Manifold Transfer Networks for Lens Distortion Rectification

Li Jing
MIT Physics
ljing@mit.edu

Lay Jain
MIT EECS
layjain@mit.edu

Rumen Dangovski
MIT EECS
rumenrd@mit.edu

Marin Soljačić
MIT Physics
soljacic@mit.edu

***Abstract*—Convolutional neural networks (CNNs), well-known for their translational invariance property on translational manifolds, are not guaranteed to generalize to images on other types of manifolds. Existing works extending CNNs' translational invariance property are limited to linear transformations such as rotation. We propose a novel framework, the *Manifold Transfer Network*, with an embedded inductive bias for any specified nonlinear manifold. Our model maps a nonlinear transformation to a linear translation on a translational manifold, making it suitable for a CNN to learn and predict. We design such a map through the solutions of a particular class of partial differential equations. We empirically apply our method to the domain of radial lens distortion rectification. In our experiments on the CelebA dataset we demonstrate superior performance of our model compared to conventional baselines.**

***Index Terms*—convolutional neural networks, image rectification, manifold learning**

Convolutional neural networks (CNNs) have been considered the default toolbox for a wide range of computer vision applications [21, 13, 12, 27, 14]. This is due to their intrinsic inductive bias matching the nature of 2D vision. However, CNNs only guarantee generalization ability within the dataset with translational invariance. This is not a good match for other types of transformations such as rotation or distortion.

Many works traditionally have tried to extend such abilities to other types of transformations. Group equivariant convolutional networks [4, 7] extend CNNs to tolerate rotation and mirror transformation. Spherical CNNs [5, 9, 20] achieve SO(3) symmetry on CNNs and hence generalize to spherical images. Another approach following the Spatial Transformer Networks [18, 19] uses an intermediate transformation to map images to a new coordinate system so that they gain translational symmetry. Furthermore, [8] use polar coordinates to map rotations into translations and [32] extends this approach to general linear transformations.

In this work, we develop a theory that finds the transfer manifold for a general transformation. This transformation will map a specified transformation to a simple shift with distances proportional to the transformation coefficient elements, which Figure 1 demonstrates. As a result, the output feature map will be suitable for CNNs to learn and predict. Coming from another line of work, self-supervised learning [1] is the state-of-the-art method for pertaining feature maps on synthetically modified data using automatically generated tasks and large quantities of unlabeled data. Instead, the goal of our theory is to propose a parallel line of methods that can learn useful feature maps based on physics-inspired manifold transfer. More specifically, we propose a method to find such transfer through partial differential equations. This transfer only needs to be performed on the data before it is fed into the network. Hence no gradient step is required and approximation can be applied to simplify the solution.

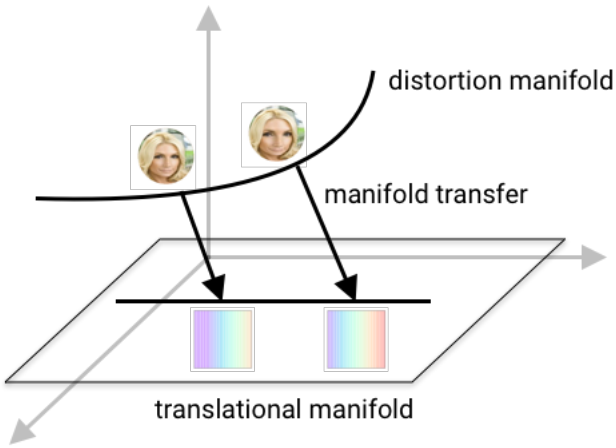

Fig. 1: An image and its corresponding distorted images with different distortion coefficient form a manifold in the high dimensional data space. Under manifold transfer, each point on this distortion manifold is mapped to a new point on a translational manifold. Each image represented by a point on this manifold can be derived from another by a simple shift. Hence, the transformed manifold will be suitable for a CNN to learn and predict due to its translational invariance.

