ITERATIVE LEARNING FOR DISTORTED IMAGE RESTORATION

Chao Wang[†], *Yi Gu*[†], *Jie Li*[†‡*], *Xinlei He*[†], *Zirui Zhang*[†], *Yuting Gao*[§] *and Chentao Wu*[†]

[†] Department of Computer Science and Engineering, Shanghai Jiao Tong University, China
[‡] MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, China
[§] Department of Electrical and Computer Engineering, Texas A&M University, US

ABSTRACT

Deep generative networks have achieved great success on distorted image restoration. However, existing deep learning approaches mainly focus on delicate module structure while ignoring the saturation problem. In this paper, we study the influence of different learning schemes on fitting capability and tackle the problem by proposing a novel iterative learning scheme. It accumulates weight importance from past episodes and guides the network to search for the optimal of current episodes based on obtained knowledge. Since public available datasets contain very few distortion types, we also release a new benchmark to explore this task. Extensive experimental evaluations on the benchmarks demonstrate that our learning approach significantly outperforms all other methods and achieves new state-of-the-art results.

Index Terms— Image Restoration, Iterative Learning, Benchmark.

1. INTRODUCTION

Liquify [1] is significantly valued to image editing. It can distort images for special effects [2, 3] and reversely restore distorted images [4, 5]. However, recovery is a challenging task which is more complicated than destruction. With the success of generative adversarial networks [6], some generative models for distorted image restoration have been introduced [7, 8, 9]. These methods formulate the distortion restoration problem as finding the appropriate warping and predicting the dense grid, which achieve the state-of-the-art results. However, existing efforts mainly focus on the model structure, one opportunity that is widely ignored is learning approach.

Typically, the neural network fits the distribution of entire dataset containing all distortion types. When the model converges, a limitation problem has been exposed: the performance cannot be improved significantly and the model remains saturated at this time. Unexpectedly, such limitations cannot be tackled by simply adding more layers. In fact, when training dataset is given, its distribution is objective and fixed. In theory, for neural network with the same architecture, no matter what learning approach is adopted, the network should converge to the same optimal. However, as reported in [10], different methods will lead to distinct difficulties for the network to fit the data distributions.

In this paper we tackle this problem from another perspective: without designing a new neural network structure but by optimizing the learning scheme. We present a novel learning approach named iterative learning to accumulate the weight importance learned in the pioneer episodes, and guide the optimizer to the next optimal solution based on the acquired knowledge. Iterative learning can provide more information for the learning of subsequent samples based on the weight importance accumulation, which is conducive to the generalization of model on more distortion types if there is enough training data. Since there is only one public available dataset which contain only four distortion types, we also introduce a more challenging benchmark named DLD. It contains ten kinds of distortion crossing large variations in degree and position. It can be used as a new benchmark to fill the vacuum.

The contributions in this paper include: 1) We propose a novel iterative learning scheme to optimize distorted image restoration. 2) We create a new benchmark on this domain which is publicly available. 3) Extensive experiments demonstrate that incorporating our learning approach into the state-of-the-art models consistently improve its performance on distorted image restoration benchmarks.

2. RELATED WORK

Distorted Image Restoration. Recently, some end-to-end deep learning methods that cope with distorted image rectification have been proposed. Ma et al. [9] first proposes DocUNet to predict the forward mapping with the implementation of two stacked U-Nets. DewarpNet [7] considers the 3D geometric property and introduce the stacked 3D and 2D regression networks. CREASE [8] introduces a new per-pixel angle loss which provides image rectification maps to optimize the rectification model. All methods focus on model

*Jie Li is the corresponding author. Chao Wang and Yi Gu are co-first authors of the manuscript. This work has been partially supported by the National Key Research and Development Program of China Nos. 2020YFB1806700, NSFC Grant 61932014, Project BE2020026 supported by the Key R&D Program of Jiangsu, China.

structure while following the traditional learning setting.

Learning Scheme. As the standard learning approach, joint learning [11, 12, 13] is still the first choice for automatic distorted image restoration. It uses all task data to train the neural network to fit the distribution of all tasks. But as the task number increases, the hidden distribution becomes difficult to fit, for the reason that joint learning just simply amalgamates the datasets deriving from different tasks into a huge one. Recently, some efforts are made to explore other optimization probabilities. Transfer learning [14, 15, 16] utilizes the parameters of trained model to the new model as initial learning setting, which optimizes and speeds up the learning efficiency. Lifelong learning [17, 18, 19] learns new tasks with only access to new task data while not forgetting previous tasks. Meta-learning [20, 21, 22] observes learning processes of different learning methods on multiple tasks, and adapts to new tasks based on meta-data. However, transfer learning and meta learning aim to adapt to new tasks quickly while forgetting the capability of solving historical tasks, lifelong learning seems to remember past tasks but does not share classes across the tasks. All these three methods are not suitable for the setting where all tasks with all classes should be learned like joint learning without any artificial class partition.

