Article

Do We Train on Test Data? Purging CIFAR of Near-Duplicates

Björn Barz * and **Joachim Denzler**

Computer Vision Group, Friedrich Schiller University Jena, Ernst-Abbe-Platz 2, 07743 Jena, Germany; joachim.denzler@uni-jena.de

*   Correspondence: bjoern.barz@uni-jena.de

check for updates

Abstract: The CIFAR-10 and CIFAR-100 datasets are two of the most heavily benchmarked datasets in computer vision and are often used to evaluate novel methods and model architectures in the field of deep learning. However, we find that 3.3% and 10% of the images from the test sets of these datasets have duplicates in the training set. These duplicates are easily recognizable by memorization and may, hence, bias the comparison of image recognition techniques regarding their generalization capability. To eliminate this bias, we provide the "fair CIFAR" (ciFAIR) dataset, where we replaced all duplicates in the test sets with new images sampled from the same domain. The training set remains unchanged, in order not to invalidate pre-trained models. We then re-evaluate the classification performance of various popular state-of-the-art CNN architectures on these new test sets to investigate whether recent research has overfitted to memorizing data instead of learning abstract concepts. We find a significant drop in classification accuracy of between 9% and 14% relative to the original performance on the duplicate-free test set. We make both the ciFAIR dataset and pre-trained models publicly available and furthermore maintain a leaderboard for tracking the state of the art.

Keywords: image classification; deep learning; reproducibility; duplicates

1. Introduction

Almost ten years after the first instantiation of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [1], image classification is still a very active field of research. The majority of recent approaches belong to the domain of *deep learning* with several new architectures of *convolutional neural networks* (CNNs) being proposed for this task every year and trying to improve the accuracy on held-out test data by a few percent points [2–8]. A key to the success of these methods is the availability of large amounts of training data [9,10]. The world wide web has become a very affordable resource for harvesting such large datasets in an automated or semi-automated manner [11–14].

A problem of this approach is that there is no effective automatic method for filtering out near-duplicates among the collected images. When the dataset is split up later into a training, a test, and maybe even a validation set, this can result in the presence of near-duplicates of test images in the training set. Usually, the post-processing with regard to duplicates is limited to removing exact pixel-level duplicates [11,12]. However, many duplicates are less obvious and might vary with respect to contrast, translation, stretching, color shift, etc. Such variations can easily be emulated by data augmentation, so that these variants will actually become part of the augmented training set.

For a proper scientific evaluation and model selection, the presence of such duplicates is a critical issue: Ideally, researchers and machine learning practitioners aim at comparing models with respect to their ability of generalizing to unseen data. With a growing number of duplicates, however, we run the risk of comparing them in terms of their capability of memorizing the training data, which increases with model capacity. This is especially problematic when the difference between the error

rates of different models is as small as it is nowadays, i.e., sometimes just one or two percent points. The significance of these performance differences, hence, depends on the overlap between test and training data. In some fields, such as fine-grained recognition, this overlap has already been quantified for some popular datasets, for example, for the Caltech-UCSD Birds dataset [15,16].

In this work, we assess the number of test images that have near-duplicates in the training set of two of the most heavily benchmarked datasets in computer vision: CIFAR-10 and CIFAR-100 [12]. We will first briefly introduce these datasets in Section 2 and describe our duplicate search approach in Section 3. In a nutshell, we search for nearest neighbor pairs between test and training set in a CNN feature space and inspect the results manually, assigning each detected pair into one of four duplicate categories. We find that 3.3% of CIFAR-10 test images and a startling number of 10% of CIFAR-100 test images have near-duplicates in their respective training sets. Some examples are shown in Figure 1.

Subsequently, we replace all these duplicates with new images from the Tiny Images dataset [17], which was the original source for the CIFAR images (see Section 4). To determine whether recent research results have already been affected by these duplicates, we finally re-evaluate the performance of several state-of-the-art CNN architectures on these new test sets in Section 5.

