Data Augmentation in Logit Space for Medical Image Classification with Limited Training Data

Yangwen Hu[1], Zhehao Zhong[1], Ruixuan Wang[1,2(✉)], Hongmei Liu[1], Zhijun Tan[1], and Wei-Shi Zheng[1,2,3]

[1] School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China
wangruix5@mail.sysu.edu.cn
[2] Key Laboratory of Machine Intelligence and Advanced Computing, MOE, Guangzhou, China
[3] Pazhou Lab, Guangzhou, China

Abstract. Successful application of deep learning often depends on large amount of training data. However in practical medical image analysis, available training data are often limited, often causing over-fitting during model training. In this paper, a novel data augmentation method is proposed to effectively alleviate the over-fitting issue, not in the input space but in the logit space. This is achieved by perturbing the logit vector of each training data within the neighborhood of the logit vector in the logit space, where the size of neighborhood can be automatically and adaptively estimated for each training data over training stages. The augmentations in the logit space may implicitly represent various transformations or augmentations in the input space, and therefore can help train a more generalizable classifier. Extensive evaluations on three small medical image datasets and multiple classifier backbones consistently support the effectiveness of the proposed method.

Keywords: Data augmentation · Logit space · Limited data

1 Introduction

Deep learning techniques have been successfully applied to intelligent diagnosis of various diseases [6,7,13]. In general, expert-level diagnosis from the intelligent systems are often based on large set of annotated training data for each disease. However, due to the existence of lots of rare diseases, costly and little time resource from clinicians, privacy concerns, and difficulty in data sharing across medical centres etc., annotated and publicly available large datasets for disease diagnosis are very limited. As a result, practical investigations of intelligent diagnosis often face the challenge of limited training data for model training.

Y. Hu and Z. Zhong—The authors contribute equally to this work.

M. de Bruijne et al. (Eds.): MICCAI 2021, LNCS 12905, pp. 469–479, 2021.
https://doi.org/10.1007/978-3-030-87240-3_45

To alleviate the over-fitting issue due to limited training data for the current task of interest, various approaches have been developed particularly for deep learning models. One group of approaches are based on transferring knowledge from a relatively large auxiliary dataset which contains different classes but in content is often similar to the dataset of the current task. The auxiliary dataset can be used to train and then fix a feature extractor for the current task, as in the matching network [22], prototypical network [16], and relation network [18], or to train a feature extractor which is then fine-tuned by the dataset of the current task, as in the meta-learning methods MAML [8] and LEO [14], or to jointly train a feature extractor with the dataset of the current task [24]. Such transfer learning techniques assume that the auxiliary dataset is annotated and similar to the dataset of the current task. However, large annotated auxiliary medical image dataset is generally not available in the scenario of intelligent diagnosis. Another group of approaches are based on various data augmentation techniques to increase the amount of the original training data. Besides the conventional data augmentations like random cropping, scaling, rotating, flipping, and color jittering of each training image, more advanced augmentation techniques have been recently developed, including Mixup [27], Cutout [4], Cutmix [26], AutoAugment [2], and RandAugment [3]. All these augmentations are performed directly on images and the types of basic augmentations (transformations) on images need to be manually designed. Besides data augmentation in the input space, augmentation in the semantic feature space has also been proposed [25], where various semantic transformations on images may be implicitly realized by perturbing each feature vector along certain feature dimensions.

This study follows the direction of data augmentation for over-fitting alleviation. Different from all the existing augmentations either in the input space or in the feature space, the proposed novel augmentation is in the (classifier's pre-softmax) logit space. Perturbations of each data in the logit space can implicitly represent various transformations in the input or feature space, and the augmented data in the logit space can help train the classifier to directly satisfy the desired property of generalizability, i.e., similar data in the logit space should come from the same class. Innovatively, the magnitude of perturbation can be adaptively estimated over the training process based on uncertainty for each logit element, where the logit uncertainty is part of the classifier model output. Experimental evaluations on multiple datasets with various classifier backbones prove the effectiveness and generalizability of the proposed method.

