Learning Transparent Test-Time Augmentation via Gradient-Free Optimization

Anonymous Authors[1]

Abstract

Deep learning models for vision tasks can hardly have satisfactory performance if tested on data discrepant from the training set without robustness-ascertaining techniques or proper domain adaption. Random data augmentation and fine-tuning are widely used to mitigate the weakness caused by distribution discrepancy. However, they usually require modifying the pretrained deep models, which becomes costly or even impractical for deployment. Unsupervised domain adaptation (UDA) was proposed to shift the unlabeled data to the training domain, achieving favorable results on many tasks. Nevertheless, existing UDA frameworks require access to training data and transfer input samples with non-interpretable deep models, which can be hindered in application domains with strict requirements of data privacy and model transparency (*e.g.*, medical). In this paper, we introduce a novel framework, Transparent Test-Time Augmentation via Gradient-Free Optimization (GraFTTA), to perform efficient instance-specific test-time augmentation with interpretable transforms. Our method optimizes a set of continuous variables that control the strength of several image operators via gradient estimation to improve the inference performance when treating pretrained models as black-boxes. With only a fraction of test labels and no training data, our GraFTTA significantly out-performance existing approahes on xxx dataset by xx%.

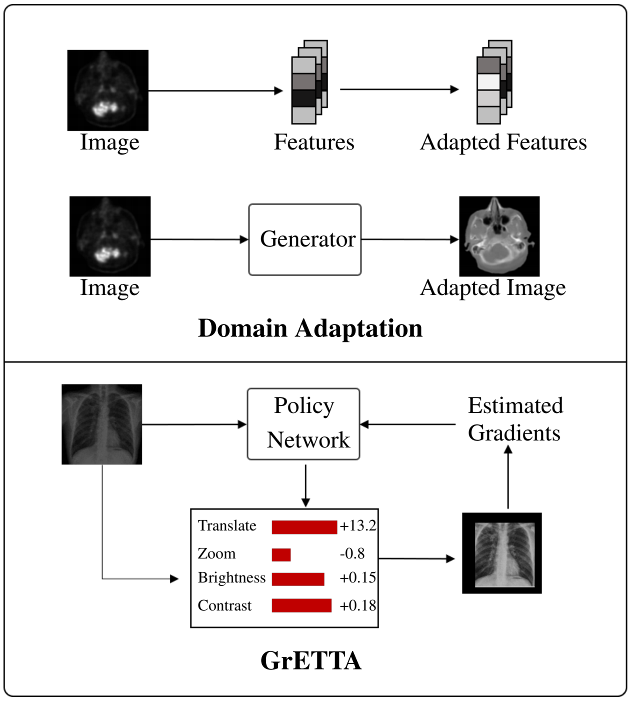

Figure 1. An illustration of the comparison between previous work and our proposed methods. On the top we illustrate the principle of image-leveland feature-level domain adaptation; On the bottom we demonstrate the pipeline of our methods GrETTA, where a policy network outputs magnitudes of understandable image transforms and is updated via estimated gradients.

1. Introduction

With the emergence of large-scale datasets (Rajpurkar et al., 2017)(Irvin et al., 2019)(Wei et al., 2020), deep learning models have achieved impressive performance for numerous biomedical image analysis tasks, from Chest X-ray classification to 3D cell instance segmentation (Rajpurkar et al., 2017).

However, unlike natural images, biomedical images tend to have large domain gaps across biology labs or clinics due to different imaging equipment, imaging parameters and tissue preparation protocols.

Worse still, we can not adopt the popular adversarial robustness or domain adaptation paradigms to close the domain gap, as the data distribution shift in real world deployments is unforeseeable during the model training.

To address it, recent works aim to learn the distribution shifts at the test time.

Given the known model, (Sun et al., 2020) fine-tunes the model parameters with the self-supervision head and (Wang et al., 2020) learns the affine transformation to align the data distribution.

However, for most biology labs and clinics, they have little knowledge of the model licensed by commercial companies (black-box) nor enough computational resource to fine-tune the large-sized deep learning model.