Similar to our work, Recht et al. [18] have recently sampled a completely new test set for CIFAR-10 from Tiny Images to assess how well existing models generalize to truly unseen data. Furthermore, they note parenthetically that the CIFAR-10 test set comprises 8% duplicates with the training set, which is more than twice as much as we have found. As opposed to their work, however, we also analyze CIFAR-100 and only replace the duplicates with new images, while leaving the remaining ones untouched. Moreover, we employ a fine-grained distinction between three different types of duplicates. We make the list of duplicates found, the new test sets, and pre-trained models publicly available at https://cvjena.github.io/cifair/.

2. The CIFAR Datasets

There are two different CIFAR datasets [12]: CIFAR-10, which comprises 10 classes, and CIFAR-100, which comprises 100 classes. Both contain 50,000 training and 10,000 test images. Neither the classes nor the data of these two datasets overlap, but both have been sampled from the same source: the Tiny Images dataset [17].

In this context, the word "tiny" refers to the resolution of the images, not to their number. Quite the contrary, Tiny Images comprises approximately 80 million images collected automatically from the web by querying image search engines for approximately 75,000 synsets of the WordNet ontology [19]. However, all images have been resized to the "tiny" resolution of 32×32 pixels.

This low image resolution combined with the much more manageable size of the CIFAR datasets allows for fast training of CNNs. Therefore, they have established themselves as one of the most popular benchmarks in the field of computer vision.
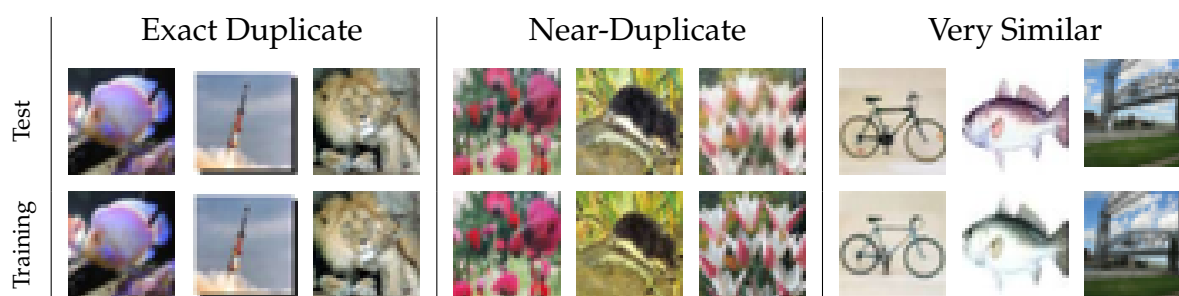

Figure 1. Examples for different types of duplicates between the CIFAR-100 test and training set. The top row shows images from the test set and the bottom row shows their nearest neighbors from the training set in a convolutional neural network (CNN) feature space. The three types of duplicates are described in the main text.

3. Hunting Duplicates

3.1. Mining Duplicate Candidates

As we have argued above, simply searching for exact pixel-level duplicates is not sufficient, since there may also be slightly modified variants of the same scene that vary by contrast, hue, translation, stretching, etc. Thus, we follow a content-based image retrieval approach [20–22] for finding duplicate and near-duplicate images: We train a lightweight CNN architecture proposed by Barz et al. [8] on the training set and then extract L^2-normalized features from the global average pooling layer of the trained network for both training and testing images. Such classification-based features have been found to be useful for comparing images with regard to their visual similarity, and thus enjoy popularity for image retrieval applications [21].

For each test image, we find the nearest neighbor from the training set in terms of the Euclidean distance in that feature space. Due to the L^2 normalization applied to the feature vectors, the Euclidean distance is equivalent to the cosine distance in this case, which is the distance metric used by most state-of-the-art image retrieval techniques [22–24]. Since we do not simply use features extracted from pre-trained models, but train the network on the training samples of the respective dataset, the feature representations are highly adapted to the domain covered by the images in the dataset. Thanks to the data augmentation applied during training, the network sees many variants of each training image and is encouraged to learn features that are largely invariant against the kind of transformations that can result in near-duplicate images.

