Test-Time Adaptation with Principal Component Analysis

Thomas Cordier, Victor Bouvier, Gilles Hénaff, Céline Hudelot

Test-Time Adaptation with Principal Component Analysis

Thomas Cordier[1,2][0000−0002−1314−9619], Victor Bouvier[3], Gilles Hénaff[1], and Céline Hudelot[2]

[1] Thales Land and Air Systems, 2 Avenue Gay-Lussac, 78990 Elancourt, France
[2] Université Paris-Saclay, CentraleSupélec, Mathématiques et Informatique pour la Complexité et les Systèmes, 91190, Gif-sur-Yvette, France
[3] Dataiku, 203 Rue de Bercy, 75012, Paris, France

Abstract. Machine Learning models are prone to fail when test data are different from training data, a situation often encountered in real applications known as distribution shift. While still valid, the training-time knowledge becomes less effective, requiring a test-time adaptation to maintain high performance. Following approaches that assume batch-norm layer and use their statistics for adaptation[19], we propose a Test-Time Adaptation with Principal Component Analysis (TTAwPCA), which presumes a fitted PCA and adapts at test time a spectral filter based on the singular values of the PCA for robustness to corruptions. TTAwPCA combines three components: the output of a given layer is decomposed using a Principal Component Analysis (PCA), filtered by a penalization of its singular values, and reconstructed with the PCA inverse transform. This generic enhancement adds fewer parameters than current methods[17, 25, 27]. Experiments on CIFAR-10-C and CIFAR-100-C[8] demonstrate the effectiveness and limits of our method using a unique filter of 2000 parameters.

Keywords: Robustness · Test-time Adaptation · Principal Component Analysis · Filtering.

1 Introduction

Deep neural networks are optimized to achieve high accuracy on their training distribution, given the hypothesis that they will be deployed on the same distribution during inference. However, distribution shift occurs in many industrial applications, for instance, when a sensor malfunctions. The accuracy of a predictive task drops as the distribution of test data shifts [8, 21]. Domain adaptation prevents such failures by jointly training on source and target data. Instead, Test-time adaptation mitigates the domain gap either by test-time training or fully test-time adaptation according to the availability of source data. Test-time training augments the training objective on source data with an unsupervised task that remains at test time to optimize domain-invariant representations. *Fully test-time adaptation* [27] does not alter training and only needs testing observations and a pre-trained model for privacy, applicability, or profit [3].

To enhance generalization, Spectral regularization [2] especially for GANs [16] and L^2-regularization are standard tools during training [20]. L^2-regularization reduces model variance for different potential training sets and constrains the model complexity by lowering the weights of its layers. Spectral normalization penalizes the weight matrices by their largest singular value to ensure the Lipschitz continuity of the neural network.

Taking inspiration from these previous works, we aim to learn the best fitting parameters of a spectral filter on a corrupted dataset without supervision. We introduce TTAwPCA, which projects a batch of inputs onto a spectral basis, filters the projected data points, and reconstructs the filtered batch. As [27], we minimize entropy to learn the parameters of the filter. This generic unsupervised learning loss makes few assumptions about the data.

In this paper, we first overview state-of-the-art test-time adaptation (Sec. 2). Then, we introduce a simple yet effective method: TTAwPCA (Sec. 3). We demonstrate its effectiveness experimentally in tackling corrupted data (Sec. 4 and we discuss our results compared with other methods (Sec. 5).

2 Related work

Unsupervised Domain Adaptation jointly adapts on source and target domain through transduction, thus requiring both simultaneously. Several properties have been optimized: cross-domain feature alignment[7, 1, 21], adversarial invariance[26, 5, 6, 9], and shared proxy tasks [24] such as predicting rotation and position. In our work, we want to use only the target domain at test time.

