Network Calibration by Temperature Scaling based on the Predicted Confidence

Lior Frenkel Jacob Goldberger
Faculty of Engineering, Bar-Ilan University, Israel
lior.frenkel@biu.ac.il, jacob.goldberger@biu.ac.il

***Abstract*—Calibrating neural networks is crucial in applications where the decision making depends on the predicted probabilities. Modern neural networks can be poorly calibrated. They tend to overestimate probabilities when compared to the expected accuracy. This results in a misleading reliability that corrupts our decision policy. We show that the magnitude of calibration error depends on the predicted confidence for each sample. This prediction confidence calibration paradigm is then applied to the concept of temperature scaling. We describe an optimization method that finds the suitable temperature scaling for each bin of a discretized value of prediction confidence. We report extensive experiments on a variety of image datasets and network architectures. Our approach achieves state-of-the-art calibration with a guarantee that the classification accuracy is not altered.**

***Index Terms*—neural networks, network calibration, temperature scaling, Expected Calibration Error (ECE)**

I. Introduction

Probabilistic machine learning algorithms output confidence scores along with their predictions. Ideally, these scores should match the true correctness probability. However, modern deep learning models still fall short in giving useful estimates of their predictive uncertainty. The lack of connection between the model's predicted probabilities and the confidence of model's predictions constitutes a key obstacle to the application of neural network models to real-world problems, such as decision-making systems. Quantifying uncertainty is especially critical in real-world tasks such as automatic medical diagnosis [1] and perception tasks in autonomous driving [2]. A classifier is said to be calibrated if the probability values it associates with the class labels match the true probabilities of the correct class assignments. Modern neural networks have been shown to be more overconfident in their predictions than their predecessors even though their generalization accuracy is higher, partly due to the fact that they can overfit on the negative log-likelihood loss without overfitting on the classification error [3].

Various confidence calibration methods have recently been proposed in the field of deep learning to overcome the overconfidence issue. Most calibration strategies perform calibration as a post processing step using an already trained model. Post-hoc scaling approaches to calibration (e.g. Platt scaling [4], isotonic regression [5], and temperature scaling [6]) are widely used. They use hold-out validation data to learn a calibration map that transforms the model's predictions to be better calibrated. Temperature scaling is the simplest and most effective calibration method [6] and is the current standard practical calibration method. Guo et al. [6] investigated several scaling models, ranging from single-parameter based temperature scaling to more complicated vector/matrix scaling. They reported poor performance for vector/matrix scaling calibration. To avoid overfitting, Kull et al. [7] suggested regularizing matrix scaling with an L_2 loss on the calibration model weights. Most of these calibration methods extend single parameter temperature scaling by making the selected temperature a linear function of the logits that are computed for the class-set. For example, in vector scaling [6], each class has its own temperature scaling.

In this study we take a different approach and propose an extension of temperature scaling that assigns a suitable temperature scaling to a given instance as a non-linear function of the confidence of the predicted class (i.e. the probability of the class with the highest logit). We show that, unlike vector and matrix scaling [7] and other recently proposed methods (e.g. [10]), we can easily find the optimal calibration parameters and no hyper parameters are needed to be tuned. The proposed calibration does not change the hard classification decision, which allows it to be applied on any trained network and guarantees to retain the original classification accuracy in all the tested cases. We evaluate our method against leading calibration approaches on various datasets and network architectures and show that it outperforms existing methods on improving the *expected calibration error* (ECE) [11] calibration measure.

II. Calibration Problem Formulation

Let x be an input vector to a classification network with k classes. The output of the network is a vector of k values $z_1, ..., z_k$. Each of these values, which are also called *logits*, represents the score for one of the k possible classes. The logits' vector is transformed into a probabilities vector by a *softmax* layer: $p(y = i|x) = \frac{exp(z_i)}{\sum_j \exp(z_j)}$. Although these values uphold the mathematical terms of probabilities, they do not represent any actual probabilities of the classes.

The predicted class for a sample x is calculated from the probabilities vector by $\hat{y} = \arg\max_i p(y = i|x) = \arg\max_i z_i$ and the predicted *confidence* for this sample is defined by $\hat{p} = p(y = \hat{y}|x)$. The *accuracy* of the model is defined by the true probability that the predicted class $\hat{y}$ is correct. The network is said to be *calibrated* if for each sample the confidence is equal to the accuracy. For example,

if we collect ten samples, each having an identical confidence score of 0.8, we then expect an 80% classification accuracy for the ten samples.

