Tune It or Don't Use It: Benchmarking Data-Efficient Image Classification

Lorenzo Brigato*[1] Björn Barz†[2] Luca Iocchi[1] Joachim Denzler[2]
[1]Sapienza University of Rome [2]Friedrich Schiller University Jena

Abstract

Data-efficient image classification using deep neural networks in settings, where only small amounts of labeled data are available, has been an active research area in the recent past. However, an objective comparison between published methods is difficult, since existing works use different datasets for evaluation and often compare against untuned baselines with default hyper-parameters. We design a benchmark for data-efficient image classification consisting of six diverse datasets spanning various domains (e.g., natural images, medical imagery, satellite data) and data types (RGB, grayscale, multispectral). Using this benchmark, we re-evaluate the standard cross-entropy baseline and eight methods for data-efficient deep learning published between 2017 and 2021 at renowned venues. For a fair and realistic comparison, we carefully tune the hyper-parameters of all methods on each dataset. Surprisingly, we find that tuning learning rate, weight decay, and batch size on a separate validation split results in a highly competitive baseline, which outperforms all but one specialized method and performs competitively to the remaining one.

1. Introduction

Many recent advances in computer vision and machine learning in general have been achieved by large-scale pre-training on massive datasets [9, 8, 23]. As the amount of data grows, the importance of methodological advances vanishes. With the number of training samples approaching infinity, a simple k-nearest neighbor classifier provides optimal performance [29]. The true hallmark of intelligence is, therefore, the ability of learning generalizable concepts from limited amounts of data.

The research area of *deep learning from small data* or *data-efficient deep learning* has been receiving increasing interest in the past couple of years [21, 1, 5, 6]. However, an objective comparison of proposed methods is difficult due to the lack of a common benchmark. Even if two works use the same dataset for evaluation, their random sub-samples of this dataset for simulating a small-data scenario will be different and not directly comparable.

Fortunately, there recently have been activities to establish common benchmarks and organize challenges to foster direct competition between proposed methods [6]. Still, they are often limited to a single dataset, *e.g.*, ImageNet [24], which comprises a different type of data than usually encountered in a small-data scenario.

Moreover, most existing works compare their proposed method against insufficiently tuned baselines [1] or baselines trained with default hyper-parameters [21, 32, 13, 27, 14], which makes it easy to outperform them. However, careful tuning of hyper-parameters, as one would do in practice, is crucial and can have a considerable impact on the final performance [4], as illustrated in Fig. 1. Here, we evaluated the performance of several combinations of learning rate and weight decay for a standard cross-entropy clas-

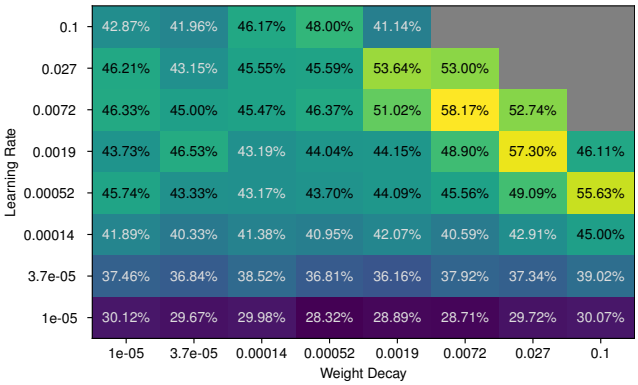


Figure 1: Classification accuracy obtained with standard cross-entropy on ciFAIR-10 with 1% of the training data for different combinations of learning rate and weight decay. Gray configurations led to divergence.

*brigato@diag.uniroma1.it
†bjoern.barz@uni-jena.de

sifier with a Wide ResNet architecture [34] trained on as few as 1% of the CIFAR-10 training data [15] and evaluated on the ciFAIR-10 test set [2] (see Section 4 for details on the training procedure). Typical default hyper-parameters such as a learning rate of 0.1 and weight decay of 1×10^{-4} as used by [6] would achieve ~46% accuracy in this scenario, which is entire 12 percentage points below the optimal performance of ~58%. Even works that do perform hyper-parameter tuning often only optimize the learning rate and keep the weight decay fixed to some default from $[1 \times 10^{-5}, 1 \times 10^{-4}]$. Such a procedure results in similarly suboptimal performance on this small training dataset, which apparently requires much stronger regularization. We can furthermore observe that the best performing hyper-parameter combinations are close to an area of the search space that results in divergence of the training procedure. This makes hyper-parameter optimization a particularly delicate endeavor.