Thus, instead of tuning the model to adapt to the test data distribution, it is more plausible to learn to modify the input image during test-time to conform to the model structure given the situation in practice.

Furthermore, biomedical experts, especially in oncology, prefer the applied image transformation to be interpretable (white-box) to make sure no artificial textures are inserted to bias the model prediction. In contrast, many works on unsupervised domain adaptation that depend on the generative models (Xia & Ding, 2020) (Pan et al., 2020) (Dong et al., 2020) will generate perceptible textures that is not comprehensible to even human experts, which will dwindle the trustworthiness of these methods.

Data augmentation is another topic that is closely-related to our motivation since most widely-used augmentation methods, like cropping, affine transforms and color space transforms are white-box image filters. However, traditional data augmentation is applied before the training phase, aiming to enhance the robustness of the neural networks. (**?**) There are, admittedly, some test-time augmentation (TTA) tricks, e.g. five-crop and ten-crop. However, these TTA methods are intended for reducing the inference variance. What we seek is actually an instance-aware augmentation policy that will alter each input image based on its content and the black-box model.

Our goal is to develop a efficient instance-aware TTA method that works under a few-shot circumstance with low computational cost. We summarize our contribution to this problem is three-fold:

- We model the test-time augmentation problem to be a simple optimization problem over a set of continuous variables that controls the intensity of the white-box image filters, hence enlarging the search space.
- We address a effective solution(GrETTA) to solve this optimization problem by gradient estimation method and outperforms existing baselines on several visual tasks.
- we present discussions about how our method will deliver best performance in few-shot learning scenarios.

2. Related Works

Test-time Augmentation Test-time adaptation for deep learning is starting to be used in computer vision (Mullapudi et al., 2019)(Shocher et al., 2018b)(Shocher et al., 2018a)(Wang et al., 2020)(Nikita et al., 2020)(Sun et al., 2020)(Ayhan & Berens, 2018) for increasing the test-time performance of visual models. For example, (Shocher et al., 2018a) shows that image super-resolution can be learned at test time simply by trying to upsample a downsampled version of the input image. (Bau et al., 2020) show that adapting the prior of a generative adversarial network to the statistics of the test image improves photo manipulation tasks. (Sun et al., 2020)performs joint optimization of image recognition and self-supervised learning with rotation prediction then uses the self-supervised objective to adapt the representation of each individual image during testing.

TTA are also applied to the adversarial attack and defense. (Goodfellow et al., 2014)(Su et al., 2019) This can be considered to be a reverse of our goal: we are trying to adapt the data so the model acquires better performance, as they are attempting to modify the data such that the black-box model has a ill performance.

The most closely-related work to our proposal is (Kim et al., 2020), which has proposed an instance-specific TTA approach based on imitation learning. Nonetheless, this method demands a large volume of training data to ensure its generization ability. This requirement brings additional computational complexity and requires access to the training data, which may be difficult in application circumstances.

Search for the Optimal Image Transforms White-box image transforms are initially used in training data augmentation to improve the robustness and generalization ability of visual models. It was first introduced as stochastic white-box image processing filters, until (Cubuk et al., 2018) proposes the search for an optimal augment policy by reinforcement learning. Many succeeding works focus on improve the efficiency of the search (Cubuk et al., 2020)(Lim et al., 2019)(Hataya et al., 2020). The search for the optimal image processing procedures are also applied in style transfer. (Hu et al., 2018).

Monte-Carlo Gradient Estimation With the gradient information being unavailable, (Rechenberg) (Owen, 2013) proposes utilizing Monte-Carlo sampling to estimate the gradients with finite differences between function evaluations. (Maheswaranathan et al., 2019) provides a improved version of such random search by revising the covariance matrix of the sample distribution and immensely raise the convergence efficiency. This method can be directly applied to our problem given an appropriate modeling, as will be stated in the next section.