A popular metric used to measure model calibration is the ECE [11], which is defined as the expected absolute difference between the model's confidence and its accuracy. Since we only have finite samples, the ECE cannot in practice be computed using this definition. Instead, we divide the interval $[0,1]$ into m equispaced bins, where the i^{th} bin is the interval $\left(\frac{i-1}{m}, \frac{i}{m}\right]$. Let B_i denote the set of samples with confidences $\hat{p}$ belonging to the i^{th} bin. The accuracy A_i of this bin is computed as $A_i = \frac{1}{|B_i|}\sum_{t\in B_i} \mathbb{1}\left(\hat{y}_t = y_t\right)$, where $\mathbb{1}$ is the indicator function, and $\hat{y}_t$ and y_t are the predicted and ground-truth labels for the t^{th} sample. A_i is the relative number of correct predictions of instances that were assigned to B_i based on the confidence. Similarly, the confidence C_i of the i^{th} bin is computed as $C_i = \frac{1}{|B_i|}\sum_{t\in B_i} \hat{p}_t$, i.e., C_i is the average confidence of all samples in the bin. The ECE can be approximated as the weighted average of the absolute difference between the accuracy and confidence of each bin:

$$\text{ECE} = \sum_{i=1}^{m} \frac{|B_i|}{n} \left|A_i - C_i\right| \tag{1}$$

where n is the number of samples in the validation set. Note that $A_i > C_i$ means the network is under-confident at the i^{th} bin and $C_i > A_i$ implies that the network is over-confident.

One disadvantage of ECE is its uniform bin width. For a well trained model, most of the samples lie within the highest confidence bins; hence, these bins dominate the value of the ECE. For this reason, we can consider another metric, AdaECE (Adaptive ECE), where bin sizes are calculated so as to evenly distribute samples between bins [12]:

$$\text{AdaECE} = \frac{1}{m}\sum_{i=1}^{m} \left|A_i - C_i\right| \tag{2}$$

such that each been bin contains $1/m$ of the data points with similar confidence values.

III. Temperature Scaling based on the Predicted Confidence

Temperature Scaling (TS) is a simple yet highly effective technique for calibrating prediction probabilities [6]. It uses a scalar temperature parameter $T > 0$ to rescale logit scores before applying the softmax function to compute the class distribution. Since the same T is used for all classes, the softmax output with scaling has a monotonic relationship with the unscaled output. To get an optimal temperature T for a trained model, we can minimize the negative log likelihood for a held-out validation dataset. In the case of a single temperature parameter, direct minimization of the ECE measure (1) on the validation set was shown to yield better calibration results [13]. This is not surprising since we directly optimize the same calibration measure on the validation set that is finally evaluated on the test set.

The essence of calibration involves manipulating the prediction confidence (the probability of the most likely class). Since the goal of calibration is making the confidence prediction more accurate, it makes sense to calibrate the network based on this quantity which is a non-linear function of the logit vector. In this study we investigated a temperature scaling calibration strategy where the most suitable temperature of a given instance depends on its predicted confidence.

Let $f : [0,1] \rightarrow [0,\infty)$ be a function from the confidence probability value to a calibration temperature. This function f is denoted Confidence based Temperature (CBT). The calibration of an instance x is done as follows. Let the network output logits be $z = (z_1, ..., z_k)$ and let $c(z) = \max_i \exp(z_i)/(\sum_j \exp(z_j))$ be the corresponding confidence. The calibrated prediction probabilities are:

$$p(y = i|x) = \frac{\exp(z_i/f(c(z)))}{\sum_{j=1}^{k}\exp(z_j/f(c(z)))}, \qquad i = 1, \ldots, k. \tag{3}$$

We use a validation set to learn the CBT function. It is difficult to directly estimate the model's accuracy distribution and therefore, similar to the computation of ECE measure, we divide the validation set into bins according to their confidence values within the unit interval. We use the ECE measure as the objective score when finding the optimal CBT function. We allow a different temperate scaling for each ECE bin which results in a piece-wise constant form of the CBT function. We chose here to define our objective function based on the adaECE variant rather than ECE since in ECE accuracy at low confidence bins are computed using a small number of validation samples which makes the temperature estimates less robust.