In this work, we establish a direct, objective, and informative comparison by re-evaluating the state of the art in data-efficient image classification. To this end, we introduce a comprehensive benchmark consisting of six datasets from a variety of domains: natural images of everyday objects, fine-grained classification, medical imagery, satellite images, and handwritten documents. Two datasets consist of non-RGB data, where the common large-scale pre-training and fine-tuning procedure is not straightforward, emphasizing the need for methods that can learn from limited amounts of data from scratch. To facilitate evaluating novel methods, we share the dataset splits of our benchmark under https://github.com/cvjena/deic.

Using this benchmark, we re-evaluate eight selected state-of-the-art methods for data-efficient image classification. The hyper-parameters of all methods are carefully optimized for each dataset individually on a validation split, while the final performance is evaluated on a separate test split. Surprisingly and somewhat disillusioning, we find that thorough hyper-parameter optimization results in a strong baseline, which outperforms seven of the eight specialized methods published in the recent literature.

In the following, we first introduce the datasets constituting our benchmark in Section 2. Then, we briefly describe the methods selected for the comparison in Section 3. Our experimental setup and training procedure are detailed in Section 4 and the results are presented in Section 5. Section 6 summarizes the conclusions from our study.

2. Datasets

Most works on deep learning from small datasets use custom sub-sampled versions of popular standard image classification benchmarks such as ImageNet [24] or CIFAR [15]. This limited variety bears the risk of overfitting research progress to individual datasets and the domain covered by them, in this case, photographs of natural scenes and everyday objects. In particular, this is not the domain typically dealt with in a small-data scenario, where specialized data that is difficult to obtain or annotate is in the focus. Additionally, very recent work showed that high performance on ImageNet does not necessarily correlate to high performance on other vision datasets [30].

Therefore, we compile a diverse benchmark consisting of six datasets from a variety of domains and with different data types and numbers of classes. We sub-sampled all datasets to fit the small-data regime, with the exception of CUB [33], which was already small enough. By default, we aimed for 50 training images per class. This full *trainval* split is only used for the final training and furthermore split into a training (~60%) and a validation set (~40%) for hyper-parameter optimization. For testing the final models trained on the trainval split, we used official standard test datasets where they existed. Only for two datasets, namely EuroSAT [12] and ISIC 2018 [7], we had to create own test splits. A summary of the dataset statistics is given in Table 1. In the following, we briefly describe each individual dataset used for our benchmark. Example images from all datasets are shown in Fig. 2.

ImageNet-1k [24] has been *the* standard benchmark for image classification for almost a decade now and also served as a basis for challenge datasets for data-efficient image classification [6]. It comprises images from 1,000 classes of everyday objects and natural scenes collected from the web using image search engines. Due to the large number of classes, a sub-sample of 50 images per class still results in a rather large training dataset compared with the rest of our benchmark.

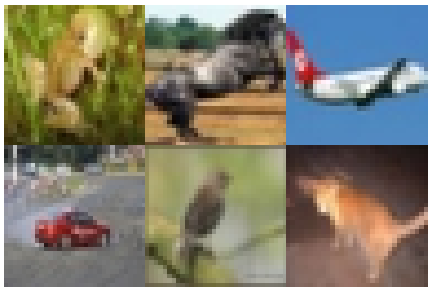
ciFAIR-10 [2] is a variant of the popular CIFAR-10 dataset [15], which comprises low-resolution images of size 32×32 from 10 different classes of everyday objects. To a large part, its popularity stems from the fact that the low image resolution allows for fast training of neural networks and hence rapid experimentation. However, the test dataset of CIFAR-10 contains about 3.3% duplicates from the training set [2], which can potentially bias the evaluation. The ciFAIR-10 dataset [2] provides a variant of the test set, where these duplicates have been replaced with new images from the same domain.

Caltech-UCSD Birds-200-2011 (CUB) [33] is a fine-grained dataset of 200 bird species. Annotating this kind of images typically requires a domain expert and is hence costly. Therefore, the dataset is rather small and only comprises 30 training images per class. Pre-training on related large-scale datasets is hence the de-facto standard for CUB [8, 19, 25, 35], which makes it particularly interesting for

Dataset	Classes	Imgs/Class	#Trainval	#Test	Problem Domain	Data Type
ImageNet-1k [24]	1,000	50	50,000	50,000	Natural Images	RGB
ciFAIR-10 [15, 2]	10	50	500	10,000	Natural Images	RGB (32x32)
CUB [33]	200	30	5,994	5,794	Fine-Grained	RGB
EuroSAT [12]	10	50	500	19,500	Remote Sensing	Multispectral
ISIC 2018 [7]	7	80	560	1,944	Medical	RGB
CLaMM [26]	12	50	600	2,000	Handwriting	Grayscale

Table 1: Datasets constituting our benchmark. Except for CUB, we use sub-samples to simulate a small-data scenario.

