Advanced Confidence Methods in Deep Learning

Yuval Meir[a], Ofek Tevet[a], Ella Koresh[a], Yarden Tzach[a], Ido Kanter[a,b,*]

[a] Department of Physics, Bar-Ilan University, Ramat-Gan, 52900, Israel.

[b] Gonda Interdisciplinary Brain Research Center, Bar-Ilan University, Ramat-Gan, 52900, Israel.

[*] Corresponding author at: Department of Physics, Bar-Ilan University, Ramat-Gan, 52900, Israel. E-mail address: ido.kanter@biu.ac.il (I. Kanter).

Abstract

The typical aim of classification tasks is to maximize the accuracy of the predicted label for a given input. This accuracy increases with the confidence, which is the maximal value of the output units, and when the accuracy equals confidence, calibration is achieved. Herein, several methods are proposed to enhance the accuracy of inputs with similar confidence, extending significantly beyond calibration. Using the first gap between the maximal and second maximal output values, the accuracy of the inputs with similar confidence is enhanced. The extension of the confidence or confidence gap to their minimal value among a set of augmented inputs further enhances the accuracy of inputs with similar confidence. Enhanced accuracies are demonstrated on EfficientNet-B0 trained on ImageNet and CIFAR-100, and VGG-16 trained on CIFAR-100. The results suggest improved applications for high-accuracy classification tasks that require manual operation for a given fraction of low-accuracy inputs.

1. Introduction

In offline and online classification tasks [1, 2], the typical aim is to maximize the accuracy of the predicted label for a given input [3, 4]. Because the trained network can fail or correctly predict the output label of a test input, estimating the accuracy of the network relies on an ensemble of inputs, that is, the test set. However, given a trained network and its accuracy, the question is whether the likelihood of the predicted label for a given input is correct. This likelihood depends on additional information, that is, the confidence provided by the propagation of an input through the trained network to the output layer.

For the prototypical classifier, the Perceptron [5, 6], the confidence is the induced field on the single output unit, resulting in above- or below- average accuracy [7], a concept that can also be extended to recurrent networks [8, 9]. For a feedforward network with several output units representing possible output labels, the predicted label is selected following the output unit with the maximal value. Using softmax normalization [10-12] for the output layer, the sum of the output units is one, and the maximal value serves as the confidence level. The likelihood to correctly predict a label is expected to increase with confidence level within the range [0, 1].

The estimation of accuracy as a function of confidence requires discretization into several bins because the validation set is finite. Hence, the fractions of correctly predicted, $f_c(i)$, and wrongly predicted, $f_w(i)$ validation inputs belonging to the i^{th} confidence bin are first calculated, resulting in the bin accuracy

$$Acc(i) = f_c(i)/(f_c(i) + f_w(i)) \qquad (1)\,,$$

where $Acc(i)$ is in the range [0, 1] and the average accuracy is obtained as follows:

$$Acc = \sum_i Acc(i) \cdot \big(f_c(i) + f_w(i)\big) = \sum_i f_c(i) \qquad (2).$$

where the bins are equally spaced in the range $[0, 1]$. For a given accuracy Acc, one can distinguish between the following three limiting cases for the distribution of $Acc(i)$, where each case is required for a different reality.

In the first scenario, $Acc(i) = Acc$, independent of i (Fig. 1a), resulting in the following entropy per input:

$$S = -Acc \cdot \ln(Acc) - (1 - Acc) \cdot \ln\left[\frac{1 - Acc}{M - 1}\right] \qquad (3),$$

where Acc is the probability for the selected output label to be correct and $\frac{1-Acc}{M-1}$ is the probability for the rest of the $M - 1$ labels, and M denotes the number of output labels. This scenario fits, for instance, the reality of automatic selection of an aisle in a store for a returned item, where a tolerance, $1 - Acc$, of mismatched aisles is acceptable following a disorder induced by the buyers. The second scenario is that for a fraction Acc $(1 - Acc)$ of the validation inputs, where the selected output label is correct (wrong) with a probability of one (Fig. 1b). The accuracy, Acc, is equal to that in the first scenario (Fig. 1a), however with lower averaged entropy per input

$$S = -(1 - Acc) \cdot \ln\left[\frac{1 - Acc}{M - 1}\right] \qquad (4).$$

This scenario fits the reality of a central storage, where returned items to aisles must be correctly placed, either automatically for a fraction Acc, or manually for a fraction $1 - Acc$. Another reality is a partially autonomous vehicle [13, 14], where manual operation is required for low-accuracy detection. The last popular scenario is calibration [11, 12, 15, 16] which aims that for each bin

