An Exploration of Consistency Learning with Data Augmentation

Connor Shorten
Florida Atlantic University
cshorten2015@fau.edu

Taghi M. Khoshgoftaar
Florida Atlantic University
khoshgof@fau.edu

Abstract

Deep Learning has achieved remarkable success with Supervised Learning. Nearly all of these successes require very large manually annotated datasets. Data augmentation has enabled Supervised Learning with less labeled data, while avoiding the pitfalls of overfitting. However, Supervised Learning still fails to be Robust, making different predictions for original and augmented data points. We study the addition of a Consistency Loss between representations of original and augmented data points. Although this offers additional structure for invariance to augmentation, it may fall into the trap of representation collapse. Representation collapse describes the solution of mapping every input to a constant output, thus cheating to solve the consistency task. Many techniques have been developed to avoid representation collapse such as stopping gradients, entropy penalties, and applying the Consistency Loss at intermediate layers. We provide an analysis of these techniques in interaction with Supervised Learning for the CIFAR-10 image classification dataset. Our consistency learning models achieve a 1.7% absolute improvement on the CIFAR-10 original test set over the supervised baseline. More interestingly, we are able to dramatically reduce our proposed Distributional Distance metric with the Consistency Loss. Distributional Distance provides a more fine-grained analysis of the invariance to corrupted images. Readers will understand the practice of adding a Consistency Loss to improve Robustness in Deep Learning.

Introduction

Data Augmentation has primarily been utilized to prevent overfitting when training Deep Neural Networks in Supervised Learning (Shorten and Khoshgoftaar 2019, Shorten et al. 2021, Cubuk et al. 2020). Modern Deep Neural Networks typically contain between 50 million and 5 billion parameters (Kaplan et al. 2020). These highly overparameterized models rely on large datasets to learn from. Without large datasets, they will learn functions with very high variance that do not generalize from the training set to unseen test points. Data Augmentation has been used to regularize the size of the training set such as to avoid learning spurious correlations between inputs and their labels (Geirhos et al. 2020). Data Augmentation commonly operates by constructing artificial points (x', y) from original (x,y) points. Here the x' denotes a transformation of the original x. The key in this formulation is that the augmented and original example share the same annotated y label.

Data Augmentation is one of the best strategies to improve the generalization of Deep Neural Networks. The standard Data Augmentation algorithm is to append the artificial examples to the original dataset. The artificial examples update the network with the same learning rule as the original data. We explore the use of a Consistency Loss between the artificial and original examples. The added loss provides more guidance for training Deep Neural Networks, particularly emphasizing invariance to label-preserving corruptions. Invariance describes making identical predictions of semantically similar inputs. Rather than solely looking at improvements on the original test set and preventing overfitting, we focus on achieving invariance to corruptions used in Data Augmentation.

Robustness is another term used in the literature to characterize invariance to corruptions (Hendrycks and Dietterich 2019). Robustness covers a wide range of tests based on how the corruption is sampled. We study the relatively simple test of using a label-preserving transformation to corrupt an original test set. For future work, Robustness may include adversarial examples in which the corruption is optimized with a gradient ascent algorithm (Hendrycks and Dietterich 2019, Dong et al. 2020). In the larger picture, Robustness also covers Distribution Shift. Distribution Shift is the phenomenon in which the test set is sampled from a different data distribution than the training set. Koh et al. collect several examples of real-world distribution shifts in the WILDS benchmark (Koh et al. 2021). These shifts range from dealing with a shift in lighting conditions across hospitals (Bandi et al. 2018), to novel locations for wildlife monitoring cameras (Beery et al. 2020), and many more. Distribution Shift may also include Domain Generalization, such as sentiment analysis trained with IMDB movie reviews and tested on Amazon product reviews (Ni et al. 2019), or image classification trained with sketch drawings and tested on photorealistic images (Peng et al. 2019).

