2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)
April 3-7, 2020, Iowa City, Iowa, USA

MULTIPLE INSTANCE LEARNING VIA DEEP HIERARCHICAL EXPLORATION FOR HISTOLOGY IMAGE CLASSIFICATION

Jan Hering, Jan Kybic

Department of Cybernetics, Faculty of Electrical Engineering,
Czech Technical University in Prague, Czech Republic

ABSTRACT

We present a fast hierarchical method to detect a presence of cancerous tissue in histological images. The image is not examined in detail everywhere but only inside several small regions of interest, called glimpses. The final classification is done by aggregating classification scores from a CNN on leaf glimpses at the highest resolution. Unlike in existing attention-based methods, the glimpses form a tree structure, low resolution glimpses determining the location of several higher resolution glimpses using weighted sampling and a CNN approximation of the expected scores. We show that it is possible to perform the classification with just a small number of glimpses, leading to an important speedup with only a small performance deterioration. Learning is possible using image labels only, as in the multiple instance learning (MIL) setting.

Index Terms— image segmentation; hierarchical; histology; microscopy; CNN

1. INTRODUCTION

We shall address the task of detecting the presence of an object or texture of interest in the image. In particular, we shall demonstrate our method on the task of detecting metastases in whole-slide histological images (WSI) of lymph nodes, as defined in the CAMELYON16 challenge [1]. In that case, pixel-level annotated training data is available, so that we know, which part of the image contains the feature of interest, in this case a tumor. We shall also consider a more difficult setting, called multiple instance learning (MIL), when only image-level annotations for the training data are available, not pixel-wise segmentations. The standard approaches usually divide the (very large) images into patches, classify each of them using a CNN, and then aggregate the patch-wise predictions [1, 2]. The current methods work well, outperforming even an expert human reader. However, these approaches are also slow, requiring the whole slide to be examined in detail.

This project was supported by the Czech Science Foundation project 17-15361S and the OP VVV project CZ.02.1.01/0.0/0.0/16_019/0000765, Research Center for Informatics.

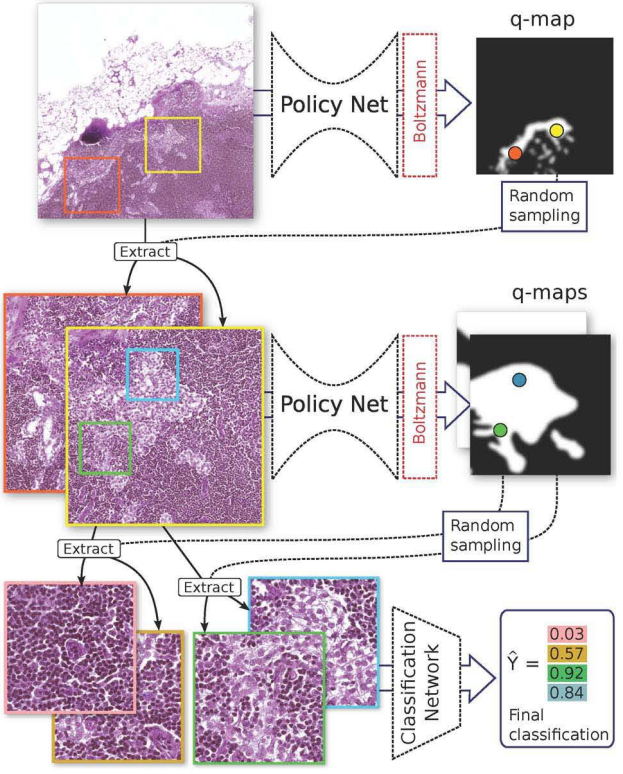

Fig. 1. Illustration of a three level hierarchical model for scaling factors 1,4, and 16. The glimpse location (color dots) is determined by random sampling from the Q heatmap calculated at the previous level.