$$Acc(i) = Confidence(i) \qquad (5).$$

This simple identity indicates, for instance, that the number of correctly classified multiple inputs (e.g. in a mini-batch) is given by their average confidence (Fig. 1c). In addition, the accuracy increases linearly with confidence, which can be easily deduced from the output layer. In this case, the entropy is expected to be higher than that in Eq. (4). However, this is not unique, because an ensemble of calibrated solutions for a given Acc exists. The number of parameters is twice the number of bins, $f_c(i)$ and $f_w(i)$, whereas the number of constraints is equal to the number of bins

$$Confidence(i) = f_c(i)/(f_c(i) + f_w(i)) \qquad (6),$$

with an additional constraint on the accuracy, as expressed in Eq. (2).

The goal of this study is to identify simple procedures based on the information obtained regarding the input and output layers to increase $Acc(i)$ beyond $Confidence(i)$ while maintaining Acc (Fig. 1d). The realization of the limiting case of non-overlapping distributions of $f_c(i)$ and $f_w(i)$ (Fig. 1b) seems unrealistic. Nevertheless, we propose methods that result in intermediate cases between this limit and calibration (Figs. 1b and c). The proposed methods rely on common back-propagation training with softmax normalization of the output layer but obtain additional information to improve $Acc(i)$. For simplicity, our validation test is taken as the entire test set. However, it can be divided into two parts, validation and test sets, and similar results can be obtained. The results are first demonstrated on EfficientNet-B0 [17] trained on ImageNet [18] and later extended to CIFAR-100 [19, 20] and VGG-16 [21].

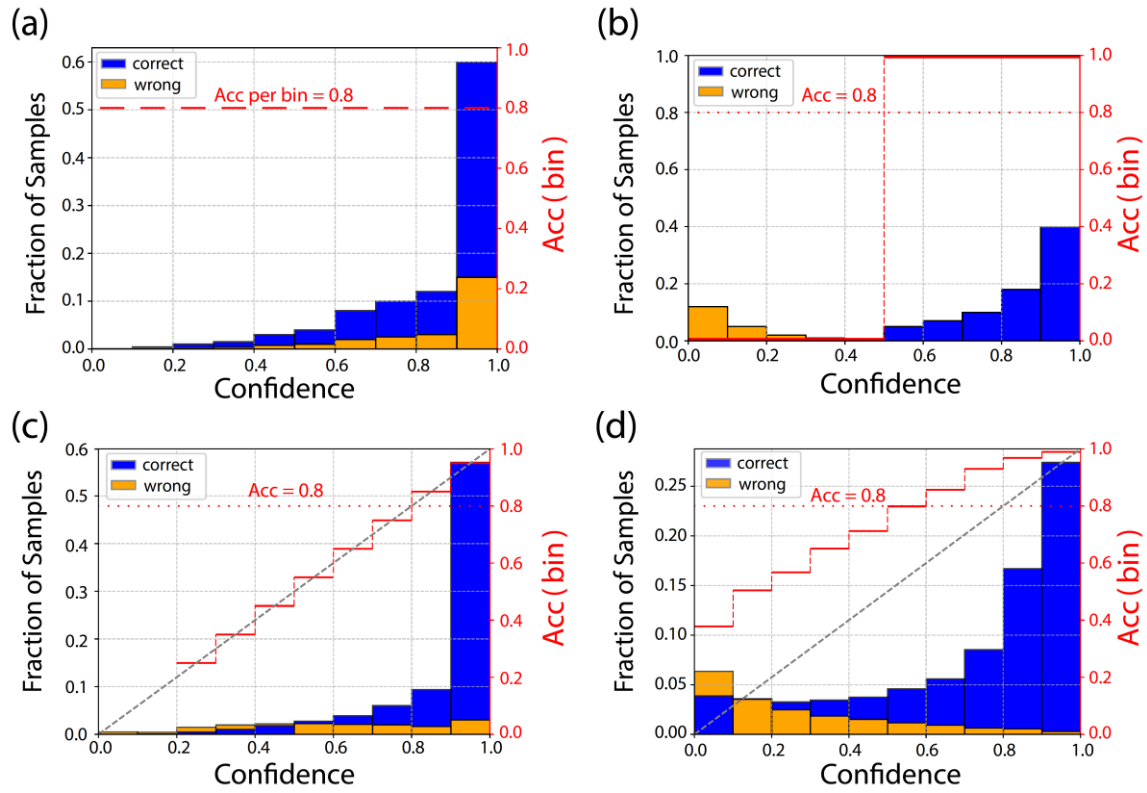

Fig. 1. Four scenarios for the accuracy per bin, $Acc(i)$ (red steps connected by dashed vertical red lines), as a function of confidence, representing the maximal value of the output units, where $Acc = 0.8$ (dotted horizontal red line). The fractions of validation inputs with correct ($f_c(i)$) or wrong ($f_w(i)$) output labels are represented by blue and orange, respectively. (a) $Acc(i) = Acc$ independent of i. (b) For a fraction 0.8 (blue), $Acc(i) = 1$, whereas for a fraction 0.2 (orange),