2 Methodology

The objective is to train a generalizable classifier with limited available training data. Assume a classifier is represented by a convolutional neural network (CNN), consisting of multiple convolutional layers (i.e., feature extractor) and one last fully connected layer (i.e., classifier head). One desired property of any generalizable classifier is that, two images should come from the same class if their feature representations (i.e., feature vectors in the feature space) from the

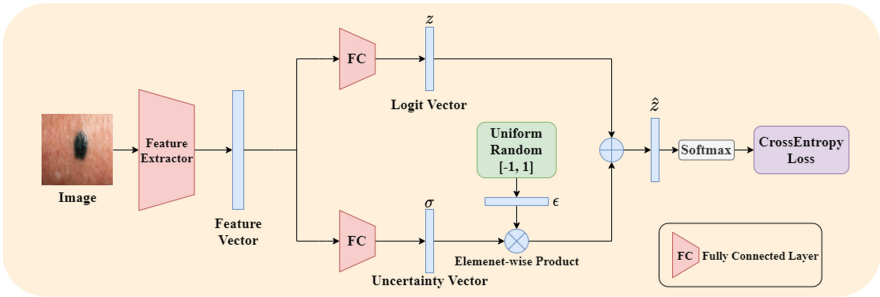

Fig. 1. Classifier training with augmentation in the logit space. Feature vector is extracted from the input image and then forwarded to two parallel fully connected layers to obtain the original logit vector and the uncertainty vector. Multiples samples of logit vectors are then generated based on the combination of the original logit vector and the uncertainty vector, with each sample fed to the softmax operator finally.

feature extractor output are similar enough. This property can be extended to the output space (i.e., from the linear transformation of feature vector, also called logit space) of the pre-softmax in the last fully connected layer. Most existing data augmentation methods try to help classifiers satisfy this property during classifier training indirectly by generating multiple transformed versions of the same image and expecting the classifier to generate correspondingly similar feature vectors (and logit vectors). However, such data augmentations cannot assure that similar feature or logit vectors would always come from the same class. With this observation, we propose a simple yet effective augmentation method in the logit space, directly helping classifier satisfy the property that similar logit feature vectors should come from the same class. Particularly, we propose training a CNN classifier which can generate not only the conventional logit vector but also the uncertainty for each logit element. Based on the logit and its uncertainty, multiple logit vector samples for each single input image can be obtained.

2.1 Classifier with Logit Uncertainty for Data Augmentation

For the logit vector $\mathbf{z}_i$ of any input image $\mathbf{x}_i$, suppose those logit vectors within its neighborhood in the logit space correspond to similar input images in content. Then sampling from the neighborhood would naturally generate multiple logit vectors associating with various input data instances from the same class. By enforcing the classifier to have same prediction for these sampled logit vectors during training, the classifier would satisfy the desire property of generalizability.

The challenge for data augmentation in the logit space is to determine the size of the neighborhood in the logit space for each input image. Manually setting a fixed neighborhood size often results in undesirable augmentation effect without considering the particular training data, training stage, and the employed CNN classifier architecture. For example, the distribution of logit vectors for each class of data may be spread in a much larger region in the logit space at the

early training stage (when the classifier has not been well trained) than that at the later training stage, or some training images of one class may be similar to images of certain other class(es) while the other images in the same class not. In these cases, it would be desirable if an adaptive neighborhood size can be automatically determined for each training data at each training stage.

