Feature extraction based on spatial co-occurrence and rotation properties for image recognition

Feature extraction based on spatial co-occurrence and rotation properties for image recognition

March 2019

Ryusuke Nosaka

Feature extraction based on spatial co-occurrence and rotation properties for image recognition

Graduate School of Systems and Information Engineering
University of Tsukuba

March 2019

Ryusuke Nosaka

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Feature extraction for image recognition

Image recognition is an essential technology in many applications such as security system, robot vision and medical diagnosis. Its importance is increasing due to the popularization of camera devices such as smartphones and network cameras year by year. Figure 1.1 shows the outline of a general process of image recognition. The process consists of the following steps. First, an object image is captured by a camera device. Next, in the feature extraction step, an image feature is extracted from the input image. The image feature represents characteristics of the object, such as shape and texture compactly. Finally, the extracted image features are fed to a classifier or a regressor to output necessary results (*e.g.*, categories, attributes, and shapes) according to a target task. To achieve high recognition performance, it is important to consider how to design and extract valid image features from an input image.

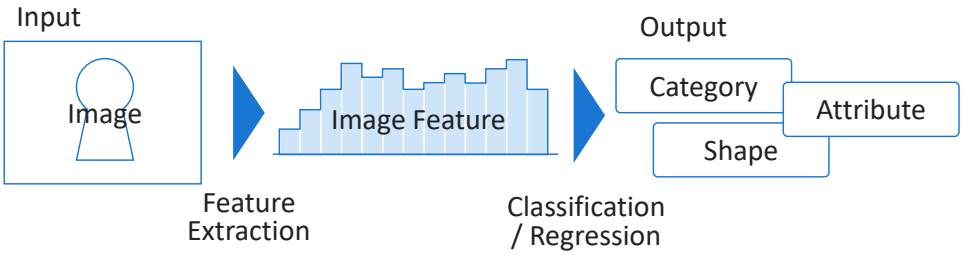
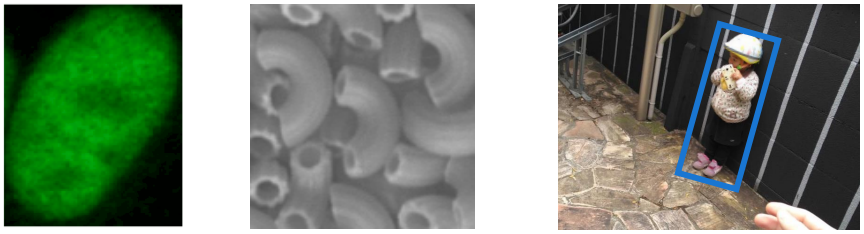


Figure 1.1: Process flow of image recognition.



(a) (b) (c)

Figure 1.2: Samples of object images that can rotate in any direction: (a) cell image in the MIVIA HEp-2 images dataset [1], (b) texture image in the Outex database [2], (c) wearable cam image with bounding box.

1.2 Basic idea of our feature extraction

To consider valid image feature extraction, in this study, we tackle the rotation of an object, which is a general phenomenon in many image recognition tasks. For example, cell and texture images can be in any orientation as shown in Figures 1.2a and b. In the object detection, objects in an input image can rotate freely depending on the relationship between the object and a camera as shown in Figure 1.2c.

For considering the rotation, we study the feature extraction from perspectives of "rotation invariant features" and "rotation variant features". Rotation invariant feature is a type of image feature that is invariant to rotation of an input image. As this type of image feature can suppress variation of the input due to the rotation, it is suitable for tasks that classify rotation invariant label of objects such as category, color names, and gender. On the other hand, the rotation variant feature is a type of image feature that is variant to rotation of an input image. This type of image feature is suitable for estimating the object's information that depends on the orientation of the object, such as posture and bounding box. Moreover, when both types of feature exist at the same time in an input image, we need to switch both image features properly according to input and situation. Therefore, in this study, we tackle the feature extraction for these three cases, (1) the rotation invariant feature for category classification, (2) the rotation variant feature for bounding box estimation, and (3) switching both image features and we propose the following four novel methods.

1.2.1 Rotation invariant feature for category classification

The study for the rotation invariant feature in the classification tasks is conducted based on local binary patterns (LBP) [3] among the conventional handcrafted fea-

tures [4, 3, 5, 6, 7, 8, 9]. This feature has many advantages *e.g.*, robustness against illumination change and low computational cost, and has been widely used in many applications such as face recognition [10], facial expression recognition [11], and texture recognition [12]. LBP represents local regions by binary patterns based on relative intensity magnitude of the local regions, and a set of binary patterns of the local regions is represented as a histogram. As rotation invariant features based on LBP, there are LBP^{ri} [13], LBP^{riu} [14], and LBP-HF [15]. In these features, rotation invariance is considered on the local region. Also, most of studies on extensions of LBP discussed how to effectively describe local regions for image recognition [16, 17, 18, 19, 20]. They ignore the spatial relationships among the local regions, causing the descriptive power them to limited.

To consider the spatial relationship among LBPs, we firstly incorporate the concept of *spacial co-occurrence* to LBP histogram. Spatial co-occurrence is often used to extract additional information related to global structures in various local region-based features [21, 22, 23]. Motivated by these methods, we introduce LBP pair histogram calculated from pairs of two LBPs. We call this feature *Co-occurrence among Adjacent LBPs* (CoALBP). CoALBP has high descriptive ability while inheriting the advantages of LBP. And then, to realize the rotation invariance, we incorporate the concept of rotation equivalence class into CoALBP. This enhanced feature is called *Rotation Invariant Co-occurrence among adjacent LBPs* (RIC-LBP). This method can provide high descriptive ability and invariance to image rotation simultaneously.

1.2.2 Rotation variant feature for bounding box estimation

In object detection, a bounding box is used to represent the region of an object roughly as shown in Figure 1.2c. When an object can rotate freely, the corresponded bounding box should rotate according to the orientation of the object. CNN-based object detector estimates the bounding box using a regressor that outputs position and shape of the bounding box [24, 25, 26, 27, 28]. To estimate a accurate bounding box using a regressor, we need to consider not only a variant feature but also how to design a regressor for the rotation. This is because the standard regressors are designed without this consideration for rotation and a bounding box estimated by the standard regressors cannot follow the orientation of the object precisely. In this study, we propose a novel network architecture, *Orientation-Aware Regression* (OAR), which deals with the orientation of the object using 2D-vector representation. This method can significantly improve the

estimation accuracy of the bounding box and the detection performance.

1.2.3 Switching rotation invariant feature and rotation variant feature

Rotation invariant features should be used when an object can rotates freely and its orientation is unconstrained. However, there are recognition tasks in which, the constraint of the orientation cannot be known in advance. For example, on object category classification with a fixed camera, a fork held by hand can rotate freely, but cars are constrained on the ground by gravity. We call whether the orientation of an input object is constrained "rotatability of the object." In this study, we propose a network module, *Rotation Invariance Switcher* (RIS), which estimates the rotatability of an input object and switches between the rotation invariant features or the rotation variant features adaptively. We applied RIS to object classification tasks and confirmed that RIS could achieve high recognition performance by observing the rotatability of the input .

1.3 Contributions

The main contributions of this study are as follow:

- We tackle the feature extraction for rotation comprehensively in these three cases, (1) the rotation invariant feature for category classification, (2) the rotation variant feature for bounding box estimation, and (3) switching both image features.
- To enhance the conventional LBP based feature, we introduce the concept of co-occurrence to the LBP histogram and propose CoALBP, which has high descriptive ability while inheriting the advantages of LBP.
- We incorporate the concept of rotation equivalence class into CoALBP and propose RIC-LBP, which can active high descriptive ability and rotation invariance simultaneously.
- For estimating rotation bounding boxes, we propose OAR, in which the object's orientation is propagated to the output bounding box in 2D-vector representation, improving the estimation accuracy of the bounding box and the detection performance.

- We propose RIS, which adaptively switches whether to use rotation invariant features or not according to an input image. We confirm this method improves classification accuracy of object classification.

1.4 Thesis organization

The rest of this thesis is organized as follows.

In Chapter 2, we review LBP and describe the basic idea and the algorithm of CoALBP and provide interpretations from the perspective of spatial patterns. In Chapter 3, we present RIC-LBP, in which the concept of rotation invariance is introduced to CoALBP described in the previous chapter. In Chapter 4, we explain the property of bounding box on rotation and propose a novel bounding box regression, OAR, with a deep neural network approach. In Chapter 5, we explain rotatability and present a novel switching modules for deep neural networks, RIS. In the final chapter, we describe conclusions of this study and future works.

Chapter 2

Co-occurrence among Adjacent LBPs

In this chapter, we provide a review of LBP and show the effectiveness of introducing spatial co-occurrence to LBP. We firstly show the algorithm and the weakness of LBP. Next, we describe the idea and the algorithm of CoALBP and show its interpretation. Then we show experimental results on face recognition task and texture classification task. Additionally, we show the performance of CoALBP in the competition of HEp-2 cells classification contest 2012 in the final of this chapter.

2.1 Local binary patterns

Local binary pattern (LBP) [3] is a binary pattern that describes a local region. LBP is obtained by thresholding the difference between a center pixel and its neighboring pixels in a local region, as shown in Figure 2.1. LBP is invariant to uniform changes of image intensity over an entire image since the binary pattern in LBP is made from the relative differences. This means that LBP is robust against the changes in illumination, which is a difficult problem in face and texture images. Besides, the process for obtaining LBPs is simple, and its computational cost is low.

In the following, we describe how to obtain LBP. Let I be an image intensity and $\boldsymbol{r} = (x, y)$ be a coordinate vector in the image. Then, an LBP at $\boldsymbol{r}$ is defined

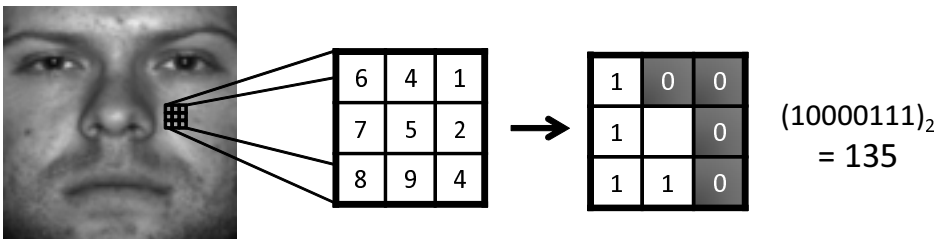


Figure 2.1: Process flow to obtain an LBP from a local region. In this example, the intensity of the center pixel is 5 and those of its neighboring pixels are 6, 4, 1, 2, 4, 9, 8 and 7. Thus, the binary pattern is "100000111" and $LBP(\boldsymbol{r}) = 135$.

as follows:

$$LBP(\boldsymbol{r}) = \sum_{i=0}^{N-1} sgn(I(\boldsymbol{r} + \Delta \boldsymbol{s}_i) - I(\boldsymbol{r}))2^i, \tag{2.1}$$

$$sgn(x) = \begin{cases} 1, & \text{if } x \geq 0, \\ 0, & \text{otherwise}, \end{cases} \tag{2.2}$$

where N is the number of neighbor pixels. $\Delta \boldsymbol{s}_i$ is displacement vector from the center pixel to neighboring pixels given by $\Delta \boldsymbol{s}_i = (s\cos(\theta_i), s\sin(\theta_i))$, where $\theta_i = \frac{2\pi}{N} i$ and s is a scale parameter of LBP. While the displacement vector is defined on a circle, for simplicity, the definition on a grid can be used, as shown in Figure 2.1. The LBP calculated for each coordinate is typically summarized into a histogram as an image feature. Note that, N is usually set to 8 and the histogram of LBPs is a $256(= 2^N)$ dimensional feature vector.

Based on LBP, many extended methods have been proposed so far [16, 17, 18, 19, 20]. Most of them are based on how to describe local regions for image recognition effectively. For example, LTP [16] encodes the magnitude relation of the local region into a ternary pattern instead of a binary pattern to represent the local region in more detail. LPQ [17] encodes the local regions by quantizing Fourier coefficients of the local regions for robustness against blur. DLBP [18] uses the Fisher separability criterion to optimize a set of pixels compared to the center pixel as training.

Recently, many studies in image recognition use CNN feature instead of handcrafted feature. Nevertheless handcrafted feature is still useful, since the performance of CNN can be further improved by combining the handcrafted and CNN features [29, 30, 31, 32, 33, 34]. Moreover, CNN feature has several problem, for example the performance of CNN depends largely on the size of training data and it is difficult to see the internal mechanism of the network architecture. To address

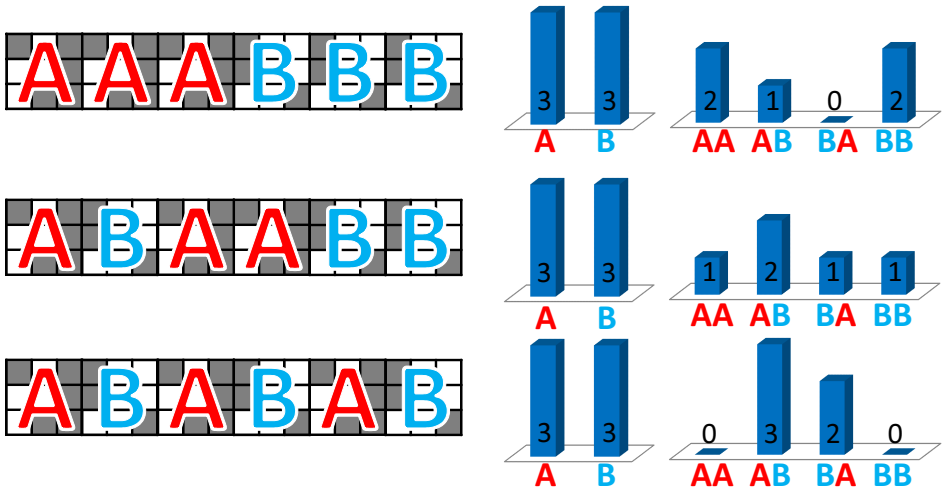


Figure 2.2: Difference of LBP histogram and CoALBP histogram. (a) Example images. (b) LBP histograms. (c) CoALBP histogram.

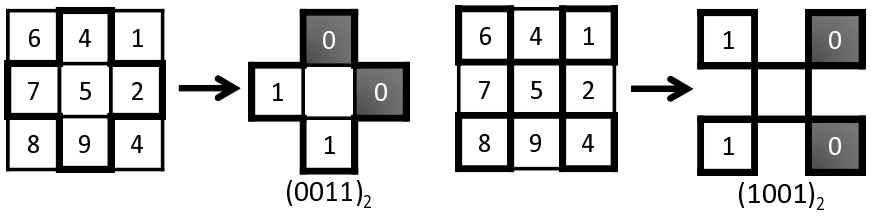
these issues, extracting handcrafted features is still cherished.

As above, LBP histogram and the extensions have many advantages. However, the LBPs in an image are forcedly packed into a single histogram. The extensions of LBP also have a similar problem. The LBP histogram generation process causes much loss of the spatial structure information of an image, especially spatial relations among the LBP. This suggests that there is still a room for further improving the performance of LBP-based features.

2.2 Co-occurrence of adjacent LBPs

To consider the spatial relationship among LBPs, we introduce the concept of *co-occurrence* to the LBP histogram. Co-occurrence is often used to extract information related to structures in various local region-based features such as Co-HOG [21], GLAC [22] and Joint Haar-like features [23]. Motivated by these methods, we introduce LBP pair histogram calculated from two considered LBPs. We call this feature, *Co-occurrence among Adjacent LBPs* (CoALBP). CoALBP is a kind of extension of the original LBP in that it consists of both the original LBP and the co-occurrence of LBPs. CoALBP has a high descriptive ability while inheriting the advantages of LBP.

We explain the effectiveness of using the co-occurrence using Figure 2.2, show-



(a) Vertical and horizontal cross (+) (b) Diagonal cross (×)

Figure 2.3: Configurations of neighboring pixels of an LBP in CoALBP.

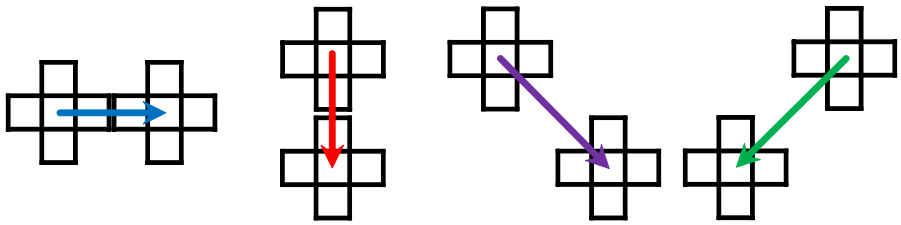


Figure 2.4: Displacements of LBP pair

ing the difference between an LBP histogram and the CoALBP histogram. The three image examples are composed of three LBP A and three LBP B, as shown in Figure 2.2a. Since the number of LBP A and LBP B in each image are the same, the LBP histograms are generated from the three images coincide with each other, as shown in Figure 2.2b. In contrast, the CoALBP histograms are generated adjacent LBP pairs and are entirely different from each other, as shown in Figure 2.2c. From this simple example, you can see that the descriptive ability of the original LBP is insufficient, and the spatial co-occurrence of LBPs is a valid requirement for realizing a higher descriptive ability.

2.2.1 Implementation of CoALBP histogram

An LBP pair at $\boldsymbol{r}$ is defined as follows:

$$P(\boldsymbol{r}, \Delta\boldsymbol{r}) = (LBP(\boldsymbol{r}), LBP(\boldsymbol{r} + \Delta\boldsymbol{r})), \tag{2.3}$$

where $\Delta\boldsymbol{r}$ is a displacement vector between an LBP pair. The LBP pair calculated for each coordinate is summarized into histograms with several displacement vectors.