Test-time adaptation indicates methods tackling the domain gap during inference. *TTT*[25] augments the supervised training objective with a self-supervised loss using source data. Only the self-supervised loss keeps adapting at test time on target domain. It relies on predicting the rotation of inputs, a visual proxy task, but designing suitable proxy tasks can be challenging. Training parameters are altered during training and test-time adaptation. *Test-time batch normalization*[22, 19] allows statistics of batch norm layers to be tracked during the distribution shift at test time. *TENT*[27] exhibits entropy minimization at test time on feature modulators extracted from spatial batch normalization to adapt to distribution shift. Entropy minimization is a generic and standard loss for domain adaptation to penalize classes overlap. Information maximization [12, 23, 10] used by [15, 17] involves entropy minimization and diversity regularization. The diversity regularizer averts collapsed solutions of entropy minimization. *SLR+IT*[17] argues that Information maximization compensates for the vanishing gradient issues of entropy minimization for high confidence predictions. Moreover, an additional trainable network shares the input samples with the tested network to partially correct the domain shift. Principal Component Analysis cuts out noisy eigenvalues to remove uncorrelated noise[14, 18]. In addition, we propose to add fully test-time learnable parameters to reduce the remaining noise of corrupted data onto the spectral basis.

3 Filtering the corrupted Singular Values

Let a neural network f_θ with parameters θ be trained to completion on a source set $X_\mathcal{D}$ of N samples from a distribution $\mathcal{D}$. Parameters θ are thus frozen after training. The initialization of our method takes place before testing. TTAwPCA is added after the jth layer. It consists of a Principal Component Analysis (PCA) and, for now, a pass-through filter. To fit its PCA, the concatenated output $A_{j,\mathcal{D}}$ of the jth layer has to be flattened from the shape N elements of the batch times c channels times the spatial dimensions $h \times w$ to a rectangular matrix of size $N \times p$ where $p = c \cdot h \cdot w$ and then mean normalized. Singular Value Decomposition breaks down the flattened training output $\tilde{A_\mathcal{D}}$ as:

$$A_{j,\mathcal{D}} = U \Lambda V^\top \tag{1}$$

where Λ is an $N \times p$ matrix of singular values, U an $N \times N$ matrix of left singular vectors and V an $p \times p$ matrix of right singular vectors. We define a hyperparameter L such that only the first L singular values are conserved. Note that this operation belongs to the training procedure.

At test time, the filter F_Γ is enabled to optimize its parameters $\Gamma = \{\gamma_i; i \in [0, L-1]\}$ of the corrupted singular values. Let the t-th batch of corrupted observation $x_t \sim D'$ be presented to the model $f_{\theta,\Gamma}$. Let $A_{j,\mathcal{D}',t}$ be the tth batched output of the jth layer. After the flatten operation and the mean normalization, $A_{j,\mathcal{D}',t}$ is projected onto the singular basis vectors by V_L, filtered by F_Γ and reconstructed by $V_L^\top$ as $O_{t,\mathcal{D}'}$ in its original basis:

$$O_{t,\mathcal{D}'} = A_{j,\mathcal{D}',t} V_L F_\Gamma V_L^\top \tag{2}$$

We designed a filter F_Γ related with L^2-regularization as demonstrated in A.1 of diagonal element $F_{i,i}$ based on the singular values Λ_L of the training set and L learning parameters γ_i:

$$F_{i,i}(\gamma_i) = \frac{\lambda_i}{\lambda_{i,i} + \mathrm{ReLU}(\gamma_i)} \tag{3}$$

The ReLU activation assures the stability of the filter.

Similarly, we designed a negative exponential filter F_Γ of diagonal element $F_{i,i}$:

$$F_{i,i}(\gamma_i) = \frac{1}{1 + \exp(\gamma_i^2 - \lambda_i)} \tag{4}$$

We denote this model $f_{\theta,\Gamma}$ composed of f_θ and TTAwPCA. The learning parameters Γ are optimised over the batch x_t using entropy minimization of model prediction $\hat{y}_t = f_{\theta,\Gamma}(x_t)$ as test-time objective.

4 Experiments

Dataset. We classify CIFAR-10-C and CIFAR-100-C [8]. Both test sets contain 10,000 images of CIFAR-10 and CIFAR-100 [13] augmented by 15 common corruptions and five severity levels.

Table 1: Episodic corruption error benchmark on CIFAR-10-C and CIFAR-100-C with the highest severity [in %].

