Neurocomputing 437 (2021) 195–205

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Comparison detector for cervical cell/clumps detection in the limited data scenario

Yixiong Liang, Zhihong Tang, Meng Yan, Jialin Chen, Qing Liu, Yao Xiang *

School of Computer Science and Engineering, Central South University, Changsha 410083, China

ARTICLE INFO

Article history:
Received 23 December 2019
Revised 29 December 2020
Accepted 2 January 2021
Available online 12 January 2021
Communicated by Zidong Wang

Keywords:
Cervical cancer screening
Object detection
Prototype representations
Few-shot learning

ABSTRACT

Automated detection of cervical cancer cells/clumps has the potential to significantly reduce error rate and increase productivity in cervical cancer screening. However, most traditional methods rely on the success of accurate cell segmentation and discriminative hand-crafted features extraction. Recently there are emerging deep learning-based methods which train Convolutional Neural Networks (CNN) to classify cell patches or to detect cells from the whole image. But the former is computationally expensive, while the latter often requires a large-scale dataset with expensive annotations. In this paper we propose an efficient cervical cancer cells/clumps detection method, called Comparison detector, to deal with the limited data problem. Specifically, we utilize the state-of-the-art proposal-based object detection method, Faster R-CNN with Feature Pyramid Network (FPN) as the baseline and replace the classification of each proposal by comparing it with the prototype representation of each category. In addition, we propose to learn the prototype representation of the background category from data instead of manually choosing them by some heuristic rules. Experimental results show that the proposed Comparison detector yields significant improvement on the small dataset, achieving a mean Average Precision (mAP) of 26.3% and an Average Recall (AR) of 35.7%, both improved by about **20**% comparing to the baseline. Moreover, when training on the medium-sized dataset, our Comparison detector gains a mAP of 48.8% and an AR of 64.0%, improving the AR by **5.1**% and the mAP by **3.6**% respectively. Our method is promising for the development of automation-assisted cervical cancer screening systems. Code and datasets are available at https://github.com/kuku-sichuan/ComparisonDetector.

1. Introduction

Cervical cytology is the most common and effective screening method for cervical cancer and premalignant cervical lesions [1], which is performed by a visual examination of cytopathological analysis under the microscope of the collected cells that have been smeared on a glass slide and stained and finally giving a diagnosis report according to the descriptive diagnosis method of the Bethesda system (TBS) [2]. Currently in developed countries, it has been widely used and has significantly reduced the number of deaths caused by related diseases, but it is still unavailable for population-wide screening in the developing countries [3], partly due to the fact that it is labor-intensive, time-consuming and expensive [4]. In addition, it is subjective and therefore has motivated lots of automated methods for the automation of cervical screening based on the image analysis techniques.

Over the past 30 years extensive research has attempted to develop automation-assisted screening methods [5,6]. Most of them try to classify a single cell into various stages of carcinoma, usually involving three steps: cell (cytoplasm and nuclei) segmentation, feature extraction and classification. The performance of these methods, however, heavily depends on the accuracy of the segmentation and the effectiveness of the hand-crafted features.

With the overwhelming success in a broad range of applications such as image classification [7,8], semantic segmentation [9], object detection [10,11] and medical imaging analysis [12–15], CNN has also been applied to cervical cell segmentation and classification [16–19]. The majority of them (e.g. [16]) are trying to take advantage of CNN to improve the segmentation accuracy of cytoplasm and nuclei, but they do not provide the needed segmentation accuracy [17,19]. Once the segmentation error is taken into account, the classification accuracy would drop [18]. To avoid the dependence on accurate segmentation, the patch-based method [19] applies CNN to classify the image patches. However, the extraction of such patches still requires the segmentation of nuclei. The recent work [18] also adopts the patch-based strategy, but

* Corresponding author.
E-mail addresses: yxliang@csu.edu.cn (Y. Liang), zhihongtang@csu.edu.cn (Z. Tang), bryant@csu.edu.cn (M. Yan), yao.xiang@csu.edu.cn (Y. Xiang).

https://doi.org/10.1016/j.neucom.2021.01.006
0925-2312/

during the inference the random-view aggregation and multiple crop testing are needed to produce the final prediction results and thereby is very time-consuming. Very recently there are emerging works [20,21] which try to exploit the contemporary object detection methods [10,11,22] to detect the cervical cytological abnormalities directly and achieve promising results on their private datasets. However, CNN-based object detection methods often need sufficient annotated data to obtain favorable generalization, but for cervical cytological abnormalities detection, collecting large amounts of data with accurate annotation is difficult partially due to the limitation by laws, the scarcity of positive samples and especially the unanimous agreement between cytopathologists [23]. To our knowledge, there is no large-scale datasets public available for this detection task.

To alleviate the limited data problem, in this paper, we propose the named *Comparison detector*, which migrates the idea of *comparison* in one/few-shot learning for image classification [24–27] into CNN-based object detection, for cervical cell/clumps detection. Specifically, we choose the state-of-the-art object detection method, Faster R-CNN [10] with FPN [11], as the baseline model and replace the original parameter classifier with a non-parametric one based on the idea of comparison each proposal with the prototype representations of each category, which are generated from the reference images. Furthermore, instead of manually choosing the reference images of the background category by some heuristic rules for generating the corresponding prototype representations, we propose to learn them from the data directly. We also investigate several important factors including the generation of prototype representations of each category and the design of head model for cervical cell/clumps detection. Our algorithm directly operates on the whole image rather than the cropped patches according to the nuclei and hereby only need one forward propagation for each image, making the inference very efficient. In addition, the proposed method is *flexible* to be integrated into other proposal-based methods.

Due to the lack of public available datasets which are directly dedicated to cervical cell/clumps detection, we collect a small dataset D_s and a medium-sized dataset D_f, on which we evaluate the performance of the proposed Comparison detector. When the model is learned from the small dataset D_s, the performance of our method is significantly better than the baseline model, i.e. Comparison detector obtains a mAP of 26.3% and an AR of 35.7%, but the baseline model only gets a 6.6% mAP and a 12.9% AR. When the model is learned from the medium-sized dataset D_f, our method achieves a mAP of 48.8% while the baseline only gets a 45.2% mAP, while in terms of AR our method improves 5.1% comparing to the baseline.

We summarize our contributions as follows: 1) We propose an end-to-end object detection method called Comparison detector to deal with the limited data problem in cervical cell/clumps detection; 2) We propose a strategy to directly learn the prototype representations of background and 3) Our method performs much better than the baseline on both small and medium-sized dataset and has the potential applications to the real automation-assisted cervical cancer screening systems.