We apply our model to lens distortion rectification. Real-world images captured by cameras with wide angles or under short distances violate pinhole camera assumptions and suffer from lens distortions. To automatically rectify distorted images that apply to general physical environments remains an important problem for the computer vision community [26, 34, 22]. Traditional methods applying prior knowledge to design special features are widely implemented in real-world applications. They typically use sharp edges [24] and other human-designed features [17] to develop a rectification transformation. For instance, [11] detect key points on faces to find the corresponding transformation. Applying CNNs to directly predict distortion coefficients has become a state-of-

the-art approach. The authors of [37] develop a CNN model based on large scale synthesized images; [35] utilize semantic segmentation as extra information to guide rectification process. These models are easy to build and deploy and achieve better performance compared to human-designed methods. However, these models do not generalize outside the scope of the training data due to the limitation of regular CNNs as discussed above. Here, we solve this problem through a manifold transfer network. We use synthetically distorted images from the large scale CelebA dataset [23] to train our model and test its performance. We compare our model to two baseline models: a standard CNN and an ablation model, and demonstrate its superior performance.

Our contributions are summarized as follows:

1) We develop a general theory to assist CNNs to go beyond translational invariance through manifold transfer.
2) We propose an approach to find such a manifold transfer method by solving a class of partial differential equations.
3) We apply our method to lens distortion rectification and demonstrate performance superior to state of the art.

I. Related Work

A. Group Equivariant Networks

Group equivariant models [4, 7] use embedded group equivariant convolutions to tolerate rotation transformations without the need of data augmentation. Spherical CNNs [5, 9, 20] extend such ability to SO(3) transformations and hence are suitable to spherical images. There are two major limitations of these approaches. First, limited types of operations can be developed to be embedded into networks. Second, high memory and latency overhead are required to support these models.

B. Spatial Transformation Networks

Standardizing images before feeding them into standard CNNs is another popular approach to tackle various distorted or transformed images. *Spatial* transformer networks [18] use an extra network to learn pose parameters for transforming distorted images back to standard. The standard image is then sent to a standard CNN, which yields higher accuracy and lower data requirements. Several follow up works, e.g. *polar* [8] and *equivariant* [32] transformer networks, apply specific transformations before applying pose predictors. Equivariant transformer networks learn general linear transformation parameters and are pipe-lined to rectify each of the distortions.

Our work is similar to this approach but we do not rely on classification targets to provide training guidance. Instead, we directly apply the spatial transformer framework to predict corresponding parameters following self-supervised learning approaches[15, 3, 25, 6]. As a result, our model can generalize to other downstream tasks. Furthermore, existing models are restricted to linear transformations or well-studied group transformations, while to the best of our knowledge, we are the first to extend it to general nonlinear transformations.

C. Distortion Rectification

Rectifying lens distorted images have a long history in image processing, yielding numerous applications. Traditional methods [17, 36, 37] detect handcrafted features from images based on prior knowledge. The authors of [16] detect horizons from images and calculate corresponding camera parameters. The authors of [11] propose to search key points for face image rectification. These methods are limited to specific data domains and are not robust against complicated environments.

Methods based on machine learning [28, 35] utilize large scale synthesized data to embed prior knowledge into models. They synthetically distort large amounts of images and train a deep learning model to reveal the corresponding distortion coefficient. In [35] the authors use semantic information as extra information to help detect distortion. They train the model in an end to end style with the distortion coefficient as intermediate supervision. In [39] the authors use spatial transformer frameworks [18] to learn and predict rectification mapping on face images. These methods are only robust within domains of images with sharp lines due to the translational invariance property of CNNs. However, no evidence exists to show these models can generalize to domains without such patterns, such as the CelebA dataset.

II. Manifold Transfer Theory

A. Manifold Transfer Operation

We assume all our N datapoints $\{\mathbf{x}_j\}_{j=1}^N$ are on a manifold in the data space, i.e. $\mathbf{x}_j \in \mathbb{R}^{H\times W\times 3}$, where H is the height and W is the width of the image in pixels, while 3 is the dimension of colors. We define a manifold operator T such that for each datapoint x on this manifold, the new datapoint x' defined by satisfying the *manifold invariance condition*

$$x'(T(\mathbf{r})) = x(\mathbf{r}) \tag{1}$$

for all $\mathbf{r} \in [0, W] \times [0, H]$ is still on this manifold.