Training Deep Neural Networks with an added Consistency Loss has seen a surge in interest. This has mostly been in application to Unsupervised Learning. In order to avoid collecting and annotating new (x,y) pairs for Supervised Learning, researchers have instead relied on Data Augmentation. As mentioned previously, Data Augmentation can create several (x',y) points from an original (x,y) annotation. In Unsupervised Learning, there are no ground-truth y annotations. However, Data Augmentation enables supervision based on the criteria that an original example and its derived augmented sample are semantically equivalent. For example, take an unlabeled image that would be annotated as a "dog" if the manual labor to do so was available. This image would still be a "dog" after being rotated 10 degrees or horizontally flipped. We note that this notion of semantic equivalence may require more care for different downstream applications. For example, downstream Computer Vision tasks such as bounding box detection (Carion et al. 2020) or keypoint estimation (Zheng et al. 2021), require more care in determining label equivalence post-augmentation, compared to image classification, which is the focus of this paper.

Most Contrastive Self-Supervised Learning algorithms additionally use semantic dissimilarity in addition to similarity comparisons. This is added to prevent representation collapse. Representation collapse describes the phenomenon where the model maps every input to a constant output. This trivial solution solves the consistency task, although the learned data representation is useless. Contrastive Self-Supervised Learning algorithms typically avoid representation collapse by using large negative batches to normalize the Consistency Loss (Chen et al. 2020, He et al. 2020).

We look to other solutions to avoid representation collapse. Large batch sizes limit the size and training time of our models. Firstly, we find that the added Supervised Loss term is strong enough to avoid collapse, depending on the alpha weighting the two loss functions. Secondly, we explore dividing the Supervised and Consistency Losses to operate at logit and vector representation levels, respectively. The vector consistency model achieves better accuracy on the original CIFAR-10 test set, although the logit consistency model is better with respect to our Distributional Distance score. Finally, we preview several modifications we intend to build on in future work. These include stopping gradients, entropy regularization, and multi-stage, rather than multi-task, learning.

Our primary contributions are as follows:

- We provide empirical results of an added Consistency Loss with Supervised Learning for CIFAR-10 image classification.
- From these results, we draw novel conclusions about the sensitivity of a multi-task loss weighting, vector similarity scoring, gradient stopping, and entropy regularization.
- We propose a novel Distributional Distance metric as a proxy for invariance in addition to corrupted test set accuracy.

Related Work

Data Augmentation

Our work is primarily related to Data Augmentation. Data Augmentation has been a very useful technique to prevent overfitting (Krizhevsky et al. 2012), and is increasingly being used in learning without manual labeling (Chen et al. 2020, He et al. 2020). Our work is similar to research in controlling the application of Data Augmentation. Cubuk et al. pioneered this work with AutoAugment, optimizing hyperparameters with respect to Data Augmentation through a controller trained with Reinforcement Learning (Cubuk et al. 2019). Cubuk et al. further simplified the problem of Augmentation optimization with RandAugment (Cubuk et al. 2019), which we use heavily in our experiments.

Contrastive Learning

Our work is also highly related to Contrastive Learning. Contrastive Learning has been especially useful in Unsupervised Learning, such as Unsupervised Data Augmentation (Xie et al. 2019), as well as Self-Supervised Learning, such as SimCLR (Chen et al. 2020) and MoCo (He et al. 2020). Our work is more closely related to Supervised Contrastive Learning from Khosla et al. (Khosla et al. 2020). These authors are looking to merge the information in manual labeling with contrastive loss functions. Differently from Isola et al., we separate the loss functions into a multi-task learning framework and do not use negative regularization.

Robustness

Consistency Learning with Augmented Data is designed to improve Robustness in Deep Learning. Robustness is one of the largest outstanding limitations of Deep Learning and has been covered by many works. We were particularly inspired by the findings of Hendrycks and Dietterich (Hendrycks and Dietterich 2019), showing that not only are Deep Neural Networks vulnerable to adversarial attacks, but common corruptions as well. Our work is related to approaches such as AugMix (Hendrycks et al. 2020), AugMax (Wang et al. 2021), and DAIR (Huang et al. 2021) that use Data Augmentation to achieve more robust Deep Learning models.

