Regularizing CNNs using Confusion Penalty Based Label Smoothing for Histopathology Images

Somenath Kuiry[1][0000−0002−8462−5547], Alaka Das[1][0000−0003−2023−6566], Mita Nasipuri[2][0000−0002−3906−5309], and Nibaran Das[2][0000−0002−2426−9915]

[1] Department of Mathematics, Jadavpur University, Kolkata-32, West Bengal, India
[2] Department of CSE, Jadavpur University, Kolkata-32, West Bengal, India
mitanasipuri@gmail.com
{skuiry.math.rs, alaka.das, nibaran.das}@jadavpuruniversity.in

Abstract. Deep Learning, particularly Convolutional Neural Networks (CNN), has been successful in computer vision tasks and medical image analysis. However, modern CNNs can be overconfident, making them difficult to deploy in real-world scenarios. Researchers propose regularizing techniques, such as Label Smoothing (LS), which introduces soft labels for training data, making the classifier more regularized. LS captures disagreements or lack of confidence in the training phase, making the classifier more regularized. Although LS is quite simple and effective, traditional LS techniques utilize a weighted average between target distribution and a uniform distribution across the classes, which limits the objective of LS as well as the performance. This paper introduces a novel LS technique based on the confusion penalty, which treats model confusion for each class with more importance than others. We have performed extensive experiments with well-known CNN architectures with this technique on publicly available Colorectal Histology datasets and got satisfactory results. Also, we have compared our findings with the State-of-the-art and shown our method's efficacy with Reliability diagrams and t-distributed Stochastic Neighbor Embedding (t-SNE) plots of feature space.

Keywords: Regularization · Label Smoothing · CNN.

1 Introduction

Deep Learning especially Convolutional Neural Networks(CNN) has shown remarkable success in various computer vision tasks in past decades and has been adopted in medical image analysis as well [14][9][12]. However, modern CNNs tend to be overconfident about their predictions making them hard to deploy in real-world scenarios [19]. Researchers proposed various regularizing techniques [20], [13], [1] [23] [2] etc. and Label Smoothing(LS) [16] is one of them. Instead of using hard labels during training, the LS introduces soft labels for training data making the training classifier more regularized. In this process, a small amount

of weight is taken from the hard label target class and redistributed with the other classes equally(see Fig.1c). Another way of viewing the effectiveness of LS is in its application in medical image analysis. In benchmark datasets like CIFAR-10, ImageNet, MNIST, etc., the annotations are well-defined and have a high annotator agreement, which is not the case for the medical domain. There is a high annotator disagreement present between different annotators in the medical field. For example, in most situations, two radiologists cannot say with 100% confidence whether a particular image of a tumor is cancerous or not. Hence, using hard labels can make a classifier overconfident about its predictions. Thus LS can capture the disagreement or lack of confidence in the training phase making the classifier more regularized. Although the vanilla LS is simple and works quite well, this limits the objective of label smoothing as it treats all the other classes with equal importance [22]. In this paper, we have introduced a novel LS technique based on the confusion penalty. Basically, in every epoch, we keep noting the model confusion for each class. Instead of giving equal weightage to the rest of the classes, we gave weightage to those classes with whom model confusion is higher(see Fig. 1d). In summary, the contributions are as follows

- We have introduced a novel Label Smoothing technique for model regularization based on confusion penalty.
- We have performed extensive experiments with well-known classifier models and got satisfactory performance.
- We have shown the feature space t-distributed Stochastic Neighbor Embedding(t-SNE) visualizations to establish the effectiveness
- The proposed method has been compared with other State-of-the-art techniques with the publicly available Colorectal histology dataset[7].

The rest of the paper is organized as follows. In section 2, we briefly discussed the previous works about LS. In section 3, we discussed the whole methodology in detail. The experiments and findings of these experiments are shown in section 4 and section 5 respectively. Finally, we gave the conclusion in section 6

2   Related works

Training with hard labels leads to overconfident models which is not appropriate for tasks with high risk factors. Label smoothing [16] was first proposed in 2016, as a regularization technique to reduce the model overconfidence. Though the method is simple and intuitive, the drawback of this is it treats all other classes except the target class equally and assigns equal weights for them.

