Self-Refining Deep Symmetry Enhanced Network for Rain Removal

Hanrong Ye\textsuperscript{1}, Xia Li\textsuperscript{1}, Hong Liu\textsuperscript{1}, Wei Shi\textsuperscript{1}, Mengyuan Liu\textsuperscript{2}, Qianru Sun\textsuperscript{3}

\textsuperscript{1}Key Laboratory of Machine Perception, Shenzhen Graduate School, Peking University
\textsuperscript{2}School of EEE, Nanyang Technological University
\textsuperscript{3}School of Computing, National University of Singapore

\{leoyhr, hongliu, ethanlee, pkusw\}@pku.edu.cn, nkliuyifang@gmail.com, qianrusun@comp.nus.edu.sg

Abstract

Rain removal aims to extract and remove rain streaks from images. Although convolutional neural network (CNN) based methods have achieved impressive results in this field, they are not equivariant to object rotation, which decreases their generalization capacity for tilted rain streaks. In order to solve this problem, we propose Deep Symmetry Enhanced Network (DSEN). DSEN extracts rotationally equivariant features of rain streaks, and then generates rain layer for image restoration. Furthermore, an efficient self-refining mechanism is designed to remove accumulated rain streaks more thoroughly. Experimental study verifies the validity of our method, with self-refining DSEN yielding the state-of-the-art performance on both synthetic and real-world rain image datasets.

Introduction

As a natural phenomenon, rain degenerates the visibility of images and affects vision algorithms severely. By restoring background information in rain scenes, rain removal is helpful for many computer vision systems including autonomous driving, intelligent photography and surveillance camera. With the rapid development of deep learning, convolutional neural network (CNN) based methods (Fu et al. 2017; Zhang and Patel 2018) have manifested promising performance on this task. However, tilted rain streaks, which are common in real-world rain images, are still difficult to remove. To address this issue, researchers elaborate new network structures and recurrent pipelines (Yang et al. 2017). These methods obtain success from the increase of network capacity, but are restricted by the inherent limitation of prevalent CNNs.

On the other hand, one of the key factors to the success of CNNs is that convolution operation (correlation operation indeed) is equivariant to translation (Goodfellow, Bengio, and Courville 2016), i.e. when the object in input image is shifted, the corresponding output (feature map) is expected to transform in a predictable way. This property ensures that CNNs work well regardless of object’s position. As real-world rain streaks are tilted at various angles, we expect the generated rain layer to follow the rotation of rain streaks. This property is called “rotation equivariance” as shown in the right column of Fig. 1. However, convolution operation in CNNs does not possess rotation equivariance, which decreases the ability of CNNs to learn representation for rotated objects (Weiler, Hamprecht, and Storath 2018; Marcos et al. 2017) including tilted rain streaks. Therefore, existing CNN based methods have difficulty in processing tilted rain streaks by their nature.

To tackle this problem, we enhance CNN with rotation symmetry and propose Deep Symmetry Enhanced Network (DSEN). Our ideas are as follow. First, we introduce a rotationally equivariant convolution layer, which can be conveniently implemented based on the ready-made convolution operation. This special convolution layer is stacked to extract deep symmetry enhanced features of rain. Then, as symmetry enhanced features need to be aggregated before decoding rain layer, we propose a symmetry aggregation block, which avoids the information loss in down-sampling of other methods (Weiler, Hamprecht, and Storath 2018). Finally, we use a regular convolution layer to decode the rain layer for image restoration.

As rain streaks overlap with each other, the accumulation effect makes it difficult to remove all rain streaks in a single stage. In order to solve this problem,
we design a multi-stage framework called self-refining mechanism based on the traits of image restoration task. To utilize information of the previous stage effectively, this framework is designed with novel information links between every two adjacent stages. Moreover, self-refining mechanism can be adapted for both symmetry enhanced CNNs and regular CNNs on image restoration task.

In summary, the primary contributions of this paper are three-fold:

- The first symmetry enhanced CNN for rain removal, which handles tilted rain streaks in images explicitly.
- Symmetry aggregation block for lossless aggregation of symmetry enhanced features.
- A nontrivial self-refining mechanism to tackle the accumulation effect of rain streaks.