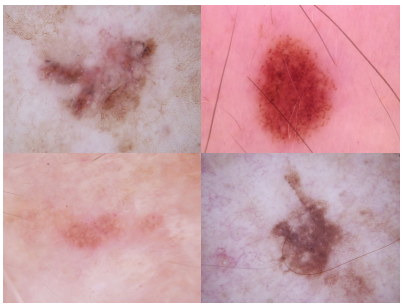
(a) ImageNet



(b) ciFAIR-10

(c) CUB

(d) EuroSAT



(e) ISIC 2018

(f) CLaMM

Figure 2: Example images from the datasets included in our benchmark. For EuroSAT, we only show the RGB bands.

research on data-efficient methods closing the gap between training from scratch and pre-training.

EuroSAT [12] is a multispectral image dataset based on Sentinel-2 satellite images of size 64x64 covering 13 spectral bands. Each image is annotated with one of ten land cover classes. This dataset does not only exhibit a substantial domain shift compared to standard pre-training datasets such as ImageNet but also a different number of input channels. This scenario renders the standard pre-training and fine-tuning procedure impossible.

Nevertheless, Helber et al. [12] adhere to this procedure by fine-tuning a CNN pre-trained on RGB images using different combinations of three out of the 13 channels of EuroSAT. Unsurprisingly, they find that the combination of the R, G, and B channel provides the best performance in this setting. This limitation to three channels due to pre-training is a waste of data and potential. In our experiments on a smaller subset of EuroSAT, we found that using all 13 channels increases the classification accuracy by 9.5% compared to the three RGB channels when training from scratch.

ISIC 2018 [7] is a medical dataset consisting of dermoscopic skin lesion images, annotated with one of seven possible skin disease types. Since medical data usually requires costly expert annotations, this domain is important to be covered by a benchmark on data-efficient deep learning. Due to the small number of classes, we increase the number of images per class to 80 for this dataset, so that the size of the training set is more similar to our other datasets.

CLaMM [26] is a dataset for **C**lassification of **La**tin **M**edieval **M**anuscripts. It was originally used in the ICFHR 2016 Competition for Script Classification, where the task was to classify grayscale images of Latin scripts from handwritten books dated 500 C.E. to 1600 C.E. into one of twelve script style classes such as *Humanistic Cursive*, *Praegothica* etc. This domain is quite different from that of typical pre-training datasets such as ImageNet and one can barely expect any useful knowledge to be extracted from ImageNet about medieval documents. In addition, the standard pre-training and fine-tuning procedure would require the grayscale images to be converted to RGB for being passed through the pre-trained network, which incurs a waste of parameters.

3. Methods

In this section, we present the methods whose performance has been re-evaluated on our benchmark using the original code, where available. We selected approaches for which the authors performed experiments on sub-sampled versions of standard computer vision datasets to prove their effectiveness for learning from small datasets.

Cross-entropy loss is the widely used standard loss function for classification. We use it as a baseline with standard network architectures and optimization algorithms.

Deep hybrid networks (DHN) represent one of the first attempts to incorporate pre-defined geometric priors via a hybrid approach of combining pre-defined and learned representations [21, 22]. According to the authors, decreasing the number of parameters to learn could make deep networks more data-efficient, especially in settings where the scarcity of data would not allow the learning of low-level feature extractors. Deep hybrid networks first perform a scattering transform on the input image generating feature maps and then apply standard convolutional blocks. The spatial scale of the scattering transform is controlled by the parameter $J \in \mathbb{N}$.

Orthogonal low-rank embedding (OLÉ) is a geometric loss for deep networks that was proposed in [16] to reduce intra-class variance and enforce inter-class margins. This method collapses deep features into a learned linear subspace, or union of them, and inter-class subspaces are pushed to be as orthogonal as possible. The contribution of the low-rank embedding to the overall loss is weighted by the hyper-parameter λ_{ole}.

Grad-ℓ_2 penalty is a regularization strategy tested in the context of improving generalization on small datasets in [3]. The ℓ_2 (squared) gradient norm is computed with respect to the input samples and used as a penalty in the loss weighted by parameter λ_{grad}. Among many regularization approaches evaluated in [3], we have chosen the grad-ℓ_2 penalty because it was among the best performing methods in the experiments with ResNet and sub-sampled versions of CIFAR-10. Since the grad-ℓ_2 penalty is proposed as an alternative to weight decay, we disable weight decay for this method. Moreover, differently from the original implementation, we enabled the use of batch normalization since, without this component, we obtained extremely low results in preliminary experiments.

