PAPER

Rethinking the Rotation Invariance of Local Convolutional Features for Content-Based Image Retrieval

Longjiao ZHAO^{†a)}, Yu WANG^{††}, *Nonmembers*, and Jien KATO^{††}, *Member*

SUMMARY Recently, local features computed using convolutional neural networks (CNNs) show good performance to image retrieval. The local convolutional features obtained by the CNNs (LC features) are designed to be translation invariant, however, they are inherently sensitive to rotation perturbations. This leads to miss-judgements in retrieval tasks. In this work, our objective is to enhance the robustness of LC features against image rotation. To do this, we conduct a thorough experimental evaluation of three candidate anti-rotation strategies (in-model data augmentation, in-model feature augmentation, and post-model feature augmentation), over two kinds of rotation attack (dataset attack and query attack). In the training procedure, we implement a data augmentation protocol and network augmentation method. In the test procedure, we develop a local transformed convolutional (LTC) feature extraction method, and evaluate it over different network configurations. We end up a series of good practices with steady quantitative supports, which lead to the best strategy for computing LC features with high rotation invariance in image retrieval.

key words: image retrieval, in-model data augmentation, in-model feature augmentation, post-model feature augmentation, local convolutional features, and rotation invariance

1. Introduction

In recent years, content-based image retrieval has received sustained attention in the computer vision community. Different with conventional object recognition tasks, which typically classify object categories with significant visual differences, the objective of image retrieval is to discriminate similar individual objects, such as different buildings. Robust image retrieval requires comprehensive image details to be captured in an invariant way. So, it is a challenging task.

In early works, low level hand-crafted visual features are mainly used to capture different aspects of images details, such as color [1]–[5], shape [6], texture [7], [8] and spatial layout [9], [10]. Recently, with the rapid popularization of convolutional neural networks (CNN) [11], [12], it has led to attempts that use pre-trained CNNs to produce the image features. Especially, local convolutional (LC) features, which are produced by the last convolutional layer in a CNN, have been studied in many works [13]–[15] and confirmed to be useful in image retrieval.

Fig. 1 The effect of rotation attacks. From top to bottom are retrieval tasks which are conducted with an original dataset, an attacked query, and an attacked database. The first column is the query image, and the rest of the columns represent the top 5 results. Boxes in red and green represent the wrong and right answer, respectively. After rotation attack, all of the retrieval results are incorrect.

Although LC features have shown better performance on the image retrieval than hand-crafted local features, they still have some significant limitations such as lack of transformations invariance, especially for rotations. For example, as show in Fig. 1, the state-of-the-art method called R-MAC [15] gives a good performance on the original Oxford5k dataset (the first row); however, when the database (the second row) or query (the third row) is attacked by rotation perturbations, the retrieval performance appears a huge decrease. This is because the LC features obtained by CNNs are designed to be translation invariant, but they are inherently sensitive to rotation transformation. This congenital shortage could lead to unimaginable miss-judgements in retrieval tasks, such as the results shown in Fig. 1. How to make the LC features become robust to rotation is an important issue.

From the viewpoint described above, this paper focuses on how to advance the rotation invariance of local convolutional (LC) features by analyzing the performance under different types of rotational attacks. In general, there are two cases where the dataset can be attacked. One is the database attack (A_d), which means that the retrieval model is required to find a target among images rotationally transformed. The other is query attack (A_q) which indicates that the model needs to find the target using a rotated query image. In this work, with the purpose of producing rotation-invariant LC

Manuscript received January 29, 2020.

Manuscript revised August 18, 2020.

Manuscript publicized October 14, 2020.

[†]The author is with the Graduate School of Informatics, Nagoya University, Nagoya-shi, 464–8601 Japan.

^{††}The authors are with the College of Information Science and Engineering, Ritsumeikan University, Kusatsu-shi, 525–8577 Japan.

a) E-mail: zhao.longjiao@e.mbox.nagoya-u.ac.jp

DOI: 10.1587/transinf.2020EDP7017

features, we propose three candidate strategies: in-model data augmentation, in-model feature augmentation and post-model feature augmentation. The in-model data augmentation strategy hypothesizes that a CNN that is trained to be rotation invariant can produce LC features with high rotation invariance, and train such a CNN by heavily augmenting the training images using image rotation. The in-model feature augmentation strategy attempts to rotate feature maps of networks so that more abundant rotation information can be given to the models. The post-model feature augmentation strategy extracts the features from multiple orientation of training images. This is done by feeding rotated images to a fixed CNN, and then collecting the activations from multiple additional orientations.