To speed up the process, we shall proceed as human experts do: We use a low resolution version of the image to identify potentially positive (suspicious or relevant) regions of interest (called *glimpses*), limiting the size of the area to be examined. We continue hierarchically, until the highest resolution is reached, where the final classification is done. This idea has been explored before, using recurrent visual attention [3] or reinforcement learning [2]. In these approaches, the glimpses are sampled in a linear sequence, each depending only on the preceding state. In contrast, we propose a tree pattern, where the contents of each glimpse determines the loca-

tion of several smaller glimpses at the next higher resolution. We focus on glimpses estimated to be positive (and ignoring the negative ones), which corresponds to an explicit MIL aggregation. Finally, in our formulation we attempt to predict the score calculated at the final level with lower-resolution glimpses, which is a well-defined and relatively easy to optimize task, sidestepping the difficulty of the notoriously difficult to optimize reinforcement learning and recurrent models.

2. METHODS

At testing time, the method proceeds recursively, see Fig. 1. It starts at level $l = 0$ with a root glimpse (rectangle) g_0 covering the whole input image I. A glimpse g_j at level l generates n_l glimpses at level $l + 1$, using a *policy* (described below). The glimpse g_j is a rectangular region denoted $I_l = R_{s_l}(I)$ around a central coordinate $\mathbf{c}_j$ extracted from the original image downscaled by a factor s_l. We choose to increase the resolution 4 times between levels, i.e. $s_l = 4^{l_{\text{final}}-l}$, where l_{final} is the final level. All glimpses at all levels have the same size, we use 304×304 pixels.

2.1. Classification

At the final level a *classifier* produces a glimpse classification $\hat{y}_j = C(g_j)$ for each glimpse g_j. The classification network C is an Inception-V3 network [4] and its output $\hat{y}_j \in [0, 1]$ is the probability that a glimpse g_j is positive, i.e., that it contains a tissue of interest, in our case a tumor. There are $N_G = \prod_{l=0}^{l_{\text{final}}-1} n_i$ leaf glimpses at level l_{final}. The final classification $\hat{Y} = \max \hat{y}_j$ over all leaf glimpses.

2.2. Glimpse sampling

For any level except the final, $l < l_{\text{final}}$, a glimpse g_j is fed into a *policy network* P_l, to obtain a *heatmap* $q_j = P_l(g_j)$ of the same size as g_j. The policy network is a U-Net architecture [5], independent for each scale. The idea is that the values in the heatmap q_j at level l, which can be calculated cheaply from reduced resolution image I_l, approximate the final classification C at the same location, which however needs full-resolution images and is therefore much more expensive to calculate. In particular, we would like

$$\big(K_{l+1} * q_j\big)(\mathbf{c}_k) = \mathrm{E}\big[\hat{y}_k\big] \quad \text{with} \quad \hat{y}_k = C(g_k) \tag{1}$$

where g_k is a glimpse at the final level $l = l_{\text{final}}$ centered at point $\mathbf{c}_k$ and K_{l+1} is a box ($\{1, 0\}$) kernel corresponding to the spatial extent of glimpses at level $l + 1$. This smoothing encourages the expected score $\mathrm{E}\big[\hat{y}_k\big]$ to be a smooth function of the glimpse center coordinates $\mathbf{c}_k$; it also simplifies the learning.

The n_l glimpse centers $\mathbf{c}$ at level $l + 1$, direct descendants of g_j, are chosen within g_j using Boltzmann exploration [6], i.e. randomly with probabilities proportional to a *transformed heatmap* Q

$$\mathrm{Prob}\big[\mathbf{c} = \mathbf{x}\big] \propto Q_l(\mathbf{x}) \tag{2}$$

$$\text{where} \quad Q_l(\mathbf{x}) = \exp\big(-\big(K_{l+1} * q_j\big)(\mathbf{x})/T\big) \tag{3}$$

where T is a temperature. The idea is the same as in simulated annealing: In the beginning, when the policy networks P_l are not well trained, we choose large temperature T and the new glimpse centers at level $l + 1$ are sampled almost uniformly within the their parent glimpse g_j. Later on, when the policy prediction accuracy improves, T is gradually decreased, so that only points with high $K * q$ values are considered. In the limit, when $T \to 0$, a maximum location would be always taken. At training time, T is decreased following a decay schedule. At test time, T is set to some small value, determined by cross-validation; for our application $T = 0.01 \sim 0.1$ seems to work well.

