



Pre-rotation Only at Inference-Stage: A Way to Rotation Invariance of Convolutional Neural Networks

Yue Fan^{1,2} · Peng Zhang³ · Jingqi Han³ · Dandan Liu⁴ · Jinsong Tang² · Guoping Zhang¹

Received: 7 January 2024 / Accepted: 26 March 2024
© The Author(s) 2024

Abstract

The popular convolutional neural networks (CNN) require data augmentation to achieve rotation invariance. We propose an alternative mechanism, Pre-Rotation Only at Inference stage (PROAI), to make CNN rotation invariant. The overall idea is to learn how the human brain observe images. At the training stage, PROAI trains a CNN with a small number using images only at one orientation. At the inference stage, PROAI introduces a pre-rotation operation to rotate each test image into its all-possible orientations and calculate classification scores using the trained CNN with a small number of parameters. The maximum of these classification scores is able to simultaneously estimate both the category and the orientation of each test image. The specific benefits of PROAI have been experimented on rotated image recognition tasks. The results shows that PROAI improves both the classification and orientation estimation performance while greatly reduced the numbers of parameters and the training time. Codes and datasets are publicly available at <https://github.com/automlresearch/FRPRF>.

Keywords Rotated image recognition · Orientation estimation · Convolutional neural networks · Rotation invariance

1 Introduction

One of the goals of machine learning research is to obtain better generalization ability using as fewer parameters as possible. Convolutional neural networks (CNN) [1, 2] are closer to this goal than fully connected networks because they share weights of convolutional filters across different image locations. As shown in Fig. 1a, such a weight-sharing mechanism provides CNNs with space shift invariance, and reduces the number of parameters of CNNs [3]. However, such a weight-sharing mechanism does not exist in the rotation dimension and CNNs still lacks rotation invariance [4]. As shown in Fig. 1b, when the input image rotates by a certain angle, the original weights of the convolutional kernels quickly and seriously mismatch with the regions to be convoluted, which usually leads to the failure of feature

extraction and classification in rotated image recognition (RIR) tasks.

To recognize arbitrarily rotated images, existing RIR researchers commonly train CNN using rotation data augmentation [5–11]. There are three implementation approaches for rotation data augmentation. The most straightforward yet widely used implementation approach of rotation data augmentation [12] is randomly rotating each training image, so that the CNN output as identical classification scores as possible for an image and its rotated versions. Such a straightforward approach does not need to adjust the architectures of CNNs, but its performance is heavily dependent on the diversity of the orientations among training images. This is to say, CNNs must learn as many orientations as possible to achieve high RIR performance. However, providing CNNs with training images at all-possible orientations is hard work. To improve RIR performance, the second implementation approach of rotation data augmentation is to build multiple rotation channels, which actively rotated the features extracted by CNN [4, 13, 14]. For example, Laptev et al. [8] uniformly rotated the input image by 24 angles in $[-180^\circ, 180^\circ]$, then applied 'maximum pooling' to features extracted in these 24 images. These 24 images are called image rotation channels, which can produce more robust rotation-invariant features than a

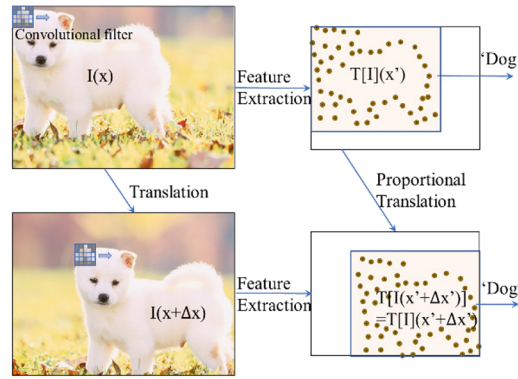
✉ Guoping Zhang
Guoping_Zhang999@163.com

¹ Central China Normal University, Wuhan, China

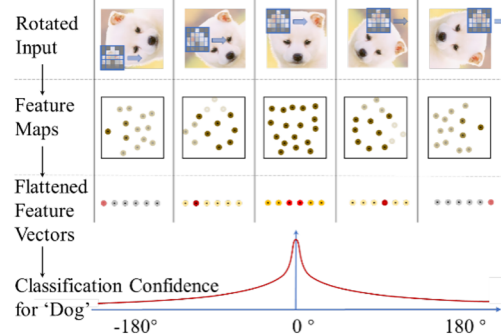
² Naval University of Engineering, Wuhan, China

³ National University of Defense Technology, Changsha, China

⁴ Yancheng Institute of Technology, Yancheng, China



(a) A translation of Δx of an object on the input image $I(x)$ causes a proportional translation of $\Delta x'$ on the feature map $T[I](x')$, i.e., $T[I(x+\Delta x)] = T[I(x'+\Delta x)]$, where $T(\cdot)$ denotes convolution operation. Such invariance of feature extraction to space shift allows CNNs to recognize objects at other locations.



(b) Rotating the input image changes the angle between the convolution kernel and the regions to be convolved, which causes feature extraction to fail and leads to incorrect classifications in the fully connected layer.

Fig. 1 Differences between the invariances of feature extraction of CNN for translated and rotated images

single image channel. But because building multiple image channels are usually computationally unfriendly, multiple rotation channels have also been built by rotating feature maps [14–18] or convolution kernels [4, 6, 7, 19]. The two approaches above take advantage of CNN's powerful function-fitting ability to ‘memorize’ images at all orientations so that the trained CNN can directly recognize images at any orientations. Different from these two approaches, other effective RIR methods are based on ‘derotation’ [20]. For example, Spatial Transformer Network (STN) [21], has been proposed to reduce the Number of training images. STN uses a spatial transformer layer to align rotated images to several canonical orientations, so the CNN only needs to learn several orientations. However, it should be noted that the training of the spatial transformer layer still relies on images at different orientations. This is to say STN still uses rotation data augmentation.

Although the rotation data augmentation has been widely used in RIR tasks, it has at least two disadvantages. First, CNN has to learn images at as many orientations as possible, which significantly increases the Number of training samples. More training samples require more parameters of CNN to achieve good generalization, and also increase the training time of CNN. Second, the outputs of CNN trained

with rotation data augmentation are independent or insensitive to the rotation of input images. This means that rotation data augmentation makes CNN lose orientation information. As a result, if the orientation need to be predicted, then CNN has to build extra orientation regression tasks [22, 23]. The abovementioned two disadvantages don't conform to how the human brain works because the human brain doesn't need to memorize images at all orientations. Psychological studies [24–26] have suggested that there is a "mental rotation" process when we recognize objects in less similar orientations in the human brain. “Mental rotation” refers to that an arbitrarily orientated mental imagery is rotated to multiple orientations and recognized multiple times until the mental imagery attains its normal orientation. Benefited from “mental rotation”, human brain is able to recognize arbitrarily oriented images by learning and memorize images at one orientation. This inspires us to develop a similar RIR mechanism for CNN to improve the RIR performance while decreasing the Number of training samples, which is named Pre-Rotation Only at Inference stage (PROAI).

From the humans “mental rotation” process we can see that rotation invariance can be achieved by share the recognition ability across different rotations. This is also the core recognition principle of PROAI to achieve rotation

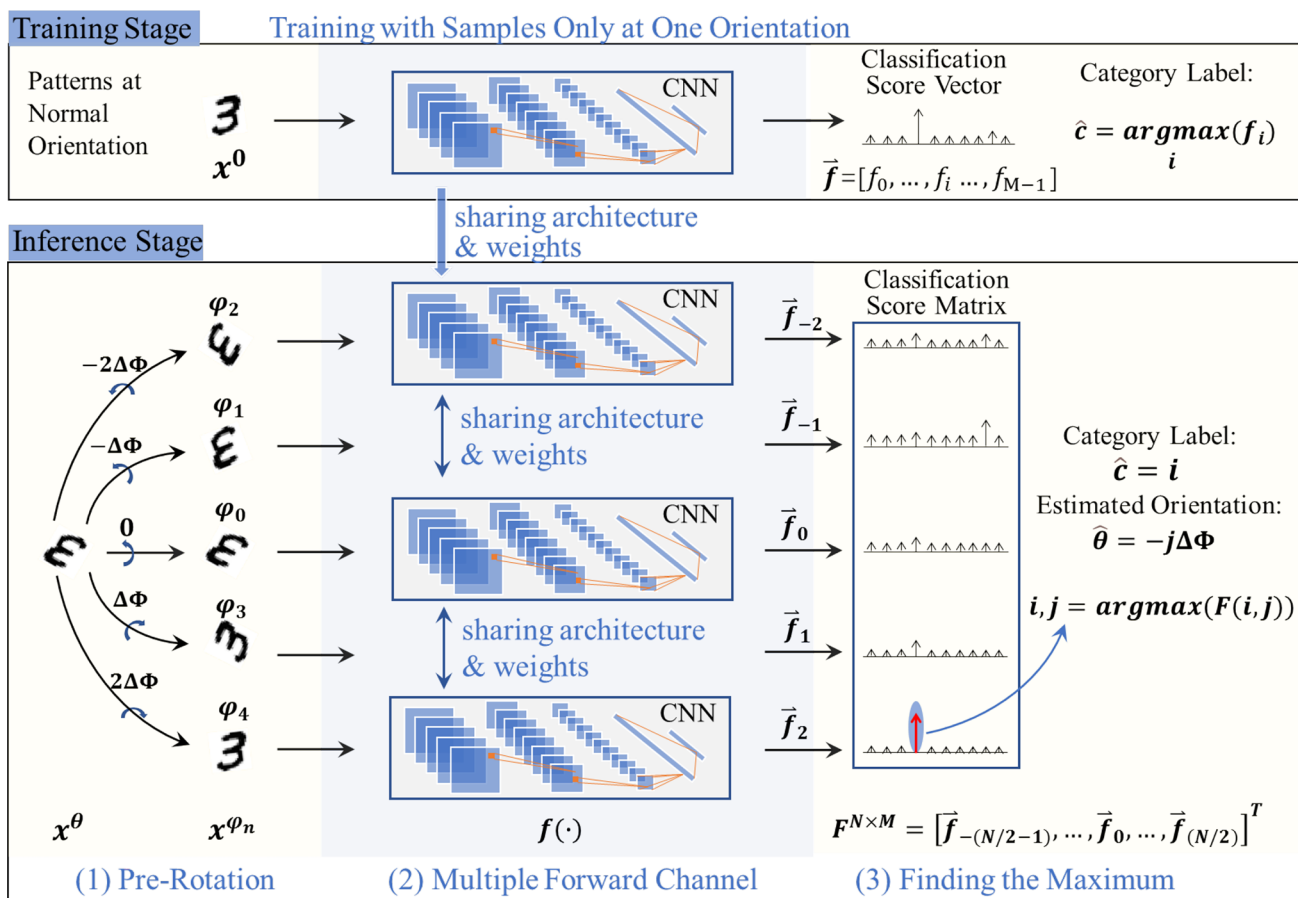


Fig. 2 Illustration of using PROAI to recognize rotated images

invariance, i.e., to share CNN weights across the rotation dimension of images. At the training stage, PROAI trains CNN with images only at one orientation, so the CNN can achieve high generalization using only a small number of parameters. Also, this CNN is supposed to correctly classify images only at the orientation. At the inference stage, PROAI generalize the recognition ability of the CNN to any other orientations through a pre-rotation operation. The pre-rotation operation rotates each test image into its all-possible orientations to generate multiple rotated versions, which are then fed into the CNN with a small number of parameters to calculate classification scores. The maximum of these classification scores is applied to simultaneously estimate both the category and the orientation of each test image.

