AttentionBoost: Learning What to Attend for Gland Segmentation in Histopathological Images by Boosting Fully Convolutional Networks

Gozde Nur Gunesli, Cenk Sokmensuer, and Cigdem Gunduz-Demir, *Member, IEEE*

***Abstract*—Fully convolutional networks (FCNs) are widely used for instance segmentation. One important challenge is to sufficiently train these networks to yield good generalizations for hard-to-learn pixels, correct prediction of which may greatly affect the success. A typical group of such hard-to-learn pixels are boundaries between instances. Many studies have developed strategies to pay more attention to learning these boundary pixels. They include designing multi-task networks with an additional task of boundary prediction and increasing the weights of boundary pixels' predictions in the loss function. Such strategies require defining what to attend beforehand and incorporating this defined attention to the learning model. However, there may exist other groups of hard-to-learn pixels and manually defining and incorporating the appropriate attention for each group may not be feasible. In order to provide an adaptable solution to learn different groups of hard-to-learn pixels, this article proposes *AttentionBoost*, which is a new multi-attention learning model based on adaptive boosting, for the task of gland instance segmentation in histopathological images. *AttentionBoost* designs a multi-stage network and introduces a new loss adjustment mechanism for an FCN to adaptively learn what to attend at each stage directly on image data without necessitating any prior definition. This mechanism modulates the attention of each stage to correct the mistakes of previous stages, by adjusting the loss weight of each pixel prediction separately with respect to how accurate the previous stages are on this pixel. Working on histopathological images of colon tissues, our experiments demonstrate that the proposed *AttentionBoost* model improves the results of gland segmentation compared to its counterparts.**

***Index Terms*—Deep learning, attention learning, adaptive boosting, gland instance segmentation, instance segmentation.**

Manuscript received June 3, 2020; revised July 14, 2020; accepted August 3, 2020. Date of publication August 11, 2020; date of current version November 30, 2020. This work was supported by the Scientific and Technological Research Council of Turkey under Project TÜBİTAK 116E075. *(Corresponding author: Cigdem Gunduz-Demir.)*

Gozde Nur Gunesli is with the Department of Computer Engineering, Bilkent University, 06800 Ankara, Turkey (e-mail: nur.gunesli@bilkent.edu.tr).

Cenk Sokmensuer is with the Department of Pathology, Medical School, Hacettepe University, 06100 Ankara, Turkey (e-mail: csokmens@hacettepe.edu.tr).

Cigdem Gunduz-Demir is with the Department of Computer Engineering and Neuroscience Graduate Program, Bilkent University, 06800 Ankara, Turkey (e-mail: gunduz@cs.bilkent.edu.tr).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMI.2020.3015198

I. Introduction

Convolutional neural networks have shown a huge success on various image classification and object detection tasks [1], [2]. For segmentation, fully convolutional networks (FCNs) have provided significant improvements in terms of both efficiency and accuracy, approaching segmentation as a dense prediction task which predicts a label for each image pixel [3]. Thus, FCNs have become a popular choice also for instance segmentation in medical images [4]. In spite of the success of FCNs trained on very large datasets, training may become difficult when small quantities of annotated data are available and when pixels of background and foreground classes are highly imbalanced. In these cases, without further adjustments, the networks tend to yield poor generalizations for pixels of a minority class as well as for hard-to-learn pixels.

The most common approach to mitigate the class-imbalance problem is to increase the relative weight of minority class predictions in a loss function. Although this approach forces the network to give more importance to learning a minority class, it may not increase the performance for hard-to-learn pixels when these pixels occur in both majority and minority classes and when they distribute unevenly in a particular class. For example, for gland instance segmentation, it is harder to learn pixels close to gland boundaries, regardless of whether they belong to the gland or the background class. Furthermore, although the number of such hard-to-learn pixels (and as a result, the total weight contribution of their predictions in the loss function) is relatively low, their correct classification greatly affects the success of the entire task since boundary pixels separate multiple gland instances from each other.

To address this problem, it has been proposed to pay more attention to the classification of boundary pixels. One proposed solution is to adjust the weights of these pixels in the loss function based on their distances to the boundary of closest gland instances [5]. The other solution is to design a multi-task network with an additional task of boundary prediction. This task is learned together with the main task of gland instance segmentation and the glands are located at the end by employing both of the predicted maps [6]–[8]. The multi-task network proposed by [9] also includes one more additional task to predict the bounding boxes of gland instances. Both of these solutions help better classify the boundary pixels since they pay more attention to decreasing

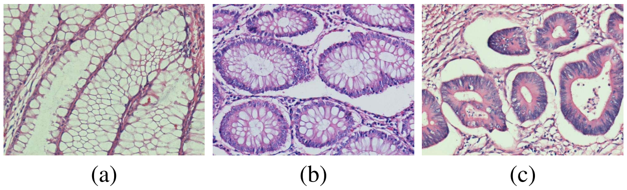

Fig. 1. Examples of histopathological images of colon glands. For gland instance segmentation, it is more difficult to correctly classify boundary pixels when two glands are very close to each other. The image shown in (a) contains such kind of glands. Additionally, these images typically contain noise and artifacts due to the tissue preparation procedures. For example, due to the density difference between glands and connective tissues (inside and outside of a gland), the fixation and sectioning procedures may result in large white artifacts outside the glands. The images given in (b) and (c) contain such kind of artifacts. It is common for algorithms to identify some of these large white artifacts as false glands. These are the images consisting of (a)-(b) normal glands and (c) cancerous glands.

mistakes that their network would make on these pixels. This attention is defined to alleviate one single mistake type related to one group of hard-to-learn pixels, namely incorrect boundary classification, and this mistake type needs to be manually identified before designing and learning the network. This manual identification is indeed a natural choice for gland instance segmentation since multiple gland instances may seem touching in histopathological images due to their nature. On the other hand, there may exist other groups of hard-to-learn pixels, and thus, other types of mistakes related to these pixels (see Fig. 1). In order for these solutions to be adaptable to additional mistake types, either new weight adjustments or new additional tasks should be defined for each mistake type separately. Nevertheless, this should be done externally and manually, which might be challenging especially when these mistakes are not related to the nature of images but to noise and artifacts. As shown in Fig. 1, histopathological images typically contain such noise and artifacts due to the tissue preparation procedures. Note that it has also been proposed to define lumen segmentation as another additional task in the multi-task architecture [8], [10]. However, this new task requires extra efforts for annotating lumen structures.

In response to these issues, this article introduces an iterative attention learning model based on adaptive boosting for gland instance segmentation.[1] This model, which we call *AttentionBoost*, proposes to learn multiple attentions directly on image data at the same time as it learns the network weights. To this end, *AttentionBoost* designs a multi-stage system that contains a fully convolutional network at each stage. Then, it proposes to modulate the attention of each network for each training image, based on pixel-wise errors of the previous stage networks, by introducing a new loss adjustment method for fully convolutional networks. This method is inspired by the Adaboost algorithm [11] and adjusts the loss weight of each pixel prediction separately with respect to how confident the previous stage networks are on their correct/incorrect predictions for the pixel. This adjustment enables the proposed *AttentionBoost* model to assign different attention levels to different pixels of the same image, according to the difficulty level of learning these pixels, as well as to adaptively select/learn what image parts (e.g., gland boundaries and artifacts) need more attention during network training. This also forces the next stages to pay more attention to learning the pixels incorrectly segmented by the previous stage networks. With this adaptive loss adjustment, *AttentionBoost* end-to-end trains its multi-stage network and combines the outputs of all stages to obtain final segmentation. For the gland segmentation task, our experiments demonstrate that this type of learning improves results not only for boundary pixels but also for other hard-to-learn pixels, mostly corresponding to false positives emerged as a result of noise and artifacts. Additionally, this work explores the possibility of using *AttentionBoost* for another instance segmentation task. Our experiments on the task of nucleus segmentation in fluorescence microscopy images show that it has the potential of increasing the segmentation performance for other tasks.

It is worth to noting that the attention mechanism was originally developed for recurrent models to give them the ability of focusing on (seeing) the relevant parts of an image for a sequential decision task [12], [13]. They can change what they see over time by adaptively weighing the contributions of different features based on past information and demand of the task at the current time. This existing attention mechanism is in the form of learning weights for feature contributions. Additionally, attention networks have been proposed to emphasize informative features and suppress less useful ones using the global distribution of channel-wise feature responses [14], [15]. This idea is also used to effectively combine 3D context information in a 2D network [16]. All these networks are different than the proposed multi-stage *AttentionBoost* model, which can change what image part each stage network needs to attend more in its training by adaptively weighing the contributions of pixel predictions in the loss function.

