

RESEARCH ARTICLE

Gastric precancerous diseases classification using CNN with a concise model

Xu Zhang^{1,2}, Weiling Hu^{3,4}, Fei Chen^{4,5}, Jiquan Liu^{1,2*}, Yuanhang Yang^{1,2}, Liangjing Wang^{4,5}, Huilong Duan^{1,2}, Jianmin Si^{3,4}

1 College of Biomedical Engineering & Instrument Science, Zhejiang University, Hangzhou, China, **2** Key Laboratory of Biomedical Engineering, Ministry of Education, Zhejiang University, Hangzhou, China, **3** Department of Gastroenterology, Sir Run Run Shaw Hospital, Zhejiang University, Hangzhou, China, **4** Institute of Gastroenterology, Zhejiang University, Hangzhou, China, **5** Department of Gastroenterology, Second Affiliated Hospital of Zhejiang University School of Medicine, Hangzhou, China

* liujq@zju.edu.cn



OPEN ACCESS

Citation: Zhang X, Hu W, Chen F, Liu J, Yang Y, Wang L, et al. (2017) Gastric precancerous diseases classification using CNN with a concise model. PLoS ONE 12(9): e0185508. <https://doi.org/10.1371/journal.pone.0185508>

Editor: Bin Liu, Harbin Institute of Technology Shenzhen Graduate School, CHINA

Received: May 11, 2017

Accepted: September 14, 2017

Published: September 26, 2017

Copyright: © 2017 Zhang et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All of the data files are available from the github. And the URL of our dataset repository is <https://github.com/jiquan/Data-Open-Access4PLoS-One.git>.

Funding: This work was supported by the National Natural Science Foundation of China (No. 31470955 to Mr Jiquan Liu)(<http://www.nsf.gov.cn/>) and the National Natural Science Foundation of China (No. 81401733 to Weiling Hu)(<http://www.nsf.gov.cn/>). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Abstract

Gastric precancerous diseases (GPD) may deteriorate into early gastric cancer if misdiagnosed, so it is important to help doctors recognize GPD accurately and quickly. In this paper, we realize the classification of 3-class GPD, namely, polyp, erosion, and ulcer using convolutional neural networks (CNN) with a concise model called the Gastric Precancerous Disease Network (GPDNet). GPDNet introduces fire modules from SqueezeNet to reduce the model size and parameters about 10 times while improving speed for quick classification. To maintain classification accuracy with fewer parameters, we propose an innovative method called iterative reinforced learning (IRL). After training GPDNet from scratch, we apply IRL to fine-tune the parameters whose values are close to 0, and then we take the modified model as a pretrained model for the next training. The result shows that IRL can improve the accuracy about 9% after 6 iterations. The final classification accuracy of our GPDNet was 88.90%, which is promising for clinical GPD recognition.

Introduction

Because gastroscopy can observe the gastrointestinal (GI) tract directly, it has been widely applied for GI examinations, which makes it more convenient for doctors to find lesions in the gastric mucosal surface. According to the evolution of cancerization, gastric lesions can be divided into three categories: advanced gastric cancer (AGC), early gastric cancer (EGC), and gastric precancerous disease (GPD). Patients with AGC can rarely be cured, and the 5-year survival rate is no more than 30% [1]. Although the 5-year survival rate of EGC can achieve 90% [2], endoscopic diagnosis of EGC is indeed difficult in most countries [3], and largely depends on the doctors' experience and the device advancement. Statistics indicate that GPDs, like erosion, polyps, and ulcers, may transform into gastric cancer if they are misdiagnosed or are not treated in time [4]. Based on these facts, it becomes more curial to intervene during the GPD process and develop computer-aided GPD recognition systems to avoid misdiagnosis before GPD transforms into EGC or AGC.

During the past decades, many computational approaches have been proposed for dealing with biomedical data. Liu et al. [5] proposed a novel method that can convert the frequency

Competing interests: The authors have declared that no competing interests exist.

profile into a series of profile-based proteins and achieved a promising result using support vector machine (SVM). Zheng et al. [6] introduced sparseness with Nonnegative matrix factorization and achieved a better tumor clustering result. A. Tartar et al. [7] have proposed a more robust CAD system with random forest classifier by using morphological features and patient information properties.

Several attempts have also been made in the field of endoscopic image lesion detection using computer vision, especially machine learning. Researchers have realized colonic polyp detection using geometric features, shape features, and texture features [8–10]. Li et al. used a local binary pattern (LBP) and SVM to classify capsule endoscopy images based on wavelet transform [11, 12]. Many works like [13–15] prove the dual-tree complex wavelet transform (DT-CWT) features to be quite effective for distinguishing different types of polyps. X. Shen et al. combined multiscale texture features with color features, which were fed into an Ada-Boost classifier to complete gastroscopic image lesion detection [16]. However, traditional machine-learning methods required handcrafted features, which were quite time-consuming and lacked of robustness.