To avoid resampling the same location multiple times, which would result in many overlapping glimpses, we perform standard *maximum suppression* — after one glimpse center $\mathbf{c}$ is chosen, the area around it corresponding to its spatial extent (identical to K_{l+1}) is set to zero. This way, the promising regions are explored more efficiently, with minimum overlap between glimpses. Borders are also zeroed for glimpses to be fully included in the parent glimpse.

2.3. Training

We have two training strategies, which we call *supervised* and *MIL*. The supervised strategy assumes that we can find out the reference (ground truth) classification y_k^* for any leaf glimpse g_k. In the CAMELYON16 dataset, we use the provided pixel-level classifications. The classification network C is trained using binary cross entropy loss function $\mathcal{L}_C$. Since each training image leads to N_G leaf glimpses, the total loss per image is

$$\sum_{g_k} \mathcal{L}_C(\hat{y}_k, y_k^*) \tag{4}$$

where the sum is taken over all training glimpses (patches) at the finest level.

The policy network is trained to minimize the sampled squared difference derived from (1) and inspired by reinforcement learning. In particular, let $g_{\tau(0)=0}, g_{\tau(2)}, \ldots, g_{\tau(l_{\text{final}})=k}$ be a sequence of glimpses (referred to as a trajectory) from the root glimpse $g_{\tau(0)=0}$ to a leaf glimpse $g_{\tau(l_{\text{final}})=k}$, with centers $\mathbf{c}_{\tau(0)}, \ldots, \mathbf{c}_{\tau(l_{\text{final}})=k}$, such that $g_{\tau(l)}$ is a parent of $g_{\tau(l+1)}$. The loss function for one input image will be a sum over all N_g trajectories τ_k for all leaf glimpses g_k.

$$\mathcal{L}_P = \sum_{\tau_k} \sum_{l=0}^{l_{\text{final}}-1} \Big(\big(K_{l+1} * q_{\tau(l)}\big)(\mathbf{c}_{l+1}) - \hat{y}_{\tau(l_{\text{final}})}\Big)^2 \tag{5}$$

The *MIL strategy* is used when pixel-level classification is not available, i.e. when only image labels Y^* are provided.

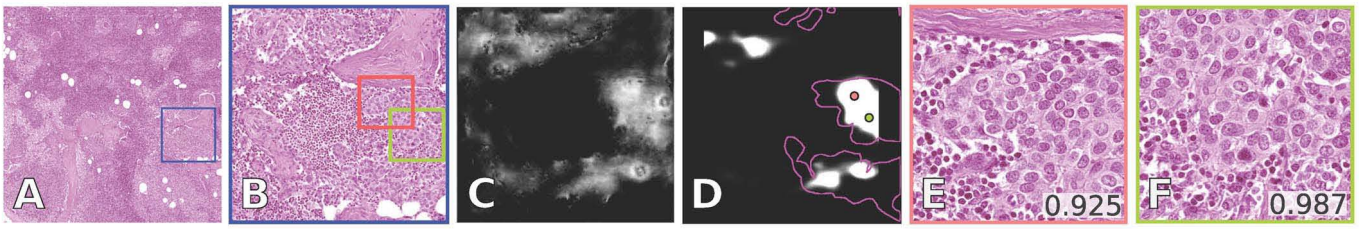

Fig. 2. (A) Input image at the initial scale with a glimpse marked in blue, (B) second level glimpse with two successor glimpses, (C) the policy-net output corresponding to B, (D) Q-map after Boltzmann transformation at $T = 0.1$ with a ground truth overlay in magenta and the two sampled glimpse centers, and (E, F) the sampled leaf glimpses and their score by the classification network.