Dataset	Method	Mean	Gauss	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	JPEG
CIFAR-10-C	No Adaptation	43.53	72.33	65.71	72.92	46.94	54.32	34.3	42.02	25.07	41.30	26.01	9.30	46.69	26.59	58.45	30.30
	BN	20.44	28.08	26.12	36.27	12.82	35.28	14.17	**12.13**	17.28	17.39	15.26	8.39	12.63	23.76	19.66	27.30
	TENT	**19.96**	28.05	26.11	36.31	12.80	35.28	14.16	12.14	**17.27**	**17.36**	15.23	8.37	**12.59**	23.77	**19.61**	27.31
	exp-TTAwPCA (ours)	20.35	**25.5**	**23.55**	**33.77**	14.82	35.04	15.24	13.76	17.73	17.43	16.09	8.62	14.58	24.44	20.00	**24.68**
	ReLU-TTAwPCA (ours)	20.42	28.10	25.99	36.13	**12.72**	**34.93**	**14.00**	12.24	17.29	17.8	**15.07**	**8.26**	13.09	**23.47**	19.76	27.41
CIFAR-100-C	No Adaptation	85.54	93.84	93.60	96.63	91.49	92.79	86.51	88.69	70.91	82.30	84.74	47.26	96.30	85.02	89.50	83.49
	BN	36.61	47.21	46.72	55.59	**27.33**	47.75	**28.23**	26.65	32.74	33.63	32.92	**21.35**	29.64	37.79	33.99	47.56
	TENT	**34.56**	**42.91**	**41.94**	**49.76**	28.27	**44.55**	28.75	27.38	**30.99**	**31.59**	**30.72**	21.88	30.81	**35.42**	**31.27**	**42.09**
	exp-TTAwPCA (ours)	37.89	45.92	45.71	54.23	32.82	47.88	31.98	30.04	33.53	35.12	36.26	22.46	32.92	39.18	34.91	45.37
	ReLU-TTAwPCA (ours)	36.62	47.41	46.80	55.50	27.61	47.76	28.28	**26.54**	32.67	33.46	32.80	21.41	**29.55**	37.67	34.25	47.53

Table 2: Online corruption error benchmark on CIFAR-10-C and CIFAR-100-C with the highest severity [in %].

Dataset	Method	Mean	Gauss	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	JPEG
CIFAR-10-C	TENT	**18.57**	**25.09**	**22.76**	**32.71**	**12.01**	**31.88**	**13.25**	**11.12**	**15.9**	**16.32** ± 0.59	**13.82**	**8.21**	**11.66**	**22.02**	**17.29**	**24.5** ± 0.43
	exp-TTAwPCA (ours)	20.28	25.42	23.44	33.92	14.79	34.81	15.18	13.71	17.52	17.53 ± 0.17	16.09	8.62	14.58	24.44	20.00	**24.68 ± 0.12**
	ReLU-TTAwPCA (ours)	20.45	28.14	25.84	36.23	12.85	35.04	14.01	12.22	17.27	17.63	15.08	8.37	13.05	23.58	19.93	27.44
CIFAR-100-C	TENT	**31.7**	**38.74**	**36.88**	**44.00**	**26.91**	**41.03**	**27.33**	**25.54**	**28.18**	**28.85**	**28.03**	**20.44**	**28.81**	**33.93**	**28.41**	**38.41**
	exp-TTAwPCA (ours)	37.89	46.02	45.8	54.15	32.56	47.87	31.91	30.14	33.62	35.19	35.98	22.33	33.08	39.18	34.93	45.51
	ReLU-TTAwPCA (ours)	36.83	47.39	46.82	55.95	27.80	48.30	28.49	26.85	32.92	33.78± 0.20	32.91	21.64	29.58	37.94	34.48	47.59± 0.10

Models. We use the publicly available pre-trained WideResNet-28-10 [28] of RobustBench [4]. We trained a model on CIFAR-100 achieving 83% accuracy on the test set, as Robustbench does not provide one. TTAwPCA is set after the first convolutional layer with only 2000 parameters for our best results on both datasets. We compare our two different filters with TENT [27] and test-time batch statistics updates[22, 19].

