frontiers in Medicine

ORIGINAL RESEARCH
published: 14 December 2021
doi: 10.3389/fmed.2021.767625

A Robust Training Method for Pathological Cellular Detector *via* Spatial Loss Calibration

Hansheng Li[1†], Yuxin Kang[1†], Wentao Yang[2†], Zhuoyue Wu[1], Xiaoshuang Shi[3], Feihong Liu[1], Jianye Liu[1], Lingyu Hu[4], Qian Ma[4], Lei Cui[1], Jun Feng[1*] and Lin Yang[1*]*

[1] School of Information Science and Technology, Northwest University, Xi'an, China, [2] Fudan University Shanghai Cancer Center, Shanghai, China, [3] Department of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China, [4] AstraZeneca, Shanghai, China

OPEN ACCESS

Edited by:
Zhuotun Zhu, Johns Hopkins Medicine, United States

Reviewed by:
Takaaki Sugino, Tokyo Medical and Dental University, Japan
Yanning Zhou, The Chinese University of Hong Kong, China

***Correspondence:**
Lei Cui
leicui@nwu.edu.cn
Jun Feng
fengjun@nwu.edu.cn
Lin Yang
linyang@nwu.edu.cn

[†]These authors have contributed equally to this work

Specialty section:
This article was submitted to Precision Medicine, a section of the journal Frontiers in Medicine

Received: *31 August 2021*
Accepted: *15 November 2021*
Published: *14 December 2021*

Citation:
Li H, Kang Y, Yang W, Wu Z, Shi X, Liu F, Liu J, Hu L, Ma Q, Cui L, Feng J and Yang L (2021) A Robust Training Method for Pathological Cellular Detector via Spatial Loss Calibration. Front. Med. 8:767625. doi: 10.3389/fmed.2021.767625

Computer-aided diagnosis of pathological images usually requires detecting and examining all positive cells for accurate diagnosis. However, cellular datasets tend to be sparsely annotated due to the challenge of annotating all the cells. However, training detectors on sparse annotations may be misled by miscalculated losses, limiting the detection performance. Thus, efficient and reliable methods for training cellular detectors on sparse annotations are in higher demand than ever. In this study, we propose a training method that utilizes regression boxes' spatial information to conduct loss calibration to reduce the miscalculated loss. Extensive experimental results show that our method can significantly boost detectors' performance trained on datasets with varying degrees of sparse annotations. Even if 90% of the annotations are missing, the performance of our method is barely affected. Furthermore, we find that the middle layers of the detector are closely related to the generalization performance. More generally, this study could elucidate the link between layers and generalization performance, provide enlightenment for future research, such as designing and applying constraint rules to specific layers according to gradient analysis to achieve "scalpel-level" model training.

Keywords: cellular detection, spatial loss calibration, sparsely annotated pathological datasets, convolutional neural network, object detection network

1. INTRODUCTION

Locating and counting cells in the pathological whole slide images (WSIs) is a direct way to find effective and important biomarkers, which is an essential and fundamental task of pathological image analysis (1–3). For instance, the spatial arrangement of tumor cells has been proved to be related to cancer grades (4, 5). Therefore, the qualitative and quantitative analysis of different types of tumors at cellular-level detection can help us better understand tumors and also explore various options for cancer treatment (6, 7).

Recently, object detection frameworks of Convolutional Neural Networks (obj-CNNs) have been proved powerful for locating instances in medical images [e.g., in CT images (8) and colonoscopy images (9)]. The big empirical success of obj-CNNs depends on the availability of a large corpus of fully annotated instances in training images (10). However, different from images of other modalities, we find two kinds of distributions of cells in pathological images, namely embedded and dense distribution, making full annotations of cellular-level instances difficult to

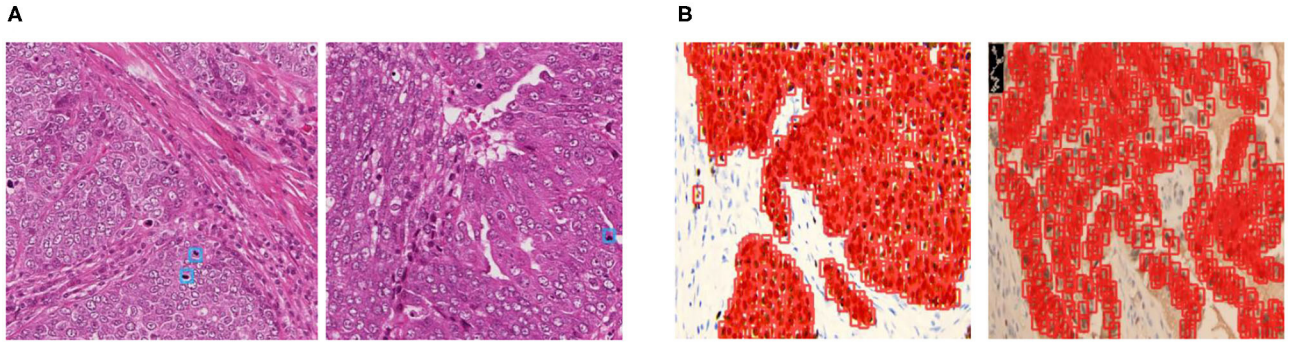

FIGURE 1 | The examples of annotations in the two kinds of datasets. **(A)** Cells of embedded distribution, which are sampled from the MITOS-ATYPIA-14 dataset. Obviously, mitoses that need to be annotated are often hidden among hundreds of other cells, tough to categorize and locate. **(B)** Cells of dense distribution in our Ki-67 dataset, usually more than hundreds of cells are required to be annotated in a small patch sampled from the whole slide image (WSI), which is an expensive and laborious task.

be guaranteed (refer to **Figure 1**). Specifically, the embedded distribution means that positive cells are hidden among hundreds of other cells, which are challenging for pathologists to categorize, locate, and then annotate. As for the dense distribution, a small patch sampled from the WSIs may contain hundreds of positive cells, making the annotation task expensive and laborious. Therefore, sparsely annotated datasets (SADs) are common in the field of the detection of cells.

In fact, when the training dataset contains a certain amount of sparse cellular annotations, the overfitting issue tends to easily occur, naturally leading to poor performance in generalization (11). In this study, we show the fundamental problems that decrease the generalization performance of the detector trained on SADs. First, deviation-loss, that is, numerous unannotated positive cells are mistaken for negative ones in the SADs, resulting in a serious miscalculated loss during training. Second, the deviation-loss dominates the early training process, and then drives the detector to learn only the features of the annotated cells, which yields the overfitting issue (Experimental testify can be seen in **Appendix A1**).

In this study, we point out that alleviating the deviation-loss during the training process can guide the detector to continuously learn the features of positive cells rather than only the annotated ones, and the SADs overfitting problem can be solved. In order to achieve that goal, the first cornerstone is how to identify those positive cells from negative ones when annotations are missing. We observe the more and more significant difference in densities between the predictions of the positive and negative cells during training (refer to **Figure 2**). Based on this observation, we propose a SADs training method named Boxes Density Energy (BDE), which utilizes densities' information to reduce the deviation-loss. Specifically, the more predictions for a cell, the more likely the cell is to be positive, and these predictions deserve smaller losses. In this way, deviation-loss disappears, and meanwhile, the overfitting problem is solved naturally.

We have conducted experiments on two datasets, namely the MITOS-ATYPIA-14 dataset (embedded distribution)[1] and the Ki-67 dataset (dense distribution), which can be seen in **Figure 1**. Sufficient experimental results prove that our training method can significantly boost the performance of SADs. More importantly, we explore the gradient in the network and find that BDE brings a significant improvement on the middle layers (20–60 layers, 80 layers in total) of the network, indicating that the network's generalization performance seems to be closely related to the middle layers of the network. This may change the current training paradigm, such as applying constraint rules to specific layers according to gradient analysis to achieve the "scalpel-level" model training.