$Acc = 0$. (c) Calibration, $Acc(i) = Confidence(i)$. (d) An intermediate case between (b) and (c), where $Acc(i)$ is typically greater than $Confidence(i)$.

2. Material and methods

2.1 Architectures and Datasets

Two different architectures were examined. VGG-16 [22] and EfficientNet-B0 [17]. VGG-16 [22] and EfficientNet-B0 [17] were trained to classify CIFAR-100, and we used the pre-trained weights of EfficientNet-B0 on the ImageNet datasets.

2.2 Data preprocessing

For VGG-16, each input pixel of an image (32×32) from the CIFAR-100 databases was divided by the maximal pixel value, 255, multiplied by 2, and subtracted by 1, such that its range was $[-1, 1]$. During the training phase, data augmentation was used, derived from the original images, by random horizontally flipping and translating up to four pixels in each direction.

For EfficientNet-B0, the images were normalized by subtracting the average value of each color and dividing by its standard deviation. For CIFAR-100, the images were also expanded from their initial size of (32×32) to (224×224) [23]. Data augmentation was also used during the training phase, which included a random horizontal flip, a random rotation of up to two degrees, a random translation of the image of up to four pixels in each direction and a shear of up to two degrees.

2.3 Optimization

The cross-entropy cost function was selected for the classification task and was minimized using the stochastic gradient descent algorithm [4, 24]. Note that the cross-entropy is a standard used measure in deep learning which is simply related to Kullback-Leibler entropy [25-27]. The maximal accuracy was determined by searching through the hyper-parameters (see below). Cross-validation was confirmed using several validation databases, each consisting a fifth of the training set examples, randomly selected. The averaged results were

in the same standard deviation (Std) as the reported average success rates [28]. The Nesterov momentum [23] and L2 regularization method [4] were applied.

2.4 Hyper-parameters

The hyper-parameters η (learning rate), μ (momentum constant [29]), and α (regularization L2 [4]) were optimized for offline learning, using a mini-batch size of 100 inputs. The learning rate decay schedule [24, 30] was also optimized. A linear scheduler was used such that it was multiplied by the decay factor, q, every Δt epochs, and is denoted below as $(q, \Delta t)$. Different hyper-parameters were used for each one of the architectures on each classification task.

VGG-16 was trained using the following hyper-parameters and decay schedule for the training to reach maximal accuracy on CIFAR-100:

VGG-16 trained on CIFAR-100				
η	μ	α	epochs	$(q, \Delta t)$
0.002	0.975	4e-3	300	$(0.65, 20)$

Table 1. Hyper-parameters for VGG-16 trained on CIFR-100

EfficientNet-B0 was trained on CIFAR-100 using the following hyper-parameters and decay schedule for the training to reach maximal accuracy:

EfficientNet-B0 trained on CIFAR-100				
η	μ	α	epochs	$(q, \Delta t)$
0.01	0.9	0.001	200	$(0.975, 1)$

Table 2. Hyper-parameters for EfficientNet-B0 trained on CIFR-100

2.5 Hardware and software

We used Google Colab Pro and its available GPUs. We used Pytorch for all the programming processes.

3. Results

3.1. EfficientNet-B0 trained on ImageNet

The training of EfficientNet on ImageNet results in $Acc = 0.77$ [28], and the bin accuracy is slightly higher than its confidence level,

$$Acc(i) > Confidence(i) \qquad (6)$$

for all bins (Fig. 2a). This scenario differs, for instance, from that of ResNet-110 trained on CIFAR100, where $Acc(i)$ is less than the calibration level [15].

For a micro-canonical ensemble of all validation inputs belonging to a bin, $Confidence(i)$, one can determine an improved estimated accuracy for an input (Fig. 2b). The confidence now is the first gap among the output units, $Confidence_{gap}$, i.e. the maximal output unit minus the second maximal output unit. Its value is bounded by the upper limit of $Confidence(i)$ (e.g. 0.5 for $0.4 < Confidence(i) \leq 0.5$ (Fig.2b)). For high gap values (> 0.35 in (Fig. 2b)) $Acc(i)$ is greater than Acc, and less than Acc for smaller gaps. High gaps distinguish better between correct and wrong output labels for the following reason. It is expected that for inputs with selected wrong output labels, the correct output label value is typically non-negligible as a result of training; hence the gap is typically reduced.