In this study, we propose applying the classifier with uncertainty estimate for determination of neighborhood size for each training data. As demonstrated in Fig. 1, the classifier head consists of two parallel fully connected layers, one layer output representing the logit vector $\mathbf{z}_i$, and the other representing the uncertainty $\boldsymbol{\sigma}_i$ of corresponding logit elements. During training, instead of feeding the logit vector $\mathbf{z}_i$ to the softmax operator as in conventional classifier, here the classifier feeds multiple samples of logit vectors $\{\hat{\mathbf{z}}_{i,k}, k = 1, \ldots, K\}$ around the original $\mathbf{z}_i$, each generated by

$$\hat{\mathbf{z}}_{i,k} = \mathbf{z}_i + \boldsymbol{\epsilon}_k \odot \boldsymbol{\sigma}_i. \tag{1}$$

$\boldsymbol{\epsilon}_k$ is a vector whose element is randomly sampled from the uniform distribution within the range $[-1, 1]$, and $\odot$ is the Hadamard (i.e., element-wise) product. Note that the element in the uncertainty vector $\boldsymbol{\sigma}_i$ is not constrained to be non-negative (although it should be in principle) considering that the negative sign of any element in $\boldsymbol{\sigma}_i$ can be absorbed into the corresponding element in the random vector $\boldsymbol{\epsilon}_k$. For each sampled logit vector $\hat{\mathbf{z}}_{i,k}$, denote the corresponding classifier output by $\hat{\mathbf{y}}_{i,k}$ which is the softmax output with $\hat{\mathbf{z}}_{i,k}$ as input, and the ground-truth one-hot vector by $\mathbf{y}_i$ for the original input image $\mathbf{x}_i$. Then, the classifier can be trained with the training set $\{(\mathbf{x}_i, \mathbf{y}_i), i = 1, \ldots, N\}$ by minimizing the cross-entropy loss L,

$$L = \frac{1}{NK} \sum_{i=1}^{N} \sum_{k=1}^{K} l(\mathbf{y}_i, \hat{\mathbf{y}}_{i,k}) = -\frac{1}{NK} \sum_{i=1}^{N} \sum_{k=1}^{K} \left(\hat{z}_{i,k,c} - \log \sum_{c'=1}^{C} \exp(\hat{z}_{i,k,c'}) \right), \tag{2}$$

where $l(\mathbf{y}_i, \hat{\mathbf{y}}_{i,k})$ is the well-known cross-entropy function to measure the difference between $\mathbf{y}_i$ and $\hat{\mathbf{y}}_{i,k}$, which can be transformed to the format on the right side of Eq. (2). $\hat{z}_{i,k,c}$ is the c-th element in $\hat{\mathbf{z}}_{i,k}$ and c is the index associated with the ground-truth class for the input $\mathbf{x}_i$, and C is the number of logit or classifier output elements. At the early stage of classifier training, the classifier has not been well trained and therefore the logit vector $\mathbf{z}_i$ would often correspond to undesired classifier output. In this case, the classifier would not be prone to generate smaller uncertainty values in $\boldsymbol{\sigma}_i$, because smaller uncertainty would cause the multiple sampled logit vectors $\{\hat{\mathbf{z}}_{i,k}, k = 1, \ldots, K\}$ often having similar undesired classifier output as from $\mathbf{z}_i$. In contrast, at later training stage when the classifier has been well trained, the classifier may correctly predict the class of the input $\mathbf{x}_i$ based on the corresponding logit vector $\mathbf{z}_i$. This generally would be associated with smaller uncertainty estimate, because larger uncertainty would cause more varied logit vector samples $\{\hat{\mathbf{z}}_{i,k}\}$ and part of the samples which are more different from $\mathbf{z}_i$ could cause undesired classifier output. Therefore, the logit uncertainty estimate can be automatically adapted during

model training. The trend of uncertainty estimate over training process has been confirmed in experiments (Sect. 3.3). Similar analysis can be performed for individual training images $\mathbf{x}_i$'s, with harder images (i.e., prone to be incorrectly classified) often associated with larger logit uncertainty. As a result, the uncertainty logit estimate $\boldsymbol{\sigma}_i$ from the classifier can be naturally used to determine the size of neighborhood around $\mathbf{z}_i$ in the logit space, based on which multiple samples in the neighborhood can be generated for each input image and used for data augmentation during model training.