Recently, deep learning using convolutional neural networks (CNNs) has achieved great success in the area of image recognition, including medical image analysis [17–19]. However, models trained from CNNs contain millions of parameters. To avoid overfitting and achieve quick convergence while training, researchers have adopted many effective tricks, including data augmentation [20, 21], ReLU for activation [22], batch normalization, Dropout [23], pretrained model, or transfer learning [24]. Some researchers, who went in a different direction, tried to modify networks for fewer parameters with state-of-the-art performance. SqueezeNet [25] was an outstanding example. SqueezeNet introduced fire modules to take the place of traditional convolutional layers, and it achieved AlexNet-level accuracy on ImageNet with 50 times fewer parameters. Thanks to these contributions, deep learning has also solved some popular issues in medical image analysis [21, 26–28]. However, deep learning shows slow development in endoscopic images. Tajbakhsh, Nima et al. [29, 30] integrate a variety of polyp features into one polyp detection system using an ensemble of CNNs. Ribeiro, Eduardo et al. [31] also realized colonic polyp classification using off-the-shelf CNN architectures. Zhu, Rongsheng et al. [32] combined a CNN for a trainable feature extractor with SVM for a classifier to realize endoscopic image lesion detection. However, these works either focused on colonic images or the networks were off-the-shelf with too many parameters that did not match well with small gastric image datasets.

To overcome these obstacles, we have proposed a new integrated CNN called the Gastric Precancerous Disease Network (GPDNet); this is a concise network and can recognize GPD precisely. Our GPDNet can help endoscopic doctors decrease misdiagnosis and screen out images containing lesions to alleviate doctors' workloads. GPDNet introduces fire modules from SqueezeNet to replace traditional convolutional layers and removes the fully connection layers to achieve fully convolutional network (FCN). Based on these efforts, our concise model is smaller and has fewer parameters. To maintain classification accuracy with fewer parameters, we have also proposed a novel fine-tuning algorithm called iterative reinforced learning (IRL). IRL is inspired by the manner in which humans learn and review. The idea of IRL is similar to dense-sparse-dense (DSD) [33]. However, DSD excessively focuses on S (sparse) stage training, which makes the training stage more complex, whereas IRL abandons the S stage, which DSD believes is the most important link. We just reset particular small weights to zero, then we retrain the networks directly. Furthermore, IRL is different from Dropout and DropConnect [34], which randomly drop units (along with their connections) from the neural networks during training. In detail, when we first train the model from scratch, we reset to zero, the parameters in the model that are close to zero. After the modification, we conduct another training. The output is a better model. In the authors' viewpoint, parameters close to

zero are not useful, and need to be reset to zero and relearned. Parameters that are far from zero contribute more to the accuracy of the model. Our research flowchart is shown in Fig 1.

Our contributions to this paper can be summarized as follows: (i) We have pioneered research on the recognition of GPD using convolutional neural networks, proving that deep learning can be applied in gastric images with state-of-the-art results. (ii) We have drawn on the experience of SqueezeNet to reduce the model size and parameters for a concise model, which could save time in gastric image classification and was promising for real-time recognition in gastric videos. (iii) We have proposed IRL to fine-tune the model parameters, which can keep the model's accuracy with a concise model.

Materials and methods

In this section, we will first concisely introduce the GPD dataset for training. Then we will establish our GPDNet and present its core components. Finally, we will describe the training stage in detail, including the fine-tuning stage for a more accurate model.

2.1. Data acquisition

To evaluate the performance of GPDNet, we collected de-identified gastroscopy images with gastric precancerous diseases from Sir Run Run Shaw Hospital, which all the patients provided written informed consent for their medical images to be published and used in this research. The number of 3-class images was 1331, including 388 images of erosion, 467 images of polyps, and 476 images of ulcer, which were labeled by two professional clinicians with all the gastroscopy images accessed anonymously. The size of the processed images we acquired was 560×475 . To expand the image set and reduce the redundant information, we extracted different sized regions of interest (ROI); then we used crop, translation, and other methods to achieve more images. After effective image augmentation, there were 3673 images in total, including 1211 images of erosion, 1218 images of polyps, and 1244 images of ulcer. To simplify our GPDNet for higher speed, we resized all the ROI images to 32×32 (see Fig 1 and Fig 2), so the image size of our GPDNet inputs is 32×32 . To avoid overfitting and make the dataset more general, for each class of images, we shuffled them at first, then we selected 300 images randomly as the testing dataset, and the rest images were regarded as the training dataset.

2.2. Architecture of GPDNet

Fig 3 shows the architecture of GPDNet. Because the input image size of our GPDNet is 32×32 , we have applied Alex's CIFAR-10 network with Caffe style as the basic net. For concise representation, we named this network "cifar-10". However, cifar-10 is an old-fashioned version. It requires more parameters due to the preserved fully connected layers, which will increase the model size. To decrease the number of parameters and maintain accuracy, we introduced the fire module from SqueezeNet, which is explained clearly in [25], to replace the convolutional layers in cifar-10. A fire module is comprised of a squeezed convolution layer (which has only 1×1 filters), feeding into an expand layer that has a mix of 1×1 and 3×3 convolution filters. Fire modules contribute to the network's concise models with fewer parameters but higher training speed. In addition, we replaced the fully connected layer in cifar-10 with a convolutional layer for GPDNet, which makes GPDNet a fully convolutional neural network. In Fig 3, the architecture of GPDNet is composed of two traditional convolution layers and two fire modules. In addition, pooling layers are adopted to reduce computation. Because we train GPDNet from scratch, we utilize Xavier to initialize convolutional kernel parameters. To avert the gradient vanishing problem, we chose ReLU as the activate function.