The organization of the study is as follows. The review of obj-CNNs and recent literature on SADs training methods is given in Section 2. Section 3 describes the proposed method in detail, and experimental results are presented in Section 4. Finally, we analyze the gradient of the trained network and conclude in Sections 5, 6, respectively.

A preliminary version of this study has been published in a conference study (12), which is only evaluated on the MITOS-ATYPIA-14 dataset. In this study, we have made significant extensions to generalize our methods on the Ki-67 dataset, aiming to provide a strong and comprehensive theory for relevant research. To be specific,

- We explore that some specific layers of CNN are strongly related to generalization performance, may provide theoretical guidance for future related research, e.g., one can improve the generalization of the network through more constraints on middle layers when training the network.
- In this study, we define the networks' training problems on SADs, from deviation-loss to the overfitting issue.

[1]MITOS-ATYPIA-14 dataset: https://mitos-atypia-14.grand-challenge.org/dataset/.

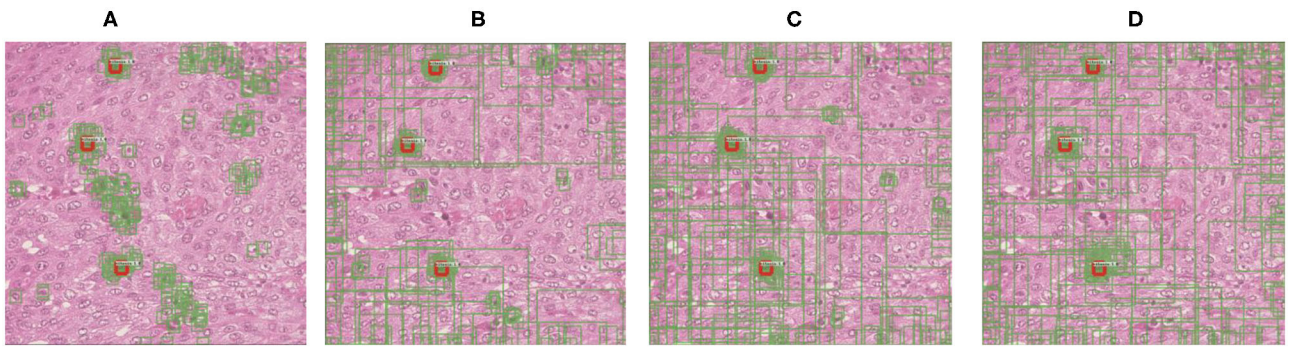

FIGURE 2 | The change of regression boxes' density during training on the MITOS-ATYPIA-14 dataset. Among them, the regression box is in green and manually annotated as red. This typical example shows that as the training process progresses, regression boxes increasingly surround the positive cells. **(A)** Regression boxes in 1 k training steps. **(B)** Regression boxes in 4 k training steps. **(C)** Regression boxes in 7 k training steps. **(D)** Regression boxes in 10 k training steps.

- This study formulated two cells' distribution in pathological images, namely embedded and dense distribution which may easily lead to SADs, and BDE can solve the SADs training problem on both embedded and dense distributions.

2. RELATED STUDY

2.1. Object Detection Networks

2.1.1. The Framework

Object detection networks can be divided into two major categories, anchor-free and anchor-based frameworks. Among them, anchor-free frameworks (13, 14) are essentially making dense predictions, receiving higher recall rates but lower accuracy results (15), which do not meet the requirement of precisely pathological image analysis. On the other hand, anchor-based frameworks are more suitable for our tasks, and can be generally divided into one-stage methods (16, 17) and two-stage methods (18, 19). Both of them first tile a large number of preset anchors on the image, then predict the category and refine the coordinates of these anchors by one or several times, finally output these refined anchors as detection results. Because two-stage frameworks refine anchors several times more than one-stage frameworks (as shown in **Figure 3**), the former has greater accuracy. Hence, we choose the two-stage Feature Pyramid Network (FPN) (19) as the baseline in this paper.

2.1.2. The Loss Function and Deviation Loss

In order to locate and recognize positive cells in the image, the object detection network has two parallel output layers to generate regression boxes (b) with probability distribution (p). The original loss (L) consists of the classification loss L_{cls} and bounding-box regression loss L_{loc}:

$$L\left(p,u,b,v\right) = L_{\text{cls}}(p,u) + L_{\text{loc}}\left(b,v\right),, \tag{1}$$

$$L_{cls}(p,u) = \sum_{k} -\left[u_k \cdot \log\left(p_k\right)\right], \tag{2}$$

$$L_{\text{loc}}\left(b,v\right) = \sum_{k} smooth_{L_1}\left(b_k - v_k\right), \tag{3}$$

$$smooth_{L_1}(x) = \begin{cases} 0.5x^2 & if\,|x| < 1 \\ |x| - 0.5 & otherwise. \end{cases} \tag{4}$$

In Equation (2), u_k represents a one-hot label for a regression box indexed by k. When k-box's Intersection Over Union (IoU) with any instance annotation higher than a threshold, is assigned with a positive one-hot label ($u_k \neq 0$), otherwise a negative ($u_k = 0$). In Equation 3, v indicates the annotated bounding-boxes.

The loss function can accurately measure the margins between p and u, b, and v on the fully annotated dataset. However, on the sparsely annotated cellular dataset, all unannotated positive cells are mistaken for negative, and u and v are translated into "untrustworthy" ground-truths. Thus, L_{cls} and L_{loc} may deviate seriously from the correct value, which we name deviation-loss. As a result, the deviation-loss confuses the training of networks, leading to limited performance.

2.2. Sparsely Annotated Datasets Training Methods

2.2.1. Pseudo-Annotation Based Methods

In order to solve the SADs training problem, pseudo-annotation based methods have been proposed and achieved success on natural images (20, 21). They first train the detector using available instance-level annotations, then generate pseudo-annotations, and merge them with the original annotations to iteratively update the detector. For example, Niitani et al. (22) trained the detector to generate annotations using the Open Images Dataset V4 (OID). They then sampled the pseudo-annotations using assumptions such as "cars should contain tires." However, such a priori assumption in the field of cell detection is unknown. Other methods based on pseudo-annotations still need a certain number of fully annotated datasets, like Yan et al. (23) and Inoue et. al. (24) employ a subset of fully annotated datasets to obtain a pre-trained detector, generating pseudo-annotations for the next training.

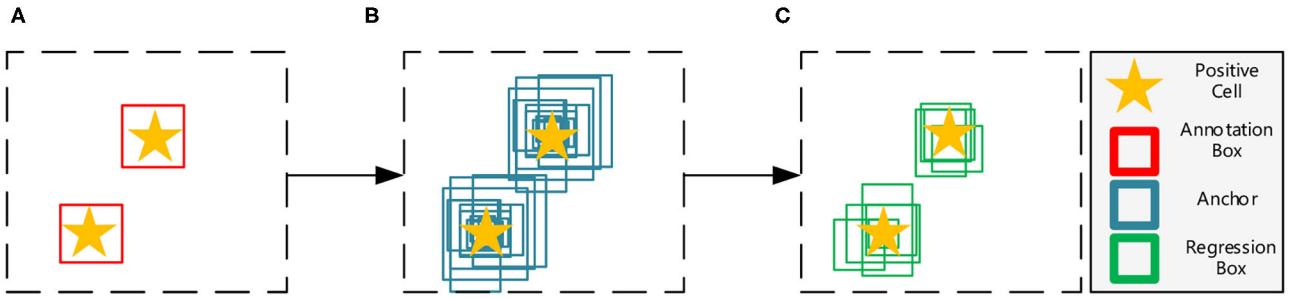

FIGURE 3 | Illustration of the anchor based two-stage framework. **(A)** The input image with manual annotations. **(B)** The first stage refines the initial thousands of anchors. **(C)** The second stage refines previous results and obtains hundreds of regression boxes.