2.2 Comparison with Relevant Techniques

The multiple sampled logit vectors $\{\hat{\mathbf{z}}_{i,k}\}$ for each input image $\mathbf{x}_i$ can be considered as the classifier logit outputs of various input images similar to the original input $\mathbf{x}_i$, whether they correspond to the transformed versions of $\mathbf{x}_i$ or different instances (e.g., more lesion spots in images) from the same class. Therefore, the data augmentation in the logit space is a generalization of existing augmentation strategies in the input data space [4,26–28], but without requiring manual choice of augmentation types (e.g., various spatial transformations) as usually used in the input data space. Our method can also be considered as an extension of the data augmentation in the feature space [21,25]. However, augmentation in feature space does not assure that similar logit vectors would come from the same class. In addition, the general decreasing trend of uncertainty values over training stages from our method reminds people of the traditional simulated annealing technique for optimization [20]. From this aspect, the uncertainty estimate in our method can be considered as the temperature parameters, tuning the classifier training process such that the optimization is less likely trapped to a poor local solution. Note that the classifier with uncertainty estimate has been proposed previously [12] for uncertainty estimate of pixel classification. We used the uncertainty estimate novelly for data augmentation to improve the performance of classification with limited training data. Different from the method [12] whose loss function is based on the difference between the ground-truth vector and the mean vector of multiple output vectors, the loss function in our method is based on the average of the differences between the ground-truth vector and each output vector. Another difference is that our method adopts the uniform sampling rather than Gaussian normal random sampling for sample generation. Such differences cause significant performance improvement from our method.

3 Experiment

3.1 Experimental Setting

The proposed method was extensively evaluated on three medical image classification datasets, Skin40, Skin8, and Xray6. Skin40 is a subset of the 198-disease skin image dataset [17]. It contains 40 skin diseases, with 60 images for each disease. Skin8 is from the originally class-imbalanced ISIC2019 challenge

dataset [1]. Based on the number (i.e., 239) of images from the smallest class, 239 images were randomly sampled from every other class, resulting in the Skin8 dataset. Xray6 is a subset of ChestXray14 dataset [23], containing six diseases of X-ray images (Atelectasis, Cardiomegaly, Emphysema, Hernia, Mass, Effusion). Based on the smallest class (i.e., Hernia) which has only 110 images, the same number of images were randomly sampled from every other class, forming the small-sample Xray6 dataset. Due to limited images for each dataset, a five-fold cross-validation scheme was adopted, with four folds for training and another fold for testing each time.

For model training on each dataset, each image was resized to 224×224 pixels after random cropping, scaling, and horizontal flipping. For testing, only the same resizing was adopted. During model training, the stochastic gradient descent (batch size 64) with momentum (0.9) and weight decay(0.0001) were adopted. The initial learning rate 0.001 was decayed by 0.1 respectively on epoch 80, 100, 110. All models were trained for 120 epochs with observed convergence. As widely adopted, the initial model parameters were from the pre-trained model based on the natural image dataset Imagenet. Unless otherwise mentioned, the number of samples K was set 5 (different numbers generally lead to similar performance from the proposed method). The average and standard deviation of classification accuracy over the five folds based on the five-fold cross-validation scheme were used to evaluate the performance of each method.

3.2 Effectiveness Evaluation

The effectiveness of our method was evaluated by comparing with various data augmentation methods, including the basic augmentation (random cropping + scaling + flipping, 'Basic' in Table 1), Mixup [27], Manifold Mixup (MM) [21], Dropblock (DB) [9], Cutmix [26], Cutout [4], and RandomErase (RE) [28]. The originally proposed training loss for uncertainty estimate in the related study [12] was also used as a baseline ('UC' in Table 1). The suggested hyper-parameter settings in the original studies were adopted here. Note that the basic data augmentation was used in all the baselines and our method by default. As Table 1 shows, our method outperforms all the data augmentation methods by a clear margin (1%–4%) on all the three medical image classifications tasks, supporting the better effectiveness of our method in alleviating the over-fitting issue. The relatively smaller average accuracy and larger standard deviation on the Xray6 dataset might be due to the smaller dataset (totally 660 images) and the highly inter-class similarity. The little performance improvement from the UC method compared to the Basic method also confirms that the different formulation of the loss function (based on individual augmented logit vector rather than mean logit vector) in our method is crucial. Unexpectedly, some advanced augmentation techniques like Mixup, MM, and RE did not perform better than the Basic method. The fine-grained difference between different diseases in the medical classification tasks might cause the failure of these augmentation techniques.