## Related Work

### Symmetry Enhanced CNNs

In this paper, we define label symmetry to distinguish it from geometry symmetry which is more common in other research fields. Label symmetry is about the symmetry property of label under symmetry transformation (e.g. translation and rotation) on input. Correspondingly, the study of symmetry enhanced CNNs focuses on remolding CNNs to process data with label symmetry effectively.

A lot of effort has been made to extend the ability of neural networks to handle label symmetry in data. Tiled CNNs (Ngiam et al. 2010) change the weight sharing strategy to a tiled pattern for learning rotation symmetry of data. Deep Symmetry Networks (Gens and Domingos 2014) generalize CNN and compute features over arbitrary symmetry groups. Dieleman et al. (2015; 2016) show that, by rotation of feature maps, symmetry enhanced CNNs are able to learn rotationally equivariant representations and achieve success on tasks including galaxy morphology prediction, plankton classification and binary segmentation (pixel-level classification). Cohen and Welling (2016) propose a mathematical framework for the study of label symmetry based on group theory, and design a symmetry enhanced CNN called G-CNN. Instead of rotating feature maps, G-CNN rotates filters to achieve rotation equivariance. Rotationally Equivariant Vector Field Network (RotEqNet) is a CNN based model that achieves rotation equivariance by applying each filter at different orientations and extracting a vector field feature map (Marcos et al. 2017). Harmonic Networks (Worrall et al. 2017) constrain convolution filters to the family of circular harmonics to achieve the steerability as well as equivariance of CNNs. Besides the development of 2D symmetry enhanced CNNs, 3D symmetry enhanced CNNs are also under spotlight for their promising application in 3D object classification (Cohen et al. 2018). From another perspective, Dumont et al. (2018) observe that symmetry enhanced CNNs are much more robust than conventional CNNs under translational, rotational and geometric adversarial attacks.

However, compared with the theoretical innovation in symmetry enhanced convolutions, experimental study of symmetry enhanced CNNs is quite limited. So far, almost all the published works on symmetry enhanced CNNs are surrounding general classification tasks: image-level classification (Zhang et al. 2018; Dumont, Maggio, and Montalvo 2018; Marcos, Volpi, and Tuia 2016; Esteves et al. 2018; Li et al. 2018a; Gens and Domingos 2014; Fasel and Gatica-Perez 2006; Chidester, Do, and Ma 2018; Cohen and Welling 2016), pixel-level classification (Li et al. 2018b), or both (Weiler, Hamprecht, and Storath 2018; Worrall et al. 2017; Marcos et al. 2017; Dieleman, Fauw, and Kavukcuoglu 2016). As exceptions, Warped Convolution (Henriques and Vedaldi 2016) and RotEqNet (Marcos et al. 2017) are evaluated on orientation estimation of cars and faces, but these tasks lack enough comparable baseline methods. Unlike these works, the proposed method is designed specifically for the well-studied image restoration problem. To the best of our knowledge, our method is the first symmetry enhanced CNN that achieves the state-of-the-art performance on well-studied image processing task other than general classification.

### Rain Removal

As for traditional methods, ID (Kang, Lin, andFu 2012) decomposes rain images in the frequency domain to extract rain layer. DSC (Luo, Xu, and Ji 2015) adopts a sparse coding manner for rain removal. LP (Li et al. 2016) separates rain image into background layer and rain layer with priors based on Gaussian Mixture Model. Zhu et al. (2017) design three image priors to assist joint bi-layer optimization. In recent years, many deep learning based methods have been proposed. DetailNet (Fu et al. 2017) is an end-to-end deep learning framework that focuses on the high-frequency detail and regresses negative residual information. To extract tilted rain streaks in various orientations, JORDER (Yang et al. 2017) removes rain streaks with a deep recurrent dilated network. These deep learning based methods mostly enjoy success from the analysis of image components and adaptation of network structure. By contrast, our method obtains insight from the label symmetry of rain streak and tackles the rain removal problem by symmetry enhanced networks, which are theoretically designed to process visual objects with label symmetry more than translation symmetry.

### Symmetry Enhanced Convolution

In this section, we present a mathematical illustration of symmetry properties (equivariance and invariance) in CNNs, introduce a symmetry enhanced convolution which would be used in DSEN, and propose a generalized definition of it.
Symmetry in CNNs

For the sake of precision, we introduce symmetry group in our following discussion. Symmetry group (group) is a set of symmetry transformations (e.g. translation and rotation), that meets certain constraints. As a simple example, group $p4$ contains all compositions of translation and rotation by 90° about any grid point.