3. ITERATIVE LEARNING SCHEME

3.1. Weight Importance Accumulation

We suppose a distortion dataset T to be learned. We randomly split T into a series of subtasks, and each subtask may contain one or more distortion types. Suppose T is divided into a group of n subtasks, $T = \{T_1, ..., T_n\}$. Each task T_i consists of n_i paired instances, $T_i = \{X_i^j, Y_i^j\}_{j=1}^{n_i}$, where X_i and Y_i denote the samples of source and target domain respectively. All subtasks are sequentially fed into the network, and only one main subtask participates in the learning in each iteration. Note that this setting is not technically incremental and distortion types of subtasks may be mixed with each other.

In order to endow each subtask with a local weight importance in solving historical subtasks. We do not regard the path integral of weights as a surrogate for past task loss function but as a regularization of prior knowledge. Based on the inference of two independent distributed tasks, we analyze the situation of multiple arbitrary tasks, and we train the network by initializing the network weights in a iterative ways. Suppose that there is a task which needs to fit the joint distribution $F(X, Y)$ on the dataset D, then we split the learning task into multiple tasks which like

$$\begin{aligned} Epoch_1: &\quad (F_1; T_1) \\ Epoch_2: &\quad (F_2, F_1; T_2, T_1) \\ &\quad \dots \\ Epoch_n: &\quad (F_n, F_{n-1}, \dots F_1; T_n, T_{n-1} \dots T_1) = (F; T) \end{aligned}$$

These n epochs need to meet the following requirements: 1) Each epoch should find which weights are local convergence, as known as the importance of the weight. 2) Let unimportant weights to fit the joint distribution.

We utilize the path integral of the weights to tell the importance of the weights. The new gradient search for the local convergence of each weight is formulated as:

$$\int_C \boldsymbol{g}(\boldsymbol{\theta}(t))d\boldsymbol{\theta} = -\sum_k \omega_i^k = \sum_k \nabla\theta_i^k \cdot (\theta_i^k - \theta_{past}^k), \quad (1)$$

where ω_i^k reflects the importance of the *k th* parameter in the network of the past tasks, i represents the current task and $\int_C \boldsymbol{g}(\boldsymbol{\theta}(t))d\boldsymbol{\theta}$ is the curve integral of model weight θ.

The learning process of the neural network is essentially to minimize the loss of between the predict value and the ground truth. In oder to maintain the important converged weights on past tasks, we add a penalty Ω. According to the calculation of the ω_i^k in Equation 1, we know that the value of ω_i^k of the converged weights is smaller than which are not converged. Therefore, we can let it be negative, and the penalty term is then defined as

$$\Omega = \sum_i \frac{\omega_i}{(\theta_i - \theta_{past})^2 + \varepsilon}. \quad (2)$$

where i is the *i th* subtask and ε is a constant.

For n epochs, in the current epoch, model weights which have been updated sightly in the previous epochs will fit the overall distribution of all tasks containing the new added subtask. And in the next epoch, locally converged weights will still maintain. This method can accumulate critical knowledge from previous tasks on-line and provide updated directions for each gradient search in current learning.

3.2. Weight Importance Guided Modification

To help the data flowing across subtasks, we shuffle between historical training samples and current ones. When learning a new subtask T_i in the *i th* iteration, the model randomly samples part of data from past sub-dataset T_{past} according to the sampling rate α. The model shuffles the current samples $\{X_i, Y_i\}$ and the past samples $\alpha\{X_{past}, Y_{past}\}$, and reorganizes T_i into a modified training set $T_i^{'}$ to be learned.

The network of the *i th* iteration H_i learns the training set $T_i^{'}$ and converges to the optimal to replicate T_i's real data distribution $\mathbb{P}_i$, where $\mathbb{P}_i$ is the set of data distributions of the first i subtasks. The most crucial part of our algorithm is the knowledge transfer used to assist in searching for the optimal in the new feature space. If only using the gradient descent method, due to the learning rate and parameter initialization, it is easy to be affected by the local optimal position in the process of finding the global extreme one.