2. Related work

2.1. Cervical cell segmentation, classification and detection

Traditional cytological criteria for classifying cervical cell abnormalities are based on the morphological changes of cells and cytoplasm, therefore there are numerous works focusing on the segmentation of cell or cell components (nuclei, cytoplasm) [28–30]. Although significant progress has been achieved recently, the segmentation of cell or cell components remains an open problem due to the large shape and appearance variation between cells [18,30,17]. Meanwhile, according to TBS rules [2], a large number of hand-crafted features are designed to describe the shape, texture and appearance characteristics of the nucleus and cytoplasm [31,6] for final classification. However, as mentioned above, the extraction of those engineered features depends on the accurate segmentation of cell or cell components. To reduce the dependency on accurate segmentation and hand-crafted feature extraction, the CNN is used to learn the features for classification recently [19,32–34]. DeepPap [18] trains a CNN to classify the cell patches centered on the nucleus centroid, but during the inference stage, DeepPap uses the random-view aggregation and multiple crops testing, which is time-consuming. Other complex CNNs such as VGG network [35] are also used for classification of cervical lesion [32]. However, all these methods only aim to improve the classification accuracy of single cell and hence when processing an image, the number of inference is proportional to the number of cell patches, which is computationally expensive. Very recently, object detection-based methods are proposed for cervical cytology analysis [20,21]. In [20], the YOLOv3 [22] is migrated to detect cervical cells/clumps and an additional classification network is trained to classify the hard objects to improve the accuracy. Due to the introduction of additional classification network, it is not an end-to-end method and is often time-consuming. In [21], the Faster R-CNN [10] and RetinaNet [36] are trained to detect 7 kinds of lesion cells. However, these detection-based methods require sufficient annotated data which is difficult to be obtained, partially due to the limitation by laws and the scarcity of positive samples, etc.

2.2. CNN-based object detection

The Overfeat [37] makes the earliest efforts to apply CNN for object detection and has achieved a significant improvement when compared to the best methods at that time which were based on the hand-crafted features. Since then, a lot of CNN-based methods [38,10,39–43,22] have been proposed for high-quality object detection, which can be roughly classified into two categories: object proposal-based and proposal-free. The road-map of proposal-based methods starts from the notable R-CNN [39] and is improved by Fast R-CNN [40] in an end-to-end manner and by Faster R-CNN [10] to quickly generate object regions, which has motivated a lot of follow-up improvements [11,41] in terms of accuracy and speed. The proposal-free methods [42,22] directly predict the bounding boxes without the proposal generation step. Generally, the proposal-free methods are conceptually simpler and much faster than the proposal-based methods, but the detection accuracy is usually behind that of the proposal-based methods [43]. Here we choose Faster R-CNN [10] with FPN [11] as our baseline model since it is still the state-of-the-arts and has great flexibility.

2.3. Few-shot learning

2.3.1. Few-shot classification

One/few-shot learning is a task of learning from just one or a few training samples per class and has been extensively discussed in the context of image recognition and classification [24–26]. Recently significant progress has been made for one/few-shot learning tackled by meta-learning or learning-to-learn strategy, which can be roughly divided into three categories: metric-based, memory-based and optimization-based. The metric-based methods [24–27] learn to compare the query image with support set images. The memory-based method [44] exploits the memory-augmented neural network to quickly store and retrieve sufficient information for each classification task, while the

optimization-based methods [45,46] aim to learn a base-model which can be fine-tuned quickly for a new classification task. Recently graph-based few-shot learning comes into fashion [47,48]. It mainly uses its flexibility to quickly establish the relationship between features. All these works only tackle image classification tasks, while our method aims at the detection.

2.3.2. Object detection with limited data

Most prior works on object detection with limited labels use semi-/weakly-supervised methods or few-example learning [49] to make use of abundant unlabeled data, whereas in limited data regime there are a few works focusing on using few-shot learning to address this problem [50,51]. Kang et al. [51] decomposes the training into base-model learning and meta-model learning and trains a meta-model to re-weight the features extracted by the base-model to assist novel object detection. However, the training of base model still needs abundant annotated data for base classes. RepMet [50] introduces a metric learning-based sub-network architecture to learn the embedding space and distribution of the training categories without using external data. However, RepMet involves an alternating optimization between the external class distribution module learning and net parameters updating, whereas our solution is a clean, single-step training framework.

3. Comparison detector

3.1. Basic architecture

The proposed Comparison detector is based on proposal-based detection framework which often consists of a backbone network for feature extraction, a RPN for generating proposals and a head for the proposal classification and bounding box regression. Here we choose Faster R-CNN [10] with FPN [11] as our baseline. We decouple the regression and classification in the head and replace the original parameter classifier with a comparison-based classifier which introduces an inductive bias, i.e. the within-class distance is less than the between-class in the features space, into the model and henceforth reduces the complexity of the model and mitigates the generalization issue with small dataset to some extent [52].

The framework of our Comparison detector is depicted in Fig. 1, which is divided into three stages. As shown in Fig. 1, at the first stage, Comparison detector generates features both for the reference images and the whole image by the FPN backbone [11], without any extra models to encode the reference images. Assume that there are K foreground categories and for each foreground category, there are n reference images with L levels pyramid features. Let F^l_{kj} be the j-th image with k-th categories' prototype representation of the l-level pyramid features, which can be calculated as follows

$$F^l_{kj} = F^l(R_{kj}), \tag{1}$$

where $F^l(\cdot)$ denotes the feature extraction function (i.e. the output of ROIPool [10] or ROIAlign [41]) at l-th level and R_{kj} is the j-th reference image of class k. Similarly, the feature P_m of the m-th object proposal x_m is generated by

$$P_m = F^l(x_m). \tag{2}$$

It should be pointed out that categories in training and test set are same in our settings unlike one/few-shot learning.

The second stage is to generate the prototype representations of each category from the reference images' pyramid features as shown in Fig. 2(a). We first obtain the k-th class pyramid feature F^l_k by average operation

$$F^l_k = \frac{1}{n}\sum_{j=1}^{n} F^l_{kj}, \tag{3}$$

and then aggregate them into the final prototype representation F_k for class k by the aggregation function $S(\cdot)$,

$$F_k = S(\{F^l_k\}). \tag{4}$$

The third stage is the head model for classification and bounding box regression (Fig. 3(c)), consisting of a few convolutional (*Conv*) and fully connected (*FC*) layers. Let $d(F_k, P_m)$ be a metric function to compute the distance between the m-th proposal feature P_m and prototype representation of the k-th category F_k (we will discuss this function in the experimental section). Each proposal's posterior probability p_k and bounding box regression b_k can be obtained by