The effectiveness of our method is further confirmed with various model backbones, including VGG16 [15], ResNet18 [10], ResNet50, SE-ResNet50 [11],

Table 1. Comparison with existing data augmentation methods on three medical image datasets. The model backbone is ResNet50.

Dataset	Basic	Mixup	MM	DB	CutMix	Cutout	RE	UC	Ours
Skin40	73.54 (1.21)	73.67 (0.75)	73.38 (1.91)	73.37 (1.05)	74.29 (1.78)	74.33 (1.29)	73.58 (1.25)	73.38 (1.65)	**76.13** (1.98)
Skin8	68.52 (2.14)	68.95 (3.14)	68.47 (1.27)	69.77 (3.10)	68.42 (1.81)	69.00 (2.05)	68.62 (1.31)	70.04 (2.87)	**72.03** (1.75)
Xray6	51.22 (5.81)	48.18 (4.85)	50.73 (3.74)	51.73 (3.98)	51.21 (3.41)	51.36 (5.08)	50.61 (4.61)	51.82 (6.83)	**52.61** (4.57)

Table 2. Performance comparison on different model backbones.

Backbone	Skin40			Skin8			Xray6		
	Basic	Cutout	Ours	Basic	Cutout	Ours	Basic	Cutout	Ours
ResNet18	71.71 (1.77)	69.58 (1.06)	**73.67** (2.10)	65.75 (2.43)	67.05 (0.67)	**68.74** (2.88)	46.52 (2.42)	46.21 (3.12)	**49.37** (2.38)
ResNet50	73.54 (1.21)	74.33 (1.29)	**76.13** (1.98)	68.52 (2.14)	69.00 (2.05)	**72.03** (1.75)	51.22 (5.81)	51.36 (5.08)	**52.61** (4.57)
VGG16	72.63 (1.74)	72.71 (1.71)	**74.92** (1.68)	68.42 (2.70)	70.62 (3.09)	**72.29** (1.66)	53.36 (4.85)	48.18 (4.80)	**54.34** (6.37)
SE-ResNet50	72.33 (1.30)	74.33 (1.23)	**75.88** (1.88)	68.27 (4.32)	69.32 (1.18)	**70.36** (2.81)	48.77 (5.39)	49.70 (4.63)	**52.34** (7.31)
EfficientNet-B3	69.67 (1.36)	69.54 (2.27)	**75.42** (1.97)	68.11 (1.80)	67.28 (2.13)	**70.98** (1.90)	50.96 (4.63)	47.42 (4.36)	**54.08** (6.21)
ViT-B_224	74.71 (0.60)	75.38 (0.84)	**77.38** (1.89)	67.16 (1.46)	67.38 (1.67)	**69.73** (2.52)	43.18 (6.81)	44.21 (5.96)	**45.30** (2.16)

EfficientNet-B3 [19], and the Transformer architecture ViT [5]. As shown in Table 2, although the classification performance varies across model backbones, our method consistently performs better than the representative baselines Basic and Cutout on each model backbone. This also suggests that our method is not limited to any specific model structure and can be applied to the training of models with various architectures. Note that more advanced backbones (e.g., EfficientNet) did not always perform better than more basic backbones (e.g., VGG and ResNet50). This again might be caused by various factors such as the fine-grained difference between different diseases, and the limited transferability of backbones from natural image dataset to small medical dataset.