In the remaining of this paper, we conduct a thorough experimental evaluation on these three candidate strategies with respect to the two kinds of attack (A_d and A_q), to find out some meaningful conclusions on how effectively each strategy can promote rotation invariance of LC features. The experiments are conducted on multiple benchmark datasets using multiple CNN architectures. We end up a series of good practices with steady quantitative supports, which lead to a best method for computing LC features with high rotation invariance in image retrieval.

The contributions of our work are: (1) We conduct a detailed and systematic evaluation on rotation invariance of LC features for image retrieval. (2) We propose the local transformed convolutional (LTC) features, which is robust against image rotation. (3) We figure out the chief factors that contribute to advancing rotation invariance for in-model and post-model strategies. (4) We summarize a series of good practices for extracting LC features with high rotation invariance.

2. Related Work

Content-based image retrieval. Most image retrieval approaches comprise a common pipeline that can be naturally divided into three parts: (1) local feature extraction; (2) image-level feature encoding; and (3) image-image similarity calculation. Local feature extraction is at the core of image retrieval issue. In early works, low level hand-crafted visual features, for example, scale-invariant feature transform (SIFT) [16] and histogram of oriented gradients (HOG) descriptors [17] are mainly used. At the image level, the Bag of Visual Words framework has been used to propose encoding methods such as voting-based encoding [18], reconstruction-based encoding [19], and super vectors [20], [21]. The Euclidean distance and cosine similarity are generally used to compare similarities between different global features. Moreover, the distance matrix learning is also a popular way in image retrieval, as shown in the [22]–[24]. In this work, we mainly discuss the methods that generate the local features with high rotation invariance.

CNN based features. Initially, the activations of fully connected layers are utilized as global features in many works such as [25] and [13]. Deep features, or the fea-

ture maps in the last convolutional layer, are currently receiving attention as they can be extracted efficiently, and can be naturally interpreted as local features ([13], [26] and [14]) of images. Most recent work has focused on how to aggregate deep features to obtain good global features. In [13] and [14], deep features were aggregated through sum-pooling based on the geometric characteristics of images and max pooling, respectively. Both techniques obtained very discriminative global features and outperformed results obtained by other techniques. R-MAC [15] encodes local features by calculating regional maximum features and has also been shown to obtain higher accuracy than most state-of-the-art methods. In this work, we incorporate R-MAC into the baseline model for all evaluations (Fig. 2.C).

Rotation invariance. In the paper [27], it shows the low capability of CNN-based features for resisting the rotation transformation, by comparing with fisher vector [22]. Generally, to advance this capability, many works ([28], [29], and [30]) proposed solution from two aspects which are network fine-tuning and feature extraction. The work [31] proposed a model named spatial transformer network (STN) to defy the rotation by studying the optimized crop or transformation to images (or activation layers). The purpose of STN is to alien the object of the image in the right angle and position automatically. Data augmentation is also a valid way to advance the rotation invariance of the network. The work [32] implements the rotations and flips on the input images randomly. The network will learn the information from different angles of an image so that the network will produce similar feature maps to the rotated images from the same data. For the other aspect, feature extraction, [27] showed that the rotation invariance of CNN activations can be advanced by fusing global features from different orientations of an image.

Based on these related works, we propose three strategies for advancing the rotation invariance of CNN features for image retrieval. Instead of aligning the input image, in-model feature augmentation is a pooling method to advance the rotation invariance by rotating feature maps randomly. In-model data augmentation provides a protocol to train a CNN that can produce similar feature maps of an image and its rotated version. Moreover, different with [27], post-model feature augmentation presents an extraction method (local transformation convolutional features) based on LC features, which include more detail information than global features. We conduct thorough and systematic experiments to evaluate the effectiveness of them.

We implement all experiments on three datasets which are Oxford Building [33], Paris [34], and Retrieval-SfM-30k [35]. Three popular networks, VGG16 [12], ResNet50 [36] and DenseNet121 [37], are employed in this work.

3. Ideas and Methods

Our objective is to enhance the robustness of LC features against image rotation. There are three key factors that influ-

Fig. 2 The outline of proposed strategies. A and C show the general flows during training (in-model) and test (post-model) procedures in image retrieval. B1, B2 and D represent the proposed methods, which are in-model data augmentation (DA), in-model feature augmentation (rotation invariance pooling) and post-model feature augmentation (local transformed convolutional features), on a single branch.

ence decisively rotation invariance of LC features, which are dataset, network structure, and feature extraction. Regarding the first factor, we introduce an in-model data augmentation strategy, and concerning the second and third factors, we propose in-model and post-model feature augmentation strategies, respectively. In this section, we describe how to implement these strategies in training procedure (in-model strategy) and test procedure (post-model strategies).