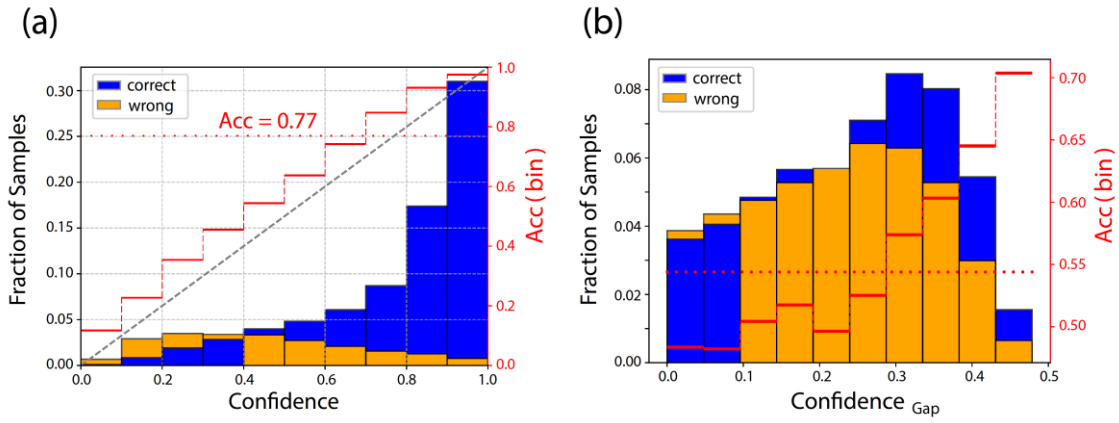

Fig. 2. Accuracy per bin, $Acc(i)$ (Eq. (1)), as a function of $Confidence$ for EfficientNet-B0 trained on ImageNet. The fractions of validation inputs with correct ($f_c\ (i)$) or wrong ($f_w\ (i)$) output labels are represented by blue and orange, respectively, and Acc by a horizontal doted red line. (a) $Acc(i)$ (red steps connected by dotted vertical red lines) and for comparison of the calibration relation (dashed black line). (b) $Acc(i)$ as a function of the first gap, $Confidence_{gap}$, the maximal output unit minus the second maximal output unit, for all inputs belonging to $Confidence$ in the range $(0.4, 0.5]$ in (a), where its $Acc(i)$ is represented by the dotted horizontal red line.

3.2. Confidence following the first gap

The zoom-in into $Confidence(i)$ and the classification of its inputs following their first gap (Fig. 2b), suggests the classification of all validation inputs following their gap (Fig. 3a). The range of $Confidence_{gap}$ is $[0, 1]$ as for the $Confidence$ (Fig. 1); however, inputs belonging to $Confidence(i)$ do not necessarily belong to $Confidence_{gap}(i)$. The accuracy per bin, $Acc(i)$, as a function of $Confidence_{gap}(i)$ (Fig. 3a) increases in comparison to $Confidence$ (Fig. 2a) and deviates further above calibration (Fig. 1c). However, accuracy, Acc, remained the same, because the fraction of inputs with low gaps and low accuracy increased. This increase in accuracy stems from a typically smaller gap for wrong output labels because the output value of the correct label typically competes with it and is expected to be non-negligible.