Two modifications must be performed in (4). First, as we no longer have a set of labeled glimpses, we start from a full training image I and use the glimpse sampling (Section 2.2) to get a set of N_G leaf glimpses g_k. Second, the glimpse labels y_j^* are replaced by the image label Y_I^*. During training, the policy network becomes sufficiently well trained to select more positive glimpses in positive images, so that the majority of glimpse labels is correct. For negative images, where all glimpses are negative by definition of the MIL problem, the label is always correct.

As shown below, training from scratch using the MIL strategy works on our data. In more difficult cases, we can pretrain both networks on a small dataset using the supervised strategy and then fine-tune using the MIL strategy on a bigger dataset with image annotations only.

3. EXPERIMENTS

3.1. Evaluation data

We extract tiles of size 5000×5000 px at the 20× magnification level from CAMELYON 16 dataset images. (Using tiles is necessary due to GPU memory limitations.) We perform a stain-specific color normalization [7]. Its by-product is a tissue mask, used to limit processing to tissue only. The training set for the binary classification experiments consists of 714 images (406 normal), the testing set has 259 images (147 normal). To assess the variance from random sampling, we repeat the evaluation for each parameter settings 10 times and report the mean and standard deviation (SD) of the image-level AUC.

3.2. Supervised training

We initialize the Inception-V3 classification network C with ImageNet-weights and train it on patches extracted at the highest resolution by minimizing the cross-entropy loss $\mathcal{L}_C$. Initially, the policy network is not used at all. Instead, we sample glimpses uniformly from known positive and negative classes. Later on, when C is already partially trained, we train two-level policy networks P_0, P_1 at levels $l = 0, 1$, with

T	N_G	mean AUC		Time
		MIL	Supervised	[s] per WSI
0.05	4	0.9166	0.9200	1.05
	8	0.9362	0.9455	1.18
	16	0.9409	0.9540	1.43
0.1	4	0.9216	0.9275	1.02
	8	0.9373	0.9534	1.17
	16	0.9428	0.9549	1.42
-	All	0.9582	0.9667	12.16
Ben Taieb [3]		0.95*	–	~ 4
Wang et al. [2]		–	0.96*	*n/a*

Table 1. Testing-set AUC comparison for different choice of the number of leaf glimpses N_G and two temperatures T. (*) values reported for the full CAMELYON16 testing set.

$l_{\text{final}} = 2$, corresponding to downscaling factors $s_0 = 16$ and $s_1 = 4$, respectively. We alternate training of P and C at odd and even epochs. The initial learning rate is 10^{-3} and a decay factor 0.1 every 1000 steps.

Results are shown in column *Supervised* in Table 1 and as solid lines in Fig. 4 and Fig. 3. As expected, reducing the number of glimpses reduces the evaluation time. For example using 8 leaf patches is about 12× faster than testing all patches, while the AUC decreases by only about 1 %. The accuracy of the reinforcement learning method [2] is between our method with $N_G = 16$ and evaluating everywhere at the finest level ('All').

3.3. Multiple instance learning scenario

In the second experiment, only weak image-level labels are used for training. The architecture parameters are the same. The classification network C except the last layer is initialized by pre-trained ImageNet weights. The policy networks P_0, P_1 are trained from scratch. We use the image-level labels

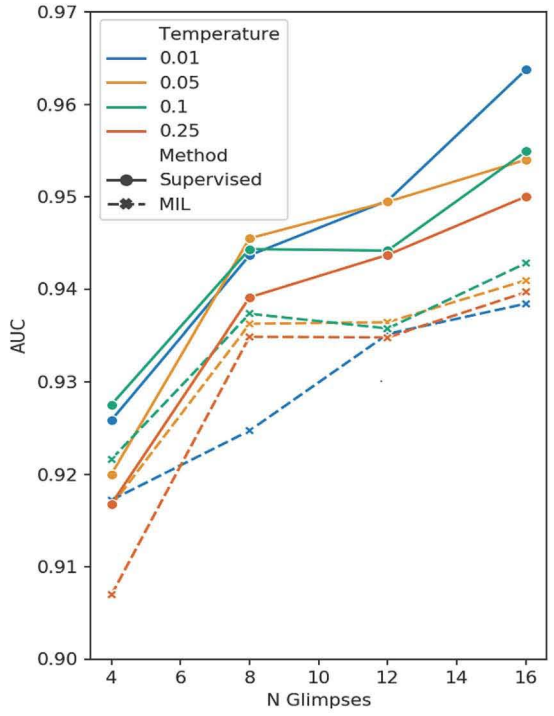

Fig. 3. Comparison of the models trained in a fully-supervised (solid line) and an MIL manner (dashed line). Each line shows the mean AUC over 10-fold repetitions.