Our assumption is that our input dataset relies on a manifold with operator T_i ("i" stands for input), which is parametrized by a vector $\boldsymbol{\phi} = [\phi_1, \phi_2] \in \mathbb{R}^2$. We want to transfer our dataset onto a new manifold with operator T_o ("o" stands for output) that satisfies equation 1. This transfer procedure can be represented by another transformation T_c ("c" for composition). In order to keep the transformations consistent, we requires T_c to satisfy the following requirement: for every data point in the input space, and for every pixel $\mathbf{r}$ on its feature space, the effect of going through transformation T_c after the original transformation T_i is identical as going through the transformation T_o after T_c. That is,

$$T_c \circ T_i[\boldsymbol{\phi}](\mathbf{r}) = T_o[\boldsymbol{\phi}] \circ T_c(\mathbf{r}) \tag{2}$$

for every $\mathbf{r} \in [0, W] \times [0, H]$. Here, transformations T_i and T_o are parameterized by the same vector parameter $\boldsymbol{\phi}$.

Specifically, we are interested in having the output as a linear translation which matches the inductive bias of CNNs, i.e.

$$T_o[\boldsymbol{\phi}](\mathbf{r}) = \mathbf{r} + \sum_j \phi_j \mathbf{e}_j \tag{3}$$

where $\{\mathbf{e}_j\}$ are the unit bases in the space of T_o (in theory we can consider a space of an arbitrary dimension M, i.e. $\boldsymbol{\phi} = [\phi_1, \ldots, \phi_M]$, even though in this paper we focus on two dimensions).

An intuitive example is that for a rotation $T_i[\boldsymbol{\phi}] \equiv T_i[\phi]$ parametrized by $\boldsymbol{\phi} = [0, \phi]$, where ϕ is a scalar. Its corresponding manifold transfer is in polar coordinates. In this case, $T_i[\phi](x, y) = (x\cos\phi - y\sin\phi, y\cos\phi + x\sin\phi)$, $T_o[\phi](x, y) = (x, y + \phi)$ and hence $T_c(x, y) = (\sqrt{x^2 + y^2}, \arg(y/x))$. Here, we propose a method to find the solution for general $T_i[\phi]$.

B. Deriving the Transfer

It is difficult to derive an analytic solution of T_c for arbitrary T_i, unless T_i is linear. Similar to [32], we propose a solution to this problem using partial differential equations to solve the nonlinear cases.

Given equation 3, we want for every dimension k in the output space,

$$\frac{\partial}{\partial \phi_j}(T_c \circ T_i[\phi](\mathbf{r}))_k = \delta_{kj} \tag{4}$$

where δ_{kj} is the Kronecker operator. More specifically,

$$\sum_l \frac{\partial \mathbf{r}_l}{\partial \phi_j} \frac{\partial}{\partial \mathbf{r}_l} T_c(\mathbf{r}')_k = \delta_{kj}, \tag{5}$$

where

$$\mathbf{r}' = T_i[\phi](\mathbf{r}). \tag{6}$$

In the following section, we will apply our theory to an application and solve equation 5 through an approximation.

III. Application to Lens Distortion

A. Lens Distortion

Images are usually represented as (2+1)D tensors with their coordinates represented by an index. In contrast, lens distorted images are spherically represented images projected onto a flat plane with coordinates mapping as projection [2, 10, 31, 33, 38]. That is $T_i(\mathbf{r}) = \tan(r/d)\mathbf{r}$. For image sizes that are larger than the lens radius d, it is appropriate to use first-order approximation, i.e. we only keep the lowest degree radial term, which is $O(r^2)$. Moreover, here we only consider radial distortion which means the angle to the origin remains constant during such a transformation.

Thus, setting $\boldsymbol{\phi} = [\phi, 0]$, the formula of the above lens transformation becomes approximately as follows

$$T_i[\boldsymbol{\phi}](\mathbf{r}) \equiv T_i[\phi](\mathbf{r}) = (1 + \phi r^2)\mathbf{r}, \tag{7}$$

where $\mathbf{r} = (x, y) \in [-1, 1] \times [-1, 1]$ is the relative coordinate of the original image with center at [0, 0], $r = (x^2 + y^2)^{1/2}$ is the absolute value of $\mathbf{r}$ and $\phi = (1/d^2) \in [0, \infty)$ is the distortion coefficient.

