Classifier Ensemble for Efficient Uncertainty Calibration of Deep Neural Networks for Image Classification

Michael Schulze, Nikolas Ebert, Laurenz Reichardt and Oliver Wasenmüller
Mannheim University of Applied Sciences, Germany
{*m.schulze, n.ebert, l.reichardt, o.wasenmueller*}*@hs-mannheim.de*

Keywords: Calibration, Uncertainty, Image Classification, SafeAI, XAI.

Abstract: This paper investigates novel classifier ensemble techniques for uncertainty calibration applied to various deep neural networks for image classification. We evaluate both accuracy and calibration metrics, focusing on Expected Calibration Error (ECE) and Maximum Calibration Error (MCE). Our work compares different methods for building simple yet efficient classifier ensembles, including majority voting and several metamodel-based approaches. Our evaluation reveals that while state-of-the-art deep neural networks for image classification achieve high accuracy on standard datasets, they frequently suffer from significant calibration errors. Basic ensemble techniques like majority voting provide modest improvements, while metamodel-based ensembles consistently reduce ECE and MCE across all architectures. Notably, the largest of our compared metamodels demonstrate the most substantial calibration improvements, with minimal impact on accuracy. Moreover, classifier ensembles with metamodels outperform traditional model ensembles in calibration performance, while requiring significantly fewer parameters. In comparison to traditional post-hoc calibration methods, our approach removes the need for a separate calibration dataset. These findings underscore the potential of our proposed metamodel-based classifier ensembles as an efficient and effective approach to improving model calibration, thereby contributing to more reliable deep learning systems.

1 INTRODUCTION

Machine learning models, particularly deep neural networks, are increasingly applied in safety critical areas such as autonomous driving (Ebert et al., 2022; Reichardt et al., 2023) and medical image analysis (Ebert et al., 2023), where incorrect decisions can have serious consequences. In these settings, achieving high accuracy and robustness (Oehri et al., 2024; Kendall and Gal, 2017) is crucial, but models must also provide reliable uncertainty estimates to assess whether their predictions can be trusted (Jiang et al., 2018). Calibration addresses this need by aligning predicted probabilities with the true likelihood of predictions being correct (Bröcker, 2009). However many machine learning models (Niculescu-Mizil and Caruana, 2005), especially deep neural networks (Guo et al., 2017), are poorly calibrated and tend to produce overconfident predictions, even when they are wrong.

Post-hoc calibration methods, which adjust the prediction scores of a trained neural network using a separate calibration dataset, are widely used to improve uncertainty estimates. Examples include Platt scaling (Platt et al., 1999), histogram binning (Zadrozny and Elkan, 2001), isotonic regression (Zadrozny and Elkan, 2002) and temperature scaling

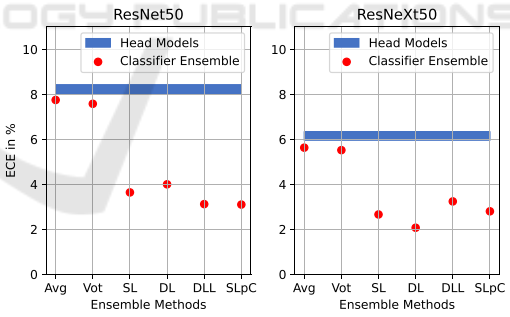

Figure 1: Expected Calibration Error (ECE) of ResNet50 (He et al., 2016) (left) and ResNeXt50 (Xie et al., 2017) (right) on CIFAR-100 (Krizhevsky et al., 2009). Each model was trained with five classifier heads initialized with different random seeds but using the same backbone. The blue area represents the ECE range for the uncalibrated classifiers. Each red dot corresponds to the ECE value achieved using different ensemble techniques. The use of metamodels (SL, DL, DLL, SLpC) significantly improves the calibration performance and reduces the ECE compared to the uncalibrated baseline.