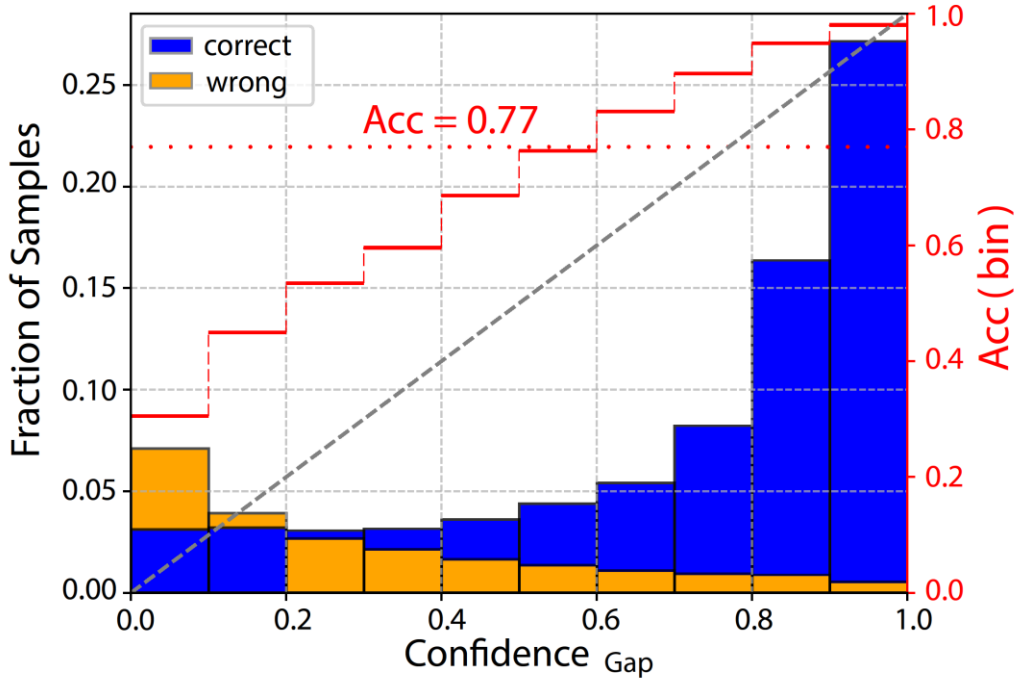

Fig. 3. Accuracy per bin, $Acc(i)$ (Eq. (1)), as a function of $Confidence_{gap}$ for EfficientNet-B0 trained on ImageNet. The fractions of validation inputs with correct ($f_c\ (i)$) or wrong ($f_w\ (i)$) output labels are represented by blue and orange, respectively, Acc (horizontal doted red line), $Acc(i)$ (red steps), and for comparison of the calibration relation (dashed black line). Note that $Acc(i)$ for $Confidence_{gap}$ is higher in comparison to $Confidence$ (Fig. 2a).

3.3. Augmentations enhance accuracy

Network training with augmented inputs [31] ensures that the test set accuracy is similar to that of all the augmented versions. This additional information is useful for enhancing $Acc(i)$ in the following ways. In the first method, the output label of a test input and its additional 24 augmented versions are calculated. Next, the predicted output label is selected following the majority, N, among the 25 output labels of the augmented inputs, and the final confidence is set as the minimum among these $N, Confidence_{min}$ (Fig. 4a). Typically, N is close to 25; however, test inputs for which $N < 10$ are present. Thereafter, $Acc(i)$ is calculated following Eq. (1) (Fig. 4a), indicating enhanced accuracy in comparison to the $N = 1$ case (Fig. 4d). This stems from the following

enhancing trend; for correctly predicted output labels, the confidence of each augmented input is expected to exhibit a slight variance around its average. However, for wrongly predicted output labels competing with the correct label, a higher variance is expected and the minimal variance is dominated by the tail of the distribution. Hence, a shift to $Confidence_{min}$ values is expected, as indicated by the comparison of the orange distributions in Figs. 2a and 4a.

Similar to Fig. 2b, an improved estimated accuracy for an input can be achieved by following the first gap in the output units of the selected augmentation with $Confidence_{min}$. For a microcanonical ensemble of inputs in the range $0.3 < Confidence_{min}(i) \leq 0.4$, for instance, accuracy is greater than the average one for $Confidence_{gap} > 0.1$ and it increases towards ~ 0.85 for $Confidence_{gap}$ approaching 0.4 (Fig. 4b).

Selecting the $Confidence_{gap}$ (Fig. 3) as the minimal gap obtained from the majority of augmented inputs with the same output label further improves accuracy, significantly above calibration (Fig. 4c). Here, the two aforementioned enhancing trends are combined to further separate the distributions of the correctly and wrongly predicted output labels (blue and orange distributions illustrated in Fig. 4c). The first effect is that the first gap is expected to be higher for the correctly predicted output label, because for a wrong output label the competition with the correct one tends to decrease the gap. Similarly, the minimal gap among the multiple first gaps, stemming from the augmented inputs, is dominated by the tail of their distribution for the wrongly predicted output labels. Hence, it is expected to decrease further and enhance accuracy. Nevertheless, Acc is unaffected because the fraction of inputs with small $Confidence_{gap}$ is higher (Fig. 4c).