Cosine loss was proposed in [1] to decrease overfitting in problems with scarce data. Thanks to an ℓ_2 normalization of the learned feature space, the cosine loss is invariant against scaling of the network output and solely focuses on the directions of feature vectors instead of their magnitude. In contrast to the softmax function used with the cross-entropy loss, the cosine loss does not push the activations of the true class towards infinity, which is commonly considered as a cause of overfitting [28, 11]. Moreover, a further increase of performance was obtained by combining the cosine with the cross-entropy loss after an additional layer on top of the embeddings learned with the cosine loss.

Harmonic networks (HN) use a set of preset filters based on windowed cosine transform at several frequencies which are combined by learnable weights [31, 32]. Similar to hybrid networks, the idea of the harmonic block is to have a useful geometric prior that can help to avoid overfitting. Harmonic networks use Discrete Cosine Transform filters which have excellent energy compaction properties and are widely used for image compression.

Full convolution (F-Conv) was proposed in [13] to improve translation invariance of convolutional filters. Standard CNNs exploit image boundary effects and learn filters that can exploit the absolute spatial locations of objects in images. In contrast, full convolution applies each value in the filter on all values in the image. According to [13], improving translation invariance strengthens the visual inductive prior of convolution, leading to increased data efficiency in the small-data setting.

Dual Selective kernel networks have been proposed and designed in [27] to be more data-efficient. The standard residual block is modified, keeping the skip connection, with two forward branches that use 1×1 convolutions, selective kernels [18] and an anti-aliasing module. To further regularize training, only one of the two branches is randomly selected in the forward and backward passes, while at inference, the two paths are weighted equally.

Besides the specialized network architecture, the original work uses a combination of three custom loss functions [27]. Despite best efforts, we were unable to derive the correct implementation from the ambiguous description of these loss functions in the paper. Therefore, we only use the DSK network architecture with cross-entropy loss.

T-vMF Similarity is a generalization of the cosine similarity that was recently presented in [14] to make modern CNNs more robust to some realistic learning situations such as class imbalance, few training samples, and noisy labels. As the name suggests, this similarity is mainly based on the von Mises-Fisher distribution of directional statistics and built on top of the heavy-tailed student-t distribution. The combination of these two ingredients provides high compactness in high-similarity regions and low similarity in heavy-tailed ones. The degree of compactness/dispersion of the similarity is controlled by the parameter κ.

4. Experimental setup

In this section, we give an overview of the experimental pipeline that we followed for a fair evaluation of the aforementioned methods on the six datasets that constitute our benchmark.

4.1. Evaluation metrics

In our benchmark, we evaluate each method on each dataset with the widely used balanced classification accuracy. This metric is defined as the average per-class accuracy, *i.e.*, the average of the diagonal in the row-normalized confusion matrix. We turned our attention toward this metric since some datasets in our benchmark do not have balanced test sets. In any case, for balanced test sets, the balanced accuracy equals the standard classification accuracy.

Since our benchmark contains multiple datasets it is hard to directly make a comparison between two methods without computing an overall ranking. Therefore, for each method, we also compute the average balanced accuracy across all datasets to provide a simple and intuitive way to rank methods. Additionally, in this manner, future methods will be easily comparable with those already evaluated.

4.2. Data pre-processing and augmentation

All input images were normalized by subtracting the channel-wise mean and dividing by the standard deviation computed on the *trainval* split. We applied standard data augmentation policies with slightly varying configurations, adapted to the specific characteristics of each dataset and problem domain. Note that none of the currently re-evaluated methods in our benchmark had as original contribution a specialized data augmentation technique. Nothing prevents the use of an augmentation-based method from partaking in the benchmark.

For datasets with a small, fixed image resolution, *i.e.*, ciFAIR-10 and EuroSAT, we perform random shifting by 12.5% of the image size and horizontal flipping in 50% of the cases. For all other datasets, we apply scale augmentation using the `RandomResizedCrop` transform from PyTorch[1] as follows: A crop with a random aspect ratio drawn from $[\frac{3}{4}, \frac{4}{3}]$ and an area between $A_{\min}$ and 100% of the original image area is extracted from the image and then resized to 224×224 pixels. The minimum fraction $A_{\min}$ of the area was determined based on preliminary experiments to ensure that a sufficient part of the image remains visible. It therefore varies depending on the dataset: We use $A_{\min} = 10\%$ for ImageNet, $A_{\min} = 20\%$ for CLaMM and $A_{\min} = 40\%$ for CUB and ISIC 2018.

For ISIC 2018 and EuroSAT, we furthermore perform random vertical flipping in addition to horizontal flipping, since these datasets are completely rotation-invariant and vertical reflection augments the training sets without drifting them away from the test distributions. On CLaMM, in contrast, we do not perform any flipping, since handwritten scripts are not invariant even against horizontal flipping.