Schulze, M., Ebert, N., Reichardt, L. and Wasenmüller, O.
Classifier Ensemble for Efficient Uncertainty Calibration of Deep Neural Networks for Image Classification.
DOI: 10.5220/0013129000003912

In *Proceedings of the 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2025) - Volume 2: VISAPP*, pages 316-323
ISBN: 978-989-758-728-3; ISSN: 2184-4321

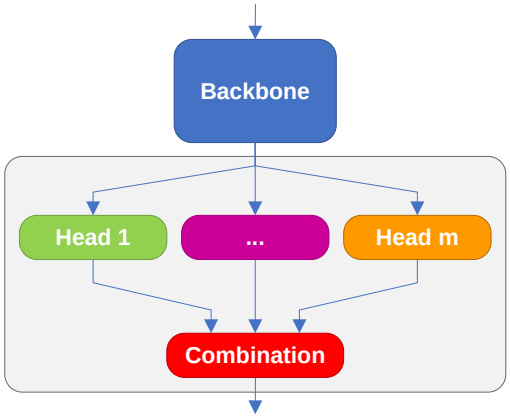

Figure 2: The principle of the classifier ensemble involves a single backbone that feeds multiple classifiers (heads). The combination method can be freely selected.

(Guo et al., 2017). Parametric methods like temperature scaling rescale the output logits of a neural network for classification using learned parameters from a calibration set. However, in many real-world scenarios with limited data, a dedicated calibration set is not available. Although non-parametric methods, such as isotonic regression, offer greater flexibility, they can reduce model accuracy after calibration. Similar to their parametric counterparts, these methods also require a dedicated calibration set.

In contrast to post-hoc calibration, ab-initio methods (Lakshminarayanan et al., 2017; Kumar et al., 2018) aim to train models that are well-calibrated from the start, incorporating uncertainty directly during training. Furthermore, deep ensembles (Lakshminarayanan et al., 2017; Wenzel et al., 2020) combine multiple models trained on the same dataset through majority voting or averaging, which enhances accuracy and reduces uncertainty. However, a disadvantage of this approach is the high computational cost associated with training several independent models. In contrast, Monte Carlo dropout (Gal and Ghahramani, 2016) follows a similar strategy by applying dropout during training and inference to randomly deactivate individual neurons, thereby creating an ensemble of models. However, this method requires repeated inference, resulting in lower accuracy and higher uncertainty compared to deep ensembles.

Thus, we propose a novel approach based on classifier ensemble (see Figure 2), which effectively combines transfer learning with ensemble methods for efficient uncertainty calibration. In contrast to traditional ensemble techniques, where multiple full-scale networks are trained separately and their predictions are combined, our method focuses on training multiple lightweight classifiers on-top of a shared backbone and utilizing their predictions collaboratively. This technique stands out by eliminating the need for an additional calibration dataset and significantly reducing computational overhead during both training and inference. By combining the strengths of transfer learning and ensemble methods, our classifier ensemble significantly reduces uncertainty while maintaining computational efficiency. Furthermore, we have proven the effectiveness of our approach in numerous analyses of different neural networks (see Figure 1) on CIFAR-100 (Krizhevsky et al., 2009) and Tiny-ImageNet (Le and Yang, 2015) benchmarks from the field of image classification.

2 RELATED WORK

2.1 Calibration Methods

During the past decade, several post-hoc methods for calibrating network outputs have been developed. Histogram binning (Zadrozny and Elkan, 2001) assigns predictions to fixed intervals and learns a calibrated score for each by minimizing the squared error loss on a calibration dataset. During inference, uncalibrated scores are replaced by these calibrated scores. Isotonic regression (Zadrozny and Elkan, 2002) generalizes this method by dynamically learning intervals from the calibration dataset, adjusting both boundaries and calibrated scores to produce a piecewise constant function. Logistic regression, or Platt scaling (Platt et al., 1999), uses uncalibrated scores as features for a regression model trained to minimize negative log-likelihood, which then calibrates the scores during prediction. Similar to Platt scaling, temperature scaling (TS) (Guo et al., 2017) uses a single scalar parameter to adjust the prediction scores based on a calibration dataset, preserving model accuracy. An extension of TS called Ensemble Temperature Scaling (Zhang et al., 2020) learns a mapping of three scaling factors instead of a single factor, resulting in a weighted combination of three TS. Parameterized Temperature Scaling (Tomani et al., 2022) extends TS by using a small neural network to learn multiple parameters for different classes instead of a single parameter for all classes.

In contrast to the mentioned post-hoc methods, deep ensembles (Lakshminarayanan et al., 2017) involve the training of multiple models on the same dataset and combining them through majority voting or averaging, enhancing accuracy and reducing uncertainty. However, this requires significant computational resources. Monte Carlo dropout (Gal and Ghahramani, 2016) combines predictions from different subnetworks by applying dropout during training

and inference. This method generates an ensemble by performing multiple inferences with different active neurons, but it generally results in lower accuracy and higher uncertainty compared to deep ensembles.