Pham et al. [11] suggested remapping targets in the field of medical image analysis to random values that are near to 1. They discovered that this approach increased model performance on the CheXpert Dataset [6] by about 1.4%. Additionally, Xi et al. [10] addressed accuracy issues by utilizing a spatial label smoothing technique to achieve sufficient performance with less reliance on well-annotated data. Krothapalli and Abbott[8] proposed an adaptive label smoothing based on relative object size within an image. For context-only

(a) Example Image

(b) Hard Label
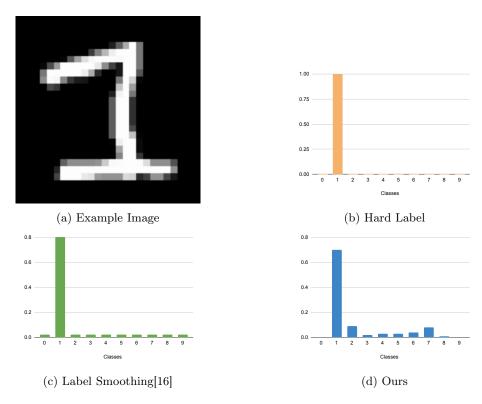
(c) Label Smoothing[16]

(d) Ours

Fig. 1: This is an example image taken from the MNIST dataset with the correct label as "1". The differences between Hard label, Vanilla label smoothing, and our label smoothing technique. Since the image has features of class "7" and "2", the weightage is given to those classes more than others, unlike the vanilla smoothing technique.

images, their method penalizes low-entropy (high-confidence) predictions while simultaneously assisting in the generation of accurate labels during training.

More recently, Wei et al. [21] utilized agreement-aware and confidence-aware label smoothing for calibrating neural networks for histopathology images. Zhang et al. [22], introduced a novel label smoothing technique Online label smoothing(OLS) on the CIFAR-100 dataset with ResNet-101 architecture.

3   Methodology

3.1   Preliminaries

Let $\mathcal{D} = \{(x_i, y_i)\}$, be the dataset where x_i and y_i denote the images and targets respectively. Let p and θ be the prediction probability distribution of the model and the target class distribution for a particular sample (x_i, y_i). In the traditional training with hard labels, a one-hot encoding(1 for the correct class and 0 for the incorrect classes) is used as the target distribution. So, for hard labels, we have $\theta(c == y_i|x_i) = 1$ and $\theta(c \neq y_i|x_i) = 0$ for $c = 1, 2, \ldots C$, where C is the total number of classes. Also, $p(c|x_i)$ is the probability score of input x_i for the class c. Hence, the traditional cross-entropy loss function for that sample (x_i, y_i) would be

$$
\begin{aligned}
\mathcal{L}_{\text{Hard}} &= -\sum_{c=1}^{C} \theta(c|x_i) \log p(c|x_i) \\
&= -\log p(c == y_i|x_i)
\end{aligned}
\tag{1}
$$

In the case of a vanilla soft label, we just replace the target distribution θ with a smooth target distribution ϕ by a weighted (α) average between the target distribution θ and a uniform distribution across the classes. The parameter α is called a smoothing parameter and the cross entropy loss function for this case is derived as

$$
\mathcal{L}_{LS} = -\sum_{c=1}^{C} \left[(1 - \alpha) * \theta(c|x_i) + \frac{\alpha}{C} \right] \log p(c|x_i)
\tag{2}
$$

3.2   Confusion Penalty based Label Smoothing(CPLS)

Though vanilla label smoothing works quite well in practice, it gives equal importance to all the other classes and overlooks the fact that not every class has similar characteristics [22]. In this paper, we introduce the Confusion Penalty Label Smoothing (CPLS), which captures the relationship or similarity between classes. The key idea here is to distribute the probabilities among the classes with similar features according to feature similarity. To find out these intra-class similarities, we keep track of the images from a particular class that is often confused with images belonging to other classes. This is done by simply saving