II. Related Work

The proposed *AttentionBoost* model mainly differs from the related networks in the following aspects: The literature contains single attention models that externally define what to attend before network training starts [5]–[7]. These attention points are manually determined as boundary pixels, assuming that these pixels are hard to learn. On the other hand, *AttentionBoost* is an error-driven multi-attention model and adaptively learns what to attend directly on image data without making any prior assumption.

AttentionBoost is also different than the iterative methods that have been proposed to correct the mistakes of a single model and refine its results. The basic idea of these methods is to decompose a segmentation task into iterative stages where image features are learned together with high-level context features from the previous map to improve the result of the next stage [17]–[20]. These methods iteratively give the next stage an image and a label map predicted by the previous stage. Their initial map is either a null label map [18] or

[1] Gland instance segmentation is a binary segmentation task where the aim is to partition an image into its glands (foreground objects) and background. in histopathological images.

a segmentation map obtained from another model [19], [20] and their final output is the label map predicted at the last stage. As opposed to our model, they learn the same task and use the same objective (loss) function at every stage, which does not explicitly force a network to change its attention to learning incorrectly segmented pixels but expects the network to implicitly learn how to correct them. On the other hand, although *AttentionBoost* uses the same task definition at all stages, since it adaptively changes the objective function from one stage to another, it can be considered that *AttentionBoost* learns a different subtask at each of these stages.

The literature consists of studies that use different weight contributions in their loss function. However, almost all these studies address the class-imbalance problem. They calculate a constant weight for each class, typically inversely proportional to its pixel frequency, and use this constant weight for all predictions of pixels belonging to the same class [21]–[23]. Different than these studies, *AttentionBoost* can assign different weights for predictions of pixels belonging to the same class by learning them directly on data. There exists a single study that attempts to learn loss weights on image data for object detection [24]. However, this previous study neither constructs multiple networks nor trains them iteratively, but it rather focuses on training a single-stage network. Each training epoch updates the loss weight for each object separately and the next epoch uses the same updated weight for all pixels of the same object. Such an approach may increase importance of learning misdetected and harder-to-learn objects in later epochs. However, since the use of a single network requires using the same network weights for all object types and since the common type of (in)correctly detected objects may still dominate the loss function, this makes harder to focus on multiple detection subtasks with different difficulty levels simultaneously. On the other hand, *AttentionBoost* defines a multi-stage network, each stage of which can use a different loss function. This allows each stage to focus on a different aspect of the task. Additionally, as opposed to *AttentionBoost*, in [24], the same loss weight is used for all pixels of the same object without considering their pixel-wise contributions.

There also exist studies that combine the Adaboost algorithm [11] with a neural network architecture [25]–[27]. However, these studies do not involve a dense prediction task and do not use an FCN, but they rather focus on image classification. To construct different learners, they either arrange different training sets or use different loss weights for training instances. These classification models have been designed for a non-dense prediction task and are beyond the scope of this article. This article uses the idea of adjusting loss weights of pixel-wise predictions for a dense prediction task.

III. Methodology

The *AttentionBoost* model proposes to train a multi-stage network that adjusts (learns) the attention of each stage automatically and to combine the outputs of all these stages for obtaining final segmentation. To this end, it introduces an attention learning mechanism for fully convolutional networks. This mechanism relies on devising a new loss adjustment method, in which the loss contribution of each pixel prediction at each stage is adjusted depending on the confidence levels of the correct/incorrect predictions of the previous stages.

The motivation behind designing such a multi-stage network is as follows: A network is trained to optimize its objective function, and thus, the definition of this function greatly affects its outputs. When there exist imbalanced data distributions and when all data points contribute to the objective function evenly, the network is biased to learning the most common patterns in the data. In this case, learning less common patterns will require adjustments in the objective function. However, making adjustments for many different patterns may not be that easy for a model that trains a single network with a single objective function. On the other hand, when the model allows training multiple (sub)networks that may use different objective (loss) functions, it is easier to make such adjustments since this gives the model an opportunity to modulate each network's attention to a different goal.

With this motivation, this article designs a multi-stage network architecture for gland segmentation in histopathological images. Each stage of this architecture trains a fully convolutional network with a different loss function. To do so, it iteratively inputs an image and a probability map estimated by the previous stage, adjusts its loss function according to this probability map, and outputs a new probability map for the next stage. This architecture is illustrated in Fig. 2. The source codes are available at www.cs.bilkent.edu.tr/~gunduz/downloads/AttentionBoost.

A. Attention Learning

Let I be an image in the training set $\mathcal{D}$, p be a pixel in training image I, and $y(p)$ be the ground truth for pixel p. Here $y(p) = 1$ if the pixel belongs to a gland instance and $y(p) = 0$ otherwise. Then, the loss function $\mathcal{L}_n$ for the n-th stage network is defined as

$$\mathcal{L}_n = \sum_{I \in \mathcal{D}} \sum_{p \in I} C_n(p) \cdot \Big(y(p) - \hat{y}_n(p)\Big)^2 \tag{1}$$

where $\hat{y}_n(p)$ is the gland probability for pixel p estimated by the n-th stage network and $C_n(p)$ is the contribution of this pixel prediction in loss function $\mathcal{L}_n$. The proposed attention learning mechanism iteratively learns contributions $C_n(p)$, for each pixel p and for each stage n, simultaneously with learning the network weights by backpropagation. In particular, this mechanism decreases loss contributions for correctly estimated pixels and increases them for incorrectly estimated ones in the framework of adaptive boosting.

To this end, it defines coefficients $\beta_n(p)$ that control how much to update the current loss contribution $C_n(p)$ for the next stage. These coefficients are used to calculate $C_{n+1}(p)$. In this work, initial loss contributions $C_0(p)$ are selected with respect to class pixel frequencies. Note that one may also select $C_0(p)$ the same for all pixels.

$$C_{n+1}(p) = \beta_n(p) \cdot C_n(p) \tag{2}$$

$$\beta_n(p) = \begin{cases} 1 - |\hat{y}_n(p) - 0.5| & \text{if } \hat{y}_n(p) \text{ is correct} \\ 1 + |\hat{y}_n(p) - 0.5| & \text{if } \hat{y}_n(p) \text{ is incorrect} \end{cases} \tag{3}$$

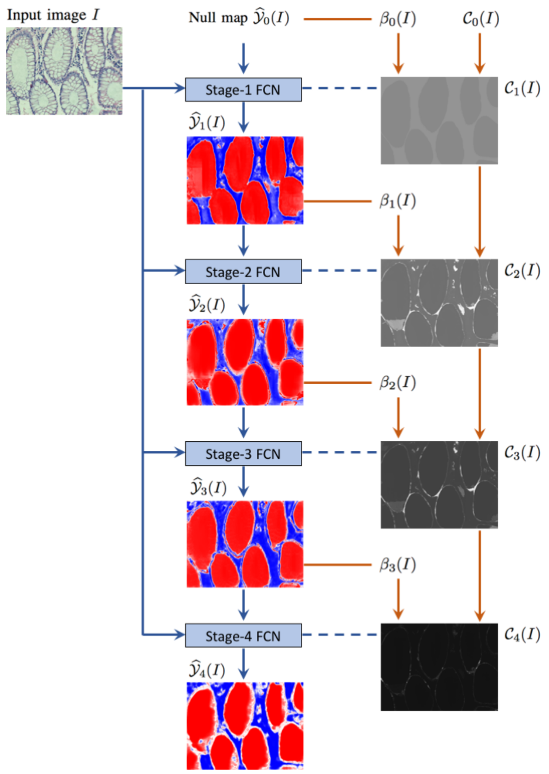

Fig. 2. An overview of the proposed multi-stage network architecture. The n-th stage network inputs an original image I and a probability map $\widehat{\mathcal{Y}}_{n-1}(I)$ estimated by the previous stage and outputs a new probability map $\widehat{\mathcal{Y}}_n(I)$ for the next stage. In training, the loss contribution map $\mathcal{C}_n(I)$ for the n-th stage is modulated by $\widehat{\mathcal{Y}}_{n-1}(I)$ and $\mathcal{C}_{n-1}(I)$ using Eqns. 2 and 3. To illustrate how this network iteratively corrects its errors for an unseen image, this figure shows the posterior maps $\widehat{\mathcal{Y}}_n(I)$ and loss contribution maps $\mathcal{C}_n(I)$ calculated for a test image. Note that the maps $\mathcal{C}_n(I)$ of this test image are calculated just for demonstration since these maps are only calculated for training images during network training. In the illustration of the contribution maps, the whiter the color of a pixel is, the higher it contributes to the corresponding loss function. The posterior maps include the probability of each pixel belonging to a gland instance. Posteriors between 1 and 0.5 are shown with increasing tints of red and posteriors between 0 and 0.5 are shown with increasing tints of blue; posteriors close to 0.5 seem whitish.