2.2 Model Ensemble

Model ensemble techniques combine multiple individual models to enhance predictive performance. The core idea is that different models may possess unique strengths and weaknesses, which can be maximized and balanced through aggregation, leading to improved overall accuracy. In a voting ensemble (Goodfellow, 2016), several models are trained on the same dataset, and their predictions are aggregated through majority voting. This approach effectively utilizes the collective intelligence of the models and is suitable when individual models exhibit similar performance levels but make distinct errors. Deep ensemble (Lakshminarayanan et al., 2017; Wenzel et al., 2020) methods involve independent training multiple neural networks, each with its own weights and parameters. Their predictions are aggregated via averaging or majority voting, capturing diverse aspects of the data and yielding more robust predictions. Bagging ensembles (Raschka et al., 2022) use bootstrapping to create multiple subsets from the training data by drawing random samples with replacement. Models are trained on these subsets, and their predictions are combined through averaging or voting. This method reduces model variance and enhances robustness against overfitting. In boosting ensembles (Raschka et al., 2022), several weak models are trained sequentially, and their predictions are combined through weighted averaging. The weights are adjusted to emphasize samples that previous models misclassified, addressing issues of high bias or underfitting. Stacking ensembles (Raschka et al., 2022) involve training multiple models on the same dataset and using their predictions as features for a meta-model. The meta-model is trained on the predictions of the base models with true labels as targets, allowing for the integration of diverse strengths and weaknesses to enhance predictive accuracy.

Unlike the ensemble methods mentioned above, we do not rely on retaining multiple full-scale networks. Instead, we retrain multiple lightweight classifiers (each comprising less than 1% of the entire model) with a strong shared backbone and utilize their predictions collaboratively. This approach effectively reduces model uncertainty and yields a well-calibrated model without the need for a dedicated calibration dataset, which is typically required by other post-hoc methods.

3 METHOD

3.1 Preliminaries

Let $X \in \mathbb{R}^D$ represent the D-dimensional input and $Y \in \{1,\dots,C\}$ represent the class labels for a classification task with C possible classes. The joint distribution of X and Y is denoted by $\pi(X,Y) = \pi(Y|X)\pi(X)$. The dataset $\mathcal{D}$ consists of N independent and identically distributed (i.i.d.) samples $\mathcal{D} = \{(X_n, Y_n)\}_{n=1}^N$, drawn from this distribution. A neural network classifier $h(X)$ outputs a predicted class $\hat{Y}$ and a corresponding logit vector $\hat{Z}$. The logits $\hat{Z}$ are then converted into a confidence score $\hat{P}$ for the predicted class $\hat{Y}$ using the softmax function σ_{SM}, where $\hat{P} = \max_c \sigma_{\text{SM}}(\hat{Z})_c$.

Uncertainty Calibration. Perfect calibration is defined as the condition where the accuracy of predictions aligns with the confidence levels across all possible confidence values (Guo et al., 2017), mathematically represented as

$$\mathbb{P}(\hat{Y} = Y|\hat{P} = p) = p \quad \text{for every } p \in [0,1]. \tag{1}$$

In contrast, miss-calibration refers to the expected discrepancy between confidence and accuracy, which can be expressed as:

$$\mathbb{E}_{\hat{P}}\left[|\mathbb{P}(\hat{Y} = Y|\hat{P} = p) - p|\right]. \tag{2}$$

Measuring Uncertainty. The Expected Calibration Error (ECE) serves as a widely used scalar metric for assessing miss-calibration (Naeini et al., 2015). It approximates Equation (2) based on the predictions $\hat{Y}$, the confidence scores $\hat{P}$ and the ground truth labels Y of a finite number of N samples. The ECE is computed by dividing the confidence scores into M equal bins B_m, calculating the average confidence (conf) and classification accuracy (acc) for each bin, and then summarizing the resulting differences. The formula for ECE is:

$$ECE^d = \sum_{m=1}^{M} \frac{|B_m|}{N} \|\text{acc}(B_m) - \text{conf}(B_m)\|_d, \tag{3}$$

where d is typically set to 1 for the L1-norm.