3. GrETTA: A gradient-free TTA method with white-box image filters

3.1. A Non-MDP Modeling

Conventionally, data augmentation tasks has been interpreted as a sequential decision-making process, as is modeled by a Markov Decision Process (MDP) (Cubuk et al., 2018)(Cubuk et al., 2020)(Lim et al., 2019)(Hataya et al., 2020)(Hu et al., 2018). A trajectory, or an episode is defined as follows

$$T = < S_1, A_1, R_1, S_2, A_2, R_2, \cdots >$$

Where S_t are states, or image instances in this setting. A_t denotes the actions, which would be various transforms executed on the image. and reward R_t is the corresponding metric to evaluate such transforms.

Typically, the decision of A_t is made by a deep neural network that outputs a distribution over the probabilities of choosing different transforms. A greedy or stochastic greedy policy will be utilized to choose the favorable one and apply it to the input image. The neural network itself will be trained via reinforcement learning (RL) algorithms.

This modeling approach, admittedly effective, has a significant defect if applied to the TTA challenge. Since we made the assumption of black-box model, the rewards' gradient with respect to the transform intensity is not accessible. In this way, the continuous RL algorithms, such as DDPG (Lillicrap et al., 2015), is not feasible in this practice. (The gradient estimation techniques is not practicable as the RL algorithm itself has a very high variance). Hence, the optimization for the transform magnitudes must be implemented among the discrete levels, consequently narrowing down the search scope. In addition, the choice among different image transforms also put an unnecessary restriction to the search space.

To resolve this deficiency, we discarded the traditional sequential decision-making modeling. Instead, every candidate transforms will be performed on the input image. The magnitude of each transforms will be manipulated by a continuous variable. That being said, we have

$$x_i = t_i(x_{i-1}, l_i) \tag{1}$$

Where $x_0 = x$ is the initial test image, and $x' = x_n$ is the augmented image that is supposed to be fed into the black-box network. t_i is a set of differentiable image transforms, and l_i denotes the corresponding magnitude of such transforms.

The target of our method is to train an effective and efficient policy network π_θ responsible for computing the optimal values of these magnitudes with image tensors as input:

$$\pi_\theta(x_0) = (l_1, l_2, \cdots, l_n)$$

.

3.2. Gradient Estimation and Random Search

Despite the fact that we have chosen differentiable image transforms, the gradients of the model outputs with respect to the adapted input images are still uncovered in our settings. In other words, we expect to optimize a function $f(x)$ (Could be cross entropy loss or any other target functions) over an n-dimensional space without ∇f being available. A common approach is to perform Gaussian sampling and estimate a descent direction with function differences. This method is also referred to as evolutionary strategies (Rechenberg) or random search. To reduce the variance, (Owen, 2013) proposes using a pair of function evaluations on antithetic direction to estimate the gradient. The estimation of the descent direction can subsequently be described as below:

$$g = \frac{\beta}{2\sigma^2 P} \sum_{i=1}^{P} \varepsilon_i \left(f(x + \varepsilon_i) - f(x - \varepsilon_i) \right) \tag{2}$$

where ε_i is sampled from a 2-dimensional Gaussian distribution $\mathcal{N}(0, \sigma^2 I)$. The scaling parameter β and the distribution variance σ^2 are considered to be hyperparameters. The estimation is unbiased due to the Monte-Carlo sampling process. Nevertheless, the algorithm is entirely built on the $2P$ queries on the black-box function and consequently has a very high variance. The choice of σ involves a trade-off between error brought by the random noises and gradient approximation. Increasing the variance will induce the inaccuracy of gradient approximation due to the high order terms in Taylor expansion (Lehman et al., 2018) while decrease it will prompt a greater error brought by the sampling noises. In practise, relatively low σ will be chosen. To rule out the perturbation of noises and achieve satisfactory optimization results, the algorithm will require an immense bulk of samples and thus is costly both in time and space complexity.

To improve the efficiency, we sample the P random samples in parallel, which introduces an extra trade-off between training batch size B and sample size P owing to the limitation of computational resources. The enlargement of batches will reduce the variance regarding training data sampling, while the increase of P will curtail the variance of gradient estimation.