Equivariance in CNNs is the property that, under certain symmetry transformation on input, corresponding output of the network is transformed in a predictable manner. Notating operators as capital letters with hat ‘$\hat{\phantom{a}}$’, for every transformation $g$ of the given group $G$, if we can find a corresponding transformation $T'$ that satisfies:

$$f(\hat{T}_g x) = \hat{T}' f(x),$$  \hspace{1cm} (1)

we say $f$ is equivariant to group $G$. Here $\hat{T}_g$ is the operator of transformation $g$. A visual illustration is shown in Fig. 1.

Invariance in CNNs indicates that under certain symmetry transformation on input, the output of networks remains invariant. Invariance can be viewed as a special case of equivariance (Lenc and Vedaldi 2015), where $\hat{T}'$ in Eq. 1 is the identity operator.

To a great extent, the success of regular CNNs results from translation equivariance of convolution layer, which can be formulated as:

$$\hat{C}(\hat{T}_t f) = \hat{T}'(\hat{C} f),$$  \hspace{1cm} (2)

where $f : \mathbb{Z}^2 \rightarrow \mathcal{R}$ is a feature map of CNN, $\hat{T}_t$ denotes the translation operator and $\hat{C}$ indicates the convolution operator.

However, mathematical study (Cohen and Welling 2016) finds that regular convolution operation is not equivariant to rotation operation, which signifies that CNNs are not able to handle the rotated objects in images effectively. To help CNNs approximate label symmetry, data augmentation method (e.g. randomly rotating training images) is widely used, but it requires higher network capacity and fails on generalization (Dieleman, Fauw, and Kavukcuoglu 2016; Lenc and Vedaldi 2015). Therefore, the best solution is to explicitly encode symmetry in the network architecture.

Group Equivariant Convolution

Before introducing a rotationally equivariant convolution operation, we recall that a regular convolution kernel of layer $l$ in CNNs takes feature maps $f : \mathbb{Z}^2 \rightarrow \mathbb{R}^{K^{l-1}}$ with $K^{l-1}$ channels as input, and can be formulated as:

$$\hat{C} f(x) = \sum_{k=1}^{K^{l-1}} \sum_{y \in \mathbb{Z}^2} f_k(y)(\hat{T}_x \psi_k(y)), \hspace{1cm} (3)$$

$$\hat{T}_x \psi_k(y) = \psi_k(y - x). \hspace{1cm} (4)$$

Here, $f_k$ is the $k^{th}$ channel of feature maps, $\hat{C}$ denotes the convolution operator of a convolution kernel in layer $l$ and $\hat{T}_x$ is the translation operator to shift the filter $\psi_k(y)$ by $x$. All functions are defined on plane $\mathbb{Z}^2$.

To obtain rotation equivariance in CNNs, a family of symmetry enhanced convolution operations, called group equivariant convolution (G-convolution), is put forward to replace the correlation operation commonly used in CNNs (Cohen and Welling 2016). G-convolution is created to be equivariant to certain symmetry groups (e.g. $p4$), and its output is called group equivariant feature maps (G-feature maps). As $\mathbb{Z}^2$ can be viewed as group $p1$ which has only translation, we rewrite the G-convolution from group $H$ to group $I$ ($H \rightarrow I$) as:

$$\hat{G}_I f^H(i) = \sum_{k=1}^{K^{l-1}} \sum_{h \in H} f_k(h)(\hat{T}_i \psi_k(h)),$$  \hspace{1cm} (5)

where $\hat{G}_I$ denotes the G-convolution operator of group $I$, $\hat{T}_i$ is the operator of transformation $i$ from group $I$ ($i \in I$) and $h$ is a transformation of group $H$. Both filter $\psi_k$ and feature maps $f^H$ are functions defined on group $H$. When Eq. 5 satisfies $I = p4$, we call it a $p4$ group equivariant convolution ($p4$ G-convolution).

S-DSEN

In this part, we introduce our symmetry enhanced network and show how it can be used to extract and remove rain from a rain image after analyzing the task of rain removal.