Settings. Episodic and online settings describe whether the model is reset after optimization on each batch or after optimization on the corruption at a given severity.

Optimization. We optimize the parameters Γ of the filter by Adam [11] for one step on both offline and episodic fully test-time adaptation settings. We set the batch size at 200 samples and the learning rate at 0,001. $L = 2000$ proved to be sufficient for our method, as shown in A.2.

5 Discussion

TTAwPCA tackles common corruptions [8] by improving the accuracy of each perturbed set. With only the 2000 parameters, TTAwPCA achieves state-of-the-art performance on various corruptions in the episodic CIFAR-10-C setting. Namely: Gaussian Noise, Shot Noise, Impulse Noise, Glass Blur, and JPEG compression for the exponential filter and Defocus Blur, Glass Blur, Motion Blur, Fog, Brightness, and Elastic Transformation for the ReLU filter whereas performing close to TENT [27] on the rest. Our method achieves a better trade-off between accuracy retrieval and the number of parameters. On the other hand,

TTAwPCA does not take advantage of the online setting and does not scale well to CIFAR-100-C. We provide intuitions to explain this observation.

TTAwPCA enables PCA to filter noisy singular values on the remaining dimensions, assuming additive noises increase singular values. However, we observe some corruptions to reduce singular values effectively, thus filtering crucial information to the tested task. A penalizing filter is unable to recover this loss of information. Adding a multiplicative parameter to each diagonal element of our filter became a subject of our interest but was found unstable. To increase stability, we normalized each singular value λ_i by its higher value: λ_0. The instability of the tested filter prevents its convergence in an online setting.

Our results on CIFAR-100-C tend to be underperforming. High similarity between classes of CIFAR-100 might be too complex for TTAwPCA to reach over-parametrized methods such as TENT. A subtle change in the first principal components of the PCA can significantly affect the discriminability of the model if corruption occurs and the classes are too close. The first convolutional layer might not be discriminative enough to perform reliable principal components. On the other hand, the following layers merge the corruption and the features relevant to the task.

We argue that TTAwPCA follows the setting of *Fully test-time adaptation* [27] as TTAwPCA does not change the training objective. TTAwPCA expects a model to have a fitted PCA after completing the training procedure. Equivalently TENT needs spatial batch normalization layers to operate.

Lastly, TTAwPCA is the only method that does not alter any training parameter. Its test-time update can be fully deactivated without reloading the model instead of TENT or batch adaptation at test time (BN). The batch normalization parameters are forgotten through their processes. PCA also offers a linear adaptation of the model.

6 Conclusion

This paper introduced a new layer called TTAwPCA, filtering the singular values to tackle the out-of-distribution shift at test time. This spectral filter, initialized after training, is optimized on the test dataset with a task agnostic loss. We compared the effectiveness of our method in an online and an episodic setting to TENT [27] on CIFAR-10-C and CIFAR-100-C [8]. We argue our technique to adapt efficiently, reaching a new state-of-the-art on some corruptions without altering training parameters. We provided explanations of the success and the flaws of spectral penalization and its connections with standard methods in Machine Learning.

A Appendix

A.1 Connection with L^2-Regularization

Let $X \in \mathbb{R}^{n \times d}$, where n is the number of samples and d is the number of features. We consider the simple case of linear regression where $Y = X\theta$ where θ is the parameter of the model. The optimal parameter are defined as follows:

$$\theta^\star := \arg\min_\theta ||Y - X\theta||^2 \tag{5}$$

and it is straightforward to observe the following closed form:

$$\theta^\star = \left(X^\top X\right)^{-1} X^\top Y \tag{6}$$

it is also straightforward to observe that:

$$\theta^\star_\gamma := \arg\min_\theta ||Y - X\theta||^2 + \gamma \cdot ||\theta||^2 \tag{7}$$

leads to the close form:

$$\theta^\star_\gamma = \left(X^\top X + \gamma I_d\right)^{-1} X^\top Y \tag{8}$$

In the following, we note $C = X^\top X$. C is has an orthogonal eigen decomposition (symmetric, positive and definite).