Let A_i and C_i be the accuracy and confidence of the points in the i-th bin set B_i. Denote the average confidence at bin i after scaling all the instances in B_i by a temperature T_i by $C_i(T_i)$:

$$C_i(T_i) = \frac{1}{|B_i|}\sum_{t\in B_i} \max_{j=1}^{k} \frac{\exp(z_{tj}/T_i)}{\sum_{l=1}^{k}\exp(z_{tl}/T_i)} \tag{4}$$

s.t. $z_{t1}, ..., z_{tk}$ are the logit values computed by the network that is fed by x_t. To find the best CBT function we look for a temperature set that minimizes the following adaECE score:

$$L(T_1, ..., T_m) = \frac{1}{m}\sum_{i=1}^{m} \left|A_i - C_i(T_i)\right|. \tag{5}$$

We can perform the minimization of L for each bin separately. We can further apply a grid search to find T_i that satisfies $A_i = C_i(T_i)$. It can be easily verified that $C_i(T_i)$ is a monotonically decreasing function of T_i. We can thus apply a fast binary search to find the optimal temperature. Note that the minimum confidence value is $1/k$ where k is the number of classes. Hence, if the average accuracy is less than $1/k$ we cannot make the confidence coincides exactly with the accuracy. This usually occurs at the lowest bin.

We can thus find temperature values $T_1, ..., T_m$ such that the adaECE score (5) of the validation is exactly zero. This does not imply, however, that the adaECE (2) score of the calibrated validation set is zero. It can easily be verified that

a network that is more confident at point x than at point y can become less confident at x than y after a temperature scaling calibration (even if the same temperature was applied to the two data points). The calibration can thus change the order of the validation points when sorted according to their confidence. This alters the partition of the validation set into bins and causes that the adaECE score (2) of the calibrated validation set is not necessarily zero. We can thus apply the optimization of the adaECE score (5) on the calibrated validation set in an iterative manner. At each step we minimize the score (5) given the current calibration of the validation set. To set the number of iterations we compute the adaECE score of the validation set after each iteration and choose the number of iterations that yields the minimal adaECE score. The number of bins is determined in a similar way.

At the inference phase we calibrate a given point x by tracking its bin membership at each iteration. Denote the temperature-set learned at the s-th iteration by $T_{s,1}, ..., T_{s,m}$. The temperature used to calibrate x at the s-th iteration is $T_{s,b(x,s)}$ such that $b(x,j)$ is the bin containing the confidence value of the calibration of x at the j-th iteration. This bin is found based on the boundaries of the learned bins and the current confidence that is computed by the scaled logits of point x at the s-th iteration:

$$z_i/(T_{1,b(x,1)} \times ... \times T_{s-1,b(x,s-1)}), \qquad i = 1, ..., m.$$

Algorithm 1 Confidence based Temperature Scaling (CBT) - Train

input: A validation dataset $x_1, ..., x_n$. Each x_t is fed into a k-class classifier network to produce logits $z_{t1}, ..., z_{tk}$.
for $s = 1, ..., S$ **do**
  Divide the dataset into m equal size sets $B_{s1}, ..., B_{sm}$ based on the confidence values.
  for $i = 1, ..., m$ **do**
    Compute the average accuracy A_{si} and confidence C_{si} based on the points in B_{si}.
    Apply a binary search to find a temperature T_{si} that satisfies $A_{si} = C_{si}(T_{si})$.
    Divide all the logits of the points in B_{si} by T_{si}.
  end for
end for
output: The temperature sets and the bins' interval borders for all the iterations.

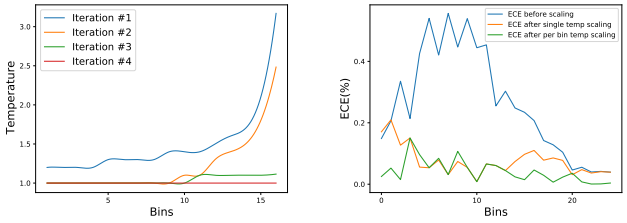
The train and inference phases of the CBT algorithm are summarized in Algorithm boxes 1 and 2 respectively. CBT has the desirable property that it does not affect the hard-decision accuracy since the same temperature scaling is applied to all the logits. This guarantees that the calibration does not impact the accuracy. Note that both vector and matrix scaling do affect model accuracy and may decrease it.

There are several other calibration methods based on non linear transformations of logits vectors. Gupta et al. [10] built a calibration function by approximating the empirical cumulative distribution using a differentiable function via splines.