$$p_k = \frac{e^{-d(F_k,P_m)}}{\sum_i e^{-d(F_i,P_m)}}, \tag{5}$$

$$b_k = b(F_k, P_m), \tag{6}$$

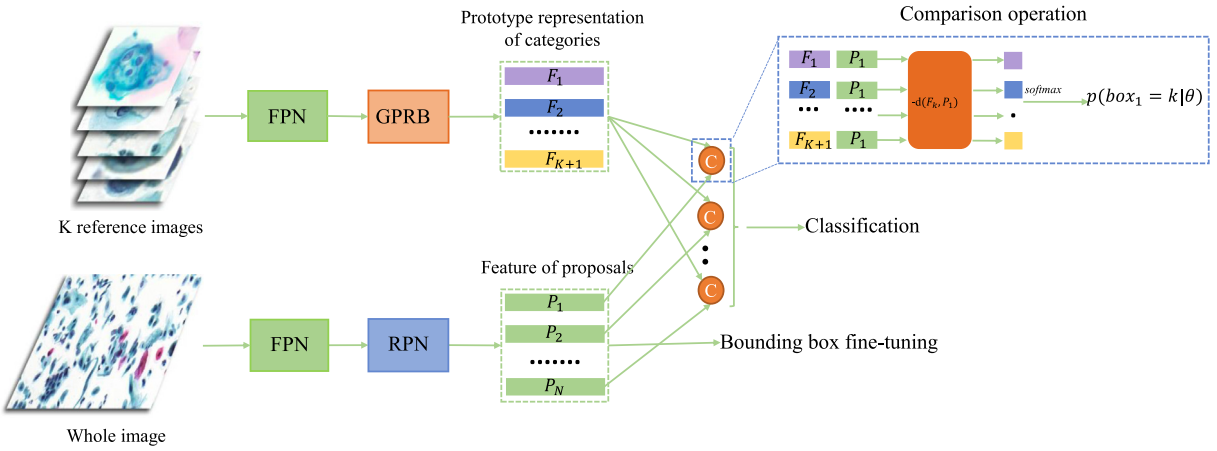


Fig. 1. The overall structure of Comparison detector. First, $n \times K$ reference images are fed into FPN backbone to obtain the features, which are then fed into the **Generating Prototype Representation Block (GPRB)** to generate the prototype representations for each category, where K is the number of categories excluding the background and n is number of images in each category. At the same time, we can obtain the features of proposals from the whole image through FPN and region proposal network (RPN). It should be noted that the FPN backbone is shared. By comparing the features of each proposal with all prototype representations, we can get the category of this proposal. Only the feature of proposals are used to fine-tune the bounding box.

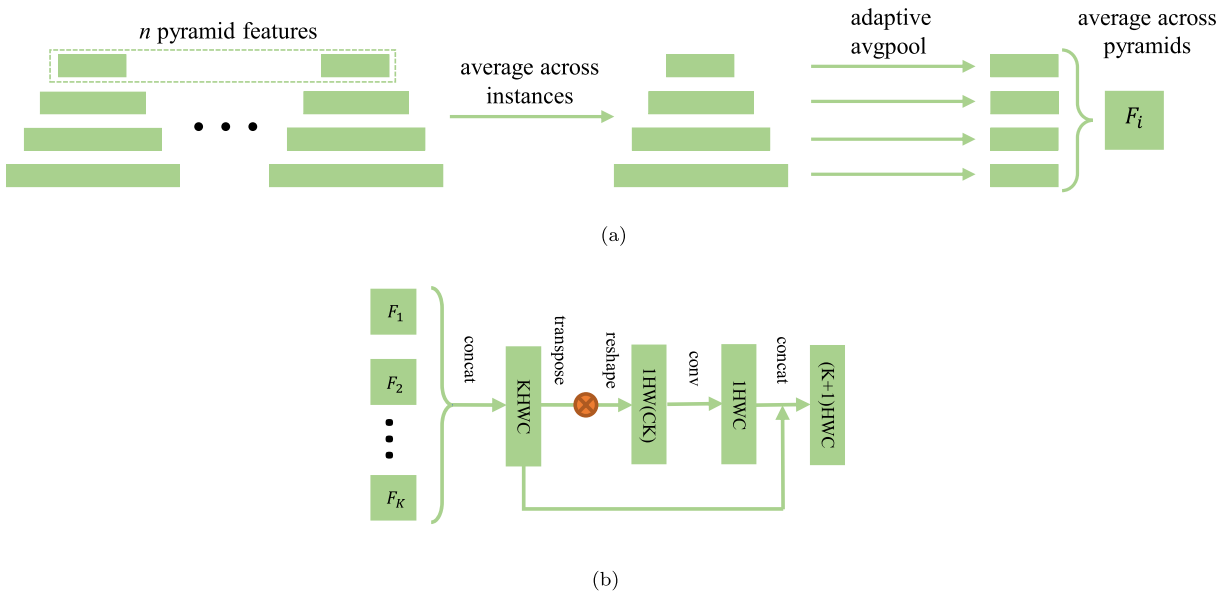


Fig. 2. *GPRB.* (a) The process of generating one foreground prototype representation from pyramid features of reference image. (b) The learning of background prototype representation from all K-class foreground prototype representations. Here W, H and C are the width, height and channels of prototype representation of each category respectively.

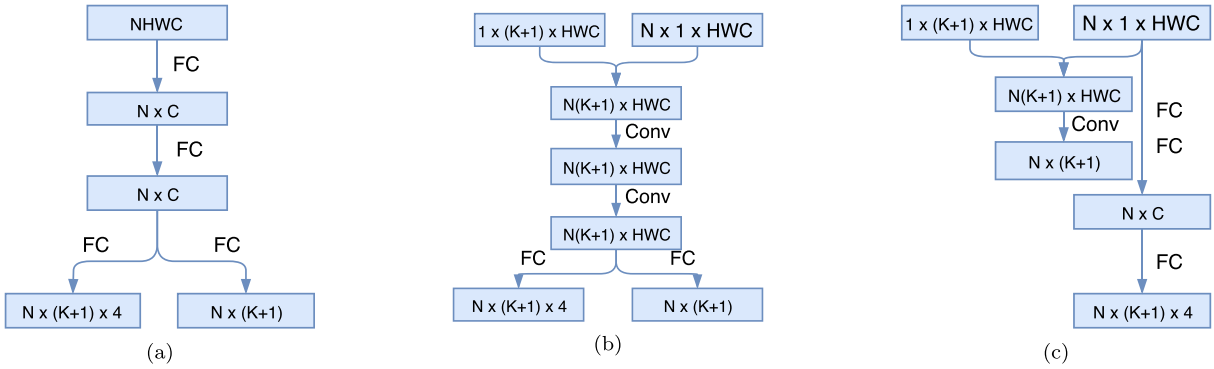


Fig. 3. *The head for classification and regression.* (a) The detection head of baseline. (b) The share module in Comparison Detector. (c) The independent module in the Comparison Detector.

where $b(\cdot,\cdot)$ denotes the bounding box regression function. The softmax function in Eq. (5) is based on the comparison between m-th proposal feature P_m and prototype representation of the k-th category F_k and therefore we denote the resulting classifier as comparison classifier. The rest is the same as Faster R-CNN with FPN [11].

Similar to Faster R-CNN [10], the objective function of Comparison detector is to minimize the total loss consisting of the RPN loss L^{rpn} and head loss L^{head}, both consisting in bounding box regression loss and classification loss. In our method, the head loss L^{head} is

$$L^{head} = \lambda L^{head}_{cls} + L^{head}_{reg}, \tag{7}$$

where λ is the parameter to balance the head classification loss L^{head}_{cls} and head box regression loss L^{head}_{reg}. All classification losses are cross-entropy loss and bounding box losses are ℓ_1-smooth loss [40].

3.2. Generating prototype representation block (GPRB)

3.2.1. Generating prototype representations of foreground categories

As mentioned above, we use n reference images for each foreground category to generate pyramid features and then obtain the corresponding prototype representation by the aggregation function $S(\cdot)$. For simplicity, we directly resize each pyramid feature which is generated by reference images to a fixed size, and then calculate prototype representation by averaging operation, i.e.

$$S(\{F^l_k\}) = \frac{1}{L}\sum_l r(F^l_k, s), \tag{8}$$

where $r(\cdot,\cdot)$ is bilinear interpolation function and s is the size of final features.