The kinds of LBPs and displacement of LBP pairs can be on the various setting,

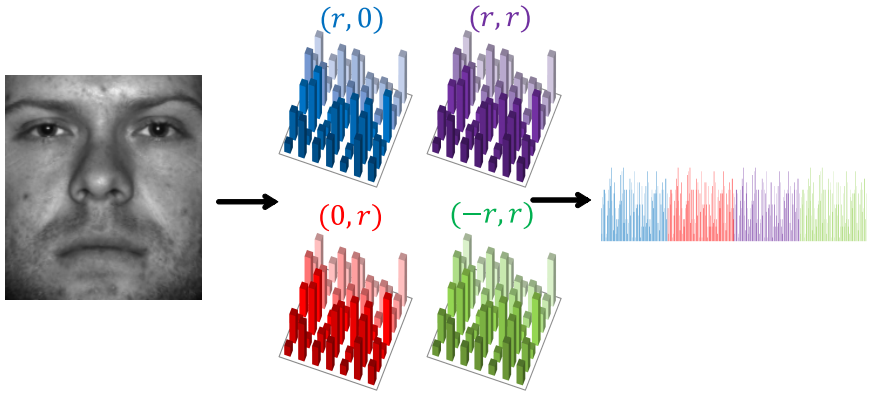


Figure 2.5: Process flow to obtain the CoALBP histogram: (a) Extracting LBPs from an image, (b) LBP pair histograms with four displacement vectors $\Delta \boldsymbol{r}$, (c) Combined histogram as feature vector.

increasing the computational cost. We give them the following several constraints in order to limit the computational cost. First, the number of neighboring pixels in LBP is set to four, *i.e.*, $N = 4$. Thus, the constrained LBP has $16 (= N_P = 2^N)$ possible patterns. Next, as a configuration of neighbor pixels, there are two choices for configuration. One is the vertical and horizontal cross configuration, which consists of only two horizontal and two vertical pixels, as shown in Figure 2.3a. In this configuration, $\Delta \boldsymbol{s}_i = (\pm s, 0), (0, \pm s)$. Another one is the diagonal cross configuration, which consists of four diagonal pixels as shown in Figure 2.3b. In this configuration, $\Delta \boldsymbol{s}_i = (\pm s, \pm s), (\pm s, \mp s)$. As a displacement of LBP pair $\Delta \boldsymbol{r}$, four orientations (*i.e.*, horizontal, vertical and two diagonal) are used. Here, $\Delta \boldsymbol{r}$ $= (r, 0), (0, r), (r, r), (-r, r)$, the value of r is an interval between an LBP pair. Figure 2.4 shows the displacements of LBP pair.

Since an LBP pair histogram has $256 (= {N_P}^2)$ dimensions and four histograms are extracted from an image, the CoALBP histogram has $1024 (= 4 \times {N_P}^2)$ dimensions. Although CoALBP histogram has high dimensionality, the computational time has a linear relationship with the number of pixels of an image. Besides, since a CoALBP is represented by an LBP pair, the CoALBP retains information of a single LBP, making it an extension of the original LBP.

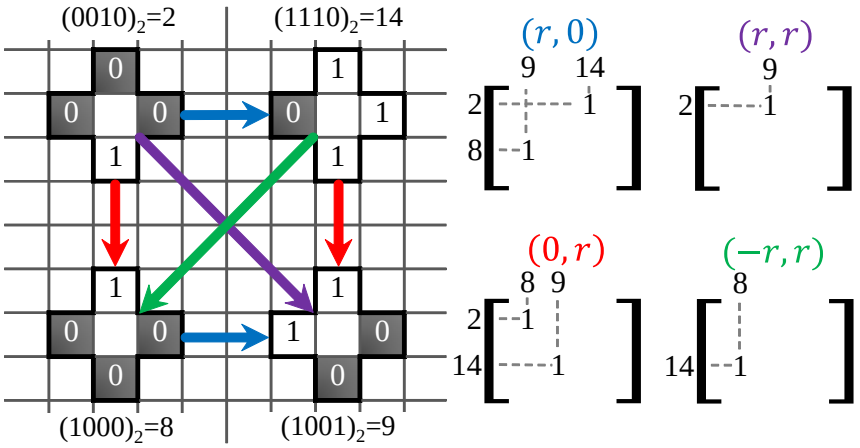


Figure 2.6: An example of obtaining the CoALBP histogram. (a) Example image. (b) LBP pair histograms $H(\Delta \boldsymbol{r})$

2.2.2 Process flow of obtaining CoALBP histogram

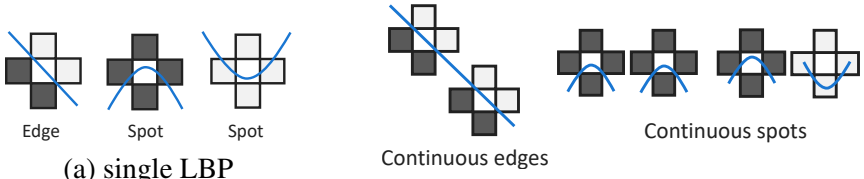
The process to obtain a CoALBP histogram consists of three steps, as shown in Figure 2.5. Firstly, LBPs are extracted at every pixel position $\boldsymbol{r}$ in the input image (Figure 2.5a). Next, a LBP pair histogram is calculated for each displacement $\Delta \boldsymbol{r}$ (Figure 2.5b). Finally, these histograms are combined to a $4N_P^2$-dimensional feature vector (Figure 2.5c).

A concrete example of extracting CoALBP histogram from an image is shown in Figure 2.6. The example image has four LBPs (Figure 2.6a). The labels of these LBPs are $2, 8, 9$ and 14, respectively. In the case of the displacement vector $\Delta \boldsymbol{r} = (r, 0)$, there are two LBP pairs ({upper left, upper right} and {lower left, lower right}) in the image. Since the labels of these pairs are $(2, 14)$ and $(8, 9)$, the frequency of them are 1 and others are 0. For other displacement vectors: $\boldsymbol{r} = (0, r), (r, r)$ and $(-r, r)$, an LBP pair histogram is similarly generated as shown in Figure 2.6b.

2.2.3 Interpretation

In this section, we explain the interpretation of CoALBP.

A single LBP is made from locally relative magnitude relations. Thus, the relative relations (*i.e.*, binary patterns) could be classified to edge and spot as shown in Figure 2.7a. An LBP pair includes not only two binary patterns but also a spa-



(a) single LBP (b) LBP pair

Figure 2.7: Interpretation of LBP and CoALBP from a perspective of represented spatial patterns.

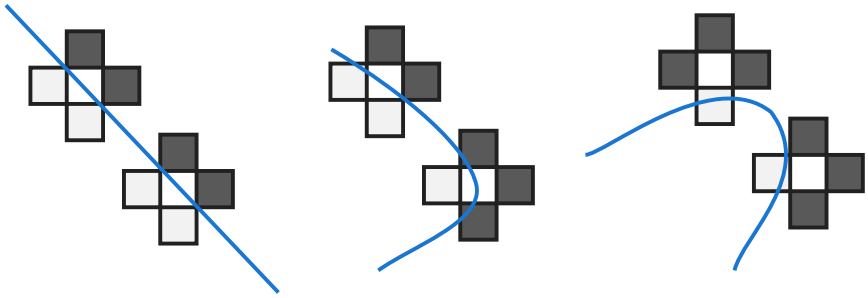


Figure 2.8: Interpretation of CoALBP from the perspective of curvature.

cial relationship between two binary patterns. Thus, it can be considered that the CoALBP represents spacial continuous edges and continuous spacial spots as shown in Figure 2.7b.

Moreover, since edges represented by binary pattern has orientation, CoALBP can also represent various curvatures as shown in Figure 2.8. Also, since CoALBP has several orientations as displacement, the curvatures are extracted in several orientations. It is hard for a widely expanded LBP to represent these patterns because the widely expanded LBP can represent only simple patterns such as edge, spot, etc.

These interpretations would help to understand the advantage of CoALBP.

2.3 Experimental evaluation

To evaluate CoALBP, face recognition in two setting (recognition under various illumination condition and image set based recognition), and texture classification without rotation variance are conducted.

Figure 2.9: Example images in the Extended Yale Face Database B.

2.3.1 Face recognition under various illumination conditions

Settings

In this experiment, we evaluated CoALBP through face recognition under various illumination conditions using the Extended Yale Face Database B [35]. Figure 2.9 shows examples of images included in the dataset. The dataset contains the faces of 38 subjects under 64 variations in illumination for each. All images are frontal face images cropped to 168×168 pixels. In this experiment, the images are resized to 88×88 pixels that are determined experimentally. Images with frontal lighting were used as a training set (one image per person). The remaining images were used as a testing set (63 images per person).

CoALBP histogram was compared with a raw image feature, a Gabor image feature [36], and the LBP histogram [3]. The raw image feature is obtained by vectorizing an input image. For the LBP histogram, we prepared three types of features, differing in the selection of surrounding pixels: eight neighboring pixels (original LBP), pixels in the vertical and horizontal cross configuration ($+$) and pixels in the diagonal cross configuration ($\times$). For CoALBP, we prepared two types of features, differing in the selection of surrounding pixels: pixels in the vertical and horizontal cross configuration (+) and pixels in the diagonal cross configuration ($\times$). The image was divided into multiple subregions. Four types of divisions (1×1, 2×2, 4×4, and 8×8) were performed. The features extracted from these subregions were integrated into a final feature. For each division, the dimension of the final proposed feature is $4N_P^2 \times 1$, $4N_P^2 \times 2^2$, $4N_P^2 \times 4^2$, $4N_P^2 \times 8^2$, respectively.

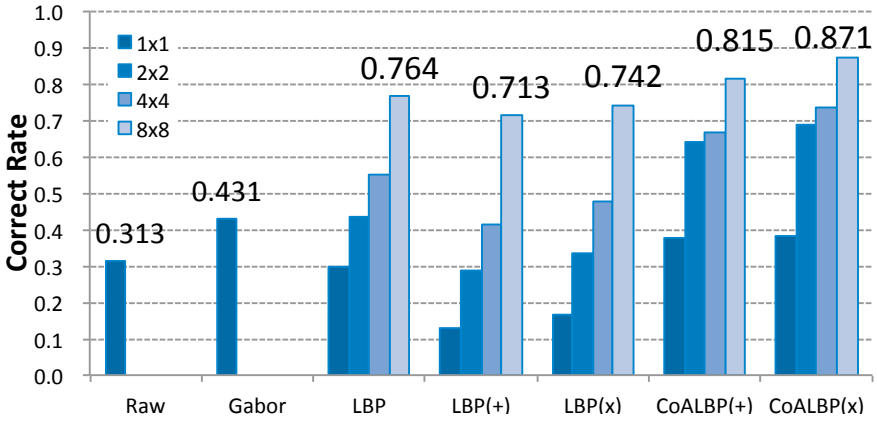


Figure 2.10: Results of face recognition under various illumination conditions.

The region division described above was not performed for the raw image feature and the Gabor image feature.

As a classifier, the nearest neighbor method with L_1 distance was used. The L_1 distance is usually used as the similarity between two histograms [37], since it has a similar characteristic to the histogram intersection defined by $S(\boldsymbol{h}_1, \boldsymbol{h}_2) = \sum_i \min(h_{1i}, h_{2i})$.

Results

Figure 2.10 shows the results of each feature. From these results, it can be confirmed that the LBP-based feature's robustness against variations in illumination, while performances of the raw image feature and the Gabor image feature were poor, due to their sensitivity to illumination.

Also, the results show that the CoALBP outperforms other features. It has achieved the best performance using parameters $s = 1$, $r = 3$, and 8×8 division. We can consider that increasing number of division increases the performance due to keeping spacial information. Moreover, we can also see that the performance of CoALBP depends on the configuration of LBP. This result implies that there is room to improve the performance by optimizing the configuration.

2.3.2 Face recognition based on image sets

Settings

In this experiment, we evaluated the effectiveness of CoALBP for the image set based face recognition.

As the dataset, face images of 109 subjects that are collected by us were used. Figure 2.11 shows some examples from the dataset. The dataset contains two types of movement; horizontal head shaking as shown in Figure 2.11a, and vertical head shaking as shown in Figure 2.11b. Each subject was taken 150 photos twice with each movement. Thus, each subject has 600 images (= 150 $\times$ 2 sets $\times$ 2 movements). The image size is 128×128 pixels. The face region was obtained by the face detection module of OpenCV [38].

Each set was split to 10 subsets, each of which has 15 images. The subset for training and the subset for testing were chosen from sets that are different from each other. Thus, the number of trial in each movement is $109 \times 10 \times 10 \times 2$.

Mutual Subspace Method (MSM) [39] is one of most effective image set based classifiers. MSM uses the canonical angles between an input subspace and references subspaces as a similarity between two image sets. We use the extension of MSM, Orthogonal MSM (OMSM) [40], which has higher discriminative ability than the original MSM.

We compared CoALBP with that of the raw image feature and LBP histogram. The raw image feature was generated by vectorizing a 16×16 grayscale image. For LBP histogram and CoALBP histogram, after the image was divided to 3×3 subregions, image feature was extracted from each subregion, and all the extracted features were concatenated into a single vector. The dimensions of feature vector for LBP and CoALBP are 256×9 and 1024×9, respectively. The scale parameter of LBP, s, was set to 2 and the interval of LBP, r, was set to 4. The extracted image feature was compressed into a low-dimensional feature by principal component analysis (PCA).

Results

Table 2.1 shows the error rates and equal error rates of all the methods. Figure 2.12 shows the ROC curves of all the methods. The results show that the performance of LBP was significantly improved by introducing co-occurrence, which is discarded in the original LBP.

Also, the results show the performance in the vertical shaking was better than

Table 2.1: Results of the image set based face recognition. "sr" indicates that the image feature is extracted from the image divided to the subregions of an input image.

(a) Horizontal

	ER [%]	EER[%]
Raw	4.67	2.49
LBP	11.40	5.09
CoALBP	1.94	1.10
LBPsr	2.10	1.11
CoALBPsr	**1.87**	**1.00**

(b) Vertical

	ER [%]	EER[%]
Raw	4.00	2.11
LBP	7.54	3.83
CoALBP	0.75	0.46
LBPsr	0.79	0.53
CoALBPsr	**0.40**	**0.25**

Table 2.2: Outex database details.

Outex ID	Classes	Image sizes	Training/Testing images
Outex_TC_00000	24	128×128	10
Outex_TC_00001	24	64×64	44
Outex_TC_00002	24	32×32	184
Outex_TC_00016	319	128×128	10

that in the horizontal shaking. This is because that the horizontal shaking has a larger variance than vertical shaking, as shown in Figure 2.11, and exact cropping a face region is difficult. For this reason, dividing an entire image into several subregions was more effective in the vertical shaking than in the horizontal shaking.

We can see that exact cropping a face region is required to realize high-performance face recognition even if using CoALBP. This indicates that, even in face recognition based on CoALBP, it is important to crop a face region precisely and to prevent mixing of a histogram of each subregion.

2.3.3 Texture recognition without rotation variance

Settings

Finally, we evaluated CoALBP using Outex_TC sets in the Outex database [2]. Table 2.2 shows the details of the dataset, Figure 2.13 shows examples of images from the dataset. Outex_TC_00000 – 00002 contains grayscale images of 24 textures that are generated by dividing large texture images at different sizes. Outex_TC_00016 contains grayscale images of 319 textures. The average image intensity value is normalized to 128, with a standard deviation of 20.

The images were randomly split between training and testing sets. This division was repeated 100 times to produce 100 evaluation sets. The average of all

Table 2.3: Results of the texture recognition without rotation variance.

	LBP			CoALBP	
Outex ID	Original	(+)	(×)	(+)	(×)
Outex_TC_00000	0.996	0.986	0.989	**0.999**	**0.999**
Outex_TC_00001	0.985	0.930	0.948	**0.989**	**0.989**
Outex_TC_00002	0.871	0.721	0.742	0.906	**0.915**
Outex_TC_00016	0.783	0.686	0.708	**0.830**	0.820

correct rates over 100 iterations was defined as the final rate.

CoALBP employed $s = 1, \cdots, 3$ and $r = 1, \cdots, 5$, and the best correct rates were used as the reported results. The nearest neighbor method with L_1 distance was used as a classifier.

Results

Table 2.3 shows the results of the experiment. The results confirm a significant advantage of CoALBP against the LBP histogram features, which are not considering the co-occurrence. The CoALBP with parameters $s = 1$, $r = 2$ achieved the best performance among all the features. From Table 2.2 and Table 2.3, we can see that the correct rates decrease as the image size becomes smaller despite the increase in the number of training images. This can be considered that these image features are based in histograms and became unstable by reducing the number of pixels in an image. Therefore, in texture recognition with histogram-based image feature, it is more important to use an input image in a large image than to increase the number of training images.

2.4 Application: HEp-2 cells classification

We applied CoALBP to HEp-2 cells classification. Our method that is a combination of CoALBP and linear SVM achieved first place in the HEp-2 cells classification contest 2012 [41]. Figure 2.14 shows the performance of methods entered in the contest. Our method achieved the highest performance in both metrics: cell level accuracy and image level accuracy. Furthermore, our method archived a similar performance with the performance by a human specialist. These results indicate the effectiveness of CoALBP. In the next chapter, we describe RIC-LBP, which has better performance than this result and analysis the performance in detail.

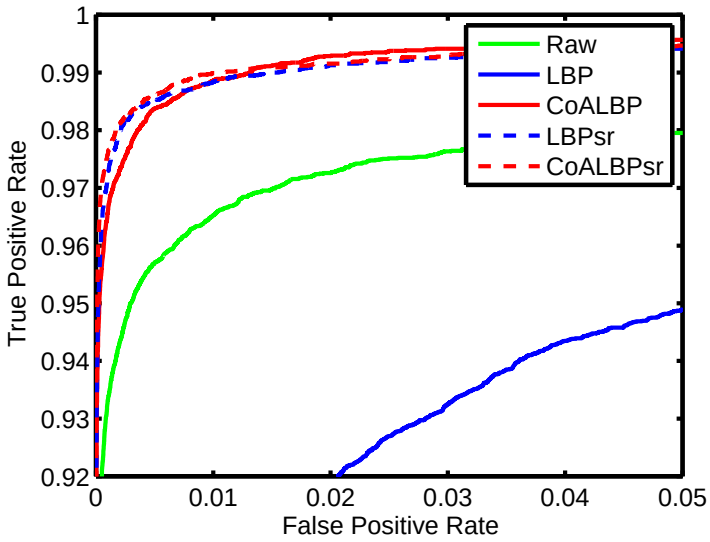
2.5 Summary

In this chapter, we proposed the CoALBP to enhance LBP employed in many applications. CoALBP is implemented by histograms of LBP pairs within several displacements, and has higher descriptive power than LBP, inheriting the advantages of LBP. Also, we have described the interpretation from perspectives of represented spatial patterns and of curvature. Then, we confirmed the effectiveness of the CoALBP on several tasks: face recognition under the various illuminations, face recognition based on image sets, and texture recognition. Besides, we have reported the results of the HEp-2 cells classification contest 2012, and have confirmed that our method had high and practical performance.