In addition to ECE, we use the Maximum Calibration Error (MCE), which captures the largest discrepancy among the intervals used to calculate the ECE, providing another measure of calibration performance. The formula for MCE is:

$$MCE = \max_{m \in \{1,\dots,M\}} |\text{acc}(B_m) - \text{conf}(B_m)| \tag{4}$$

3.2 Classifier Ensemble for Uncertainty Calibration

A commonly used method for calibrating model outputs is deep ensembles (Lakshminarayanan et al., 2017) (see Section 2), where multiple models are trained on the same data and combined during inference. However, this approach requires substantial time and computational resources, as it necessitates training several models from scratch and performing multiple inferences.

In contrast, our novel classifier ensemble approach divides the model into a backbone and a head (classifier), with the backbone responsible for computing features and being significantly larger than the head, which maps these features to target classes. Notably, we only re-train the heads while keeping the pre-trained backbone frozen. The individual classifiers are subsequently combined using model ensemble techniques, as illustrated in Figure 2.

3.2.1 Train Strategies

The training of a classifier ensemble is conducted in multiple steps. Initially, a base model is created and trained on the training dataset, after which it is saved. Subsequently, a new base model is created, and the weights from the previously trained model are loaded. Following the principles of transfer learning, the weights are frozen, and only the head is newly constructed and then trained again on the training data. This process is repeated as many times as necessary to form the desired number of heads for the classifier ensemble, as illustrated in Figure 3.

Such a separate training approach offers several advantages. It allows the use of different head architectures, such as varying the number of layers or incorporating dropout. Additionally, diverse data augmentation strategies or different subsets of the dataset can be applied during each head's training, akin to the bagging ensemble method described in Section 2.

3.2.2 Ensemble Methods

In the final step of the classifier ensemble, the different heads must be combined, as shown in Figure 2. Our proposed methods for combining these heads include averaging, voting, and the use of metamodels. Averaging involves summing the individual outputs of the heads and dividing the total by the number of heads. In voting, a majority decision is made by selecting the most frequent predicted values across the heads.

Alternatively, metamodels can be used, where the classifiers are combined using additional learnable

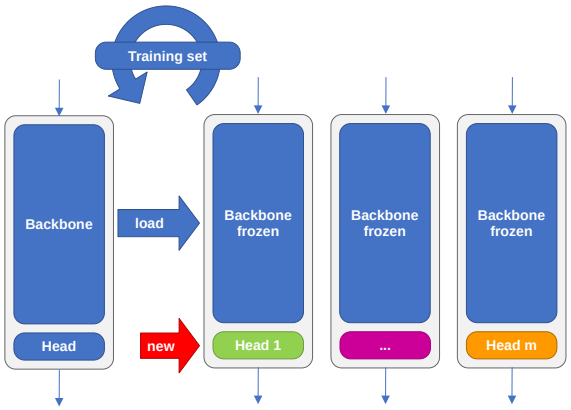

Figure 3: Training Process of our classifier ensemble.

parameters. One approach involves concatenating the outputs of all m heads and applying a fully connected layer, where the input consists of the combined predictions from the heads, yielding $m \cdot C$ input features, while the output remains the original C classes.

The architecture can be further extended with additional hidden layers, nonlinearities, or dropout, as long as the structure supports $m \cdot C$ input and C output features. Another variant is to link the head outputs class-wise with a fully connected layer. In this case, a separate fully connected layer is used for each class, with each layer having m input features and a single output, leading to a total of C fully connected layers, one for each class.

In the studies conducted in Section 4.2, we performed a thorough comparison of all methods. However, no single approach consistently outperformed the others across different networks and datasets. Nevertheless, all methods demonstrated a significant improvement compared to the uncalibrated baseline.

4 EVALUATION

Our evaluation is divided into three sections. Section 4.1 first provides a detailed overview of the data and training settings used for all our experiments. Next, in section 4.2.1, we conduct an extensive study with CIFAR-100 (Krizhevsky et al., 2009). Finally, in Section 4.2.2, we use Tiny ImageNet (Le and Yang, 2015) for further evaluations.

4.1 Datasets, Training, and Ensemble Configuration

Datasets. To evaluate our novel classifier ensemble, we utilize the CIFAR-100 (Krizhevsky et al., 2009) dataset. CIFAR-100 consists of 50,000 training im-

Table 1: Comparison of accuracy, ECE and MCE in percent of uncalibrated heads and their combination to the classifier ensemble with ResNet (He et al., 2016) variants on CIFAR-100 (Krizhevsky et al., 2009).