3.2. Manual Annotation

Given the nearest neighbors of each test image in the training set and their distance in the learned feature space, it would be easy to capture the majority of duplicates by simply thresholding the distance between these pairs. However, such an approach would result in a high number of false positives as well. Therefore, we inspect the detected pairs manually, sorted by increasing distance. In a graphical user interface (GUI) depicted in Figure 2, the annotator can inspect the test image and its duplicate, their distance in the feature space, and a pixel-wise difference image.

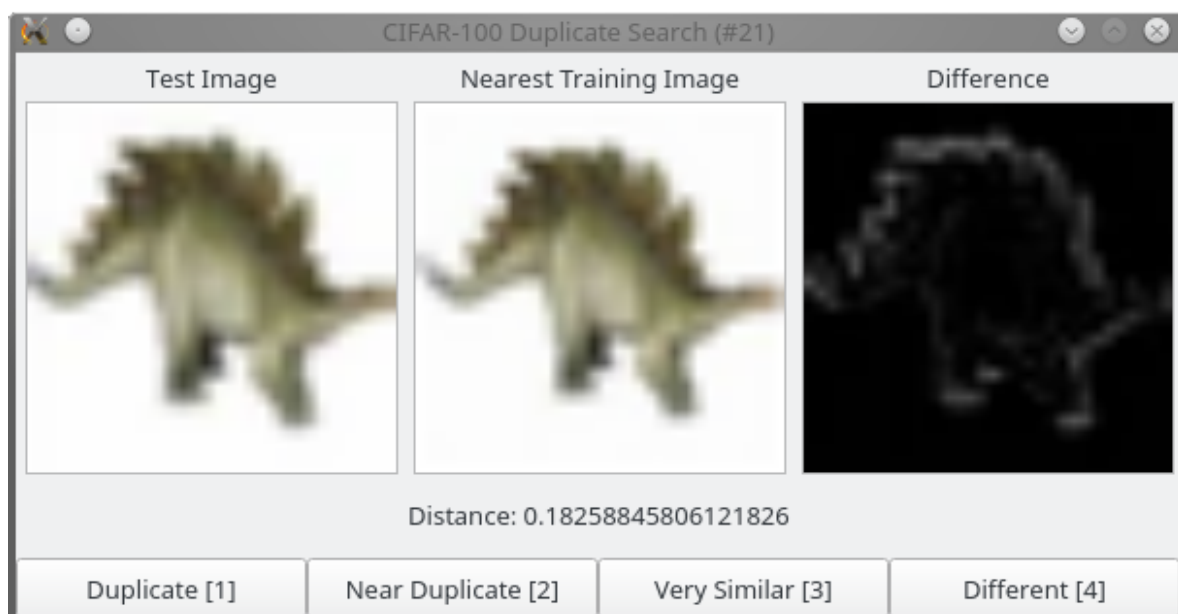The pair is then manually assigned to one of four classes:

Figure 2. GUI for duplicate annotation.

Exact Duplicate Almost all pixels in the two images are approximately identical.
Near-Duplicate The content of the images is exactly the same, i.e., both originated from the same
    camera shot. However, different post-processing might have been applied to this original scene,
    e.g., color shifts, translations, scaling, etc.
Very Similar The contents of the two images are different, but highly similar, so that the difference
    can only be spotted at the second glance.
Different The pair does not belong to any other category.

Figure 1 shows some examples for the three categories of duplicates from the CIFAR-100 test set, where we picked the 10th, 50th, and 90th percentile image pair for each category, according to their distance. In the remainder of this paper, the word "duplicate" will usually refer to any type of duplicate, not necessarily to exact duplicates only.