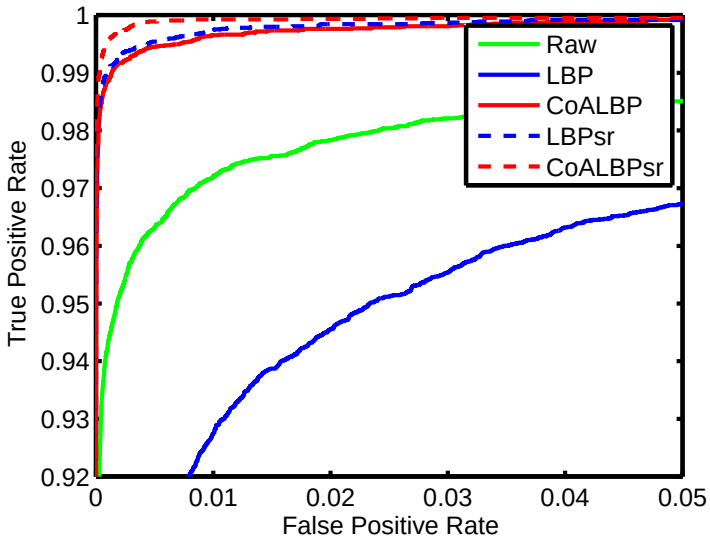
(a) Horizontal

(b) Vertical

Figure 2.11: Example images for image set based face recognition experiment.

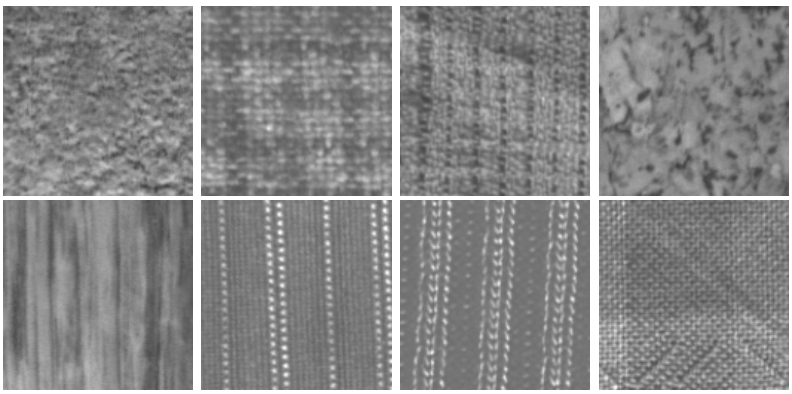


(a) Horizontal

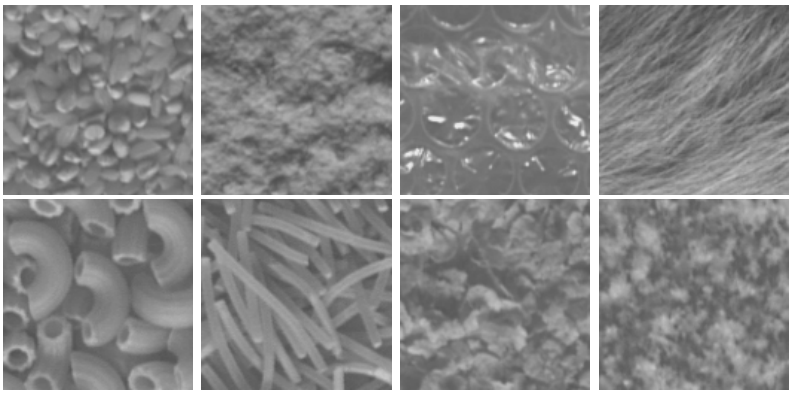


(b) Vertical

Figure 2.12: ROC curves of the image set based face recognition.



(a) Outex_TC_00000 – 00002



(b) Outex_TC_00016

Figure 2.13: Example images in Outex database.

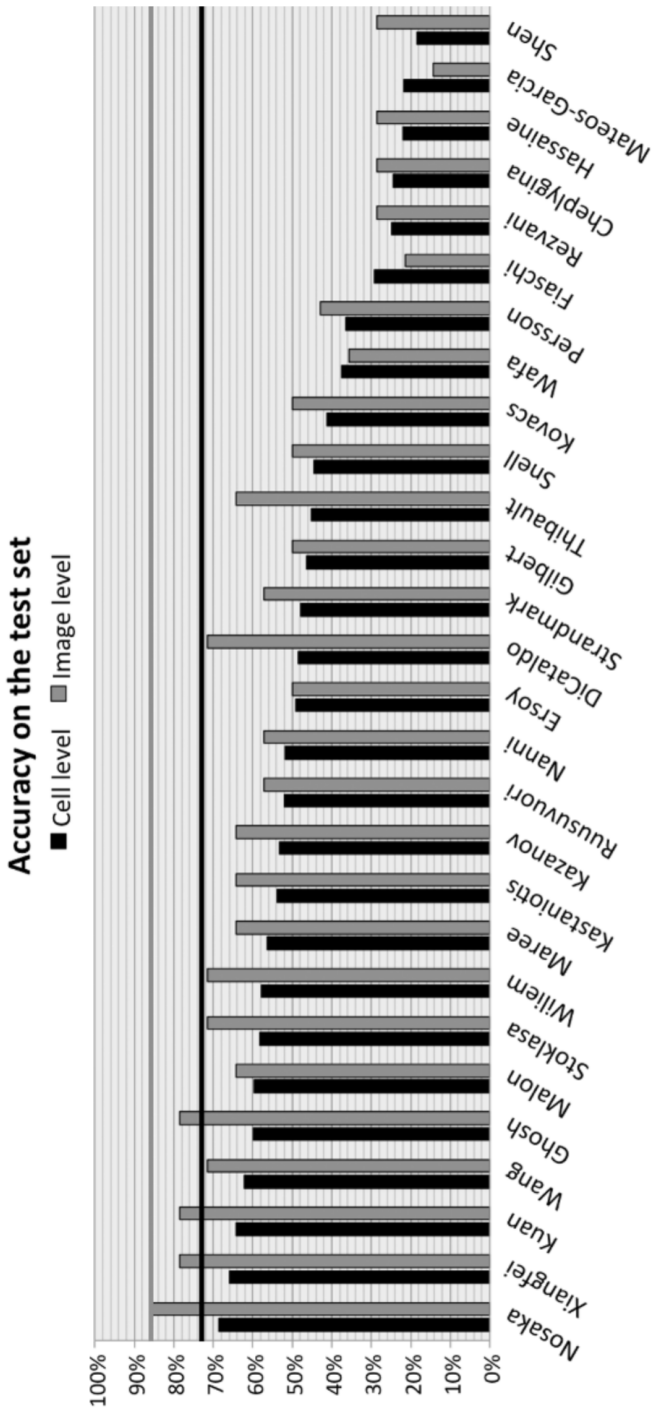


Figure 2.14: Results of the HEp-2 cells classification contest 2012. Horizontal lines indicate performance by a specialist. Our method (Nosaka) is 1st place and achieved a similar performance with the performance by the specialist. This image is created from an image in [41]

Chapter 3

Rotation Invariance of Co-occurrence among Adjacent LBPs

In this chapter, we describe RIC-LBP, in which the concept of rotation invariance is introduced to CoALBP described in the previous chapter. Then, we demonstrate the effectiveness of the combination of rotation invariance and co-occurrence in texture classification task and HEp-2 cells classification task. Finally, we provide the experimental results of the method proposed on the HEp-2 cell sorting task in detail.

3.1 Introduction

The rotation invariance is an essential characteristic for tasks such as texture recognition and cell classification. In such tasks, the category of the input object is invariant against the rotation of the object. Thus, the output of a classifier is required to be rotation invariant. In general, this property is considered in the feature extraction step by utilizing a rotation invariant feature, as the output of a classifier is not necessarily rotation invariant even if the classifier is trained with a set of images rotated in various directions. Moreover, the rotation invariant feature can effectively reduce the variation of input by its rotation invariance.

Several LBP-based features with rotation invariance have already been proposed [13, 14, 15]. LBP^{ri} [13] and LBP^{riu2} [14] obtain rotation invariance by summarizing frequencies of binary patterns that are rotation equivalent each other.

It means that an LBP histogram is suppressed along rotation to get rotation invariance, and the descriptive ability of LBP is lost. Therefore, it is essential to extract spatial information to improve descriptive ability.

To overcome the problem, we propose an extended CoALBP feature, Rotation Invariance of Co-occurrence among Adjacent LBPs (RIC-LBP). RIC-LBP obtains the rotation invariance by considering rotation equivalent LBP pairs, not rotation equivalent single LBP such like LBP^{ri} and LBP^{riu2}. Thus, RIC-LBP can provide the rotation invariance and the high descriptive ability of CoALBP.

3.2 Proposed method

3.2.1 Idea

To introduce rotation invariance to CoALBP, consider rotation invariance of LBP pair. The simplest way to embed rotation invariance is to attach a rotation invariant label to each LBP pair.

For example, in Figure 3.1 there are two types of LBP pairs, each having four configurations. The same label is attached to each of these eight LBP pairs because each LBP pair is equal to the others regarding rotation. This relation among LBP pairs is called *rotation equivalence*; a set of rotation equivalent LBP pairs is called a *rotation equivalence class* of LBP pairs. The LBP pairs in Figure 3.1 constitute one rotation equivalence class. This strategy is inspired from LBP^{ri} [13] and LBP^{riu2} [14].

Nevertheless, finding such rotation equivalent LBP pairs is difficult since the number of possible LBP combinations is huge and the relation between LBPs of the LBP pair can switch each other by rotation. To solve this problem, we automatically detect pairs with the same value by using a computational algorithm that is described in the next section. As a result, RIC-LBP can simultaneously provide a high descriptive ability and invariance to image rotation.

3.2.2 Rotation equivalence class of LBP pair

As shown in Figure 3.1, the upper LBP pairs are equivalent to LBP pairs that have been rotated 180 degrees from the lower LBP pairs. This indicates that, for finding the rotation equivalent LBP pairs, it is necessary to consider only two cases: (i) a case in which LBP pairs of $\theta = 0, \pi/4, \pi/2, 3\pi/4$ have rotation equivalence and (ii) a case in which LBP pairs that are rotated by 180 degrees have rotation

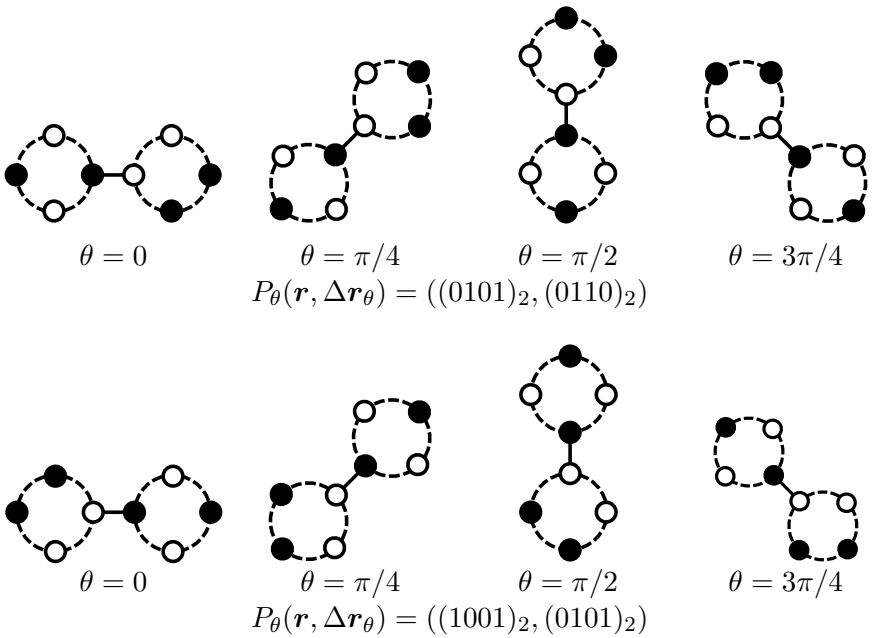


Figure 3.1: An example of the rotation equivalence class. The same label is attached to these LBP pairs. Here, black and white circles indicate binary patterns; black circle is "0" and white circle is "1".

equivalence.

First, in order to consider case (i), we modify the definition of LBP pair. The modified LBP pair is written as follows:

$$P_\theta(\boldsymbol{r}, \boldsymbol{r} + \Delta\boldsymbol{r}_\theta) = (LBP_\theta(\boldsymbol{r}), LBP_\theta(\boldsymbol{r} + \Delta\boldsymbol{r}_\theta)), \tag{3.1}$$

$$LBP_\theta(\boldsymbol{r}) = \sum_{i=0}^{N-1} sgn(I(\boldsymbol{r} + \Delta\boldsymbol{s}_{i,\theta}) - I(\boldsymbol{r}))2^i, \tag{3.2}$$

$$\Delta\boldsymbol{s}_{i,\theta} = (s\cos(\theta_i + \theta), s\sin(\theta_i + \theta)), \tag{3.3}$$

where θ indicates orientation of the displacement. By this modification, the arrangement of the LBP pairs is modified from a grid-like arrangement to a circular arrangement. Also, θ acts neighboring pixels of LBP as like the bias of the rotation angle in LBP, according to the orientation of the displacement, neighboring pixels also rotate. Thus, the LBP pair of each configuration has the same value regarding rotation.

Next, we consider case (ii). In this case, we use a rule that an LBP pair that is rotated 180 degrees from $P_\theta(\boldsymbol{r}, \boldsymbol{r} + \Delta\boldsymbol{r}_\theta)$ is equal to $P_{\theta+\pi}(\boldsymbol{r} + \Delta\boldsymbol{r}_\theta, \boldsymbol{r})$. According

Algorithm 1 Calculate a mapping table M.

Input: N // number of neighbor pixels.
Output: M // mapping table ($N_P \times N_P$ matrix)

```
id ⇐ 1, N_P ⇐ 2^N
for i = 0, ··· , N_P − 1 do
  for j = 0, ··· , N_P − 1 do
    if M(i, j) = null then
      i' ⇐ i ≫ N/2,  j' ⇐ j ≫ N/2
      M(i, j) ⇐ id,  M(j', i') ⇐ id
      id ⇐ id + 1
    end if
  end for
end for
```

to this rule, we can consider that these LBP pairs have rotation equivalence. We implement this rule by a mapping table M that has a label for each LBP pair. The mapping table M is generated by using Algorithm 1. In Algorithm 1, "$\gg$" is a circular shift; also, "$i' = i \gg N/2$" means to rotate LBP i by 180 degree (*e.g.*, $i = (1000)_2$ becomes $i' = (0010)_2$).

By using mapping table M, we define a rotation invariant label for an LBP pair at $\boldsymbol{r}$ (*i.e.*, RIC-LBP) as follows:

$$P_\theta^{RI}(\boldsymbol{r}) = M(P_\theta(\boldsymbol{r}, \Delta\boldsymbol{r}_\theta)). \tag{3.4}$$

Finally, a histogram of RIC-LBPs is generated from $P_\theta^{RI}(\boldsymbol{r})$ for $\boldsymbol{r}$ in the entire image and θ in all displacement orientations.

Since the number of the rotation equivalence classes for the LBP pairs determines the dimension of the RIC-LBP histogram vector, we describe this in more detail as follows. The number of possible LBP pairs is $N_P^2 \times 4$. By considering case (i), the number of possible patterns becomes N_P^2. Moreover, by considering case (ii), the number of possible patterns is halved. Here, we consider a symmetric LBP pair as shown in Figure 3.2; the number of symmetric LBP pairs is N_P. Therefore, the number of rotation equivalence classes is $N_P(N_P + 1)/2$.

3.2.3 Process flow of obtaining RIC-LBP histogram

We explain how to generate the RIC-LBP histogram from an input image with Eq. (3.1) and the mapping table M.

First, we explain how Eq. (3.1) and mapping table M work using Figure 3.3.

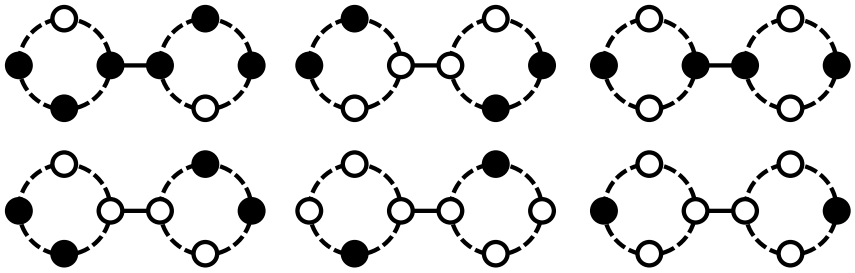


Figure 3.2: Examples of symmetric LBP pair.

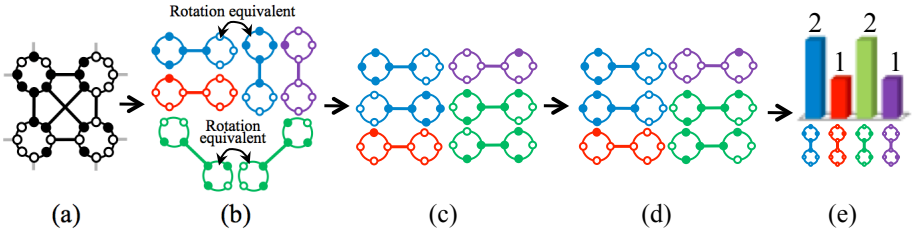


Figure 3.3: An example of obtaining RIC-LBP. LBP pairs are colored by colors that indicates rotation equivalence class. (a) Example image. (b) LBP pairs of the example image. (c) Labeling of each LBP pair using Eq. (3.1). (d) Re-labeling of each LBP pair by applying the mapping table M. (e) RIC-LBP histogram.

The example image has four LBPs (Figure 3.3a). The image is decomposed into six LBP pairs (Figure 3.3b). We then have two sets of LBP pairs that have rotation equivalence as indicated by arrows in Figure 3.3b. By Eq. (3.1), the effect of configurations is removed from these LBP pairs, as shown in Figure 3.3c. By utilizing mapping table M, these pairs are arranged as shown in Figure 3.3d. As we can see, LBP pairs in Figure 3.3d are rotation invariant. By such a process, we obtain a RIC-LBP histogram of the example image, as shown in Figure 3.3e.

Next, we explain the overall process flow to obtain a RIC-LBP histogram of an image using Figure 3.4. Firstly, compute $LBP_\theta(\boldsymbol{r})$ at every pixel $\boldsymbol{r}$ throughout the entire input image (Figure 3.4a). Next, compute a histogram of $P_\theta(\boldsymbol{r}, \Delta\boldsymbol{r}_\theta)$ (Figure 3.4b). Finally, combine the histogram using mapping table M and obtain a histogram of $P_\theta^{RI}(\boldsymbol{r})$ (Figure 3.4c). Mapping table M is calculated offline. The final histogram is $N_P(N_P+1)/2$ dimensional vector and is applied to a classifier.