the confusion matrix of the validation data. Let $M_n = (m_{ij})$ denote the confusion matrix of validation data in epoch number n, where $n = 1, 2...$ etc. M_n is a $C \times C$ matrix where the rows denote the true class and the columns denote the predicted classes. Thus each cell j of row i of M_n indicates out of the total sample from class i, how many sample is classified as class j. To use this information as a target distribution, we normalize each confusion matrix row-wise, which will now indicate the confusion distribution of each class. We then take this distribution and use this in the next epoch as a smooth label using the following loss function

$$\mathcal{L}_{\text{CPLS}} = -\sum_{c=1}^{C} m_{ic} \log p(c|x_i) \tag{3}$$

The training procedure is shown in the Figure 2. In the next section, we discussed the detailed experimental setup of this work.
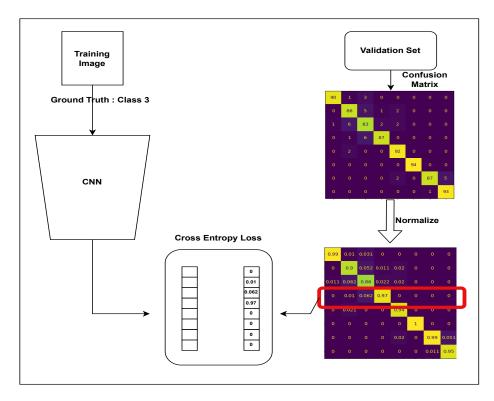
4   Experiments

Fig. 2: The training procedure of our CPLS method.

Algorithm 1 Training Procedure with CPLS

Constants: $\mathcal{D}_{train} = \{x_i, y_i\}$ = Training data with labels
$\quad \mathcal{D}_{val} = \{x_i, y_i\}$ = Validation data with labels
$\quad p(c|x_i)$ = Softmax output of image x_i for class c
$\quad M_n = (m_{ij})$ = Confusion Matrix of validation set. Initialize with Identity matrix
$\quad$ N = Threshold
$\quad 0 < \beta < 1$
Ensure:
1: **if** EPOCH $\leq$ N **then**
2:
3: $\quad$ **for** x_i to $\mathcal{D}_{train}$ **do**
4: $\qquad \mathcal{L}_{\text{Hard}} = -\log p(c == y_i|x_i)$
$\qquad$ Loss $= \mathcal{L}_{\text{Hard}}$
5: $\quad$ **end for**
6: $\quad$ **return** Loss
7: **else if** EPOCH $>$ N **then**
8:
9: $\quad$ **for** x_i to $\mathcal{D}_{train}$ **do**
10: $\qquad \mathcal{L}_{\text{CPLS}} = -\sum_{c=1}^{C} m_{ic} \log p(c|x_i)$
$\qquad$ Loss $= \beta\mathcal{L}_{\text{Hard}} + (1 - \beta)\mathcal{L}_{\text{CPLS}}$
11: $\quad$ **end for**
12: $\quad$ **for** x_i to $\mathcal{D}_{val}$ **do**
13: $\qquad$ Calculate Confusion matrix M_n
$\qquad M_n = \text{Normalized}(M_n)$
14: $\quad$ **end for**
15: $\quad$ **return** Loss
16: **end if**

4.1 Dataset

In this work, we have considered the publicly available Colorectal_histology [7]. This dataset consists of 5000 histological images of human colorectal cancer. Each image has dimensions of 150×150 pixels and belongs to one of eight classes: 'TUMOR', 'STROMA', 'COMPLEX', 'LYMPHO', 'DEBRIS', 'MUCOSA', 'ADIPOSE', and 'EMPTY'. This dataset is well-balanced, with an equal number of samples for each class. For training, validation, and testing, we divided the data in a $70 : 15 : 15$ ratio, respectively.