3.2.2. Learning the prototype representations of background category

There are many negative proposals generated by RPN, so R-CNN [39] adds an additional background category to represent them. In our Comparison detector, we need to select n reference images for each category and therefore also need to select reference images for background category. However, because of the overwhelming diversity, selecting reference images of background is exceedingly difficult. We notice that a background

region is considered as a proposal indicating that it has certain similarity with the foreground categories. Therefore, we can infer that its prototype representation is a combination of different foreground categories in the most case. So we propose to learn its prototype representation of background category from the foreground categories' prototype representations. Specifically, we first transpose the channels and reshape the tensor and then we use a simple 1×1 convolution operation to generate the prototype representations of background category. Finally, we concatenate all prototype representations together, as shown in Fig. 2(b).

3.3. The head for classification and regression

As shown in Fig. 3(a), the baseline model's head first transforms the proposal features and then followed by two branches, one for classification and the other for the offset predicting of the bounding box. For our Comparison detector, due to the introduction of the reference images, we need to re-organise the head. There are two choices according to whether sharing the features between bounding box regression and classification. One is that bounding box regression branch and classification branch are not shared, as shown in Fig. 3(c). Unlike the baseline model, the classifier and bounding box regressor in the head of Comparison detector are independent (independent module). And the bounding box regressor only uses the features of proposal to predict the offset of the bounding box, i.e.

$$\begin{aligned} d(F_i, P_m) &= FC(m(F_i, P_m)), \\ b(F_i, P_m) &= FC(FC(FC(P_m))), \end{aligned}$$

where $m(F_i, P_m) = Conv_3(Conv_1(|F_i - P_m|^2))$.[1] Another choice is to share features for both classification and regression, as shown in Fig. 3(b), which means

$$b(F_i, P_m) = FC(m(F_i, P_m)).$$

We call this method as shared module.

3.4. Strategies for selecting reference images

In our Comparison detector, we need to choose the reference images for each foreground category. An intuitive way is to select them according to the Bethesda atlas [2]. However, there are significant differences between the given atlas and our data due to the variations of the preparation and digitization of slide. Hence we resort to other feasible data-driven alternatives. We randomly select about 150 instances of each category from the training sets. The shortest side of these instances is greater than 16 pixels. In this manner we get a total of 1,560 instances as candidate reference images from training sets, from them we can select suitable instances in these objects as our reference images.

There are two possible ways. The first is to randomly choose several instances of each category as the reference images. The second is to first map all 1,560 objects into the feature space through the ImageNet pre-trained model [8] to get the features of each object and then use t-SNE [53] for feature dimension reduction (Fig. 4). Based on the results of t-SNE, we empirically obtain the number of clusters in each category. Then we use it as the parameter for K-means. Finally, based on the result of K-means, we choose the instances which are the closest to the center of clusters as reference images.

[1] The subscript denotes kernel size of convolution.

Fig. 4. *t-SNE* visualization of reference images. Same background color represents they belong to the same category. After t-SNE learning, the images of same category will cluster together, so that the number of clusters in each category can be observed.

4. Experimental results and analysis

4.1. Experimental setups and implementation

To our knowledge, there are no publicly available benchmarks for cervical cell object detection in the community, thus we establish a database[2] consisting of 7,410 cervical microscopical images which are cropped from the whole slide images (WSIs) obtained by Pannoramic MIDI II digital slide scanner. The corresponding specimens are prepared by Thinprep methods stained with Papanicolaou stain. Conforming to TBS categories [2], 48,587 object instance bounding boxes were labeled by experienced pathologists which belong to 11 categories, namely ASC-US (ascus), ASC-H (asch), low-grade squamous intraepithelial lesion (lsil), high-grade squamous intraepithelial lesion (hsil), squamous-cell carcinoma (scc), atypical glandular cells (agc), trichomonas (trich), candida (cand), flora, herps, actinomyces (actin). Fig. 5 shows some examples of each category in our database. We randomly divide the dataset into training set D_f which contains 6,666 images, test set which contains 744 images for experiment. To verify the performance of Comparison detector on the small dataset, we build a small-sized dataset D_s via randomly selecting 762 images from D_f while try to keep the distribution of each foreground category in D_s is as close as the one in D_f. The distributions of categories in small dataset D_s, median-sized dataset D_f and test set are shown in the Fig. 6.

In following experiments, we use ResNet50 [8] with FPN [11] as backbone network. Parameters are initialized with the pre-trained model on ImageNet. For the reference images, we rescale them to size of 224×224 to meet the requirement of the pre-trained model. During the training, the initial learning rate is 0.001, and then decreased by a factor of 10 at 35-th and 50-th epoch. Training is stopped after 60 epochs and we adopt early stopping tricks. The

[2] Codes and datasets are available at https://github.com/kuku-sichuan/ComparisonDetector.

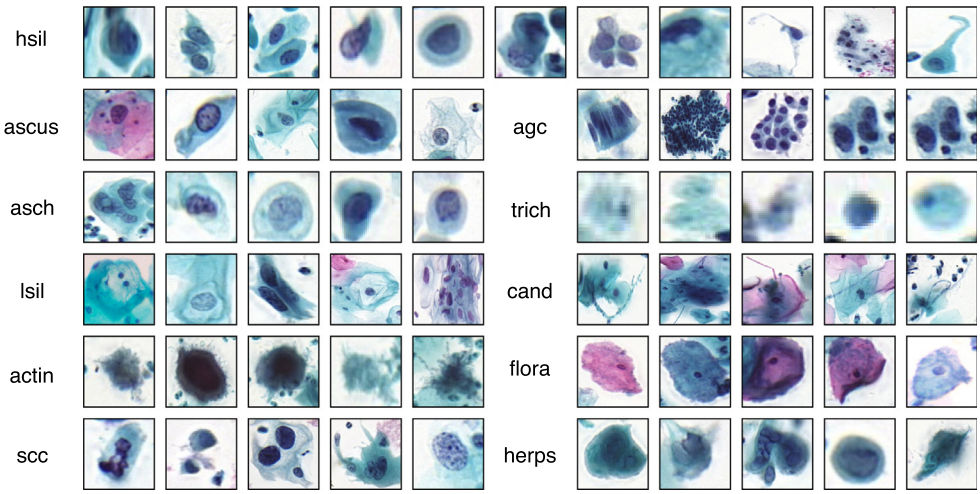


Fig. 5. Selected samples of abnormal cervical cells/clumps. The same category of cells have multiple features due to the presence of cell cluster and mucus. Some subsets of different categories have extremely similar features.

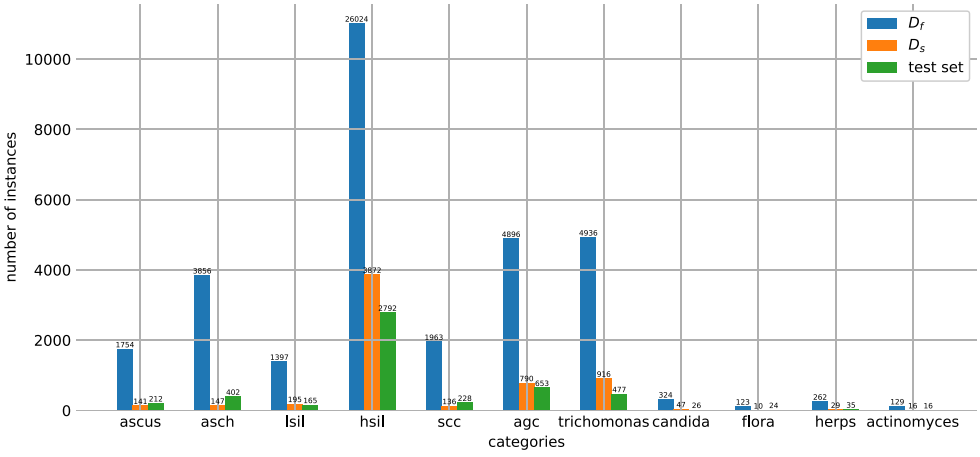


Fig. 6. The distribution of categories on D_s, D_f and test set.

mini-batch size is set to 2 and the weight decay and the momentum are 0.0001 and 0.9, respectively.