	Model	ResNet18			ResNet34			ResNet50			ResNet101		
		Acc.	ECE	MCE	Acc.	ECE	MCE	Acc.	ECE	MCE	Acc.	ECE	MCE
Baseline	Head 1	75.08	4.41	27.47	76.74	5.60	15.66	77.21	8.22	25.01	78.06	8.76	21.06
	Head 2	74.95	4.69	16.26	76.76	5.74	15.00	77.15	8.30	23.80	78.06	8.83	23.14
	Head 3	75.07	4.65	27.25	76.86	5.56	27.76	77.45	8.06	23.73	78.12	8.68	22.40
	Head 4	75.22	4.40	24.07	76.89	5.92	17.64	77.39	8.04	25.96	78.20	8.64	23.26
	Head 5	75.25	4.46	14.11	76.71	5.75	15.20	77.06	8.43	25.51	78.05	8.76	23.11
Classifier Ensemble (ours)	Avg.	75.06	4.43	10.66	76.83	5.75	19.12	77.28	7.75	25.13	78.07	8.43	20.17
	Vot.	74.96	4.25	11.97	76.81	5.12	14.27	77.29	7.58	24.97	77.99	8.31	22.85
	SL	74.39	**2.59**	**8.35**	76.45	3.48	11.37	77.17	3.64	9.62	77.30	2.81	8.61
	DL	74.29	2.93	8.44	76.37	**3.44**	9.10	76.89	4.00	10.52	77.44	**2.71**	**7.57**
	DLL	74.73	3.51	10.22	76.75	3.83	11.12	77.32	3.12	11.10	77.94	3.39	10.66
	SLpC	74.99	4.11	11.71	76.73	4.01	**7.73**	77.11	**3.10**	**9.35**	78.22	3.70	9.12

ages and 10,000 test images with 100 classes. In addition to the experiments on the CIFAR-100 dataset, we also conducted experiments on the Tiny ImageNet (Le and Yang, 2015) dataset consisting of 100,000 training images and 5,000 test images of 200 classes.

Base Training. As outlined in Section 3, the first step in training our classifier ensemble is the standard pretraining of a base model (backbone + head), which serves as the foundation for subsequent steps. For this work, a Stochastic Gradient Descent (SGD) optimizer with momentum of 0.9 and weight decay of 5e-04 is used. During the 200 training epochs, the basic learning rate of 0.1 is gradually adjusted by a factor of 0.2 using a multi-stage scheduler. A batch size of 128 and a basic data augmentation strategy, including random cropping, padding, horizontal flipping and random rotation is used.

Training Heads. The individual heads are created by loading the base model. As described in Section 3, the backbone weights are frozen and only the classifier (head) is reinitialized with new random seeds. The new classifier is then trained using an SGD optimizer with an initial learning rate of 0.1. The learning rate for training the heads is adjusted using a Plateau-Min-Scheduler, which monitors the validation loss. If the loss does not improve within a specified number of epochs, the learning rate is reduced by multiplying it by a factor of 0.5. Additionally, early stopping with a patience of 15 epochs is applied, terminating the training if no further improvements are observed.

All heads used in this work consist of a single fully connected layer. Each base model is trained with five distinct heads, which are saved and later combined into an ensemble. Since each head contains only a few parameters, the training process is very fast.

Ensemble Configuration. For classifier ensemble without a metamodel, two combination methods were explored: mean averaging and majority voting. When using a metamodel to combine the heads, additional training is required. Four metamodels were implemented and analyzed: Single-Layer (SL), Double-Layer (DL), Double-Layer-Large (DLL), and Single-Layer-per-Class (SLpC).

The SL metamodel combines the outputs of the heads through a single fully connected layer. The DL metamodel adds a second layer with ReLU activation and dropout, where the first layer reduces the number of neurons. In contrast, the DLL metamodel doubles the number of neurons in the first layer compared to the DL model. The SLpC metamodel takes a different approach, using a dedicated fully connected layer for each class, where the concatenated head outputs are connected with class-specific layers.

As the metamodels introduce additional parameters, they require training on the training dataset. This training is performed over 20 epochs using an SGD optimizer with an initial learning rate of 0.0002, along with a Plateau-Min-Scheduler to adjust the learning rate. After training, the metamodel with the lowest validation loss is selected for deployment.