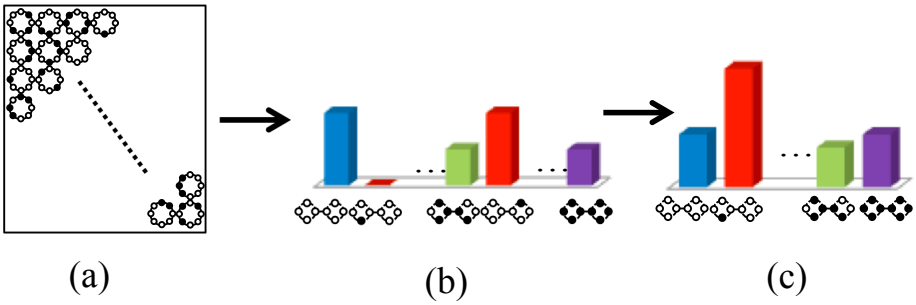
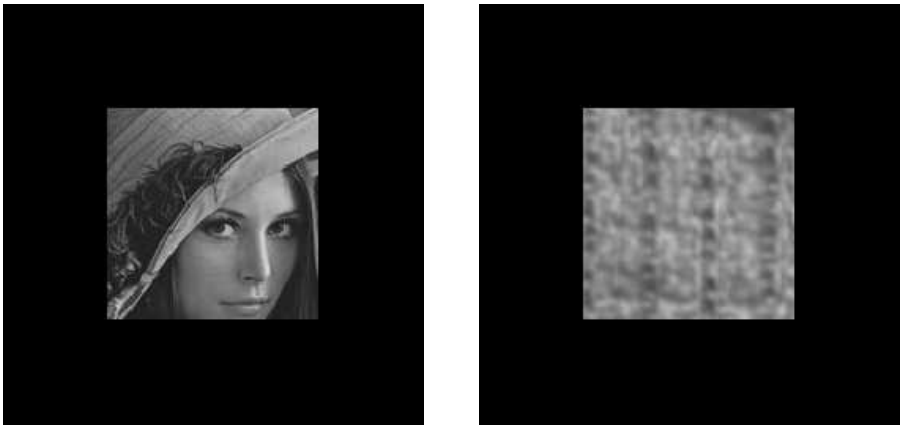


Figure 3.4: Process flow to obtain the RIC-LBP histogram. (a) Input LBP image. (b) Histogram of $P_\theta(\boldsymbol{r}, \Delta\boldsymbol{r}_\theta)$. (c) RIC-LBP histogram.



(a) Lena (b) Texture

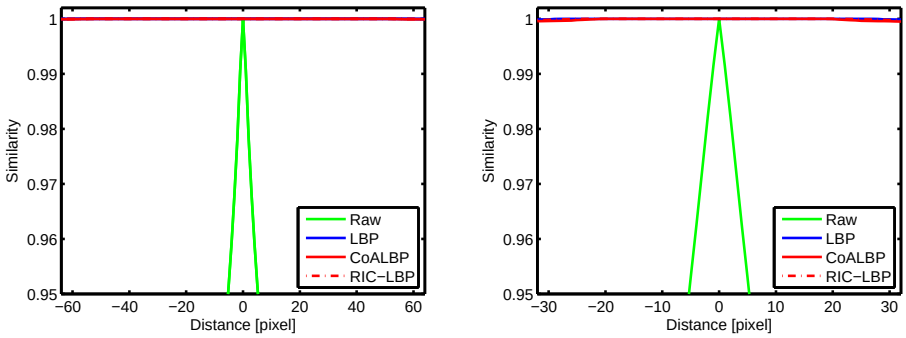
Figure 3.5: Test images used in the evaluation of the robustness against shift in position.

3.3 Experimental evaluation

As an experimental evaluation, firstly, we investigated basic properties: robustness against position variations and rotation variations. Then we evaluated two recognition tasks: texture recognition including rotation variations and HEp-2 cells classification. In particular, in the experiment with HEp-2 cells classification, performance with RIC-LBP was investigated in more detail.

3.3.1 Fundamental properties

Robustness against position variations



(a) Lena (b) Texture

Figure 3.6: Robustness against variations in position of the object. X axis indicates the distance from the center of the image.

Settings In this experiment, we evaluated the robustness of the proposed features against variations of the position of an object. The evaluation procedure is as follows. First, objects were placed at the center of the images, respectively, as shown in Figure 3.5, and an image feature $\boldsymbol{f}$ was extracted from the image. Then image features $\boldsymbol{f}'$ were extracted by the same method as that used in the initial position while shifting the position of the object from the left to right side in the image. The correlation-based similarities between $\boldsymbol{f}$ and $\boldsymbol{f}'$ were calculated to measure the robustness the proposed features.

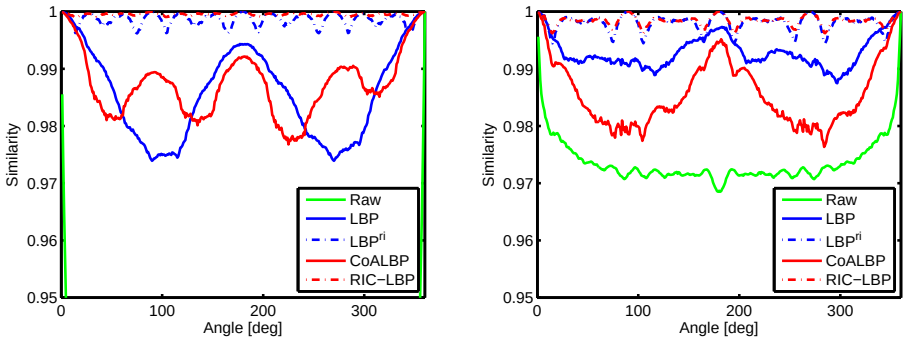
Results Figure 3.6 shows the result of the experiment. This result shows that the variations of position do not influence the similarity of LBP-based feature. This is because LBP-based feature is based on histograms with position invariant. From the result, we can confirm the proposed features are very robust against variations of position.

Robustness against image rotation

Settings In this experiment, we evaluated the robustness of the proposed features against image rotation. The evaluation procedure is as follows. First, the face and texture used in the first experiment were set respectively, as shown in Figure 3.7, and image feature $\boldsymbol{f}$ was extracted from the image. Then, image feature $\boldsymbol{f'}$ were extracted by the same method as that used in the first orientation while rotating the object from 0 to 360 degrees. The correlation-based similarities between $\boldsymbol{f}$ and $\boldsymbol{f'}$ were calculated to measure the robustness of the proposed features.

(a) Lena (b) Texture

Figure 3.7: Test images used in the evaluation of the robustness against image rotation.



(a) Lena (b) Texture

Figure 3.8: Robustness against rotation of an object.

Results Figure 3.8 shows the result of the experiment. RIC-LBP outputs higher similarity than raw image feature, LBP and CoALBP, in all angles. Although RIC-LBP is more robust against image rotation, it can be confirmed that RIC-LBP is not rotation invariant perfectly except for 90, 180 and 270 degrees. This is because that RIC-LBP assumes rotation angle of LBP pair is eight orientation as similar with LBP^{r}i. Even though, RIC-LBP is very robust against also image rotation.

Computational cost

Settings In this experiment, we measured the computational time of performing the proposed feature extraction by using a single of the Intel Xeon with 2.53GHz. Each method was implemented using MATLAB R2011a. Refer to the author's

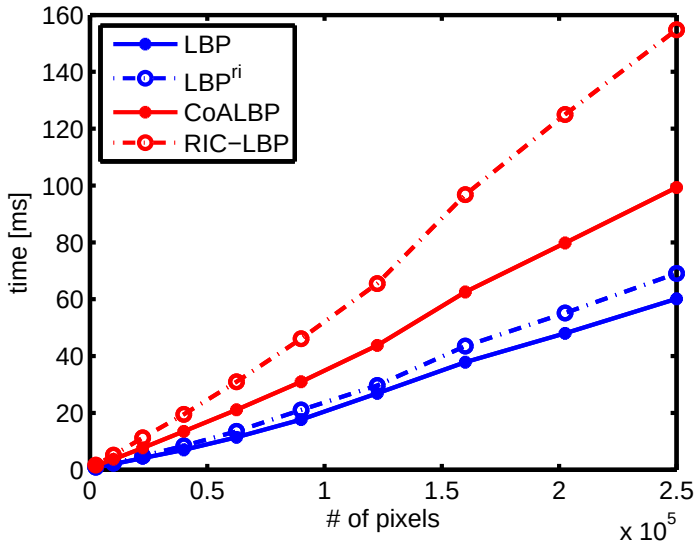


Figure 3.9: Computational time for each feature extraction method.

website[1] for the implementation of LBP and LBP^{ri}.

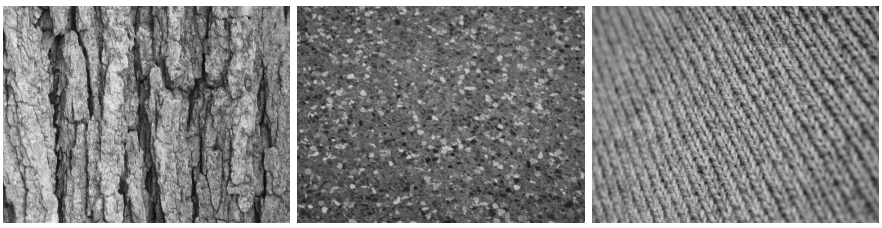
Results Figure 3.9 shows the computational time of each feature. The results show that the computational time of the proposed features was longer than that of others. This is because that four histograms are required to be extracted from an image in the proposed features. Even though, the speed to extract the proposed features is fast enough for real applications. For processing a 300×300 pixels image, computational time of CoALBP and RIC-LBP were 31ms and 46ms, respectively.

3.3.2 Texture recognition with rotation variance

Settings We evaluated RIC-LBP for texture recognition with rotation variance using the UIUC texture database [42].

The database contains texture images of 25 classes. Each class consists of 40 images of size 640×480 pixels. The images in the dataset have rotation variations. Some examples of texture images are shown in Figure 3.10. The images of each class were randomly split into training and testing sets. This division was repeated 20 times to produce 20 evaluation sets. The average of all correct rates over 20 iterations was defined as the final rate. The parameters of the LBP features were set to the same setting as in Section 2.3.3.

[1] http://www.cse.oulu.fi/CMV/Downloads



bark1 floor1 knit

wall water wood1

Figure 3.10: Example images in UIUC texture database.

Table 3.1: Accuracy of the texture recognition with rotation variance.

Method	Correct Rate [%]
LBP [3]	82.55
LBP^{ri} [14]	83.51
LBP^{riu2} [14]	57.33
CoALBP	81.49
LBP-HF [15]	93.60
RIC-LBP	88.27

Results Accuracy of each method is shown in Table 3.1. The performance of RIC-LBP was better than that of almost all the other conventional LBP methods, such as LBP^{ri} and LBP^{riu2}. However, the LBP-HF method, which utilizes the discrete Fourier transform, achieved better performance than RIC-LBP. This is because RIC-LBP considers rotation at local regions, whereas LBP-HF considers the rotation of the entire image. LBP-HF is thus better suited for this type of texture dataset, which contains global rotation equivalence images. This experimental result indicates that the performance of RIC-LBP for the texture dataset may be further improved by also considering the rotation of the entire image by using a method such as the discrete Fourier transform.

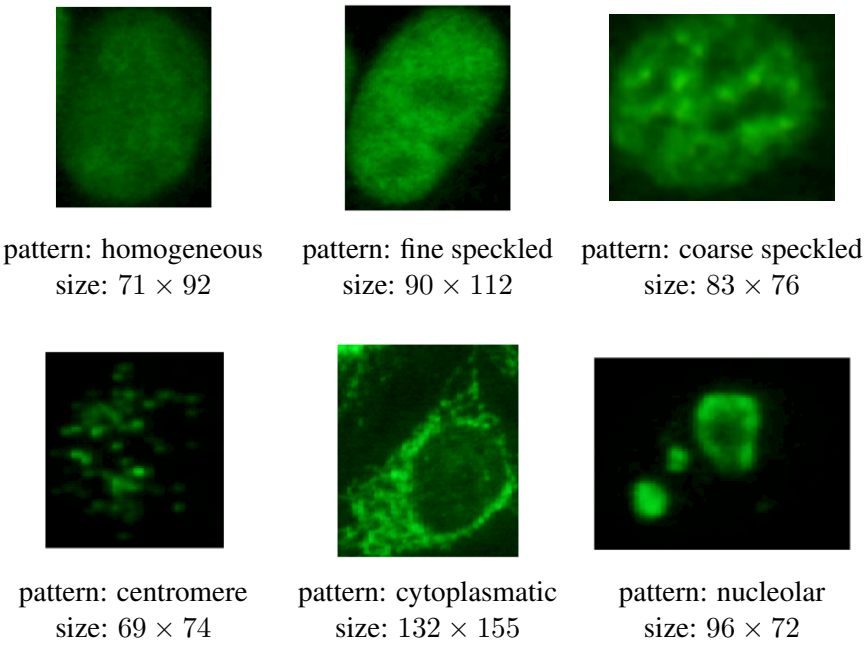


Figure 3.11: Example images in HEp-2 cell dataset. "size" indicates the size of the displayed image; other images not displayed have different sizes.

3.3.3 HEp-2 cells classification

Settings

Dataset For the experimental evaluation, we used the MIVIA HEp-2 images dataset [1], which is a dataset of IIF images. The dataset consists of 28 slide images, each containing several cells (min: 13 cells; max: 119 cells). The total number of cells is 1455. The average size of the cell images is about 86×87 pixels. The cells in the images were manually segmented and annotated by experts. Each image was assigned one of six types of staining patterns, namely, *homogeneous*, *fine speckled*, *coarse speckled*, *centromere*, *cytoplasmatic*, and *nucleolar*, as shown in Figure 3.11. Each image was also assigned and one of two types of intensity patterns, namely, *positive* or *intermediate*. As shown in Figure 3.13, intermediate intensity images are darker than positive intensity images, and noise may be higher. Note that the cells in each slide image were assigned the same type of staining pattern and the same type of intensity pattern.

Evaluation protocol For a fair comparison, the performance evaluation was conducted using two protocols; these protocols are following the special issue of the

journal *Pattern Recognition* on analysis and recognition of indirect immunofluorescence images [43].

One protocol was performed by dividing the cells into training and test sets in the same way as the HEp-2 cell classification contest that was hosted by ICPR 2012 [44]. The training set contained 14 slide images (721 cells). The test set contained 14 slide images (734 cells). The other protocol was performed using the leave-one-out protocol overall 28 slide images: for each slide image in the dataset, the classifier was trained using the other 27 slide images. Since the leave-one-out protocol is more reliable, we mainly discuss using the results of that protocol.

For each protocol, results were reported at the cell level and the slide level [2]. At the cell level, the result was based on the prediction for the staining pattern of each cell. At the slide level, the result was based on the prediction for the staining pattern of each slide image using the staining pattern most frequently assigned to the cells within that image.

Training and inference Figure 3.12 shows the overview of the HEp-2 cells classification process. We describe each process as follows.

As preprocessing that is a common process of training and inference phase, a cell image is converted to a grayscale image by extracting only the green component of the RGB color space.

Next, an image feature (*e.g.*, LBP and RIC-LBP) is extracted from the preprocessed image. The size of a stained HEp-2 cell depends strongly on the type of its staining pattern, but RIC-LBPs cannot deal with large changes in the cell size. To cope with this problem, RIC-LBPs is extracted with several parameters corresponding to different scales, and finally, the extracted RIC-LBPs are combined into a final feature vector. This multi-scale feature extraction incorporates the ability to absorb changes in the cell size into RIC-LBP. In this experiments, the parameters corresponding to three scales, *i.e.*, $(s, d) = (1, 2), (2, 4), (4, 8)$, were used. Thus, the final feature vector was a 136×3-dimensional vector.

As classifier for high-accuracy and high-speed classification, we apply linear classification by an SVM classifier and a one-versus-all strategy to handle multi-class classification. The linear SVM is trained using extracted features from both the original images and the synthesized images. In this experiments, we used LIBLINEAR [45] due to its special design for linear SVM.

For training, we synthesize additional training images from the collected im-

[2] For understandable, we use "slide level" instead of "image level" used in [43].

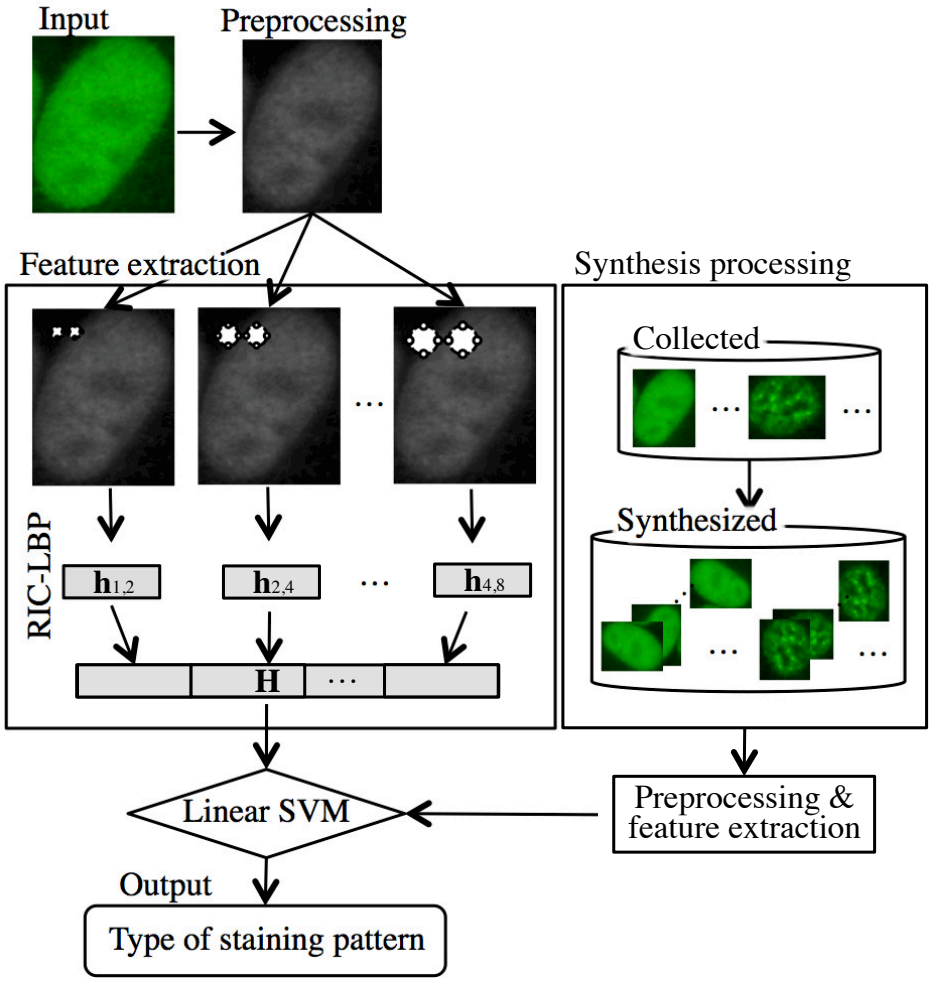


Figure 3.12: Overview of HEp-2 cells classification process.