B. Manifold Transfer Solution

Here, we derive an approximate solution for lens distortion transformation. Combining Equations 7, 2 and 3, we obtain

$$T_c\left((1 + \phi r^2)\mathbf{r}\right) = T_c(\mathbf{r}) + \phi \mathbf{e_1}. \tag{8}$$

Since there are multiple solutions to 8, we choose the simplest one in which T_c is independent of the direction of $\mathbf{r}$. We can isolate the variables in the equation by $r = |\mathbf{r}|$ and the angle to the origin. Therefore there exists a solution where the first element of $T_c(\mathbf{r})$ only depends on r and the second element only depends on the angle to the origin, i.e. we consider the following ansatz

$$T_c(\mathbf{r}) = [X(r), Y(\theta)], \tag{9}$$

where r is the absolute value of $\mathbf{r}$ and θ is the angle to the origin. We have that

$$X\left(r + \phi r^3\right) = X(r) + \phi \tag{10}$$

for any r. Note that Y can be an arbitrary function, since we do have constraints from equation 8, given equation 9.

Now, we take the Taylor expansion of the LHS of equation 10

$$X(r + \phi r^3) = X(r) + \frac{dX}{dr}(\phi r^3) + O(\phi r^3)^2$$

and with this substitution, in equation 10 the term $X(r)$ cancels out, and thus after diving by ϕ on both sides we obtain that

$$\frac{dX}{dr} r^3 + \frac{O(\phi r^3)^2}{\phi} = 1. \tag{11}$$

To solve equation 11 we assume that r is small enough, so that the term $O(\phi r^3)^2/\phi$ vanishes. This is a reasonable assumption in our experiments, since the sizes of the images that we consider are small. With this assumption we derive an approximate solution of equation 11 as follows

$$X(r) = C - 1/(2r^2) \tag{12}$$

for any constant C.

In practice, we need to sample transformed images through $T_c(\mathbf{r})$ instead of sampling through $\mathbf{r}'$. i.e. when sampling an output pixel value $\mathbf{r}$, we are looking for the corresponding coordinate $\mathbf{r}'$ in the original input image. Therefore, we need to find the solution of $\mathbf{r} = T_c^{-1}(\mathbf{r}')$ where $\mathbf{r}'$ represents pixels in the output feature map. Hence, we obtain the final transformation formula as follows

$$T_c^{-1}(x, y) = \left[\sqrt{\frac{1}{2(C - x)}}\sin(y), \sqrt{\frac{1}{2(C - x)}}\cos(y)\right] \tag{13}$$

where C is arbitrary and we will tune it as a hyperparameter. It it straightforward to confirm that $X(T_c^{-1}(x, y)) = x$ and since we can choose Y freely, we set $Y(\theta) := y$, hence using equation 9 we obtain that

$$T_c\left(T_c^{-1}(x, y)\right) = (x, y),$$

as desired.

We demonstrate the effectiveness of the manifold transfer networks in Figure 2. The first row demonstrates one input image and its corresponding distorted heatmap. The second row demonstrates the corresponding feature maps after the transformation T_c. The feature maps become a linear shift along the x-axis. This property is suitable for CNNs to learn and share certain features within every layer of feature maps.

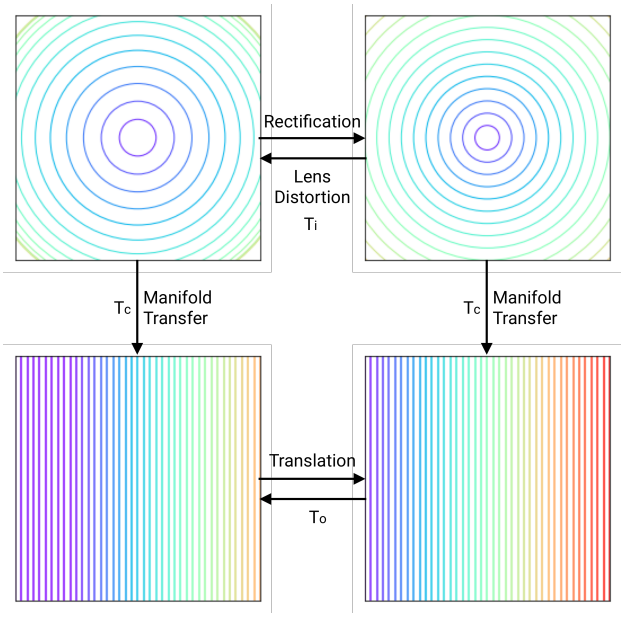

Fig. 2: The heatmap of a standard image and its corresponding lens distorted one. After a manifold transfer, the new heatmap becomes a simple shift of the original, which reflects equation 2.