4.2 Results

Eight different base models were developed and trained on the CIFAR-100 dataset, with five distinct heads trained for each base model. The results for various ResNet models (He et al., 2016) are displayed in Table 1, while more advanced models such as DenseNets (Huang et al., 2017), ResNeXt (Xie et al., 2017) and GoogLeNet (Szegedy et al., 2015) are shown in Table 2. All heads were combined with our classifier ensembles using different methods, including mean averaging, majority voting and different metamodels called Single-Layer (SL), Double-Layer

Table 2: Comparison of accuracy, ECE and MCE in percent of uncalibrated heads and their combination to the classifier ensemble with ResNeXt (Xie et al., 2017), DenseNet (Huang et al., 2017) and GoogLeNet (Szegedy et al., 2015) on CIFAR-100 (Krizhevsky et al., 2009).

	Model	ResNeXt50			DenseNet121			DenseNet169			GoogLeNet		
		Acc.	ECE	MCE	Acc.	ECE	MCE	Acc.	ECE	MCE	Acc.	ECE	MCE
Baseline	Head 1	77.15	6.05	15.06	77.55	4.74	10.20	78.43	4.00	10.28	75.66	6.93	16.65
	Head 2	77.06	5.99	12.28	77.45	4.75	12.02	78.56	3.92	9.30	75.84	7.02	19.36
	Head 3	76.86	6.14	13.56	77.55	4.97	13.04	78.57	4.08	9.64	75.74	6.96	18.68
	Head 4	76.91	6.34	16.08	77.43	4.70	10.49	78.42	4.05	9.02	75.71	7.04	18.88
	Head 5	77.27	5.96	15.09	77.49	4.49	11.75	78.51	4.04	8.60	75.66	7.07	19.15
Classifier Ensemble (ours)	Avg.	77.13	5.63	13.26	77.56	4.27	9.77	78.44	3.68	8.85	75.74	6.72	17.35
	Vot.	76.99	5.52	13.55	77.52	4.29	10.41	78.53	3.61	8.21	75.77	6.37	15.74
	SL	76.45	2.66	7.53	77.27	2.95	11.51	78.24	2.46	**8.05**	75.09	**3.55**	**7.94**
	DL	76.27	**2.07**	**5.97**	77.18	**2.69**	19.08	77.66	2.23	8.88	75.36	4.44	8.68
	DLL	76.57	3.24	9.57	77.51	2.79	11.58	78.09	**2.19**	8.86	75.43	4.84	10.36
	SLpC	77.22	2.80	8.99	77.39	2.95	**7.86**	78.56	3.09	10.88	75.73	4.91	10.32

(DL), Double-Layer Large (DLL) and Single-Layer per Class (SLpC). The tables summarize the results for architectures, highlighting accuracy (Acc.), Expected Calibration Error (ECE), and Maximum Calibration Error (MCE). Individual heads are presented as baseline, where each head paired with the backbone represents a different variation due to the unique random seed applied. The tables present the results for mean averaging and majority voting, followed by the outcomes of the trained metamodels.

4.2.1 Results for CIFAR-100

The results across different ResNet variants on CIFAR-100 (see Table 1) reveal a consistent trend. For individual heads, the accuracy remains fairly consistent across all ResNet models, with ResNet101 achieving the highest accuracy (78.22%). However, this also corresponds with higher ECE and MCE values, indicating issues with model calibration. Mean averaging as an ensemble method yields slight improvements in accuracy and moderate reductions in ECE, particularly in ResNet50 and ResNet101, though the calibration improvements are not substantial. Majority voting offers better calibration than mean averaging, resulting in lower ECE and MCE, but accuracy is marginally lower compared to mean averaging.

The metamodels, particularly the SL and DL approaches, show the most significant reductions in ECE and MCE across all ResNet variants, especially for ResNet101. Although these methods slightly decrease accuracy, the calibration improvement is substantial. The DLL and SLpC models also exhibit strong calibration performance, with SLpC performing notably well in terms of ECE for ResNet50.

Table 3: Comparison of accuracy, ECE and MCE in percent of classic model ensemble with ResNet18 (He et al., 2016) on CIFAR-100 (Krizhevsky et al., 2009).