The $|\hat{y}_n(p) - 0.5|$ term in Eqn. 3 quantifies how confident the n-th stage network is on its estimation for pixel p. Since $0 \leq |\hat{y}_n(p) - 0.5| \leq 0.5$, $\beta_n(p)$ will converge to its minimum value of 0.5 if the current network correctly estimates p and if it is very confident on this correct estimation. In this case, loss contribution $C_{n+1}(p)$ becomes smaller, which forces the next stage network to decrease its attention to learning this pixel p. On the other hand, if the current network incorrectly estimates p but if it is very confident on this incorrect estimation, $\beta_n(p)$ will converge to its maximum value of 1.5. This time, loss contribution $C_{n+1}(p)$ becomes larger, which forces the next stage network to increase its attention to learning p. Thus, coefficients $\beta_n(p)$, which are calculated based on the estimations of the current stage network, are used to modulate the attention of the next stage network.

After calculating loss contributions $C_{n+1}(p)$ using Eqn. 3, they are normalized for correctly estimated pixels of a training image I and its incorrectly estimated pixels separately, such that $\sum C_{n+1}(p) = 1$ for all correctly estimated pixels $p \in I$ and $\sum C_{n+1}(q) = 1$ for all incorrectly estimated pixels $q \in I$.

B. Base Model for Each Stage

This work uses the FCN architecture given in Fig. 3 at all of its stages. The FCN at the n-th stage inputs a normalized RGB image I and the probability map $\widehat{\mathcal{Y}}_{n-1}(I) = \{\hat{y}_{n-1}(p)\}_{p\in I}$ that is estimated for this image by the previous stage network and outputs the probability map $\widehat{\mathcal{Y}}_n(I) = \{\hat{y}_n(p)\}_{p\in I}$. In order to employ the same base model at all stages, a null map is used for $\widehat{\mathcal{Y}}_0(I)$ where $\hat{y}_0(p) = 0.5$ for all pixels.

This FCN architecture consists of an encoder and a decoder path that are connected by symmetric connections. It is similar to the one proposed in [5] where extra dropout layers are added to reduce overfitting. It has convolution layers with 3×3 filters and pooling/upsampling layers with 2×2 filters. It uses the sigmoid activation function at its last layer and the ReLu activation function elsewhere. Note that by using this architecture, our multi-stage network is fit in the memory of a GPU during end-to-end training of its four networks.

C. Multi-Stage Network Training

Normalized RGB images I in the training set $\mathcal{D}$ and their ground truth maps $\mathcal{Y}(I) = \{y(p)\}_{p\in I}$ are fed to the network and the overall multi-stage network is trained with a multi-task objective function $\mathcal{L}_{sum}$, which is the sum of all loss functions, in an end-to-end manner using backpropagation. At each epoch, the forward pass calculates loss contributions $\mathcal{C}_n(I) = \{C_n(p)\}_{p\in I}$ for each image I from the first stage to the last one iteratively, as described in Sec. III-A. Then, loss functions $\mathcal{L}_n$ are updated according to the calculated loss contributions and the backward pass updates the weights of all stage networks at once by differentiating the updated loss functions. Note that weight updates for the n-th stage network are mainly affected by the loss function $\mathcal{L}_n$ defined for this stage. However, since its estimated posteriors $\widehat{\mathcal{Y}}_n$ are the inputs of the next stage network, these weight updates are also affected by the loss functions of later stages, but with a smaller extent as gradients start vanishing while backpropagating these later stage loss functions. The training procedure is illustrated in the supplementary material [29].

D. Gland Instance Segmentation

After training its multi-stage network, for image I, the *AttentionBoost* model averages the probability maps estimated by all stages and uses a seed-controlled region growing algorithm on the average map $\widehat{\mathcal{Y}}_{avg}(I) = \{\hat{y}_{avg}(p)\}_{p\in I}$ to locate gland instances. This algorithm identifies gland and background seed regions that are to be grown as follows: It first assigns each pixel p to a label $l(p)$ based on its average probability $\hat{y}_{avg}(p)$ and a confidence parameter α.

$$l(p) = \begin{cases} \text{gland} & \text{if } \hat{y}_{avg}(p) \geq 0.5 + \alpha \\ \text{background} & \text{if } \hat{y}_{avg}(p) \leq 0.5 - \alpha \\ \text{uncertain} & \text{otherwise} \end{cases} \quad (4)$$

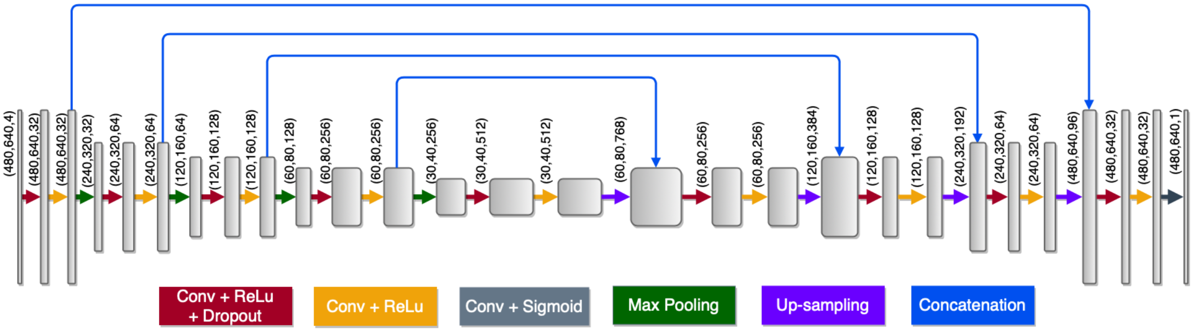

Fig. 3. Architecture of the FCN used as the base model. Each box represents a feature map with its dimensions and number of channels being indicated in order on its right. Each arrow corresponds to an operation which is distinguishable by its color.

It then finds connected components of the gland pixels and the background pixels, separately, and identifies the components larger than an area threshold A_{thr} as the seed regions. It relabels the pixels of eliminated small components with the uncertain class. At the end, it grows the seed regions onto the uncertain pixels with respect to their average probabilities. That is, it grows the seeds pixel by pixel starting from the least uncertain pixel to the most uncertain one. Note that for a gland seed region, the least uncertain pixel is the one with the highest $\hat{y}_{avg}(p)$ and for a background seed region, it is the one with the lowest $\hat{y}_{avg}(p)$. Each grown gland seed region is considered as a gland instance in the final segmentation map. The boundaries of these gland instances are smoothed by majority filtering with a filter size $f_{size} = 3$.

IV. Experiments

A. Dataset

We test our model on a dataset of 200 microscopic images of colon biopsy samples obtained from the Pathology Department Archives of Hacettepe University School of Medicine. These samples are hematoxylin-and-eosin stained tissue sections containing normal and cancerous (colon adenocarcinomatous) glands. Their images are taken using a Nikon Coolscope Digital Microscope with a 20× objective lens. The image resolution is 480 × 640.

The dataset is divided into training and test sets (Table I). The training set is further split into training images, on which the backpropagation algorithm learns the network weights, and validation images, which are used for early stopping. The training and validation images are also employed to select the parameters of the gland instance segmentation step. All networks are trained for five times, using different training and validation images. For that, the training set is split into five folds. For each run, images in four folds are used as the training images (for learning the network weights) and those in the remaining fold are used as the validation images (for early stopping). Afterwards, using each of these five trained networks, our model as well as the comparison methods and ablation studies are evaluated on the test set.

B. Implementation Details

The multi-stage network is implemented in Python using the Keras deep learning framework. The network is trained on a GPU (GeForce GTX 1080 Ti). It is trained from scratch with an early stopping approach based on the loss calculated for the validation images. The batch size is 1 and the drop-out factor is 0.2. The learning rate and the momentum value are adaptively adjusted using the AdaDelta optimizer.

TABLE I
Number of Images and Glands in the Training and Test Sets

	Number of images		Number of glands	
	Training	Test	Training	Test
Normal	50	50	744	621
Cancerous	50	50	370	367
Total	100	100	1114	988

C. Evaluation

Results are quantitatively assessed using three criteria: 1) the object-level F-score to assess what percentage of gland objects (instances) are detected correctly, 2) the object-level Dice index to assess how accurately the pixels of the segmented gland objects overlap with those of their matching (maximally overlapping) ground truth objects, and 3) the object-level Hausdorff distance to assess the shape similarity between the segmented gland objects and their matching ground truth objects. Note that these measures were also used in the GlaS Challenge Contest [28]. The detailed definitions of these measures are given in the supplementary material [29].