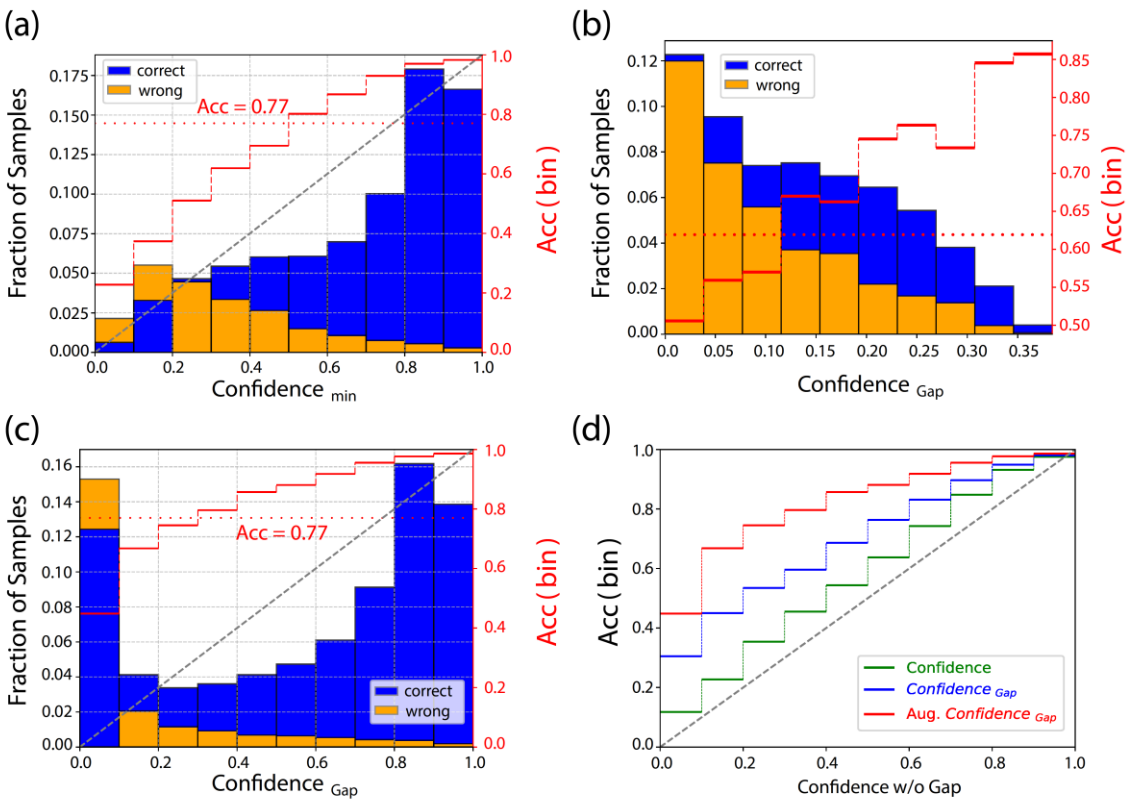

Fig. 4. Enhanced confidence using augmentations for EfficientNet-B0 trained on ImageNet. (a) $Acc(i)$ as a function of minimal confidence, $Confidence_{min}$, for an output label, selected by the most common one among the 25 augmented inputs. Acc (horizonal dotted red line) and calibration (dashed black line), in panels (a–c). (b) Similar to Fig. 2b, for inputs belonging to $0.3 < Confidence_{min} \leq 0.4$. (c) Similar to (a), where $Confidence_{min}$ is replaced by $Confidence_{gap}$, the minimal first gap that is obtained from the most common output label of the augmented inputs. (d) $Acc(i)$ as a function of $Confidence$ (green) as illustrated in Fig. 2a, $Confidence_{gap}$ (blue) as illustrated in Fig. 3, and $Confidence_{gap}$ (red) using augmentations as illustrated in Fig. 4a.

3.4. EfficeintNet-B0 and VGG-16 trained on CIFAR-100

The results presented for EfficientNet-B0 trained on ImageNet are extended to a different dataset, CIFAR-100, and to a different deep architecture, VGG-16 [21, 32].

Applying the aforementioned procedures to EfficientNet-B0 trained on CIFAR-100 result in similar trends (Figs. 5a and 4d). The accuracy as a function of $Confidence$ is less than the calibration level (Fig. 5a, green). Enhanced accuracy is obtained for $Confidence_{gap}$ (Fig. 5a, blue), which is further enhanced for $Confidence_{gap}$ among 25 augmentations for each input (Fig. 5a, red), representing the same trends as illustrated in Fig. 4d. Similar trends are obtained for VGG-16 trained on CIFAR-100 (Fig. 5b), where the enhancement of the $Confidence_{gap}$ (blue) over $Confidence$ (green) is less than that illustrated in Fig. 5a.