When learning task i, we initialize the previously learned weights to the current network, and then add a penalty term to the weights that exert a crucial impact on previous tasks. This term will change less during optimization to achieve weight

importance guided modification. In order to strengthen this effect, we randomly sample the data set of past tasks and put it into the data set of the current task. In the process of knowledge transfer, using previous task samples can better guide the model to converge to the optimum.

The original optimization uses SGD to optimize the parameters of the network in the whole dataset. We redefine this learning process and perform iterative learning of sub-tasks. In the first sub-task, we initialize the parameters randomly, and let $\Omega_0 = 0$. After learning we update the Ω_1 through the path integral of parameters on time which guides the move direction during the learning process. In the next iteration, we first shuffle the subtask as mentioned above, then we initialize the parameters by the past results of subtasks and train the network based on the modification loss where important parameters move little during gradient descent. Finally, we calculate the Ω of the next round. The overall iterative learning scheme can be summarized in Algorithm 1.

Algorithm 1: Iterative learning scheme

for *N subtasks* **do**
  if *subtask* == T_1 **then**
    $\Omega_0 = 0$;
    Train the network : θ_1;
    update for all parameters $\Omega_1 : \Omega_1^k \leftarrow \frac{\nabla\theta_1^k \cdot (\theta_1^k - \theta_0^k)}{(\theta_1^k - \theta_0^k)^2 + \varepsilon}$;
  else
    randomly sample episodes from T_{past};
    Reorganize $T_i : T_i^{'} \leftarrow T_i + \alpha T_{past}$;
    initialize $\theta_i : \theta_i \leftarrow \theta_{past}$;
    calculate $loss^{'} : loss^{'} \leftarrow loss + \frac{\lambda}{2} \sum_j \Omega_{past}^j (\theta_j - \theta_{past,j}^*)^2$;
    Train the network on $T_i^{'} : \theta_i$;
    update the importance Ω_{past} for all parameters : $\Omega_{past}^k \leftarrow \Omega_{past}^k + \frac{\nabla\theta_i^k \cdot (\theta_i^k - \theta_{past}^k)}{(\theta_i^k - \theta_{past}^k)^2 + \varepsilon}$;
    update $T_{past} : T_{past} \leftarrow T_i + T_{past}$;
  end
end

4. EXPERIMENT

Dataset. To the best of our knowledge, DFD [23] is the only public available dataset for this task while containing only four types of distortions. Therefore, we create a complex dataset as a supplementary benchmark for evaluation. We consider ten common types of distortion: affine, barrel, piecewise linear, pin cushion, projective, similarity, sinusoid, skew, wave and non-reflective similarity.

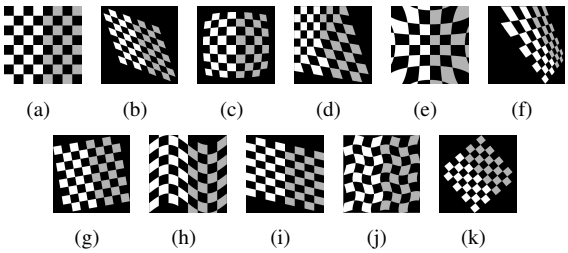

Fig. 1. Some image distortion results. (a) Original image. (b) Affine. (c) Barrel. (d) Piecewise linear. (e) Pin cushion. (f) Projective. (g) Similarity. (h) Sinusoid. (i) Skew. (j) Wave. (k) Non-reflective similarity.

We use a two dimensional geometric transformation to achieve these distortions. This transformation is a mapping that associates each point in a Euclidean plane with another point in the same plane. In these distortions, the geometric transformation is defined by a rule that tells how to map the point (x, y) to another point (u, v) with Cartesian coordinating. We choose the following geometric transformation method for modification, an it can be formulated as $[up\ vp\ wp] = [x\ y\ w] \times T$, where $u = \frac{up}{wp}$, $v = \frac{vp}{wp}$, w and p are coefficients for 3D projection space. T is a matrix $(a_{ij})_{3\times3}$ different for each distortion entry. Therefore, the overall transformation for each point can be formulated as:

$$\begin{cases} u = \dfrac{a_{11}x + a_{21}y + a_{31}}{a_{13}x + a_{23}y + a_{33}} \\ v = \dfrac{a_{12}x + a_{22}y + a_{32}}{a_{13}x + a_{23}y + a_{33}} \end{cases} \tag{3}$$

We select GoogleLandmarks-v2 [24] and randomly select 100,000 images as our samples. We use the above formulation to perform ten distortion transformations on these images with random distortion center and scaling. For further study we divide the whole dataset into training and validation sets with nine to one ratio. This new dataset, namely Distorted Landscape Dataset (DLD), is employed as a new benchmark together with DFD to evaluate the restoration model. Figure 1 illustrates some distortion results. Note that for iterative learning, we randomly partition the samples of DFD and DLD into several groups as subtasks.