C. Centroid Prediction

In order to assist CNNs to predict coefficients and take advantage of the linear translational property, we use the centroid layer proposed in [8]. The definition of the centroid layer is shown in Algorithm 1. It multiplies a softmax of the input and a positional encoding value along the last dimension. Compared to simple softmax (used by [28]), the centroid layer takes advantage of the relative spatial information of different slots and fully utilizes input information. Furthermore, a centroid layer is differentiable and more stable in practice.

Input : A Tensor x with last dimension representing the coefficient shift.
Output: A Tensor y with same dimension as x without last dimension
$\mathsf{d} \leftarrow \mathsf{x}.shape[-1]$
$\mathsf{x} \leftarrow \texttt{softmax}(\mathsf{x}, axis = -1)$
$\mathsf{rg} \leftarrow [1/\mathsf{d}, 2/\mathsf{d}, ..., 1]$
$\mathsf{y} \leftarrow \mathsf{x} * \mathsf{rg}$
$\mathsf{y} \leftarrow \texttt{average}(\mathsf{y}, axis = -1)$

Algorithm 1: Centroid Layer

D. Rectification Transformation

Once we get the distortion coefficient, the last step is to apply the inverse lens distortion transformation: the rectification transformation $T^{-1}[\phi]$. We derive this analytically after solving the cubic equation equation 7 and obtain the following

$$T^{-1}[\phi](\mathbf{r}) = \frac{(q + \sqrt{q^2 + p^3})^{\frac{1}{3}} + (q - \sqrt{q^2 + p^3})^{\frac{1}{3}}}{r}\mathbf{r} \quad (14)$$

where $p = 1/(3\phi)$, $q = r/(2\phi)$. Similar to the original lens distortion, this rectification transformation only acts on the radial dimension.

E. Rectification Networks

In this section, we demonstrate the architecture of manifold transfer networks. During training, we directly supervise the network to predict the distortion coefficient from a distorted image. During the rectification runtime, as shown in Figure 3, the coefficient from the network is used to generate a sampling grid. Then the rectified image is sampled from the original distorted image on this grid.

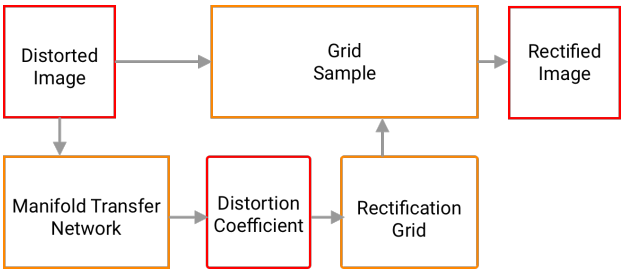

Fig. 3: The runtime model for image rectification. A manifold transfer network predicts the distortion coefficient which is used to generate a mapping grid. Then the distorted image is sampled onto this grid to generate the rectified image.

The prediction network contains three parts: a manifold transfer, a CNN and a centroid prediction layer. See Table I. The original distorted image is sampled onto the new grid using a transformation based on equation 13. It is then followed by five blocks of Conv-BatchNorm-ReLU similar to the VGG architecture [30]. Each convolution uses a 3x3 kernel with a 2x2 stride except the last layer with stride 1x2. The linear layer is then used to mix up the y-axis which corresponds to the radial dimension in the original image. Lastly, the centroid layer followed by a linear layer is used to predict the final coefficient.

We use two baseline models in our experiments. First, a standard CNN model without the manifold transfer layer or the centroid prediction layer. Second, a polar transformer model similar to [8] that uses a simple polar coordinate transformation instead of our manifold transfer. This polar transformation also maps lens distortion onto the x-axis of the output, but not to a linear translation, which yields complicated distortion dynamics.