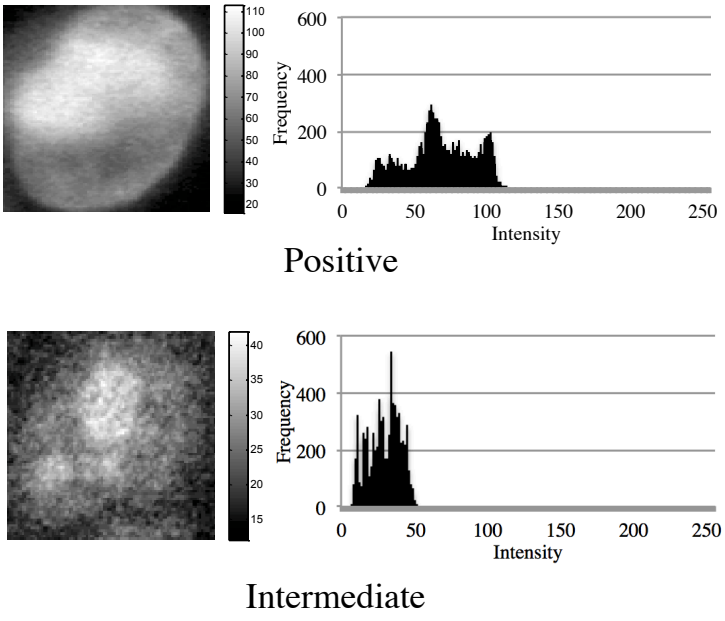


Figure 3.13: Example images and intensity histograms of each intensity pattern. Staining patterns of these images are homogeneous patterns. The brightness and contrast of these images were modified for better visibility.

ages in advance to deal with the global rotation of an input image. This synthesis process is realized by rotating the collected images by various angles.

Result: comparison with LBP-based features

This section shows a comparison of various LBP-based image features. To focus on the effectiveness of each LBP feature, the classifier was trained using only the given images without synthesis processing. RIC-LBP was compared with LBP [3][3], LBP^{ri} [13][3], LBP^{riu2} [14][3], LBP-HF [15][3], CLBP_S_M [46][4], and CoALBP [47].

The LBP-based features had two parameters to be considered: the number of neighbor pixels, N, and the radius of LBP, s, in Eq.(2.1). For CoALBP and RIC-LBP, the interval between the LBP pair, d, was added. The parameters of all the methods were tuned to obtain the best performances through preliminary experiments. Table 3.2 shows the parameters used for each method in our experiments. Here, note that three features extracted with three different parameter sets were combined into a final feature vector to deal with the change in scale, as described

[3]code : http://www.cse.oulu.fi/wsgi/CMV/Downloads/LBPMatlab
[4]code : http://www4.comp.polyu.edu.hk/~cslzhang/papers.htm

Table 3.2: Parameters of each method.

Method	(N,s)
LBP	(8,1), (8,2), (8,4)
LBP^{ri}	(8,1), (8,2), (8,4)
LBP^{riu2}	(8,1), (16,2), (16,4)
LBP-HF	(8,1), (16,2), (16,4)
CLBP_S_M	(8,1), (8,2), (8,4)

Method	(N,s,d)
CoALBP	(4,1,2), (4,2,4), (4,4,8)
RIC-LBP	(4,1,2), (4,2,4), (4,4,8)

Table 3.3: Accuracy rates (%) obtained with LBP-based image features in leave-one-out protocol.

Method	LBP	LBP^{ri}	LBP^{riu2}	LBP-HF
Cell level	49.48%	47.07%	51.75%	58.14%
Slide level	57.14%	57.14%	57.14%	71.42%

Method	CLBP_S_M	CoALBP	RIC-LBP
Cell level	56.90%	59.03%	**66.46%**
Slide level	64.28%	71.42%	**78.57%**

previously for RIC-LBP.

Table 3.3 shows the accuracy rate of each feature in the leave-one-out protocol. CoALBP outperformed LBP, LBP^{ri}, and LBP^{riu2}; these methods are often used for HEp-2 cell classification. From the results, we can confirm the effectiveness of using co-occurrence. Furthermore, although CoALBP won the 2012 HEp-2 cells classification contest, RIC-LBP achieved the best performance among the other features. The results demonstrate the effectiveness of using both co-occurrence and rotation robustness for HEp-2 cell classification.

Result: effectiveness of the synthesis process

This section shows the effectiveness of synthesized training images. The additional training images were synthesized by rotating the original training images. The number of synthesized images increases as the step angle of rotation decreases. However, for RIC-LBP, the number of different training images does not increase linearly due to rotation equivalence. In our settings, the relationship between the

Table 3.4: Setting of rotation angles for the synthesis process.

RIC-LBP	
Number of valid training images	Step angle
Original	–
Original $\times$ 2	45°
Original $\times$ 3	60°
Original $\times$ 9	20°
Original $\times$ 18	5°

CoALBP	
Number of valid training images	Step angle
Original	–
Original $\times$ 6	60°
Original $\times$ 8	45°
Original $\times$ 12	30°
Original $\times$ 18	20°

number of training images and the step angle is summarized in Table 3.4. For instance, the number of training images for RIC-LBP is three times as many as that of the original set when the step angle is set to 60°. On the other hand, the number of training images for CoALBP is six folds that of the original set under the same condition.

Figure 3.14 shows the relationship between the accuracy rate and the number of synthesized training images. From the results, we can confirm that the best accuracy rate was 70.65%, and the synthesis process significantly improved the accuracy rate. Besides, the accuracy rates of RIC-LBP were better than those of CoALBP for all settings. This is because the method using CoALBP considered only global rotation in the synthesis process, whereas the method using RIC-LBP considered both local and global rotations. Therefore, the combination of RIC-LBP and the synthesis process is effective for HEp-2 cell classification.

Result: detail of performance

Tables 3.5, 3.6, and 3.7 show accuracy rates and confusion matrices of the experiments using the contest protocol and the leave-one-out protocol, respectively. Tables 3.8 and 3.9 show the details of the classification results for each slide image in the leave-one-out protocol. The confusion matrices in the contest protocol and the leave-one-out protocol show similar trends as follows.

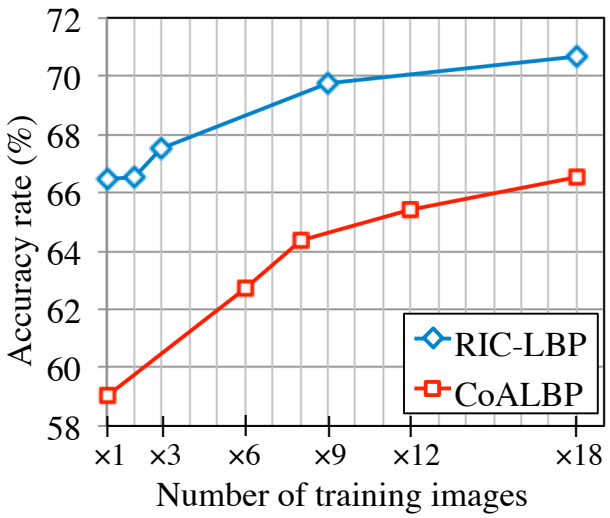


Figure 3.14: Change in accuracy rate (cell level) against the number of training images.

Table 3.5: Average accuracy rates of the proposed method for the contest protocol and the leave-one-out protocol.

	Contest	Leave-one-out
Cell level	68.53%	70.65%
Slide level	71.43%	85.71%

From the confusion matrices in Tables 3.6 and 3.7, we can confirm that the accuracy rates for the cytoplasmatic pattern and the centromere pattern are higher than for the other types. This is because the cytoplasmatic pattern has a more characteristic shape (fibers) when compared with the others. The centromere pattern is also different from the other types, in that it has several individual spots.

On the other hand, the accuracy rates of the coarse speckled pattern and the fine speckled pattern were not very high, as shown in Tables 3.6 and 3.7. Misclassifications in this case that can be summarized as belonging to one of the following two cases.

In the first case, the fine speckled pattern and coarse speckled pattern were misclassified as one another. For example, Images #2 and #17 were classified as incorrect types in the leave-one-out protocol, as shown in Tables 3.8 and 3.9. In particular, Image #17, a coarse speckled pattern, was completely misclassified as

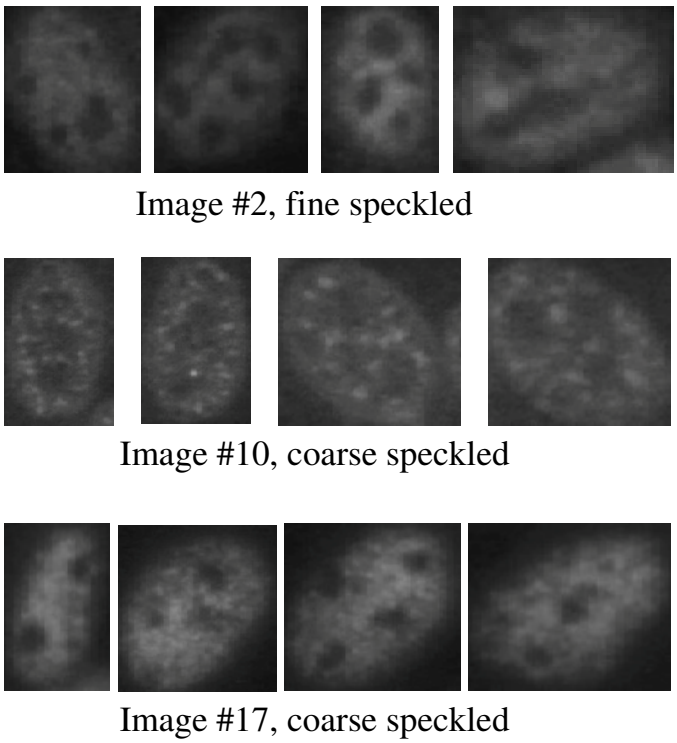


Figure 3.15: Examples of the fine speckled pattern and the coarse speckled pattern. The brightness and contrast of these images were modified for better visibility.

a fine speckled pattern. This is because the appearances of these staining patterns are very similar to each other, as shown in Figure 3.15.

In the second case, the fine speckled pattern and coarse speckled pattern were misclassified as the centromere pattern. For example, Images #10 and #15 were classified as the centromere pattern in the leave-one-out protocol, as shown in Tables 3.8, and 3.9. This misclassification happened because the difference between the coarse speckled pattern and the centromere pattern is very small. The coarse speckled pattern and the centromere pattern are different only in the unstable local fine textures under strong noise; they have similar small spots.

Besides, Figure 3.16 shows that the accuracy rates for intermediate intensity were about 20 percentage points lower than for positive intensity. This implies that there is room for improvement of the proposed method and its robustness against noise.

Table 3.6: Confusion matrices of the proposed method for the experiment using the contest protocol.

Cell level	Prediction											
	CE		HO		NU		CO		FI		CY	
CE	**90.6%**	**(135)**	0.67%	(1)	8.05%	(12)	0%	(0)	0.67%	(1)	0%	(0)
HO	1.66%	(3)	**63.88%**	**(115)**	2.77%	(5)	5%	(9)	24.44%	(44)	2.22%	(4)
NU	7.19%	(10)	20.14%	(28)	**71.94%**	**(100)**	0%	(0)	0%	(0)	0.71%	(1)
CO	13.86%	(14)	1.98%	(2)	5.94%	(6)	**43.56%**	**(44)**	32.67%	(33)	1.98%	(2)
FI	40.35%	(46)	9.64%	(11)	0.87%	(1)	0%	(0)	**49.12%**	**(56)**	0%	(0)
CY	0%	(0)	0%	(0)	3.92%	(2)	9.8%	(5)	0%	(0)	**86.27%**	**(44)**

Slide level	Prediction											
	CE		HO		NU		CO		FI		CY	
CE	**100%**	**(3)**	0%	(0)	0%	(0)	0%	(0)	0%	(0)	0%	(0)
HO	0%	(0)	**100%**	**(2)**	0%	(0)	0%	(0)	0%	(0)	0%	(0)
NU	0%	(0)	0%	(0)	**100%**	**(2)**	0%	(0)	0%	(0)	0%	(0)
CO	33.33%	(1)	0%	(0)	0%	(0)	**33.33%**	**(1)**	33.33%	(1)	0%	(0)
FI	50%	(1)	0%	(0)	0%	(0)	0%	(0)	**50%**	**(1)**	0%	(0)
CY	0%	(0)	0%	(0)	0%	(0)	0%	(0)	0%	(0)	**100%**	**(2)**

CE: centromere; HO, homogeneous; NU: nucleolar; CO: coarse speckled; FI: fine speckled; CY: cytoplasmic.

Table 3.7: Confusion matrices of the proposed method for the experiment using the leave-one-out protocol.

Cell level	Prediction											
	CE		HO		NU		CO		FI		CY	
CE	**81.51%**	**(291)**	0.28%	(1)	4.2%	(15)	7.84%	(28)	5.88%	(21)	0.28%	(1)
HO	1.21%	(4)	**73.03%**	**(241)**	11.21%	(37)	4.54%	(15)	9.69%	(32)	0.3%	(1)
NU	10.78%	(26)	17.84%	(43)	**68.04%**	**(164)**	2.9%	(7)	0%	(0)	0.41%	(1)
CO	7.14%	(15)	2.85%	(6)	0.47%	(1)	**67.14%**	**(141)**	21.9%	(46)	0.47%	(1)
FI	13.94%	(29)	25.96%	(54)	0%	(0)	14.9%	(31)	**45.19%**	**(94)**	0%	(0)
CY	0%	(0)	0%	(0)	0.91%	(1)	10.09%	(11)	0%	(0)	**88.99%**	**(97)**

Slide level	Prediction											
	CE		HO		NU		CO		FI		CY	
CE	**100%**	**(6)**	0%	(0)	0%	(0)	0%	(0)	0%	(0)	0%	(0)
HO	0%	(0)	**100%**	**(5)**	0%	(0)	0%	(0)	0%	(0)	0%	(0)
NU	0%	(0)	0%	(0)	**100%**	**(4)**	0%	(0)	0%	(0)	0%	(0)
CO	20%	(1)	0%	(0)	0%	(0)	**60%**	**(3)**	20%	(1)	0%	(0)
FI	25%	(1)	25%	(1)	0%	(0)	0%	(0)	**50%**	**(2)**	0%	(0)
CY	0%	(0)	0%	(0)	0%	(0)	0%	(0)	0%	(0)	**100%**	**(4)**

CE: centromere; HO, homogeneous; NU: nucleolar; CO: coarse speckled; FI: fine speckled; CY: cytoplasmic.

Table 3.8: Percentage (numbers) of predicted staining patterns in each slide image (1/2).

Image#	Cell-level prediction												Slide-level prediction	Ground truth
	CE		HO		NU		CO		FI		CY			
1	0%	(0)	96.72%	(59)	0%	(0)	0%	(0)	3.27%	(2)	0%	(0)	HO	HO
2	**2.08%**	**(1)**	**50%**	**(24)**	**0%**	**(0)**	**12.5%**	**(6)**	**35.41%**	**(17)**	**0%**	**(0)**	**HO**	**FI**
3	98.87%	(88)	0%	(0)	0%	(0)	0%	(0)	0%	(0)	1.12%	(1)	CE	CE
4	7.57%	(5)	43.93%	(29)	46.96%	(31)	1.51%	(1)	0%	(0)	0%	(0)	NU	NU
5	2.12%	(1)	89.36%	(42)	4.25%	(2)	0%	(0)	4.25%	(2)	0%	(0)	HO	HO
6	2.94%	(2)	0%	(0)	0%	(0)	92.64%	(63)	4.41%	(3)	0%	(0)	CO	CO
7	66.07%	(37)	0%	(0)	14.28%	(8)	19.64%	(11)	0%	(0)	0%	(0)	CE	CE
8	33.92%	(19)	3.57%	(2)	53.57%	(30)	8.92%	(5)	0%	(0)	0%	(0)	NU	NU
9	0%	(0)	17.39%	(8)	0%	(0)	32.6%	(15)	50%	(23)	0%	(0)	FI	FI
10	**39.39%**	**(13)**	**9.09%**	**(3)**	**0%**	**(0)**	**33.33%**	**(11)**	**18.18%**	**(6)**	**0%**	**(0)**	**CE**	**CO**
11	0%	(0)	4.87%	(2)	2.43%	(1)	73.17%	(30)	19.51%	(8)	0%	(0)	CO	CO
12	0%	(0)	0%	(0)	0%	(0)	75.51%	(37)	22.44%	(11)	2.04%	(1)	CO	CO
13	95.65%	(44)	0%	(0)	0%	(0)	4.34%	(2)	0%	(0)	0%	(0)	CE	CE
14	39.68%	(25)	1.58%	(1)	3.17%	(2)	23.8%	(15)	31.74%	(20)	0%	(0)	CE	CE

CE: centromere; HO, homogeneous; NU: nucleolar; CO: coarse speckled; FI: fine speckled; CY: cytoplasmic.

Table 3.9: Percentage (numbers) of predicted staining patterns in each slide image (2/2).

Image#	Cell-level prediction												Slide-level prediction	Ground truth
	CE		HO		NU		CO		FI		CY			
15	**44.44%**	**(28)**	**14.28%**	**(9)**	**0%**	**(0)**	**4.76%**	**(3)**	**36.5%**	**(23)**	**0%**	**(0)**	**CE**	**FI**
16	94.73%	(36)	0%	(0)	2.63%	(1)	0%	(0)	2.63%	(1)	0%	(0)	CE	CE
17	**0%**	**(0)**	**5.26%**	**(1)**	**0%**	**(0)**	**0%**	**(0)**	**94.73%**	**(18)**	**0%**	**(0)**	**FI**	**CO**
18	0%	(0)	76.19%	(32)	0%	(0)	19.04%	(8)	2.38%	(1)	2.38%	(1)	HO	HO
19	93.84%	(61)	0%	(0)	6.15%	(4)	0%	(0)	0%	(0)	0%	(0)	CE	CE
20	4.34%	(2)	10.86%	(5)	82.6%	(38)	0%	(0)	0%	(0)	2.17%	(1)	NU	NU
21	4.91%	(3)	40.98%	(25)	18.03%	(11)	4.91%	(3)	31.14%	(19)	0%	(0)	HO	HO
22	0%	(0)	69.74%	(83)	20.16%	(24)	3.36%	(4)	6.725%	(8)	0%	(0)	HO	HO
23	0%	(0)	25.49%	(13)	0%	(0)	13.72%	(7)	60.78%	(31)	0%	(0)	FI	FI
24	0%	(0)	9.58%	(7)	89.04%	(65)	1.36%	(1)	0%	(0)	0%	(0)	NU	NU
25	0%	(0)	0%	(0)	4.16%	(1)	12.5%	(3)	0%	(0)	83.33%	(20)	CY	CY
26	0%	(0)	0%	(0)	0%	(0)	5.88%	(2)	0%	(0)	94.11%	(32)	CY	CY
27	0%	(0)	0%	(0)	0%	(0)	15.78%	(6)	0%	(0)	84.21%	(32)	CY	CY
28	0%	(0)	0%	(0)	0%	(0)	0%	(0)	0%	(0)	100%	(13)	CY	CY

CE: centromere; HO, homogeneous; NU: nucleolar; CO: coarse speckled; FI: fine speckled; CY: cytoplasmic.