Compared with existing RIR methods, PROAI has made the following two contributions: (1) Architectures and weights of the entire CNN are shared across the rotation dimension for the first time, which allows CNN no longer need to learn rotated images at arbitrary orientations in RIR tasks, reducing both the Number of free parameters of CNN and training time. (2) PROAI builds an orientation-related learning task for

CNN, enabling CNN to estimate images' orientations without adding extra orientation regression tasks.

2 Pre-rotation Only at Inference-Stage (PROAI)

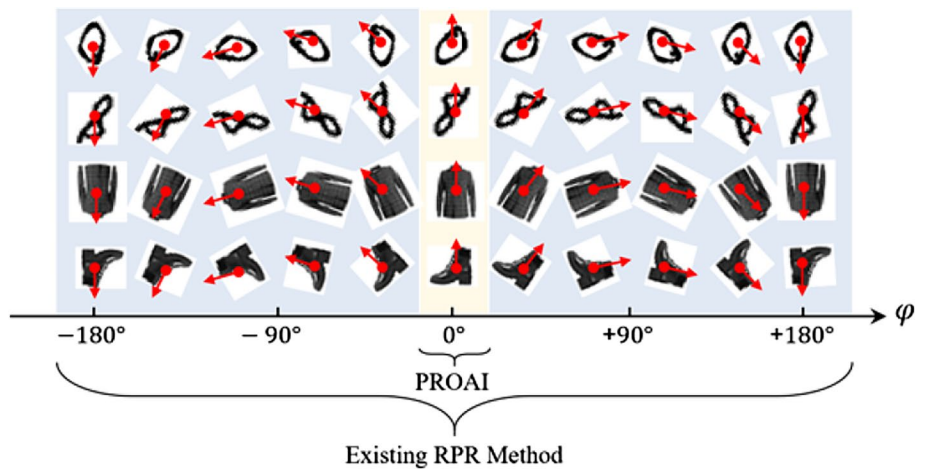
The workflow of PROAI illustrated in Fig. 2 is divided into two stages, i.e., the training stage and the inference stage. At the training stage, a CNN, which have a small number of parameters, is trained by images only at one orientation. As a result, this CNN is able to recognize images only at one orientation. At the inference stage, a multi-channel weight-sharing mechanism generalizes this recognition ability to images at any other orientations, so images at any other orientations can also be recognized.

2.1 Training Procedure of PROAI

2.1.1 Training Procedure

Figure 3 shows rotated images from MNIST [27] and Fashion MNIST [28] datasets. As it is shown, the angle between

Fig. 3 Illustration of rotation orientations of images (red arrows are assumed to be the top of objects)



the top of the image object and the positive direction of the y-axis is defined to be the rotation orientation of the image. In this paper, the orientations of images are denoted as φ ($\varphi \in [-180^\circ, +180^\circ]$), and the arbitrarily rotated image of the object with category i can be expressed as x_i^φ ($i \in Z$). Specifically, $\varphi = 0$ (the positive direction of the y-axis) refers to the normal orientation, and x_i^0 refers to the images with category i at normal orientation.

Based on the abovementioned definition of normal orientation, the training loss of PROAI is defined in Eq. (1). We design the following modified cross-entropy loss function for PROAI. Owing to the fact that PROAI only uses x_i^0 to train CNN, the cross entropy loss function $L[f(\cdot), i]$ is calculated through Eq. (1)

$$L[f(x_i^0), i] = - \sum_{c=1}^C y_c \cdot \log[f_c(x_i^0)] \tag{1}$$

where $f(\cdot)$ represents the output of a CNN, i is the category labels of an image x_i^0 . C is the Number of sample categories in the training dataset, and $c \in [1, C]$. y is a one-hot vector for the category i , y_c is the c th element of y , and $f_c(x_i^0)$ is the c th element of $f(\cdot)$.

One CNN trained with images only at one orientation is supposed to output higher classification scores for images at the orientation than images at other orientations. Therefore, PROAI makes CNN output the maximum classification scores only at normal orientation. In other words, PROAI makes CNN output peak value at the normal orientation. Such property is formulated by:

$$\begin{cases} \max[f(x_j^\varphi)] \leq \max[f(r^\varphi(x_i^0))], & ' = ' \text{ holds when } j = i \\ \max[f(x_i^\varphi)] \leq \max[f(x_i^0)], & ' = ' \text{ holds when } \varphi = 0 \end{cases} \tag{2}$$

where i, j are category labels of image objects, $r^\varphi(\cdot)$ represents rotating an image by an angle of φ , $x_i^\varphi = r^\varphi(x_i^0)$.

2.1.2 Comparison with the Training Procedures of Existing RIR Methods

Figure 4 compares of PROAI with existing RIR methods from three aspects: training images, network architectures, and annotation.

In terms of training images, PROAI trains CNNs using images only at normal orientation, while existing methods have tried to make CNN 'memorize' images at as many orientations as possible (see Fig. 4 or Fig. 3). Take the implementation approach of Rotation Data Augmentation (RDA) integrated from Pytorch¹ as an example, each training image will be rotated to a random orientation in each epoch of training. That is to say, each training image will be transformed into a new version in each training epoch. Noting the Number of total training epochs as N_{epoch} , then data augmentation requires N_{epoch} times more training images than PROAI.

In terms of network architecture, PROAI requires CNN with smaller parameters than existing RIR methods. This is because PROAI is required to learn images only at one orientation, and the variation of the training image dataset of PROAI is apparently smaller than that of RDA methods [29]. To achieve better RDA performance, existing RIR methods must improve CNNs' parameters using more complex network architectures. For example, Transformation-Invariant Pooling (TIPooling) obtains the final rotation invariant features by pooling the features extracted from multiple image rotation channels [8]; Oriented Response Networks (ORN) and RotEqNet [6] create multiple rotation channels by rotating convolutional filters to extract rotation invariant features [7]; STN trains a complex spatial transformation layer to

¹ <https://github.com/pytorch/vision/blob/main/torchvision/transforms/transforms.py>, RandomRotation Module.



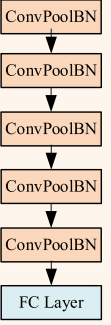
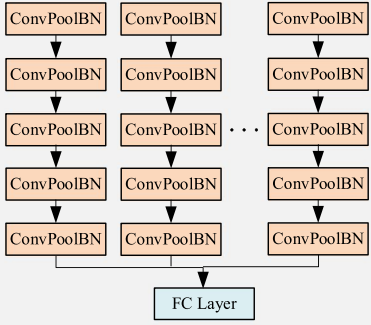
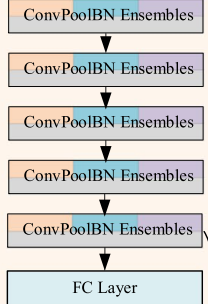
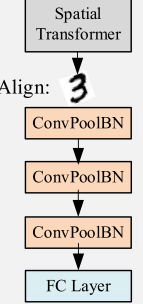
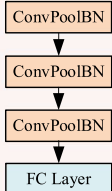
Methods	RDA	TIPooling	ORN	STN	PROAI
Training Samples	Randomly Oriented Instances: 				
Network Architecture During Training Stage					
Label	Category (Angle label is also required in orientation estimation task)				Category

Fig. 4 Comparison of the training stages of PROAI and existing RIR methods

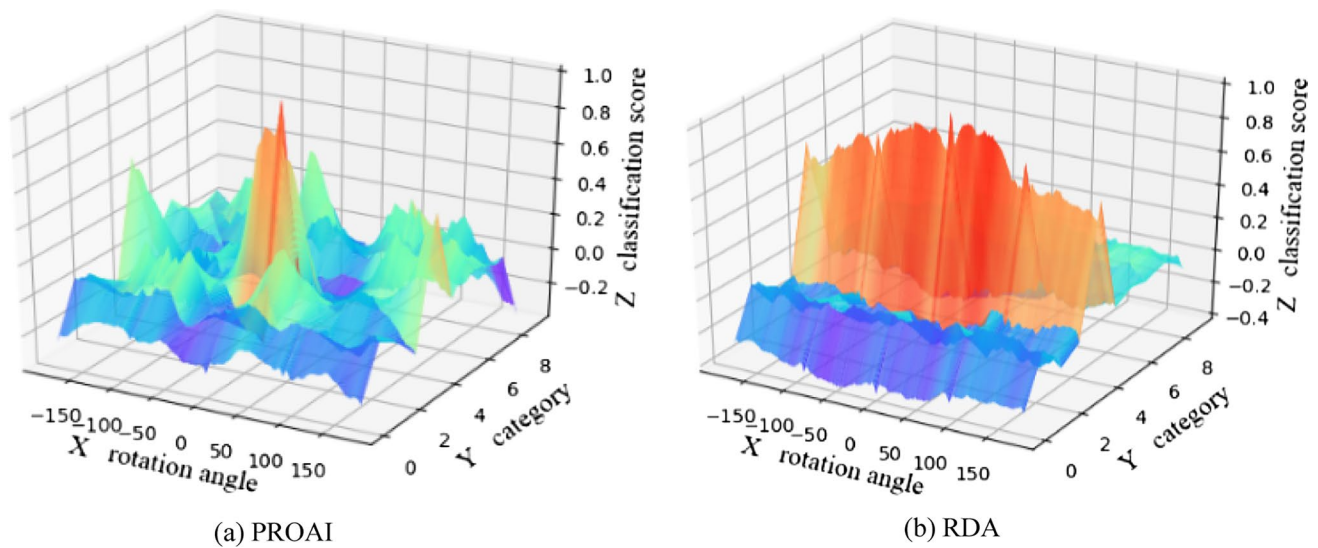


Fig. 5 The output of CNNs trained by PROAI and RDA to images at different orientations

align images to a similar orientation before using CNN to recognize images [21].