Obviously, such an iterative process brings uncontrollability into the training process, e.g., a bad pseudo-annotation generator may significantly influence the final results. In addition, there is not much consensus on how to utilize the pseudo-annotations until now, especially for object detection (22), e.g., determining the optimal number of iterations is tricky, therefore, it is urgent to solve the SADs training problem in a non-iterative way. Besides, considering that such methods are relatively difficult to replicate, with respect to, empirical and tricky parameter selection or special requirements of the forms of datasets, this study does not include such methods in the comparative experiment.

2.3. Loss-Calibration Based Methods

Compared with pseudo-annotation based methods, the loss-calibration methods for solving noise labels are more relevant to our study. The meaning of noise labels is wrong labels or missing labels (25, 26). These methods aim to reduce noise labels by establishing loss functions that are more noise-tolerant. For example, Müller et al. (27) softens the labels by adding a uniform distribution. Wang et al. (28) assumes that the network will become more and more reliable as the training continues and proposes reducing the loss gradually to reduce the influence of noise labels. However, these loss calibration methods also inevitably reduce the core contributions of correct labels for the training of the network. On the contrary, our BDE utilizes the regression boxes' density to encourage correct predictions and give relatively more significant losses to wrong predictions, whether the label is missing or not.

It is worth noting that in view of the class imbalance problem they, the have put forward many loss weighting schemes (17, 29). However, these methods may cause relatively large losses to correct predictions lacking corresponding annotations, which makes them ineffective on SADs.

3. BOXES DENSITY ENERGY

The overall process of our proposed BDE is shown in **Figure 4**. BDE is proposed to encourage the correct predictions of unannotated positive cells to ignore the adverse effect of the deviation-loss, which can be summarized into five core steps. **Figure 4A** A sparsely annotated image is inputted for the training. At the second stage of the detector, each cell is surrounded by some regression boxes automatically that we regard as a group. **Figure 4B** Boxes Density: Calculate the average distance between each box and the others. **Figure 4C** Boxes Energy: Normalized operation by dividing the Box Density by the maximum distance between all boxes. **Figure 4D** Calculate the original total loss. **Figure 4E** BDE loss: Calibrate the original loss with Boxes Energy to guide the detector training in the right direction.

3.1. Boxes Density

The boxes density can be measured by the average distance between each box, so that denser boxes have smaller average distances than isolating ones. The density of a box indexed by i can be represented as:

$$\text{Density}(b_i) = \frac{1}{N}\sum_{j}^{N} D\left(b_i, b_j\right), \tag{5}$$

where N is the number of boxes per image, D is the distance function, we choose Manhattan distance (Equation 6) in this study considering the less computational cost.

$$D\left(b_i, b_j\right) = \left|x_i - x_j\right| + \left|y_i - y_j\right|, \tag{6}$$

In which, the x_i and y_i represent the x-coordinate and y-coordinate of the center point of the box indexed by i.

We can prove that the average distance can measure the density effectively; if we treat regression boxes around a cell as a group, and assume that we have k groups $\{G_1, \ldots, G_j, \ldots G_k\}$. Meanwhile, there are $\{m_1, \ldots, m_j, \ldots, m_k\}$ boxes in the corresponding group.

For simplicity, we assume that the distances within a group are all close to 0, the distances between the groups are all d, and the total number of boxes is N, which means that $N = \sum_{l=1}^{k} m_L$. Thus, the average distance of each box in the j-group is Equation (7). This indicates that the box in a denser group (larger m_j)of the j-group has a smaller density value.

$$\text{Density}(b_i) = \frac{0 \times m_j + (N - m_j) \times d}{N} = d \times (1 - \frac{m_j}{N}). \tag{7}$$

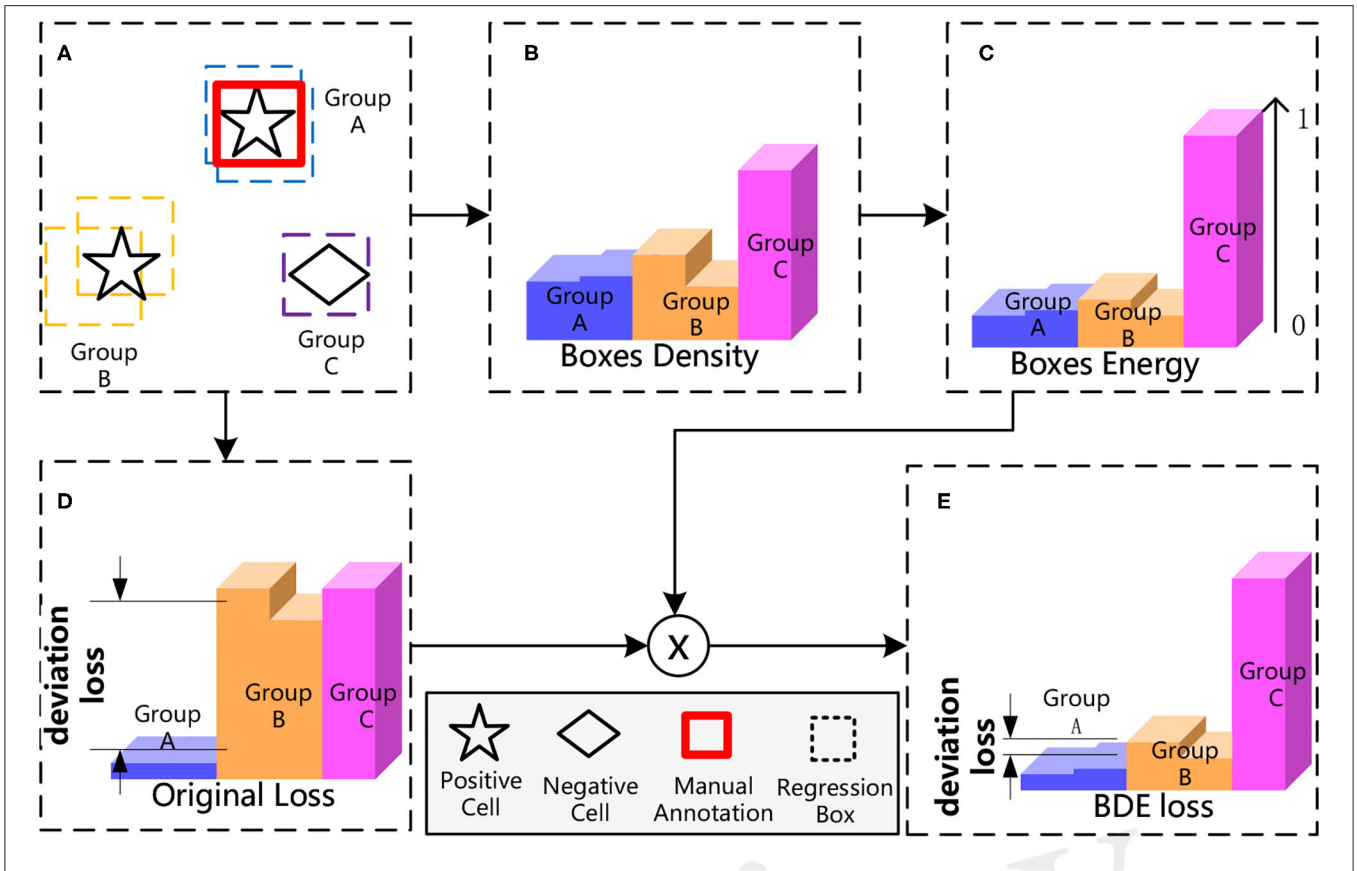

FIGURE 4 | The process of Boxes Density Energy (BDE). The core idea of BDE is to use the distribution of regression boxes in **(A)**, to obtain the Boxes Energy of **(C)**, and to correct the deviation loss in **(D)**.