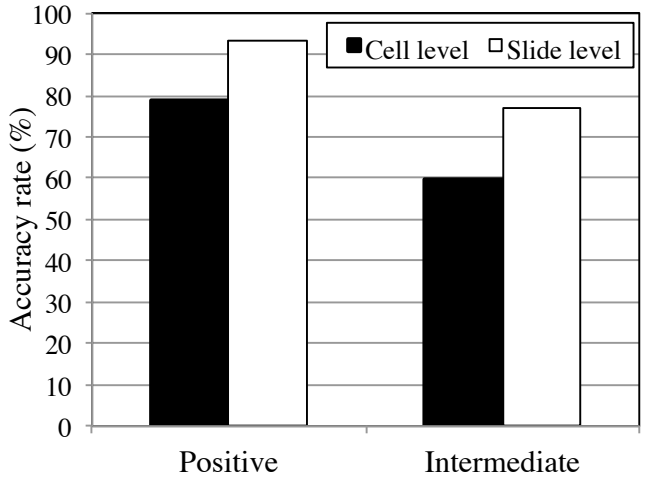


Figure 3.16: Accuracy rates for each intensity pattern.

3.4 Summary

In this chapter, we have focused on rotation invariance in LBP and have proposed RIC-LBP. In this method, rotation equivalent LBP pairs are assigned the same label, and a histogram of them are used as an image feature. RIC-LBP has rotation invariance, inheriting the advantages of CoALBP. We have confirmed the effectiveness of the RIC-LBP, *i.e.*, the combination of spatial co-occurrence and rotation invariance on texture recognition and HEp-2 cells classification. On the HEp-2 cells classification experiment, the effectiveness of the synthesis of training images by rotation has been confirmed.

Chapter 4

Oriented Bounding Box Estimation with Rotation Variant Feature

In this chapter, we propose orientation-aware regression for oriented bounding box estimation. Firstly, we describe the background of the oriented bounding box estimation and explain the property of the oriented bounding box. Next, we explain architectures and components of the oriented-aware regression. Finally, we show the experimental results of the bounding box estimation accuracy and the detection performance.

4.1 Introduction

The oriented bounding box is used in oriented object detection for the surveillance applications such as vehicle/ship detection from aerial images [48, 26, 27] and person detection with wearable/top-view cameras [49, 50]. The development of convolutional neural networks based detectors such as Faster R-CNN [24] and SSD [25] has led to an increasing amount of research into oriented object detection utilizing CNNs [26, 27, 51, 52, 28]. While a typical CNN detector outputs an axis-aligned bounding box as the object shape using a bounding box regressor, the detector for oriented objects outputs an oriented bounding box that approximates the object shape more precisely than the axis-aligned bounding box, it is tight-fit and reduces background clutter. Thus, the oriented bounding box is preferable for many applications.

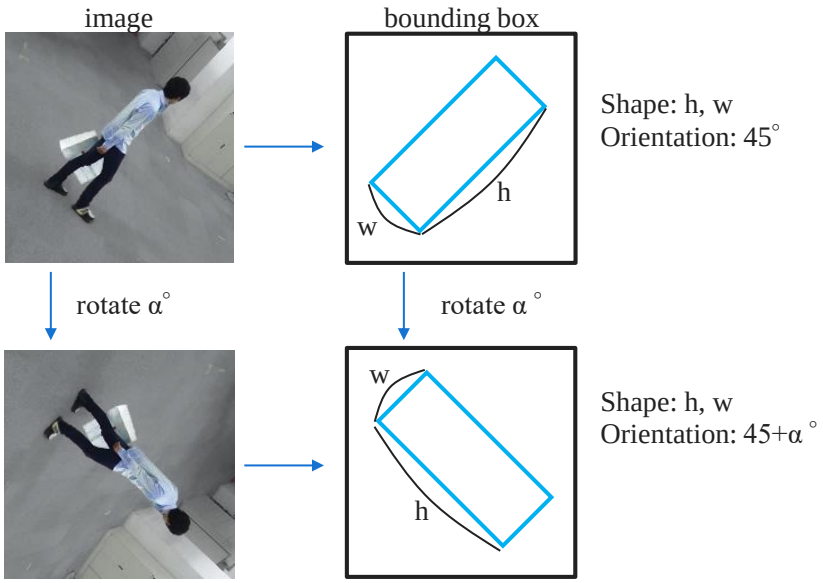


Figure 4.1: Rotation covariance of oriented bounding box.

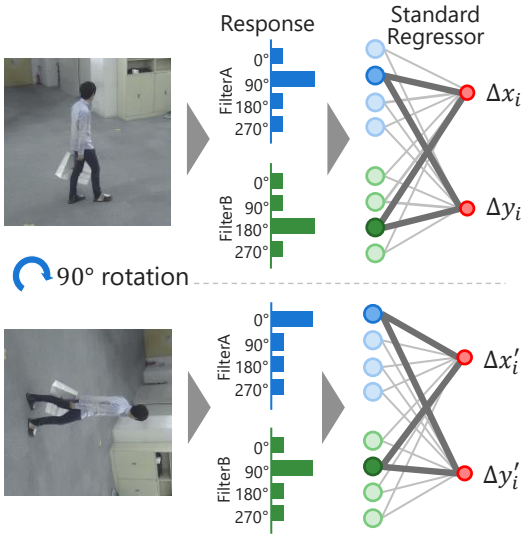
The oriented bounding box has a remarkable property about the relationship between the orientation of an object and one of a bounding box. More concretely, when an object rotates, then a bounding box of the object also rotates with the same angle; we call this property between the object and the bounding box to *rotation covariance*. As shown in Figure 4.1, shape (*i.e.*, width and height) is invariant to rotation. On the other hand, the orientation of the bounding box varies according to the rotation, *i.e.*, it is covariant to the rotation.

In oriented bounding box estimation, since the regressor needs to perceive the object's orientation, extracting the orientation that is correlated with the orientation of the object is critically important [28]. An efficient network extracting the oriented information is ORConv [53], which is an extension of convolutional layers which convolves with filters rotated by multiple angles. The responses from the rotated filters are rotation variant and are input as an oriented feature to the regressor.

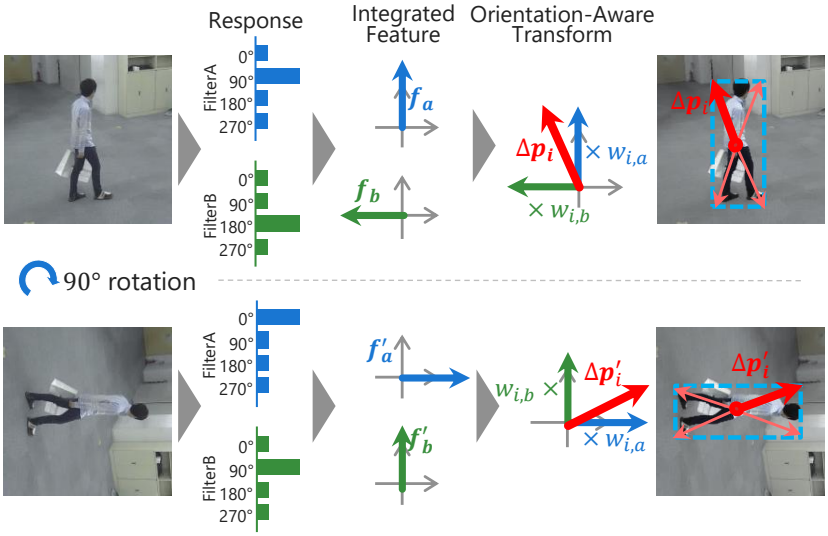
While a commonly used architecture for the bounding box regressor usually consists of linear layers (*i.e.*, full-connection layers, and convolutional layers) and ReLU activations, the standard regressor does not directly utilize the orientation of the oriented feature leading to degraded accuracy of the bounding box estimation. As a result, bounding box estimated by the standard regressor cannot be rotation covariant. We explain more detail using Figure 4.2. Consider an object

rotated by 90 degrees. The filter's responses obtained from the rotated object are matched with cyclic-shifted responses that are obtained from the non-rotated object as shown in Figure 4.2a, indicating that the object's orientation correlates with the orientation where the filter responds. However, the orientation associated with the filter's response, such as the filter rotation angle, is not propagated to the regressor and only the filter's response is input to the regressor. Therefore the regressor cannot directly learn the relationship between the response and object orientation. Moreover, as described in Section 4.3.4, even if the orientation of the response is input to the regressor, the standard regressor perturbs the orientation, and the resulting estimated bounding box becomes sensitive to rotation and is not rotation covariant.

In this study, we propose *Orientation-Aware Regression* for oriented bounding box estimation, which conserves the orientation of the response to the output. This method integrates the response and orientation as a feature, then performs orientation-aware transforms through which the orientation of the integrated feature is maintained. As shown in Figure 4.2b, when the object is rotated by some angle, the integrated feature is also rotated by the same angle. The orientation-aware transform also has a similar property that when the object is rotated, the output of the transform is also rotated by the same angle. As a result, the estimated bounding box becomes rotation covariant. This improves the precision of the estimated bounding box and robustness against object rotation.



(a) Standard regression



(b) Orientation-aware regression

Figure 4.2: Comparison of standard regression and orientation-aware regression. Angles under "Response" indicate the filter rotation angle. (a) The standard regressor must learn the relationships between the response and orientation for each object, since the response is dependent on the orientation. (b) Use of the integrated feature allows the regressor to output an oriented bounding box since the integrated feature can maintain the orientation information.

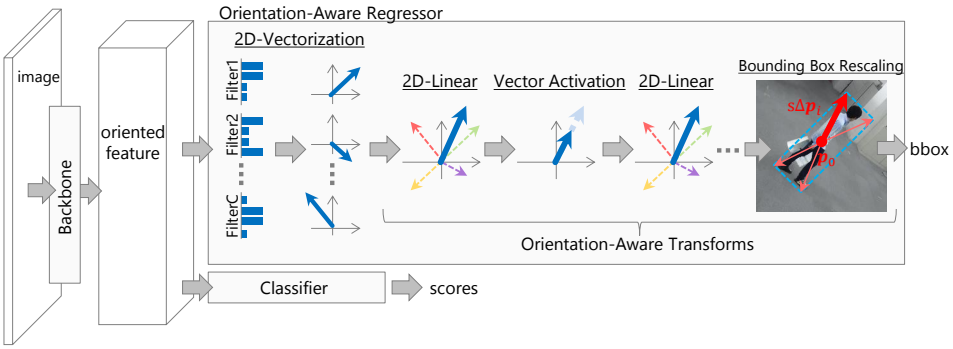


Figure 4.3: Overview of detector with orientation-aware regression.

4.2 Related work

Bounding box regression Felzenszwalb *et al.* [54] first proposed the bounding box regression for object detection. They used linear least-squares regression for bounding box estimation for improved detection performance. Based on this work, CNN-based detectors such as Faster R-CNN [24] and SSD [25] estimate the bounding box using a regressor that consists of linear layers and ReLU activations. Recent detectors for oriented objects [26, 27, 28] take over the regressor and estimate the oriented bounding box using a similar regressor.

Oriented feature by neural networks There are some studies for dealing with rotation for object classification [55, 56, 53, 57, 58, 59, 60, 61]. For example, Laptev *et al.* proposed TI-Pooling [55], which rotates an input image by multiple angles and summarizes the features extracted from the rotated images. Worrall *et al.* [61] rewrote convolution with circular harmonics and handled rotation in complex space. In this study, we employ ORConv [53], which is an extension of the convolutional layer which encodes the orientation to which the filter responds. In particular, ORConv rotates convolutional filters by multiple angles and produces the response of the rotated filters. Due to its small filter size (*i.e.*, 3×3), ORConv can perform with reasonable amounts of computation and be integrated into modern architectures [62, 63, 64, 65].

4.3 Orientation-aware regression

4.3.1 Overview

An overview of the detector is shown in Figure 4.3. The detector consists of three components: 1) backbone 2) classifier and 3) the orientation-aware regressor. The backbone extracts an oriented feature from an input image using convolutional layers. The oriented feature is used to predict class-wise scores by the classifier and the four vertices of the bounding box by the regressor. The regressor consists of 2D-Vectorization and orientation-aware transforms (2D-Linear, vector activation, and bounding box rescaling). The regressor uses multiple 2D-Linear and vector activation similar to standard regression.

4.3.2 Oriented feature extraction

To extract the oriented feature, the backbone uses ORConv as the convolutional layer. Let $h_{c,k}$ and $h'_{c,k}$ denote the features of the input and output of the ORConv layer respectively, corresponding to the c-th channel and the k-th orientation. The output feature is computed as

$$h'_{c_o,k_o} = \sum_{c_i,k_i} F^{c_o,\theta_{k_o}}_{c_i,k_i} * h_{c_i,k_i}, \tag{4.1}$$

where $*$ is the convolution operator and $F^{c_o,k_o}_{c_i,k_i}$ is the c_o-th filter rotated by $\theta_{k_o} = 360k_o/K$ degrees. In practice, K is set to 8.

Figure 4.4 shows exmaples of feature maps extracted with a network with ORConv layers. As we can see, it is clearly shown that the ORConv conducts for each specific orientation. The extracted features are rotated and shifted according to the orientation of the input image.

As mentioned in Section 4.2, ORConv can be applied to various architectures. In our experiments, we used an architecture similar to wide-ResNet [66] as the backbone.

4.3.3 2D-vectorization

The angle θ_{k_o} for filter rotation is not included explicitly in the feature h'_{c_o,k_o}, even though this angle is valuable as information about the orientation of the feature. To

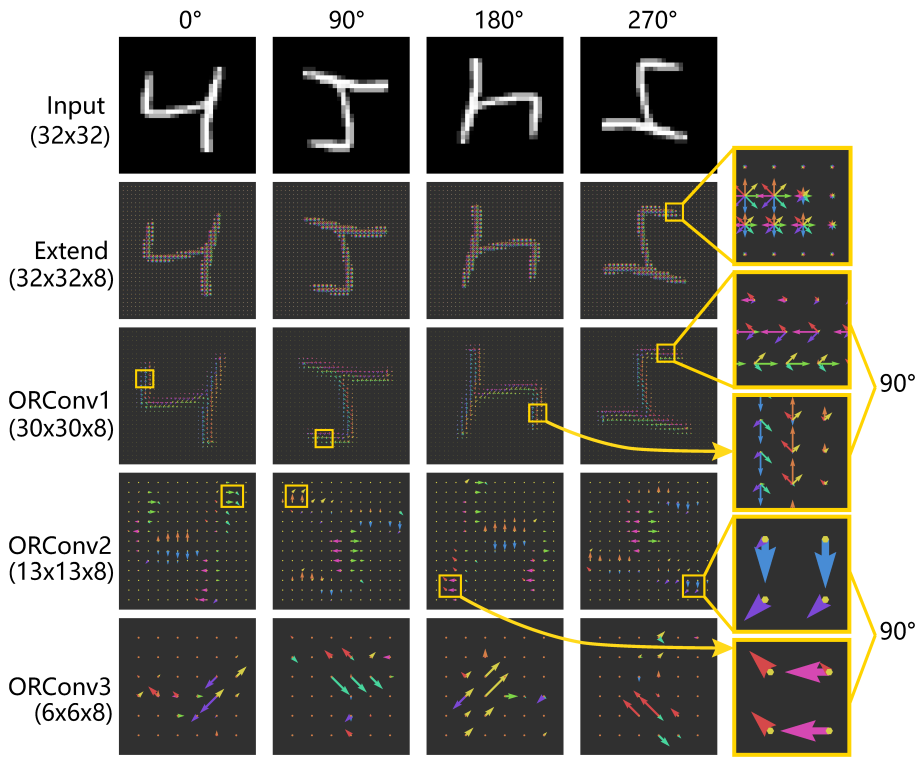


Figure 4.4: Example feature maps extracted by a network with ORConv layers trained on the rotated MNIST dataset. The orieginal of this figure is in [53].

utilize this angle, the oriented feature is converted into a 2D-vector form as follows:

$$\tilde{\boldsymbol{h}}_{c,k} = (h_{c,k}\cos(\theta_k), h_{c,k}\sin(\theta_k)), \tag{4.2}$$

where $\theta_k = 360k/K$ degrees. The 2D-vectors are summarized along the orientation axis. For simplicity, we use average pooling as:

$$\boldsymbol{h}_c = \frac{1}{K}\sum_{k=0}^{K-1}\tilde{\boldsymbol{h}}_{c,k}. \tag{4.3}$$

The orientation of $\boldsymbol{h}_c$ is consistent with the orientation in which the filter exhibits a strong response. Thus when the object is rotated, the orientation of $\boldsymbol{h}_c$ also rotated.

4.3.4 Orientation-aware transforms

To use the 2D-vector effectively, the regressor consists of transforms which conserve the orientation of the vector. One of these transforms is a function that multiplies each element of the input with the same coefficient. The transforms described below (2D-Linear, vector activation, and bounding box rescaling) are based on this function and, as a result, the orientation of the oriented feature is propagated to output effectively.

2D-Linear In contrast to the standard linear function (*e.g.*, full-connection layer, and convolutional layer) which scales each element with different parameters perturbing its orientation, we scale each element with the same coefficient. We use a restricted linear-combination as follows:

$$\boldsymbol{h}_{c_o} = \sum_{c_i} w_{c_o,c_i}\boldsymbol{h}_{c_i}, \tag{4.4}$$

where $\boldsymbol{h}_{c_i}$ and $\boldsymbol{h}_{c_o}$ are input and output 2D-vectors respectively, and w_{c_o,c_i} is a learnable parameter. Since each element is scaled with the same coefficient, the orientation is maintained, allowing $\boldsymbol{h}_{c_o}$ to rotate as the object rotates.