$$C = U^\top D U \tag{9}$$

where $U \in \mathbb{U}(d)$ which is the unitary group $U^\top U = I_d$. We note the basis change of X as follows:

$$\tilde{X} := XU^\top \tag{10}$$

By construction, $\tilde{X}$ has a diagonal covariance,

$$\tilde{X}^\top X = UX^\top XU^\top = UX^\top XU^\top = UCU^\top = D \tag{11}$$

Now, what happens when regressing from $\tilde{X}$ to obtain $\tilde{\theta}^\star$:

$$\tilde{\theta}^\star := D^{-1}\tilde{X}Y \tag{12}$$

Now,

$$\tilde{X}\tilde{\theta}^\star = \tilde{X}D^{-1}\tilde{X}^\top Y = Y \tag{13}$$

$$X\underbrace{U^\top D^{-1} U X^\top Y}_{=\theta} = \tilde{X}D^{-1}\tilde{X}^\top Y = Y \tag{14}$$

$$X\underbrace{U^\top (D+\gamma I_d)^{-1} U X^\top Y}_{=\theta} = \tilde{X}D^{-1}\tilde{X}^\top Y = Y \tag{15}$$

$$\theta^\star \tilde{X} = \theta^\star X \tag{16}$$

Let break the equation of $\theta^\star_\gamma$:

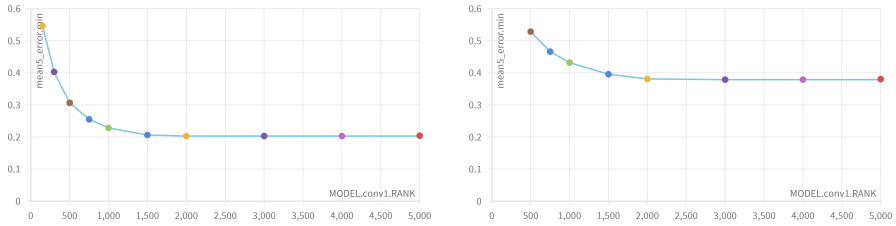
$$\theta^\star_\gamma = \left(X^\top X + \gamma I_d\right)^{-1} X^\top Y \tag{17}$$
$$= \left(U^\top (D + \gamma I_d) U\right)^{-1} X^\top Y \tag{18}$$
$$= U^\top (D + \gamma I_d)^{-1} U X^\top Y \tag{19}$$
$$= U^\top \underbrace{D(D + \gamma I_d)^{-1}}_{F_\gamma} D^{-1} U X^\top Y \tag{20}$$
$$= U^\top F_\gamma U U^\top D^{-1} U X^\top Y \tag{21}$$
$$= U^\top F_\gamma U \theta^\star_0 \tag{22}$$

where F_γ is a diagonal matrix such that:

$$F_{\gamma,i,i} = \frac{\lambda_i}{\lambda_i + \gamma} \tag{23}$$

where λ_i is the $i-$th eigen-value of C.

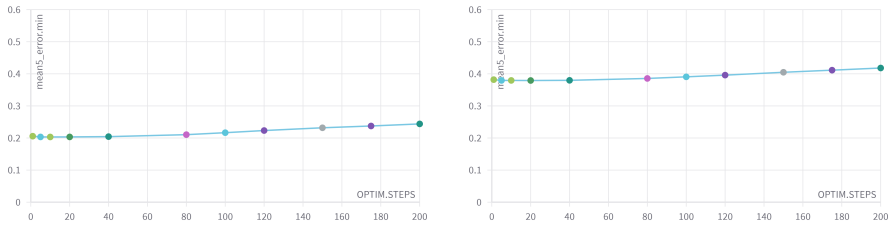
A.2 Ablation Studies



(a) CIFAR-10-C (b) CIFAR-100-C

Fig. 1: Episodic mean error along all corruptions at severity 5 for different rank of the PCA of TTAwPCA.