Rain Model

Original rain image $O$ can be decomposed into the background layer $B$ and the rain layer $R$:

$$O = B + R.$$  \hspace{1cm} (6)

As long as we extract rain layer $R$ from the image, we are able to restore the rain-free background layer $B$ with Eq. 6 (Yang et al. 2017).

Deep Symmetry Enhanced Network

We illustrate the proposed DSEN in Fig. 2, which shows how rain layer is learned in a supervised way. To process images with arbitrary sizes, the proposed model is designed as a fully convolutional network with neither fully connected layer nor pooling operation. Specifically, DSEN consists of three function blocks: Symmetry enhanced convolutional block, symmetry aggregation block and decoder block.

Symmetry Enhanced Convolutional Block:

This component adopts $p4$ G-convolution as its basic operation to keep equivariant to rotation transformation while extracting high-level features through deep architecture. It composes of two kinds of G-convolution: $p4$ G-convolution on $\mathbb{Z}^2$ (P4Conv22) and $p4$ G-convolution on $p4$ (P4ConvP4). To process the
plane information of input images, P4ConvZ2 is compulsory as the first layer. For the rest layers, P4ConvP4 is stacked to increase the network capacity for abstracting high level features, and the depth can be adjusted depending on practical need. Residual connection (He et al. 2016) is adopted on each P4ConvP4 layer.

For implementation, since every transformation \( g \in p4 \) can be decomposed into a translation \( x \) and a rotation \( r \) from the corresponding point group \( C_4 \) (rotation by 90°, 180°, 270°), P4ConvZ2 and P4ConvP4 can be rewritten as:

\[
\hat{G}_{p4} f^{Z2}(x, r) = \sum_k \sum_{y \in \mathbb{Z}^2} f_k^{Z2}(y)(\hat{T}_x(\hat{T}_r \psi_k(y))),
\]

(7)

\[
\hat{G}_{p4} f^{P4}(x, r) = \sum_k \sum_{y \in \mathbb{Z}^2} \sum_{s \in C_4} f_k^{P4}(y, s)(\hat{T}_x(\hat{T}_r \psi_k(y, s))),
\]

(8)

where \( \hat{T}_x \) denotes the operator of translation \( x \) and \( \hat{T}_r \) denotes the operator of rotation \( r \). For P4ConvZ2 (in Eq. 7), its input \( f^{Z2} \) and filter \( \psi_k \) are both defined on plane \( \mathbb{Z}^2 \), while the output \( \hat{G}_{p4} f^{Z2} \) is a function on the group \( p4 \). For P4ConvP4 (in Eq. 8), its input, filter, and output are all defined on group \( p4 \).

In brief, to calculate \( p4 \) G-convolution, we need to rotate the convolution filters by 90°, 180°, 270° and 360°, then call the off-the-shelf convolution algorithm. Therefore, each regular channel of the \( p4 \) G-feature maps has four orientation channels corresponding to the four cyclic rotation transformations in \( C_4 \).

**Symmetry Aggregation Block:** The G-feature maps generated by symmetry enhanced convolutional block have 4 orientation channels for each regular channel. To aggregate rotationally equivariant features, orientation pooling method (Weiler, Hamprecht, and Storath 2018) pools over the orientation channels within each regular channel on binary segmentation task. However, since different orientation channels encode rain streaks from corresponding aspects, and rain streaks are not strictly perpendicular to each other, orientation pooling method causes information loss. To remedy this problem, we propose to apply a convolution operation on all orientation channels of G-feature maps. Additionally, as the channels of aggregated feature maps contribute differently to rain layer, an Squeeze-and-Excitation block for channel-wise recalibration. Each convolution or G-convolution layer is followed by an activation function, except that the decoder block is designed with a 3-channel 1 × 1 convolution layer for rain layer generation. Best viewed in color.

**Decoder block** This module is a regular convolution layer with 1 × 1 kernel size to decode and generate the three-channel (RGB) rain layer \( R \). Solving the inverse problem of Eq. 6, we obtain the restored background layer \( B \), which is supervised by ground truth background image with an Euclidean loss function.

**Self-refining Mechanism**

For image restoration problem, the output of system shares the same domain as its input. For example, output of the rain removal model is still an image of the same scene with fewer rain streaks. Thus the output of image restoration networks can be reused as input for finer processing. This multi-stage recurrent pipeline encourages the reuse of parameters to exploit the potential capacity of base models. For rain removal, this pipeline could be used to tackle the accumulation effect of rain streaks overlapping. However, simply repeating the network faces gradient vanishing problem at training. Therefore, we need to improve the gradient flow between deeper and shallower stages for the recurrent structure.