Vector activation For an activation function conserving the orientation of inputs, we use a norm-based activation function that normalizes the input by employing "squashing" [67] defined as:

$$\text{squashing}(\boldsymbol{h}_c) = \frac{\|\boldsymbol{h}_c\|^2}{1+\|\boldsymbol{h}_c\|^2}\frac{\boldsymbol{h}_c}{\|\boldsymbol{h}_c\|}. \tag{4.5}$$

This activation change only the norm of the input vector and keeps the orientation of the input vector.

Bounding box rescaling and estimation The regressor outputs four displacements from the reference point to each vertex of the oriented bounding box. The displacements $\Delta \boldsymbol{p}_i = (\Delta x_i, \Delta y_i)$, $(i = 1, 2, 3, 4)$ is converted as:

$$\boldsymbol{p}_i = \boldsymbol{p}_o + s\Delta \boldsymbol{p}_i, \tag{4.6}$$

where $\boldsymbol{p}_i$ is location of i-th vertex of the estimated bounding box, $\boldsymbol{p}_o$ is reference point location and s is a predefined scale parameter that is set according to the standard size of the target object. In our experiments, s was set to 16. The orientation-aware regression estimates Δx_i and Δy_i together and the bounding box rescaling multiplies them with same scale parameter, whereas the standard regression estimates them independently and rescales them with different parameters [24]. Therefore, the bounding box estimated by the orientation-aware regression is rotation covariant.

4.4 Experimental evaluation

Evaluations on the person detection task with a sliding window were conducted. In this experiment, an input image is a cropped image, and target outputs are a score for person/non-person classification, and a bounding box of the person. The proposed method was evaluated on the bounding box estimation performance and the detection performance.

4.4.1 Settings

Dataset

For the evaluation, a subset of the COCO dataset [68] was used. First, the images were cropped using ground-truth bounding boxes of all categories with large margins and the object in each image was centered. These cropped images were then randomly rotated and re-cropped. Then the re-cropped images are resized to 64×64 pixels. Finally, the re-cropped and resized images were categorized into two classes, positive person images, and negative non-person images. The datasets for training and testing were constructed from the train-2014 set and val-2014 set,

respectively.[1] For training, 3,951 images were used for positive training and 38,291 images for negative training. For testing, 1,705 and 20,011 images were used for positive and negative testing, respectively. For data-augmentation during training, the images were randomly flipped and shifted.

Network architecture

The network used in the experiments is composed of a backbone, a classifier, and a bounding box regressor. To evaluate the proposed method, we used the four architectures as shown in Figure 4.5. For the backbone, we employed a wide-ResNet [66] based network, composed of a convolutional layer and four res-blocks. The number of channels of each convolutional layer was parameterized with the widening factor α. For the Conv architecture (Figure 4.5a), $\alpha = 4$ and for the architectures (Figure 4.5b-d), $\alpha = 0.5$. The ORConv architectures utilized the OR-Align [53] in the classifier, which transforms the input features using a SIFT-like operation for rotation-invariance. Architecture (d) outputs four vectors as an oriented bounding box. Since Architectures (a)-(c) estimate Δx_i and Δy_i independently, these regressors output eight values.

Cross-entropy loss was used for the classifier. The sum of the two losses was optimized, momentum SGD is used with an iteration of 200 epochs as the optimizer. The learning rate was initially set to 0.02 and was decayed by 0.5 at 60, 120 and 160 epochs. Momentum and weight decay are 0.9 and 0.0005, respectively. The mini-batch size is 128. Since the dataset was unbalanced, a mini-batch was constructed by sampling equally for each class.

Evaluation metric

As a metric for evaluation, we use "Oriented IoU" (OIoU), which measures the accuracy of the bounding box region and the accuracy of the orientation. OIoU is defined as:

$$\mathrm{OIoU} = \mathrm{IoU}(B_p, B_g)\mathrm{AoO}(\boldsymbol{n}_p, \boldsymbol{n}_g) \tag{4.7}$$

$$\mathrm{IoU}(B_p, B_g) = \frac{\mathrm{Area}(B_p \cap B_g)}{\mathrm{Area}(B_p \cup B_g)} \tag{4.8}$$

$$\mathrm{AoO}(\boldsymbol{n}_p, \boldsymbol{n}_g) = \max\left(0, \boldsymbol{n}_p \cdot \boldsymbol{n}_g\right), \tag{4.9}$$

[1] All images used are licensed CC-BY-2.0, CC-BY-SA-2.0 or CC-BY-ND-2.0.

where B_p and B_g are the predicted and the ground-truth bounding boxes respectively, and $\boldsymbol{n}_p$ and $\boldsymbol{n}_g$ are the orientations of the predicted and ground-truth bounding boxes, respectively. The orientation is calculated from the corner vertices of the bounding box, a normalized vector from the center point of the bottom of the bounding box to the center point at the top of the bounding box. In this experiment, the threshold of OIoU for judging whether the estimation result is correct or not is 0.5.

4.4.2 Results

Bounding box regression performance

The results of the fundamental performance of the oriented bounding box regression without classification are shown in Figure 4.6. The results for Architecture (a) and Architecture (b) are similar to standard regression. When 2D-vectorization is added to standard regression (Architecture (c)), the recall becomes worse since the standard linear layer and activation are not suitable for 2D-Vector. Concretely, ReLU included in the standard regression truncates a negative value to zero and changes the orientation of the 2D-vectors drastically. On the other hand, using orientation-aware regression (Architecture (d)) drastically improved recall and outperformed the other architectures, indicating that the combination of 2D-vectorization and the orientation-aware transform is crucial.

Figure 4.7 and Table 4.1 show the distribution and statistics of standard deviation of OIoU. Here, the standard deviation of OIoU is calculated per image from an image rotated randomly, and the distribution and statistics are obtained from the standard deviations of all images in the dataset. The standard deviation of Architectures (b)-(d) are lower than Architecture (a), indicating that ORConv stabilizes the estimated bounding box. Furthermore, the lowest standard deviation is Architecture (d), showing that the proposed method improves not only the accuracy of the bounding box but also the robustness against rotation. This is due to the efficient propagation of the orientation of the oriented feature to output. Examples of the estimation results and robustness are shown in Figures 4.8, 4.9, 4.10. It is clear that the bounding box estimated by Architecture (a) varies by image rotation and is inconsistent and inaccurate. On the other hand, the bounding box estimated by Architecture (d) was more robust against image rotation and more accurate.

Table 4.1: Average of standard deviation of OIoU when the input image is rotated. Standard deviation is calculated per image.

Architecture	Avg. of Std.
(a) Conv+Standard Reg.	0.0930
(b) ORConv+Standard Reg.	0.0786
(c) ORConv+2DVec+Standard Reg.	0.0894
(d) ORConv+Ori.-Aware Reg.(ours)	**0.0704**

Detection performance

Detection performance was evaluated using a sliding window for both regression and classification. The result for each architecture is shown in Figure 4.11. The results are similar to the previous experiments, showing that improving the accuracy of the bounding box leads to improvements in detection performance, and also argues for the effectiveness of the proposed method.

4.5 Summary

In this chapter, we have focused on rotation variant feature by CNN and rotation covariance, which is a rotation property between an object and an oriented bounding box, and have proposed Orientation-Aware Regression (OAR) for oriented bounding box estimation using CNN. OAR consists of 2D-vectorization and orientation-aware transforms, converting the oriented feature to the oriented bounding box effectively. We experimentally confirmed the bounding box estimated by OAR becomes more accurate and robust against rotation and improves detection performance.

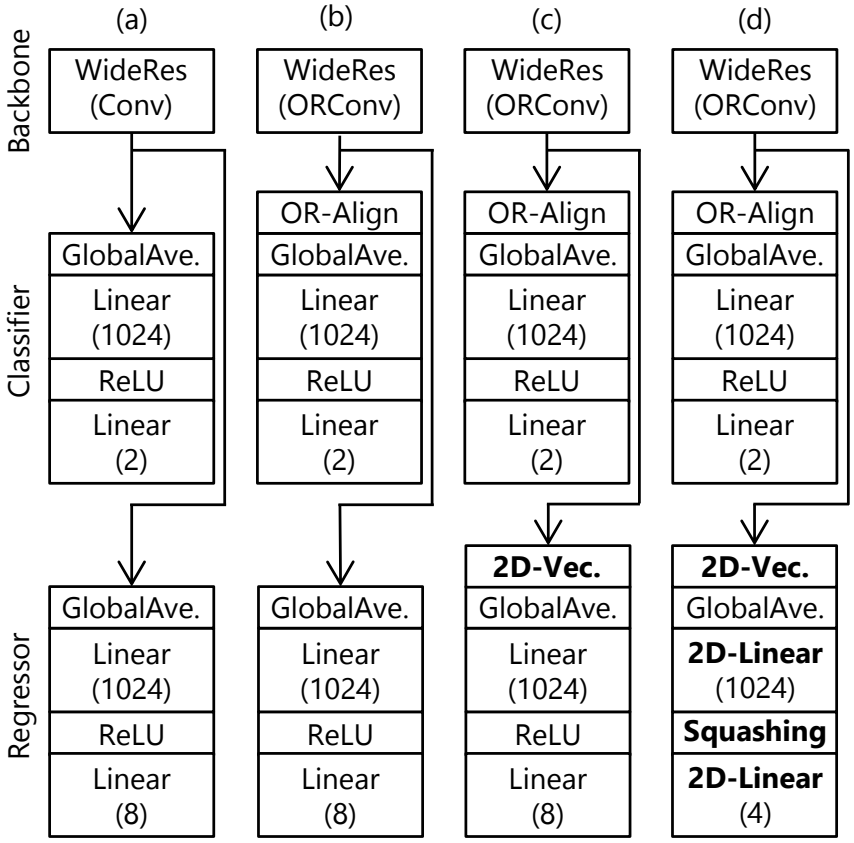


Figure 4.5: Architectures used in the bounding box evaluation experiments. "GlobalAve." is global-average pooling. The number in the bracket indicates the number of output channels. Architecture (a) uses standard convolution layer in the backbone and its regressor is standard regression. Architecture (b)-(d) uses ORConv in the backbone. Architecture (b) uses a standard regression. The regressor of Architecture (c) uses standard regression added 2D-vectorization. Architecture (d) uses rotation-aware regression. The bounding box scaling is not shown.

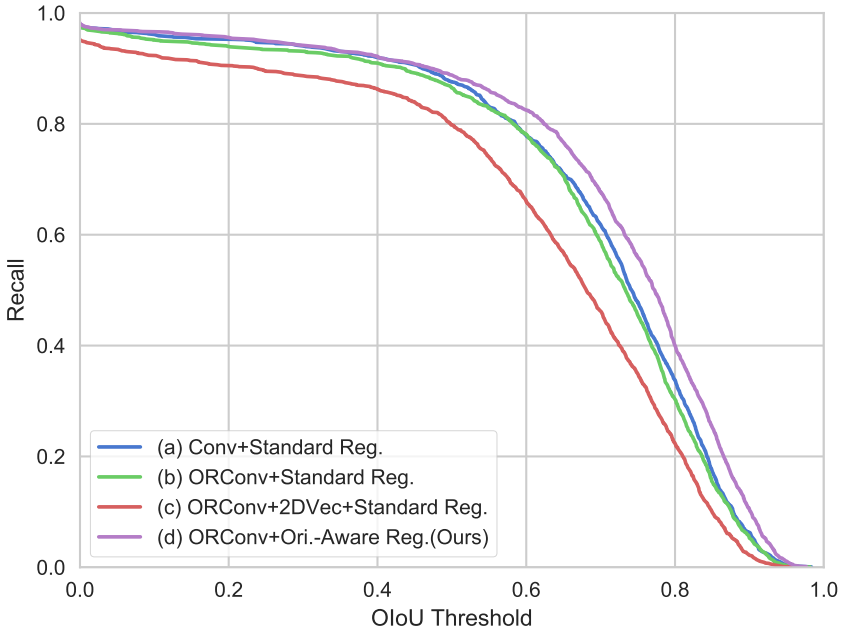


Figure 4.6: OIoU threshold and recall curves.

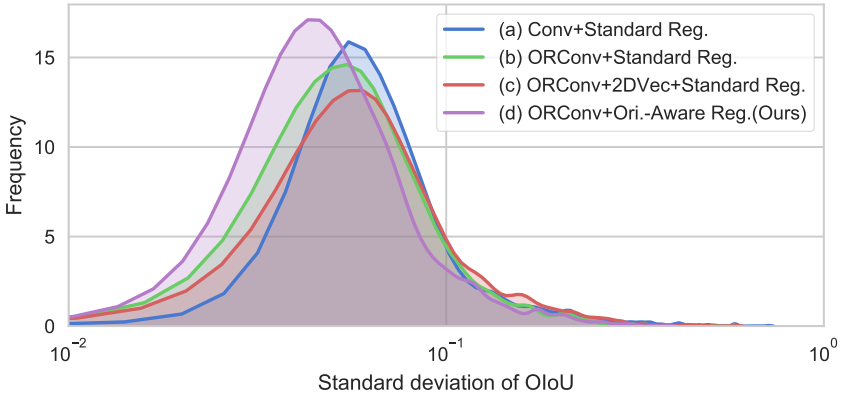


Figure 4.7: Distribution of standard deviation of OIoU when the input image is rotated. Standard deviation is calculated per image.

Figure 4.8: Estimation results of the bounding box regression (1/3). Upper images are the results of the proposed method, Architecture (d). Lower images are results of Architecture (a). Green box denotes true positive. Red box denotes false positive. Blue dash box denotes ground-truth box. Orange dot denotes the upper left corner of the bounding box. Threshold of OIoU is 0.5. The original images are from the website: https://www.flickr.com/photos/impuls-f/2434568700.

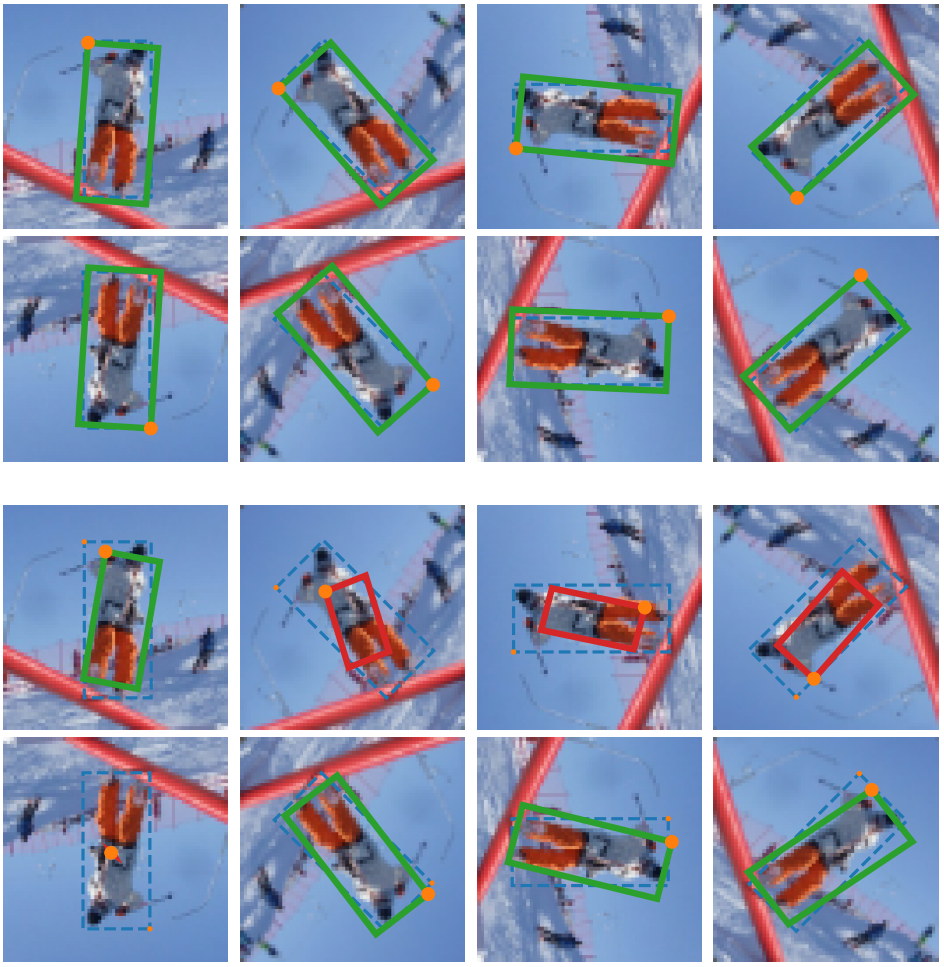


Figure 4.9: Estimation results of the bounding box regression (2/3). Upper images are the results of the proposed method, Architecture (d). Lower images are results of Architecture (a). Green box denotes true positive. Red box denotes false positive. Blue dash box denotes ground-truth box. Orange dot denotes the upper left corner of the bounding box. Threshold of OIoU is 0.5. The original images are from the following websites: https://www.flickr.com/photos/badkleinkirchheim/6856285019.

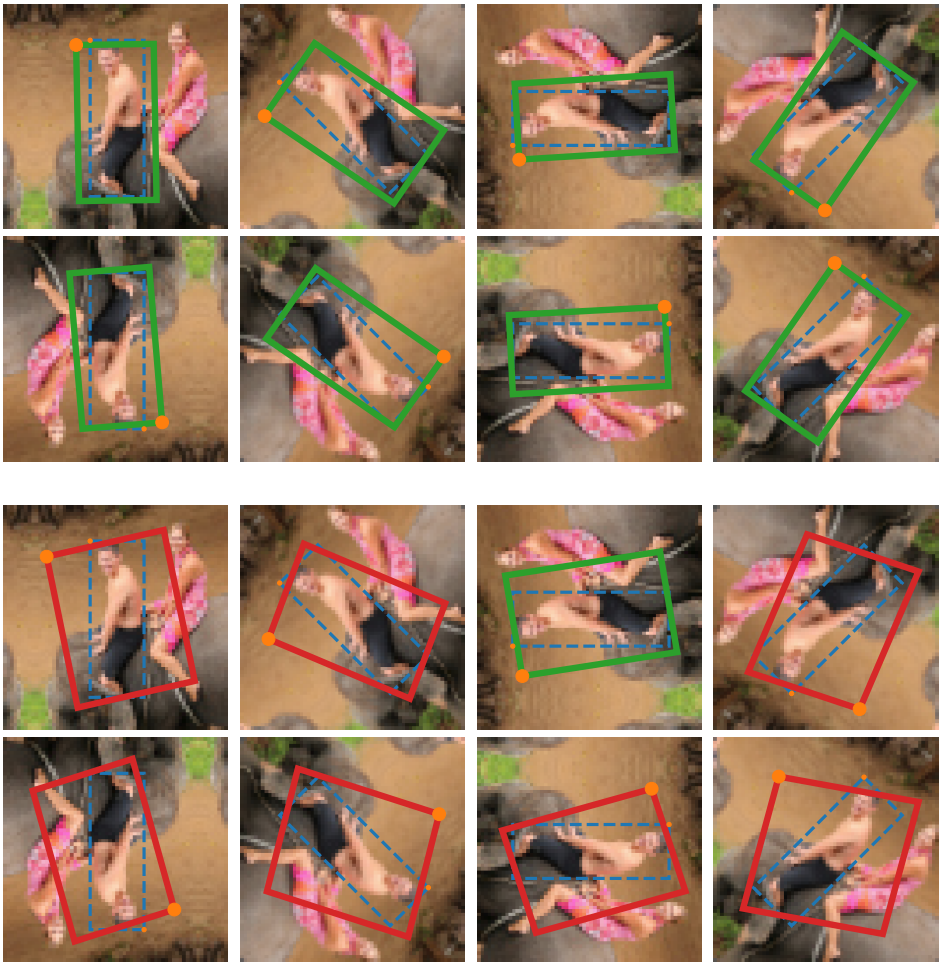


Figure 4.10: Estimation results of the bounding box regression (3/3). Upper images are the results of the proposed method, Architecture (d). Lower images are results of Architecture (a). Green box denotes true positive. Red box denotes false positive. Blue dash box denotes ground-truth box. Orange dot denotes the upper left corner of the bounding box. Threshold of OIoU is 0.5. The original images are from the following websites: https://www.flickr.com/photos/tedmurphy/6132078023.