PCA rank and parameters of the filter Our experiments investigated how many parameters are enough to tackle corrupted data points. While these results only apply to CIFAR-10-C and CIFAR-100-C, we experienced that 2000 parameters are enough to effectively train a model to regain accuracy after a distributional shift at test time. In Figure 1, we show the mean error on all corruptions at severity 5 for different ranks of the PCA on both datasets. We averaged over three runs for each PCA rank with minor variations. The optimization has been done in an episodic setting.



(a) CIFAR-10-C (b) CIFAR-100-C

Fig. 2: Online mean error along all corruptions at severity 5 for different number of learning steps of TTAwPCA.

Optimizing steps As shown in [17], error degrades over optimization steps as entropy minimization lacks target distribution regularization. Still, this effect is minor compared with the accuracy retrieval achieved by our simple method.

A.3 Insight on CIFAR-10-C

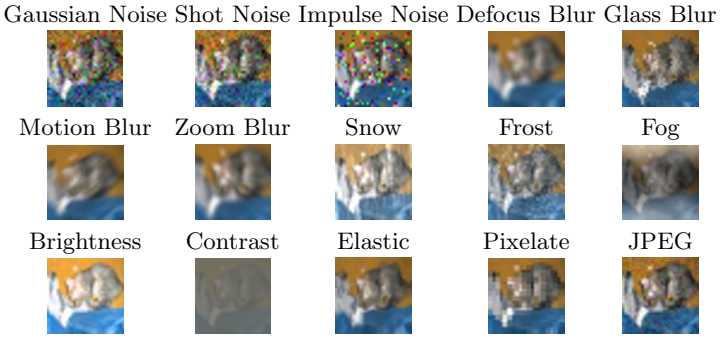


Fig. 3: CIFAR-10-C [8] consists of 15 corrupted versions of the CIFAR-10 test dataset [13] with 5 levels of severity (level 5 here).

A.4 Insight on Principal Component Analysis

Principal Component Analysis (PCA) linearly separates multivariate systemic variation from noise. Consider A an $N \times p$ data matrix. PCA defines its principal components as the $q \leq p$ unit vectors such that the i-th vector satisfy orthogonality with the first $i-1$ and best fits the direction of data. The process performs a change of basis on the data according to the principal components. They are computed by Singular Value Decomposition (SVD) of A and ranked by the

corresponding singular value scale. Thus irrelevant principal components can be ignored.

Incremental PCA can be performed if the dataset is too large to fit in the memory. Incremental PCA uses an amount of memory independent of the number of input data samples to build a low-rank approximation.

References

1. Baochen, S., Jiashi, F., Kate, S.: Correlation Alignment for Unsupervised Domain Adaptation, p. 153–171. Springer (2017)
2. Bartlett, P.L., Foster, D.J., Telgarsky, M.: Spectrally-normalized margin bounds for neural networks. In: Advances in Neural Information Processing Systems. Curran Associates, Inc. (2017), https://proceedings.neurips.cc/paper/2017/file/b22b257ad0519d4500539da3c8bcf4dd-Paper.pdf
3. Chidlovskii, B., Clinchant, S., Csurka, G.: Domain adaptation in the absence of source domain data. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 451–460 (2016). https://doi.org/https://doi.org/10.1145/2939672.2939716
4. Croce, F., Andriushchenko, M., Sehwag, V., Debenedetti, E., Flammarion, N., Chiang, M., Mittal, P., Hein, M.: Robustbench: a standardized adversarial robustness benchmark (2021)
5. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: Bach, F., Blei, D. (eds.) Proceedings of the 32nd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 37, pp. 1180–1189. PMLR, Lille, France (07–09 Jul 2015), http://proceedings.mlr.press/v37/ganin15.html
6. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., March, M., Lempitsky, V.: Domain-adversarial training of neural networks. Journal of Machine Learning Research **17**(59), 1–35 (2016), http://jmlr.org/papers/v17/15-239.html
7. Gretton, A., Smola, A., Huang, J., Schmittfull, M., Borgwardt, K., Schölkopf, B.: Covariate shift and local learning by distribution matching, pp. 131–160. MIT Press, Cambridge, MA, USA (2009)
8. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=HJz6tiCqYm
9. Hoffman, J., Tzeng, E., Park, T., Zhu, J.Y., Isola, P., Saenko, K., Efros, A., Darrell, T.: CyCADA: Cycle-consistent adversarial domain adaptation. In: Dy, J., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 80, pp. 1989–1998. PMLR (10–15 Jul 2018), http://proceedings.mlr.press/v80/hoffman18a.html
10. Hu, W., Miyato, T., Tokui, S., Matsumoto, E., Sugiyama, M.: Learning Discrete Representations via Information Maximizing Self-Augmented Training. In: Proceedings of the 34th International Conference on Machine Learning. pp. 1558–1567. PMLR (2017)
11. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), http://arxiv.org/abs/1412.6980
12. Krause, A., Perona, P., Gomes, R.: Discriminative clustering by regularized information maximization. In: Lafferty, J., Williams, C., Shawe-Taylor, J., Zemel, R., Culotta, A. (eds.) Advances in Neural Information Processing Systems. vol. 23. Curran Associates, Inc. (2010), https://proceedings.neurips.cc/paper/2010/file/42998cf32d552343bc8e460416382dca-Paper.pdf