3.2. Boxes Energy and Loss Calibration

The main idea of our proposed method is that the more prediction boxes around a cell, the cell is more likely to be positive, and therefore, the predictions should have a smaller loss. The density of each box has been modeled, however, the range of density is not normalized. Therefore, we use Equation (8) to convert the Boxes Density to Boxes Energy which is normalized from 0 to 1. Afterward, Boxes Energy can be utilized as a weight of L_{cls} and L_{loc} (refer to Equations 9, 10). By that, the deviation loss is alleviated by calibrating the original loss.

$$\text{Energy}(b_i) = \frac{\text{Density}(b_i)}{\max(D(b))}. \tag{8}$$

$$L_{cls}^{BDE}(p,u) = \sum_k \left[1_{u_k=0}\left(Energy\left(b_k\right)\right) + 1_{u_k\neq 0}\right]\cdot\left[-u_k\cdot\log\left(p_k\right)\right], \tag{9}$$

$$L_{loc}^{BDE}(b,v,u) = \sum_k \left[1_{u_k=0}\left(Energy\left(b_k\right)\right) + 1_{u_k\neq 0}\right] \cdot\left[\text{smooth}_{L_1}\left(b_k - v_k\right)\right]. \tag{10}$$

In Equations 9, 10, u_k equals zero indicates the one-hot label of the box indexed by k is negative. With the loss-calibration of BDE, the detector can be trained along the right direction on the SADs. For example, if the box indexed by k is mistaken for negative (u_k is zero) due to SADs, but has a small $Energy(b_k)$, then, the original deviation-loss is calibrated by the term of $Energy(b_k)$. Finally, the total loss is improved from Equation (1) to:

$$L^{BDE}\left(p,u,b,v\right) = L_{cls}^{BDE}(p,u) + L_{\text{loc}}^{BDE}\left(b,v\right). \tag{11}$$

4. EXPERIMENTS

We utilize the FPN (19) with the backbone resnet50 (30) as the baseline. Our method is also compared with the representative loss-calibration methods, namely Label Smooth (LS) (27) and ProSelfLC (28). In Section 4.3, we conduct experiments to detect mitosis on the 2014 MITOS-ATYPIA Grand Challenge dataset and to detect tumor-cells on the Ki-67 dataset in Section 4.4. These two datasets can represent embedded and dense annotations. Experimental results demonstrate that BDE outperforms other methods on the SADs significantly, and BDE can address the training problem of SADs of both embedded and dense annotations.

4.1. Description and Implementation Details

The experiments for KI-67 and 2014 MITOS-ATYPIA datasets set the same hyperparameters. The inputted image is resized to the resolution of 800 × 800 pixels. The number of training steps is 10 k. The learning rate is initially set to 0.001 and is divided by 10 at 5 k and 7.5 k steps. In order to objectively evaluate our method, we perform 4-fold cross-validation on the MITOS-ATYPIA-14 dataset and 3-fold cross-validation on the Ki-67 dataset. We implement our framework with the open source software library TensorFlow version 1.12.0 on a workstation equipped with two NVIDIA GeForce 2080 Ti GPUs.

4.2. Evaluation Metrics

The average precision (AP) and recall are used for performance evaluation. The recall is defined as the proportion of all positive examples ranked above a given rank. Precision is the proportion of all examples above that rank that are from the positive class. The AP summarizes the shape of the precision/recall curve and is defined as the mean precision at a set of eleven equally spaced recall levels [0, 0.1,..., 1]:

$$AP = \frac{1}{11} \sum_{r \in \{0, 0.1, \ldots, 1\}} p_{\text{interp}}(r). \tag{12}$$

The precision at each recall level r is interpolated by taking the maximum precision measured for a method for which the corresponding recall exceeds r:

$$p_{\text{interp}}(r) = \max_{\tilde{r}:\tilde{r} \geq r} p(\tilde{r}), \tag{13}$$

where $p(\tilde{r})$ is the measured precision at recall $\tilde{r}$ (31).

4.3. Experiments on the 2014 MITOS-ATYPIA Grand Challenge Dataset (embedded annotations)

4.3.1. Data Description

We have conducted experiments on the 2014 MITOS-ATYPIA Grand Challenge Dataset (MITOS-ATYPIA-14 dataset). The data samples were scanned by two slide scanners Aperio Scanscope XT and Hamamatsu Nanozoomer 2.0-HT, whole-slide histological images (WSIs) stained with standard hematoxylin and eosin (H&E) dyes. The centroids pixels of mitoses were manually annotated *via* two senior pathologists. In a situation of contradiction between the pathologists, the third one will provide the final say.

We choose the train-set of WSIs scanned from Hamamatsu Nanozoomer 2.0-HT, and we sample 393 patches that contain 743 mitoses with a sliding window of resolution of 1,663 × 1,485 pixels. Annotations for training the FPN are generated by 32×32 bounding boxes centered on all centroids pixels. For the MITOS-ATYPIA-14 dataset, we refer to the original data as a fully annotated dataset. Meanwhile, we randomly delete annotations until there is only one per training image and name it as an extremely sparse dataset. It is worth noting, we only conduct the sparse operations on the training dataset, and the testing dataset is intact.

4.3.2. Results of MITOS-ATYPIA-14 dataset

Boxes Density Energy can improve recall results on the fully annotated dataset. **Table 1** lists the recall and AP results on the fully annotated dataset. For the AP results, all methods have lower AP results than the baseline (FPN), which demonstrates that when loss-calibration methods are introduced to the training on fully annotated embedded annotations, interfering with the network's accuracy. On the other hand, for the recall results, BDE can improve the recall results significantly. FPN, LS, and ProSelfLC achieve 89.8, 85.5, and 88.7% average recall, respectively. While BDE achieves 94.6%, exceeding that of FPN by 4.8%.

Boxes Density Energy improves the network's performance in all aspects on the sparsely annotated dataset. As shown in **Table 2**, BDE outperforms other methods significantly on both AP and recall results. However, LS's overall performance is reduced compared with the baseline, which indicates that the assumption of annotation-distribution of LS is incompatible in the embedded annotations, whose positive and negative samples are extremely unbalanced.

4.4. Experiments on the Ki-67 Dataset (Dense Annotations)

4.4.1. Data Description

The Ki-67 dataset is used for training FPN to detect tumor-cells and count their number. We have 206 patches with a resolution of 1,080 × 1,920 pixels sampled from WSIs, and the pathologists try their best to annotate all the tumor cells with key points in all patches. Finally, 21,025 tumor cells have been annotated. Then, we generate 32×32 bounding boxes centered on all key points.

4.4.1.1. The SAD of the Ki-67 Dataset

For the Ki-67 dataset, considering that there is an average of 102 annotated tumor cells in each patch, so we can retain different annotation rates to train the network to fully validate BDE, e.g., the retentive rate is 0.1 if 10% of annotations are retained. We have carried out experiments starting from the retentive rate of 0.1 and increasing it to 1 by 0.1. We believe that if the retentive rate is below 0.5, then the dataset we can define as a SADs because the number of unannotated instances is greater than the number of annotated instances in such a dataset. Experimental results have demonstrated the BDE can significantly boost the performance of networks trained on that SADs.