To this end, we design a novel self-refining mechanism with a stage-wise link called skip concatenation, which resembles a combination of dense connection (Huang...
et al. (2017) and skip connection (He et al. 2016). We illustrate our framework with a two-stage example of DSEN with self-refining mechanism (S-DSEN) in Fig. 3. The pipeline of stage I follows DSEN. In stage II, the G-feature maps of each P4ConvP4 layer at stage I is extracted and processed by an intermediate P4ConvP4 convolution layer generating the intermediate G-feature maps. The intermediate G-feature maps are concatenated with input of the same layer at stage II and then fed into the next layer. L2 loss functions of all stages are calculated and summed for supervised learning.

**Figure 3:** A two-stage example of S-DSEN. The output of stage I ($O'$) is reused as the input of stage II. The pipeline of stage I is the same as DSEN. In stage II, the G-feature maps of each P4ConvP4 layer at stage I is extracted and processed by an intermediate P4ConvP4 convolution layer generating the intermediate G-feature maps. The intermediate G-feature maps are concatenated with input of the same layer at stage II and then fed into the next layer. Different from dense connection, the links only exist between adjacent stages with stage-wise weight sharing, thus the increase of stage number would not add any model parameters. Formally, with stage-wise concatenation, the input $f^l_s$ of P4ConvP4 layer $l$ at stage $s$ is:

$$f^l_s = [\hat{G}^l_{\text{inter}}G^l_{s-1}, G^{l-1}_sf^{l-1}_s],$$  \hspace{1cm} (9)

where $\hat{G}^l_{\text{inter}}$ and $G^l_{\text{inter}}$ denote intermediate G-convolution operator and layer G-convolution operator of layer $l$ correspondingly. [...] refers to the concatenation of G-feature maps. In this way, the information flow between neighbor stages is improved. For optimization, L2 loss functions are calculated and summed at the output of every stage.

**Experiments and Analysis**

In this section, we perform an introspective study and measure against latest baselines on both synthetic and real-world datasets. Experiments are designed to answer the following questions:

**Q1 Overall Analysis** Does DSEN exceed its regular CNN counterparts?

**Q2 Component Analysis** How do different components contribute to S-DSEN?

**Q3 Peer Comparison** Does the proposed approach outperform the state-of-the-art methods?

**Benchmark**

We evaluate DSEN and S-DSEN on the rain removal problem.

**Datasets** Yang et al. (2017) synthesize three classes of rain images: Rain100L, Rain200L and Rain100H. The former two datasets have only one rain streak direction in images, while the hardest Rain100H is synthesized with five rain streak directions. Zhang et al. (2017) synthesize dataset with 800 image pairs containing rain streaks of various intensities and orientations, hence we call it Zhang800. To evaluate the ability of algorithms to extract tilted rain streaks, Rain100H and Zhang800, which are the most difficult cases, are chosen for objective evaluation. According to the official split, in Rain100H there are 1800 image pairs for training and 100 for testing; In Zhang800, there are 700 image pairs for training and 100 for testing. We randomly select 100 image pairs from training set for validation on both datasets. For subjective evaluation, we adopt both synthetic and real-world rain images collected from Zhang800, Rain100H as well as Internet.

**Evaluation Metrics** Two commonly used image quality measures are adopted: Structural Similarity Index (SSIM) (Wang et al. 2004) and Peak Signal to Noise Ratio (PSNR) (Huynh-Thu and Ghanbari 2008), for quantitative study. SSIM is widely used for measuring the similarity between two images with perceptual consideration, and PSNR focuses on the fidelity of restored images. Both indexes indicate better image restoration quality for higher value.

**Implementation Details** Since rain removal is a local image processing problem, to accelerate the training process, training images are randomly cropped to the size of $64 \times 64$. Deep learning models are trained on an Nvidia GTX 1080Ti GPU with Pytorch implementation, setting the batch size to 64 unless otherwise stated. LeakyReLU is used as activation function with negative slope set to 0.2. We adopt Adam algorithm (Kingma and Ba 2014) for optimization and set the learning rate to 0.0005 initially, which is divided by 10 at step 15000 and 17500. All trainable parameters are randomly initialized.