4.3. Architecture and optimizer

To perform a fair comparison, we use the same backbone CNN architecture for all methods. More precisely, for ciFAIR-10, we employ a Wide Residual Network (WRN) [34], precisely WRN-16-8, which is widely used in the existing literature for data-efficient classification on CIFAR. For all other cases, the popular and well-established ResNet-50 (RN50) architecture [10] is used. Note that we made changes to the architecture when that was an original contribution of the paper, but all those changes were applied to the selected base architecture. Due to the high popularity of residual networks, the majority of the selected approaches were originally tested with a RN/WRN backbone. This fact allowed us to perform a straightforward porting of the network setup, when necessary.

We furthermore employ a common optimizer and training schedule across all methods and datasets to avoid any kind of optimization bias. We use standard stochastic gradient descent (SGD) with a momentum of 0.9 and weight decay and a cosine annealing learning rate schedule [20], which reduces the learning rate smoothly during the training process. The initial learning rate and the weight decay factor are optimized for each method and dataset individually together with any method-specific hyper-parameters as detailed in the next subsection. The total number of training epochs for each dataset was chosen according to preliminary experiments.

[1] https://pytorch.org/vision/stable/transforms.html#torchvision.transforms.RandomResizedCrop

Hyper-Parameter	ImageNet	ciFAIR-10	CUB	EuroSAT	ISIC 2018	CLaMM
Learning Rate			`loguniform`(1e-4, 0.1)			
Weight Decay			`loguniform`(1e-5, 0.1)			
Batch Size	{8, 16, 32}	{10, 25, 50}	{8, 16, 32}	{10, 25, 50}	{8, 16, 32}	{8, 16, 32}
Epochs	200 (500)	500	200	500	500	500
HPO Trials	100	250	100	250	100	100
Grace Period	10	50	10	25	25	25

Table 2: Summary of hyper-parameters searched/used with ASHA [17]. Method specific hyper-parameters were included in the search space but not included in this table due to space limitations. An epoch number in parentheses means that a higher number of epochs was used for the final training than for the hyper-parameter optimization.

4.4. Hyper-parameter optimization

As we have discussed in Section 1 and shown in Fig. 1, the choice of hyper-parameters has a substantial effect on the classification performance that should in no case be underestimated. Careful hyper-parameter optimization (HPO) [4] is therefore not only crucial for applying deep learning techniques in practice but also for a fair comparison between different methods, so that each can obtain its optimal performance. Comparing against an untuned baseline with default hyper-parameters is as good as no comparison at all.

For our benchmark, we hence first tune the hyper-parameters of each method on each individual dataset using a training and a validation split, which are disjoint from the test set used for final performance evaluation (see Section 2). For any method, we tune the initial learning rate and weight decay, sampled from a log-uniform space, as well as the batch size, chosen from a pre-defined set. Details about the search space are provided in Table 2. In addition to these general hyper-parameters, any method-specific hyper-parameters are tuned as well simultaneously, considering the boundaries used in the original paper, if applicable, or lower and upper bounds estimated by ourselves.

For selecting hyper-parameters to be tested and scheduling experiments, we employ Asynchronous HyperBand with Successive Halving (ASHA) [17] as implemented in the Ray library[2]. This search algorithm exploits parallelism and aggressive early-stopping to tackle large-scale hyper-parameter optimization problems. Trials are evaluated and stopped based on their accuracy on the validation split.

Two main parameters need to be configured for the ASHA algorithm: the number of trials and the grace period. The former controls the number of hyper-parameter configurations tried in total while the latter the minimum time after which a trial can be stopped. Since the number of trials corresponds to the time budget available for HPO, we choose larger values for smaller datasets, where training is faster. The grace period, on the other hand, should be large enough to allow for a sufficient number of training iterations before comparing trials. Therefore, we choose larger grace periods for smaller datasets, where a single epoch comprises fewer training iterations. The exact values for each dataset as well as the total number of training epochs can be found in Table 2. These values were determined based on preliminary experiments with the cross-entropy baseline.

[2] https://docs.ray.io/en/master/tune/

4.5. Final training and evaluation

After having completed HPO using the procedure described above, we train the classifier with the determined configuration on the combined training and validation split and evaluate the balanced classification accuracy on the test split. To account for the effect of random initialization, this training is repeated ten times and we report the balanced average accuracy.

5. Results

In the following, we first present the results of re-evaluating the eight methods described in Section 3 and the baseline on our benchmark introduced in Section 2, after carefully tuning all methods on each dataset. Then, we compare the performance obtained by our re-implementations, including the baseline, with other values published in the literature.