4.4.2. The Quantization Results

We evaluate the performance of our BDE which is trained on datasets with different retentive-rates, and observe that BDE is a robust training method, which is hardly affected by the quality of data annotations. For example, in **Table 3**, when the retentive-rate is dropped from 1.0 (original) to 0.1, BDE's AP result dropped from 49.02 to 46.45%, only reducing by 2.57%. On the other hand, FPN decreased by 23.88%, and LS decreased by 27.17%, and ProSelfLC decreased by 21.05%.

Similarly, **Table 4** lists the recall results of different methods trained on different retentive-rates. When the retentive-rate decreases from 1.0 to 0.1, BDE only reduces recall results by

TABLE 1 | The recall and average precision (AP) results on the fully annotated MITOS dataset (original dataset).

Method	Fold1		Fold2		Fold3		Fold4		Avg. Recall	Avg. AP
	Recall	AP	Recall	AP	Recall	AP	Recall	AP		
FPN (Baseline)	80.2	41.8	89.4	46.9	95.8	44.6	93.6	60.7	89.8	**48.5**
LS (27)	75.6	36.7	84.6	47.7	91.6	41.7	90.4	64.2	85.5	47.6
ProSelfLC (28)	80.2	32.7	86.5	40.6	95.2	40.3	93.1	62.7	88.7	44.1
BDE (ours)	90.6	40.7	93.3	42.3	99.4	43.2	95.0	59.1	**94.6**	46.3

TABLE 2 | The recall and AP results on the sparsely annotated MITOS dataset (retain one annotation in each image).

Method	Fold1		Fold2		Fold3		Fold4		Avg. Recall	Avg. AP
	Recall	AP	Recall	AP	Recall	AP	Recall	AP		
FPN (Baseline)	69.8	34.5	81.7	32.9	94.6	37.4	88.1	55.9	83.6	40.2
LS (27)	65.8	24.6	71.1	30.4	86.8	33.9	83.4	54.6	76.7	35.8
ProSelfLC (28)	80.2	28.8	84.6	28.4	95.8	30.1	85.7	50.1	86.5	34.3
BDE (ours)	88.5	41.8	89.4	37.1	95.8	40.2	91.3	60.1	**91.3**	**44.8**

TABLE 3 | The AP results on different annotations-retentive rates on the Ki-67 dataset.

Retentive rate	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
FPN (Baseline)	26.22	33.57	39.92	41.94	43.88	45.15	46.17	47.28	48.22	50.1
LS (27)	24.30,	37.39	41.01	44.16	45.48	46.47	47.69	48.96	50.01	51.47
ProSelfLC (28)	30.67	38.85	43.07	45.37	46.57	**47.72**	**48.79**	**49.91**	**50.87**	**51.72**
BDE (ours)	**46.45**	**46.36**	**46.24**	**46.71**	**46.94**	47.52	47.24	48.05	48.60	49.02

TABLE 4 | The recall results on different annotations-retentive rates on the Ki-67 dataset.

Retentive rate	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
FPN (Baseline)	38.84	44.81	48.34	49.84	51.01	51.73	52.39	53.14	53.64	54.22
LS (27)	31.45	45.23	48.01	50.82	51.93	52.84	**53.46**	**54.00**	**54.43**	**54.69**
ProselfLC (28)	43.43	48.24	50.28	51.62	52.25	52.78	53.31	53.90	54.16	54.38
BDE (ours)	**52.70**	**53.07**	**53.12**	**53.32**	**53.25**	**53.68**	53.37	53.82	53.95	53.29

0.59%. While FPN, LS, and Proself LC decreased by 15.38, 23.24, and 10.95%, respectively. Furthermore, from **Figures 5**, **6**, the robustness and stability of BDE can be demonstrated from the perspective of AP results and recall results' curves. Our method is almost unaffected by sparse annotations. In particular, when the retentive rate is in the range of 0.1–0.5, that is, sparse annotation, BDE achieves significant improvements.

4.4.3. The Qualitative Results

In **Figure 7**, we list some detection results produced by different methods. A score threshold of 0.6 is used for display. Obviously, other methods trained on the sparsely annotated dataset (the retentive rates is 0.1) tend to miss tumor cells, while our method largely avoids that mistake. Meanwhile, our BDE trained on the 0.1 retentive rate even achieve better performance than other methods trained on the 0.4 retentive rate.

5. LAYER-LEVEL GRADIENT ANALYSIS

5.1. Why Need Layer-Level Gradient Analysis

The gradient of a kernel is obtained by taking the chain derivative of the loss with respect to the weight, so that, the larger the weight of the kernel, not only its gradient is smaller but it also indicates that the kernel is more important. Thus, by comparing gradients of the same kernel but trained by different methods, we can know the advantages and disadvantages of training methods for this kernel. However, there are usually more than thousands of kernels in a single network, and it is not instructive to understand the superiority of kernel-level training. On the other hand, the same layer's kernels are responsible for similar feature extractions, e.g., kernels of a specific layer extract edges from different angles. Naturally, all kernels' average gradients in each

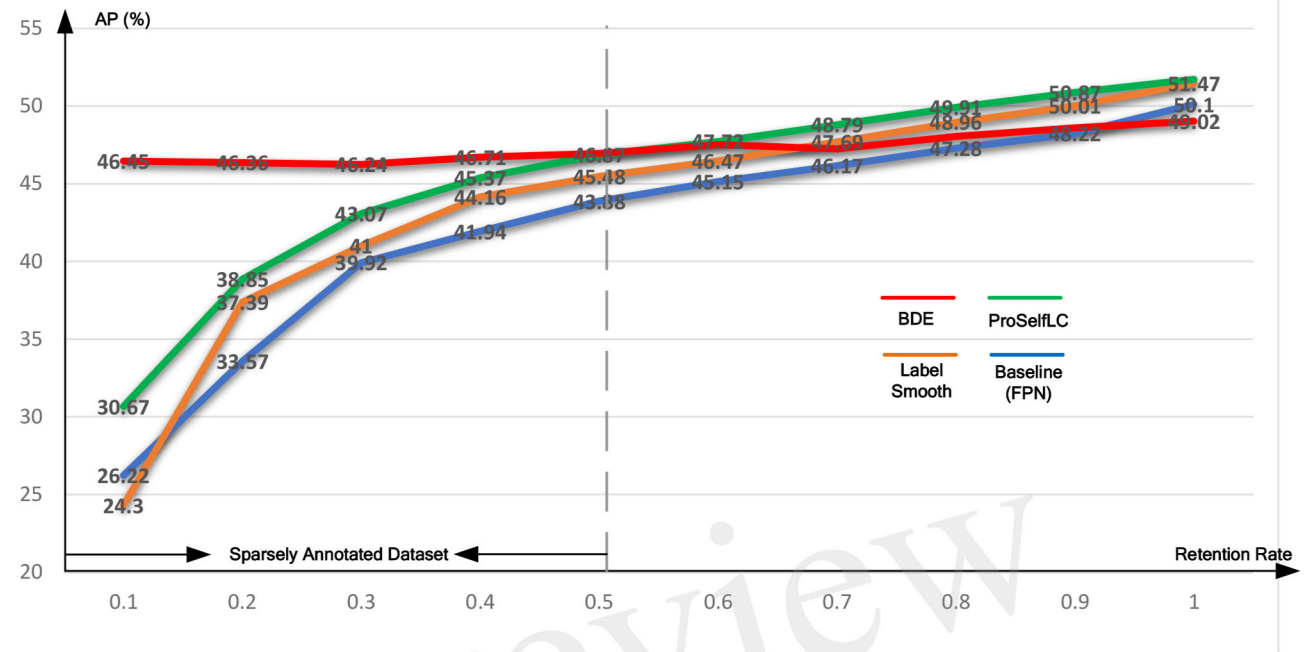

FIGURE 5 | The average precision (AP) test-results curve of different methods trained on the Ki-67 dataset of different retentive rates. The horizontal coordinate stands for different retentive rates and the vertical coordinate for AP(%).