**Overall Analysis (Q1)**

In this part, we study the overall effectiveness of DSEN. Firstly we build a DSEN with 4 P4ConvP4 layers in symmetry enhanced convolutional block (7 convolution layers including G-convolution layers and regular convolution layers in total). Except for decoder block, the channel number of all filters are set to be 10 with the kernel size of $5 \times 5$. Then we design a CNN counterpart with exactly the same structure as DSEN but replace all the G-convolutions by regular convolutions and double the number of channels per filter to keep parameter size approximately invariant. Since data augmentation helps CNNs learn label symmetry, for comparison, we apply random rotation ($-30^\circ \sim 30^\circ$) and C4 data augmentation (random rotation by $90^\circ$, $180^\circ$, and $270^\circ$) on training images of regular CNN (CNN+DA and CNN+C4DA).
Figure 4: Restored samples of synthetic (O_syn) and real-world (O_real) images generated by: DetailNet (Fu et al. 2017), JORDER-R (Yang et al. 2017) and S-DSEN. PSNR and SSIM are calculated for synthetic cases. S-DSEN generates images with fewer rain streaks and more details evidently.

Table 1: PSNR and SSIM of regular CNN and DSEN on Rain100H and Zhang800. The best results are highlighted in bold.

| Dataset | Rain100H | Zhang800 |
|---------|----------|----------|
| Metric  | PSNR     | SSIM     | PSNR     | SSIM     | Parameter Size |
| CNN     | 23.36    | 0.7557   | 22.88    | 0.8127   | 52113         |
| CNN+DA  | 22.21    | 0.7055   | 24.04    | 0.8044   | 52113         |
| CNN+C4DA| 22.74    | 0.7320   | 22.49    | 0.7891   | 52113         |
| DSEN    | **23.60**| **0.7618**| 23.39    | **0.8162**| 50958         |

Table 2: PSNR and SSIM on Rain100H and Zhang800. The best results are highlighted in bold.

| Dataset     | Rain100H | Zhang800 |
|-------------|----------|----------|
| Metric      | PSNR     | SSIM     | PSNR     | SSIM     |
| DSEN_maxpool| 22.83    | 0.7359   | 22.30    | 0.7916   |
| DSEN_avgpool| 22.86    | 0.7382   | 22.30    | 0.7916   |
| DSEN_w/o_SE| 23.46    | 0.7595   | 22.53    | 0.8053   |
| DSEN       | 23.60    | 0.7618   | 23.39    | 0.8162   |
| MCNN        | 23.70    | 0.7226   | 22.90    | 0.7965   |
| MCNN+SR     | 26.36    | 0.8395   | 23.52    | 0.8362   |
| MDSEN       | 23.98    | 0.7295   | 23.48    | 0.8000   |
| MDSEN+SC    | 25.84    | 0.8182   | **23.70**| 0.8269   |
| S-DSEN      | **27.16**| **0.8589**| 23.64    | **0.8379**|

Component Analysis (Q2)

In this experiment, we study the effect of model components, especially symmetry aggregation block and self-refining mechanism.

Symmetry Aggregation Block We compare the proposed symmetry aggregation block with orientation pooling methods (Weiler, Hamprecht, and Storath 2018) including orientation max-pooling (DSEN_maxpool) and orientation average-pooling (DSEN_avgpool). For ablation study of the Squeeze-and-Excitation (SE) block, we remove it from DSEN (DSEN_w/o_SE). Experiment results in Table 2 indicate that orientation max-pooling and average-pooling share very close performance. Compared with orientation average pooling, DSEN improves PSNR by 3.24% and 4.89% respectively on Rain100H and Zhang800. DSEN outperforms its variants on all indicators.