We used a single annotator and stopped the annotation once the class "Different" has been assigned to 20 pairs in a row. In addition to spotting duplicates of test images in the training set, we also search for duplicates within the test set, since these also distort the performance evaluation. Note that we do not search for duplicates within the training set. In the worst case, the presence of such duplicates biases the weights assigned to each sample during training, but they are not critical for evaluating and comparing models.

3.3. Duplicate Statistics

We found 891 duplicates from the CIFAR-100 test set in the training set and 104 duplicates within the test set itself. In total, 10% of the test images have duplicates. The situation is slightly better for CIFAR-10, where we found 286 duplicates in the training and 39 in the test set, amounting to 3.25% of the test set. This is probably due to the much broader type of object classes in CIFAR-10: We suppose it is easier to find 5000 different images of birds than 500 different images of maple trees, for example.

The vast majority of duplicates belongs to the category of near-duplicates, as can be seen in Figure 3. It is worth noting that there are no exact duplicates in CIFAR-10 at all, as opposed to CIFAR-100. This might indicate that the basic duplicate removal step mentioned by Krizhevsky et al. [9] for the construction of CIFAR-10 has been omitted during the creation of CIFAR-100.

Table 1 lists the classes with the most duplicates for both datasets. The only classes without any duplicates in CIFAR-100 are "bowl", "bus", and "forest".

On the subset of test images with duplicates in the training set, the ResNet-110 [2] models from our experiments in Section 5 achieve error rates of 0% and 2.9% on CIFAR-10 and CIFAR-100, respectively. These are in stark contrast to the 5.3% and 26.1% obtained by this model on the full test set, verifying that even near-duplicate and highly similar images can be classified far too easily by memorization.
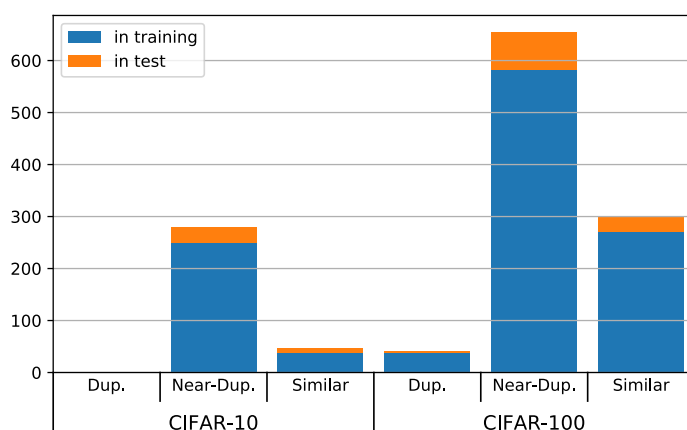

Figure 3. Number of duplicates per type between test and training set (blue) and within the test set (orange).

CIFAR-10		CIFAR-100	
frog	59	cockroach	33
automobile	55	orange	29
airplane	47	lawn mower	28
deer	40	apple	26
bird	26	skunk	26
horse	20	keyboard	23
dog	19	leopard	23
ship	17	wolf	21
truck	14	worm	20
cat	11		

Table 1. The classes with the most duplicates.

4. The Duplicate-Free ciFAIR Test Dataset

To create a fair test set for CIFAR-10 and CIFAR-100, we replace all duplicates identified in the previous section with new images sampled from the Tiny Images dataset [17], which was also the source for the original CIFAR datasets.

We took care not to introduce any bias or domain shift during the selection process. To this end, each replacement candidate was inspected manually in a graphical user interface (see Figure 4), which displayed the candidate and the three nearest neighbors in the feature space from the existing training and test sets. We approved only those samples for inclusion in the new test set that could not be considered duplicates (according to the definitions in Section 3) of any of the three nearest neighbors.