For the cervical cell images, annotators are prone to take a higher threshold when label the objects due to the low discrimination between them. In addition, multiple nearby objects with the same category will be marked as one, so the performance of methods can not be well reflected by mAP [39]. Therefore, we adopt both mAP (mean Average Precision) and AR (Avearage Recall) as evaluation metrics. Particularly, if one method achieves same mAP and higher AP when comparing to the other, it is considered to be superior to the other.

4.2. Ablation studies

We run a number of ablations to analyze Comparison detector. An elementary summary of results can be found in Table 1, where all methods are trained on the D_f while the detection performance is based on the test set.

Learning the prototype representation of background. In order to compare the effect of learning background, we randomly select some background images from the proposals to obtain the features of background (model A). As shown in Table 1, it only obtains a mAP of 31.4% while the model of learning the prototype representation of background category (model B) gains a mAP of 34.1%. Notice that Model B has only a few more parameters than Model B, which validates the effectiveness of our background pro-

Table 1
Ablation studies. We train on D_f and evaluate on the test set. The ℓ_2-distance is used as the metric in comparison classifier.

Model	Learning background	Independent mode	All pyramid features	Bounding box regression	mAP	AR	Params
A		√	√	√	31.4	49.3	41.29 M
B	√	√	√	√	34.1	**53.3**	41.99 M
C	√	√		√	32.7	50.8	41.99 M
D	√		√	√	**41.0**	51.3	42.00 M
E	√			√	38.9	49.8	42.00 M
F	√	√	√		37.7	51.1	**28.10** M

Table 2
Comparative performance with or without the balanced bounding box regression loss and classification loss. The metric outside of the bracket is mAP while the one in bracket is AR.

Model	B	D	F
wo/balancing	34.1(53.3)	41.0(51.3)	37.7(51.1)
w/balancing	**43.7(60.7)**	**43.5(58.9)**	**38.8(52.3)**

Table 3
Different comparison classifier. The numbers in the brackets denote the result after balancing the loss.

Comparator	ℓ_2-distance	Parameterized ℓ_2-distance	concat
mAP	34.1(43.7)	38.2(**44.5**)	40.7(42.5)
AR	53.3(60.7)	56.8(**61.6**)	49.1(58.1)

totype representation learning method. Furthermore, because the background prototype representation is learned from the prototype representation of foreground categories, the gradient propagation will also have promising effect on the optimization of other prototype representation. In order to make sure whether this effect is beneficial, we stop gradient propagation at the fork position in Fig. 2(b). The performance declines, with a mAP of 33.0% and an AR of 52.6%.

Prototype representations of foreground categories. In our approach, as shown in Eq. (8), we use all pyramid features to generate prototype representation of each category. Another choice is to only use the last level pyramid features as the category of prototype, i.e. $S(\{F_k^l\}) = F_k^5$, resulting in the model C. As shown in Table 1, model C only gets a mAP of 32.7%, which is lower than model B by 1.4%. Similarly, in Table 1 the model D and model E have the same number of parameters and the only difference between them is whether to use all pyramid features. When using all pyramid features, model D achieves better performance than model E both in terms of mAP and AR. All these results show that using all pyramid features will benefit the final performance, in that it can combine features of multiple scales, which not only fuses the semantics information of the high-level layers and detailed information of the low layers, but also takes objects of different size into account.

Head bounding box regression vs. head classification. As mentioned before, in independent module, the box regression function $b(\cdot,\cdot)$ is the same as baseline model because our experiments show that removing one layer will make the result worse. The model B corresponds to the independent module while model D is shared module in Table 1, which shows that shared module performs much better than independent module. Furthermore, we drop the operation of bounding box regression in the head (model F) which reduces the number of parameters greatly. It's weird that without the head bonding box regression (i.e. only using the RPN regression results), model F gets a mAP of 37.7% which performs slightly better than model B. This phenomenon goes against common sense that performing bounding box regression twice is often better than just once. We empirically conjecture that the importance of classification should be more important than bounding box regression [15]. So we adjust the weight coefficient λ in the Eq. (7) to balance the classification loss and bounding box regression loss in head. Here we select $\lambda = 5$. The results are shown in Table 2. After balancing these two kinds of loss, the performance is greatly improved.

Table 5
Comparison of different the reference images selection strategies.

Method	Fixed mode	Random mode	t-SNE
mAP	44.5	42.8	**45.9**
AR	61.6	61.0	**63.5**

Distance metrics. We evaluate three distance metrics in the comparison classifier. The first is the well-known ℓ_2-distance which means $d(P_m, F_k) = M(|F_k - P_m|^2)$, here $M(\cdot)$ represents averaging function. Instead of the predefined one, we also try to learn the metric. The second is the parameterized ℓ_2-distance, i.e. $d(P_m, F_k) = Conv_7(|F_k - P_m|^2)$. The last one, following [27], we try to concatenate P_m and F_k in depth and then learn the metric function. We also test the balancing between the two branches in head and the corresponding results are shown in Table 3, where the numbers shown in brackets are obtained by setting $\lambda = 5$. From it we can see that combing the balancing tricks and the parameterized ℓ_2-distance, we can obtain the best performance with a mAP of 44.5%. Combining with the results shown in Table 2, we find that it is universal that the balancing trick can improve performance, so we adopt this trick in all the following experiments.

Strategies for selecting reference images. We further investigate the scheme of randomly choosing reference images. We try two methods, and the first is to randomly choose 3 instances of each category (this number is limited by GPU's memory) as the reference images (`fixed mode`) while the second one is to randomly select 5 candidates of each category in those objects. Then the model randomly selected three of the five candidates as templates in training, but five in testing (`random mode`). In addition, we also choose reference images by applying t-SNE and K-means. During the training of t-SNE, we adopt the following parameters setting, i.e. the hyper-parameters are 30 for perplexity, 1 for learning rate, and 10 for label supervision. Through the t-SNE visualization, we can empirically obtain the number of clusters for each category, which is then set as the `n_clusters` parameter for K-means. The number of clusters is shown in Table 4. Finally, according to the results of K-means, we choose the instance which is the closest to the center of clusters as reference images. As show in Table 5, by applying t-SNE and K-means, we can obtain a mAP of 45.9%, while the one of `fixed mode` and `random mode` is 44.5% and 42.8%, respectively. It indicates that the selection reference images via t-SNE and K-means can boost the performance.