Methodology

The primary objective of these experiments is to see the improvement of Consistency Learning with Data Augmentation. More particularly, we study performance on image classification. Image classification is a common task in Computer Vision in which tensor representations of images made up of RGB pixel matrices are assigned categorical class labels. We experiment with the CIFAR-10 academic benchmark for image classification (Krizhevsky 2009).

In the following equations, we pass predictions into a KL-divergence loss function, abbreviated as KL. This is a common technique to measure the distance between probability distributions, such as logit representations of image labels. We subscript the predicted ys to denote how they have been augmented. In this case, we only subscript with RandAug (Cubuk et al. 2019), but we leave this notation for the generality of targeting generalization to a specific augmented distribution. For example, we may want to compare the consistency to Rotation-augmented images in future work, or particular domains for achieving Domain Generalization. Our predicted ys without any subscript denotes that the input image has not been augmented. Equation 1 illustrates a standard Supervised Learning loss function. Equation 2 illustrates the added Consistency Loss between original and aug-

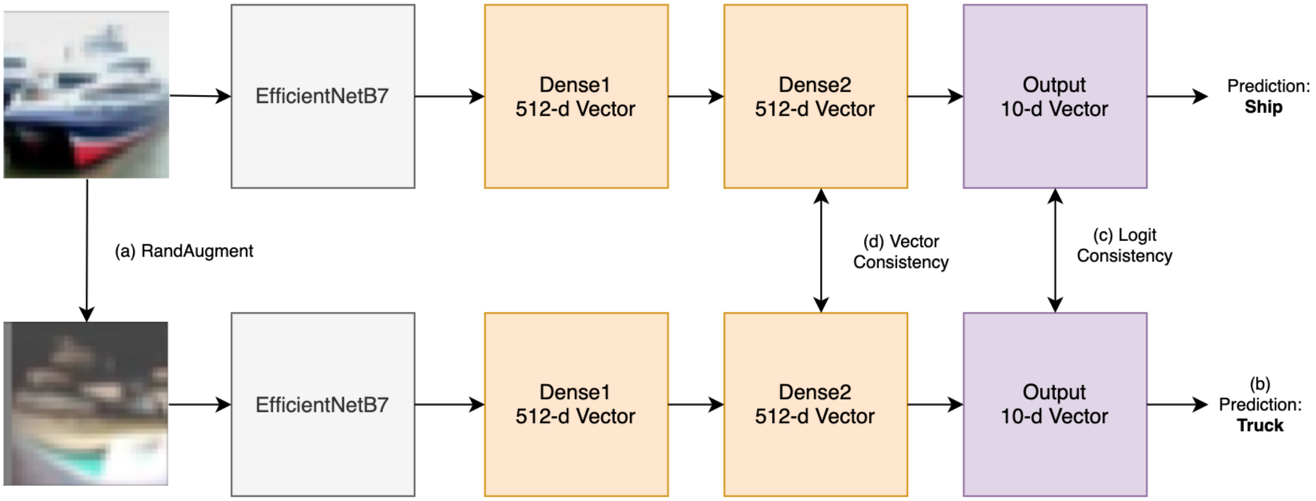

Figure 1: Illustration of the Consistency Loss framework we explore in this study. Section (a) visualizes the RandAugment transformation of an original image. Section (b) denotes the forward pass of a Deep Neural Network. In our experiments, we mostly use the EfficientNetB7 (Tan and Le 2019) architecture to process images. The representation from EfficientNetB7 is then passed through 3 fully connected layers. Section (c) references the logit representation of a predicted image. The argmax of this representation shown in Section (c) is used to determine the prediction, which is denoted in Section (b). Section (d) references an intermediate vector representation from the Neural Network. We explore an added Consistency Loss in representations (c) and (d) in our experiments.