Self-refining Mechanism (SR) To validate SR, which works on not only DSEN but also regular CNNs, we construct two multi-stage networks based on the CNN in Q1: with or without SR (MCNN+SR and MCNN). Both networks adopt 8-stage recurrent architecture. Experiment results in Table 2 show that with self-refining mechanism, the performance of regular CNN soars with SSIM and PSNR improved by 16.18% and 11.22% separately on Rain100H. Similarly, eight-stage S-DSEN raises SSIM and PSNR by 17.74% and 13.26% on Rain100H compared with its counterpart without SR (MDSEN). The performance of plain recurrent networks (MCNN and MDSEN) are even worse than their single-stage base models in Table 1 concerning SSIM. This serves as evidence for the importance of stage-wise information links in recurrent architecture and shows the validity of self-refining mecha-
Table 3: Comparison of S-DSEN with different stage number on Rain100H and Zhang800.

| Dataset   | Rain100H     | Zhang800    |
|-----------|--------------|-------------|
| Metric    | PSNR | SSIM   | PSNR | SSIM   |
| S-DSEN s2 | 25.04 | 0.8028 | 23.53 | 0.8270 |
| S-DSEN s4 | 26.24 | 0.8328 | 23.74 | 0.8340 |
| S-DSEN s6 | 26.90 | 0.8520 | 23.66 | 0.8364 |

Figure 5: Comparison of S-DSEN with different stage number. Experiments are conducted on Rain100H with the batch size of 32.

Comparison with the State-of-the-art Methods (Q3)

In this subsection, we compare our S-DSEN with other rain removal methods in Table 4. As our approach is the first symmetry enhanced CNN used in image restoration, we compare it with regular CNN based methods. S-DSEN outperforms the competing algorithms by a large margin.

Table 4: Comparison of quantitative results in terms of PSNR and SSIM. The best results are highlighted in bold.

| Dataset   | Rain100H     | Zhang800    |
|-----------|--------------|-------------|
| Metric    | PSNR | SSIM   | PSNR | SSIM   |
| ID (Kang, Lin, and Fu 2012) | 14.02 | 0.5239 | 18.88 | 0.5832 |
| DSC (Luo, Xu, and Ji 2015) | 15.66 | 0.4225 | 18.56 | 0.5996 |
| LP (Li et al. 2016) | 14.26 | 0.5444 | 20.46 | 0.7297 |
| DetailNet (Fu et al. 2017) | 22.26 | 0.6928 | 21.16 | 0.7420 |
| JORDER (Yang et al. 2017) | 22.15 | 0.6736 | 22.29 | 0.7922 |
| JORDER-R (Yang et al. 2017) | 23.45 | 0.7490 | 22.29 | 0.7922 |
| DSEN     | 23.60 | 0.7618 | 23.39 | 0.8162 |
| S-DSEN   | **27.16** | **0.8589** | 23.64 | 0.8379 |

As subjective evaluation is important for rain removal algorithms, a comparison of restored images generated by several methods from both synthetic and real-world rain images are shown in Fig. 4 and more samples are listed in supplementary material. The proposed method generates images with fewer rain streaks and more details.

Conclusions

This paper investigates the label rotation symmetry of rain streaks and proposes a novel symmetry enhanced network to remove them from images. Moreover, a handy self-refining mechanism, which works on both regular CNNs and symmetry enhanced CNNs, is designed to tackle the accumulation effect of rain streaks with economical model parameter size. S-DSEN outperforms its regular CNN counterparts with the state-of-the-art results on both synthetic and real-world datasets. Since this work is inspired by the theoretic development of symmetry enhanced CNNs, the success of S-DSEN suggests the predictive power of symmetry theory developed in CNNs, leading to a deeper understanding of steerable deep learning.

References

Chidester, B.; Do, M. N.; and Ma, J. 2018. Rotation equivariance and invariance in convolutional neural networks. arXiv preprint arXiv:1805.12301.

Cohen, T. S., and Welling, M. 2016. Group equivariant convolutional networks. In ICML, 2990–2999.

Cohen, T. S.; Geiger, M.; Khler, J.; and Welling, M. 2018. Spherical CNNs. In ICLR.

Dieleman, S.; Fauw, J. D.; and Kavukcuoglu, K. 2016. Exploiting cyclic symmetry in convolutional neural networks. In ICML, 1889–1898.

Dieleman, S.; Willott, K. W.; and Dambre, J. 2015. Rotation-invariant convolutional neural networks for galaxy morphology prediction. Monthly Notices of the Royal Astronomical Society 450(2):1441–1459.

Dumont, B.; Maggio, S.; and Montalvo, P. 2018. Robustness of rotation-equivariant networks to adversarial perturbations. arXiv preprint arXiv:1802.06627.