Furthermore, we followed the labeler instructions provided by Krizhevsky et al. [12] for CIFAR-10. However, separate instructions for CIFAR-100, which was created later, have not been published. By inspecting the data, we found that some of the original instructions seem to have been relaxed for this dataset. For example, CIFAR-100 does include some line drawings and cartoons as well as images containing multiple instances of the same object category. Both types of images were excluded from CIFAR-10. Therefore, we also accepted some replacement candidates of these kinds for the new CIFAR-100 test set.

We term the datasets obtained by this modification as ciFAIR-10 and ciFAIR-100 ("fair CIFAR"). They consist of the original CIFAR training sets and the modified test sets, which are free of duplicates. ciFAIR can be obtained online at https://cvjena.github.io/cifair/.

5. Re-Evaluation of the State of the Art

Two questions remain: Were recent improvements to the state-of-the-art in image classification on CIFAR actually due to the effect of duplicates, which can be memorized better by models with higher capacity? Does the ranking of methods change given a duplicate-free test set?
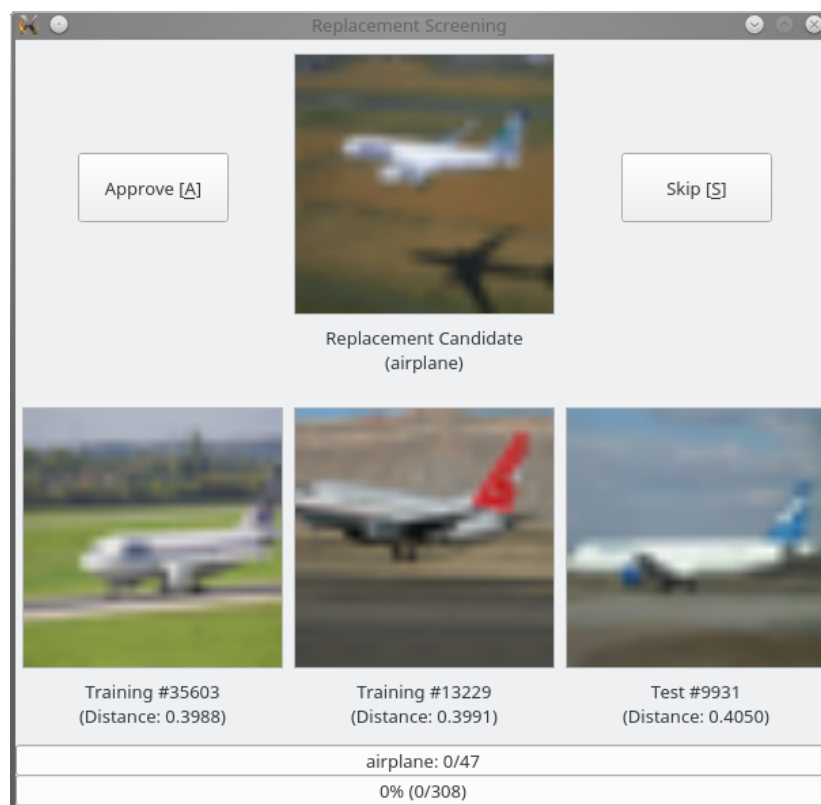

Figure 4. GUI for replacement candidate selection.

Table 2. Classification error rate of various CNN architectures on the original CIFAR test sets and the modified ciFAIR test sets. The best value in each column is highlighted in bold font.

Model	CIFAR-10				CIFAR-100			
	CIFAR	ciFAIR	Gap	Rel. Gap	CIFAR	ciFAIR	Gap	Rel. Gap
Plain-11 [8]	5.91%	6.43%	0.52%	8.80%	27.82%	31.34%	3.52%	12.65%
ResNet-110 [2]	5.26%	5.77%	0.51%	9.70%	26.05%	29.24%	3.19%	**12.25%**
WRN-28-10 [3]	3.89%	4.25%	0.36%	9.25%	18.95%	21.48%	2.53%	13.35%
DenseNet-BC (L=190, k=40) [5]	3.90%	4.20%	**0.30%**	**7.69%**	18.62%	21.02%	2.40%	12.89%
ResNeXt-29 (8x64d) [4]	**3.56%**	**3.95%**	0.39%	10.96%	18.38%	20.84%	2.46%	13.38%
PyramidNet-272-200 [6]	3.58%	4.00%	0.42%	11.73%	**17.05%**	**19.38%**	**2.33%**	13.67%