4.2 Experimental Setup

We implemented various state-of-the-art CNN classifiers available in the PyTorch library, including DenseNet-121[5], GoogLeNet[15], ResNet-18[4], Inception V3[17], and EfficientNet [18]. These classifiers were trained from scratch with similar hyperparameters, which made the experiment protocol simple and minimalistic. We did not use any kind of data augmentation as well to see the effectiveness of the label smoothing. We take these three models for further experiments with label smoothing. For the loss function, we have used the $\mathcal{L}_{\text{CPLS}}$

loss initially. The problem that we faced here is that in the starting phase, the model did not have any information about the confusion matrix, and with a random initialization, the model's performance degrades as it lacks the hard label from the beginning and the model tends to diverge. Hence, we have employed a hybrid loss function $\mathcal{L}$ as described in Equation 4, and the whole training process is given in Algorithm 1.

$$\mathcal{L} = \beta\mathcal{L}_{\mathbf{Hard}} + (1 - \beta)\mathcal{L}_{\mathbf{CPLS}} \qquad (4)$$

During training, we employ the hard label loss $\mathcal{L}_{\mathbf{Hard}}$ for a few epochs then we introduce the new loss function $\mathcal{L}$ into the training. The advantage of this is that initially, the model gets more and more confident about the images for a few epochs and then the new loss function keeps the model from over-fitting. Also, the model can have a more accurate confusion matrix for the updated loss to work with. Each classifier was trained with the same hyperparameters to make the experiment protocol simple. These models were trained for around 50 epochs with NVIDIA GeForce Quadro P5000 16 GB GPU in PyTorch environment. The loss curves are shown in the Figure 3.

5 Result and Discussion

A well-calibrated classifier's prediction probability reflects the true likelihood of the event of interest. For example, if a classifier predicts N number of images with the highest probability equal to 0.8, then 80% of those N images should be correct. The Expected Calibration Error(ECE)[3] measures this model calibration by calculating the weighted average error between the prediction confidence(probabilities) and accurate prediction percentage. In practice, we divide the confidence range(0 to 1) into a few bins(n) and calculate the weighted average of the differences between the accurate prediction percentage and the prediction confidence. The mathematical expression is given as

$$\text{ECE} = \sum_{m=1}^{n} \frac{|B_m|}{n} |\operatorname{acc}(B_m) - \operatorname{conf}(B_m)|$$

Here, B_m is a particular bin, $acc(B_m)$ is the percentage of correct classification, and $conf(B_m)$ is the confidence of those samples. The lower ECE indicates a well-calibrated model and is suitable for application. In this paper, we have considered the overall testing accuracy and Expectation Calibration Error(ECE)[3] as the evaluation metric for our primary classifier ResNet-18, Inception V3, EffienetNet V2, GooLeNet, and DenseNet-121.

In Table 1, we have shown the Testing accuracy and ECE obtained by different classifiers. Each network is trained with State-of-the-art smoothing techniques like vanilla label smoothing, and online label smoothing to compare with ours. From Table 1, we can see that vanilla and online label smoothing perform better than training with hard labels for almost every classifier apart from Efficient Net in terms of Test accuracy and ECE. However, our method CPLS outperforms every other in most situations. For GoogLeNet, Online label smoothing
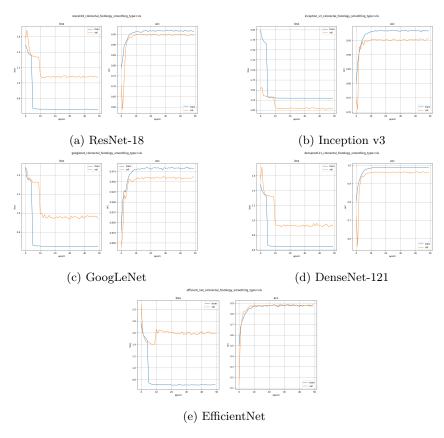
(a) ResNet-18

(b) Inception v3

(c) GoogLeNet

(d) DenseNet-121

(e) EfficientNet

Fig. 3: Loss and Accuracy curves of all the classifiers. Blue and Orange are for training and validation sets respectively

Table 1: Comparision of Testing Accuracy and ECE with Hard label, vanilla Soft label[16], Online label smoothing[22], and our CPLS method. Here the terms hard, vanilla, and ols represent Hard label, Vanilla Soft label, and Online label smoothing respectively.