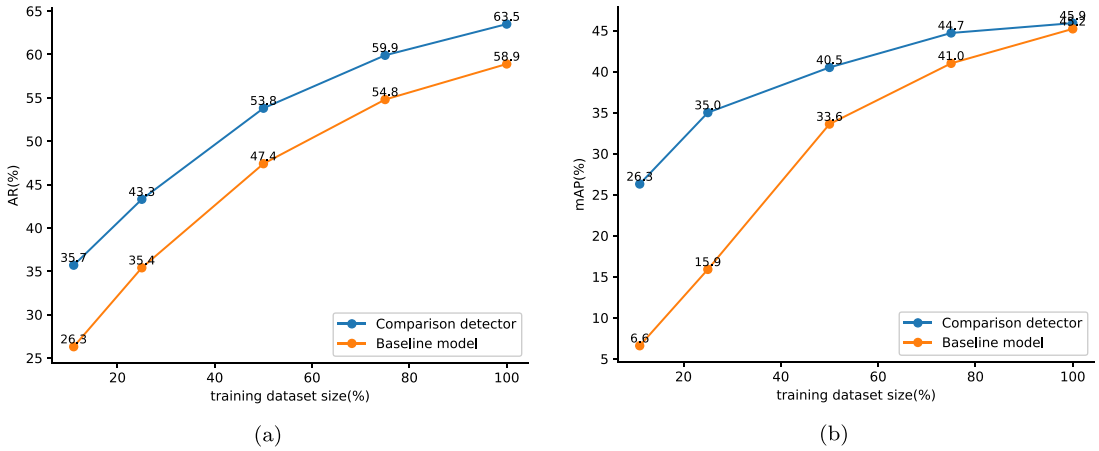
Performance on small training dataset. To evaluate the performance of our method on small training dataset, we also train the model on D_s and the comparative results are shown in Table 6. When training on the median-sized dataset D_f, Comparison detector only gets a **0.7** mAP improvement and a 4.6 AR improvement. As described in Section 4.1, some correct predictions may be identified as false positives. Therefore, there is a distinct increase in AR, but little improvement in mAP. However, when training on the D_s, the performance of Comparison detector is completely superior to the one of the baseline. It achieves a mAP of 26.3% which indicates our method alleviates the small training size problem to some extent.

Table 4
Number of clusters for each category.

Category	ascus	asch	lsil	hsil	scc	agc	trich	cand	flora	herps	actin
Number of clusters	3	4	2	4	2	3	1	2	2	4	1

Table 6
The results of learning on different size datasets. We regard Faster-RCNN with Feature Pyramid Network as baseline model. The performance of Comparison detector and baseline model with training on different datasets are shown in the following.

Method	dataset	AR	mAP	ascu	asch	lsil	hsil	scc	agc	trich	cand	flora	herps	actin
Baseline model	D_s	12.9	6.6	11.0	2.0	23.7	21.6	0.0	3.5	0.0	11.5	0.0	0.0	0.0
Comparison detector	D_s	**35.7**	**26.3**	10.5	1.7	42.8	32.3	0.8	40.5	37.5	24.1	6.9	45.0	46.6
Baseline model	D_f	58.9	45.2	27.2	6.7	41.7	35.3	18.6	57.3	46.7	72.2	57.3	83.0	51.4
Comparison detector	D_f	**63.5**	**45.9**	27.4	6.7	41.7	40.1	21.8	54.5	45.0	65.5	63.5	68.1	70.5



(a) (b)

Fig. 7. Comparative performance with different training size. (a) AR. (b) mAP.

Table 7
The performance of model with different β.

β	0	0.3	0.5	0.7	1
mAP	45.2	45.6	**48.8**	42.0	45.9
AR	58.9	59.8	**64.0**	58.0	63.5

Improved Comparison detector. Generally speaking, when the annotated data is sufficient, the performance of methods based on few-shot learning does not always perform as well as traditional supervised learning. To verify this, we compare the performance of Comparison detector with the baseline on different training sample size. As shown in Fig. 7, as the number of training images increases, the performance of both models is improved, but the growth rate of performance to Comparison detector is not as high

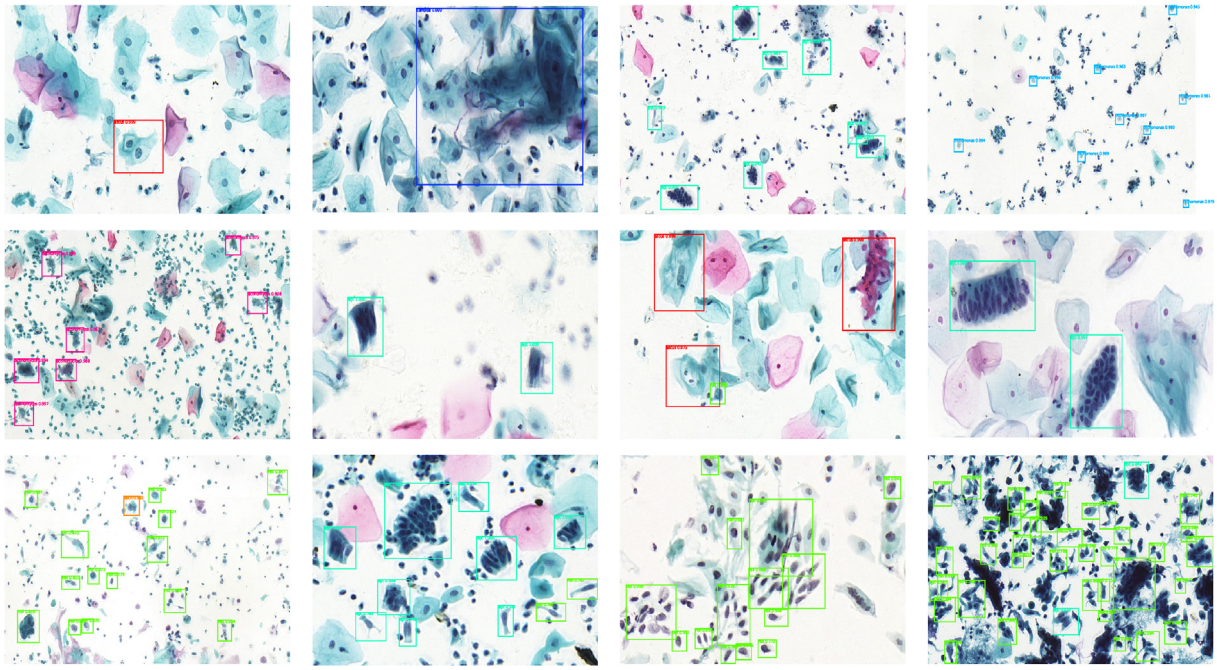


Fig. 8. Detection results on the test set using the Improved Comparison detector. These results are based on ResNet-50 with FPN as the backbone, achieving a mAP of 48.8% and AR of 64.0%.

Table 8
The performance of different methods. The data in brackets represent the time complexity which has been optimized. FPS represents Frame Per Second. "*" indicates Comparison detector with an additional linear classifier.