To answer these questions, we re-evaluate the performance of several popular CNN architectures on both the CIFAR and ciFAIR test sets. Unfortunately, we were not able to find any pre-trained CIFAR models for any of the architectures. Thus, we had to train them ourselves, so that the results do not exactly match those reported in the original papers. However, we used the original source code, where it has been provided by the authors, and followed their instructions for training (i.e., learning rate schedules, optimizer, regularization, etc.).

The results are given in Table 2. Besides the absolute error rate on both test sets, we also report their difference ("gap") in terms of absolute percent points, on the one hand, and relative to the original performance, on the other hand.

On average, the error rate increases by 0.41 percent points on CIFAR-10 and by 2.73 percent points on CIFAR-100. The relative difference, however, can be as high as 12%. The ranking of the architectures did not change on CIFAR-100, and only Wide ResNet and DenseNet swapped positions on CIFAR-10.

6. Conclusions

In a laborious manual annotation process supported by image retrieval, we have identified a startling number of duplicate images in the CIFAR test sets that also exist in the training set. We have argued that it is not sufficient to focus on exact pixel-level duplicates only. In contrast, slightly modified variants of the same scene or very similar images bias the evaluation as well, since these can easily be matched by CNNs using data augmentation, but will rarely appear in real-world applications. We, hence, proposed and released a new test set called ciFAIR, where we replaced all those duplicates with new images from the same domain.

A re-evaluation of several state-of-the-art CNN models for image classification on this new test set led to a significant drop in performance, as expected. In fact, even simple models such as ResNet-110 classified almost all the near-duplicate images perfectly, achieving error rates on this subset that are an order of magnitude better than on the remaining non-duplicate images. This observation underpins that the possibility of memorizing the training data is heavily exploited by state-of-the-art CNN architectures. In combination with the high fraction of near-duplicates in the test set (as high as 10% on CIFAR-100), this effect has the potential to bias the evaluation and development of deep learning approaches for image recognition substantially.

A positive result, on the other hand, is that the relative ranking of the models did not change considerably. Thus, the research efforts of the community do not seem to have overfitted to the presence of duplicates in the test set yet: CNN architectures performing better than others on the original test still provide higher accuracy on the new ciFAIR test set. However, all models we tested have sufficient capacity to memorize the complete training data. In the past, the high fraction of near-duplicates in the test data might, hence have prevented the publication of less over-parameterized approaches that are computationally more efficient and might generalize just as well as other models, but fail to memorize the duplicates.

We encourage all researchers training models on the CIFAR datasets to evaluate their models on ciFAIR, which will provide a better estimate of how well the model generalizes to new data. To facilitate comparison with the state-of-the-art further, we maintain a community-driven leaderboard at https://cvjena.github.io/cifair/, where everyone is welcome to submit new models. We will only accept leaderboard entries for which pre-trained models have been provided, so that we can verify their performance.