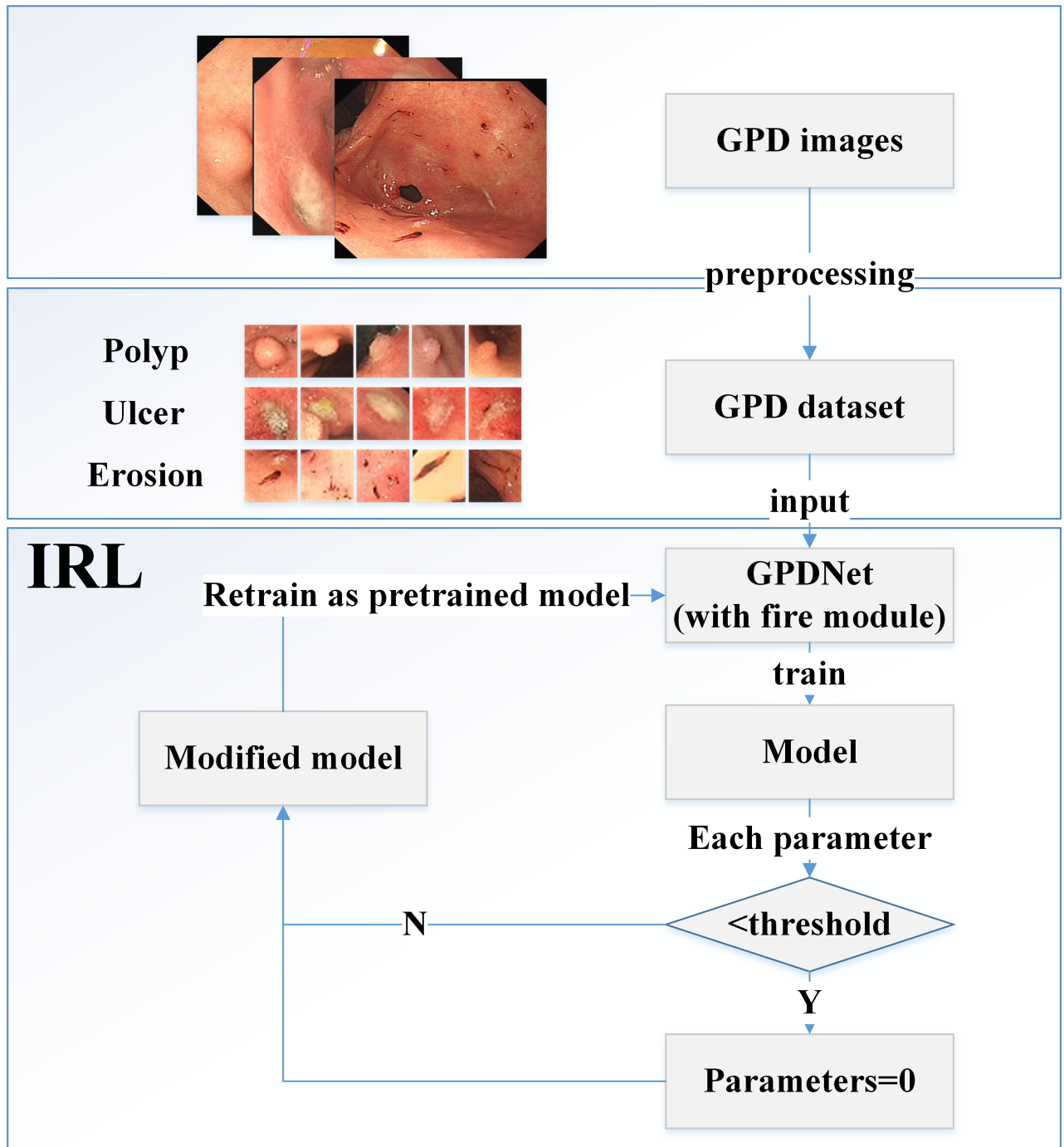


Fig 1. Research flowchart, including IRL algorithm.

<https://doi.org/10.1371/journal.pone.0185508.g001>

2.3. Training and a novel fine-tuning method for GPDNet

As described earlier, we split our dataset $D\{(X_i, l_i) | X_i \in images, l_i \in labels\}$ into a test set containing about 25% of the subjects and the training set containing about 75%. In total, the number of categories $L = 3$ for classification.