Layer	Output Dimension
Transfer	224x224x3
Conv2d	112x112x3
Conv2d	56x56x32
Conv2d	28x28x32
Conv2d	14x14x32
Conv2d	7x14x32
Linear	14x32
Centroid	1x32
Linear	1

TABLE I: Manifold transfer network architecture. The *Transfer* layer is the non-parametric mapping discussed above. The middle part of the architecture is a standard CNN with groups of convolution and batch normalization. The last *Centroid* layer is a replacement of *softmax* which helps predicting the coefficient.

IV. Experimental Results

We first show our data preparation and training procedures. Then qualitatively and quantitatively we evaluate a manifold transfer network versus a standard CNN and an ablation model.

Our model is implemented using the PyTorch framework[1]. All images are formalized into a resolution of 224x224. The experiments are carried out on two Nvidia 1080 GPUs.

A. Data Preparation

We built our training dataset based on the large-scale CelebFaces Attributes (CelebA) Dataset [23]. CelebA contains more than 200K celebrity images covering large pose variations and background clutter. We synthetically distort each image based on Equation 7 with a randomly sampled distortion coefficient. Specifically, each distortion coefficient is uniformly sampled within the range $[0, 1.0]$. We build a distortion grid with respect to each distortion coefficient. Each distortion grid has a size 224x224. Then the distorted images are sampled by these grids using bilinear interpolation. In order to solve the edge issue, we mask out the pixels outside the circle of $r = 1$.

In our training process, we use 100k images for training, 20k images for validation and 20k images for testing.

Each image is also associated with its corresponding distortion coefficient. In addition, for the test set, the original images are sampled to 224x224 without distortion as ground truth, which is used for rectification evaluation quantitatively and qualitatively.

B. Training the Coefficient Predictor

We use the Mean Squared Error (MSE) as a loss function to supervise the distortion coefficient prediction directly. In the training process, we use an Adam optimizer with a learning rate 0.001 and a batch size 32. Each model is trained for 200 epochs with early stopping based on the error on the validation set.

[1]https://github.com/jingli9111/manifold-transfer-networks

MSE scores on the test set show that compared to standard CNNs and the ablation model, our manifold transfer network achieves significantly higher prediction accuracy.

Model	MSE (10^{-3})
Standard CNN	1.85
Polar Transformer	1.77
Manifold Transfer	**0.09**

TABLE II: The MSE score on the test set measures the prediction accuracy of the distortion coefficient of a) a standard CNN b) a polar transformer network c) a manifold transfer network. The manifold transfer network achieves significantly higher prediction accuracy.

C. Images Rectification

We show rectified face images using our manifold transfer and the input images and ground truths. All images are sampled with 224x224 pixels and masked on $r > 1$ to avoid the manifold transfer model taking advantage of just learning the edge.

Figure 4 contains both severely and barely distorted images. The manifold transfer network is able to accurately predict the distortion coefficient and hence rectify the image successfully.

D. Quantitative Evaluation

We quantitatively evaluate the rectification performance of each model by comparing the output images to the ground truth using synthesized test dataset. We use Peak Signal-to-Noise Ratio (PSNR) [2] to measure the difference of the generated images to the corresponding ground truth images. We also use non-rectified images as a baseline. Table III shows that the manifold transfer network model significantly outperforms the standard CNN model and the polar transformer model.

Model	PSNR
No Rectification	16.53
Standard CNN	18.28
Polar Transformer	18.55
Manifold Transfer	**19.59**

TABLE III: Quantitative evaluation of rectification performances of a) a standard CNN model b) a polar transformation model c) a manifold transfer network. The PSNR measures the difference of the rectified images compared the corresponding ground truth. The manifold transfer network significantly outperforms standard CNN model and the polar transformation model.