3.1 In-Model Data Augmentation

A direct interpretation for why the CNNs show high sensitivity to rotation transformation is that training data are usually not abundant enough for learning such rotation invariance. Therefore, fine-tuning a pre-trained network by feeding the images after some rotation transformation is supposed to be helpful to advance rotation invariance. It is necessary to evaluate the effects of the ratio of rotated images to non-rotated images (in the training data) on rotation invariance.

Based on this idea, we randomly select input images, rotate them into a random angle $\alpha \in [0^\circ, 360^\circ]$, and add these images to the dataset, which is used for fine-tuning the model. We define a rotation parameter k that means the rotation percentage of input data in each iteration of fine-tuning (e.g. $k = 0.3$ means 30% images are rotated). In the experiments described later, we set the k to be a number of values, for instance, $[0.3, 0.5, 0.7]$ corresponding to the low, middle, and high ratios of rotated input images, to investigate quantitatively how data augmentation affects rotation invariance. This is so-called in-model data augmentation (DA) strategy. Figure 2.B1 shows architecture of DA.

3.2 In-Model Feature Augmentation

The other reason that the network cannot resist the rotation perturbations is supposed to be that the pooling layers of the network do not have enough compatibility with image rotations. Strictly speaking, since simple max pooling for the convolutional layers has a limitation on spatial invariance of feature location, CNNs are inherently sensitive to rotation transformations of input data. Based on this idea, we propose a new architecture called rotation invariance pooling (RIP), which is expected to advance the anti-rotation capability of the network itself. We call this in-model feature augmentation.

The RIP improves the spatial invariance of the feature position by rotating randomly the feature maps of the network that are learned from the original input data. Theoretically, it will increase the compatibility for defying rotation of the model. The RIP is composed of a rotation layer and a down-sampling layer. Inspired by the spatial transformed network (STN), the rotation layer is to generate a grid by using affine transformation shown in Eq. (1):

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}, \quad (1)$$

where the x_i^s and y_i^s are the coordinates of source image, and the x_i^t and y_i^t are coordinates of the target image. The θ means the rotation angle controlled by an activation function $H(r)$ and an angle α as shown in Eq. (2):

$$\begin{aligned} \theta &= H(r) \cdot \alpha, \quad \alpha \in [10^\circ, 350^\circ] \text{ in step of } 10^\circ; \\ H(r) &= \begin{cases} 1, & r \geq \text{margin} \\ 0, & \text{others} \end{cases}, \quad r \sim U(0, 1), \end{aligned} \quad (2)$$

where r and *margin* respectively represent the activation factor and activation margin. The activation factor r follows uniform distribution $U(0, 1)$. When r is larger than the activation margin, the rotation operation is activated, otherwise, the rotation operation is non-activated. Furthermore, we utilize the bilinear sampling to generate and down-sample the feature maps. Figure 2.B2 shows architecture with single RIP.

3.3 Post-Model Feature Augmentation

Sections 3.1 and 3.2 discussed how to promote the rotation invariance of LC features in the training procedure. In this section, we focus on the way to advance the rotation invariance of LC features in the test procedure by using a fixed pre-trained network. Generally, as already mentioned, the pre-trained CNNs are inherently sensitive to image rotations because of a lack of orientation information. We hypothesize that the rotation invariance of LC features can be advanced by extracting and assembling the features that are related to rotation transformation with various rotation angles.

We propose a method to obtain local convolutional features which contain multi-angle information of an image. The features produced by this method are called local transformed convolutional features (LTC features). We define that t is a rotation transformation matrix. To extract rotation information, the image I need to be transformed into $I \cdot t$, before feeding into the network. The feature maps of the rotated image $I \cdot t$ is expressed as X_{It} with the dimension $m' \times n' \times D$ ($m' \times n'$ is the size of the image, and D is the number of feature maps). To recover the correspondences concerning the original input image I , the inverse transformation t^{-1} is applied on the feature maps X_{It} , which results in the transformed feature maps as

$$X_I^t = X_{It} \cdot t^{-1}, \quad (3)$$

where t^{-1} is the inverse matrix of t which represents the inverse transformation.

We defined a set of transformation matrices $T = \{t_i\}, i = 1, \dots, K$, and fusion function F . Therefore, the LTC feature can be described by

$$f_{LTC} = F(X_I, X_I^{t_1}, \dots, X_I^{t_i}, \dots, X_I^{t_k}), X_I^{t_i} \in \mathbb{R}^{m \times n \times D}. \quad (4)$$

Using this feature extraction method in the test procedure is called post-model feature augmentation. Figure 2.D shows architecture of LTC.

4. Experiments

In this section, we give the details of our experiments that are conducted for verify the ideas and methods described in Sect. 3.