Model	ResNet18			
	Acc.	ECE	MCE	Params
Model 1	74.89	5.96	27.61	11.22 M
Model 2	75.03	6.26	17.23	11.22 M
Model 3	74.20	9.72	22.02	11.22 M
Model 4	73.08	11.30	25.82	11.22 M
Model 5	71.66	7.98	18.26	11.22 M
Ensemble	75.85	6.91	18.32	56.10 M

In line with the findings in Table 1 for ResNet variants, the more advanced models presented in Table 2 display a comparable pattern. While individual heads achieve competitive accuracy, they consistently exhibit higher calibration errors, with GoogLeNet showing particularly elevated ECE and MCE values.

Mean averaging and majority voting marginally reduce calibration errors across all models, particularly in DenseNet and ResNeXt. However, these reductions are not as significant as those seen with the use of metamodels. The trained metamodels, particularly the DL and SL approaches, yield substantial improvements in calibration metrics. The DL metamodel delivers the lowest ECE and MCE values for ResNeXt and DenseNet, with notable performance in reducing calibration errors while maintaining accuracy. The SLpC model also demonstrates good calibration, especially for DenseNet169, which achieves a balance between low ECE and high accuracy.

Compared to the previous table for ResNet models, these results further highlight the effectiveness of classifier ensembles with metamodels in reducing calibration errors, with DL consistently performing well

across architectures. However, the trade-off between accuracy and calibration remains present, as seen with the slight dip in accuracy in some metamodel approaches. Overall, classifier ensembles incorporating metamodels continue to significantly enhance model calibration across various architectures, building on the trends observed with the ResNet variants.

As reference, a traditional horizontal model ensemble using ResNet18 was also evaluated. The results are shown in Table 3. This approach aggregates models from different checkpoints during training and combines their outputs using mean averaging. When comparing the ResNet18 results from Table 3 with those in Table 1, some distinct trends can be observed.

The accuracy of the classical model ensemble in Table 3 reaches 75.85%, which is slightly higher than the individual heads, where the highest accuracy is 75.25%. However, the calibration errors, particularly the ECE and MCE, remain relatively high in the classical ensemble, with 6.91% and 18.32%, respectively. In contrast, our classifier ensembles using metamodels in Table 1 consistently achieve much lower calibration errors, with the SL and DL approaches reducing the ECE to 2.59% and 2.93%, respectively, while also minimizing the MCE.

Another notable difference is the parameter count. The classical ensemble significantly increases the number of parameters to 56.1 M, whereas the classifier ensembles with metamodels only introduce minor increases in parameter count (approximately 3%). Thus, while the classical ensemble offers slightly improved accuracy, it does so at the cost of significantly higher calibration errors and a substantial increase in model size compared to the classifier ensemble methods.

4.2.2 Results for Tiny ImageNet

Table 4 presents the results of ResNet18 trained on the Tiny ImageNet dataset, comparing individual heads and various classifier ensemble methods in terms of accuracy (Acc.), Expected Calibration Error (ECE) and Maximum Calibration Error (MCE).

The individual heads achieve accuracy scores around 63.3%, with ECE values between 5.84% and 6.43%, and MCE values ranging from 15.33% to 18.06%. The classifier ensemble methods reduce calibration errors, with the DLL approach notably lowering the ECE to 3.13% and MCE to 6.62%, significantly outperforming the other methods in terms of calibration. However, the accuracy of the ensemble methods slightly decreases compared to the individual heads.

Table 4: Comparison of accuracy, ECE and MCE in percent of uncalibrated heads and their combination to the classifier ensemble with ResNet18 (He et al., 2016) on Tiny ImageNet (Le and Yang, 2015).

	Model	ResNet18		
		Acc.	ECE	MCE
Baseline	Head 1	63.41	6.02	16.56
	Head 2	63.11	5.91	15.33
	Head 3	63.23	6.43	18.06
	Head 4	63.39	6.05	15.89
	Head 5	63.34	5.84	17.09
Classifier Ensemble (ours)	Avg.	63.32	5.86	16.43
	Vot.	63.32	5.67	14.65
	SL	62.69	5.00	11.35
	DL	62.05	5.09	13.19
	DLL	62.58	**3.13**	**6.62**
	SLpC	63.26	4.92	8.93

5 CONCLUSIONS

In this study, we explored various ensemble techniques using multiple deep learning architectures on the CIFAR-100 and Tiny ImageNet datasets. Our focus was on evaluating the accuracy and calibration performance of our novel classifier ensembles, particularly in reducing Expected Calibration Error (ECE) and Maximum Calibration Error (MCE).