Esteves, C.; Allen-Blanchette, C.; Zhou, X.; and Daniilidis, K. 2018. Polar transformer networks. In ICLR.
Fasel, B., and Gatica-Perez, D. 2006. Rotation-invariant neoperceptron. In ICPR, 336–339.
Fu, X.; Huang, J.; Zeng, D.; Huang, Y.; Ding, X.; and Paisley, J. 2017. Removing rain from single images via a deep detail network. In CVPR, 1715–1723.
Gens, R., and Domingos, P. 2014. Deep symmetry networks. In NIPS, 2537–2545.
Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. Deep Learning. MIT Press. http://www.deeplearningbook.org.
He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Identity mappings in deep residual networks. arXiv preprint arXiv:1603.05027.
Henriques, J. F., and Vedaldi, A. 2016. Warped convolutions: Efficient invariance to spatial transformations. arXiv preprint arXiv:1609.04382.
Hu, J.; Shen, L.; and Sun, G. 2017. Squeeze-and-excitation networks. arXiv preprint arXiv:1709.01507.
Huang, G.; Liu, Z.; Maaten, L. v. d.; and Weinberger, K. 2017. Densely connected convolutional networks. In CVPR, 4700–4709.
Huynh-Thu, Q., and Ghanbari, M. 2008. Scope of validity of PSNR in image/video quality assessment. Electronics Letters 44(13):800–801.
Kang, L.-W.; Lin, C.-W.; and Fu, Y.-H. 2012. Automatic single-image-based rain streaks removal via image decomposition. IEEE TIP 21(4):1742–1755.
Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
Lenc, K., and Vedaldi, A. 2015. Understanding image representations by measuring their equivariance and equivalence. In CVPR, 991–999.
Li, Y.; Tan, R. T.; Guo, X.; Lu, J.; and Brown, M. S. 2016. Rain streak removal using layer priors. In CVPR, 2736–2744.
Li, J.; Yang, Z.; Liu, H.; and Cai, D. 2018a. Deep rotation equivariant network. Neurocomputing 290:26 – 33.
Li, X.; Yu, L.; Fu, C. W.; and Heng, P. A. 2018b. Deeply supervised rotation equivariant network for lesion segmentation in dermoscopy images. arXiv preprint arXiv:1807.02804.
Luo, Y.; Xu, Y.; and Ji, H. 2015. Removing rain from a single image via discriminative sparse coding. In ICCV, 3397–3405.
Marcos, D.; Volpi, M.; Komodakis, N.; and Tuia, D. 2017. Rotation equivariant vector field networks. In ICCV, 5058–5067.
Marcos, D.; Volpi, M.; and Tuia, D. 2016. Learning rotation invariant convolutional filters for texture classification. In ICPR, 2012–2017.
Ngiam, J.; Chen, Z.; Chua, D.; Koh, P. W.; Le, Q. V.; and Ng, A. Y. 2010. Tiled convolutional neural networks. In NIPS, 1279–1287.
Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: From error visibility to structural similarity. IEEE TIP 13(4):600–612.
Weiler, M.; Hamprecht, F. A.; and Storath, M. 2018. Learning steerable filters for rotation equivariant cnns. In ICCV, 849–858.
Worrall, D. E.; Garbin, S. J.; Turmukhambetov, D.; and Brostow, G. J. 2017. Harmonic networks: Deep translation and rotation equivariance. In CVPR, 5028–5037.
Yang, W.; Tan, R. T.; Feng, J.; Liu, J.; Guo, Z.; and Yan, S. 2017. Deep joint rain detection and removal from a single image. In CVPR, 1357–1366.
Zhang, H., and Patel, V. M. 2018. Density-aware single image de-raining using a multi-stream dense network. In CVPR.
Zhang, X.; Xie, Y.; Chen, J.; Wu, L.; Ye, Q.; and Liu, L. 2018. Rotation invariant local binary convolution neural networks. IEEE Access 6:18420 – 18430.
Zhang, H.; Sindagi, V.; and Patel, V. M. 2017. Image de-raining using a conditional generative adversarial network. arXiv preprint arXiv:1701.05957.
Zhu, L.; Fu, C.-W.; Lischinski, D.; and Heng, P.-A. 2017. Joint bi-layer optimization for single-image rain streak removal. In CVPR, 2526–2534.