4.1 Dataset

Oxford5k and Paris6k. The Oxford Building dataset (Ox-

ford5k) and Paris dataset (Paris6k) are published by a research group from Oxford university [33], [34]. They include 5,062 images and 6,412 images, respectively. Of which, Oxford5k include 11 landmarks and Paris6k include 12 landmarks. There are 55 query images for both datasets. Oxford5k and Paris6k are only used in the test procedure.

Attacked dataset. As mentioned before, we focus on two attacks cases: database attack (A_d) and query attack (A_q). The attacked datasets of these two cases are produced based on the Oxford5k and Paris6k. To simulate A_d situation, the two original datasets are rotated randomly within a range of $[[0^\circ, 360^\circ]]$. For each dataset, we impose three degrees of attack on it: Heavy attack ($k = 0.7$), medium attack ($k = 0.5$) and low attack ($k = 0.3$), that is, 70%, 50%, and 30% images in the dataset are randomly rotated, respectively. On the other hand, to simulate the A_q situation, the query images of two datasets are rotated by an angle within the range of $[0^\circ, 360^\circ]$, in the step of 10° (the same settings are used in the [27], [38]).

Retrieval-SfM-30k. During the training procedure, we employ the Retrieval-SfM-30k dataset that is provided by [35]. The dataset includes 28,599 images collected from Flickr, which are taken in popular 713 cities and landmarks across the world. This dataset includes 22,156 images, with 5,974 query images for training, and includes 6,403 images with 1,691 query images for validation.

4.2 Network Pre-Finetuning

Because of the particularity of the image retrieval task, a pre-trained network for classification tasks is not able to produce suitable local convolutional features for retrieval tasks. As the image retrieval task discriminates differences between two similar individual objects belonging to the same category, the local features must have more detail information than which is required for classification tasks. For this reason, it is necessary to fine-tune the network so that the model is well adapted to image retrieval tasks. In this work, to produce good deep features, we utilize a loss function called triplet loss. This loss function is confirmed to be more suitable for image retrieval than other often used loss function.

Based on the idea of linear margin nearest neighborhood (LMNN) [39], the triplet loss takes the influence of three inputs into account which are query image X_q , positive image X_p and negative image X_n . By optimizing the loss function, the network can make the distance between images with the same label smaller, while making the distance between images with different label further. This can be represented by Eq. (5):

$$\begin{aligned} L_T(W, X_q, X_p, X_n) \\ = \frac{1}{2} \max(0, m + \|X_q - X_p\|^2 - \|X_q - X_n\|^2). \end{aligned} \quad (5)$$

To implement the triplet loss, we rebuild the network with triplet structure that is also used in the [31]–[33]. The

Table 1 Details of pre-trained networks.

Network	Depth	Input size	LC features (D)
VGG16	5+3	$224 \times 224 \times 3$	512
ResNet50	5+1	$224 \times 224 \times 3$	2048
DenseNet121	5+1	$224 \times 224 \times 3$	1024

new network includes three branches with the same structure for three inputs, and all sub-networks sharing the same parameters as shown in Fig. 2.A.

In this work, we mainly focus on three widely used pre-trained networks which are VGG16, ResNet50 and Densenet121, as shown in the Table 1. All fine-tuning and feature extraction steps are implemented on the Matconvnet25, with GPU GTX 1080Ti and RTX 2080.

4.3 Encoding Method

After obtaining local convolutional features, instead of implementing maximum pooling directly, we employ the regional maximum activation of convolutions (R-MAC) [15] to generate the global features. Assumed that x_k represent the k^{th} feature map of 3D tensor f . Under the rectangular region $R \in [1, m] \times [1, n]$, the regional maximum activation is given by

$$F_R = [F_{R,1} \dots F_{R,k} \dots F_{R,D}], F_{R,i} = \text{MAX}_{p \in R} x_k(p) \quad (6)$$

All of the regional maximum features are implemented the sum and l_2 normalization at last. Following the encoding step, the principal component analysis (PCA) is used to decrease the dimension of the global features. Features from all of the network will be reduced to 512 dimensions in all of the experiments. We use the mean average precision (mAP) to evaluate the performance of each method.

5. Experimental Results and Discussion

5.1 In-Model Data Augmentation

In this section, we evaluate the effect of data augmentation in different cases and seek the best data augmentation method.

Some results of A_d situation are shown in the Table 2. We fine-tuned and evaluated the influence of k (the ratio of rotated images augmented) on the VGG16 network. The baseline model is the original fine-tuned VGG16 ($k = 0$). All the results show that the accuracy of the image retrieval decreases as the database attack is intensifying. Additionally, we can see that accuracy has significant improvement after data augmentation, and moreover, as the ratio k increases, the accuracy improvement is on an obvious increasing trend. It indicates that high ratio of rotated images performs well for advancing the rotation invariance of VGG16 network (we will talk about other models later). The highest ratio ($k = 0.7$) of rotated images shows the best performance on the A_d case, which increased the accuracy by 11.32% at most.