D. Parameter Selection

AttentionBoost uses two external parameters in its gland instance segmentation step. These are the confidence parameter α to identify certain pixels for region growing and the area threshold A_{thr} to eliminate small regions. The grid search on the training and validation images is used to select their values. The test images are not used in this selection at all. For that, all combinations of $\alpha = \{0.05, 0.10, 0.15, 0.20, 0.25\}$ and $A_{thr} = \{250, 500, 750, 1000\}$ are considered and the one that yields the highest Dice index for the first trained network (run for the first fold) is selected. The same set of parameters is used for the other trained networks (runs for the other four folds). The selected values are $\alpha = 0.15$ and $A_{thr} = 250$. The effects of this parameter selection to the model's performance are further analyzed in the supplementary material [29]. Note that the same procedure is used to select the parameters of the comparison methods and the ablation studies.

TABLE II

QUANTITATIVE RESULTS OF *AttentionBoost* AND THE COMPARISON METHODS. THESE ARE THE AVERAGE OF THE TEST SET RESULTS OBTAINED AT FIVE DIFFERENT RUNS (FOLDS) TOGETHER WITH THEIR STANDARD DEVIATIONS

	Normal glands			Cancerous glands			All glands		
	F-score	Dice	Hausdorff	F-score	Dice	Hausdorff	F-score	Dice	Hausdorff
AttentionBoost	96.2 ± 0.5	95.0 ± 0.3	23.3 ± 2.1	90.4 ± 0.8	91.7 ± 0.5	46.5 ± 2.3	94.0 ± 0.2	93.4 ± 0.1	34.7 ± 0.9
Boundary-loss-adjustment	92.8 ± 1.2	90.6 ± 1.1	50.3 ± 6.8	89.3 ± 0.8	90.2 ± 0.5	57.9 ± 4.9	91.5 ± 0.7	90.4 ± 0.6	54.0 ± 3.5
Multi-task	95.1 ± 0.7	93.7 ± 1.0	31.0 ± 5.6	87.4 ± 1.3	90.7 ± 0.6	51.1 ± 3.6	92.2 ± 0.7	92.2 ± 0.3	40.9 ± 2.4
Iterative	89.5 ± 1.3	85.8 ± 1.3	77.5 ± 8.3	90.3 ± 0.5	91.6 ± 0.5	49.9 ± 3.6	89.8 ± 0.8	88.6 ± 0.9	64.0 ± 5.7

E. Comparisons

We compare our model with three approaches implemented based on the previously reported FCNs [5], [6], [18]. The first two, the *boundary-loss-adjustment* and *multi-task* methods, are single-stage models that pay more attention to predicting gland boundaries. However, as opposed to our model, these methods require a prior definition of what to attend and include this definition in their system design. The last one is a multi-stage *iterative* method, each stage of which also inputs an image and a segmentation (probability) map from the previous stage and outputs another segmentation map for the next stage. However, different than our model, it always uses the same loss function at all stages. It neither explicitly forces its network to modulate its attention to learning incorrectly predicted pixels nor employs adaptive boosting for this purpose. We use this *iterative* method to understand the effectiveness of using adaptive boosting for a dense prediction task. The details are given below. Note that all these methods use the same network architecture (Fig. 3) as their base models. However, for fair comparisons, we keep the number of their parameters (network weights) on par with ours by selecting an appropriate number of feature maps in their first convolutional layers. This selection will affect the number of feature maps in the other convolutional layers, since this base model uses the U-Net architecture, which doubles the number of feature maps after each pooling layer and halves it after each upsampling layer. For these comparison methods as well as the ablation studies, the number of the feature maps and the number of parameters are provided in the supplementary material [29].

1) Boundary-Loss-Adjustment Method: It pays attention to more correctly predicting pixels close to boundaries. Thus, it increases the loss contributions of such pixels. For that, it uses a U-Net model that adjusts loss contributions of all pixels based on their distances to the boundary of closest gland instances [5]. Pixels predicted as gland by this trained network typically form undersegmented components for multiple gland instances that are close to each other; some of them are connected to each other by narrow bridges. Thus, to improve the results of this method, the gland pixels are postprocessed as follows: They are eroded by a disk structuring element, eroded components smaller than an area threshold are eliminated, and the remaining components are separately dilated by using the same structuring element.

2) Multi-Task Method: It pays more attention to learning boundary pixels by designing a multi-task architecture, similar to the DCAN model [6]. This architecture defines an additional task for boundary prediction and concurrently learns it together with the segmentation task. After training, this method locates gland instances by subtracting the predicted boundary pixels from the predicted gland pixels and applying postprocessing that finds large connected components on the subtracted map and dilates them with a disk structuring element.

TABLE III

AVERAGE NUMBER OF THE TYPES OF MISTAKES THAT *AttentionBoost* AND THE COMPARISON METHODS MAKE ON THE TEST IMAGES TOGETHER WITH THEIR STANDARD DEVIATIONS

	Undersegm ground truths	False segmented objects	Small oversegm objects	Missing ground truths
AttentionBoost	53.0 ± 6.1	11.4 ± 3.4	36.6 ± 6.8	40.2 ± 2.0
Boundary-loss-adj	157.4 ± 12.3	22.8 ± 1.8	16.8 ± 2.8	33.6 ± 5.3
Multi-task	88.0 ± 14.1	51.4 ± 25.8	22.0 ± 13.5	33.0 ± 7.0
Iterative	207.0 ± 23.9	17.6 ± 5.1	18.2 ± 7.6	27.6 ± 7.4

3) Iterative Method: It uses the same multi-stage network of *AttentionBoost* and iteratively trains this network as proposed in [18]. However, it uses the same loss function at all of its stages and does not use adaptive boosting at all. After its training, the segmentation maps produced by its different stages are aggregated and postprocessed using the same steps of the proposed *AttentionBoost* model.

V. RESULTS

Table II reports the quantitative results of the proposed *AttentionBoost* model and the comparison methods. These are the average of the test set results obtained at five different runs (folds) together with their standard deviations. This table also presents the results for the segmentation of normal and cancerous glands separately. These results show that *AttentionBoost* is more successful at detecting and segmenting gland instances (higher F-score and Dice index values) as well as it yields more accurate gland shapes (lower Hausdorff distances). This is attributed to the ability of our model to automatically learn what to attend in images and also to focus on different types of mistakes. To explore this further, we examine the following types of mistakes the methods make in their segmentations, visually (Fig. 4) and quantitatively (Table III).

(i) Undersegmented ground truth objects: Let S and G be sets of segmented gland objects and ground truth objects, respectively. A ground truth object $g \in G$ is considered as undersegmented if a segmented gland object $s \in S$ intersects with at least 50 percent of g but also intersects with at least 50 percent of another ground truth object $g' \in G$. This mistake type commonly occurs when a method cannot correctly predict the labels of pixels close to gland boundaries. This is the mistake type that most of the previous methods have attempted to solve by either adjusting the weights of boundary pixels in