Algorithm 2 Confidence based Temperature Scaling (CBT) - Inference

input: A data point x with network outputs logits $z_1, ..., z_k$ and a division of bins B obtained from training step.
for $s = 1, ..., S$ **do**
  Compute the confidence: $c = \max_i \frac{\exp(z_i)}{\sum_j \exp(z_j)}$.
  Find l s.t. c is in the bin defined by B_{sl}.
  $z_i \leftarrow z_i/T_{sl}, \qquad i = 1, ..., k$
end for
output: The calibrated prediction is:

$$p(y = i|x) = \frac{\exp(z_i)}{\sum_j \exp(z_j)}, \qquad i = 1, ..., k$$



(a) (b)

Fig. 1: (a) Optimal temperature for each bin in four iterations of CBT for Imagenet trained with ResNet152. (b) ECE values for each bin before and after temperature scaling for Imagenet trained with ResNet152.

Note that this calibration method can change the accuracy. Ji et al. [14] extended TS to a bin-wise setting, denoted by BTS, by setting separate temperatures for various confidence subsets. The main difference between BTS and our approach is the learning procedure. BTS is trained by maximizing the log-likelihood function. We proposed an iterative scheme that directly minimizes the gap between the confidence and the accuracy at each bin. In the next section we show that CBT consistently yields better calibration results than BTS on a large set of tasks.

IV. Experimental Results

We first illustrate the CBT algorithm on the ImageNet dataset [15] with network architectures ResNet152 [16] and DenseNet161 [17] trained with a cross-entropy loss. Fig. 1a presents the CBT temperature in each bin that minimizes the adaECE score for each one of the four iterations performed by the algorithm. As we go up in the bins' range, we can see an increase in the optimal temperature per bin. This is because samples in higher bins are more over-confident than samples in lower bins, so that higher temperatures for samples in these high over-confident bins should bring the average confidence closer to the average accuracy. However, a single temperature does not take this difference in over-confidence through bins into account. We can also see that the optimal

Dataset	Model	Cross-Entropy			Brier Loss			MMCE			LS-0.05		
		Pre T	TS	CBT	Pre T	TS	CBT	Pre T	TS	CBT	Pre T	TS	CBT
CIFAR-100	ResNet-50	17.52	3.42	**1.25 (5)**	6.52	3.64	**1.08 (2)**	15.32	2.38	**1.25 (5)**	7.81	4.01	**1.30 (3)**
	ResNet-110	19.05	4.43	**1.95 (3)**	7.88	4.65	**2.19 (2)**	19.14	3.86	**2.11 (3)**	11.02	5.89	**1.82 (2)**
	Wide-ResNet-26-10	15.33	2.88	**1.43 (5)**	4.31	2.70	**1.37 (4)**	13.17	4.37	**1.40 (3)**	4.84	4.84	**0.93 (2)**
	DenseNet-121	20.98	4.27	**2.20 (3)**	5.17	2.29	**1.50 (5)**	19.13	3.06	**1.79 (2)**	12.89	7.52	**1.52 (5)**
CIFAR-10	ResNet-50	4.35	1.35	**0.75 (2)**	1.82	1.08	**1.02 (4)**	4.56	1.19	**0.63 (2)**	2.96	1.67	**1.05 (2)**
	ResNet-110	4.41	1.09	**0.74 (2)**	2.56	1.25	**0.81 (2)**	5.08	1.42	**0.26 (4)**	2.09	2.09	**0.65 (3)**
	Wide-ResNet-26-10	3.23	0.92	**0.80 (2)**	1.25	1.25	**0.88 (2)**	3.29	0.86	**0.69 (5)**	4.26	1.84	**0.50 (2)**
	DenseNet-121	4.52	1.31	**0.39 (2)**	1.53	1.53	**0.75 (2)**	5.10	1.61	**0.73 (2)**	1.88	1.82	**0.78 (2)**
Tiny-ImageNet	ResNet-50	15.32	5.48	**1.25 (3)**	4.44	4.13	**0.99 (4)**	13.01	5.55	**1.19 (3)**	15.23	6.51	**1.30 (3)**

TABLE I: ECE (%) computed for different approaches for pre-temperature scaling, post-single temperature scaling and post confidence-based temperature scaling (with the number of iterations in brackets). $T \approx 1$ indicates an innately calibrated model.