3.3. Surrogate Gradients and the guided search

To reduce the high variance in Vanilla ES and boost the query efficiency, (Maheswaranathan et al., 2019) proposes using surrogate gradients as a prior knowledge to perform guided search in the input space $\mathcal{X}$.

To address it, suppose we have a surrogate information regarding the gradients at a given point in $\mathcal{X}$. This surrogate gradients have the same dimensions as the gradient vectors, and are to a certain degree correlated with the true gradients (though being biased). During a parameter update iteration, if we have observed previous k surrogate gradients, we can generate a subspace with regards to them. Using QR decomposition we can denote this subspace with $n \times k$ orthogonal basis matrix U, i.e. $UU^T = I_k$. These basis can then be exploited to guide the random search by providing a revised covariance matrix for the Gaussian distribution:

$$\Sigma = \frac{\alpha}{n} I_n + \frac{1-\alpha}{k} UU^T$$

and ε_i will be drawn from the modified distribution $\mathcal{N}(0, \sigma^2\Sigma)$. (Maheswaranathan et al., 2019) justifies the validity of this revision by computing the expectation of the estimated gradients with second-order Taylor expansion approximation and proving g is a descent direction.

Given the fact that the guided search is likely to acquire a descent direction, it's somewhat biased. By acquiring a faster convergence than the random search, the guided search might well fall into a less-satisfactory local minimum, while random search is expected to achieve nearly as good a result as gradient descent. This is yet another trade-off between variance and bias.

As to the surrogate gradient generation, , we train a student model in advance via the knowledge distillation method (Hinton et al., 2015). We consider the original black-box model as the teacher model ϕ, and train the student model by fitting the outputs of the model. The pretraining objective is the mean square error (MSE):

$$l_{student} = \underset{x \sim D_{\text{test}}}{\hat{\mathbb{E}}} [\text{MSE}(\phi(x), f(x))]$$

We then take the gradients of the student model $\nabla\phi(x)$ to be the surrogate information. To prevent numerical errors, we perform clipping to the sampled perturbations ε_i.

3.4. Search Space and Optimization Target

In our implementation, a set of differentiable image transforms are required. Following the precedent work[AutoAug, RandomAug, Fast AutoAug, Learning loss], we use various transforms including geometry transforms, color space transforms, and sharpness filter. We adopt the implementation of a recently-released computer vision library kornia (Riba et al., 2020). The detailed description of the chosen transforms for each tasks can be found in the next section about the experiment settings. In practise, we found different subsets of these transforms ought to be employed when dealing with different visual tasks and datasets.

To ascertain the regularity of the transform magnitudes, we appointed a feasible range for each transforms, as is listed in table 1. The output l_i of our policy network will be normalized to $(0, 1)$ by the sigmoid function $\sigma(\cdot)$ and then regularized to the predefined ranges. Note that if the sigmoid-normalized magnitude is 0.5, we implicitly executes an identity transform on the image, which will naturally jettison the inappropriate transforms. The sharpness filter is an exception, since its feasible range is not synthetic with regards to the identity transform, we perform a trick by taking the absolute value of the normalized output as the transform magnitude.

Name	TranslateX	TranslateY	ZoomX	ZoomY
Min	-20	20	0.7	0.7
Max	-20	20	1.3	1.3
Name	Rotate	Brightness	Contrast	Sharpness
Min	-15	-0.3	0.8	0
Max	15	0.3	1.2	0.3

Table 1. The predefined range of magnitude for each transform. Note that the sharpness filter is an exception since its range is not synthetic, ad perform the absolute value trick to resolve this problem.

The differentiable transforms are then applied to the image with these regularized magnitudes and these adapted image tensors will be posited into the black-box model and acquire an target function evaluation. The target function can be chosen with flexibility considering the discrepant characteristics of various visual tasks and datasets. Using the surrogate gradient and the query result, we can estimate the gradient with respect to the magnitudes. And subsequently the descent direction of the policy network parameters will be attained through backwarding process. We initialize the last linear layer of our policy network to be all zero in light of starting up our optimization from identity transforms. (since $\sigma(0) = 0.5$).