Comparison Models. Since our approach is not dependent on any specific network architecture, we deploy the following state-of-the-art models as our baseline structure: DocUNet [9], DewarpNet [7] and CREASE [8]. We incorporate our scheme with these representative models, and compare our new integrated models with the prior state-of-the-art ones. For fair comparison, we maintain consistent parameter settings and training techniques for each baseline structure.

Implement Details. We follow the experiment setup of the closely related work for evaluation metrics and baseline implementations. We explore the restoration performance through both qualitative and quantitative analysis. The quali-

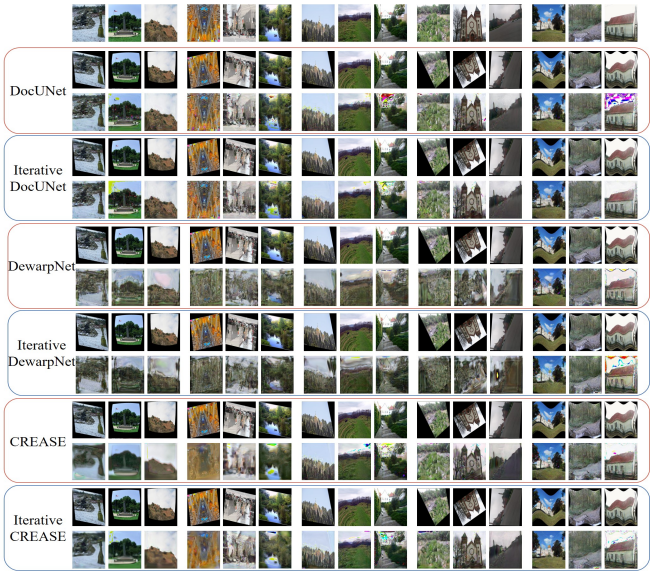

Fig. 2. Comparison of qualitative results on DLD using model DocUNet, DewarpNet and CREASE with different learning approaches. In each column from top to bottom: original, distorted and restoration images, respectively.

tative evaluation mainly relies on visual perception while the quantitative evaluation utilizes PSNR and SSIM to quantify the performance. Each baseline model is trained on the entire dataset while for weight importance equipped versions, the training set is divided into a sequence of subtasks at random to which each model is exposed. After learning the sequence of all subtask sets, we evaluate the ultimate performance of each model on the validation set. In order to make a fair comparison with the state-of-the-art methods, we ensure that all models have been trained to convergence with the same training epochs and no image exists in the validation set and training set simultaneously. All experiments are conducted on NVIDIA RTX 2080Ti GPUs.

Quantitative results. We report the quantitative evaluation on the benchmarks DFD and DLD, as shown in Table 1. The evaluations on DFD and DLD show that the iterative learning method obtains the highest numerical results in terms of PSNR and SSIM on all models. It illustrates the good generalization of our approach on all kinds of distortions. However, the restoration results of all models on DLD decrease in both metrics compared with those of DFD, which reflects the difficulty of our proposed benchmark and the large gap between DFD and DLD. We can conclude that our method surpasses all other learning methods and achieves great improvements across all the models.

Qualitative results. To visually explore the performance of different approaches on distorted image restoration, we take the promising results of DLD benchmark as one example, as presented in Figure 2. The visual results in the bottom panel of each block in the figure show that all iterative

Model	Method	DFD		DLD	
		PSNR	SSIM	PSNR	SSIM
DocUNet	Baseline	22.28	0.64	15.37	0.46
	IL	27.25	0.85	22.59	0.64
DewarpNet	Baseline	23.87	0.71	20.29	0.55
	IL	28.35	0.87	22.60	0.66
CREASE	Baseline	25.04	0.74	20.97	0.57
	IL	30.05	0.91	23.92	0.72

Table 1. Quantitative evaluations in terms of PSNR and SSIM on benchmark DFD and DLD. Higher values are better.

Method	PSNR	SSIM
DocUNet	15.37	0.46
DocUNet+Shuffle	17.65	0.50
DocUNet+WIGM	20.98	0.57
DocUNet+Shuffle+WIGM	22.59	0.64

Table 2. PSNR and SSIM results of each module on Iterative DocUNet trained on DLD. Higher values are better.

models obtain the perceptually convincing images and perform better than their corresponding baseline models. Some reinforced generation artifacts exist in the produced outputs of baseline models which further illustrates that our method has good generalization on a variety of distorted images. Besides, it is worth noting that the deeper the distortion degree is, the more details are lost in the restored image. Overall, our method achieves the most visually satisfactory results which corrects various distortions and partially or completely supplements the missing details.