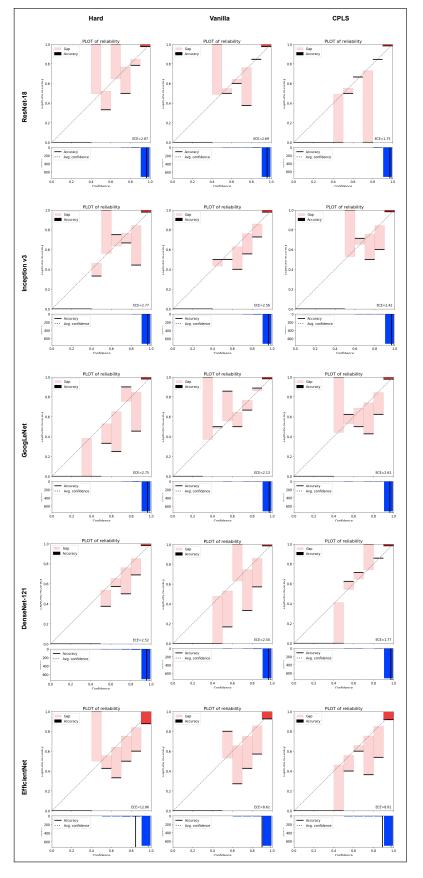| Networks	Accuracy	ECE
Resnet 18 + hard	0.9521	2.75
Resnet 18 + vanilla	0.9534	2.69
Resnet 18 + ols	0.9521	2.76
Resnet 18 + cpls	**0.9601**	**1.75**
InceptionV3 + hard	0.9627	2.77
InceptionV3 + vanilla	0.9654	2.56
InceptionV3 + ols	0.9601	2.48
InceptionV3 + cpls	**0.9694**	**2.42**
DenseNet 121 + hard	0.9694	2.52
DenseNet 121 + vanilla	0.9654	2.5
DenseNet 121 + ols	0.964	2.55
DenseNet 121 + cpls	**0.9734**	**1.77**
GoogLeNet + hard	0.964	2.75
GoogLeNet + vanilla	0.972	2.61
GoogLeNet + ols	**0.98**	**1.89**
GoogLeNet + cpls	0.964	2.13
EfficientNet + hard	**0.8989**	12.86
EfficientNet + vanilla	0.851	**8.62**
EfficientNet + ols	0.8789	9.09
EfficientNet + cpls	0.8896	8.81

Fig. 4: Comparison of Reliability Diagrams between all the classifiers with hard label, vanilla label smoothing, and our technique.

has the best performance. For EfficinetNet, hard labels achieve the best test accuracy but the vanilla label smoothing has the lowest ECE. In Figure 4, we have presented the Reliability Diagrams of each network trained with Hard label, Vanilla soft label, and our CPLS. These diagrams correspond to the ECE scores of each network.

For further analysis, we have shown the TSNE plots of feature space of each network trained with hard label, vanilla soft label, and our CPLS in Figure 5. In almost all situations, vanilla soft label created better clusters than the hard label. However, our CPLS techniques created better clusters in the feature space. This indicates the efficacy of our novel CPLS technique.

6 Conclusion

In this paper, we proposed a novel label smoothing techniques for model regularization. To create such a smooth label from a hard label, we utilized the confusion between frequent classes as the primary information from the validation dataset. Unlike the vanilla label smoothing, our technique does not give importance to each class equally. ResNet-18, Inception V3, GooLeNet, and EfficientNet were used in this experiment on the Colorectal Histology dataset. The Expected Calibration Error(ECE) is 1.35%, 0.5%, and 0.61% less on average than the hard label, vanilla label smoothing, and Online Label smoothing respectively. From the TSNE plots (Figure 5) of feature space, we can see that our CPLS can create better clusters than hard labels and vanilla soft labels, which proves the effectiveness of our novel label smoothing technique. Though in terms of accuracy and ECE, our method works well, it makes the models under-confident in some cases which can be seen in the Reliability Diagrams(Figure 4). In the future, we would like to address this issue. Also, we would like to use this technique in the domain of image segmentation as well.

Disclosure of Interests. The authors have no conflict of interest to disclose.