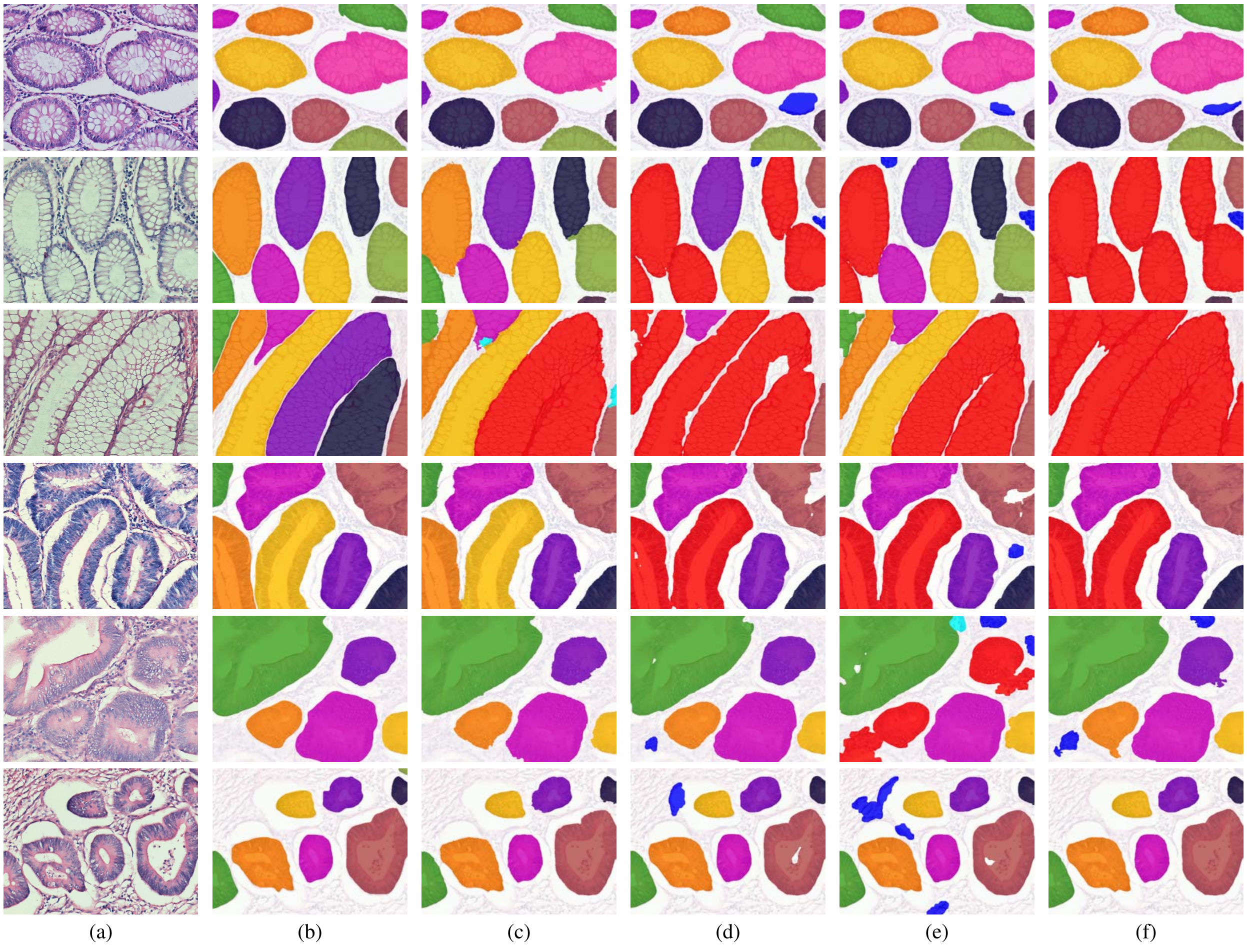

Fig. 4. (a) Example test set images containing normal (first three rows) and cancerous (last three rows) glands. (b) Ground truths. Results of the (c) *AttentionBoost*, (d) *boundary-loss-adjustment*, (e) *multi-task*, and (f) *iterative* methods. To better emphasize differences, different colors are used for different types of mistakes in these segmentation results: red for all undersegmented glands, blue for all false positives, and cyan for all small oversegmented objects. Note that these visual results are obtained for the first fold (using the first trained network); the visual results for the other folds show similar characteristics.

the loss function [5] or defining boundary prediction as an additional task in a multi-task architecture [6], [7].

(ii) False positives: A segmented gland object $s \in S$ is considered as false positive if it does not intersect with at least 50 percent of any $g \in G$. In our experiments, we observe this mistake type due to two main reasons. The first one is segmenting non-gland regions as gland objects. These non-gland regions are typically located around white artifacts, which are usually formed as a result of the tissue preparation procedures. Such an example can be seen in the first row of Fig. 4(d). The second reason is oversegmenting small objects in a gland, usually close to its boundary. Two such small oversegmented objects can be seen in the third row of Fig. 4(c). To distinguish these two types of false positives, we call $s \in S$ a false segmented object if it does not intersect with at least 50 percent of any $g \in G$ and if any $g' \in G$ does not intersect with at least 50 percent of s. On the other hand, we call it a small oversegmented object, if it does not intersect with at least 50 percent of any $g \in G$ but if a ground truth object $g' \in G$ intersects with at least 50 percent of s.

(iii) False negatives: A ground truth object $g \in G$ is considered as false negative (missing object) if at least its 50 percent does not intersect with any $s \in S$.

The average number of the mistake types for the test set are reported in Table III and visual results obtained on exemplary test images are provided in Fig. 4. They demonstrate that *AttentionBoost* leads to the best results both for undersegmentations, which emerge as a result of misclassifying boundary pixels, and for false segmented objects, which are dislocated due to not differentiating true gland pixels from those of non-gland regions mostly containing noise and artifacts. These are the two most common mistake types for this gland instance segmentation task and our proposed model improves results for both at the same time, in contrast to its counterparts, which are good at either one mistake type or the other.

The *iterative* method, which is also a multi-stage model but uses the same loss function at all of its stages, is successful to eliminate false positives. However, it cannot sufficiently improve boundary pixel prediction throughout its stages, which leads to a significantly higher number of undersegmentations. This suggests the benefits of automatically adjusting the loss functions of consecutive stages via adaptive boosting. The *boundary-loss-adjustment* method improves the results of the *iterative* method. However, it still leads to many undersegmentations due to incorrectly segmenting pixels in between adjacent gland objects. The *multi-task* method gives

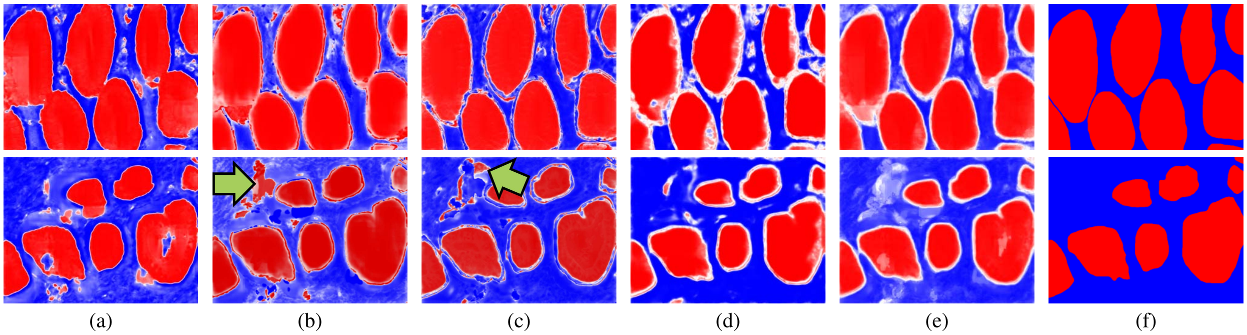

Fig. 5. (a) Posterior map $\hat{\mathcal{Y}}_1(I)$ generated by the first stage. (b) Posterior map $\hat{\mathcal{Y}}_2(I)$ generated by the second stage. (c) Posterior map $\hat{\mathcal{Y}}_3(I)$ generated by the third stage. (d) Posterior map $\hat{\mathcal{Y}}_4(I)$ generated by the fourth stage. (e) Average posterior map $\hat{\mathcal{Y}}_{avg}(I)$ obtained by aggregating the posterior maps of all stages. (f) Posterior map $\mathcal{Y}(I)$ produced by the ground truth segmentation. These maps include pixel posteriors where 1 indicates that a pixel belongs to the gland class and 0 indicates that it belongs to the background. Posteriors between 1 and 0.5 are shown with increasing tints of red and posteriors between 0 and 0.5 are shown with increasing tints of blue. Note that in these images posteriors close to 0.5 seem whitish. This figure points to the complementariness of the estimated maps. For example, the red regions shown with the arrows in (b) and (c) include incorrectly estimated pixels. However, most of these pixels are incorrectly estimated in either one map or the other. Thus, when all maps are averaged, the resulting map does not contain these incorrectly estimated pixels.

relatively better results for undersegmentations. On the other hand, this method is effective for this specific mistake type at the expense of locating more false positives, as also seen in Fig 4(e). This indicates the effectiveness of learning multiple attentions directly on image data instead of externally defining a specific attention type beforehand. Note that *AttentionBoost* misses slightly more ground truth objects. However, in our experiments, we observe that most of them correspond to small ground truth objects close to image edges. The one at the upper-right corner of the image shown in the last row of Fig. 4(b) is an example of such small objects.

The improvement in the results is attributed to the following: *AttentionBoost* is a multi-stage and an error-driven multi-attention learning model, each stage of which is able to pay a different level of attention to learning different parts (pixels) of an image. This enables each stage to produce a segmentation (posterior) map complementary to those of the other stages. The maps of different stages are complementary on incorrect predictions, especially for hard-to-learn pixels, since it is usually quite difficult for a single network to produce correct predictions for all such pixels. By having such complementary maps, errors in one map may be compensated by another. Thus, when these maps are aggregated, it is expected to obtain more robust predictions. This can be seen in the posterior maps produced for the two exemplary test images (Fig. 5).