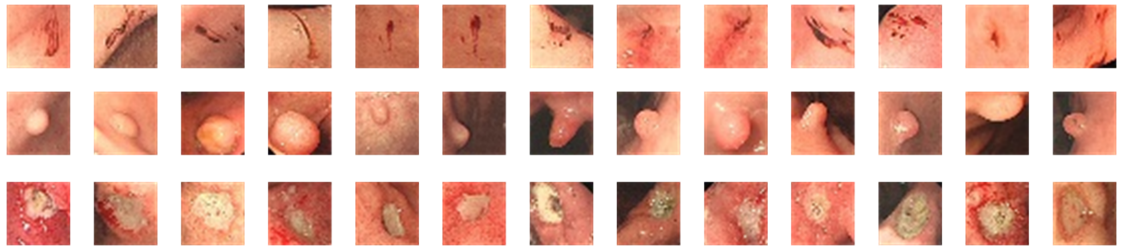


Fig 2. Some samples of GPD training images. The top row denotes erosion lesions. The middle row denotes polyps. The bottom row denotes ulcers. All of them may develop into EGC if they are misdiagnosed during screening.

<https://doi.org/10.1371/journal.pone.0185508.g002>

We train the GPDNet using mini-batch stochastic gradient descent. Our loss function can be described as formula (1), where $P(l_i | X_i; (W, b))$ indicates the probability of X_i being correctly classified as l_i . We aim to solve optimal parameters (W, b) to minimize loss.

$$Loss = \sum_{X_i, l_i \in D} -\log(P(l_i | X_i; (W, b))) \tag{1}$$

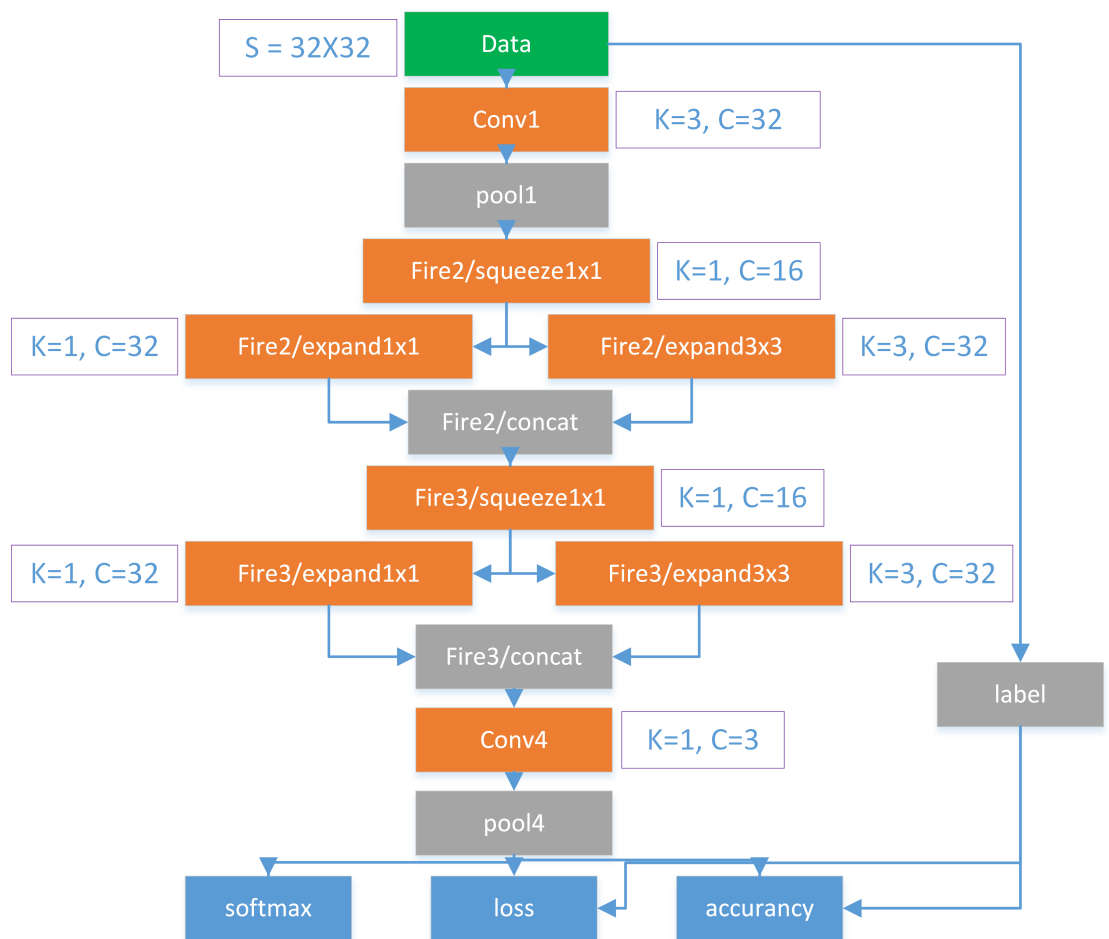


Fig 3. Architecture of GPDNet. K denotes kernel size; C denotes channel or number of feature maps; S denotes input image size.

<https://doi.org/10.1371/journal.pone.0185508.g003>

The mini-batch size is $N = 32$. The basic learning rate is 0.001, and we have adopted “step” as our learning rate decay policy. The weight decay is set as 0.9. The training undergoes 50 epochs. After the training is done, the performance of the model acquired is not satisfactory due to using fewer parameters to represent so many images. However, we introduced a novel fine-tuning method IRL to improve the model, which shows that models with fewer parameters can also achieve an outstanding result.