Ablation Study. We investigate the effectiveness of each component in the iterative learning. We take the state-of-the-art DocUNet as the baseline model for example. As presented in Table 2, DocUNet reaches the lower bound in terms of PSNR and SSIM. Both the shuffle mechanism and WIGM module increase PSNR and SSIM. However, only WIGM cannot match the distortion task due to the differences and uncertainties between distortion types, and single shuffle is difficult for the model to fit the implicit function mapping. When employing both two modules to the baseline model, it reaches the best performance.

5. CONCLUSION

In this paper, we explore the challenging distorted image restoration task. In order to fill the vacuum in the public available dataset, we release a new dataset containing ten types of distortions. We analyze the reasons for model saturation and propose a knowledge-driven iterative learning scheme. We propose a weight importance guided modification to accumulate the crucial knowledge and improve the gradient descent search for future learning. We use the state-of-the-art models as the baseline and evaluate all learning strategies on the benchmarks. Extensive experiments demonstrate that our iterative learning method significantly improves the performance of existing models, and can be easily integrated with other models on different domains in a plug-and-play manner.

6. REFERENCES

[1] Gavin Cromhout, Josh Fallon, Nathan Flood, Katy Freer, Jim Hannah, Francine Spiegel, Pete Walsh, and James Widegren, "Liquifying faces," in *Photoshop Elements 2 Face Makeovers*, pp. 135–152. Springer, 2003.

[2] George J Kingsnorth, Gavin Cromhout, Janee Aronoff, Dan Caylor, and Pete Walsh, "Effects and filters," in *Photoshop Elements 2 Tips and Tricks*, pp. 80–117. Springer, 2003.

[3] Colin Smith and Al Ward, "Magic and monsters," in *Photoshop Most Wanted 2: More Effects and Design Tips*, pp. 83–93. Springer, 2003.

[4] Adobe Illustrator, "Photoshop," *Graphpad Prism for*.

[5] Katrin Eismann and Wayne Palmer, *Photoshop restoration & retouching*, Peachpit Press, 2006.

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[7] Sagnik Das, Ke Ma, Zhixin Shu, Dimitris Samaras, and Roy Shilkrot, "Dewarpnet: Single-image document unwarping with stacked 3d and 2d regression networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 131–140.

[8] Amir Markovitz, Inbal Lavi, Or Perel, Shai Mazor, and Roee Litman, "Can you read me now? content aware rectification using angle supervision," in *European Conference on Computer Vision*. Springer, 2020, pp. 208–223.

[9] Ke Ma, Zhixin Shu, Xue Bai, Jue Wang, and Dimitris Samaras, "Docunet: document image unwarping via a stacked u-net," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4700–4709.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[12] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[13] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[14] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2010.

[15] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell, "Deep domain confusion: Maximizing for domain invariance," *arXiv preprint arXiv:1412.3474*, 2014.

[16] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan, "Learning transferable features with deep adaptation networks," in *International conference on machine learning*. PMLR, 2015, pp. 97–105.

[17] Zhizhong Li and Derek Hoiem, "Learning without forgetting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[18] Friedemann Zenke, Ben Poole, and Surya Ganguli, "Continual learning through synaptic intelligence," in *International Conference on Machine Learning*. PMLR, 2017, pp. 3987–3995.

[19] Chenshen Wu, Luis Herranz, Xialei Liu, Joost van de Weijer, Bogdan Raducanu, et al., "Memory replay gans: Learning to generate new categories without forgetting," in *Advances in Neural Information Processing Systems*, 2018, pp. 5962–5972.

[20] Sachin Ravi and Hugo Larochelle, "Optimization as a model for few-shot learning," 2016.

[21] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra, "Matching networks for one shot learning," *arXiv preprint arXiv:1606.04080*, 2016.

[22] Chelsea Finn, Pieter Abbeel, and Sergey Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1126–1135.

[23] Yi Gu, Yuting Gao, Jie Li, Chentao Wu, and Weijia Jia, "Distorted image restoration using stacked adversarial network," *arXiv preprint arXiv:2011.05784*, 2020.

[24] Tobias Weyand, Andre Araujo, Bingyi Cao, and Jack Sim, "Google landmarks dataset v2-a large-scale benchmark for instance-level recognition and retrieval," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2575–2584.