In terms of annotation, PROAI only have to annotate category labels for training images, no matter in classification or orientation estimation tasks. As a contrast, existing RIR methods have to annotate both category or angle labels for classification or orientation estimation tasks.

The differences in the training procedures shown in Fig. 4 result in different CNN classification scores for rotated images. Take rotated handwritten digit recognition task as an example, the output classification scores of CNN to a

digit “4” in different orientations are shown in Fig. 5. In Fig. 5a, the CNN trained with images only at normal orientation outputs higher classification scores on the correct category label ‘4’ for images that are oriented close to the normal orientation and outputs lower classification scores on wrong category labels or other orientations. The output in Fig. 5a satisfies Eq. (1). For comparison, as shown in Fig. 5b, rotation data augmentation makes the CNN insensitive to changes of orientation, resulting in similar responses for different orientations on the correct category label ‘4’, i.e., $f(x_i^0) \approx f(x_i^p)$.

Methods	RDA	TIPooling	ORN	STN	PROAI
Test Samples	Test Samples with Random Orientations				
Preprocess	\mathfrak{m}	Prerotation: $\mathfrak{m} \ \mathfrak{3} \ \mathfrak{w} \ \mathfrak{e}$	\mathfrak{m}	\mathfrak{m}	Prerotation: $\mathfrak{m} \ \mathfrak{m} \ \mathfrak{3} \ \mathfrak{3} \ \mathfrak{w} \ \mathfrak{w} \ \mathfrak{e} \ \mathfrak{e}$
Network Architecture During Inference Stage					
Output	Category (Angle estimation is possible only if orientation is learned during training stage)				Category & Angle

Fig. 6 Comparison of the inference stages of PROAI and existing RIR methods

2.2 Inference Procedure of PROAI

2.2.1 Inference Procedure

The inference procedure of PROAI illustrated in Fig. 2 is composed of three steps.

(1) Pre-rotate a test image into multiple orientations.

Given a test image at an orientation of θ , $x^\theta \in \mathbb{R}^{H \times W \times d}$ (H, W, d are the height, width, and number of color channels of the test image). It is firstly rotated by N angles that are uniformly distributed in $[-\Phi, +\Phi]$. The angle interval for rotation is $\Delta\Phi$, $\Delta\Phi=2\Phi/N$. The rotated images of the test image are denoted as,

$$x^{\varphi_n} = r^{\pm\Delta\varphi_n}(x^\theta) \tag{3}$$

$$\Delta\varphi_n = \pm n \cdot \Delta\Phi, \quad \varphi_n = \theta \pm n \cdot \Delta\Phi \tag{4}$$

where $\forall n = -\left(\frac{N}{2} - 1\right), -\left(\frac{N}{2} - 2\right), \dots, 0, \dots, \frac{N}{2} - 1, \frac{N}{2}$, $0^\circ \leq \Phi \leq 180^\circ$.

(2) Calculate the classification score matrix of multiple forward channels.

Applying a cluster of CNN sharing the same architecture and weights with the CNN trained with images only at normal orientation to independently conduct forward calculation in each channel, the classification score vector \vec{f}_n of the n th channel can be obtained as,

$$\vec{f}_n = f(x^{\varphi_n}) = [f_0, \dots, f_m, \dots, f_{M-1}] \tag{5}$$

where, f_m is the classification confidence that belongs to the m th class image, $m \in [0, \dots, M - 1]$, where M denotes the number of all possible categories. For the M -class classification problem, f_n is a vector with length of M . All these classification score vectors are arranged to form a classification

score matrix $F = [\vec{f}_{-(N/2-1)}^T, \dots, \vec{f}_0^T, \dots, \vec{f}_{N/2}^T]$, where $F \in \mathbb{R}^{N \times M}$.

(3) Estimating category and orientation by finding the maximum classification score matrix.

According to Eq. (1), the CNN trained with images only at normal orientation outputs higher classification scores for images oriented close to the normal orientation and outputs lower classification scores for images oriented far away from the normal orientation (see Fig. 5a). Therefore, both the category \hat{c} and the orientation $\hat{\theta}$ of the test image can be estimated from the coordinate of the maximum F , i.e.,

$$\hat{c} = k, \quad \hat{\theta} = -j \cdot \Delta\Phi \tag{7}$$

where,

$$(j, k) = \arg \max_{(n,m)} \{\text{soft max}[F^{N \times M}(n, m)]\} = \arg \max_{(n,m)} [F^{N \times M}(n, m)] \tag{8}$$

2.2.2 Comparison with the Inference Procedures of Existing RIR Methods

Figure 6 compares PROAI with existing RIR methods from three aspects, i.e., preprocessing of test images, network architectures, and outputs.

In terms of the preprocessing of test images, PROAI pre-rotates each test image into multiple orientations to build image rotation channels, while RDA, ORN, and STN directly classify rotated images using a single channel.

In terms of the network architectures, PROAI builds a cluster of copies of the CNN trained with normal images, while existing methods use the same CNN architecture with training stages. It should be noted that although TIPooling,

and ORN also use multiple forward channels in the inference stage, PROAI is essentially different from them. This is because the weights of CNNs in existing methods are learned from all-possible orientations, while the weights of the CNN in PROAI are trained with normal images first and then shared across the rotation channels.

In terms of the outputs, PROAI simultaneously outputs the category and orientation for each test image, while the existing methods only predict what they have learned at the training stage.

3 Results and Discussions

MNIST and Fashion MNIST have been commonly used to evaluate RIR performances [6, 8, 21]; hence they are applied to evaluate the performance of PROAI. The performance evaluation of PROAI is divided into training and inference stages. At the training stage, the parameters and training time of PROAI are compared with existing RIR methods. At the inference stage, the classification accuracy of PROAI is firstly compared with existing RIR methods. Then the orientation estimation precision of PROAI is compared with a CNN angle regressor. The CNN angle regressor is tailor-designed and trained in a supervised orientation regression task because existing RIR methods have not reported orientation estimation precision yet. Comparing the classification accuracies achieved by PROAI with existing RIR methods, which can demonstrate that PROAI can achieve higher classification accuracies in RIR tasks. And PROAI can achieve higher orientation estimation performance in RIR tasks.

In addition to quantitatively evaluating PROAI on MNIST and Fashion MNIST, PROAI has also been applied to the rotated face recognition task of FDDB dataset [30], and the underwater rotated target recognition task of SCTD dataset [31].

3.1 Preparation of Dataset and Design of CNN Architecture

3.1.1 Preparation of RIR Dataset

Two RIR datasets, Rotated MNIST and Rotated Fashion MNIST, are prepared for RIR experiments in this paper.

Firstly, Rotated MNIST is generated by randomly rotating the images in MNIST within a certain angle range. Different angle ranges have been used by existing researches, which typically include $[0^\circ, 0^\circ]$, $[-90^\circ, 90^\circ]$, $[-180^\circ, 180^\circ]$. Correspondingly, Rotated datasets generated by these three angle ranges are denoted as rot0, rot180, and rot360. If assuming that the images in the original MNIST are normal oriented, then rot0 can be taken as the training dataset for PROAI. As a result, PROAI only require rot0 to train CNNs

at the training stage, while existing RIR methods have to use rot180, rot360 to train CNNs. At the inference stage, rot180, and rot360 of the test dataset were applied to compare the classification accuracies of PROAI with existing RIR methods.

In addition to the classification task, PROAI also conducted orientation estimation task on Rotated MNIST. Therefore, the rotation angles of images must be labeled to form orientation labels. PROAI does not require orientation labels of training images, but existing RIR research must use them for training supervised orientation regressors. At the inference stage, orientation labels of validation or test images are applied to evaluate the orientation estimation precisions.

Secondly, Rotated Fashion MNIST is generated using the abovementioned method. The Rotated Fashion MNIST is obtained by applying rotational transformation on Fashion MNIST, and is much more difficult to recognize than Rotated MNIST. For example, the data augmentation method can achieve 95% test accuracy on Rotated MNIST, but only 77% test accuracy on the Rotated Fashion MNIST. In addition, because the image orientation of Fashion MNIST is very close to the ideal normal orientation (see Fig. 3), Rotated Fashion MNIST can effectively examine whether PROAI is able to extend the recognition ability of CNN for images in one orientation to other arbitrary orientations.

Both in Rotated MNIST and Rotated Fashion MNIST, the numbers of images in training, validation, test sets are 50,000, 10,000, 10,000, and the image sizes are 28×28 .

3.1.2 Design of CNN Architectures

To fairly compare PROAI with existing RIR methods, RDA was applied to compare with PROAI. The reason for choosing RDA for comparison is two-folded: (1) RDA is the most practical RIR method. (2) Neither RDA nor PROAI needs to change the architecture of CNNs. The CNN architecture designed for PROAI and RDA is shown in Table 1. The designed CNN has a similar architecture with the CNN architecture used in the experiments of RIR classification [8]. This CNN architecture is composed of classic convolution and maxpooling layer to extract feature. The extracted features are transformed by the 'ReLU' activation function. To improve the generalization of the architecture, Batch Normalization (BN) [32] layers has been added after each 'ReLU' activation layer. At the output of the CNN, the 'Softmax' activation function is applied to transform the output. The output of the CNN is a classification score vector with length C (C is the Number of sample categories in the training dataset).