References

1. Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.
2. Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Dropblock: A regularization method for convolutional networks. *Advances in neural information processing systems*, 31, 2018.
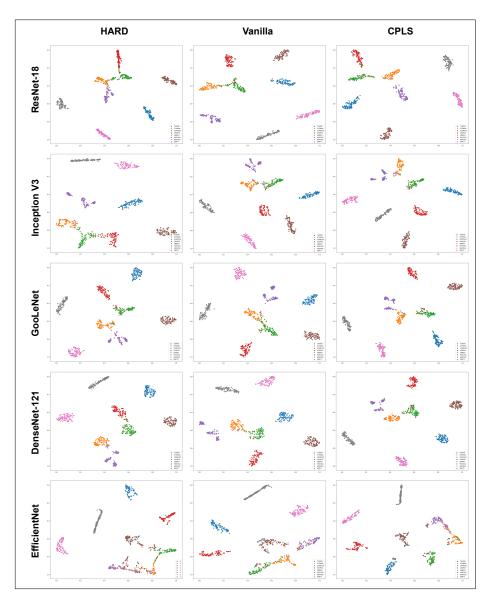
Fig. 5: t-SNE plots of feature space for all the classifiers trained with the hard label, vanilla label smoothing, and our technique.

3. Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.

4. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

5. Forrest Iandola, Matt Moskewicz, Sergey Karayev, Ross Girshick, Trevor Darrell, and Kurt Keutzer. Densenet: Implementing efficient convnet descriptor pyramids. *arXiv preprint arXiv:1404.1869*, 2014.

6. Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 590–597, 2019.

7. Jakob Nikolas Kather, Cleo-Aron Weis, Francesco Bianconi, Susanne M Melchers, Lothar R Schad, Timo Gaiser, Alexander Marx, and Frank Gerrit Z"ollner. Multiclass texture analysis in colorectal cancer histology. *Scientific reports*, 6:27988, 2016.

8. Ujwal Krothapalli and Lynn Abbott. One size doesn't fit all: Adaptive label smoothing. 2020.

9. Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

10. Xi Ouyang, Zhong Xue, Yiqiang Zhan, Xiang Sean Zhou, Qingfeng Wang, Ying Zhou, Qian Wang, and Jie-Zhi Cheng. Weakly supervised segmentation framework with uncertainty: A study on pneumothorax segmentation in chest x-ray. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part VI 22*, pages 613–621. Springer, 2019.

11. Hieu H Pham, Tung T Le, Dat Q Tran, Dat T Ngo, and Ha Q Nguyen. Interpreting chest x-rays via cnns that exploit hierarchical disease dependencies and uncertainty labels. *Neurocomputing*, 437:186–194, 2021.

12. Muhammad Imran Razzak, Saeeda Naz, and Ahmad Zaib. Deep learning for medical image processing: Overview, challenges and the future. *Classification in BioApps: Automation of Decision Making*, pages 323–350, 2018.

13. Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014.

14. Dinggang Shen, Guorong Wu, and Heung-Il Suk. Deep learning in medical image analysis. *Annual review of biomedical engineering*, 19:221–248, 2017.

15. Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

16. Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

17. Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

18. Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

19. Deng-Bao Wang, Lei Feng, and Min-Ling Zhang. Rethinking calibration of deep neural networks: Do not be afraid of overconfidence. *Advances in Neural Information Processing Systems*, 34:11809–11820, 2021.

20. Hongxin Wei, Renchunzi Xie, Hao Cheng, Lei Feng, Bo An, and Yixuan Li. Mitigating neural network overconfidence with logit normalization. In *International Conference on Machine Learning*, pages 23631–23644. PMLR, 2022.

21. Jerry Wei, Lorenzo Torresani, Jason Wei, and Saeed Hassanpour. Calibrating histopathology image classifiers using label smoothing. In *International Conference on Artificial Intelligence in Medicine*, pages 273–282. Springer, 2022.

22. Chang-Bin Zhang, Peng-Tao Jiang, Qibin Hou, Yunchao Wei, Qi Han, Zhen Li, and Ming-Ming Cheng. Delving deep into label smoothing. *IEEE Transactions on Image Processing*, 30:5984–5996, 2021.

23. Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.