13. Krizhevsky, A.: Learning multiple layers of features from tiny images. Technical report, University of Toronto (2009)
14. Li, B.: A principal component analysis approach to noise removal for speech denoising. In: 2018 International Conference on Virtual Reality and Intelligent Systems (ICVRIS). pp. 429–432 (2018). https://doi.org/10.1109/ICVRIS.2018.00111
15. Liang, J., Hu, D., Feng, J.: Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In: International Conference on Machine Learning (ICML). pp. 6028–6039 (July 13–18 2020)
16. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. In: International Conference on Learning Representations (2018), https://openreview.net/forum?id=B1QRgziT-
17. Mummadi, C.K., Hutmacher, R., Rambach, K., Levinkov, E., Brox, T., Metzen, J.H.: Test-time adaptation to distribution shift by confidence maximization and input transformation (2021), https://arxiv.org/pdf/2106.14999.pdf
18. Murali, Y., Babu, M., Subramanyam, D., Prasad, D.: Pca based image denoising. Signal & Image Processing **3** (04 2012). https://doi.org/10.5121/sipij.2012.3218
19. Nado, Z., Padhy, S., Sculley, D., D'Amour, A., Lakshminarayanan, B., Snoek, J.: Evaluating prediction-time batch normalization for robustness under covariate shift (2020)
20. Neyshabur, B., Bhojanapalli, S., McAllester, D., Srebro, N.: Exploring generalization in deep learning. In: Advances in Neural Information Processing Systems. Curran Associates, Inc. (2017), https://proceedings.neurips.cc/paper/2017/file/10ce03a1ed01077e3e289f3e53c72813-Paper.pdf
21. Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., Lawrence, N.D.: Dataset shift in machine learning. In: MIT Press (2009)
22. Schneider, S., Rusak, E., Eck, L., Bringmann, O., Brendel, W., Bethge, M.: Improving robustness against common corruptions by covariate shift adaptation. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 11539–11551. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper/2020/file/85690f81aadc1749175c187784afc9ee-Paper.pdf
23. Shi, Y., Sha, F.: Information-theoretical learning of discriminative clusters for unsupervised domain adaptation. In: Langford, J., Pineau, J. (eds.) Proceedings of the 29th International Conference on Machine Learning (ICML-12). pp. 1079–1086. ICML '12, Omnipress, New York, NY, USA (July 2012)
24. Sun, Y., Tzeng, E., Darrell, T., Efros, A.A.: Unsupervised domain adaptation through self-supervision (2019)
25. Sun, Y., Wang, X., Zhuang, L., Miller, J., Hardt, M., Efros, A.A.: Test-time training with self-supervision for generalization under distribution shifts. In: ICML (2020)
26. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)
27. Wang, D., Shelhamer, E., Liu, S., Olshausen, B., Darrell, T.: Tent: Fully test-time adaptation by entropy minimization. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=uXl3bZLkr3c
28. Zagoruyko, S., Komodakis, N.: Wide residual networks. In: BMVC (2016)