Model	Faster R-CNN with FPN [11] (baseline)	RetinaNet [36]	Comparison detector	*Comparison detector
mAP	45.2	45.2	45.9	**48.8**
AR	58.9	56.6	63.5	**64.0**
FPS	7.81	**8.80**	7.49	7.48

as baseline model. Therefore, in order to make Comparison detector more flexible, in addition to our comparison-based classifier, we add a linear classifier along with bounding box regression, which is the same as baseline model. The final classification result is the weighted average of the comparison classifier and the linear classifier as follows

$$p_k = \beta p_{kc} + (1-\beta)p_{kl}, \tag{9}$$

where p_{kc} and p_{kl} are the results of comparison classifier and linear classifier respectively. We also combine the two classification losses in the same way when training the model. Actually, we expand the classification loss of head L_{cls}^{head} in Eq. (7) as

$$\lambda L_{cls}^{head} = \beta L_c + (1-\beta)L_l, \tag{10}$$

where L_c and L_l are both cross entropy loss and represent the classification loss of comparison classifier and linear classifier respectively. When β is set to 1, it is equivalent to original Comparison detector while when β equals to 0, it is degenerated into the baseline. We adjust the value of β so that the model can combine the advantages of these two kinds of classifiers. As shown in Table 7, when the beta is equal to 0.5, the model achieves the best performance. The detection results of some samples are shown in Fig. 8.

4.3. Comparisons with state-of-the-art methods

Our method is based on the state-of-the-art Faster R-CNN [10] with FPN [11], but redesigns the head sub-networks. In order to better reflect the effect of our method, we also compare our method with RetinaNet [36] which is another state-of-the-arts object detector. The comparative results are shown in Table 8, where we also list the inference speed of different methods. When evaluating the inference speed of our method, we pre-compute the prototype representation of each category which can avoid repeated calculation. From Table 8, we can see that our methods perform better than Faster R-CNN [10] and RetinaNet [36] both in terms of mAP and AR, but the inference speed are slightly slower than them due to introduction of the extra parameters in classification head.

5. Conclusion

In this work, we propose to apply contemporary CNN-based object detection methods for automated cervical cell/clumps detection. To deal with the limited training dataset, we develop the comparison classifier based on the comparison with the reference images of each category, which can be used as a plug-in module in proposal-based object detectors. Instead of manually choosing the reference images of the background by some heuristic rules, we present a scheme to learn them from the data directly. We also investigate several important modules including the generation of prototype representations of each category and the design of head model for cervical cell/clumps detection. Experimental results show that compared with the baseline, our method improves the mAP by **19.7**% and the AR by **22.8**% when trained on the small training dataset. After combining linear classifier and comparison classifier in the model, our method improves the mAP by **3.6**% and the AR by **5.1**% when trained on the medium-sized training dataset. It should be noticed that our method directly operates on the whole image rather than the extracted patches based on the nuclei and hereby only needs one forward propagation for each image, making the inference extremely efficient. In addition, the proposed method is *flexible* to be integrated into other proposal-based methods.

CRediT authorship contribution statement

Yixiong Liang: Conceptualization, Methodology, Investigation, Writing - original draft, Writing - review & editing. **Zhihong Tang:** Conceptualization, Methodology, Software, Validation, Formal analysis, Writing - original draft, Writing - review & editing. **Meng Yan:** Investigation, Data curation, Writing - original draft. **Jialin Chen:** Data curation, Writing - original draft. **Qing Liu:** Writing - original draft, Writing - review & editing. **Yao Xiang:** Resources, Supervision, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research was supported by the National Natural Science Foundation of China (Nos. 61672542 and 61972419), Natural Science Foundation of Hunan Province of China (No. 2020JJ4120), and Changsha Science and Technology Project (kh1902014).

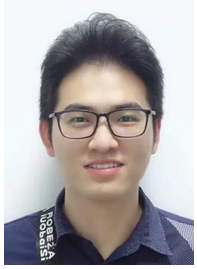
References

[1] E. Davey, A. Barratt, L. Irwig, S.F. Chan, P. Macaskill, P. Mannes, a.M. Saville, Effect of study design and quality on unsatisfactory rates, cytology classifications, and accuracy in liquid-based versus conventional cervical cytology: a systematic review, The Lancet 367 (9505) (2006) 122–132..
[2] R. Nayar, D.C. Wilbur, The Bethesda System for Reporting Cervical Cytology: Definitions, Criteria, and Explanatory Notes, Springer, 2015.
[3] D. Saslow, D. Solomon, H.W. Lawson, M. Killackey, S.L. Kulasingam, J. Cain, F.A. Garcia, A.T. Moriarty, A.G. Waxman, D.C. Wilbur, et al., American cancer society, american society for colposcopy and cervical pathology, and american society for clinical pathology screening guidelines for the prevention and early detection of cervical cancer, CA: A Cancer Journal for Clinicians 62 (3) (2012) 147–172.
[4] E. Bengtsson, P. Malm, Screening for cervical cancer using automated analysis of PAP-Smears, Computational and Mathematical Methods in Medicine 2014 (2014) 1–12.
[5] L. Zhang, H. Kong, C. Ting Chin, S. Liu, X. Fan, T. Wang, S. Chen, Automation-assisted cervical cancer screening in manual liquid-based cytology with hematoxylin and eosin staining, Cytometry Part A 85 (3) (2014) 214–230.
[6] H.A. Phoulady, M. Zhou, D.B. Goldgof, L.O. Hall, P.R. Mouton, Automatic quantification and classification of cervical cancer via adaptive nucleus shape modeling, in: IEEE International Conference on Image Processing (ICIP), IEEE, 2016, pp. 2658–2662.
[7] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[8] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016, pp. 770–778..
[9] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: CVPR, 2015, pp. 3431–3440.
[10] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (6) (2016) 1137–1149.
[11] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017, pp. 936–944.
[12] G. Litjens, T. Kooi, B.E. Bejnordi, A.A.A. Setio, F. Ciompi, M. Ghafoorian, J.A. Van Der Laak, B. Van Ginneken, C.I. Sánchez, A survey on deep learning in medical image analysis, Medical Image Analysis 42 (2017) 60–88.