Since PROAI and RDA require CNNs with different numbers of parameters to achieve each best generalization performance. The numbers of parameters of the CNN in Table 1

Table 1 The CNN architecture designed for PROAI and RDA

Layer	Parameters and channel size
Input	Size: 32×32
Convolution	Kernel: 3×3 , channel: $N1$
ReLU (+BN layer)	
Max pooling	Kernel: 2×2 , stride: 2
Feature map1	Size: $16 \times 16 \times N1$
Convolution	Kernel: 3×3 , channel: $M1 \times N1$
ReLU (+BN layer)	
Max pooling	Kernel: 2×2 , stride: 2
Feature map2	Size: $8 \times 8 \times (M1 \times N1)$
Convolution	Kernel: 3×3 , channel: $M2 \times N1$
ReLU (+BN layer)	
Max pooling	Kernel: 2×2 , stride: 2
Feature map3	Size: $4 \times 4 \times (M2 \times N1)$
Linear	$1 \times 1 \times N2$
Dropout	
Linear, softmax	$1 \times 1 \times C$

were adjusted by changing four variable architecture hyper-parameters, which includes the Number of channels for the Initial Convolutional layer ($N1$), the number of neurons for the Fully Connected layer ($N2$), and two multipliers for the channel numbers ($M1$ and $M2$). $N1$, $N2$, $M1$, $M2$ were set to adapt the requirements of different RIR methods and datasets. Specifically, PROAI requires smaller number of parameters than RDA, and CNNs designed for Rotated Fashion MNIST requires larger number of parameters than Rotated MNIST (Because images in Fashion MNIST are more complex than that in MNIST, more convolution filters are used than that for MNIST). Architecture hyper-parameters, numbers of parameters, and numbers of computations of CNNs, which are designed in the following experiments, are listed in Table 2. The designed architecture hyper-parameters in Table 2 can help PROAI or RDA achieve each high generalization performance. The Numbers of parameters and computations in Table 2 were calculated by python 'thop' library.²

The reason for choosing such a traditional and simple architecture shown in Table 1 is to demonstrate that PROAI is not picky for CNN architectures. In addition to this architecture, one of the most popular CNN architectures ResNet [33], and the CNN architectures automatically designed by a state-of-the-art Neural Architecture Search (NAS) algorithm [34] are also applied to evaluate the RIR performance of PROAI.

3.1.3 Evaluation Metrics

Two important evaluation metrics of RIR tasks are classification accuracy and orientation estimation precision. These two metrics are calculated through the following Eqs. (9) and (10). The classification accuracy A represents the proportion of the Number of correct classifications to the Number of all targets, i.e.,

$$A = \frac{1}{N_I} \sum_{n=1}^N I(y_n = \hat{y}_n) \quad (9)$$

where N_I is the Number of images in a dataset, $I(\cdot)$ refers to the indicator function. The A calculated on the training, validation, and test datasets are also referred to as the training accuracy, validation accuracy, and test accuracy. The higher the A is, the more accurate the classification is.

The orientation estimation precision is evaluated using the Mean Absolute Error (MAE), which is calculated by,

$$MAE = \frac{1}{N_I} \sum_{i=1}^{N_I} |\varphi_i - \hat{\varphi}_i| \quad (10)$$

where φ_i and $\hat{\varphi}_i$ is the ground-truth value and the estimated value of the rotated the i -th image. The lower the MAE value is, the more precise the orientation estimation is.

Besides, other practical metrics are also calculated to evaluate the performance of training and inference of PROAI. At the training stage, the training time and the number of parameters of CNNs used in PROAI and RDA are calculated and compared. Training time represents the total time required for all epochs of training, of which the unit is second (s). The number of parameters represents the Number of learnable parameters for a CNN. In this paper, mega (M) is used as the unit of the number of parameters. At the inference stage, the inference time, classification accuracies, and orientation estimation precisions of PROAI are calculated and compared with those of RDA. The inference time refers to the total time required for inferencing all the images in the test set, of which the unit is second (s).

3.2 Results at the Training Stage

This section evaluates the training time and number of parameters of CNNs training by PROAI. As a comparison, the results of CNNs trained with RDA are also provided. The comparison demonstrates that the training method of PROAI can effectively reduce both the training time and the number of parameters of CNNs.

² <https://github.com/automlresearch/pytorch-OpCounter/>.

Table 2 Architecture hyper-parameters, numbers of parameters, calculations of CNNs in the experiments

Train and val datasets	CNN name	Architecture hypermeters				Numbers of parameters		Numbers of computations	
		N1	M1	M2	N2	Params (M)	Params (%)	MACs	MACs (%)
Rotated MNIST	CNN (large)@M	40	2	4	5120	13.32	100.00	28.49	100.00
MNIST	CNN (small)@M	40	1.5	2	256	0.39	2.92	9.18	32.22
Rotated Fashion MNIST	CNN (large)@F	60	2	4	5120	20.04	100.00	53.77	100.00
Fashion MNIST	CNN (small)@F	60	1.5	2	256	1.31	6.54	19.99	37.28

3.2.1 Training Time

PROAI trains and validate CNN on the rot0 datasets of Rotated MNIST and Rotated Fashion MNIST. The CNN architectures designed in Table 2 are trained on these two datasets. Specifically, the CNNs with small numbers of parameters, CNN(small)@M and CNN(small)@F are applied to be trained on Rotated MNIST and Rotated Fashion MNIST, respectively. The cross-entropy loss function in Eq. (1) and the Stochastic Gradient Descent (SGD) algorithm are used to optimize the parameter of the two CNNs. For the training on these two datasets: the momentums of SGD are 0.9, the batch sizes of training images are 64, and the training epochs are 360.

As a comparison, RDA is applied to train and validate CNN on the rot180 or rot360 datasets of Rotated MNIST and Rotated Fashion MNIST. The CNNs with large numbers of parameters, CNN(large)@M and CNN(large)@F are applied to be trained on Rotated MNIST and Rotated Fashion MNIST, respectively. Other training hyperparameters are set to be the same as that of PROAI.

The curves for cross-entropy loss in relation to training epochs are shown in Fig. 7. The cross-entropy loss curves of RDA are the results of training CNNs on rot360. As shown in Fig. 7a and b, both the training and validation losses of PROAI decrease and converge more quickly than RDA. Also, the loss curves of PROAI are reduced to significantly lower values than that of RDA. These results demonstrate that PROAI is of a faster training convergence. Figure 7c and d can also confirm this argument.

To quantitatively evaluate how faster the training of PROAI is. The CNN(small)@M is trained by PROAI and the CNN(large)@M is trained by RDA on Rotated MNIST for many times. For each time of training, the overall training epochs is increased in [3, 480]. After each time of training, the classification accuracies on training and validation datasets are calculated using Eq. (9). Meanwhile, the overall training time is recorded. As a result, the training and validation accuracies in relation to training time can be drawn as Fig. 8. As it is shown, PROAI cost significantly shorter training time while achieve higher validation

accuracies. In addition, PROAI and RDA cost 84 and 390 epochs to achieve each best generalization performance. Under this condition, the training time of PROAI (t_{PROAI}) is only 10.3% of that of RDA (t_{RDA}).

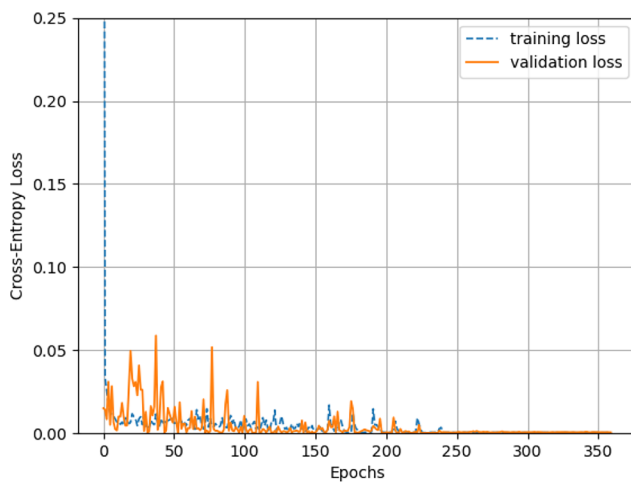
3.2.2 Numbers of Parameters

PROAI trains CNN on rot0 dataset, so it requires CNN with smaller numbers of parameters than RDA does. This is the reason that we have designed CNNs with small number of parameters CNN(small)@M and CNN(small)@F for PROAI, and CNN(large)@M and CNN(large)@F for RDA.

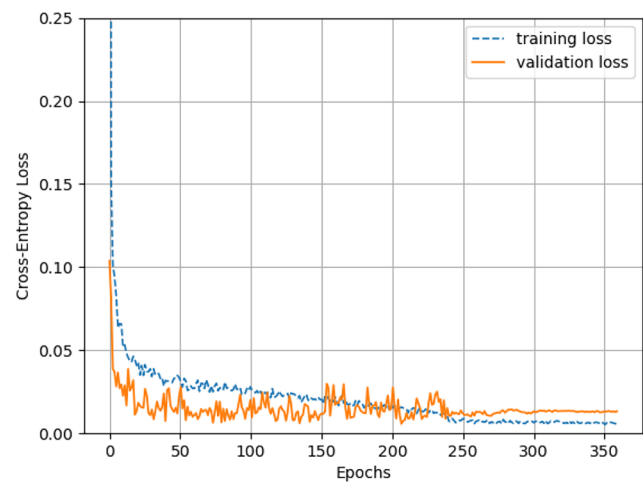
To validate that PROAI requires smaller numbers of parameters of CNN than RDA. RDA and PROAI are used to train both CNNs with larger and small numbers of parameters. The trained CNN is then applied to infer the images in the validation dataset. Meanwhile, the validation accuracies are calculated through Eq. (9). By observing the difference in validation accuracies, the preferences for the number of parameters of different methods can be concluded.

We train CNN(small)@M and CNN(large)@M on rotated MNIST using PROAI and RDA, and we can calculate the validation accuracies shown in Table 3. As can be seen, when training CNNs with PROAI, the CNN with a smaller number of parameters, i.e., CNN(small)@M, achieves higher validation accuracy. In contrast, when training CNNs with RDA, the CNN with a larger number of parameters, i.e., CNN(large)@M, achieves higher validation accuracy. These two results imply that PROAI requires smaller numbers of parameters of CNN than RDA. That is also to say that PROAI can reduce the number of parameters of CNNs. For Rotated MNIST dataset, the number of parameters of CNN(small)@M is only 2.92% of that of CNN(large)@M.