To explore it further, we conduct two ablation studies in which all stage networks share weights. Here it is worth to noting that both of these ablation studies still use the proposed attention mechanism to adjust the loss function used by their each stage network. In the first study, each stage network uses the base model given in Fig. 3. In this case, it is observed that the maps estimated by different stage networks are not complementary enough, which might be the reason of obtaining lower performance measures (Table IV). Of course, when the weights are shared, the number of parameters to be learned decreases (in our experiments, from 31,387,780 to 7,846,945), which also decreases the convergence time of network training. In our experiments, the number of epochs at the stopping time decreases from 79.4 to 32.2 on the average; their convergence plots are provided in the supplementary material [29]. However, the computational time required by each epoch remains almost the same (26-27 seconds) since the training procedure unfolds the network to be learned.

TABLE IV
ABLATION STUDY RESULTS OBTAINED ON THE TEST IMAGES

	F-score	Dice	Hausdorff
AttentionBoost	94.0 ± 0.2	93.4 ± 0.1	34.7 ± 0.9
AttentionBoost (shared weights)	91.3 ± 1.2	91.5 ± 0.7	44.9 ± 2.5
AttentionBoost (shared weights ×2)	92.7 ± 1.3	92.4 ± 0.7	40.2 ± 3.3
AttentionBoost (w/o normalization)	91.9 ± 0.7	91.7 ± 0.9	44.2 ± 4.3

The second ablation study is conducted to understand whether the performance decrease is a result of weight sharing or due to the decrease in the parameter number. For that, we double the number of feature maps in the base model, which gives a network with 31,379,521 parameters. The experiments show that although this new network improves the performance, it is still worse than the original *AttentionBoost* model, which does not use weight sharing. Additionally, the increase in the parameter number greatly increases the convergence time. The number of epochs becomes 184.8 on the average and the computational time for each epoch becomes approximately 70 seconds since the network is unfolded during training.

Another important factor that affects the complementariness of the segmentation maps is the amount of updates in loss contributions from one stage to another. In the proposed model, this is controlled by coefficients β_n, which increase the loss contributions for incorrectly segmented pixels and decrease them for correctly segmented ones. If this increase is too little compared to the decrease, the correctly segmented pixels may dominate the loss function since the number of these pixels (thus, their total contribution to the loss function) is usually high. This typically results in producing non-complementary maps. On the other hand, if the increase is too much, later stage networks may completely give up their attentions to learning the correctly segmented pixels. This may cause to produce complementary but low quality maps for these pixels at later stages, which affects the final segmentation since all maps are aggregated at the end. To balance these two factors, our model normalizes the loss contributions for correctly and incorrectly segmented pixels separately (Sec. III-A). The ablation study that does not use any normalization shows that this normalization is important to obtain better results (Table IV). Note that one can also set this trade-off by changing the definition of β_n. For example, one may use a multiplier (or an exponent) term in its definition and may select different multipliers (exponents) for correctly and incorrectly segmented pixels. This possibility can be investigated as future work.

A. Stage Number Analysis

The *AttentionBoost* model uses a multi-stage network architecture. The number of stages (base models) in this architecture determines the number of segmentation maps that the multi-stage network can produce. It is important to keep in mind that the later stage maps are produced to correct the mistakes made by the previous stage networks. The stage number should be selected considering the complexity of the task at hand. In our experiments, this number is selected as four for the gland segmentation task.

When the stage number is not selected large enough, the multi-stage network cannot produce an adequate number of maps that correct the mistakes of the previous stages; this yields lower performances. In our experiments, such kind of performance decrease is observed when the stage number is selected as two (Table V). On the contrary, when it is more than necessary, it may also negatively affect the performance, as also seen in our results when the stage number is more than five. The reason might be the following: After a certain point, at a given stage, loss contributions of many correctly segmented pixels that were also correctly segmented by many previous stage networks become relatively smaller than those of the correctly segmented pixels that were correctly segmented by only a few stage networks. Note that although the loss contributions are normalized for the correctly segmented pixels, this may not be that effective at later stages since this normalization treats all correctly segmented pixels in the same way, without considering how many times they were correctly segmented by the previous stage networks. Thus, after a certain point, the networks may generate uncertain predictions, which are close to 0.5, for such pixels. This negatively affects the performance since the segmentation maps generated by every stage are averaged at the end. This problem might be alleviated by designing more sophisticated algorithms to aggregate these maps. Such designs are considered as future work.

B. Robustness Analysis

We examine the robustness of *AttentionBoost* to pixel-level noise. For that, normal noise $\mathcal{N}(0, \sigma^2)$ is added to each pixel[2] in each image and the experiments are repeated for different values of the standard deviation σ. Note that since the noise is generated probabilistically, the experiments are repeated three times for each fold. Figure 6 depicts the average performance obtained on the test images as a function of σ. This figure shows that *AttentionBoost* is robust to pixel-level noise to a certain degree. However, as expected, the performance decreases for larger amounts of noise.

TABLE V
TEST SET RESULTS AS A FUNCTION OF THE NUMBER OF STAGES IN THE MULTI-STAGE NETWORK

No of stages	F-score	Dice	Hausdorff
2	92.9 ± 0.2	92.6 ± 0.2	40.0 ± 1.7
3	93.5 ± 0.4	93.1 ± 0.4	36.7 ± 3.9
4	94.0 ± 0.2	93.4 ± 0.1	34.7 ± 0.9
5	93.7 ± 0.3	93.3 ± 0.1	35.2 ± 0.8
6	93.0 ± 0.8	92.8 ± 0.6	37.6 ± 2.8
7	91.9 ± 0.7	92.3 ± 1.0	40.3 ± 6.1

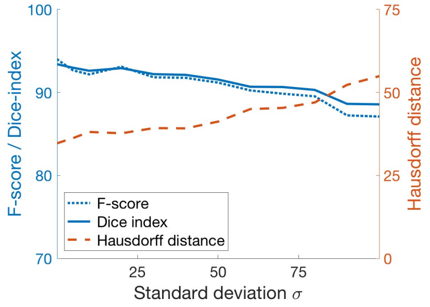

Fig. 6. Effects of pixel-level noise on performance measures. All results are obtained on the test images.

VI. DISCUSSION

A. Base Model Selection

This article introduces the multi-stage *AttentionBoost* model with the objective of increasing the learning power of a single-stage network. Each stage of *AttentionBoost* consists of a single-stage network (base model) that adaptively learns what to attend from the mistakes of its previous stage networks. In our experiments, we use a simple U-Net network as the base model (see Fig. 3) and compare *AttentionBoost* with its counterparts, which also focus on increasing the learning power of the same network but by using different learning strategies. However, the proposed method is not limited with the use of this selected base model and can be used to increase the performance of other networks. To better understand this trait of the proposed method, we compare the performance of a single-stage network, which does not use any of these learning strategies, with its corresponding *AttentionBoost* model that uses the same single-stage network as its base model.

For this purpose, three single-stage networks with different complexities are employed. The most basic one is the U-Net network, which was originally selected for our experiments. This network is the same with that of the *boundary-loss-adjustment* method but its training does not use any loss adjustment based on the closeness of pixels to gland boundaries. The second one is similar to this first network except that it uses G-convolutions instead of the standard ones in its convolution layers. These G-convolutions were proposed to be used for gland instance segmentation by the Rota-Net model [10]. In addition to using the G-convolutions, the third network has residual blocks given in [10].

[2] The value of a noisy pixel p_{noise} is set to 0 if $p_{noise} < 0$ after adding the normal noise. Likewise, it is set to 255 if $p_{noise} > 255$.

TABLE VI
COMPARISON OF SINGLE-STAGE NETWORKS AND THEIR CORRESPONDING *AttentionBoost* MODELS ON THE TEST IMAGES

	F-score	Dice	Hausdorff
Single-stage networks (two classes)			
SingleStage (U-Net)	89.1 ± 1.0	88.2 ± 1.0	66.7 ± 5.1
SingleStage (G-Conv)	91.2 ± 0.7	90.0 ± 0.9	57.6 ± 4.6
SingleStage (G-Res)	93.7 ± 0.3	92.2 ± 0.1	43.2 ± 1.2
Single-stage networks (three classes)			
SingleStage (U-Net)	92.2 ± 0.9	92.0 ± 0.3	43.0 ± 1.3
SingleStage (G-Conv)	94.0 ± 0.6	93.1 ± 0.8	39.6 ± 4.8
SingleStage (G-Res)	94.4 ± 0.5	93.7 ± 0.1	35.4 ± 0.5
Multi-stage networks			
AttentionBoost (U-Net)	94.0 ± 0.2	93.4 ± 0.1	34.7 ± 0.9
AttentionBoost (G-Conv)	95.3 ± 0.3	94.5 ± 0.3	29.2 ± 2.7
AttentionBoost (G-Res)	95.9 ± 0.3	95.1 ± 0.4	25.9 ± 1.6

These three single-stage networks are first trained for the two-class classification problem, where the classes are gland and background. Gland pixels estimated by these three networks are postprocessed by following the steps used for the *boundary-loss-adjustment* method. The test set results obtained by the single-stage networks as well as their corresponding *AttentionBoost* models are reported in Table VI. This table shows that the performance increases with the increasing complexity of a network. It also reveals that *AttentionBoost* can further increase the performance of a given network regardless of its complexity.