[13] A. Esteva, B. Kuprel, R.a. Novoa, J. Ko, S.M. Swetter, H.M. Blau, S. Thrun, Dermatologist-level classification of skin cancer with deep neural networks, Nature 542 (7639) (2017) 115..
[14] Y. Liang, R. Kang, C. Lian, Y. Mao, An end-to-end system for automatic urinary particle recognition with convolutional neural network, Journal of Medical Systems 42 (9) (2018) 165.
[15] Y. Liang, Z. Tang, M. Yan, J. Liu, Object detection based on deep learning for urine sediment examination, Biocybernetics and Biomedical Engineering 38 (4) (2018) 661–670.
[16] A. Tareef, Y. Song, H. Huang, Y. Wang, D. Feng, M. Chen, W. Cai, Optimizing the cervix cytological examination based on deep learning and dynamic shape modeling, Neurocomputing 248 (2017) 28–40.
[17] Z. Lu, G. Carneiro, A.P. Bradley, D. Ushizima, M.S. Nosrati, A.G. Bianchi, C.M. Carneiro, G. Hamarneh, Evaluation of three algorithms for the segmentation of overlapping cervical cells, IEEE Journal of Biomedical and Health Informatics 21 (2) (2017) 441–450.
[18] L. Zhang, L. Lu, I. Nogues, R.M. Summers, S. Liu, J. Yao, DeepPap: Deep convolutional networks for cervical cell classification, IEEE Journal of Biomedical and Health Informatics 21 (6) (2017) 1633–1643.
[19] O.N. Jith, K. Harinarayanan, S. Gautam, A. Bhavsar, A.K. Sao, DeepCerv: Deep neural network for segmentation free robust cervical cell classification, in: Computational Pathology and Ophthalmic Medical Image Analysis, Springer, 2018, pp. 86–94.
[20] Y. Xiang, W. Sun, C. Pan, M. Yan, Z. Yin, Y. Liang, A novel automation-assisted cervical cancer reading method based on convolutional neural network, Biocybernetics and Biomedical Engineering 40 (2) (2020) 611–623.
[21] C. Zhang, D. Liu, L. Wang, Y. Li, X. Chen, R. Luo, S. Che, H. Liang, Y. Li, S. Liu, et al., Dccl: A benchmark for cervical cytology analysis, in: International Workshop on Machine Learning in Medical Imaging, Springer, 2019, pp. 63–72.
[22] J. Redmon, A. Farhadi, Yolov3: An incremental improvement, arXiv preprint arXiv:1804.02767..
[23] M.H. Stoler, M. Schiffman, et al., Interobserver reproducibility of cervical cytologic and histologic interpretations: realistic estimates from the ASCUS-LSIL triage study, JAMA 285 (11) (2001) 1500–1505.
[24] G. Koch, R. Zemel, R. Salakhutdinov, Siamese neural networks for one-shot image recognition, in: International Conference on Machine Learning (ICML) Deep Learning Workshop, vol. 2, 2015..
[25] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al., Matching networks for one shot learning, in: Advances in Neural Information Processing Systems, 2016, pp. 3630–3638.
[26] J. Snell, K. Swersky, R. Zemel, Prototypical networks for few-shot learning, in: Advances in Neural Information Processing Systems, 2017, pp. 4077–4087.
[27] F.S.Y. Yang, L. Zhang, T. Xiang, P.H. Torr, T.M. Hospedales, Learning to compare: relation network for few-shot learning, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2018.
[28] L. Zhang, H. Kong, C.T. Chin, S. Liu, Z. Chen, T. Wang, S. Chen, Segmentation of cytoplasm and nuclei of abnormal cells in cervical cytology using global and local graph cuts, Computerized Medical Imaging and Graphics 38 (5) (2014) 369–380.
[29] L. Zhang, H. Kong, S. Liu, T. Wang, S. Chen, M. Sonka, Graph-based segmentation of abnormal nuclei in cervical cytology, Computerized Medical Imaging and Graphics 56 (2017) 38–48.
[30] H. Lee, J. Kim, Segmentation of overlapping cervical cells in microscopic images with superpixel partitioning and cell-wise contour refinement, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshop, 2016, pp. 63–69..
[31] Y. Marinakis, G. Dounias, J. Jantzen, Pap smear diagnosis using a hybrid intelligent scheme focusing on genetic algorithm based feature selection and nearest neighbor classification, Computers in Biology and Medicine 39 (1) (2009) 69–78.
[32] N. Song, Q. Du, Classification of cervical lesion images based on cnn and transfer learning, in: 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC), IEEE, 2019, pp. 316–319.
[33] O.E. Aina, S.A. Adeshina, A. Aibinu, Classification of cervix types using convolution neural network (cnn), in: 2019 15th International Conference on Electronics, Computer and Computation (ICECCO), IEEE, 2019, pp. 1–4.
[34] A.A. Abdullah, A.F.D. Giong, N.A.H. Zahri, Cervical cancer detection method using an improved cellular neural network (cnn) algorithm, Indonesian Journal of Electrical Engineering and Computer Science 14 (1) (2019) 210–218.
[35] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: International Conference on Learning Representations, 2015..
[36] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980–2988.
[37] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, Y. LeCun, Overfeat: Integrated recognition, localization and detection using convolutional networks, in: International Conference on Representation Learning (ICLR), 2014..
[38] W. Chu, D. Cai, Deep feature based contextual model for object detection, Neurocomputing 275 (2018) 1035–1042.
[39] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2014, pp. 580–587.
[40] R. Girshick, Fast R-CNN, in: International Conference on Computer Vision (ICCV), 2015, pp. 1440–1448..
[41] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in: International Conference on Computer Vision (ICCV), IEEE, 2017, pp. 2980–2988..
[42] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real-time object detection, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016, pp. 779–788.
[43] S. Zhang, L. Wen, X. Bian, Z. Lei, S.Z. Li, Single-shot refinement neural network for object detection, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2018.
[44] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, T. Lillicrap, Meta-learning with memory-augmented neural networks, in: International Conference on Machine Learning (ICML), 2016, pp. 1842–1850.
[45] S. Ravi, H. Larochelle, Optimization as a model for few-shot learning, in: International Conference for Learning Representations (ICLR), 2017.
[46] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in: International Conference on Machine Learning, 2017, pp. 1126–1135..
[47] V. Garcia, J. Bruna, Few-shot learning with graph neural networks, arXiv preprint arXiv:1711.04043..
[48] L. Liu, T. Zhou, G. Long, J. Jiang, C. Zhang, Learning to propagate for graph meta-learning, in: Advances in Neural Information Processing Systems, 2019, pp. 1037–1048.
[49] X. Dong, L. Zheng, F. Ma, Y. Yang, D. Meng, Few-example object detection with model communication, IEEE Transactions on Pattern Analysis and Machine Intelligence 41 (7) (2018) 1641–1654.
[50] E. Schwartz, L. Karlinsky, J. Shtok, S. Harary, M. Marder, S. Pankanti, R. Feris, A. Kumar, R. Giries, A.M. Bronstein, RepMet: Representative-based metric learning for classification and one-shot object detection, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2019, pp. 5197–5206.
[51] B. Kang, Z. Liu, X. Wang, F. Yu, J. Feng, T. Darrell, Few-shot object detection via feature reweighting, in: IEEE International Conference on Computer Vision (ICCV), IEEE, 2019, pp. 8420–8429.
[52] P.W. Battaglia, J.B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, et al., Relational inductive biases, deep learning, and graph networks, arXiv Preprint arXiv:1806.01261..
[53] L.V.D. Maaten, G. Hinton, Visualizing data using t-SNE, Journal of Machine Learning Research 9 (Nov) (2008) 2579–2605.

Yixiong Liang is currently an Associate Professor of Computer Science in Central South University. Between 2011 and 2012, he was a visitor at the Robotics Institute, Carnegie Mellon University. From 2005 to 2007, he was a Postdoctoral Fellow in Institute of Automation, Chinese Academy of Science. He received the Ph.D., M.S. and B.S. degrees from Chongqing University, China, in 2005, 2002 and 1999, respectively. His research interests include computer vision and machine learning.

Zhihong Tang is currently a master student in the School of Computer Science and Engineering in Central South University, China. His research interests include computer vision and medical image analysis.



Meng Yan is currently a master student in the School of Computer Science and Engineering in Central South University, China. His research interests include medical image processing, machine learning and computer vision.

Qing Liu received the bachelor degree and Ph.D. degree in computer science and technology from the Central South University, Changsha, China, in 2011 and 2017 respectively. She is currently a lecturer at Central South University. She was a postdoc researcher at Central South University from 2017–2019. Dr. Liu has authored or co-authored more than 10 papers in journals and conferences. Her research interests include salient object detection and medical image analysis.

Jialin Chen is currently a master student in the School of Computer Science and Engineering in Central South University, China. His research interests include image fusion, computer vision and machine learning.

Xiang Yao received the Ph.D. degree from Central South University, in 2011. She is now an instructor in the School of Computer Science and Engineering, Central South University. Her research interests include computer vision and image processing.