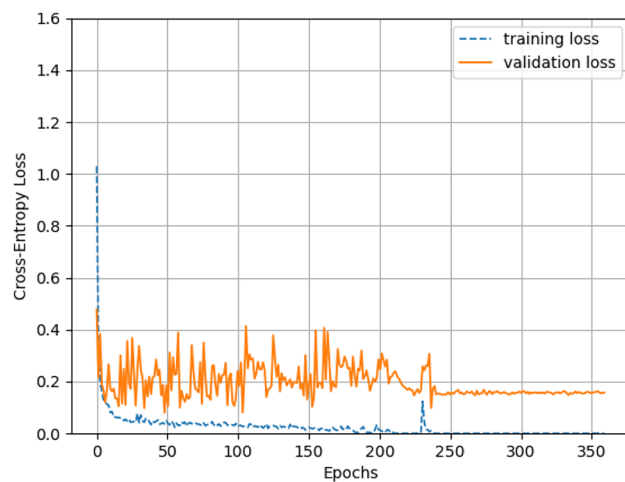
Moreover, we also train CNN(small)@F and CNN(large)@F on Rotated Fashion MNIST using PROAI and RDA, and we can calculate the validation accuracies shown in Table 4. As can be seen from the validation accuracies, PROAI makes the CNN with a smaller number of parameters achieve high validation accuracy, and RDA



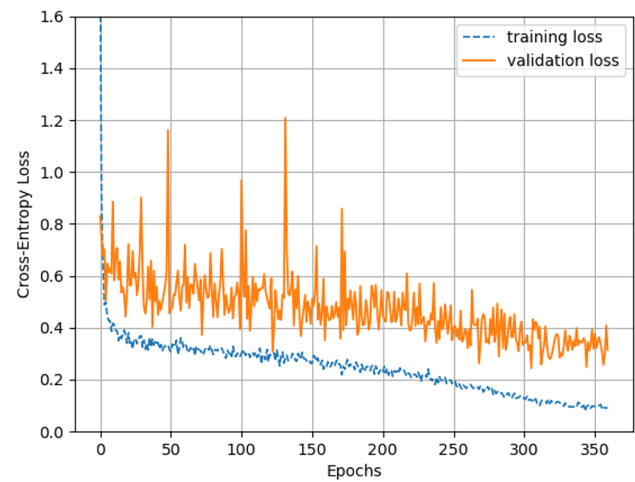
(a) PROAI on Rotated MNIST



(b) RDA on Rotated MNIST



(c) PROAI on Rotated Fashion MNIST



(d) RDA on Rotated Fashion MNIST

Fig. 7 Cross-entropy loss in relation to training epochs

makes the CNN with larger number of parameters achieve higher validation accuracy. These results also imply that PROAI can reduce the number of parameters of CNNs.

3.2.3 Discussions

Both the results in Tables 3 and 4 show that PROAI can reduce the number of parameters of CNN in RIR tasks. This is in agreement with the fact that the training set of PROAI contains images only at one orientation while the training set of RDA method contains images at all possible orientations. The variation of images within the training set of PROAI is smaller, so the optimal number of parameters is lower. The advantages of the reductions of the number of parameters is threefold: (1) the CNN requires less

storage space to save weight and less memory footprint to run. (2) The number of calculations of CNN is reduced. With the same network architecture and image size, the CNN with a lower number of parameters is usually less computationally intensive. As a result, the computation required by the CNN of PROAI is lower than that of the CNN of the data augmentation method. As shown in the “Numbers of Calculations” column in Table 2, the number of calculations of the CNN are reduced to 32.22% and 37.28% of that of the RDA method on the two datasets. (3) The CNN weights converge faster during training. As shown in Fig. 7, PROAI achieves the highest generalization using 84 training epochs, while the data augmentation method requires 390 epochs.

Fig. 8 Classification accuracies in relation to training time

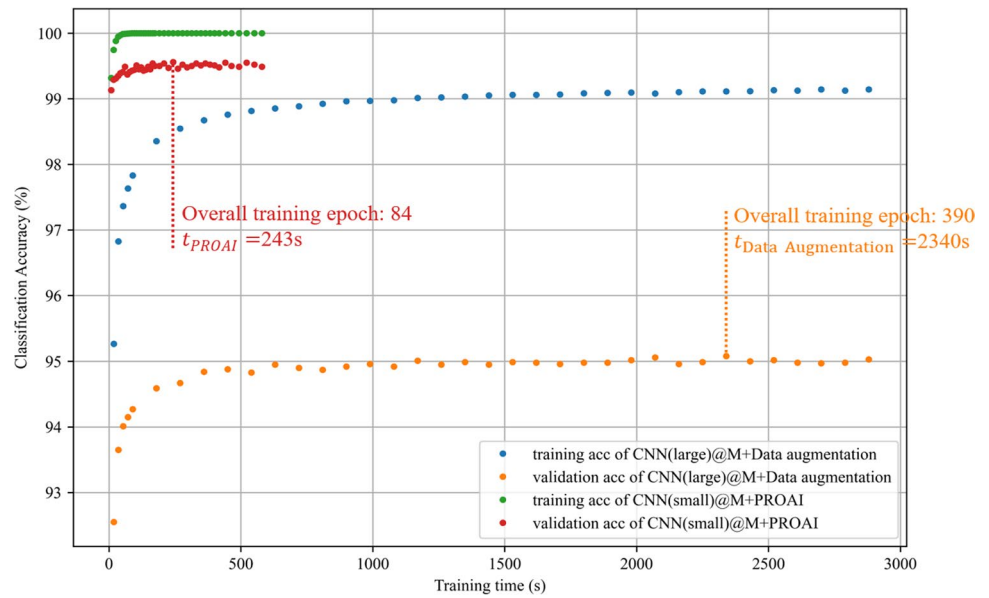


Table 3 Validation accuracies on Rotated MNIST of CNNs trained with PROAI and RDA

Training and validation datasets	Methods	Params (%)	Validation accuracies (%)
Rotated MNIST rot360—> rot360	RDA	CNN (large)@M	100.00
		CNN (small)@M	2.92
Rotated MNIST rot0—> rot0	PROAI	CNN (large)@M	100.00
		CNN (small)@M	2.92

Table 4 Validation accuracies on Rotated Fashion MNIST of CNNs trained with PROAI and RDA

Training and validation datasets	Methods	Params (%)	Validation accuracies (%)
Rotated MNIST rot360—> rot360	RDA	CNN (large)@M	100.00
		CNN (small)@M	2.92
Rotated MNIST rot0—> rot0	PROAI	CNN (large)@M	100.00
		CNN (small)@M	2.92

Due to the abovementioned advantages (2) and (3), the total training time of PROAI becomes significantly lower than that of RDA.

3.3 Results at the Inference Stage

This section first compares the classification accuracies achieved by PROAI with existing RIR methods, which can demonstrate that PROAI can achieve higher classification accuracies in RIR tasks.

Then this section compares orientation estimation precisions achieved by PROAI with existing RIR methods. But because existing RIR methods have not reported orientation

estimation precisions yet, two tailor-designed CNN angle regressors are trained using the orientation labels of rotated MNIST and rotated Fashion MNIST. The results of orientation estimation experiments can demonstrate that PROAI can achieve higher orientation estimation performance in RIR tasks.

Finally, the inference time of PROAI is quantitatively evaluated.

3.3.1 Classification Accuracies

This sub-section compares the classification accuracy achieved by PROAI with existing RIR methods. The

Table 5 Classification accuracy on rotated MNIST

Methods		Params (%)	Rot0—> Rot180 (%)	Rot0—> Rot360 (%)	Rot180—> Rot180 (%)	Rot360—> Rot360 (%)
Existing Methods	STN [21]	100.40	44.41	–	97.12	98.07
	ORN [7]	17.80	83.79	–	98.58	98.88
	TIPooling [8]	108.87	–	–	–	98.74
	DA (CNN(large)@M)	100	58.63	44.79	96.82	94.65
PROAI	CNN (small)@M	2.92	99.2	99.2	–	–

Table 6 The classification accuracies under different CNN architectures

Methods		Params (%)	Rot0—> Rot180 (%)	Rot0—> Rot360 (%)	Rot180—> Rot180 (%)	Rot360—> Rot360 (%)
DA	DA (ResNet50)	141.09	62.73	46.13	96.86	95.29
	DA (CNN (large)@M)	100	58.63	44.79	96.82	94.65
	DA (DARTSNet@12)	19.87	64.05	51.7	98.83	98.6
PROAI	ResNet18	87.83	97.4	97.4	–	–
	CNN (small)@M	2.92	99.2	99.2	–	–
	DARTSNet@6	9.38	99.37	99.37	–	–

classification accuracies in this section are calculated by applying Eq. (9) to the validation dataset. The rotated MNIST was first applied to calculate the classification accuracies of PROAI and RDA. For PROAI, the inference procedure proposed in Sect. 2.2.1 was conducted, each test image was rotated into 36 orientations to form 36 rotation channels, in which the CNN architecture and weights were shared with CNN(small)@M trained in Table 3. The classification accuracies of PROAI and existing RIR methods are shown in Table 5.

In the first row of Table 5, the text before the arrow indicates the dataset type used in the training stage. The text after the arrow indicates the dataset type used in the inference stage, the first column gives the numbers of parameters of each CNNs (take CNN(large)@M as the reference), and the second to fifth columns show the classification accuracies of each method on the rot0, rot180, and rot360 test sets, respectively. As can be seen, the results in “rot0—> rot180” and “rot0—> rot360” columns show that PROAI using CNN(small)@M significantly increases the classification accuracy, even though the training data is not augmented. For example, in the “rot0—> rot180” and “rot0—> rot360” columns, PROAI increase the classification accuracy to 99.2%. This result is not only higher than the RDA method but also greater than the existing state-of-the-art accuracy (98.88%) achieved by the rotation-invariant ORN.