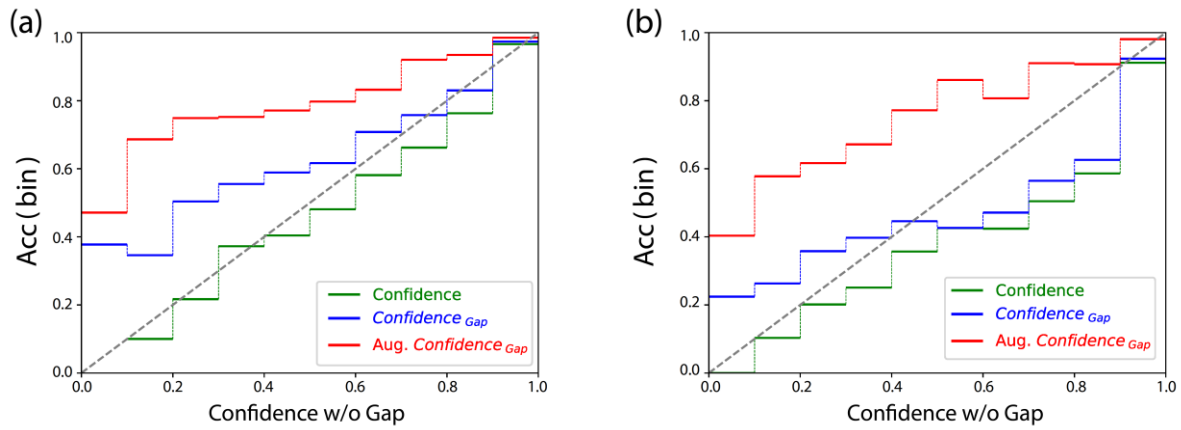

Fig. 5. Enhanced $Acc(i)$ for a different dataset and deep architecture using similar presentation as illustrated in Fig. 4d. (a) EfficientNet-B0 trained on CIFAR-100. (b) VGG-16 trained on CIFAR-100.

4. Conclusions

Several methods were proposed to enhance the accuracy of a given input, indicating that the actual information in the input and output layers is not fully utilized by the $Confidence$ measure.

The first method was within the standard $Confidence$ measure. For a microcanonical ensemble of inputs belonging to a bin (Figs. 2b and 4b) accuracy was enhanced as a function of the first gap, which is the difference

between the maximal and the second maximal output units. For a fraction of these inputs with large first gap accuracy was significantly above the bin's accuracy, $Acc(i)$ (Fig. 2b).

The second method replaced the $Confidence$ measure with $Confidence_{gap}$, the first gap as defined in the first method but for all validation inputs. The range of $Confidence_{gap}$ remained $[0, 1]$ and the profile of its accuracy was enhanced in comparison with $Confidence$ (Fig. 3). This relies on the trend that the confidence of the correct and wrong output labels might be the same. However, the first gap for the wrong one is expected to be smaller because in such events, the output of the correct label is not negligible.

The third method relied on several augmented inputs that were expected to yield similar confidence levels when the predicted label was correct. The new confidence measure was the minimum confidence among the augmented inputs, selecting their most common output label (Fig. 4a). The enhanced accuracy trend relies on the expectation that for an input selecting a wrong output label, some augmented inputs generate a significantly lower minimal confidence in comparison to inputs selecting the correct output label.

The fourth method combined the second and third methods (Fig. 4c). The confidence measure was replaced by the minimal first gap among the augmented inputs, selecting their most common output label. The abovementioned two accuracy enhancement trends are valid in this scenario and result in the maximal accuracy profile among the four presented methods. Enhancing the accuracy profile even further using a more complex confidence measure, for instance, depending on several gaps, may be possible; however, this requires further research.

For a given accuracy, the scenario for minimal entropy is given by two non-overlapping distributions for the correct and wrong output labels (Fig. 1b); however, approaching or approximating such a reality is uncertain. Temperature scaling was found to affect the accuracy profile and to achieve calibration [11, 12, 15, 16]. It adds a temperature coefficient to the softmax activation function which scales the output values. Hence, this temperature scaling mechanism, along with one of the proposed advanced accuracy methods might be used to approximate a profile with minimal entropy (Fig. 1b).

However, the possible demand for varying temperatures for each bin is expected to increase the complexity of such optimization procedures.