The results show that while individual heads achieve reasonable accuracy, they often exhibit high calibration errors, particularly on larger models. Simple ensemble techniques such as mean averaging and majority voting provide modest improvements in calibration but fail to significantly lower the ECE and MCE. In contrast, metamodel-based ensemble methods consistently outperform these basic techniques in terms of calibration, with our Double-Layer and Double-Layer Large methods beeing particularly effective in reducing both ECE and MCE, albeit with slight reductions in accuracy.

Compared to traditional model ensembles, classifier ensembles with metamodels demonstrated similar improvements in calibration with far fewer parameters, offering a more efficient approach to improving model reliability. These findings suggest that integrating metamodels into classifier ensembles can provide a robust solution for enhancing the calibration of deep learning models, making them more reliable in real-world applications.

Future work could explore the scalability of these methods to even larger datasets and architectures, as well as their potential in more complex tasks like object detection requiring highly calibrated predictions.

ACKNOWLEDGEMENTS

This research was partly funded by Albert and Anneliese Konanz Foundation, the German Research Foundation under grant INST874/9-1 and the Federal Ministry of Education and Research Germany in the project M^2Aind-DeepLearning (13FH8I08IA).

REFERENCES

Bröcker, J. (2009). Reliability, sufficiency, and the decomposition of proper scores. *Quarterly Journal of the Royal Meteorological Society*.

Ebert, N., Mangat, P., and Wasenmüller, O. (2022). Multitask network for joint object detection, semantic segmentation and human pose estimation in vehicle occupancy monitoring. In *Intelligent Vehicles Symposium (IV)*.

Ebert, N., Stricker, D., and Wasenmüller, O. (2023). Transformer-based detection of microorganisms on high-resolution petri dish images. In *International Conference on Computer Vision Workshops (ICCVW)*.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning (ICML)*.

Goodfellow, I. (2016). Deep learning.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks. In *International Conference on Machine Learning (ICML)*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jiang, H., Kim, B., Guan, M., and Gupta, M. (2018). To trust or not to trust a classifier. *Advances in Neural Information Processing Systems (NeurIPS)*.

Kendall, A. and Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems (NeurIPS)*.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images. *Technical report*.

Kumar, A., Sarawagi, S., and Jain, U. (2018). Trainable calibration measures for neural networks from kernel mean embeddings. In *International Conference on Machine Learning (ICML)*.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems (NeurIPS)*.

Le, Y. and Yang, X. (2015). Tiny imagenet visual recognition challenge. *CS 231N*.

Naeini, M. P., Cooper, G., and Hauskrecht, M. (2015). Obtaining well calibrated probabilities using bayesian binning. In *AAAI Conference on Artificial Intelligence*.

Niculescu-Mizil, A. and Caruana, R. (2005). Predicting good probabilities with supervised learning. In *International Conference on Machine Learning (ICML)*.

Oehri, S., Ebert, N., Abdullah, A., Stricker, D., and Wasenmüller, O. (2024). Genformer – generated images are all you need to improve robustness of transformers on small datasets. In *International Conference on Pattern Recognition (ICPR)*.

Platt, J. et al. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*.

Raschka, S., Liu, Y. H., and Mirjalili, V. (2022). *Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python*. Packt Publishing Ltd.

Reichardt, L., Ebert, N., and Wasenmüller, O. (2023). 360deg from a single camera: A few-shot approach for lidar segmentation. In *International Conference on Computer Vision Workshops (ICCVW)*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tomani, C., Cremers, D., and Buettner, F. (2022). Parameterized temperature scaling for boosting the expressive power in post-hoc uncertainty calibration. In *European Conference on Computer Vision*.

Wenzel, F., Snoek, J., Tran, D., and Jenatton, R. (2020). Hyperparameter ensembles for robustness and uncertainty quantification. *Advances in Neural Information Processing Systems (NeurIPS)*.

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. (2017). Aggregated residual transformations for deep neural networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zadrozny, B. and Elkan, C. (2001). Learning and making decisions when costs and probabilities are both unknown. In *International Conference on Knowledge Discovery and Data Mining*.

Zadrozny, B. and Elkan, C. (2002). Transforming classifier scores into accurate multiclass probability estimates. In *International Conference on Knowledge Discovery and Data Mining*.

Zhang, J., Kailkhura, B., and Han, T. Y.-J. (2020). Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning. In *International Conference on Machine Learning (ICML)*.