V. Discussion

In this work, we restrict the discussion to a single parameter, e.g. radial lens distortion. We believe that rectification for other types of distortion will also benefit from our corresponding

[2]The definition is $\mathrm{PSNR} = 20\log_{10}(\mathrm{MAX}) - 10\log_{10}(\mathrm{MSE})$ where $\mathrm{MSE} = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - T(i,j)]^2$

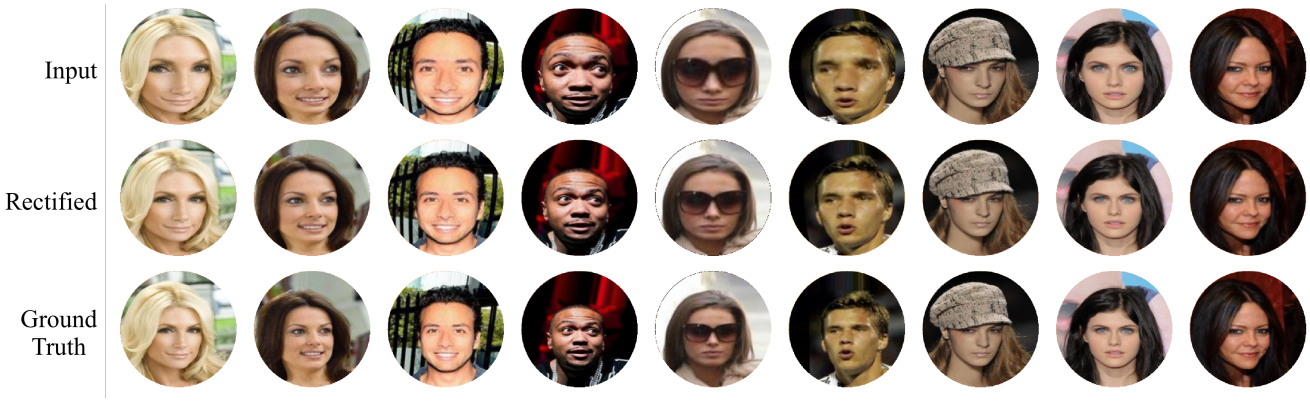

Fig. 4: The rectification performance on synthesized images. The three images in each group represent a) input distorted images b) rectified images from the manifold transfer model c) ground truths. The input are a mixture of images with different distortion coefficients. For example, the last input images are weakly distorted. Manifold transfer network accurately predicts the corresponding coefficient from raw input. Apart from a small scaling differences, the manifold transfer network generates rectified images with satisfactory quality.

manifold transfer networks. For other types of transformation, the theoretical derivation is significantly harder and numerical errors may have stronger influences. Also, pipe-lined rectification modules can be helpful to combine different types of distortion rectification [32].

Our model has superior performance on face images. However, we did not reach robust performance on a general dataset, e.g. ImageNet [29]. We suspect that face images have similar patterns and may store good information under transformations, while general features in ImageNet may be clasped within the manifold transfer. To extend our model to general images remains our future work.

VI. Conclusion

In this paper, we proposed manifold transfer networks that extend CNNs' translational invariance property to nonlinear transformations. The embedded prior knowledge helps CNNs to learn and predict on any nonlinear manifold. We theoretically derived the transformation formula and an approach to find such transfer through partial differential equations. We applied our model to lens distorted face image rectification and achieved superior performance compared to several baseline models. Finally, we claimed that computer vision models embedded with task-specific inductive bias may be a general path to boost CNNs to real-world applications.

Acknowledgments

Research supported in part by the Army Research Office through the Institute for Soldier Nanotechnologies under contract No. W911NF-18-2-0048, in part by the MIT–SenseTime Alliance on Artificial Intelligence, in part upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Agreement No. HR00111890042, as well as in part by MIT AIIA.

References

[1] Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, et al. A cookbook of self-supervised learning. *arXiv preprint arXiv:2304.12210*, 2023.

[2] D. C. Brown. Close-range camera calibration. *Photogrammetric Engineering*, 37:855–866, 1971.

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709, 2020.

[4] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *ICML*, 2016.

[5] Taco Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. *ArXiv*, abs/1801.10130, 2018.

[6] Rumen Dangovski, Li Jing, Charlotte Loh, Seungwook Han, Akash Srivastava, Brian Cheung, Pulkit Agrawal, and Marin Soljačić. Equivariant contrastive learning. *arXiv preprint arXiv:2111.00899*, 2021.

[7] Sander Dieleman, Jeffrey De Fauw, and Koray Kavukcuoglu. Exploiting cyclic symmetry in convolutional neural networks. In *ICML*, 2016.

[8] Carlos Esteves, Christine Allen-Blanchette, Xiaowei Zhou, and Kostas Daniilidis. Polar transformer networks. *ArXiv*, abs/1709.01889, 2017.