[1]T.L. Watkin, A. Rau, M. Biehl, The statistical mechanics of learning a rule, Reviews of Modern Physics, 65 (1993) 499.
[2]E. Agliari, F. Alemanno, M. Aquaro, A. Barra, F. Durante, I. Kanter, Hebbian dreaming for small datasets, Neural Networks, (2024.106174 (
[3]Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature, 521 (2015) 436-444.
[4]J. Schmidhuber, Deep learning in neural networks: An overview, Neural networks, 61 (2015) 85-117.
[5]F. Rosenblatt, The perceptron, a perceiving and recognizing automaton Project Para, Cornell Aeronautical Laboratory, 1957.
[6]M. Minsky, S. Papert, Perceptrons cambridge, MA: MIT Press. zbMATH.(1969) ,
[7]L. Ein-Dor, I. Kanter, Confidence in prediction by neural networks, Phys Rev E, 60 (1999) 799.
[8]A. Barra ,A. Bernacchia, E. Santucci, P. Contucci, On the equivalence of Hopfield networks and Boltzmann machines, Neural Networks, 34 (2012) 1-9.
[9]A. Barra, G. Genovese, F. Guerra, Equilibrium statistical mechanics of bipartite spin systems, Journal of Physics A: Mathematical and Theoretical, 44 (2011) 245002.
[10]T. Pearce, A. Brintrup, J. Zhu, Understanding softmax confidence and uncertainty, arXiv preprint arXiv:2106.04972.(2021) ,
[11]M. Minderer, J. Djolonga, R. Romijnders, F. Hubis, X. Zhai, N. Houlsby ,D. Tran, M. Lucic, Revisiting the calibration of modern neural networks, Advances in Neural Information Processing Systems, 34 (2021) 15682-15694.
[12]Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. Dillon, B. Lakshminarayanan, J. Snoek, Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift, Advances in neural information processing systems, 32.(2019)
[13]M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L.D. Jackel, M . Monfort, U. Muller, J. Zhang, End to end learning for self-driving cars, arXiv preprint arXiv:1604.07316.(2016) ,
[14]A. Goldental, I. Kanter, A minority of self-organizing autonomous vehicles significantly increase freeway traffic flow, Journal of Physics A: Mathematical and Theoretical, 53 (2020) 414001.
[15]C. Guo, G. Pleiss, Y. Sun, K.Q. Weinberger, On calibration of modern neural networks, in: International conference on machine learning, PMLR, 2017, pp. 1321-1330.
[16]C. Wang, Calibration in deep learning: A survey of the state-of-the-art, arXiv preprint arXiv:2308.01222.(2023) ,
[17]M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International conference on machine learning, PMLR, 2019, pp. 6105-6.114
[18]J. Deng, A large-scale hierarchical image database, Proc. of IEEE Computer Vision and Pattern Recognition, 2009.(2009) ,
[19]A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images.(2009) ,
[20]P. Singh, V.K. Verma, P .Rai, V.P. Namboodiri, Hetconv: Beyond homogeneous convolution kernels for deep cnns, International Journal of Computer Vision, 128 (2020) 2068-2088.
[21]S. Liu, W. Deng, Very deep convolutional neural network based image classification using small training sample size, in: 2015 3rd IAPR Asian conference on pattern recognition (ACPR), IEEE, 2015, pp. 730-734.
[22]K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556.(2014) ,
[23]R .Keys, Cubic convolution interpolation for digital image processing, IEEE transactions on acoustics, speech, and signal processing, 29 (1981) 1153-1160.

[24]K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770-778.
[25]A.E. Abbas, A. H. Cadenbach, E. Salimi, A Kullback–Leibler view of maximum entropy and maximum log-probability methods, Entropy-Switz, 19 (2017) 232.
[26]A. Barra ,M. Beccaria, A. Fachechi, A new mechanical approach to handle generalized Hopfield neural networks, Neural Networks, 106 (2018) 205-222.
[27]A. Fachechi, A. Barra, E. Agliari, F. Alemanno, Outperforming RBM feature-extraction capabilities by “dreaming” mechanism, IEEE transactions on neural networks and learning systems.(2022) ,
[28]Y. Meir, Y. Tzach, S. Hodassman, O. Tevet, I. Kanter, Towards a universal mechanism for successful deep learning, Scientific Reports, 14 (2024) 5881.
[29]A. Botev, G. Lever ,D. Barber, Nesterov's accelerated gradient and momentum as approximations to regularised update descent, in: 2017 International joint conference on neural networks (IJCNN), IEEE, 2017, pp. 1899-1903.
[30]K. You, M. Long, J. Wang, M.I. Jordan, How does learning rate decay help modern neural networks?, arXiv preprint arXiv:1908.01878.(2019) ,
[31]L. Perez, J. Wang, The effectiveness of data augmentation in image classification using deep learning, arXiv preprint arXiv:1712.04621.(2017) ,
[32]Y. Meir ,Y. Tzach, R.D. Gross, O. Tevet, R. Vardi, I. Kanter, Enhancing the accuracies by performing pooling decisions adjacent to the output layer, Sci Rep-Uk, 13 (2023) 13385.