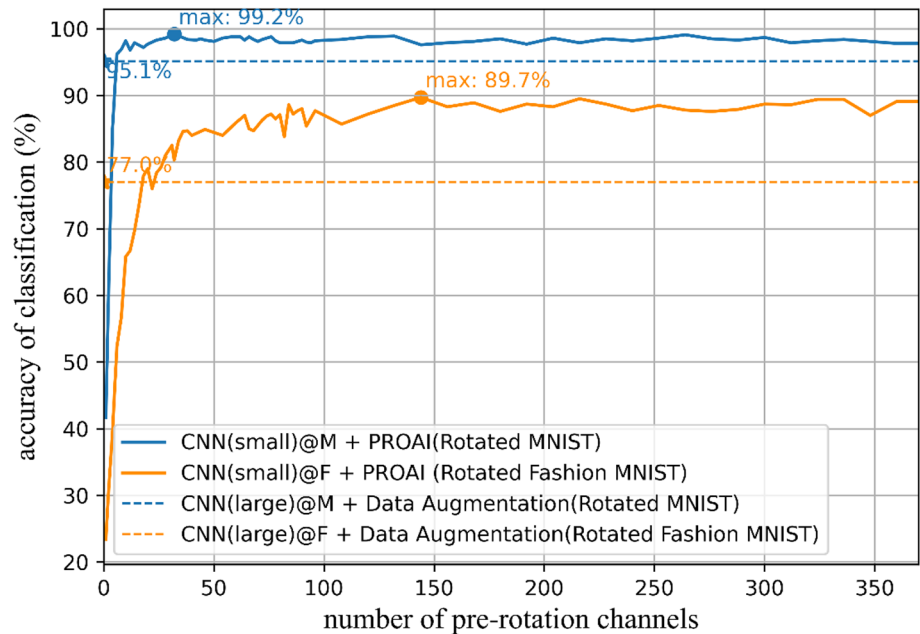
The results of CNN(small)@M in Table 5 has revealed that PROAI can use a simple CNN architecture to achieve higher RIR performance than RDA. To demonstrate PROAI can also generalize to other CNN architectures, we

also use two new CNN architectures to evaluate the RIR performance of PROAI. One of the architectures is one of the most popular CNN architectures ResNet [33]; another CNN architecture is called DARTSNet, which is automatically designed for MNIST dataset by a state-of-the-art Neural Architecture Search (NAS) algorithm, Differentiable Architecture Search (DARTS) [34]. DARTSNet@6, which is composed of 6 levels of computation cells, was applied in PROAI, while DARTSNet@12, which is composed of 12 levels of computation cells, was applied in RDA. The classification accuracies achieved by PROAI and RDA under different CNN architectures are shown in Table 6.

As shown in Table 6, in the first place, DARTSNet@6 can improve the classification accuracy of PROAI and achieve state-of-the-art classification accuracy 99.37%, while ResNet18 has decreased the performance of PROAI. A possible cause for the drop in accuracy is that the number of parameters of ResNet18 is still big for applying PROAI to rotated MNIST. This result again proves that, PROAI only requires CNN architectures with small numbers of parameters. In the second place, for each kind of CNN architecture, PROAI has achieved higher classification accuracies than RDA. This reveals again that PROAI increases classification accuracy. From the above results, two conclusions can be drawn: (1) when using the same CNN architectures in PROAI and RDA, PROAI can always outperform RDA on classification accuracy. (2) Designing appropriate CNN architecture can improve the RIR performance of PROAI, and the neural architecture search can be applied to design the CNN architectures for PROAI.

Table 7 Classification accuracies on rotated fashion MNIST

Methods	Params (%)	Rot0—> Rot180 (%)	Rot0—> Rot360 (%)	Rot180—> Rot180 (%)	Rot360—> Rot360 (%)
DA (CNN(large)@F)	100.00	32.07	22.81	79.77	77.0
PROAI (CNN(small)@F)	11.42	89.70	89.70	—	—

Fig. 9 Classification accuracies with respect to the Number of rotation channel

In addition to rotated MNIST, rotated fashion MNIST was also applied to calculate the classification accuracies of PROAI and RDA. For this dataset, the rotation channel N of PROAI was set to 144, the classification accuracy is shown in Table 7. As can be seen, the accuracy of PROAI is greater than RDA by 9.93%. This result implies that PROAI is also effective for Rotated Fashion MNIST, in which the image pattern is more complex.

To examine the relationship between the RIR performance of PROAI and the Number of pre-rotation channels, the “rot0- > rot360” classification accuracy curves of PROAI were calculated by adjusting the Number of pre-rotation channels in [1, 360]. The classification accuracy curve with respect to the number of rotation channels is shown in Fig. 9. As can be seen, on both datasets, the classification accuracies noticeably increase first and then stabilize when increasing the Number of rotation channel numbers. On rotated MNIST, the highest classification accuracy 99.2% is achieved when N is 36, which is 3.9% greater than the highest classification accuracy achieved by RDA. On Rotated Fashion MNIST, the highest classification accuracy is achieved when N is 144, which is 12.7% greater than the highest classification accuracy achieved by RDA.

Figure 9 also shows that the number of pre-rotation channels required for PROAI to achieve the best performance is different for different RIR datasets, which implies the inference time is also different. The inference time, the choice of N will be discussed in Sects. 3.3.3 and 3.3.4.

3.3.2 Orientation Estimation Precisions

This sub-section evaluates the orientation estimation precision achieved by PROAI. The evaluation metric of orientation estimation precision is MAE. By comparing the predicted value of orientation and the ground truth, the MAE for orientation estimation can be calculated using Eq. (10). Because the performance of PROAI is in relation to the Number of pre-rotation channels, the MAEs achieved by PROAI were calculated in the condition of different numbers of pre-rotation channels. The numbers of pre-rotation channels are increased from 1 to 360. As a result, the MAE curves are plotted with blue and orange lines in Fig. 10.

For comparison, the MAEs achieved by RDA were also calculated. But because existing RIR methods have not reported the results for orientation estimation task, the MAE for orientation estimation of PROAI was compared with a tailor-designed CNN angle regressor. This angle

Fig. 10 Mean absolute error of orientation estimation with respect to the Number of rotation channels

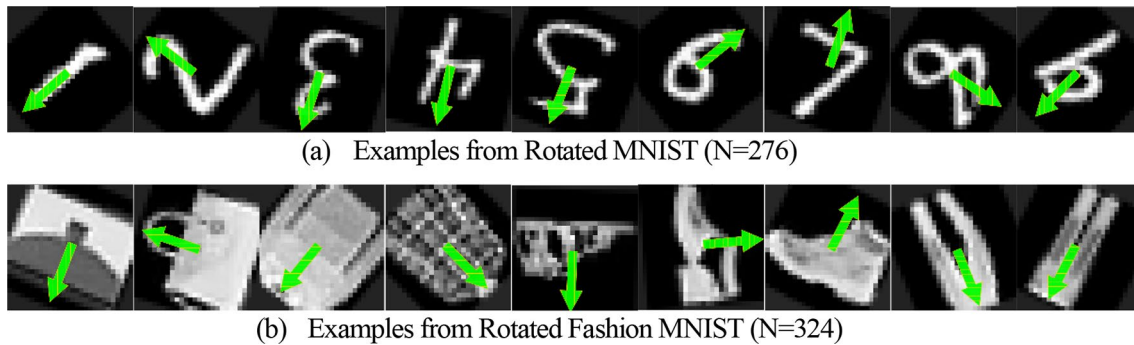
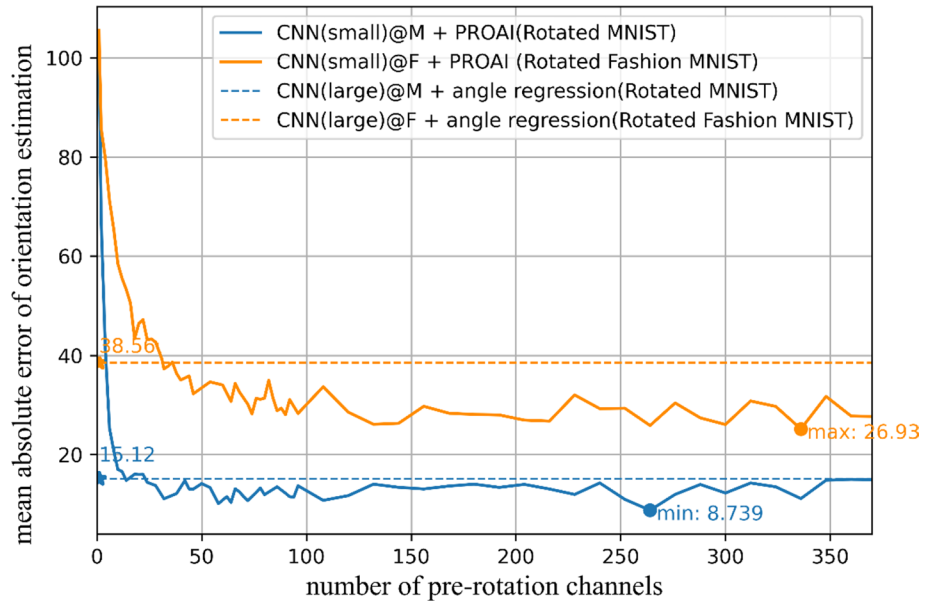


Fig. 11 Examples of rotated images with the orientations estimated by PROAI

regressor was designed by adding two output neurons to CNN(large)@M and CNN(large)@F, and these two neurons are responsible for regressing the sine and cosine values of the image rotation angle [22, 23]. Since the supervised orientation regressors need to be trained with images at any possible rotation angles, they are also taken as RDA methods. The blue and orange dash lines in Fig. 10 indicate the MAEs on Rotated MNIST, Rotated Fashion MNIST achieved by supervised orientation regressors.

It can be observed that the MAE curves of PROAI noticeably decrease first and then stabilize when increasing the rotation channel numbers. On rotated MNIST, the lowest MAE 8.739° is achieved when N is 276, which is 6.38° lower than what can be achieved by the supervised angle regressor. On Rotated Fashion MNIST, the lowest MAE 26.93° is achieved when N is 324, which is 11.63° lower than the supervised angle regressor.

Figure 10 demonstrates that PROAI can estimate orientation more precisely than the supervised angle regressor. Examples of rotated images labeled with the orientations estimated by PROAI are shown in Fig. 11.

3.3.3 Inference Time

This sub-section compares the inference time of PROAI with existing RIR methods. The inference time here refers to the total time required for inferring all images in test sets. In the experiment, parallel computing is adopted for inference, and the batch size of a test set is 64; the computer used in the inference experiment is equipped with a Core i9 CPU and a NVIDIA 3090 GPU.