IRL is inspired by the manner in which humans iteratively learn things. In the authors’ opinion, the value of a parameter can reflect its contribution to the model’s accuracy. The larger the value (absolute value) is, the more impact the parameter will have on the model’s accuracy. In other words, the smaller the value is, the less impact the parameter will have. Consider an extreme condition: If the value equals zero, then the corresponding parameter will be regarded as insignificant, and should be retrained or discarded. To conclude, for an arbitrary parameter α in the model, if α is less than the given threshold, it will be set as zero.

$$\forall \alpha \in Model \begin{cases} \alpha = 0, \text{ if } (|\alpha| \leq \text{threshold}). \\ \alpha = \alpha, \text{ otherwise.} \end{cases} \quad (2)$$

In this paper, we set the threshold = 0.001. If the model is modified, we will treat this model as the pretrained model for the next training. After several training iterations, the output model should be better.

Results and discussions

In this section, we will discuss our results from three perspectives. First, we will analyze the correspondence between the model’s accuracy and the value of the parameters. Second, we will discuss whether the threshold value and the number of iterations for the IRL have an impact on the model. Finally, we will compare our CNN architecture with fire modules and without fire modules and analyze GPDNet’s testing results. All the experiments and training tasks were conducted on a Nvidia Tesla K20m GPU (5GB).

3.1 Model accuracy versus the value of parameters

Because the value of the model parameters is between -1 and 1, we divided the range [0,1] into ten subranges, [0,0.001), [0.001,0.005), [0.005,0.01), [0.01,0.05), [0.05,0.1), [0.1,0.15), [0.15,0.2), [0.2,0.3), [0.3,0.4), and [0.4,1]. For each subrange, we set all the model parameters of that range as zero, then we tested the modified model’s accuracy with the same test dataset. It’s worth noting that these ten subranges were not divided equally due to the number of parameters distributed unevenly. The result is shown in Fig 4.

From Fig 4, when we have completed IRL, we can find that in [0.01,0.05), there are 3335 parameters. The accuracy drops very little if we set them to zero. However, if we set [0.3,0.4) with 392 parameters or [0.4,1] with 155 parameters to zero, the accuracy drops rapidly. This shows that the value of a parameter has an incredible impact on the model’s accuracy. Specifically, the larger the parameter is, the more impact it has on the model’s accuracy. We can also see an interesting change in the distribution of parameters after using IRL. Even though the number of parameters whose value is less than 0.05 changes slightly, the number of parameters between 0.05 and 0.15 have decreased, which leads to a rapid increase in parameters greater than 0.15. This also shows that the increasing number of large parameters contributes to the improvement of model accuracy.