5.1. Data-efficient image classification benchmark

Table 3 presents the average balanced classification accuracy over 10 runs with different random initializations for all methods and datasets. We performed Welch's t-test to assess the significance of the advantage of the best method for each dataset in comparison to all others. Most results are significantly worse on a level of 5% than the best method on the respective dataset, with only two exceptions: T-vMF Similarity on ciFAIR-10 and Harmonic Networks on CLaMM perform similar to the best method on these

Method	ImageNet	ciFAIR-10	CUB	EuroSAT	ISIC 2018	CLaMM	Average
Cross-Entropy Baseline	44.97	**58.22**	71.44	90.27	67.19	**75.34**	67.90
Deep Hybrid Networks [21, 22]	38.69	54.21	52.54	91.15	59.64	65.74	60.33
OLÉ [16]	43.05	54.92	63.32	89.29	62.89	71.42	64.15
Grad-ℓ_2 Penalty [3]	25.21	51.03	51.94	79.33	60.21	65.10	55.47
Cosine Loss [1]	37.22	52.39	66.94	88.53	62.42	68.89	62.73
Cosine Loss + Cross-Entropy [1]	44.39	51.74	70.80	88.77	64.52	69.29	64.92
Harmonic Networks [31, 32]	**46.36**	56.50	**72.26**	**92.09**	**70.42**	*74.59*	**68.70**
Full Convolution [13]	36.58	55.00	64.90	90.82	61.70	63.33	62.06
Dual Selective Kernel Networks [27]	45.21	54.06	71.02	91.25	64.78	61.51	64.64
T-vMF Similarity [14]	42.79	*57.50*	67.43	88.53	65.37	66.40	64.67

Table 3: Average balanced classification accuracy in % over 10 runs for each task and across all tasks. The best value per dataset is highlighted in bold font. Numbers in italic font indicate that the result is not significantly worse than the best one on a significance level of 5%.

datasets, which is the baseline in both cases.

This leads us to the main surprising finding of this benchmark: When tuned carefully, the standard cross-entropy baseline is very competitive with the published methods specialized for deep learning from small datasets. On ciFAIR-10 and CLaMM, it actually is the best performing method. It obtains the second rank on CUB and ISIC 2018, the third rank on ImageNet, and the fourth on EuroSAT. The baseline scores an average balanced accuracy across all datasets of 67.90%, which beats all other methods except Harmonic Networks by a large margin (the next best average accuracy is only 64.92%).

Harmonic Networks are the overall champion of our benchmark, with an average balanced accuracy of 68.70%. On the four datasets where they outperform the baseline, however, they only surpass it by 1%-5%.

Overall, the finding that the vast majority of recent methods for data-efficient image classification does not even achieve the same performance as the baseline is sobering. We attribute this to the fact that the importance of hyper-parameter optimization is immensely underestimated, resulting in misleading comparisons of novel approaches with weak and underperforming baselines.

5.2. Published baselines are underperforming

We show further evidence of why tuning the hyper-parameters and not neglecting the baseline in scenarios with small datasets is fundamental to perform a fair comparison between different methods.

We analyzed the original results reported for the methods considered in our study and selected those that shared a similar setup. Note that due to the lack of a standard benchmark and the common practice of randomly sub-sampling large datasets, we are unable to conduct a fair comparison with the same dataset split, training procedure, etc. Still, our benchmark shares the base dataset and network architecture with the selected cases. Therefore, we believe that this analysis is suitable for supporting our point regarding the common practice of comparing tuned proposed methods with underperforming baselines.

The results of this analysis are shown in Table 4. Deep Hybrid Networks and Harmonic Networks were originally tested with a WRN-16-8 on CIFAR-10 while Full Convolution employed RN50 on ImageNet. In both cases, training sets were comprised of 50 images per class. Our baseline clearly outperforms the original baselines by large margins (Table 4, left part). More precisely, our models surpass the reported ones by ~12, ~6, and ~18 percentage points on the CIFAR and ImageNet setups. Recall also that the ciFAIR-10 test set is slightly harder than the CIFAR-10 one due to the removal of duplicates [2].

On the contrary, for the case of the proposed methods (Table 4, right part), the difference between ours and original results is sharply less evident. Our DHN and HN slightly underperform the original ones by a ~0.5 and ~2 percentage points, respectively. However, this was expected due to the higher difficulty of ciFAIR-10. On ImageNet instead, our F-Conv model outperforms the original one by ~5 percentage points, confirming once again that careful HPO can further boost the performance.