Table 2 Results of in-model data augmentation (DA) in the situation of database attack.

method	high attack	medium attack	low attack
k=0	33.12%	43.62%	58.79%
k=0.3	38.60%	49.89%	61.04%
k=0.5	42.32%	48.65%	61.81%
k=0.7	44.44%	49.80%	62.85%

Fig. 3 Results of in-model data augmentation in the situation of query attack. Four different types of lines correspond to the different values of k . The high ratio of rotated images in training data increase the rotation invariance of LC features.

The results of A_q case are given in Fig. 3. They show the similar tendency of rotation effect with baseline model. Effected by three inflection points (rotation angle at 90° , 180° , and 270°), curves present ‘w’ shape, where accuracy drop suddenly at 90° and 270° , but it returns around 180° . It is obvious that after data augmentation, the accuracy disparity among three inflection points decrease considerably. That means data augmentation mitigates the influence caused by rotation angle varying of a query image. The effect of k is the same as the A_d . With the increase of rotation parameter k , the rotation invariance of LC features is improved. When k achieved 0.7, the LC features showed the best performance on defying rotation.

We found in-model data augmentation advance the rotation invariance by decreasing the disparity of rotation angles. Furthermore, the high rotation ratio of input images advances the orientation variety of the training dataset, and makes it easy to obtain a model with multi-orientation information.

5.2 In-Model Feature Augmentation

For in-model feature augmentation strategy, since we found multiple RIP showed low accuracy on both A_q and A_d cases in the preliminary study, we only show and discuss the effect of a single RIP for VGG16 (other models will be discussed later). We evaluate the position of the RIP at first. There are five pooling layers (each in a convolutional block) in the network, and we evaluate the effect of RIP for each pooling layer. After fine-tuning the VGG16 network by the RIP, we

extract LC features and conduct the experiments for the A_q and A_d cases. Finally we evaluated the effect of three activation margins (0.3,0.5,0.7) on the rotation invariance of networks.

The results of in-model feature augmentation on the A_d are shown in the Table 3. Among the five candidate positions of RIP, only the last convolutional block (conv5) shows

Table 3 Results of in-model feature augmentation in the situation of database attack.

method	high attack	medium attack	low attack
original	33.12%	43.62%	58.79%
conv1+0.3	17.69%	17.93%	16.80%
conv1+0.5	28.68%	36.32%	45.26%
conv1+0.7	29.93%	29.92%	23.88%
conv2+0.3	22.37%	30.27%	40.93%
conv2+0.5	25.14%	32.86%	44.02%
conv2+0.7	27.76%	36.38%	49.37%
conv3+0.3	30.05%	38.10%	49.49%
conv3+0.5	30.28%	39.09%	51.19%
conv3+0.7	31.04%	39.66%	51.97%
conv4+0.3	29.53%	40.19%	52.89%
conv4+0.5	29.47%	38.74%	51.60%
conv4+0.7	27.08%	36.21%	51.94%
conv5+0.3	33.32%	44.75%	60.99%
conv5+0.5	33.45%	44.72%	59.87%
conv5+0.7	33.39%	44.72%	60.11%

Fig. 4 Results of in-model feature augmentation in the situation of query attack. Five colors represent the different position of RIP. Three marks indicate three different activation margins. The baseline is shown by the blue line without any mark. Among five convolutional blocks, the conv1 gives the best performance.

Fig. 5 The effects of two strategies (data augmentation and feature augmentation) with respect to ResNet50 and DenseNet121 are similar with VGG16. Especially, the LTC+DA method based on all the networks achieves the best robustness against rotations.

the slight improvement on rotation invariance. The results of the first three blocks show that with the margin increasing, the performance is nearly on a distinct going up trend, while the last two blocks show an disordered tendency. Figure 4 summarizes the result of RIP on the A_q . Compared with baseline model, the RIP shows little effect on rotation invariance except for the first block (conv1). The conv5 with margin 0.3 presents the best performance. In conclusion, the RIP improves the A_d case to a limited extent, but almost no contribution to the A_q case. For the A_d case, this is because the feature maps of conv5 is able to catch the global features of objects [43], rotating the feature maps of conv5 adds the orientation information to the model without destroying the spatial structure of the targets. On the other hand, for the A_q case, this is because the spatial structure of objects is modified by the RIP, which leads to a low retrieval performance. The RIP emphasizes rotation information too much so that it ignores important features of original images.