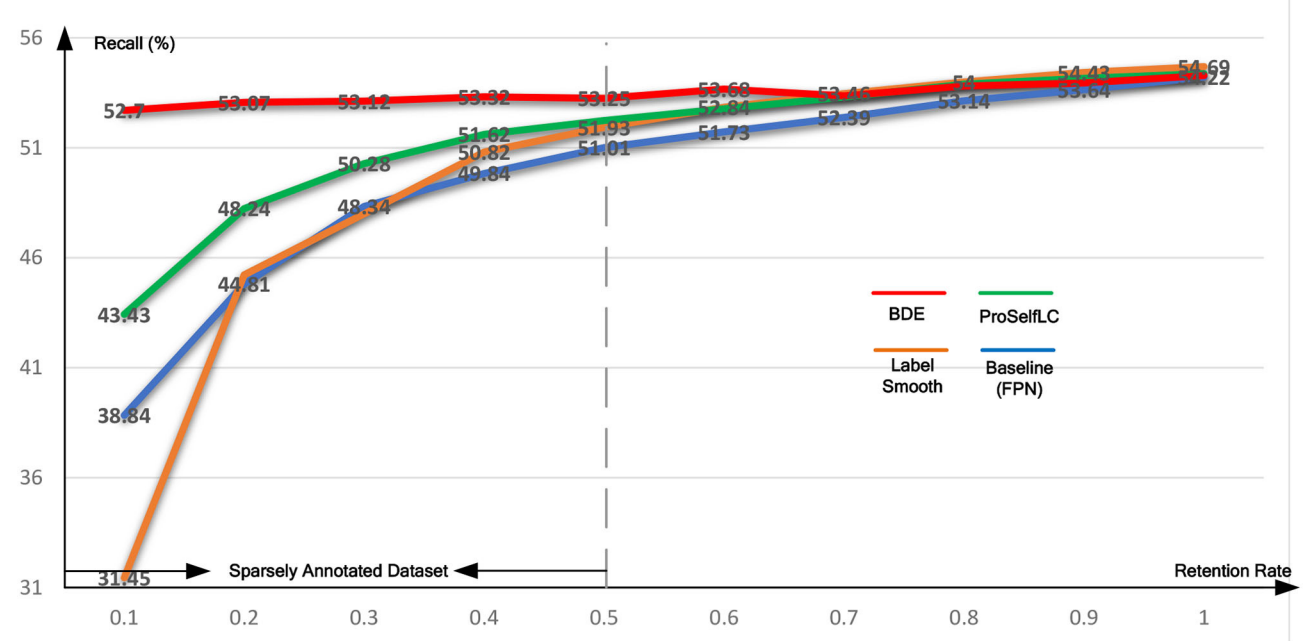

FIGURE 6 | The recall test-results curve of different methods trained on the Ki-67 dataset of different retentive rates. The horizontal coordinate stands for different retentive rates and the vertical coordinate for recall(%).

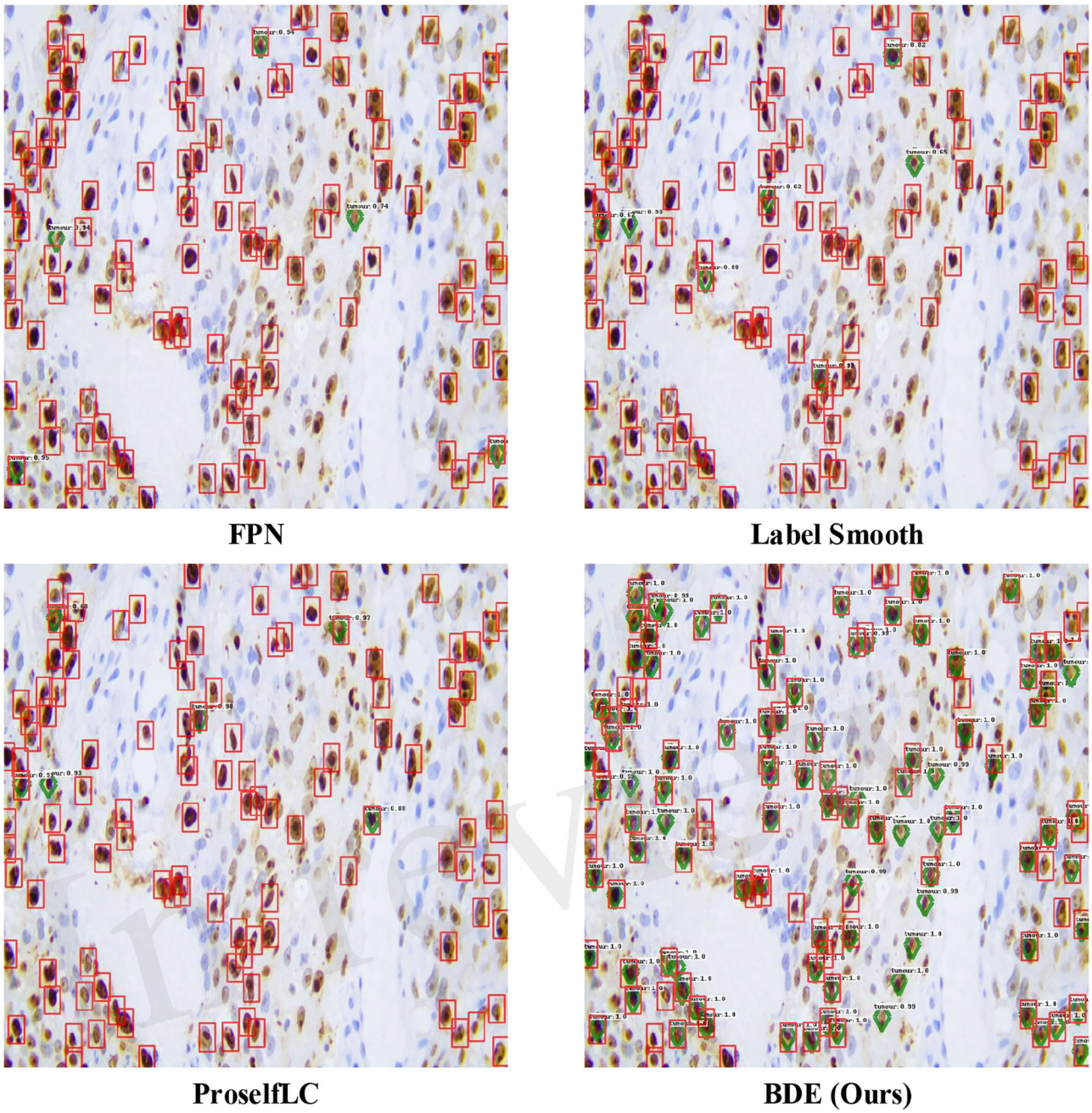

FIGURE 7 | Examples of detection results on the Ki-67 dataset. These results are trained on the dataset with the retentive rate of 0.1 and tested on the test-set. Predictions are drawn in green diamond, and manual annotations are in red boxes. A score threshold of 0.6 is used for display.

layer can be used as an objective evaluation standard for feature extraction ability. Therefore, we analyze the gradient of each layer to investigate why BDE can improve the performance.