3.3 Model Component Choice and Effect of Hyper-parameters

Effect of Random Sampler: During model training, uniform random sampler was adopted to generate multiple samples in the logit space based on each training data. Compared to the samples from Gaussian normal random sampler, samples from uniform random sampler are more varied within the neighborhood of each original logit vector, and therefore may correspond to more different augmentations in the input space. Since more augmentations can represent more different training data, it is expected that uniform random sampler would perform better than Gaussian normal random sampler in improving model performance. This is confirmed with different model backbones on Skin40 (Fig. 2).

Although samples from Gausian normal random sampler already increase the model performance compared to basic data augmentation in the input space, uniform random sampler helps further improve the performance consistently.

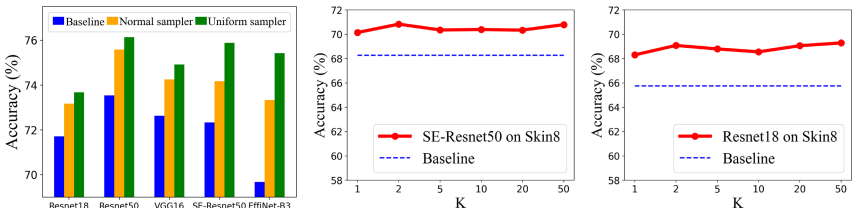

Fig. 2. Effect of random sampler (left) and sampling number K (middle and right).

Effect of Sampling Number K: While K was set 5 for all above experiments, we expect different K would result in similar classification performance because multiple iterations of training would equivalently generate various samples in the logit space. As Fig. 2 (middle and right) shows, with the two backbones SE-Resnet50 (middle) and Resnet18 (right) respectively, the performance of the classifier is relatively stable with respect to the number of samples K and all are clearly better than that from the Basic augmentation ('Baseline' in Fig. 2).

Effect of Uncertainty Estimate: Our method can directly estimate the uncertainty of logit elements for each training data at each training stage, and we expect the uncertainty decreases over training stages as analyzed in Sect. 2.1. This is confirmed in Fig. 3, which shows that the average (absolute) uncertainty over all training images generally decrease over training epochs. The adaptive uncertainty estimate can help adjust the size of neighborhood for sampling over training stages and training images. From Table 3, it is clear that automatic and adaptive uncertainty estimate works consistently well, while fixed uncertainty value even within the range of automatically estimated uncertainty (e.g., $\sigma = 0.05, 0.15, 0.35$) works well only on specific dataset with specific model backbone, supporting the necessity and effectiveness of adaptive uncertainty estimate for data augmentation in the logit space.

Table 3. Effect of adaptive uncertainty estimate on classification performance.

Uncertainty σ	Skin40			Skin8		
	ResNet18	ResNet50	SE-ResNet50	ResNet18	ResNet50	SE-ResNet50
Adaptive σ	73.67	76.13	75.88	68.74	72.03	70.36
Fixed $\sigma = 0.05$	73.79	74.08	75.54	69.01	71.35	70.35
Fixed $\sigma = 0.15$	73.45	76.20	74.92	68.36	70.99	69.47
Fixed $\sigma = 0.35$	73.38	75.67	75.87	68.31	72.08	69.62

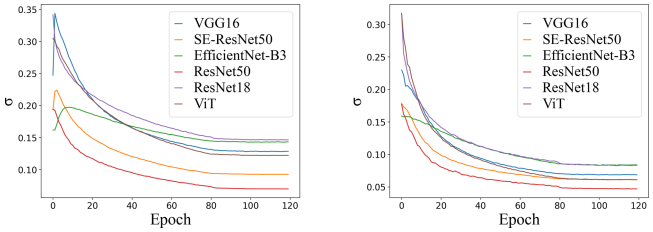

Fig. 3. Uncertainty estimate over training stages on Skin40 (left) and Skin8 (right).

4 Conclusion

To alleviate the over-fitting due to limited training data, a novel data augmentation method is proposed, not in the input or feature space, but in the logit space. Experimental evaluations on multiple datasets and model backbones confirm the effectiveness of the proposed method for improving classification performance. In future work, the proposed method will be applied to more medical image analysis tasks including imbalanced classification and classification on large datasets.