Dataset	Model	Uncalibrated	TS	Vector Scaling	MS-ODIR	Dir-ODIR	Spline	BTS	CBT
CIFAR-100	ResNet-110	18.480	2.428	2.722	3.011	2.806	1.868	1.907	**1.818**
	ResNet-110-SD	15.861	1.335	2.067	2.277	2.046	1.766	1.373	**1.299**
	Wide-ResNet-32	18.784	1.667	1.785	2.870	2.128	1.672	1.796	**1.317**
	DenseNet-40	21.159	**1.255**	1.598	2.855	1.410	2.114	1.336	1.307
	Lenet-5	12.117	1.535	1.350	1.696	2.159	**1.029**	1.659	1.249
CIFAR-10	ResNet-110	4.750	1.224	1.092	1.276	1.240	1.011	1.224	**0.982**
	ResNet-110-SD	4.135	0.777	0.752	**0.684**	0.859	0.992	1.020	0.867
	Wide-ResNet-32	4.512	0.905	**0.852**	0.941	0.965	1.003	1.064	1.049
	DenseNet-40	5.507	1.006	1.207	1.250	1.268	1.389	0.957	**0.904**
	Lenet-5	5.188	1.999	1.462	1.504	**1.300**	1.333	1.865	1.614
ImageNet	DenseNet-161	5.720	2.059	2.637	4.337	3.989	**0.798**	1.224	0.845
	ResNet-152	6.545	2.166	2.641	5.377	4.556	**0.913**	1.165	0.935
SVHN	ResNet-152-SD	0.877	0.675	0.630	0.646	0.651	0.832	**0.535**	0.537

TABLE II: ECE for top-1 predictions (in %) using 25 bins (with the lowest in bold and the second lowest underlined) on various image classification datasets and models with different calibration methods.

temperatures given to each bin are reduced through iterations until all temperatures are converged to 1. Fig. 1b shows the ECE score in each bin before and after calibration with TS and CBT for ResNet152 trained on a test set from ImageNet. The ECE in bin i is defined by: $\mathrm{ECE_i} = \frac{|B_i|}{n} |A_i - C_i|$ such that the total ECE score is the sum of the ECE of all bins. The results show that the ECE in each bin achieved by CBT is almost always lower or equal to the ECE achieved by TS.

We implemented the CBT method on various image classification tasks to test the algorithm's performance. The experiment setup followed the setup of [13] and included several pre-trained deep neural networks which are available online [1], trained on the following image classification datasets:

1) **CIFAR-10** [15]: This dataset has 60,000 color images of size 32×32, divided equally into 10 classes. We used a train/validation/test split of 45,000/5,000/10,000 images.
2) **CIFAR-100** [15]: This dataset has 60,000 color images of size 32×32, divided equally into 100 classes. We again used a train/validation/test split of 45,000/5,000/10,000 images.
3) **Tiny-ImageNet** [18]: Tiny-ImageNet is a subset of ImageNet with 64×64 dimensional images, 200 classes and 500 images per class in the training set and 50 images per class in the validation set. The image dimensions of Tiny-ImageNet are twice those of the CIFAR-10/100 images.

Table I compares the ECE% (computed using 15 bins) obtained by evaluating the test set. Although adaECE was used as the objective function in our algorithm, ECE is still the standard way to report calibration results, so we used it to compare our calibration results with previous studies. The results are divided into ECE before calibration, after scaling by a single temperature (TS) and after our Confidence based Temperature Scaling (CBT). The optimal TS was achieved by a greedy algorithm to minimize the ECE calibration score [13]. The CBT was trained on 16 bins over a validation set. Along with the cross-entropy loss, we tested our results on three other models which were trained on different loss functions:

1) **Brier loss** [19]: The squared error between the predicted softmax vector and the one-hot ground truth encoding.
2) **MMCE** (Maximum Mean Calibration Error) [20]: A continuous and differentiable proxy for calibration error that is normally used as a regulariser alongside cross-entropy.
3) **Label smoothing** (LS) [21]: Given a one-hot ground-

[1] https://github.com/torrvision/focal_calibration

truth distribution $\mathbf{q}$ and a smoothing factor α (hyper-parameter), the smoothed vector $\mathbf{s}$ is obtained as $\mathbf{s}_i = (1-\alpha)\mathbf{q}_i + \alpha(1-\mathbf{q}_i)/(k-1)$, where $\mathbf{s}_i$ and $\mathbf{q}_i$ denote the i^{th} elements of $\mathbf{s}$ and $\mathbf{q}$ respectively, and k is the number of classes. Instead of $\mathbf{q}$, $\mathbf{s}$ is treated as the ground truth. The reported results were obtained from LS-0.05 with $\alpha = 0.05$, which was found to achieve the best performance [13].