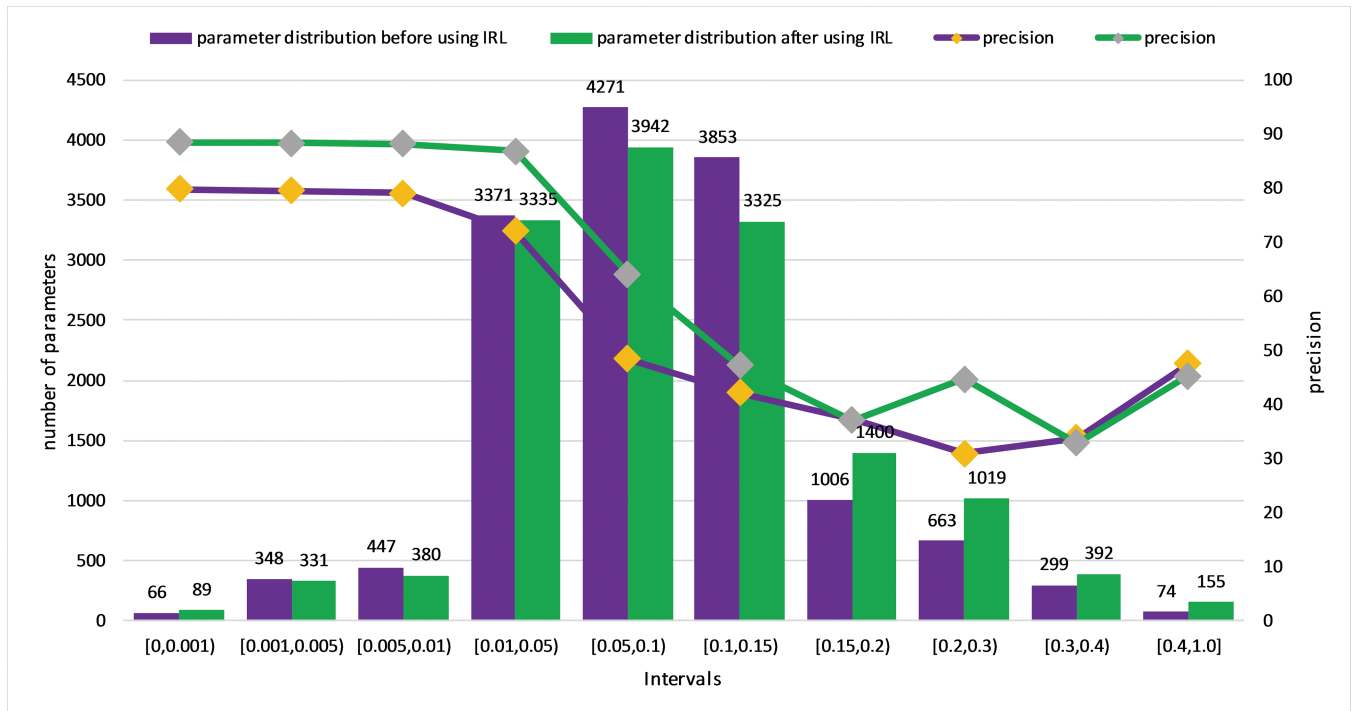


Fig 4. The number of parameters of each range, corresponding to the modified model’s accuracy if we set all the parameters in that range as zero. The purple bar denotes the parameter distribution before using IRL, which corresponds to the purple line. The green bar denotes the parameter distribution after using IRL, which corresponds to the green line. The super-parameters of IRL are threshold = 0.001 and iterations = 5.

<https://doi.org/10.1371/journal.pone.0185508.g004>

3.2 The choice of iterations and threshold of IRL

To find the optimal threshold and iterations, we set the thresholds as 0.001, 0.005, 0.01, 0.015, 0.05, 0.1, 0.15, 0.2, and 0.3, respectively. For each threshold, we iterated 7 times. Accuracy figures are listed in Table 1. From Table 1, we can determine the following: (1) If we set the threshold too large, the final model accuracy will decrease quickly (even not converge) as expected. The reason is that too large a threshold will break the model’s basic structure. IRL will not be effective anymore in this situation. We should choose a suitable threshold, which can avoid impacting the model’s basic structure. (2) When we have chosen a proper threshold, the model accuracy will tend to relative stability after several iterations, and this “proper threshold” will have little influence on the model accuracy. (3) For GPDNet, when the threshold < 0.05, the model’s basic structure can be preserved, so we choose the IRL threshold = 0.001 in Fig 1 and Table 2.

Table 1. Influence on model accuracy of the IRL threshold and iteration times.

Threshold \ Iteration	0.001	0.005	0.01	0.015	0.05	0.1	0.15	0.2	0.3
0	79.74	79.74	79.74	79.74	79.74	79.74	79.74	79.74	79.74
1	86.1	86.31	86.21	85.88	86.42	86.75	83.73	78.77	35.67
2	86.85	87.07	88.64	86.85	87.39	87.50	84.38	48.06	32.97
3	88.36	87.61	87.71	87.61	88.04	87.82	84.81	80.06	32.97
4	88.04	87.61	88.15	88.14	87.93	88.04	84.81	82.11	32.97
5	88.15	87.93	88.58	88.04	88.36	87.61	84.91	46.77	32.97
6	88.47	87.93	88.36	88.90	88.04	87.93	85.56	82.22	32.97
7	88.47	88.58	88.36	88.58	88.25	87.82	86.42	83.19	32.97

<https://doi.org/10.1371/journal.pone.0185508.t001>

Table 2. Comparison between two architectures.

	GPDNet without fire modules	GPDNet with fire modules
Number of fire modules	0	2
Have fully connected layers	Y	N
Number of parameters	144928	14368
Model size (caffe model)	568KB	60KB
Model's accuracy without IRL	87.39%	79.74%
Model's accuracy with IRL (iteration = 7, threshold = 0.001)	88.47%	88.47%
Model's best accuracy (%)	89.22%	88.90%
Time consuming (classify 900 images)	3.15s	2.65s

<https://doi.org/10.1371/journal.pone.0185508.t002>

3.3 Comparison between GPDNet with fire modules and without fire modules

Fig 2 displays some samples of GPD training images. To further verify the model trained from GPDNet, we have determined more indicators on the training models between GPDNet with fire modules and without fire modules. The result is shown in Table 2.

Table 2 depicts two GPDNets. The first column denotes GPDNet without fire modules. The second column denotes GPDNet with fire modules. The architecture of GPDNet without fire modules is based on cifar-10. We only change the output number from 10 to 3.

Compared with GPDNet without fire modules, the parameters of GPDNet with fire modules are reduced 10.0 times and the model size is reduced 9.5 times. If we do not use the IRL method, the accuracy is only 79.74%. However, if we apply IRL, the best accuracy can reach 88.9%, which improves about 9%, and the result is comparable to GPDNet without fire modules. Small networks with few parameters will improve computational efficiency, which in turn improves the speed of GPD classification. To confirm this, we classified 900 images, which comprised the test dataset, using the GPDNet models with and without fire modules. The result shows that GPDNet without fire modules costs 3.15 s, while GPDNet with fire modules only requires 2.65 s.

Conclusion

In this paper, we established a CNN called GPDNet to classify three categories of gastric precancerous diseases. We utilize the fire module as the basic component of the architecture and the size of the concise model is reduced 9.5 times. The number of parameters is reduced 10.0 times. Time consumed in recognition has also decreased due to fewer parameters and smaller model size. However, if we train the model from scratch, the accuracy can only reach 80%, which is not enough for clinical applications. In this paper, we propose IRL to reinforce learning the parameters iteratively. IRL's threshold should be less than 0.05 and there should be more than four iterations. The result shows that IRL can improve the accuracy to 88.9%, which is comparable with GPDNet without fire modules. Our work is promising for recognition of gastric precancerous disease, which will help doctors avoid misdiagnosis.