mented images. Equation 3 uses the bar notation to symbolize consistency on vector representations, rather than logit outputs. Equation 4 introduces the [SG], stop-gradient, operator motivated by avoiding representation collapse.

$$Loss = KL(\hat{y}_{RandAug}, y) \tag{1}$$

$$Loss = KL(\hat{y}_{RandAug}, y) + \alpha * KL(\hat{y}, \hat{y}_{RandAug}) \tag{2}$$

$$Loss = KL(\hat{y}_{RandAug}, y) + \alpha * KL(\overline{y}, \overline{y}_{RandAug}) \tag{3}$$

$$Loss = KL(\hat{y}_{RandAug}, y) + \alpha * KL(\hat{y}, \hat{y}_{RandAug}[SG]) \tag{4}$$

Distributional Distance

We propose a novel contribution of using Distributional Distances to measure invariance to augmentation. This metric can be generalized to arbitrary notions of semantic equivalence. Distributional Distance is calculated by vectorizing the entire dataset and then averaging the vector distances produced by the compared models. Vectorizing the dataset refers to iterating through each instance and running a forward pass to compress the high-dimensional inputs into intermediate vector representations. These vectors can be stored on the disk for comparison, or distance can be computed online to save memory. Intermediate vector representations produced by Deep Neural Networks are illustrated in Figure 1(d). We use L1 distance in our experiments for the sake of simplicity and leave angular, cosine, or hamming vector distances for future work. The objective of computing Distributional Distance is to drill deeper into the data representation, rather than just the final prediction. We use the DD(A, B) notion to symbolize the Distributional Distance of A and B. Distributional Distance is a symmetric metric, meaning that DD(A, B) is equivalent to DD(B, A). In our experiments, we compute Distributional Distance at the logit level.

Our models are explicitly trained to minimize the Distributional Distance between the original CIFAR-10 data and the RandAugment-transformed CIFAR-10 data. We additionally compute the distance to a Rotate-transformed CIFAR-10 set for the sake of comparison. Ideally, learning invariance to RandAugment should generalize to Rotation and other augmentations. However, we do not achieve that goal with our Consistency Loss. We additionally report the Supervised test accuracy as another proxy for performance, although the emphasis of this paper is Robustness and learning to be consistent with label-preserving augmentations.

Experimental Results

Consistency Learning

Our experiments begin with the training objectives shown in Equations 1 and 2. This describes the Supervised Learning task, as well as a Multi-Task objective between label- and consistency-based loss function. The results of this experiment are shown in Table 1. The added Consistency Loss improves the absolute performance on the original test set

Metric	SL	SL + CL
Original Train	98.9%	**99.5%**
Original Test	86.2%	**86.9%**
DD(Original, RandAugment)	0.341	**0.275**
DD(Original, Rotate)	0.834	**0.818**
DD(Rotate, RandAugment)	0.882	**0.861**
RandAugment Train	93.0%	**97.9%**
RandAugment Test	80.6%	**83.1%**
Rotate Train	61.3%	**62.7%**
Rotate Test	56.4%	**57.1%**

Table 1: The added Consistency Loss improves the Supervised Learning baseline on every metric we consider (highlighted in bold). We find a very encouraging reduction in Distributional Distance from the original data to RandAugment-transformed data. Reducing this metric from 0.341 to 0.275 signals that the model makes much more similar predictions on RandAugment-transformed images with the added Consistency Loss.

Metric	1e-2	5e-3	1e-3
Original Train	96.1%	99.5%	**99.8%**
Original Test	84.1%	**86.9%**	86.1%
DD(Original, RandAugment)	0.386	**0.275**	0.305
DD(Original, Rotate)	0.852	**0.818**	0.838
DD(Rotate, RandAugment)	0.902	**0.861**	0.89
RandAugment Train	87.9%	**97.9%**	96.8%
RandAugment Test	78.1%	**83.1%**	81.6%
Rotate Train	58.9%	**62.7%**	61.8%
Rotate Test	54.5%	**57.1%**	55.9%