The comparative calibration results are presented in Table I. The number of iterations of the CBT algorithm for each comparison appears in brackets. As can be seen, the ECE score after CBT calibration was lower than the ECE after TS in all cases.

In another set of experiments, we followed the setup in [10]. In addition to CIFAR-10 and CIFAR-100, we evaluated our CBT method on the SVHN dataset [22] and ImageNet [18]. Pre-trained network logits are available online [2]. The CBT was compared to TS, vector scaling, two variants of matrix scaling [7], BTS [14] and Spline fitting [10]. As shown in Table II, CBT achieved the best or second best results in most cases.

V. Conclusion

Calibrated confidence estimates of predictions are critical to increasing our trust in the performance of neural networks. As interest grows in deploying neural networks in real work decision making systems, the predictable behavior of the model will be a necessity especially for critical tasks as automatic navigation and medical diagnosis. In this work, we introduced a simple and effective calibration method based the prediction confidence. Most calibration methods are trained by optimizing the cross entropy score. CBT function learning can be done by explicitly optimizing the adaECE measure. We compared our CBT method to various state-of-the-art methods and showed that it yielded the lowest calibration error in many of our experiments. CBT is very easy to train and there is no need to tune any hyper-parameter. we believe that it can be used in place of the standard temperature scaling method.

Acknowledgment

The research was partially supported by the Israeli Ministry of Science & Technology.

References

[1] Cynthia S Crowson, Elizabeth J Atkinson, and Terry M Therneau, "Assessing calibration of prognostic risk scores," *Statistical Methods in Medical Research*, vol. 25, no. 4, pp. 1692–1706, 2016.

[2] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mane, "Concrete problems in AI safety," *arXiv preprint arXiv:1606.06565*, 2016.

[3] Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf, "Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate the problem," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[4] John Platt et al., "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in Large Margin Classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[5] Bianca Zadrozny and Charles Elkan, "Transforming classifier scores into accurate multiclass probability estimates," in *International Conference on Knowledge Discovery and Data Mining (KDD)*, 2002.

[6] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger, "On calibration of modern neural networks," in *International Conference on Machine Learning (ICML)*, 2017.

[7] Meelis Kull, Miquel Perello Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song, and Peter Flach, "Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with Dirichlet calibration," in *Advances in Neural Information Processing Systems (NeurIPs)*, 2019.

[8] Lior Frenkel and Jacob Goldberger, "Network calibration by class-based temperature scaling," in *The European Signal Processing Conference (EUSIPCO)*, 2021.

[9] Lior Frenkel and Jacob Goldberger, "Calibration of medical imaging classification systems with weight scaling," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2022.

[10] Kartik Gupta, Amir Rahimi, Thalaiyasingam Ajanthan, Thomas Mensink, Cristian Sminchisescu, and Richard Hartley, "Calibration of neural networks using splines," in *International Conference on Learning Representations (ICLR)*, 2021.

[11] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht, "Obtaining well calibrated probabilities using bayesian binning," in *AAAI Conference on Artificial Intelligence*, 2015.

[12] Khanh Nguyen and Brendan O'Connor, "Posterior calibration and exploratory analysis for natural language processing models," *arXiv preprint arXiv:1508.05154*, 2015.

[13] Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip HS Torr, and Puneet K Dokania, "Calibrating deep neural networks using focal loss," in *Advances in Neural Information Processing Systems (NeurIPs)*, 2020.

[14] Byeongmoon Ji, Hyemin Jung, Jihyeun Yoon, Kyungyul Kim, and Younghak Shin, "Bin-wise temperature scaling (BTS): Improvement in confidence calibration performance through simple scaling techniques," in *IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019.

[15] Alex Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., Department of Computer Science, University of Toronto, 2009.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[17] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger, "Densely connected convolutional networks," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[18] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[19] Glenn W Brier, "Verification of forecasts expressed in terms of probability," *Monthly weather review*, vol. 78, no. 1, pp. 1–3, 1950.

[20] Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain, "Trainable calibration measures for neural networks from kernel mean embeddings," in *International Conference on Machine Learning*, 2018.

[21] Rafael Müller, Simon Kornblith, and Geoffrey Hinton, "When does label smoothing help?," *arXiv preprint arXiv:1906.02629*, 2019.

[22] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng, "Reading digits in natural images with unsupervised feature learning," in *NIPS Workshop*, 2011.

[2] https://github.com/markus93/NN_calibration