Author Contributions

Conceptualization: Jiquan Liu.

Data curation: Weiling Hu, Fei Chen, Liangjing Wang.

Funding acquisition: Jiquan Liu.

Investigation: Xu Zhang, Weiling Hu, Fei Chen.

Methodology: Xu Zhang, Jiquan Liu, Yuanhang Yang.

Project administration: Jiquan Liu.

Resources: Weiling Hu, Fei Chen.

Software: Xu Zhang, Yuanhang Yang.

Supervision: Weiling Hu, Jiquan Liu, Liangjing Wang, Huilong Duan, Jianmin Si.

Validation: Xu Zhang, Weiling Hu, Fei Chen, Liangjing Wang.

Visualization: Xu Zhang.

Writing – original draft: Xu Zhang.

Writing – review & editing: Weiling Hu, Jiquan Liu.

References

1. Ajani JA, Bentrem DJ, Besh S, D'Amico TA, Das P, Denlinger C, et al. Gastric cancer, version 2.2013. *Journal of the National Comprehensive Cancer Network*. 2013; 11(5):531–46. PMID: [23667204](https://pubmed.ncbi.nlm.nih.gov/23667204/)
2. Isobe Y, Nashimoto A, Akazawa K, Oda I, Hayashi K, Miyashiro I, et al. Gastric cancer treatment in Japan: 2008 annual report of the JGCA nationwide registry. *Gastric Cancer*. 2011; 14(4):301–16. <https://doi.org/10.1007/s10120-011-0085-6> PMID: [21894577](https://pubmed.ncbi.nlm.nih.gov/21894577/)
3. Yao K. The endoscopic diagnosis of early gastric cancer. *Annals of Gastroenterology*. 2012; 26(1):11.
4. Weck MN, Brenner H. Prevalence of chronic atrophic gastritis in different parts of the world. *Cancer Epidemiology and Prevention Biomarkers*. 2006; 15(6):1083–94.
5. Liu B, Zhang D, Xu R, Xu J, Wang X, Chen Q, et al. Combining evolutionary information extracted from frequency profiles with sequence-based kernels for protein remote homology detection. *Bioinformatics*. 2014; 30(4):472–9. <https://doi.org/10.1093/bioinformatics/btt709> PMID: [24318998](https://pubmed.ncbi.nlm.nih.gov/24318998/)
6. Zheng CH, Huang DS, Zhang L, Kong XZ. Tumor Clustering Using Nonnegative Matrix Factorization With Gene Selection. *IEEE Transactions on Information Technology in Biomedicine*. 2009; 13(4):599–607. <https://doi.org/10.1109/TITB.2009.2018115> PMID: [19369170](https://pubmed.ncbi.nlm.nih.gov/19369170/)
7. Tartar A, Kılıç N, Akan A, editors. A new method for pulmonary nodule detection using decision trees. 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC); 2013 3–7 July 2013.
8. Summers RM, Yao J, Pickhardt PJ, Franaszek M, Bitter I, Brickman D, et al. Computed tomographic virtual colonoscopy computer-aided polyp detection in a screening population. *Gastroenterology*. 2005; 129(6):1832–44. <https://doi.org/10.1053/j.gastro.2005.08.054> PMID: [16344052](https://pubmed.ncbi.nlm.nih.gov/16344052/)
9. Paik DS, Beaulieu CF, Rubin GD, Acar B, Jeffrey RB Jr, Yee J, et al. Surface normal overlap: a computer-aided detection algorithm with application to colonic polyps and lung nodules in helical CT. *Medical Imaging, IEEE Transactions on*. 2004; 23(6):661–75.
10. Ameling S, Wirth S, Paulus D, Lacey G, Vilarino F. Texture-based polyp detection in colonoscopy. *Bildverarbeitung für die Medizin 2009*: Springer; 2009. p. 346–50.
11. Li B, Meng MQ-H. Texture analysis for ulcer detection in capsule endoscopy images. *Image and Vision computing*. 2009; 27(9):1336–42.
12. Li B, Meng MQ-H, editors. Capsule endoscopy images classification by color texture and support vector machine. *Automation and Logistics (ICAL), 2010 IEEE International Conference on*; 2010: IEEE.
13. Häfner M, Liedlgruber M, Uhl A. Colonic polyp classification in high-definition video using complex wavelet-packets. *Bildverarbeitung für die Medizin 2015*: Springer; 2015. p. 365–70.
14. Häfner M, Gangl A, Liedlgruber M, Uhl A, Vécsei A, Wrba F, editors. Pit pattern classification using extended local binary patterns. *Information Technology and Applications in Biomedicine, 2009 ITAB 2009 9th International Conference on*; 2009: IEEE.
15. Häfner M, Uhl A, Vecsei A, Wimmer G, Wrba F, editors. Complex wavelet transform variants and discrete cosine transform for scale invariance in magnification-endoscopy image classification. *Information*