5.2. How to Analyze the Gradient

We analyze the mean value of the gradients in each layer of the network by computing the back-propagation *via* the testing loss. Specifically, for a layer indexed by l, whose mean gradient (μ_l) can be computed as follow:

$$\mu_l = \sum_{k=1}^{k=K} \frac{1}{K} \cdot A_{l,k}; \quad \mu_l \in R^1, \tag{14}$$

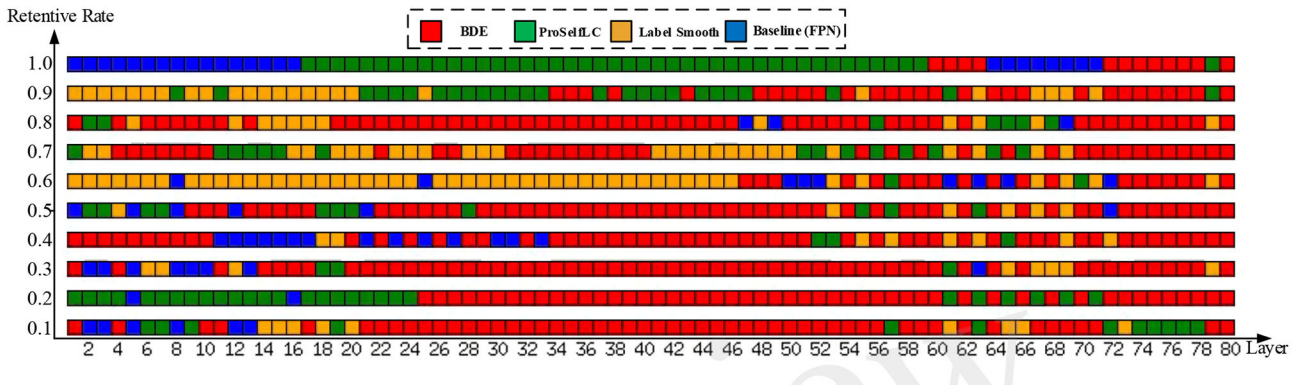

FIGURE 8 | The gradient is visualized on the whole network (80 layers) with different training methods and different retention rate annotations (from 0.1 to 1.0, bottom to top). Each row in this figure contains 80 grids representing 80 layers, and the color of each grid encodes which training method obtains the minimum gradient at a layer, e.g., the red grid representing BDE has the minimum gradient for a specific layer.

in which, K is the number of convolution kernels in the layer indexed by l, and $A_{l,k}$ can be obtained by Equation (15).

$$A_{l,k} = \frac{1}{d \times w \times h} \sum_{i}^{d} \sum_{j}^{w} \sum_{m}^{h} G_{l,k}^{i,j,m}, \tag{15}$$

where $G_{l,k}$ is the gradient of k-th convolutional kernel in the l-th layer. Meanwhile, d, w, and h are the depth, the width, and the height of this kernel. $G_{l,k}$ can be computed by Equation (16).

$$G_{l,k} = \sum_{i}^{N} \frac{1}{N} \cdot |\frac{\partial L_{test}^{i}}{\partial W_{l,k}}|; \quad G_{l,k} \in R^{d \times w \times h}, \tag{16}$$

where L_{test}^{i} represents the loss computed on the i-th testing image, and there are N testing images, and $W_{l,k}$ is the weights of the k-th convolutional kernel in the l-th layer. Further, the gradient represents the direction whether it is positive or negative, so that we perform an absolute operation on the calculated gradient.

5.3. Visualization and Discussion of the Gradient

As shown in **Figure 8**, we visualize the layer-level gradient of the networks (with 80 layers), which are trained on the Ki-67 dataset (retentive rates range from 0.1 to 1), and the gradient is obtained by the testing loss of the Ki-67 dataset. For each layer, we compare whose gradient is trained on different methods. Specifically, a grid with different colors indicates which method can obtain the minimum gradient, e.g., a red grid shows that our approach reduces the test gradient for a particular layer.

We can observe from **Figure 8** when the network is trained on a dataset whose retentive rate below 0.5, BDE improves most of the middle layers (roughly 20–60 layers), which does not seem to happen by accident. Therefore, we can further presume that the generalization performance improvement of the cell detection task is closely related to the middle layers of the network.

6. CONCLUSION

In this study, through theoretical analysis and experimental verification, we identify that the detector trained on sparsely annotated cellular datasets may fall into overfitting due to deviation-loss. In order to address the training limitation, we propose a novel training method, which is utilized to calibrate the deviation-loss based on the cues provided by the density of regression boxes. Extensive experiments demonstrated the strength of BDE to significantly improve the training performance of the cellular detector, even with 90% of annotations missing, the performance of our method is barely affected. Thus, our proposed BDE might enable better and faster development of accurate cellular detection. More importantly, through the visual analysis of the network gradient, we find that the improvement of generalization performance is closely related to the middle layer of the network, which is expected to provide a new theoretical direction for future research.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

AUTHOR CONTRIBUTIONS

HL: conception and design of study. FL, XS, and LH: drafting the manuscript. YK, ZW, and JL: analysis and/or interpretation of data. LY, WY, and QM: acquisition of data. LC and JF: funding acquisition. All authors contributed to the article and approved the submitted version.

FUNDING

This study is supported by the National Natural Science Foundation of China (NSFC grant no. 6207326), and the Natural Science Foundation of Shaanxi Province of China (2021JQ-461, 2020JM-387).

ACKNOWLEDGMENTS

The authors would like to thank the medical team at AstraZeneca China and the technical team at DeepInformatics++ for their scientific comments on this study.