Table 2: Bolded values denote superior metric results. Unfortunately, the added Consistency Loss shows a sensitivity to the hyperparameter controlling the relative weighting of the Consistency and Supervised Learning losses. This table is showing a monotonically decreasing weighting from 0.01 (1e-2) to 0.001 (1e-3). We leave it to future work to explore hyperparameter optimization and dynamic loss scaling.

by 0.7%. We find further improvements on the original test set with the vector consistency described later on. More interestingly, the Distributional Distance between the original and RandAugment test sets are reduced from 0.341 to 0.275 (approximately a 20% relative decrease). This highlights the effectiveness of using a Consistency Loss to achieve invariant predictions with augmentations.

Weighting the Consistency Loss

We find hyperparameter sensitivity manifested in the Multi-Task weighting in Equation 2. Table 2 illustrates the results of monotonically decreasing alpha weightings from left to right. We see a significant variance in the performance of models depending on this hyperparameter. In future work, we intend to explore a curriculum of scheduling this loss weighting. For example, in the beginning of training the representations are very different and the loss has a significantly higher magnitude than the supervised update. However, as training progresses this loss becomes much more manageable and the small loss weighting is limiting the contribution of the loss towards convergence.

Components of Consistency Learning

In this section, we report our findings from exploring the objectives shown in Equations 2 to 4. These objectives are primarily designed to avoid representation collapse. We are also motivated by the exploration of alternative constructions of the Consistency Loss. We find interesting differences in the performance of vector versus logit consistency, sensitivity to the loss weighting, and negative results with stopping gradients.

Vector Consistency The following experiment applies the Consistency Loss at the vector, rather than logit, representation. This is shown in Equation 3, using the bar versus hat notation to communicate vector versus logit representations, respectively. The concept of intermediate vector representations is also illustrated in Figure 1(d). The vector representation has more dimensions of comparison, which makes the Consistency Loss a more challenging task. The magnitude of this loss is much larger than the Supervised Loss. To mitigate this, we weight this loss much smaller than the logit Consistency Loss at 1e-5. Our best performing logit model was weighted at 5e-3 (2x larger). The Supervised Loss is only applied at the logit level, which has another interesting property of dividing up the layers of supervision. Shown in Table 3, we find an exciting improvement on the original CIFAR-10 test set by applying the Consistency Loss at the vector level. However, we do not improve the Distributional Distance over the logit consistency model. These results are biased by calculating Distributional Distance at the logit level. We think it is best to achieve logit-level invariance before drilling deeper into the representation such as these intermediate vectors.

Multi-Stage Training We are mostly interested in applying the Consistency Loss in a Multi-Task framework with the Supervised Loss. We also consider Multi-Stage training, also known as Fine-Tuning. Fine-Tuning is a very common procedure in Transfer Learning, in which a model is first trained on one task and then trained on another task. The key distinction is that the model only applies one of the loss functions at a time. We continue training the Supervised Learning model from Table 1 with 300 epochs of the added Consistency Loss. As shown in Table 4, We see an improvement over the Supervised Learning model, but this falls short of the Multi-Task model.

Stopping Gradients As shown in Eq. 4, we stop the gradients applied to one of the predictions in the Consistency Loss. More particularly, we stop the gradients on the prediction of the RandAugment-transformed image in the Consistency Loss. This prediction still contributes to the overall loss in the Supervised Learning update, but not the Consistency Loss. By stopping the gradient in the second predic-

Metric	Vector CL	Logit CL
Original Train	99.4%	**99.5%**
Original Test	**88.1%**	86.9%
DD(Original, RandAugment)	0.304	**0.275**
DD(Original, Rotate)	**0.803**	0.818
DD(Rotate, RandAugment)	**0.857**	0.861
RandAugment Train	94.6%	**97.9%**
RandAugment Test	**83.2%**	83.1%
Rotate Train	**63.2%**	62.7%
Rotate Test	**59.2%**	57.1%

Table 3: Comparison of applying the Consistency Loss at the vector versus logit representation. We find promising results applying the loss at the vector representation. Not surprisingly, the logit model is better at Distributional Distance calculated at the logit level. This may be an important detail depending on the downstream application such as robustness to logit-level predictions versus semantic similarity based on vector distance.