[9] Carlos Esteves, Christine Allen-Blanchette, Ameesh Makadia, and Kostas Daniilidis. Learning so(3) equivariant representations with spherical cnns. *International Journal of Computer Vision*, 128:588 – 600, 2019.

[10] W. Faig. Calibration of close-range photogrammetry systems: Mathematical formulation. *Photogrammetric Engineering and Remote Sensing*, 41:1479–1486, 1975.

[11] Ohad Fried, Eli Shechtman, Dan B. Goldman, and Adam Finkelstein. Perspective-aware manipulation of portrait photos. *ACM Trans. Graph.*, 35:128:1–128:10, 2016.

[12] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2013.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask r-cnn. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[15] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. *ArXiv*, abs/1911.05722, 2019.

[16] Yannick Hold-Geoffroy, Kalyan Sunkavalli, Jonathan Eisenmann, Matt Fisher, Emiliano Gambaretto, Sunil Hadap, and Jean-François Lalonde. A perceptual measure for deep single image camera calibration. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2354–2363, 2017.

[17] Ciarán Hughes, Patrick Denny, Martin Glavin, and Edward Jones. Equidistant fish-eye calibration and rectification by vanishing point extraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:2289–2296, 2010.

[18] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. *ArXiv*, abs/1506.02025, 2015.

[19] Sékou-Oumar Kaba, Arnab Kumar Mondal, Yan Zhang, Yoshua Bengio, and Siamak Ravanbakhsh. Equivariance with learned canonicalization functions. In *International Conference on Machine Learning*, pages 15546–15566. PMLR, 2023.

[20] Risi Kondor, Zhen Lin, and Shubhendu Trivedi. Clebsch-gordan nets: a fully fourier space spherical convolutional neural network. *ArXiv*, abs/1806.09231, 2018.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[22] Xiao-Yu Li, Bo Zhang, Pedro V. Sander, and Jing Liao. Blind geometric distortion correction on images through deep learning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4850–4859, 2019.

[23] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[24] Rui Melo, Michel Antunes, João Pedro Barreto, Gabriel Falcão Paiva Fernandes, and Nuno Gonçalves. Unsupervised intrinsic calibration from a single frame using a "plumb-line" approach. *2013 IEEE International Conference on Computer Vision*, pages 537–544, 2013.

[25] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. *ArXiv*, abs/1912.01991, 2019.

[26] James Pritts, Zuzana Kukelova, Viktor Larsson, and Ondřej Chum. Radially-distorted conjugate translations. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1993–2001, 2017.

[27] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2015.

[28] Jiangpeng Rong, Shiyao Huang, Zeyu Shang, and Xianghua Ying. Radial lens distortion correction using convolutional neural networks trained with synthesized images. In *ACCV*, 2016.

[29] Olga Russakovsky, Jun Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252, 2014.

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[31] C. C. Slama. Manual of photogrammetry. *American Society of Photogrammetry*, 1980.

[32] Kai Sheng Tai, Peter Bailis, and Gregory Valiant. Equivariant transformer networks. In *ICML*, 2019.

[33] J. Weng, Cohen. P, and M. Herniou. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:965–980, 1992.

[34] Zhucun Xue, Nan Xue, Gui-Song Xia, and Weiming Shen. Learning to calibrate straight lines for fisheye image rectification. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1643–1651, 2019.

[35] Xiaoqing Yin, Xinchao Wang, J. S. Yu, Maojun Zhang, Pascal Fua, and Dacheng Tao. Fisheyerecnet: A multi-context collaborative deep network for fisheye image rectification. *ArXiv*, abs/1804.04784, 2018.

[36] Xianghua Ying and Zhanyi Hu. Can we consider central catadioptric cameras and fisheye cameras within a unified imaging model. In *ECCV*, 2004.

[37] Mi Zhang, Jian Yao, Menghan Xia, Kuntai Li, Yi Zhang, and Yaping Liu. Line-based multi-label energy optimization for fisheye image rectification and calibration. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4137–4145, 2015.

[38] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, Nov 2000. ISSN 1939-3539. doi: 10.1109/34.888718.

[39] Erjin Zhou, Zhimin Cao, and Jian Sun. Gridface: Face rectification via learning local homography transforma-

tions. In *ECCV*, 2018.