REFERENCES

1. Li C, Wang X, Liu W, Latecki LJ. DeepMitosis: mitosis detection via deep detection, verification and segmentation networks. *Med Image Anal*. (2018) 45:121–33. doi: 10.1016/j.media.2017.12.002
2. Xing F, Su H, Neltner J, Yang L. Automatic Ki-67 counting using robust cell detection and online dictionary learning. *IEEE Trans Biomed Eng*. (2014) 61:859. doi: 10.1109/TBME.2013.2291703
3. Xing F, Cornish TC, Bennett T, Ghosh D, Yang L. Pixel-to-pixel learning with weak supervision for single-stage nucleus recognition in Ki67 images. *IEEE Trans Bio Med Eng*. (2019) 66:3088–97. doi: 10.1109/TBME.2019.2900378
4. Sirinukunwattana K, Raza SEA, Tsang YW, Snead DR, Cree IA, Rajpoot NM. Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images. *IEEE Trans Med Imaging*. (2016) 35:1196–206. doi: 10.1109/TMI.2016.2525803
5. Lewis JS, Ali S, Luo J, Thorstad WL, Madabhushi A. A quantitative histomorphometric classifier (QuHbIC) identifies aggressive versus indolent p16-positive oropharyngeal squamous cell carcinoma. *Am J Surg Pathol*. (2014) 38:128. doi: 10.1097/PAS.0000000000000086
6. Schmidt U, Weigert M, Broaddus C, Myers G. Cell detection with star-convex polygons. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Granada: Springer (2018). p. 265–73.
7. Zhou Y, Chen H, Xu J, Dou Q, Heng PA. IRNet: instance relation network for overlapping cervical cell segmentation. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Shenzhen: Springer (2019). p. 640–8.
8. Xie H, Yang D, Sun N, Chen Z, Zhang Y. Automated pulmonary nodule detection in CT images using deep convolutional neural networks. *Pattern Recognit*. (2019) 85:109–19. doi: 10.1016/j.patcog.2018.07.031
9. Zhang R, Zheng Y, Poon CC, Shen D, Lau JY. Polyp detection during colonoscopy using a regression-based convolutional neural network with a tracker. *Pattern Recognit*. (2018) 83:209–19. doi: 10.1016/j.patcog.2018.05.026
10. Xu M, Bai Y, Ghanem B, Liu B, Gao Y, Guo N, et al. Missing labels in object detection. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. Long Beach, CA (2019).
11. Zhang C, Bengio S, Hardt M, Recht B, Vinyals O. Understanding deep learning requires rethinking generalization. *arXiv preprint* arXiv:161103530. (2016).
12. Li H, Han X, Kang Y, Shi X, Yan M, Tong Z, et al. A novel loss calibration strategy for object detection networks training on sparsely annotated pathological datasets. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Lima: Springer (2020). p. 320–9.
13. Tian Z, Shen C, Chen H, He T. FCOS: fully convolutional one-stage object detection. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul: IEEE (2020).
14. Zhou X, Wang D, Krähenbühl P. Objects as Points. *arXiv*. (2019) *arXiv-1904*.
15. Zhang S, Chi C, Yao Y, Lei Z, Li SZ. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Virtual (2020) p. 9759–68.
16. Huang R, Pedoeem J, Chen C. YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers. In: *2018 IEEE International Conference on Big Data (Big Data)*. Seattle, WA: IEEE (2018). p. 2503–10.
17. Lin TY, Goyal P, Girshick R, He K, Dollár P. Focal loss for dense object detection. *In: Proceedings of the IEEE International Conference on Computer Vision*. Venice: IEEE (2017). p. 2980–8.
18. Ren S, He K, Girshick R, Sun J. Faster r-cnn: towards real-time object detection with region proposal networks. In: *Advances in Neural Information Processing Systems*. Montreal, QC (2015). p. 91–99.
19. Lin TY, Dollár P, Girshick R, He K, Hariharan B, Belongie S. Feature pyramid networks for object detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu, HI (2017). p. 2117–25.
20. Zhang X, Wei Y, Feng J, Yang Y, Huang TS. Adversarial complementary learning for weakly supervised object localization. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. SaltLake City, UT (2018). p. 1325–34.
21. Zhang X, Wei Y, Kang G, Yang Y, Huang T. Self-produced guidance for weakly-supervised object localization. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Munich (2018). p. 597–613.
22. Niitani Y, Akiba T, Kerola T, Ogawa T, Sano S, Suzuki S. Sampling techniques for large-scale object detection from sparsely annotated objects. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Long Beach, CA (2019). p. 6510–8.
23. Yan Z, Liang J, Pan W, Li J, Zhang C. Weakly-and semi-supervised object detection with expectation-maximization algorithm. *arXiv preprint* arXiv:170208740. (2017).
24. Inoue N, Furuta R, Yamasaki T, Aizawa K. Cross-domain weakly-supervised object detection through progressive domain adaptation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT: IEEE (2018) p. 5001–9.
25. Sukhbaatar S, Bruna J, Paluri M, Bourdev L, Fergus R. Training convolutional networks with noisy labels. *arXiv preprint* arXiv:14062080. (2014).
26. Patrini G, Rozza A, Krishna Menon A, Nock R, Qu L. Making deep neural networks robust to label noise: a loss correction approach. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu, HI (2017). p. 1944–52.
27. Müller R, Kornblith S, Hinton GE. When does label smoothing help? In: *Advances in Neural Information Processing Systems*. Vancouver, BC (2019). p. 4694–703.
28. Wang X, Hua Y, Kodirov E, Clifton DA, Robertson NM. ProSelfLC: progressive self label correction for training robust deep neural networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Virtual (2021).
29. Li B, Liu Y, Wang X. Gradient harmonized single-stage detector. In: *Computer Vision and Pattern Recognition*. Salt Lake City, UT (2018).
30. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Las Vegas, NV: IEEE (2016). p. 770–8.
31. Everingham M, Van Gool L, Williams CK, Winn J, Zisserman A. The pascal visual object classes (voc) challenge. *Int J Comput Vis*. (2010) 88:303–38. doi: 10.1007/s11263-009-0275-4

Conflict of Interest: WY, LH, and QM were employed by the company AstraZeneca, Shanghai, China.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

APPENDIX

Overfitting Issues When Datasets Are Sparsely Annotated

Figure A1A in Appendix exhibits the loss curve of a standard object detector trained on the KI-67 dataset at different cellular-level retentive annotation rates. Before 3,000 steps, the detector trained on the datasets with a lower retentive annotation rate leads to a larger loss, which indicates that the deviation-loss dominates the training process. After that, lower retentive annotation rates lead to smaller losses, which indicates that the detector tends to focus on the annotated instances and then drives the overfitting issue. As shown in **Figure A1B** In Appendix, our method can significantly solve the overfitting issue.

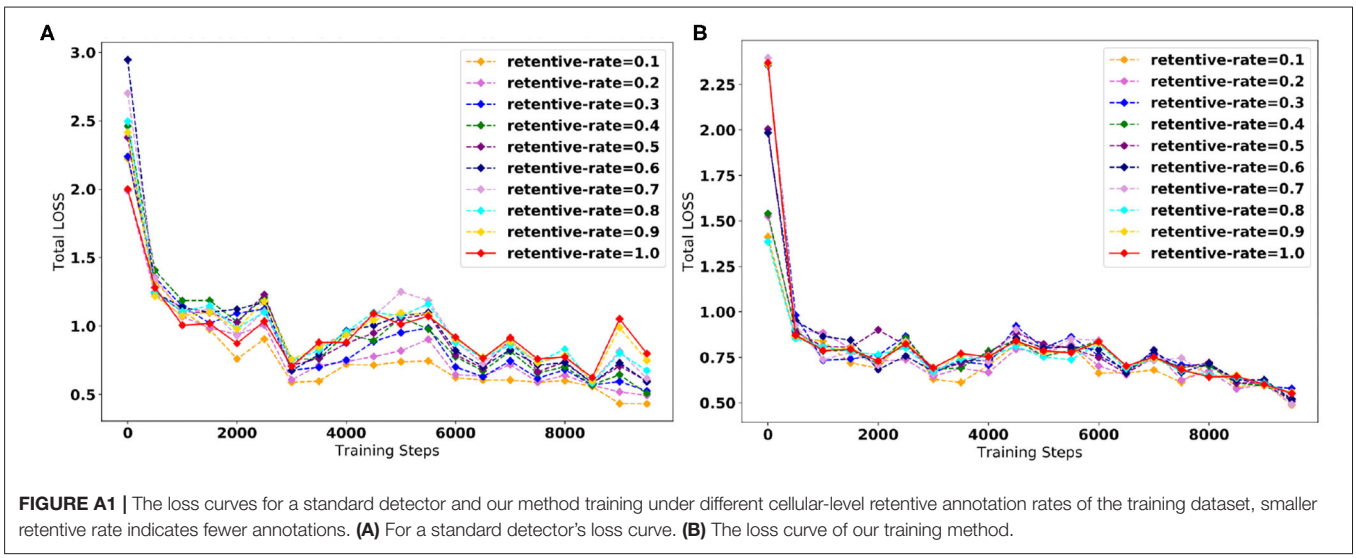

FIGURE A1 | The loss curves for a standard detector and our method training under different cellular-level retentive annotation rates of the training dataset, smaller retentive rate indicates fewer annotations. **(A)** For a standard detector's loss curve. **(B)** The loss curve of our training method.