As summarized above, to a greater or lesser extent, two in-model strategies both show the validity to advance the rotation invariance of networks (VGG16). Compared with

Table 4 Transformation combinations for image retrieval.

index	setting
comb1	[0,90]
comb2	[0,90,180,270]
comb3	[0,90,180,270,flip, flip+90]
comb4	[0,90,180,270,flip, flip+90, flip+180, flip+270]

Table 5 Results of post-model feature augmentation in the situation of database attack.

method	heavy attack	medium attack	low attack
original	33.12%	43.62%	58.79%
comb1+CAT	35.45%	44.97%	59.72%
comb1+max	44.96%	50.04%	58.02%
comb1+mean	48.64%	52.83%	58.93%
comb2+CAT	36.99%	46.19%	61.19%
comb2+max	54.19%	56.84%	60.38%
comb2+mean	56.95%	58.54%	60.86%
comb3+CAT	39.65%	47.34%	61.03%
comb3+max	54.48%	57.18%	60.59%
comb3+mean	56.9%	58.8%	60.73%
comb4+CAT	38.7%	47.48%	61.76%
comb4+max	54.03%	57.11%	60.49%
comb4+mean	57.52%	59.95%	62.25%

of the datasets. Compared with the baseline, it improved accuracy by 22% at most.

Figure 6 shows a comparison of the effects of LTC features in A_q situation. Compared with the baseline, all the combinations of LTC show better performance on defying the rotations. Except concatenation, LTC features enhance

Oxford5k_Aq

mAP

Rotation Angle

Legend:

- comb1+CAV
- comb2+CAV
- comb3+CAV
- comb4+CAV
- comb1+MAA
- comb2+MAA
- comb3+MAA
- comb4+MAA
- comb1+Mean
- comb2+Mean
- comb3+Mean
- comb4+Mean
- Rotation

Fig. 6 Results of post-model feature augmentation in the situation of query attack. Within different combinations, the comb4 with average pooling achieved the best performance.

Fig. 7 Visualization of three strategies and baseline for two attack situations. The first row of each block shows the query image and the average precision of each query. The histograms represent the AP of each method, which is illustrated in the first column of the figure (the same color of the bar represents the same method). For each method, the top 10 results are shown in the block. The red and green boxes represent the wrong and right of the resulting images.

Table 6 Evaluate the combination of the best performing strategies under other networks.

Network	Oxford5k			Paris6k		
	heavy attack	medium attack	low attack	heavy attack	medium attack	low attack
VGG16	33.12%	43.62%	58.79%	62.82%	67.37%	72.12%
VGG16-LTC	57.52%	59.95%	62.25%	75.10%	76.32%	76.99%
VGG16-DA	44.44%	49.80%	62.85%	68.97%	71.50%	74.54%
VGG16-DA-LTC	64.40%	64.45 %	66.52 %	75.87%	76.64%	77.66%
ResNet50	36.20%	45.74%	60.43%	67.41%	71.72%	75.60%
ResNet50-LTC	65.79%	67.11%	70.34%	79.98%	80.66%	81.60%
ResNet50-DA	60.40%	58.31%	65.60%	74.10%	75.43%	76.53%
ResNet50-DA-LTC	64.98%	66.37 %	68.08 %	76.68%	77.43%	77.71%
DenseNet	36.20%	40.20%	54.60%	57.82%	63.31%	68.81%
DenseNet-LTC	55.97%	63.94%	65.53%	77.38%	78.2%	79.45%
DenseNet-DA	40.14%	43.63%	52.48%	62.86%	65.55%	69.66%
DenseNet-DA-LTC	60.48%	61.12 %	62.66 %	74.35%	74.88%	75.83%

the accuracy greatly at the typical angles which are involved in transformation combination. Therefore, it is not difficult to find that the curves become more flatter (rotation invariance) when feature maps include more orientation information. In this case, same as A_d situation, comb4 with average pooling shows the best performance.

It is easy to see that the concatenation fusion shows disappointing performance in all the experiments. We think this is because of too much repetitive information in LTC features. Since concatenation just simply connects all the feature maps from the beginning to the end, the important activations do not get the attention they deserve to do. Although PCA is implemented to decrease the repetition, it decreases the amount of rotation information simultaneously. On the other hand, max pooling and average pooling have comparable performance in some of experiments. It is because both of them picked up the important activations of feature maps.

5.4 Experiments with Other Networks

Based on the experiments we implemented previously, we select the best performing strategies, namely, in-model data augmentation and post-model feature augmentation, which are related to training and test procedures, respectively. Pursuant to the two strategies, we further conducted thorough experiments to evaluate the effects of previous works over ResNet50 and DenseNet121 under two attack cases, A_d and A_q .