From this analysis it seems clear that proposed methods are usually tuned to obtain an optimal or near-optimal result while baselines are trained with default hyper-parameters that have been found useful for large datasets but do not

Cross-Entropy Baseline			
Publication	Dataset	Network	Accuracy
[21]	CIFAR-10	WRN-16-8	46.5 ± 1.4
[32]	CIFAR-10	WRN-16-8	52.2 ± 1.8
Ours	ciFAIR-10	WRN-16-8	58.22 ± 0.9
[13]	ImageNet	RN50	26.39
Ours	ImageNet	RN50	44.97 ± 0.3

Other Methods			
Method	Dataset	Network	Accuracy
DHN [21]	CIFAR-10	WRN-16-8	54.7 ± 0.6
DHN (Ours)	ciFAIR-10	WRN-16-8	54.21 ± 0.4
HN [32]	CIFAR-10	WRN-16-8	58.4 ± 0.9
HN (Ours)	ciFAIR-10	WRN-16-8	56.50 ± 0.5
F-Conv [13]	ImageNet	RN50	31.1
F-Conv (Ours)	ImageNet	RN50	36.58 ± 0.4

Table 4: Summary of ours/published results of the cross-entropy baseline (left) and other methods (right) on similar setups.

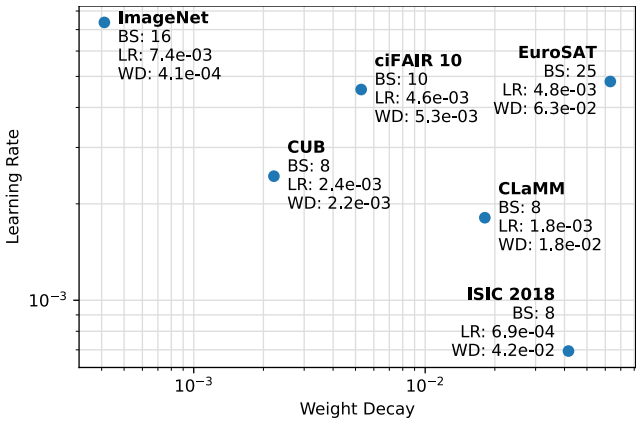


Figure 3: Hyper-parameters found with ASHA [17] for the cross-entropy baseline. BS = batch size, LR = learning rate, WD = weight decay.

necessarily generalize to smaller ones.

5.3. Optimal hyper-parameters

For reproducibility, but also to gain further insights into hyper-parameter optimization for small datasets, we show the best hyper-parameter combinations found during our search for the cross-entropy baseline in Fig. 3.

We can observe that small batch sizes seem to be beneficial, despite the use of batch normalization. While the learning rate exhibits a rather small range of values from 0.7×10^{-3} to 7.4×10^{-3} across datasets and spans only one order of magnitude, weight decay varies within a range of two orders of magnitude from 4.1×10^{-4} to 1.8×10^{-2}.

Furthermore, learning rate and weight decay appear to be negatively correlated. Higher learning rates are usually accompanied by smaller weight decay factors. The same correlation can be observed in Fig. 1. A quantitative analysis over the hyper-parameters of all methods used in our study instead of only the baseline yields a correlation of $r = -.28$, $p = .02$. After taking the logarithm of learning rate and weight decay, the correlation is strengthened to $r = -.58$, $p < .01$.

6. Conclusions

In this paper, we laid the foundation for fair and appropriate comparisons among modern data-efficient image classifiers. The motivations that brought us to our work are mainly two-fold: the lack of a common evaluation benchmark with fixed datasets, architectures, and training pipelines; and the experimental evidence of weak assessments of baselines due to a lack of careful tuning.

The re-evaluation of eight selected state-of-the-art methods guided us to the surprising and sobering conclusion that the standard cross-entropy loss ranks second in our benchmark only behind Harmonic Networks and, competes with or outperforms the remaining methods.

With these results in mind, we conclude that the importance of hyper-parameter optimization is immensely undervalued and should be taken into account in future studies to elude misleading comparisons of new approaches with weak and underperforming baselines. The publication of our benchmark heads towards this direction and is considered by ourselves an important contribution for the community of data-efficient image classification.