To observe the relationship between the pre-rotation channel numbers and the inference time, the pre-rotation channel number gradually increases, and the corresponding

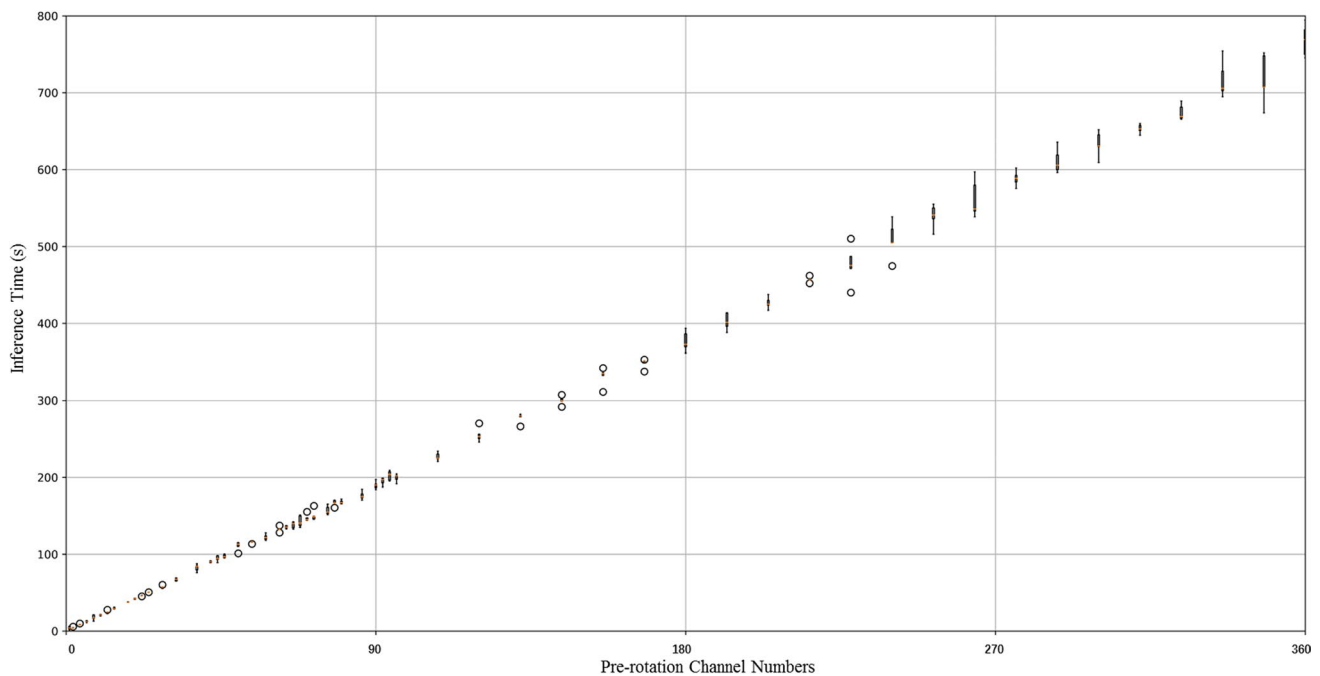


Fig. 12 Inference time of PROAI in relation to pre-rotation channel numbers

inference time is recorded. For each Number of pre-rotation channels, the inference experiment is conducted 5 times to record 5 independent measures of the inference time. These five different measures of the inference time are plotted using the boxplot. Then, the inference time curve of PROAI is shown in Fig. 12. As shown in Fig. 12, the inference time of PROAI has a linear relationship with the pre-rotation channel number. Also, it can be observed from Fig. 12 that the variance of the inference time increases with the pre-rotation channel number.

Figures 9 and 10 have demonstrated that the classification and the orientation estimation performance of PROAI increase with the pre-rotation channel number. Figure 12 has demonstrated the inference time of PROAI increases with the pre-rotation channel number. Next, it is valuable to study how many pre-rotation channel numbers are required by PROAI to outperform the classification and the orientation estimation performance of RDA. To answer this question, the pre-rotation channel number of PROAI is gradually increased. Then the classification performance and the orientation estimation performance are evaluated for each pre-rotation channel number. As a result, the curves for classification accuracies and mean absolute errors of orientation estimation are shown in Fig. 13. As a comparison, the results of RDA are added.

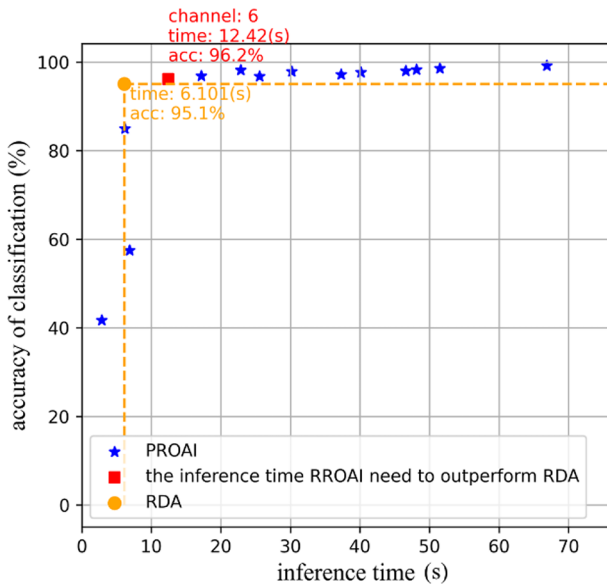
Figure 13a, b shows the classification accuracies with respect to inference time. The classification accuracies of PROAI in relation to the inference time are plotted with blue stars, and the inference time of RDA in relation to the

inference time is plotted with an orange circle. As these two figures shown, the inference time and classification accuracy achieved by single-channel PROAI is lower than the RDA, but the classification accuracies become greater when the inference time increases. PROAI outperforms RDA when the numbers of the pre-rotation channel are 6 and 18. These two points are plotted with red squares in the two figures, and the corresponding inference time of PROAI is 12.66 s and 79.74 s, which are 2.079 and 6.254 times that of RDA.

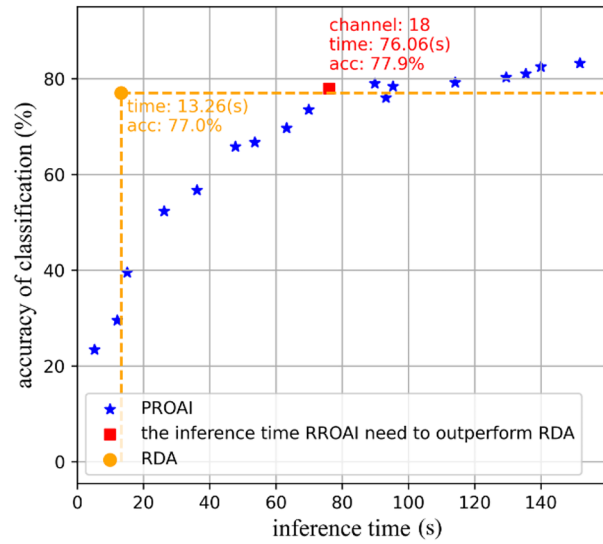
Figure 13c, d shows the MAEs for orientation estimation with respect to inference time. The MAEs for orientation estimation of PROAI in relation to the inference time are plotted with blue stars, and the MAE of RDA in relation to the inference time is plotted with an orange circle. As these two figures shown, when using a single channel, the inference time of PROAI is lower than RDA, while its MAEs for orientation estimation are higher than RDA. When increasing the pre-rotation channels, the MAEs become lower, but the inference time increases. When the Number of the pre-rotation channels increases to 8 and 24, PROAI outperforms RDA methods on Rotated MNIST and Rotated Fashion MNIST. These two points are plotted with red squares in the two figures, and the corresponding inference time of PROAI is 16.88 s and 106.3 s, which are 2.772 and 8.337 times that of RDA.

3.3.4 Discussions

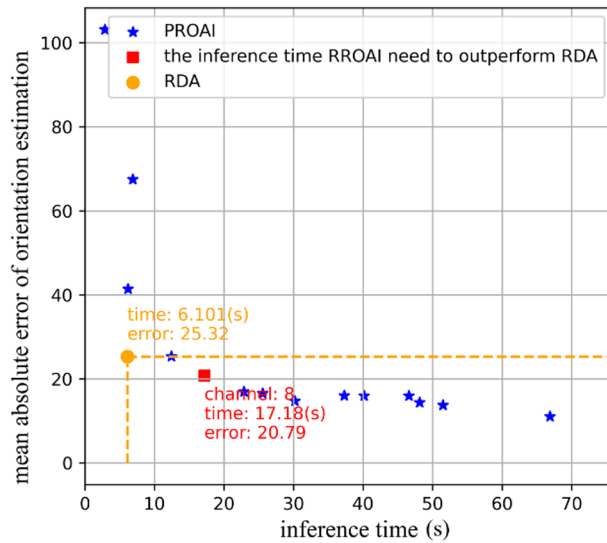
Figures 8 and 9 demonstrate that the RIR performance increase first and then stabilize when increasing the rotation



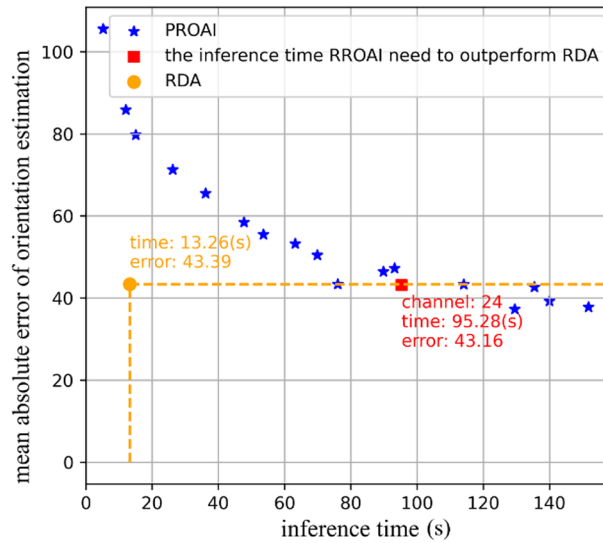
(a) Classification accuracies on Rotated MNIST vs inference time



(b) Classification accuracies on Rotated Fashion MNSIT vs inference time



(c) Mean absolute errors on Rotated MNIST vs inference time



(d) Mean absolute errors on Rotated Fashion MNSIT vs inference time

Fig. 13 Classification accuracies and mean absolute errors of orientation estimation vs inference time

channel numbers. Different datasets require a different number of rotation channels to achieve each highest RIR performance. For the Rotated MNIST dataset, PROAI achieves the highest classification accuracy when N is set to 36, but for the Rotated Fashion MNIST dataset, PROAI achieves the highest classification performance when N is set to 144. The reason that the PROAI requires different numbers of pre-rotation channels on the two datasets is that the orientations of handwritten digits in MNIST are distributed in a certain range, so that the CNN trained directly on this

dataset can recognize handwritten digits in a certain range of orientations rather than only a single orientation. As a result, a smaller number of pre-rotation channels is able to make PROAI achieve the highest classification accuracy on Rotated MNIST. By contrast, since the image orientations of Fashion MNIST are more concentrated in a small range, the trained CNN can only recognize images within a small range of orientations. As a result, it requires more rotation channels for PROAI to achieve high classification accuracy on Rotated Fashion MNIST. As can be seen from the above

analysis, whether the image orientation is normal or not, it is effective to use PROAI's weight-sharing mechanism in the inference stage to generalize the CNN's ability to recognize image at one or some orientations to any other arbitrary orientation. This implies that PROAI has the function of improving the generalization abilities of any trained CNNs. Since this inference procedure and its function are similar to test-time augmentation, future research on PROAI can focus on the explanation of test-time augmentation.