- Technology and Applications in Biomedicine (ITAB), 2010 10th IEEE International Conference on; 2010: IEEE.
16. Shen X, Sun K, Zhang S, Cheng S, editors. Lesion detection of electronic gastroscopy images based on multiscale texture feature. *Signal Processing, Communication and Computing (ICSPCC)*, 2012 IEEE International Conference on; 2012: IEEE.
 17. Shen D, Wu G, Suk H-I. *Deep Learning in Medical Image Analysis. Annual Review of Biomedical Engineering*. 2017(0).
 18. Litjens G, Kooi T, Bejnordi BE, Setio AAA, Ciompi F, Ghafoorian M, et al. A survey on deep learning in medical image analysis. *arXiv preprint arXiv:170205747*. 2017.
 19. Seebock P. *Deep learning in medical image analysis. Master's thesis, Vienna University of Technology, Faculty of Informatics*. 2015.
 20. Miao S, Wang ZJ, Liao R. A CNN Regression Approach for Real-Time 2D/3D Registration. *IEEE transactions on medical imaging*. 2016; 35(5):1352–63. <https://doi.org/10.1109/TMI.2016.2521800> PMID: [26829785](https://pubmed.ncbi.nlm.nih.gov/26829785/)
 21. Setio AAA, Ciompi F, Litjens G, Gerke P, Jacobs C, Riel SJv, et al. Pulmonary Nodule Detection in CT Images: False Positive Reduction Using Multi-View Convolutional Networks. *IEEE Transactions on Medical Imaging*. 2016; 35(5):1160–9. <https://doi.org/10.1109/TMI.2016.2536809> PMID: [26955024](https://pubmed.ncbi.nlm.nih.gov/26955024/)
 22. Pereira S, Pinto A, Alves V, Silva CA. Brain Tumor Segmentation Using Convolutional Neural Networks in MRI Images. *IEEE transactions on medical imaging*. 2016; 35(5):1240–51. <https://doi.org/10.1109/TMI.2016.2538465> PMID: [26960222](https://pubmed.ncbi.nlm.nih.gov/26960222/)
 23. Srivastava N, Hinton GE, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*. 2014; 15(1):1929–58.
 24. Tajbakhsh N, Shin JY, Gurudu SR, Hurst RT, Kendall CB, Gotway MB, et al. Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning? *IEEE Transactions on Medical Imaging*. 2016; 35(5):1299–312. <https://doi.org/10.1109/TMI.2016.2535302> PMID: [26978662](https://pubmed.ncbi.nlm.nih.gov/26978662/)
 25. Iandola FN, Moskewicz MW, Ashraf K, Han S, Dally WJ, Keutzer K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 1MB model size. *arXiv preprint arXiv:160207360*. 2016.
 26. Anthimopoulos M, Christodoulidis S, Ebner L, Christe A, Mougiakakou S. Lung Pattern Classification for Interstitial Lung Diseases Using a Deep Convolutional Neural Network. *IEEE Transactions on Medical Imaging*. 2016; 35(5):1207–16. <https://doi.org/10.1109/TMI.2016.2535865> PMID: [26955021](https://pubmed.ncbi.nlm.nih.gov/26955021/)
 27. Kamnitsas K, Ledig C, Newcombe VF, Simpson JP, Kane AD, Menon DK, et al. Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. *Medical Image Analysis*. 2017; 36:61–78. <https://doi.org/10.1016/j.media.2016.10.004> PMID: [27865153](https://pubmed.ncbi.nlm.nih.gov/27865153/)
 28. Albarqouni S, Baur C, Achilles F, Belagiannis V, Demirci S, Navab N. AggNet: Deep Learning From Crowds for Mitosis Detection in Breast Cancer Histology Images. *IEEE Transactions on Medical Imaging*. 2016; 35(5):1313–21. <https://doi.org/10.1109/TMI.2016.2528120> PMID: [26891484](https://pubmed.ncbi.nlm.nih.gov/26891484/)
 29. Tajbakhsh N, Gurudu SR, Liang J, editors. Automatic polyp detection in colonoscopy videos using an ensemble of convolutional neural networks. 2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI); 2015: IEEE.
 30. Zhang R, Zheng Y, Mak WCT, Yu R, Wong SH, Lau J, et al. Automatic Detection and Classification of Colorectal Polyps by Transferring Low-level CNN Features from Non-Medical Domain. *IEEE Journal of Biomedical and Health Informatics*. 2016.
 31. Ribeiro E, Uhl A, Wimmer G, Häfner M. Exploring Deep Learning and Transfer Learning for Colonic Polyp Classification. *Computational and Mathematical Methods in Medicine*. 2016; 2016.
 32. Zhu R, Zhang R, Xue D, editors. Lesion detection of endoscopy images based on convolutional neural network features. 2015 8th International Congress on Image and Signal Processing (CISP); 2015: IEEE.
 33. Han S, Pool J, Narang S, Mao H, Tang S, Elsen E, et al. Dsd: Regularizing deep neural networks with dense-sparse-dense training flow. *arXiv preprint arXiv:160704381*. 2016.
 34. Wan L, Zeiler M, Zhang S, Cun YL, Fergus R, editors. Regularization of neural networks using dropconnect. *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*; 2013.