In a nutshell, our method can be summarized in algorithm 1.

3.5. Few-Shot Learning and Generalization

Unlike the methodology described in (Kim et al., 2020), we do not exploit any of the data used to train the model, as in accordance with our assumption that the training data and model structure are unavailable. Since our method also demands the groundtruth labels to train the policy network, we consider our problem to be a few-shot learning problem. In practice, we only make use of a relatively modest proportion of the test labels (1% to 20%, variant due to the concrete problem).

Our method proves to have a good generalization capability and low variance. Nevertheless, we here discuss several popular approaches that might enhance the robustness of our policy model

Algorithm 1 (GrETTA) Training

```
Input: input image $x$, black-box model $f$, surrogate student model $\phi$, policy model $\pi_\theta(i)$, A set of differentiable transforms $\{t_i\}_{i=1}^n$, target loss function $\mathcal{T}$, number of iterations $N$, number of samples $P$, descent direction scaling factor $\beta$, weight factor $\alpha$ sample distribution variance $\sigma^2$, learning rate $\eta$.
Output: optimized policy network $\pi_\theta^*$

Initialize the last layer of $\pi_\theta$ to be all zero;
for epoch in $\{0, 1, 2, ..., N\}$ do
    Compute magnitudes $(l_1, \cdots, l_n) \leftarrow \sigma(\pi_\theta(x))$;
    Regularize $l_i$ by the predetermined ranges;
    Apply transforms and get the output $x'$ by formula 1;
    Compute target evaluation $J = \mathcal{T}(\phi(x'))$;
    Attain surrogate gradient $\gamma \leftarrow \nabla_l J$;
    Update guiding subspace $U$ with $\gamma$;
    Define covariance matrix $\Sigma \leftarrow \frac{\alpha}{n} I_n + \frac{1-\alpha}{k} U U^T$;
    for $i \leftarrow 1$ to $P$ do
        Sample perturbation $\varepsilon_i \sim \mathcal{N}(0, \sigma^2 \Sigma)$;
        Query antithetic model output $f(x + \varepsilon_i), f(x - \varepsilon_i)$;
    end for
    $g \leftarrow \frac{\beta}{2\sigma^2 P} \sum_{i=1}^P \varepsilon_i \left(\mathcal{T}(f(x + \varepsilon_i)) - \mathcal{T}(f(x - \varepsilon_i)\right)$;
    Compute the descent direction $\Delta\theta$ by backwarding $g$;
    Update the policy network with $\theta \leftarrow \theta - \eta \Delta\theta$;
end for
return $\pi_\theta$.
```

Ensemble Learning. Inspired by the idea of bagging in Random Forest, we can use different subsets of the image transforms to train on bootstrapping samples and then average the outputs of these policy networks to increase the robustness and the generalization ability.

Train-time Data Augmentation. One classic and effective way to train robustness-ascertained networks is to perform training data augmentation, which will expose the network to a wider scope of data distributions. We also add stochastic training data augmentations to observe whether there are prominent improvement to the performance of our method. However, the sampling procedure in our algorithm already to a certain extent enlarge the input space for our network training, thus the training data augmentation is not a necessity.

4. Experiments

4.1. Toy Example: image classification on modified MNIST

Basic Settings. We create a simple toy example on MNIST to verify our method. Since MNIST is too easy for convolutional neural networks(CNNs), we build a harder version of modified MNIST by executing random color space transformations (including random changes on brightness, contrast, saturation and hue level), and random affine transformations, on each image instances. (implemented by the ColorJitter transform in pytorch). We call this dataset MNIST-m for convenience.

We first train a simple 3-layer CNN on the orginal MNIST dataset until it achieves a considerable accuracy(98.90%). Then we evaluate this model on MNIST-m. To our expectation, the accuracy on this modified version of MNIST drops to around 65%.