Acknowledgement. This work is supported by the National Natural Science Foundation of China (No. 62071502, U1811461), the Guangdong Key Research and Development Program (No. 2020B1111190001, 2019B020228001), and the Meizhou Science and Technology Program (No. 2019A0102005).

References

1. Combalia, M., et al.: Bcn20000: Dermoscopic lesions in the wild. arXiv preprint arXiv:1908.02288 (2019)
2. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: learning augmentation strategies from data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
3. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.: Randaugment: practical automated data augmentation with a reduced search space. Adv. Neural. Inf. Process. Syst. **33**, 18613–18624 (2020)
4. DeVries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552 (2017)
5. Dosovitskiy, A., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
6. Esteva, A., Kuprel, B., Novoa, R.A., Ko, J., Swetter, S.M., Blau, H.M., Thrun, S.: Dermatologist-level classification of skin cancer with deep neural networks. Nature **542**, 115–118 (2017)
7. Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., Cui, C., Corrado, G., Thrun, S., Dean, J.: A guide to deep learning in healthcare. Nat. Med. **25**, 24–29 (2019)
8. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: Proceedings of the 34th International Conference on Machine Learning, vol. 70, pp. 1126–1135 (2017)

9. Ghiasi, G., Lin, T.Y., Le, Q.V.: Dropblock: a regularization method for convolutional networks. In: Advances in Neural Information Processing Systems, vol. 31 (2018)
10. 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, pp. 770–778 (2016)
11. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7132–7141 (2018)
12. Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? In: Advances in Neural Information Processing Systems, vol. 30 (2017)
13. Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., Van Der Laak, J.A., Van Ginneken, B., Sánchez, C.I.: A survey on deep learning in medical image analysis. Med. Image Anal. **42**, 60–88 (2017)
14. Rusu, A.A., Rao, D., Sygnowski, J., Vinyals, O., Pascanu, R., Osindero, S., Hadsell, R.: Meta-learning with latent embedding optimization. In: 7th International Conference on Learning Representations (2019)
15. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: 3rd International Conference on Learning Representations (2015)
16. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
17. Sun, X., Yang, J., Sun, M., Wang, K.: A benchmark for automatic visual classification of clinical skin disease images. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 206–222. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_13
18. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H.S., Hospedales, T.M.: Learning to compare: relation network for few-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1199–1208 (2018)
19. Tan, M., Le, Q.V.: Efficientnet: rethinking model scaling for convolutional neural networks. In: Proceedings of the 36th International Conference on Machine Learning, vol. 97, pp. 6105–6114 (2019)
20. Van Laarhoven, P.J., Aarts, E.H.: Simulated annealing. In: Simulated Annealing: Theory and Applications, pp. 7–15 (1987)
21. Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Lopez-Paz, D., Bengio, Y.: Manifold mixup: Better representations by interpolating hidden states. In: Proceedings of the 36th International Conference on Machine Learning. vol. 97, pp. 6438–6447 (2019)
22. Vinyals, O., Blundell, C., Lillicrap, T., kavukcuoglu, k., Wierstra, D.: Matching networks for one shot learning. In: Advances in Neural Information Processing Systems, vol. 29 (2016)
23. Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., Summers, R.M.: ChestX-ray8: Hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3462–3471 (2017)
24. Wang, Y., Yao, Q., Kwok, J.T., Ni, L.M.: Generalizing from a few examples: a survey on few-shot learning. ACM Comput. Surv. **53**(3), 63:1–63:34 (2020)
25. Wang, Y., Huang, G., Song, S., Pan, X., Xia, Y., Wu, C.: Regularizing deep networks with semantic data augmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)

26. Yun, S., Han, D., Chun, S., Oh, S.J., Yoo, Y., Choe, J.: Cutmix: regularization strategy to train strong classifiers with localizable features. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 6022–6031 (2019)
27. Zhang, H., Cissé, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: beyond empirical risk minimization. In: 6th International Conference on Learning Representations (2018)
28. Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, pp. 13001–13008 (2020)