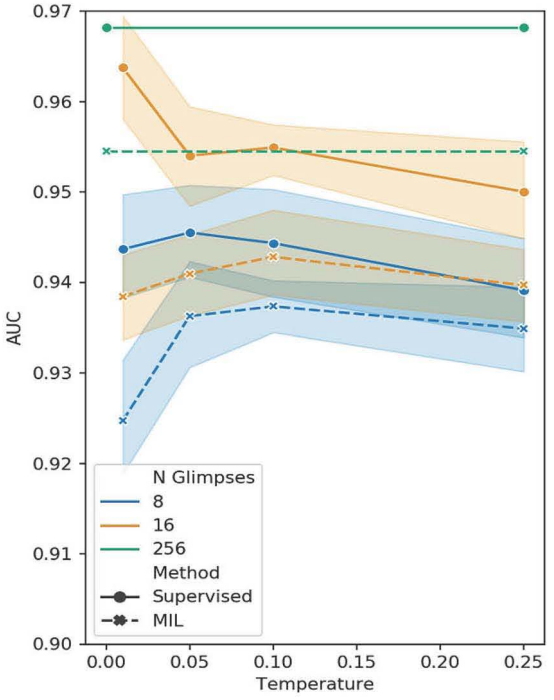

Fig. 4. Comparison of the models trained in a fully-supervised (solid line) and an MIL manner (dashed line). Each line shows the mean AUC over 10-fold repetitions, with $\pm\sigma$ error bands.

Y^* as glimpse labels y_k^*, as it is common in MIL. Results are shown in column *MIL* in Table 1, and by dashed lines in Fig 4 and Fig. 3. As expected, the model trained from weak labels (MIL) performs slightly worse compared to the supervised one but the difference is small, about $1\,\%$. Both the speed and performance of the attention method [3] is between our method run with with $N_G = 16$ and evaluating everywhere at the finest level ('All'). Glimpse and heatmap examples from the MIL training scenario are shown in Fig. 2.

4. CONCLUSION AND DISCUSSION

We have shown that our sparse hierarchical detection method is faster than dense methods with only a small performance drop, it is also faster than existing attention-based methods. The comparison to [3, 2] regarding AUC tends to be optimistic as the experiments were run only on a subsection of the full CAMELYON dataset. We observed experimentally that our method is easier to train, possibly because the tree exploration pattern is more powerful than a linear one for this task, and perhaps because predicting a score is easier than predicting a location. Our method can learn from weak, image-level labels, solving the MIL problem. It can be easily adapted to other detection tasks.

5. REFERENCES

[1] B. E. Bejnordi *et al.*, "Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer," *Jama*, vol. 318, no. 22, pp. 2199–2210, 2017.

[2] D. Wang *et al.*, "Deep learning for identifying metastatic breast cancer," 2016, arXiv:1606.05718.

[3] A. BenTaieb and G. Hamarneh, "Predicting cancer with a recurrent visual attention model for histopathology images," in *MICCAI*, 2018, pp. 129–137.

[4] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception architecture for computer vision," in *CVPR*, 2016, pp. 2818–2826.

[5] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015, pp. 234–241.

[6] A. Lazaric, M. Restelli, and A. Bonarini, "Reinforcement learning in continuous action spaces through sequential Monte Carlo methods," in *NIPS*, 2008, pp. 833–840.

[7] B. E. Bejnordi *et al.*, "Stain specific standardization of whole-slide histopathological images," *IEEE Trans. Med. Imag.*, vol. 35, no. 2, pp. 404–415, 2015.