We then build a GrETTAtrainer with the transforms mentioned in table 1. The backbone of the policy network is a 2-layer CNN. Since we need a student model to provide surrogate gradients, we also train a 2-layer CNN by fitting the output of the original model (3-layer CNN) with the whole data of MNIST-m (but without the labels). A small proportion (from 1% to 20%) of the MNIST-m data is drawn and is used as the training set, and a validation set of the same size as the training set is divided. The rest of the data is considered the testing set. The policy network is trained on the training set and we perform model selection according to the performance on the validation set. The final results that we report is the selected policy network's performance on the testing set.

We employ both the random search and the guided search as the gradient estimator in our experiments and compare their effectiveness. The target loss function is the common cross-entropy (CE) loss. As for the hyperparamters, we set the scaling factor $\beta = 1$, the weight factor $\alpha = 1$, the distribution variance $\sigma^2 = 0.01$, the number of surrogate gradients $k = 1$, and the number of samples $P = 20$.

For the optimizer, we chose the stochastic gradient descent (SGD) with momentum set to 0.9. To accelerate the training process, we perform learning rate decay. The learning rate $\eta = 0.02$ and we train the policy network for 100 epochs.

To demonstrate the potential of such tasks, we conduct some experiments to find the oracle upper bound for the white-box TTA problem. We optimize the policy network using the gradients of the loss function with regards to the network parameters $\nabla_\theta J$. This upper bound is called the white-box oracle.

We also compare the performance of our method with several popular test-time augmentation baseline, like five-crop and ten-crop. We demonstrate our results in Table .

Performance Analysis. As is stated in the aforementioned paragraph, we used different proportions of train data to test our method. Without the TTA transform, the accuracy of the pretrained black-box model has a rather poor performance of around 65%. The tradition TTA methods, like

Methods	Proportion of test labels that are utilized for training of the policy network					
	1%	2%	5%	10%	15%	20%
Without TTA	65.84%	65.58%	65.37%	65.31%	65.69%	64.90%
Five-crop	67.96%	67.91%	67.56%	68.21%	68.00%	67.62%
Ten-crop	55.48%	55.42%	54.00%	55.18%	55.64%	55.23%
GrETTA(Random Search)	**79.04%**	**82.83%**	**83.60%**	**85.43%**	**86.36%**	**90.00%**
GrETTA(Guided Search)	78.76%	81.39%	83.51%	82.78%	85.63%	85.18%
White-box Oracle Upper Bound	82.53%	84.32%	88.51%	89.32%	90.21%	90.65%

five-crop or ten-crop, does not improve the prediction accuracy prominently. Our proposed method, however, achieve satisfactory result, and raise the accuracy by 12.88% with only 1% of the test label and by up to 25.10% with 20% of the test label.

Our method demonstrates a favorable ability of generalization with few-shot samples. Even with a very modest proportion(1%) of data for training, our method accomplishes a noticeable improvement compared to the traditional TTA method.

As is discussed in section 3.3, due to the trade-off between variance and bias, the random search will gradually approach the white-box oracle upper bound as the proportion increases, while the guided search will achieve a little worse result but with a much better query efficiency. This projection is verified by the experiment result. The difference between the white-box oracle and the random search narrows down to 0.65% as the data proportion approaches 20%.

5. Conclusion

In this paper, we illustrate the importance of developing methods to perform understandable test-time augmentation with black-models on unseen test data. We show that the TTA problem in visual tasks can be modeled as the optimization of a set of continuous variables that controls the magnitude of differentiable image transforms. With the help of effective gradient-estimation method, we propose **Gr**adient **E**stimation **T**est **T**ime **A**ugmentation, which achieves prominent improvement on several visual tasks, and presents a desirable quality of few-shot generalization. By performing guided search with the assistance of surrogate gradients generated by an apprentice network that fits the black-box model outputs, we mitigate the high variance in the gradient-estimation, and consequently attain a faster and stabler convergence. Our method GrETTAprovides a novel application scenario for the data augmentation, and shows a great potential in handling inter-domain generalization with restricted resources.

References

Ayhan, M. S. and Berens, P. Test-time data augmentation for estimation of heteroscedastic aleatoric uncertainty in deep neural networks. 2018.