Moreover, we visualized the results of three methods of VGG16 as shown in Fig. 7. For the A_q situation, effected by rotation angle 90° , the top 10 retrieval results of the baseline model are almost wrong. It is easy to see that the method including LTC features showed robustness against rotations. The A_d situation is given at the right side of Fig. 7. From the retrieval results, we can see that rotated positive images come to the top, which is the situation we are aiming for.

In summary, for the VGG16, the combination of in-model data augmentation (DA) and post-model feature augmentation (LTC) achieved the best performance of rotation invariance, as expected. For deeper network, like ResNet and DenseNet, although the LTC features show the highest accuracy under two attack cases, the combination of LTC and DA shows the least fluctuation.

6. Conclusion

In this work, we provide a thorough and systematic evaluation for advancing rotation invariance of LC features. According to the results, the following conclusions are obtained: 1) in-model data augmentation (DA) advances the rotation invariance because it decreases the accuracy disparity between different orientations, and 2) post-model feature augmentation (LTC) defies the rotation attack by advancing accuracy of appointed orientations. Based on our experiments, the best robust model against rotation is the combination of in-model data augmentation (DA: rotation pa-

rameter $k = 7$) and post-model feature augmentation (LTC: 0,90,180,270,flip, flip+90, flip+180, flip+270).

Acknowledgments

This work is supported by the JSPS Grant-in-Aid for Early Career Scientists (No. 20K19831) and JSPS Grant-in-Aid for Challenging Research (Pioneering) (No. 20K20399).

References

- [1] X.-Y. Wang, B.-B. Zhang, and H.-Y. Yang, "Content-based image retrieval by integrating color and texture features," *Multimedia tools and applications*, vol.68, no.3, pp.545–569, 2014.
- [2] J.-M. Guo, H. Prasetyo, and J.-H. Chen, "Content-based image retrieval using error diffusion block truncation coding features," *IEEE Transactions on Circuits and Systems for Video Technology*, vol.25, no.3, pp.466–481, 2014.
- [3] X. Duanmu, "Image retrieval using color moment invariant," 2010 Seventh International Conference on Information Technology: New Generations, pp.200–203, IEEE, 2010.
- [4] H. Shao, Y. Wu, W. Cui, and J. Zhang, "Image retrieval based on mpeg-7 dominant color descriptor," 2008 The 9th International Conference for Young Computer Scientists, pp.753–757, IEEE, 2008.
- [5] M.M. Islam, D. Zhang, and G. Lu, "Automatic categorization of image regions using dominant color based vector quantization," 2008 Digital Image Computing: Techniques and Applications, pp.191–198, IEEE, 2008.
- [6] D. ping Tian et al., "A review on image feature extraction and representation techniques," *International Journal of Multimedia and Ubiquitous Engineering*, vol.8, no.4, pp.385–396, 2013.
- [7] G.A. Papakostas, D.E. Koulouriotis, and V.D. Tourassis, "Feature extraction based on wavelet moments and moment invariants in machine vision systems," *Human-Centric Machine Vision*, p.31, 2012.
- [8] Z. Mehmood, S.M. Anwar, N. Ali, H.A. Habib, and M. Rashid, "A novel image retrieval based on a combination of local and global histograms of visual words," *Mathematical Problems in Engineering*, vol.2016, 2016.
- [9] B. Zafar, R. Ashraf, N. Ali, M. Ahmed, S. Jabbar, and S.A. Chatzichristofis, "Image classification by addition of spatial information based on histograms of orthogonal vectors," *PloS one*, vol.13, no.6, 2018.
- [10] N. Ali, K.B. Bajwa, R. Sablatnig, and Z. Mehmood, "Image retrieval by addition of spatial information based on histograms of triangular regions," *Computers & Electrical Engineering*, vol.54, pp.539–550, 2016.
- [11] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, ACM New York, NY, USA, vol.60, no.6, pp.84–90, 2017.
- [12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [13] A.B. Babenko and V. Lempitsky, "Aggregating local deep features for image retrieval," *The IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015.
- [14] H. Azizpour, A.S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, "From generic to specific deep representations for visual recognition," *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp.36–45, 2015.
- [15] G. Toliás, R. Siciu, and H. Jégou, "Particular object retrieval with integral max-pooling of cnn activations," *arXiv preprint arXiv:1511.05879*, 2015.
- [16] D.G. Lowe, "Object recognition from local scale-invariant features," *The Proceedings of the Seventh IEEE International Conference on Computer vision*, pp.1150–1157, IEEE, 1999.