The results in Fig. 11 show that the inference time of PROAI is proportional to the number of channels, which is dependent on the dataset type, the task type, and the expected performance. In other words, PROAI can adjust the rotation image recognition performance by adjusting the inference time. This recognition performance includes rotated classification performance and additional orientation estimation performance that are obtained without learning orientation labels. To outperform RDA methods, different additional inference time is required by PROAI in different datasets. Therefore, it is necessary to set a reasonable number of pre-rotation channels for the best trade-off between RIR performance and inference time. In the experiments of Sect. 3.3.1, no greater than three times of the inference time is sufficient to obtain better RIR performance than RDA. In the experiments in Sect. 3.3.2, no greater than nine times of the inference time is sufficient to obtain better orientation estimation performance than RDA. Although PROAI requires longer inference time, it should be emphasized that the computation of each channel of PROAI is completely independent, so parallel computation can be applied to effectively accelerate the inference in practice.

In summary, the experiments in Sects. 3.3.1 and 3.3.2 demonstrate that PROAI can achieve state-of-the-art performance in both the rotated image classification and the orientation estimation task on both Rotated MNIST and Rotated Fashion MNIST datasets.

4 Conclusion

While existing rotated image recognition methods focus on making CNN "memorize" as many images as possible during the training stage, this paper has proposed a novel rotated image recognition mechanism, PROAI, which simulates the mental rotation process of the human brain. At the training stage, images at only one orientation are learned by CNN. At the inference stage, images at any other orientation are fed into a cluster of CNNs sharing the same architecture and weight to calculate classification scores, of which the maximum value has been successfully applied to simultaneously estimate both the category and the orientation of each test image. PROAI has significantly reduced the parameters and

training time of CNN in RIR tasks and also achieved state-of-the-art classification accuracies and orientation estimation precisions on several datasets.

The main limitations associated with the PROAI method is that the multi-channel inference architecture of PROAI costs more computation power, therefore, the inference time of PROAI is proportional to the Number of channels. However, since the computation of each channel of PROAI is completely independent, parallel computation can be applied to effectively accelerate the inference in practice. In addition, it is necessary to process the trade-off between performance and inference time to set a reasonable number of pre-rotation channels so that we can achieve both high accuracy and high inference speed.

PROAI achieves state-of-the-art performance on the rotated digits recognition and rotated fashion recognition tasks. Since the inference procedure of PROAI is similar to test-time augmentation, it holds promise using the method of PROAI to explain the test-time augmentation. Also, it would be beneficial to generalize PROAI to process other kinds of image transformations, such as scale transformation, etc.

Author Contributions Yue Fan Writing: original draft preparation; review and editing; Peng Zhang made substantial contributions to conception and design, literature searches and analyses; Jingqi Han participated in revising the article and gave final approval of the version to be submitted; Dandan Liu: conception and design of the manuscript and interpretation of data, literature searches and analyses, clinical evaluations; Jinsong Tang participated in revising the article and gave final approval of the version to be submitted; Guoping Zhang participated in revising the article and gave final approval of the version to be submitted.

Funding This work is supported by the National Natural Science Foundation of China (No.61901503).

Availability of Data and Materials The datasets generated during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The author declares that there is no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Alex, K., Ilya, S., Geoffrey, H.: ImageNet classification with deep convolutional neural networks. *Commun. ACM* **60**, 84–89 (2017)
- LeCun, Y.: Generalization and network design strategies. *Connect. Perspect.* **19**, 143–155 (1989)
- Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, December 4–9, 2017, Long Beach, CA, USA, pp. 3856–3866 (2017)
- Mei, S., Jiang, R., Ma, M., et al.: Rotation-invariant feature learning via convolutional neural network with cyclic polar coordinates convolutional layer. *IEEE Trans. Geosci. Remote Sens.* **61**, 1–13 (2023)
- Quiroga, F.M., Torrents-Barrena, J., Lanzarini, L.C., et al.: Invariance measures for neural networks. *Appl. Soft Comput.* **132**, 109817 (2023)
- Marcos, D., Volpi, M., Komodakis, N., et al.: Rotation equivariant vector field networks. In: *IEEE International Conference on Computer Vision, ICCV 2017*, Venice, Italy, October 22–29, 2017, pp. 5058–5067 (2017)
- Zhou, Y., Ye, Q., Qiu, Q., et al.: Oriented response networks. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, Honolulu, HI, USA, July 21–26, 2017, pp. 4961–4970 (2017)
- Laptev, D., Savinov, N., Buhmann, J.M., et al.: TI-POOLING: transformation-invariant pooling for feature learning in convolutional neural networks. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, Las Vegas, NV, USA, June 27–30, 2016, pp. 289–297 (2016)
- Cohen, T., Welling, M.: Group equivariant convolutional networks. In: *Proceedings of the 33rd International Conference on Machine Learning, ICML 2016*, New York City, NY, USA, June 19–24, 2016, pp. 2990–2999 (2016)
- Worrall, D.E., Garbin, S.J., Turmukhambetov, D., et al.: Harmonic networks: deep translation and rotation equivariance. *CVPR 2017*, 7168–7177 (2017)
- Bruintjes, R.-J., Motyka, T., van Gemert, J.: What affects learned equivariance in deep image recognition models? *CoRR abs/2304.02628* (2023)
- Simard, P.Y., Steinkraus, D., Platt, J.C.: Best practices for convolutional neural networks applied to visual document analysis. In: *7th International Conference on Document Analysis and Recognition (ICDAR 2003)*, 2-Volume Set, 3–6 August 2003, Edinburgh, Scotland, UK, pp. 958–962 (2003)
- Zheng, X., Sun, H., Lu, X., et al.: Rotation-invariant attention network for hyperspectral image classification. *IEEE Trans. Image Process.* **31**, 4251–4265 (2022)
- Li, J.: Rotation equivariance of deep convolutional neural network (in Chinese). A Dissertation Submitted to Zhejiang University for the Degree of Master of Engineering, 4th March (2019)
- Shi, Y., Fu, B., Wang, N., et al.: Spectral-spatial attention rotation-invariant classification network for airborne hyperspectral images. *Drones* **7**(4), 240 (2023)
- Fang, G., Ba, S., Gu, Y., et al.: Automatic classification of galaxy morphology: a rotationally-invariant supervised machine-learning method based on the unsupervised machine-learning data set. *Astron. J.* **165**(2), 35 (2023)
- Gens, R., Domingos, P.M.: Deep symmetry networks. In: *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014*, December 8–13 2014, Montreal, Quebec, Canada, pp. 2537–2545 (2014)
- Dieleman, S., Fawc, J.D., Kavukcuoglu, K.: Exploiting cyclic symmetry in convolutional neural networks. In: *Proceedings of the 33rd International Conference on Machine Learning, ICML 2016*, New York City, NY, USA, June 19–24, 2016, pp. 1889–1898 (2016)
- Mo, H., Zhao, G.: RIC-CNN: rotation-invariant coordinate convolutional neural network. *CoRR abs/2211.11812* (2022)
- Wei, C., Ni, W., Qin, Y., et al.: RiDOP: a rotation-invariant detector with simple oriented proposals in remote sensing images. *Remote Sens.* **15**(3), 594 (2023)
- Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. In: *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015*, December 7–12, 2015, Montreal, Quebec, Canada, pp. 2017–2025 (2015)
- Massa, F., Marlet, R., Aubry, M.: Crafting a multi-task CNN for viewpoint estimation. In: *Proceedings of the British Machine Vision Conference 2016, BMVC 2016*, York, UK, September 19–22, 2016 (2016)
- Penedones, H., Collobert, R., Fleuret, F., et al.: Improving Object Classification using Pose Information. L'IDIAP Laboratory, École Polytechnique Fédérale de Lausanne. <https://infoscience.epfl.ch/record/192574> (2012)
- Koriat, A., Norman, J.: What is rotated in mental rotation? *J. Exp. Psychol. Learn. Memory Cognit.* **10**(3), 421–434 (1984)
- Shepard, R.N., Metzler, J.: Mental rotation of three-dimensional objects. *Science (New York, N.Y.)* **171**(3972), 701–703 (1971)
- Sun, F., Morita, M., Stark, L.W.: Comparative patterns of reading eye movement in Chinese and English. *Percept. Psychophys.* **37**(6), 502–506 (1985)
- Lecun, Y., Bottou, L., Bengio, Y., et al.: Gradient-based learning applied to document recognition. *Proc. IEEE* **86**(11), 2278–2324 (1998)
- Xiao, H., Rasul, K., Vollgraf, R.: Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *CoRR abs/1708.07747* (2017)
- Larochelle, H., Erhan, D., Courville, A.C., et al.: An empirical evaluation of deep architectures on problems with many factors of variation. In: *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007)*, Corvallis, Oregon, USA, June 20–24, 2007, pp. 473–480 (2007)
- Shi, X., Shan, S., Kan, M., et al.: Real-time rotation-invariant face detection with progressive calibration networks. In: *CVPR 2018*, Salt Lake City, USA, pp. 2295–2303 (2018)
- Zhang, P., Tang, J., Zhong, H., et al.: Self-trained target detection of radar and sonar images using automatic deep learning. *IEEE Trans. Geosci. Remote Sens.* (2021). <https://doi.org/10.1109/TGRS.2021.3096011>
- Bjorck, J., Gomes, C.P., Selman, B., et al.: Understanding batch normalization. In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018*, December 3–8, 2018, Montréal, Canada, pp. 7705–7716 (2018)
- Kaiming, H., Xiangyu, Z., Shaoqing, R., et al.: Deep residual learning for image recognition. In: *IEEE Conference on Computer Vision*, pp. 770–778 (2016)
- Hanxiao, L., Karen, S., Yiming, Y.: DARTS: differentiable architecture search. In: *7th International Conference on Learning Representations, ICLR (2019)*

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.