Bau, D., Strobelt, H., Peebles, W., Zhou, B., Zhu, J.-Y., Torralba, A., et al. Semantic photo manipulation with a generative image prior. *arXiv preprint arXiv:2005.07727*, 2020.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 702–703, 2020.

Dong, J., Cong, Y., Sun, G., Zhong, B., and Xu, X. What can be transferred: Unsupervised domain adaptation for endoscopic lesions segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4023–4032, 2020.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Hataya, R., Zdenek, J., Yoshizoe, K., and Nakayama, H. Faster autoaugment: Learning augmentation strategies using backpropagation. In *European Conference on Computer Vision*, pp. 1–16. Springer, 2020.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Hu, Y., He, H., Xu, C., Wang, B., and Lin, S. Exposure: A white-box photo post-processing framework. *ACM Transactions on Graphics (TOG)*, 37(2):1–17, 2018.

Irvin, J., Rajpurkar, P., Ko, M., Yu, Y., Ciurea-Ilcus, S., Chute, C., Marklund, H., Haghgoo, B., Ball, R., Shpanskaya, K., et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 590–597, 2019.

Kim, I., Kim, Y., and Kim, S. Learning loss for test-time augmentation. *Advances in Neural Information Processing Systems*, 33, 2020.

Lehman, J., Chen, J., Clune, J., and Stanley, K. O. Es is more than just a traditional finite-difference approximator. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 450–457, 2018.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Lim, S., Kim, I., Kim, T., Kim, C., and Kim, S. Fast autoaugment. In *Advances in Neural Information Processing Systems*, pp. 6665–6675, 2019.

Maheswaranathan, N., Metz, L., Tucker, G., Choi, D., and Sohl-Dickstein, J. Guided evolutionary strategies: Augmenting random search with surrogate gradients. In *International Conference on Machine Learning*, pp. 4264–4273. PMLR, 2019.

Mullapudi, R. T., Chen, S., Zhang, K., Ramanan, D., and Fatahalian, K. Online model distillation for efficient video inference. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3573–3582, 2019.

Nikita, M., Botond, M., Attila, K.-F., Reka, H., and Horvath, P. Test-time augmentation for deep learning-based cell segmentation on microscopy images. *Scientific Reports (Nature Publisher Group)*, 10(1), 2020.

Owen, A. B. Monte carlo theory, methods and examples. *Monte Carlo Theory, Methods and Examples. Art Owen*, 2013.

Pan, F., Shin, I., Rameau, F., Lee, S., and Kweon, I. S. Unsupervised intra-domain adaptation for semantic segmentation through self-supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3764–3773, 2020.

Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C., Shpanskaya, K., et al. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv:1711.05225*, 2017.

Rechenberg, I. „evolutionsstrategie-optimierung technisher systeme nach prinzipien der biologischen evolution “,(1973) frommann-holzboog. *Stuttgart, Germany*.

Riba, E., Mishkin, D., Ponsa, D., Rublee, E., and Bradski, G. Kornia: an open source differentiable computer vision library for pytorch. In *The IEEE Winter Conference on Applications of Computer Vision*, pp. 3674–3683, 2020.

Shocher, A., Bagon, S., Isola, P., and Irani, M. Ingan: Capturing and remapping the” dna” of a natural image. *arXiv preprint arXiv:1812.00231*, 2018a.

Shocher, A., Cohen, N., and Irani, M. “zero-shot” super-resolution using deep internal learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3118–3126, 2018b.

Su, J., Vargas, D. V., and Sakurai, K. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019.

Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A. A., and Hardt, M. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning (ICML)*, 2020.

Wang, D., Shelhamer, E., Liu, S., Olshausen, B., and Darrell, T. Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

Wei, D., Lin, Z., Franco-Barranco, D., Wendt, N., Liu, X., Yin, W., Huang, X., Gupta, A., Jang, W.-D., Wang, X., et al. Mitoem dataset: Large-scale 3d mitochondria instance segmentation from em images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 66–76. Springer, 2020.

Xia, H. and Ding, Z. Structure preserving generative cross-domain learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4364–4373, 2020.