References

[1] Björn Barz and Joachim Denzler. Deep learning on small datasets without pre-training using cosine loss. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2020. 1, 4, 7

[2] Björn Barz and Joachim Denzler. Do we train on test data? purging CIFAR of near-duplicates. *Journal of Imaging*, 6(6), 2020. 2, 3, 7

[3] Alberto Bietti, Grégoire Mialon, Dexiong Chen, and Julien Mairal. A kernel perspective for regularizing deep neural networks. In *International Conference on Machine Learning (ICML)*, 2019. 4, 7

[4] Bernd Bischl, Martin Binder, Michel Lang, Tobias Pielok, Jakob Richter, Stefan Coors, Janek Thomas, Theresa Ullmann, Marc Becker, Anne-Laure Boulesteix, et al. Hyperparameter optimization: Foundations, algorithms, best practices and open challenges. *arXiv preprint arXiv:2107.05847*, 2021. 1, 6

[5] Lorenzo Brigato and Luca Iocchi. A close look at deep learning with small data. In *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021. 1

[6] Robert-Jan Bruintjes, Attila Lengyel, Marcos Baptista Rios, Osman Semih Kayhan, and Jan van Gemert. VIPriors 1: Visual inductive priors for data-efficient deep learning challenges. *arXiv preprint arXiv:2103.03768*, 2021. 1, 2

[7] Noel Codella, Veronica Rotemberg, Philipp Tschandl, M Emre Celebi, Stephen Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael Marchetti, et al. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (ISIC). *arXiv preprint arXiv:1902.03368*, 2019. 2, 3

[8] Yin Cui, Yang Song, Chen Sun, Andrew Howard, and Serge Belongie. Large scale fine-grained categorization and domain-specific transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4109–4118, 2018. 1, 2

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. 1

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5

[11] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 4

[12] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019. 2, 3

[13] Osman Semih Kayhan and Jan C van Gemert. On translation invariance in CNNs: Convolutional layers can exploit absolute spatial location. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 4, 7, 8

[14] Takumi Kobayashi. T-vMF similarity for regularizing intra-class feature distribution. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 5, 7

[15] Alex Krizhevsky. *Learning multiple layers of features from tiny images*. 2009. 2, 3

[16] José Lezama, Qiang Qiu, Pablo Musé, and Guillermo Sapiro. OLÉ: Orthogonal low-rank embedding-a plug and play geometric loss for deep learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 4, 7

[17] Liam Li, Kevin Jamieson, Afshin Rostamizadeh, Ekaterina Gonina, Jonathan Ben-tzur, Moritz Hardt, Benjamin Recht, and Ameet Talwalkar. A system for massively parallel hyperparameter tuning. *Conference of Machine Learning and Systems*, 2020. 6, 8

[18] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. Selective kernel networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 4

[19] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear CNN models for fine-grained visual recognition. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1449–1457, 2015. 2

[20] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations (ICLR)*, 2017. 5

[21] Edouard Oyallon, Eugene Belilovsky, and Sergey Zagoruyko. Scaling the scattering transform: Deep hybrid networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 1, 4, 7, 8

[22] Edouard Oyallon, Sergey Zagoruyko, Gabriel Huang, Nikos Komodakis, Simon Lacoste-Julien, Matthew Blaschko, and Eugene Belilovsky. Scattering networks for hybrid representation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018. 4, 7

[23] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. Technical report, OpenAI blog, 2019. 1

[24] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 1, 2, 3

[25] Marcel Simon, Erik Rodner, Trevor Darell, and Joachim Denzler. The whole is more than its parts? from explicit to implicit pose normalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 2

[26] Dominique Stutzmann. Clustering of medieval scripts through computer image analysis: towards an evaluation protocol. *Digital Medievalist*, 10, 2016. 3, 4

[27] Pengfei Sun, Xuan Jin, Wei Su, Yuan He, Hui Xue, and Quan Lu. A visual inductive priors framework for data-efficient image classification. In *European Conference on Computer Vision (ECCV) Workshops*, pages 511–520, Cham, 2020. Springer International Publishing. 1, 4, 5, 7

[28] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016. 4

[29] Antonio Torralba, Rob Fergus, and William T Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(11):1958–1970, Nov 2008. 1

[30] Lukas Tuggener, Jürgen Schmidhuber, and Thilo Stadelmann. Is it enough to optimize cnn architectures on imagenet? *arXiv preprint arXiv:2103.09108*, 2021. 2

[31] Matej Ulicny, Vladimir A Krylov, and Rozenn Dahyot. Harmonic networks for image classification. In *BMVC*, 2019. 4, 7

[32] Matej Ulicny, Vladimir A Krylov, and Rozenn Dahyot. Harmonic networks with limited training samples. In *European Signal Processing Conference (EUSIPCO)*, 2019. 1, 4, 7, 8

[33] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 2, 3

[34] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Edwin R. Hancock Richard C. Wilson and William A. P. Smith, editors, *British Machine Vision Conference (BMVC)*, pages 87.1–87.12. BMVA Press, September 2016. 2, 5

[35] Heliang Zheng, Jianlong Fu, Tao Mei, and Jiebo Luo. Learning multi-attention convolutional neural network for fine-grained image recognition. In *IEEE International Conference on Computer Vision (ICCV)*, volume 6, 2017. 2