Author Contributions: Björn Barz was responsible for conceptualizing this study, creating the mentioned software for near-duplicate search, annotating duplicate candidate pairs, compiling the new ciFAIR test sets, conducting the experiments, and writing the manuscript. Prof. Dr. Joachim Denzler contributed by supervising the work, proof-reading and editing the manuscript, and acquiring funding. All authors have read and agreed to the published version of the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252.
2. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
3. Zagoruyko, S.; Komodakis, N. Wide Residual Networks. In *British Machine Vision Conference (BMVC)*; Richard C., Wilson, E.R.H., Smith, W.A.P., Eds.; BMVA Press: Guildford, UK, 2016; pp. 87.1–87.12. doi:10.5244/C.30.87.
4. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated Residual Transformations for Deep Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5987–5995.
5. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
6. Han, D.; Kim, J.; Kim, J. Deep Pyramidal Residual Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6307–6315.
7. Real, E.; Aggarwal, A.; Huang, Y.; Le, Q.V. Regularized Evolution for Image Classifier Architecture Search. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), Honolulu, HI, USA, 27 January – 1 February 2019; pp. 4780–4789.
8. Barz, B.; Denzler, J. Deep Learning is not a Matter of Depth but of Good Training. In Proceedings of the International Conference on Pattern Recognition and Artificial Intelligence (ICPRAI), Montreal, QC, Canada, 14–17 May 2018; pp. 683–687.
9. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Lake Tahoe, NV, USA, 3–8 December 2012; pp. 1097–1105.
10. Sun, C.; Shrivastava, A.; Singh, S.; Gupta, A. Revisiting Unreasonable Effectiveness of Data in Deep Learning Era. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 843–852.
11. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009; pp. 248–255.
12. Krizhevsky, A.; Hinton, G. *Learning Multiple Layers of Features from Tiny Images*; Technical report; University of Toronto: Toronto, ON, Canada, 2009.

13. Huiskes, M.J.; Lew, M.S. The MIR Flickr Retrieval Evaluation. In Proceedings of the MIR '08: Proceedings of the 2008 ACM International Conference on Multimedia Information Retrieval, New York, NY, USA, 30–31 October 2008; ACM: New York, NY, USA, 2008.

14. Wu, B.; Chen, W.; Fan, Y.; Zhang, Y.; Hou, J.; Huang, J.; Liu, W.; Zhang, T. Tencent ML-Images: A Large-Scale Multi-Label Image Database for Visual Representation Learning. In IEEE Access **2019**, *7*, 172683–172693.

15. Wah, C.; Branson, S.; Welinder, P.; Perona, P.; Belongie, S. *The Caltech-UCSD Birds-200-2011 Dataset*; Technical Report CNS-TR-2011-001; California Institute of Technology: Pasadena, CA, USA, 2011.

16. Jaderberg, M.; Simonyan, K.; Zisserman, A.; Kavukcuoglu, K. Spatial Transformer Networks. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Montreal, Canada, 7–12 December 2015; pp. 2017-2025.

17. Torralba, A.; Fergus, R.; Freeman, W.T. 80 Million Tiny Images: A Large Data Set for Nonparametric Object and Scene Recognition. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **2008**, *30*, 1958–1970.

18. Recht, B.; Roelofs, R.; Schmidt, L.; Shankar, V. Do CIFAR-10 Classifiers Generalize to CIFAR-10? *arXiv* **2018**, arXiv:1806.00451.

19. Fellbaum, C. *WordNet*; Wiley Online Library: Hoboken, NJ, USA, 1998.

20. Smeulders, A.W.; Worring, M.; Santini, S.; Gupta, A.; Jain, R. Content-based Image Retrieval at the End of the Early Years. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **2000**, *22*, 1349–1380.

21. Babenko, A.; Slesarev, A.; Chigorin, A.; Lempitsky, V. Neural Codes for Image Retrieval. In Proceedings of the European Conference on Computer Vision (ECCV), Zürich, Switzerland, 6–12 September 2014; pp. 584–599.

22. Babenko, A.; Lempitsky, V. Aggregating Local Deep Features for Image Retrieval. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1269–1277.

23. Husain, S.S.; Bober, M. Improving Large-Scale Image Retrieval Through Robust Aggregation of Local Descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1783–1796.

24. Revaud, J.; Almazan, J.; Rezende, R.S.; Souza, C.R.d. Learning With Average Precision: Training Image Retrieval With a Listwise Loss. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Seoul, Korea, 27–28 October 2019; pp. 5107–5116.