The Rota-Net model considers gland instance segmentation as a three-class classification problem, where the classes are inner gland, gland contour, and background [10]. Defining two separate classes for inner glands and gland contours can indeed be considered as a learning strategy to give extra emphasis to learning gland boundaries. To also understand the effectiveness of this strategy, the three aforementioned single-stage networks are trained for this three-class classification. After training the networks, their estimated maps are postprocessed by first finding large connected components of inner gland pixels and then dilating them with a disk structuring element. Note that besides this three-class classification task, Rota-Net defines an additional task of lumen segmentation in its decoder path. However, the single-stage networks do not contain this task since it requires extra annotations of lumen structures, which are not available. The results obtained by these single-stage networks are also reported in Table VI. They show that the use of an additional gland contour class increases the performance. However, the proposed *AttentionBoost* model, which still uses two classes, is more effective and gives better results.

In all these *AttentionBoost* models, the network used as the base model is identical at all stages. However, it is also possible to use different base models at different stages. To investigate this possibility, we first define three other base models whose architectures contain different numbers of layers and feature maps. These are also U-Net networks, which double the number of feature maps after a pooling layer and halves it after a upsampling layer. The architectures of these models are provided in the supplementary material [29]. We refer each base model by two numbers: the number of its pooling/upsampling layers (blocks) and the number of feature maps in its first convolution layer. For example, the base model given in Fig. 3 is referred as *4b32f*. We then implement the following two variants of the *AttentionBoost* model.

TABLE VII
TEST SET RESULTS OBTAINED WHEN DIFFERENT BASE MODELS ARE USED AT DIFFERENT STAGES OF THE *AttentionBoost* MODEL

	F-score	Dice	Hausdorff
AttentionBoost (same model: *4b32f*)	94.0 ± 0.2	93.4 ± 0.1	34.7 ± 0.9
AttentionBoost (different models: *4b32f*, *4b32f*, *4b64f*, and *4b64f*)	93.8 ± 0.7	93.3 ± 0.3	34.6 ± 0.6
AttentionBoost (different models: *3b32f*, *3b64f*, *4b32f*, and *4b64f*)	94.0 ± 0.4	93.2 ± 0.4	35.5 ± 1.6

- The first variant uses *4b32f*, *4b32f*, *4b64f*, and *4b64f* from the first stage to the fourth one.
- The second variant uses *3b32f*, *3b64f*, *4b32f*, and *4b64f* from the first stage to the fourth one.

The test set results of these variants are reported in Table VII. These results together with those in Table VI indicate that *AttentionBoost* does not require a particular network architecture for its base model(s) and can be used to improve the performance. However, its overall performance, of course, depends on the learning power of the selected base model(s).

B. Using AttentionBoost for Nucleus Instance Segmentation

In order to explore the possibility of applying it on another segmentation task, we test *AttentionBoost* on the task of cell nucleus segmentation in fluorescence microscopy images. To this end, we test it on a dataset, which is available at www.cs.bilkent.edu.tr/∼gunduz/downloads/NucleusSegData/. This dataset contains 61 images of 3329 nuclei of cells taken from the Huh7 and HepG2 liver cancer cell lines and stained with nuclear Hoechst 33258 [31], [32]. Its training set contains 1126 nuclei of 25 images (ten Huh7 and 15 HepG2 cell line images). In our experiments, we train the networks of our model and the comparison methods for five times by using a different portion of the training images for early stopping. The parameters of all methods are also selected on this training set, as explained in Sec. IV-D. The dataset has two separate test sets: the Huh7 test set contains 891 nuclei of 11 Huh7 cell line images and the HepG2 test set that contains 1312 nuclei of 25 HepG2 cell line images. Note that it is more difficult to segment nuclei in the HepG2 cell line since HepG2 cells tend to grow in more overlays than Huh7 cells, which leads to more overlapping nuclei in the images of the HepG2 cell line.

The quantitative and visual results for the test sets are presented in Table VIII and Fig. 7. They show that *AttentionBoost* gives accurate results for both of the Huh7 and HepG2 test sets. The *boundary-loss-adjustment* and *multi-task* methods, which are single-stage models that pay more attention to predicting nucleus boundaries lead to better segmentations than the *iterative* method, which is also a multi-stage model but uses the same loss function at all of its stages. In our experiments, it is observed that the *iterative* method frequently yields undersegmentations due to not correctly predicting

TABLE VIII

QUANTITATIVE RESULTS OF *AttentionBoost* AND THE COMPARISON METHODS FOR NUCLEUS SEGMENTATION IN FLUORESCENCE MICROSCOPY IMAGES. THESE ARE THE AVERAGE OF THE TEST SET RESULTS OBTAINED AT FIVE DIFFERENT RUNS TOGETHER WITH THEIR STANDARD DEVIATIONS

	Huh7 test set			HepG2 test set		
	F-score	Dice	Hausdorff	F-score	Dice	Hausdorff
AttentionBoost	95.4 ± 0.5	91.2 ± 0.7	5.8 ± 0.6	91.9 ± 0.8	87.0 ± 0.8	9.5 ± 0.7
Boundary-loss-adjustment	93.7 ± 0.6	89.2 ± 0.6	7.0 ± 0.3	88.7 ± 0.6	84.5 ± 0.7	11.4 ± 1.7
Multi-task	93.4 ± 0.9	88.3 ± 0.9	8.2 ± 1.0	86.9 ± 0.5	83.9 ± 0.7	11.5 ± 1.1
Iterative	86.4 ± 0.8	79.5 ± 0.6	18.7 ± 0.5	80.0 ± 0.9	73.9 ± 0.5	25.8 ± 0.6

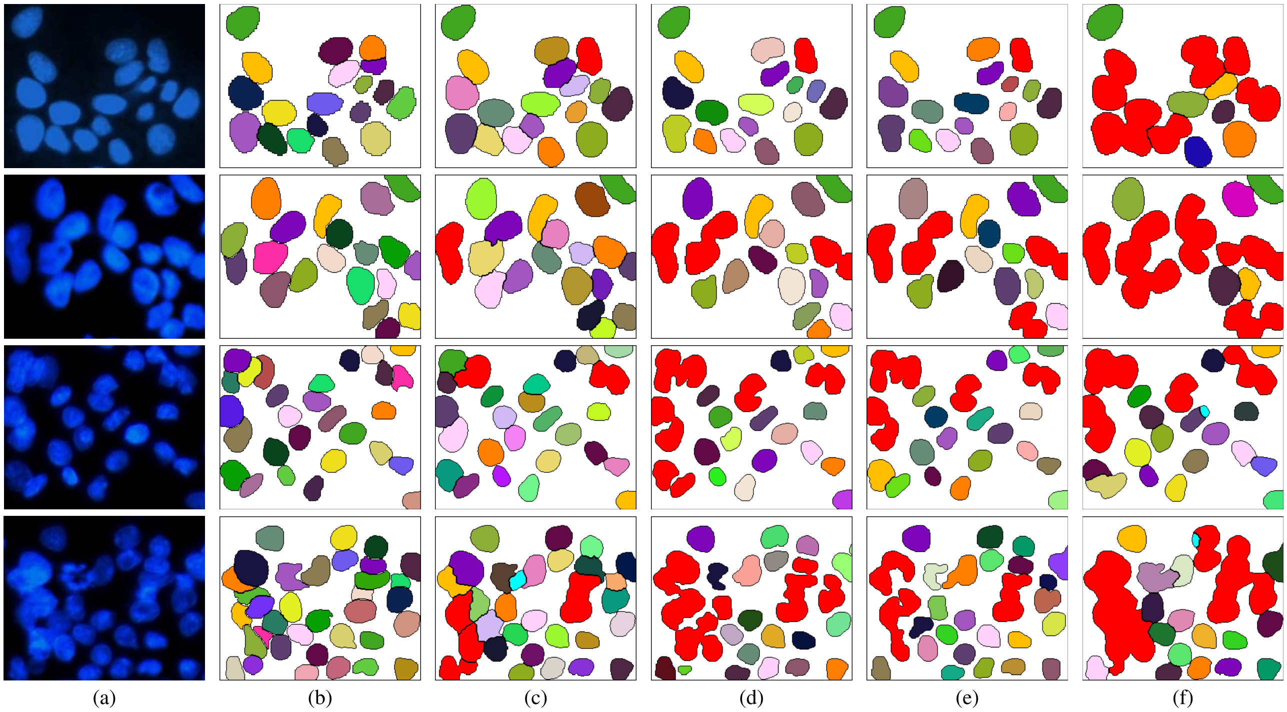

Fig. 7. (a) Example fluorescence microscopy images of the Huh7 (first two rows) and HepG2 (last two rows) cell lines. (b) Ground truths. Results of the (c) *AttentionBoost*, (d) *boundary-loss-adjustment*, (e) *multi-task*, and (f) *iterative* methods. These are subimages cropped out of the test set images. The subimage sizes are scaled for better visualization. Likewise, to better emphasize differences, different colors are used for different types of mistakes in these segmentation results: red for all undersegmented cells, blue for all false positives, and cyan for all small oversegmented objects.