- [17] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, pp.886–893, IEEE, 2005.
- [18] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," Proceedings Ninth IEEE International Conference on Computer Vision, 2003.
- [19] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp.1794–1801, IEEE, 2009.
- [20] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," Computer Vision–ECCV 2010, vol.6314, pp.143–156, 2010.
- [21] H. Jegou, F. Perronnin, M. Douze, J. Sánchez, P. Perez, and C. Schmid, "Aggregating local image descriptors into compact codes," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.34, no.9, pp.1704–1716, 2012.
- [22] Z. Cai, W. Gao, Z. Yu, J. Huang, and Z. Cai, "Feature extraction with triplet convolutional neural network for content-based image retrieval," 2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA), pp.337–342, IEEE, 2017.
- [23] F. Yan, K. Mikolajczyk, and J. Kittler, "Multiple kernel learning via distance metric learning for interactive image retrieval," International Workshop on Multiple Classifier Systems, vol.6713, pp.147–156, Springer, 2011.
- [24] J. Zhou, X. Liu, W. Liu, and J. Gan, "Image retrieval based on effective feature extraction and diffusion process," Multimedia Tools and Applications, vol.78, no.5, pp.6163–6190, 2019.
- [25] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," European conference on computer vision, vol.8695, pp.392–407, Springer, 2014.
- [26] A. Sharif Razavian, J. Sullivan, A. Maki, and S. Carlsson, "A baseline for visual instance retrieval with deep convolutional networks," International Conference on Learning Representations, May 7–9, 2015, San Diego, CA, ICLR, 2015.
- [27] V. Chandrasekhar, J. Lin, O. Morère, H. Goh, and A. Veillard, "A practical guide to cnns and fisher vectors for image instance retrieval," Signal Processing, vol.128, pp.426–439, 2016.
- [28] D. Marcos, M. Volpi, N. Komodakis, and D. Tuia, "Rotation equivariant vector field networks," Proceedings of the IEEE International Conference on Computer Vision, pp.5048–5057, 2017.
- [29] L. Sifre and S. Mallat, "Rotation, scaling and deformation invariant scattering for texture discrimination," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.1233–1240, 2013.
- [30] G. Cheng, P. Zhou, and J. Han, "Rifd-cnn: Rotation-invariant and fisher discriminative convolutional neural networks for object detection," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.2884–2893, 2016.
- [31] M. Jaderberg, K. Simonyan, A. Zisserman, et al., "Spatial transformer networks," Advances in neural information processing systems, pp.2017–2025, 2015.
- [32] P.Y. Simard, D. Steinkraus, and J.C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," ICDAR, 2003.
- [33] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," IEEE Conference on Computer Vision and Pattern Recognition, CVPR'07, pp.1–8, IEEE, 2007.
- [34] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2008.
- [35] F. Radenović, G. Tolias, and O. Chum, "Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples," European Conference on Computer Vision, vol.9905, pp.3–20, Springer, 2016.
- [36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Proceedings of the IEEE conference on computer vision and pattern recognition, pp.770–778, 2016.
- [37] G. Huang, Z. Liu, L. Van Der Maaten, and K.Q. Weinberger, "Densely connected convolutional networks," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.4700–4708, 2017.
- [38] G. Takacs, V. Chandrasekhar, S. Tsai, D. Chen, R. Grzeszczuk, and B. Girod, "Unified real-time tracking and recognition with rotation-invariant fast features," 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp.934–941, IEEE, 2010.
- [39] K.Q. Weinberger, J. Blitzer, and L.K. Saul, "Distance metric learning for large margin nearest neighbor classification," Advances in neural information processing systems, pp.1473–1480, 2006.
- [40] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning fine-grained image similarity with deep ranking," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.1386–1393, 2014.
- [41] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," International Workshop on Similarity-Based Pattern Recognition, pp.84–92, Springer, 2015.
- [42] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.815–823, 2015.
- [43] M.D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," European conference on computer vision, vol.8689, pp.818–833, Springer, 2014.

Longjiao Zhao received the M.S. degree in Information Science, from Nagoya University in 2018. She is currently a Ph.D. Candidate with the Graduate School of Informatics, Nagoya University.

Yu Wang received the M.S. degree in Information Science and Ph.D. degree in Engineering, from Nagoya University, in 2010 and 2013, respectively. He is currently an assistant professor with the Graduate School of International Development, Nagoya University.

Jien Kato received the M.E. and Ph.D. degrees in Information Engineering from Nagoya University in 1990 and 1993, respectively. Then she became an assistant professor at Toyama University. She was a visiting researcher at the University of Oxford from 1999 for one year. She became an associate professor at the Graduate School of Engineering of Nagoya University in 2000. She has been a professor at the College of Information Science and Engineering of Ritsumeikan University since 2018. Her

research interests include object recognition, visual event recognition and machine learning. She is a member of IEICE, IPSJ and JSAI, and also a senior member of IEEE.