Metric	SL	SL, then CL	SL + CL
Original Train	98.9%	**99.9%**	99.5%
Original Test	86.2%	86.5%	**86.9%**
DD(Original, RandAugment)	0.341	0.307	**0.275**
DD(Original, Rotate)	0.834	0.839	**0.818**
DD(Rotate, RandAugment)	0.882	0.881	**0.861**
RandAugment Train	93.0%	97.8%	**97.9%**
RandAugment Test	80.6%	81.7%	**83.1%**
Rotate Train	61.3%	62.5%	**62.7%**
Rotate Test	56.4%	56.0%	**57.1%**

Table 4: Comparing Multi-Stage to Multi-Task Learning. Multi-Stage Learning is a common protocol in Transfer Learning. We might view Consistency Learning as a fine-tuning step after Supervised Learning, however, we do not find promising results with this technique.

Metric	Stopped Gradients	Baseline CL
Original Train	98.6%	**99.5%**
Original Test	85.6%	**86.9%**
DD(Original, RandAugment)	0.341	**0.275**
DD(Original, Rotate)	0.88	**0.818**
DD(Rotate, RandAugment)	0.927	**0.861**
RandAugment Train	92.1%	**97.9%**
RandAugment Test	81.1%	**83.1%**
Rotate Train	58.6%	**62.7%**
Rotate Test	53.9%	**57.1%**

Table 5: Results of stopping gradients in the second term of the Consistency Loss. We do not find promising results for this in our experiments.

tion, we are only updating the original representation to be more like the frozen augmented counterpart. Allowing both gradients may fall into optimization troubles as they simultaneously try to converge to each other. However, as shown in Table 5, we do not find this to be the case. In our experiments, stopping the gradients and using Eq. 4 does not improve over using both gradients in Eq. 2. This is likely due to the additional regularization of the Supervised Learning loss. In future work we intend to explore an exponential moving average (EMA) of the RandAugment prediction, as well as the use of a Teacher's prediction for the RandAugment counterpart in a Knowledge Distillation framework.

Conclusion

In conclusion, we have presented empirical results and considerations for the use of Consistency Learning with augmented data. This is a new way to utilize augmented data in contrast to Supervised Learning updates which have the same treatment for augmented and unaugmented data. We have further proposed Distributional Distance as a strategy to measure the invariance of Deep Neural Networks. Distributional Distance enables us to look beyond the test accuracy or beyond similarly misclassified examples and see exactly how far augmentations change data representations. We have explored many additional techniques as well such as the importance of the Multi-Task loss weighting, applying Consistency Learning to intermediate representations, Multi-Stage versus Multi-Task Learning, and considerations with entropy regularization. We hope that this work inspires future interest in using Consistency Learning to achieve Robustness in Deep Learning.

References

Bandi P., Geessink O., Manson Q., Dijk M. V., et al. 2018. From Detection of Individual Metastases to Classification of Lymph Node Status at the Patient Level: The CAMELYON17 Challenge. In IEEE Transactions on Medical Imaging.

Beery S., Cole E., and Gjoka A. 2020. The iWildCam 2020 Competition Dataset. In arXiv:2004.10340.

Carion N., Massa F., Synnaeve G., Usunier N., Kirillov A., and Zagoruyko S. 2020. End-to-End Object Detection with

Transformer. In arXiv:2005.12872.

Chen T., Kornblith S., Norouzi M., and Hinton G. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In International Conference on Machine Learning.

Cubuk E. D., Zoph B., Shlens J., and Le Q. V. 2020. RandAugment: Practical Automated Data Augmentation with a Reduced Search Space. In Neural Information Processing Systems.

Cubuk E. D., Zoph B., Mane D., Vasudevan V., Le Q. V. 2019. AutoAugment: Learning Augmentation Policies from Data. In IEEE Conference on Computer Vision and Pattern Recognition.