boundary pixels. On the contrary, by adjusting the loss function of each stage adaptively, our proposed *AttentionBoost* model better predicts these pixels, and as a result, gives the best segmentation results for the task of nucleus segmentation in fluorescence microscopy images. This indicates the potential of using *AttentionBoost* for other instance segmentation tasks.

VII. CONCLUSION

This article presents an error-driven multi-attention learning model for gland instance segmentation. This model, which we call *AttentionBoost*, relies on designing a multi-stage network and adaptively learning what image parts (pixels) each stage needs to attend and the level of this attention directly on image data. To this end, it introduces a new loss adjustment mechanism that uses adaptive boosting for a dense prediction task for the first time. This mechanism modulates the attention of each stage to correct the mistakes of its previous stages, by adjusting the loss weight of each pixel separately according to how confident the previous stages are on their predictions for this pixel. We tested our model for the task of gland instance segmentation in histopathological images. Furthermore, we also showed the applicability of our model to the task of nucleus instance segmentation in fluorescence microscopy images. Our experiments revealed that the proposed *AttentionBoost* model, which enables to learn different attentions for different pixels at the same stage as well as to learn multiple attentions for the same pixel at different stages, leads to more accurate segmentation results compared to the existing approaches.

For an unseen image, *AttentionBoost* uses simple averaging to aggregate the posterior maps estimated by all stages of the multi-stage network. One future research direction is to investigate different ways to combine these maps. For example, one can aggregate multiple intermediate outputs by weighted average and learn the weights by another network [30]. As a preliminary study, a similar approach is followed: After training the *AttentionBoost* (U-Net) model, the final posterior of each pixel is estimated as a linear combination of those estimated by all stages. In this preliminary study, the weights in the linear combination are learned on pixels of the training and validation images using a logistic regression classifier. That is, the final posterior of pixel p is expressed as $\hat{y}_{final}(p) = sigmoid(\sum_{n=1}^{4} \omega_n \hat{y}_n(p) + \omega_0)$ and the weights ω_n as well as the bias ω_0 are learned with the gradient descent algorithm. Then, gland instances are located on image I by applying the proposed seed-controlled region growing algorithm on the new map $\widehat{\mathcal{Y}}_{final}(I) = \{\hat{y}_{final}(p)\}_{p \in I}$. The experiments reveal that this new map estimation leads to a slight increase in the performance measures; the F-score, Dice index, and Hausdorff distance become 94.14, 93.91, and 32.32, respectively. One may further increase these measures using a more complex classifier. The investigation of this use is left as future work.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[3] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[4] G. Litjens *et al.*, "A survey on deep learning in medical image analysis," *Med. Image Anal.*, vol. 42, pp. 60–88, Dec. 2017.

[5] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Intervent.*, 2015, pp. 234–241.

[6] H. Chen, X. Qi, L. Yu, Q. Dou, J. Qin, and P.-A. Heng, "DCAN: Deep contour-aware networks for object instance segmentation from histology images," *Med. Image Anal.*, vol. 36, pp. 135–146, Feb. 2017.

[7] Y. Xu *et al.*, "Gland instance segmentation by deep multichannel side supervision," in *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Intervent.*, 2016, pp. 496–504.

[8] S. Graham *et al.*, "MILD-net: Minimal information loss dilated network for gland instance segmentation in colon histology images," *Med. Image Anal.*, vol. 52, pp. 199–211, Feb. 2019.

[9] Y. Xu *et al.*, "Gland instance segmentation using deep multichannel neural networks," *IEEE Trans. Biomed. Eng.*, vol. 64, no. 12, pp. 2901–2912, Dec. 2017.

[10] S. Graham, D. Epstein, and N. Rajpoot, "Rota-Net: Rotation equivariant network for simultaneous gland and lumen segmentation in colon histology images," in *Proc. Eur. Congr. Digit. Pathol.* Cham, Switzerland: Springer, 2019, pp. 109–116.

[11] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.

[12] V. Mnih, N. Heess, and A. Graves, "Recurrent models of visual attention," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, Dec. 2014, pp. 2204–2212.

[13] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2015, pp. 2048–2057.

[14] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2011–2023, Aug. 2020.

[15] H. Sun, C. Li, B. Liu, H. Zheng, D. Dagan Feng, and S. Wang, "AUNet: Attention-guided dense-upsampling networks for breast mass segmentation in whole mammograms," 2018, *arXiv:1810.10151*. [Online]. Available: http://arxiv.org/abs/1810.10151

[16] Y. Zhou, W. Huang, P. Dong, Y. Xia, and S. Wang, "D-UNet: A dimension-fusion U shape network for chronic stroke lesion segmentation," 2019, *arXiv:1908.05104*. [Online]. Available: http://arxiv.org/abs/1908.05104

[17] Z. Tu and X. Bai, "Auto-context and its application to high-level vision tasks and 3D brain image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 10, pp. 1744–1757, Oct. 2010.

[18] K. Li, B. Hariharan, and J. Malik, "Iterative instance segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3659–3667.

[19] S. Gidaris and N. Komodakis, "Detect, replace, refine: Deep structured prediction for pixel wise labeling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5248–5257.

[20] A. Romero, M. Drozdzal, A. Erraqabi, S. Jégou, and Y. Bengio, "Image segmentation by iterative inference from conditional score estimation," 2017, *arXiv:1705.07450*. [Online]. Available: http://arxiv.org/abs/1705.07450

[21] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2650–2658.

[22] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[23] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Cham, Switzerland: Springer, 2017, pp. 240–248.

[24] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[25] H. Schwenk and Y. Bengio, "Boosting neural networks," *Neural Comput.*, vol. 12, no. 8, pp. 1869–1887, Aug. 2000.

[26] L. Wang, B. Zhang, J. Han, L. Shen, and C.-S. Qian, "Robust object representation by boosting-like deep learning architecture," *Signal Process., Image Commun.*, vol. 47, pp. 490–499, Sep. 2016.

[27] S. Han, Z. Meng, A.-S. Khan, and Y. Tong, "Incremental boosting convolutional neural network for facial action unit recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 109–117.

[28] K. Sirinukunwattana *et al.*, "Gland segmentation in colon histology images: The GlaS challenge contest," 2016, *arXiv:1603.00275*. [Online]. Available: http://arxiv.org/abs/1603.00275

[29] G. N. Gunesli, C. Sokmensuer, and C. Gunduz-Demir, "AttentionBoost: Learning what to attend for gland segmentation in histopathological images by boosting fully convolutional networks (supplementary material)," Dept. Comput. Eng., Bilkent Univ., Ankara, Turkey, Tech. Rep. BU-CE-2001, 2020. [Online]. Available: http://www.cs.bilkent.edu.tr/tech-reports/2020/BU-CE-2001.pdf

[30] E. Cha, J. Jang, J. Lee, E. Lee, and J. Chul Ye, "Boosting CNN beyond label in inverse problems," 2019, *arXiv:1906.07330*. [Online]. Available: http://arxiv.org/abs/1906.07330

[31] C. F. Koyuncu, R. Cetin-Atalay, and C. Gunduz-Demir, "Object-oriented segmentation of cell nuclei in fluorescence microscopy images," *Cytometry A*, vol. 93, no. 10, pp. 1019–1028, Oct. 2018.

[32] S. Arslan, T. Ersahin, R. Cetin-Atalay, and C. Gunduz-Demir, "Attributed relational graphs for cell nucleus segmentation in fluorescence microscopy images," *IEEE Trans. Med. Imag.*, vol. 32, no. 6, pp. 1121–1131, Jun. 2013.