Dong Y., Fu Q., Yang X., Pang T., Su H., Xiao Z., and Zhu J. 2020. Benchmarking Adversarial Robustness on Image Classification. In IEEE Conference on Computer Vision and Pattern Recognition.

Geirhos R., Jacobsen J., Michaelis C., Zemel R., Brendel W., Bethge M., Wichmann F. A. 2020. Shorcut learning in deep neural networks. In Nature Machine Intelligence.

Goodfellow I. J., Shlens J., and Szegedy C. 2015. Explaining and Harnessing Adversarial Examples. In arXiv:1412.6572.

He K., Fan H., Wu Y., Xie S., and Girshick R 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In IEEE Conference on Computer Vision and Pattern Recognition.

Hendrycks D., Mu N., Cubuk E. D., Zoph B., Gilmer J., Lakshminarayanan B. 2020. AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainity. In International Conference of Learning Representation.

Hendrycks D. and Dietterich T. 2019. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In International Conference on Learning Representations.

Hinton G., Vinyals O., and Dean J. 2015. Distilling the Knowledge in a Neural Network. In Neural Information Processing Systems.

Huang T., Halbe S., Sankar C., Amini P., Kottur S., Geramifard A., Razaviyayn M., and Beirami A. 2021. DAIR: Data Augmented Invariant Regularizaation. In arXiv:2110.11205.

Jung A. B., Wada K., Crall J., Tanaka S., et al. 2020. Imgaug. Url: https://github.com/aleju/imgaug, accessed February 1st, 2020.

Kaplan J., McCandlish S., Henighan T., Brown T. B., Chess B., Child R., Gray S., Radford A., Wu J., and Amodei D. 2020. Scaling Laws for Neural Language Models. In arXiv:2001.08361.

Khosla P., Teterwak P., Wang C., Sarna A., Tian Y., Isola P., Maschinot A., Liu C., and Krishnan D. 2020. Supervised Contrastive Learning. In Neural Information Processing Systems.

Koh P. W., Sagawa S., Marklund H., Xie S. M., Zhang M., et al. 2021. WILDS: A Benchmark of in-the-Wild Distribution Shifts. In arXiv:2012.07421.

Krrizhevsky A. 2009. Learning Multiple Layers of Features from Tiny Images.

Krizhevsky A., Sutskever I., and Hinton G. E. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In Neural Information Processing Systems.

Ni J., Li J., and McAuley J. 2019. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. In Empirical Methods in Natural Language Processing.

Peng X., Bai Q., Xia X., Huang Z., Saenko K., and Wang B 2019. Moment Matching for Multi-Source Domain Adaptation. In International Conference on Computer Vision.

Shorten C. and Khoshgoftaar T. M. 2019. A survey on Image Data Augmentation for Deep Learning. In Journal of Big Data.

Shorten C., Khoshgoftaar T. M., and Furht B. 2021. Text Data Augmentation for Deep Learning. In Journal of Big Data.

Tan M. and Le Q. V. 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In International Conference on Machine Learning.

Wang H., Xiao C. Kossaifi J., Yu Z., Anandkumar A., and Wang Z 2021. AugMax: Adversarial Composition of Random Augmentations for Robust Training. In Neural Information Processing Systems.

Xie Q., Dai Z., Hovy E., Luong M., Le Q. V. 2019. Unsupervised Data Augmentation for Consistency Training. In Neural Information Processing Systems.

Zheng C., Wu W., Yang T., Zhu S., Chen C., Liu R., Shen J., Kehtarnavaz N., and Shah M. 2021. Deep Learning-Based Human Pose Estimation: A Survey. In arXiv:2012.13392.

Zheng C., Wu W., Yang T., Zhu S., Chen C., Liu R., Shen J., Kehtarnavaz N., and Shah M. 2021. Deep Learning-Based Human Pose Estimation: A Survey. In arXiv:2012.13392.