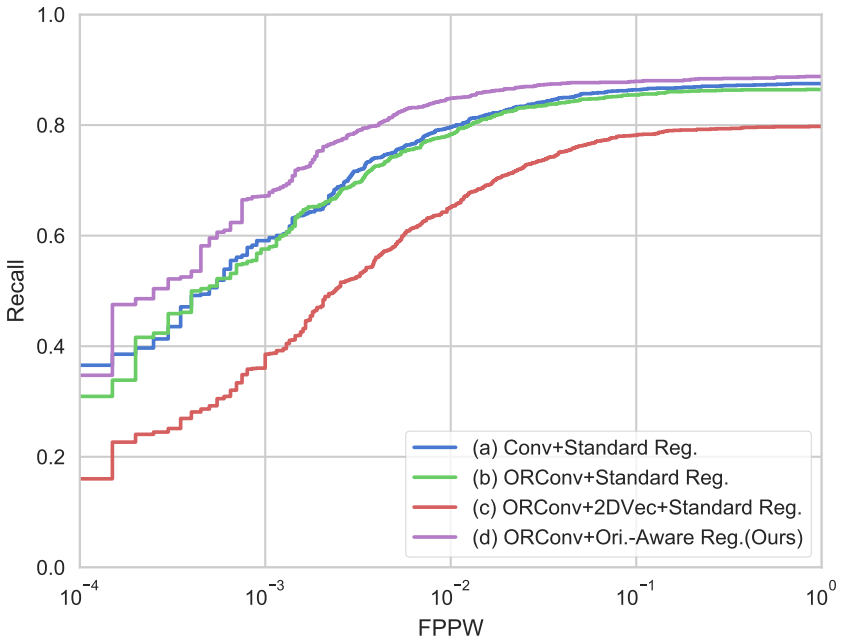


Figure 4.11: FalsePositivesPerWindow(FPPW)-Recall curves. Threshold of OIoU is 0.5.

Chapter 5

Rotation Invariance Switcher based on Rotatability

In this chapter, we propose a novel switching module based on the rotatability of objects, named rotation invariance switcher (RIS).

Firstly, we explain rotatability, showing some examples. Then, we explain the concept of the proposed method RIS and architectures for CNN, and extensions of it. Finally, we show the experimental results for classification performance and analysis of the behavior of the RIS.

5.1 Introduction

As described in Chapter 3, rotation invariance is a key feature for image recognition of objects with fully unconstrained orientation such as object detection and semantic segmentation of aerial images [51], texture classification [14, 69], scene text detection [28], rotated face detection [70], and the segmentation of medical images [71].

In this thesis, we denote objects which are unconstrained in their 3D orientation as *rotatable* objects. For example, “airplanes” and “cutleries” can appear in many different orientations (Fig. 5.1a). Features extracted from rotatable objects are problematic in that they have large variations due to the multitude of possible orientations. To suppress the number of variations, a feature is usually converted into a rotation invariant feature using rotation invariance transforms such as SIFT-like invariance [53] and max-pooling invariance [55].

On the other hand, some objects are naturally constrained in an upright posi-

(a) rotatable

(b) non-rotatable

Figure 5.1: Examples of rotatable and non-rotatable objects. Rotatable objects are unconstrained in their 3D orientation, while non-rotatable objects are constrained in an upright position. The image IDs in the MS-COCO dataset [68] for each image are following (from left to right): (a) 360701, 231862, 143258, 298276. (b) 577835, 246688, 116834, 157618.

tion, and are denoted in this thesis as *non-rotatable* objects. "Trucks" and "boats" are constrained in a plane due to the ground or water surface (Fig. 5.1b). Features extracted from these objects have lower variations compared to rotatable objects, and rotation invariance transforms do not substantially decrease this variation. Moreover, the orientation of non-rotatable objects can be used as a discriminative feature, which invariably is lost through the use of rotation invariance transforms.

In most image recognition systems, both rotatable and non-rotatable objects are commonly included as targets for classification. A priori selection in the use of rotation invariance will invariably lead to degraded prediction accuracy for either the rotatable or non-rotatable object, depending on the which mode was selected. Therefore the use of rotation invariance cannot be chosen in advance but must be selected adaptively.

In this study, we propose *RIS,* a *rotation invariance switcher* which learns about object rotatability. RIS estimates the rotatability of the features in an input image and dynamically switches between using rotation invariance or rotation variance based on a mechanism similar to the soft attention module. We develop four RIS

modules which switch between different axes, and multiple modules are applied to different network layers, estimating the rotatability at different scales. Rotatability is learned without explicit labels of rotatability and RIS enables the observation of rotatability from different viewpoints.

5.2 Related work

5.2.1 Rotation invariance with deep neural networks

As discussed in Chapter 3, there has been much research conducted in rotation invariance, such as SIFT [7] and rotation invariant LBP [14, 15]. Rotation invariance has also been the subject of intense study after the development of convolutional neural networks [55, 53, 61, 69]. TI-Pooling [55] rotates an input image by multiple angles and summarizes the features extracted from the rotated images using max-pooling along the rotation axis. Worrall *et al.* [61] rewrote convolution with circular harmonics and handled rotation in complex space. Oriented response networks [53] obtain the oriented responses by rotating filters on each convolutional layer and accumulates (OR-Pooling) or aligns (OR-Align) the outputs of the last convolutional layer.

5.2.2 Soft attention for image recognition

The design of our proposed method is inspired by the soft attention module employed in [65, 72, 73, 74, 75]. Srivastava *et al.* [72] used soft attention to regulate the shortcut path, enabling the training of very deep networks. Wang *et al.* [73] proposed the attention module using the hourglass module, inspired by advances in semantic segmentation. SENet [65] gates channels softly by weighting the feature map along the channel axis using an attention-based module. Jetley *et al.* [74] proposed an attention module that learns a 2D-attention map under a convex constraint. Chen *et al.* [75] estimated the scale of objects using a scale-oriented attention map and merged the results of multi-scale images for semantic segmentation.

5.3 Rotation invariance switcher

In this section, we present a novel switching module based on the rotatability of objects, named rotation invariance switcher. In this section we introduce a new

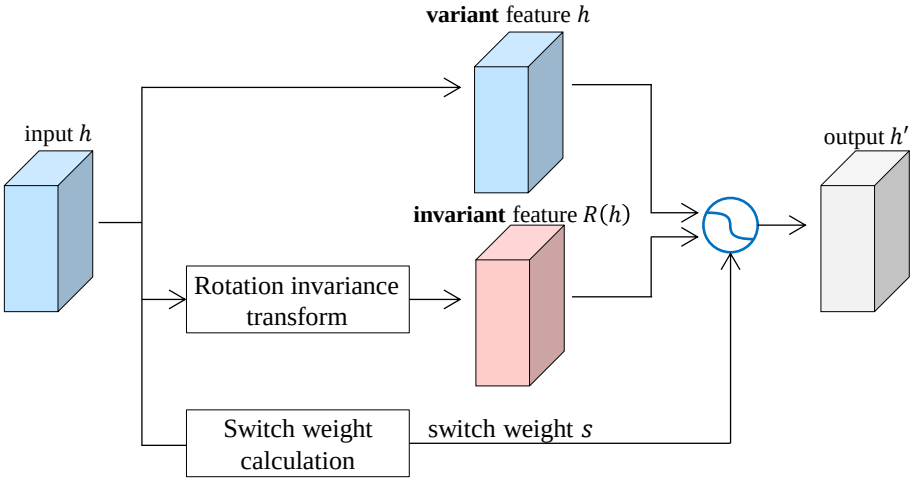


Figure 5.2: RIS overview.

switching module based on the rotatability of the object, called the rotation invariant switcher. First we will explain rotatability and give some examples. Next, we explain the concept of the proposed method RIS and implementation of various variations.

5.3.1 CNN with rotation invariance

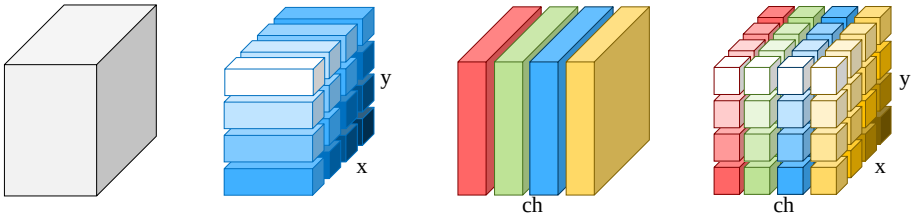
To efficiently handle rotation, we employ ORConv [53] explained in Section 4.3.2, an extension of the convolutional layer which encodes the orientation to which a filter responds. As an image rotates, the angle at which a filter strongly responds also changes. That is, ORConv features are rotation variant. Rotation invariance transforms the feature into a rotation invariant feature.

For simplicity, we employ SIFT-like invariance that cyclic-shifts input features along the orientation axis with the maximum element as the origin. This invariance keeps the feature dimension, while others (*e.g.*, max-pooling based invariance) reduce the feature dimension with the resulting loss of order.

5.3.2 Rotation invariance switch

RIS adaptively switches rotation variant features h and rotation invariant features $\mathrm{R}(h)$ as shown in Fig. 5.2. The switching mechanism is achieved with an attention-like architecture, namely:

$$h' = (1 - s)h + s\mathrm{R}(h), \tag{5.1}$$



(a) image-wise (b) position-wise (c) channel-wise (d) position/channel-wise

Figure 5.3: RIS switcher axes.

where switch weight $s = \mathrm{S}(h; w)$ and $\mathrm{S}(\cdot)$ is a function which calculates switch weight s with learnable parameter w. Implementation of them is presented in the following sections.

The input determines the switch weight s. When the input is expected to be non-rotatable, the switch weight is small, and the output is close to the rotation variant feature h. On the other hand, when the input is expected to rotatable, the switch weight is large, and the output is close to the rotation invariant feature $\mathrm{R}(h)$. Thus the switch weight is crucial as it allows direct insight into the rotatability of the input.

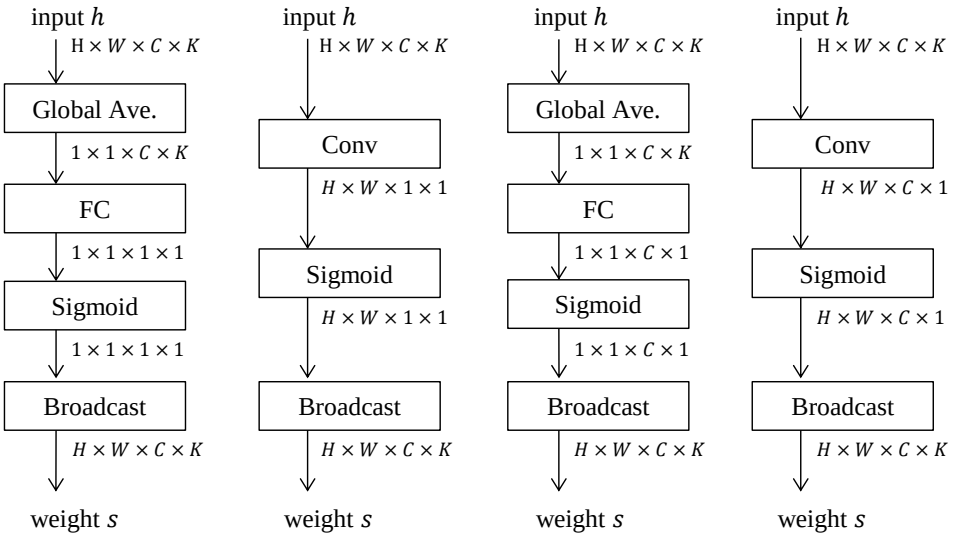
RIS can also be considered as a rotatability classifier, even though RIS is trained without an explicit rotatability label. That is, RIS learns the rotatability inherent in the input from the training data in an unsupervised manner.

5.3.3 Use of different axes

RIS can utilize different axes for switching to observe rotatability from different viewpoints. The most straightforward axis is to switch per image, but RIS can also switch on a much granular level such as a position axis, channel axis and a combination of both position and channel axes (Fig. 5.3).

The image-wise switcher (Fig. 5.3a) calculates a weight for the entire image, which is shared throughout the entire feature map. The position-wise switcher (Fig. 5.3b) calculates a weight for each position of the input feature map, sharing the weight along the channel axis. The channel-wise switcher (Fig. 5.3c) calculates a weight for each channel of the input feature map, sharing the weight among all positions. The position/channel-wise switcher (Fig. 5.3d) is a combined position and channel-wise switcher and calculates a weight for each position and channel.

The switch weight along each axis indicates the rotatability from different viewpoints. For the image-wise switcher, the switch weight indicates the rotata-



(a) image-wise (b) position-wise (c) channel-wise (d) pos/ch-wise

Figure 5.4: Architectures to calculate switch weight. The output size of each step is denoted under each block. "Global Ave." indicates a global average pooling layer. "FC" indicates a full-connection layer. "Broadcast" expands the input to target shape by repeating elements of the input.

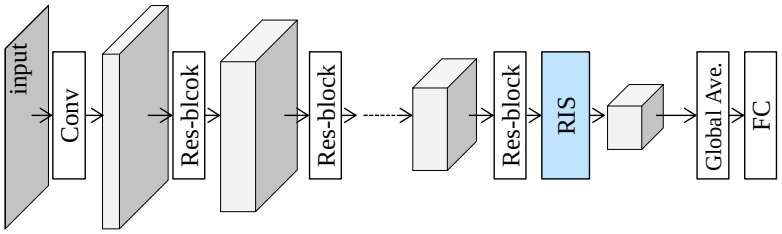
bility of the entire image. For the position-wise and channel-wise switchers, the switch weight indicates the local rotatability and channel rotatability, respectively.

The networks of $\mathrm{S}(\cdot)$ to calculate the switch weight for each axis are shown in Fig. 5.4. The image and channel-wise switchers compress the feature map using global average pooling and then perform full-connection. The position-wise and position/channel-wise switchers calculate the switch weight convolutionally to obtain a weight for each position.

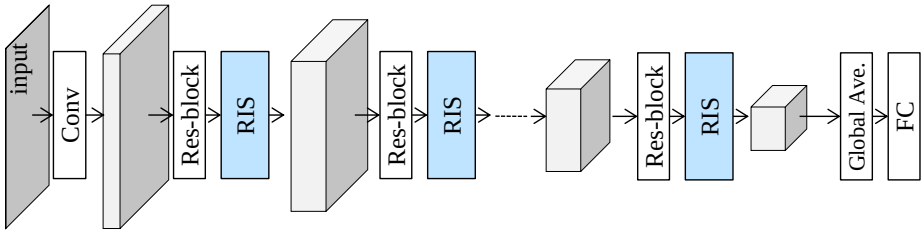
5.3.4 Multiple RIS modules

RIS is incorporated into a network consisted of multiple convolutional or res-block layers. As shown in Fig. 5.5a, RIS can be situated after the final res-block in a manner similar [53]. Moreover, since RIS adaptively switches depending on the input, RIS can be situated every after res-block (Fig. 5.5b).

As mentioned above, the rotatability depends on the input intrinsically. This can also be considered that the rotatability and the category of the input depend on each other. Thus the single RIS architecture (Fig. 5.5a) estimates the rotatability at the layer close to the output, showing the global rotatability. On the other hand, the



(a) single RIS



(b) multiple RIS

Figure 5.5: Network architectures incorporating RIS. (a) RIS is situated after the final res-block. (b) RIS is situated after each res-block. "Global Ave." indicates a global average pooling layer. "FC" indicates a full-connection layer.

multiple RIS architectures (Fig. 5.5b) can be regarded as dividing the rotatability estimation problem at different scales and progressively estimating the rotatability through all the RIS modules. Thus the multiple RIS architectures enable to handle the local and global rotatability.

5.4 Experimental evaluation

Evaluations of RIS were conducted on two types of tasks, object classification and person/non-person classification.

5.4.1 Object classification

Setting

The CIFAR-10 and CIFAR-100 datasets [76] were used to evaluate object classification. The CIFAR-10 dataset consists of 60,000 32×32 images in 10 categories. From each category, 5,000 images were used for training and 1,000 images for testing. Similarly, the CIFAR-100 dataset consists of 60,000 32×32 images in 100 categories and 500 and 100 images from each class were used for training and testing, respectively.

Table 5.1: Error ratios [%] on the CIFAR-10 and CIFAR-100 datasets. Rotation invariance is abbreviated as "RI."

Model		CIFAR-10	CIFAR-100
w/o RIS	non RI	6.06	28.18
	RI	6.07	28.22
RIS	image-wise	**5.75**	**27.96**
	position-wise	**5.77**	**27.59**
	channel-wise	**5.87**	**27.63**
	pos/ch-wise	**5.86**	**27.75**

The network was a wide-ResNet [66]-based network consisting of one convolutional layer (*conv1*), three res-blocks (*res2, res3, res4*), RIS, global average polling and full-connection, corresponding to the single RIS architecture shown in Fig. 5.5a. For comparison, two other types of networks were used. One is a network without both RIS and rotation invariance, in that the output of res4 is directly used for classification. The other network uses rotation invariance instead of RIS, and the output of res4 is transformed to a rotation invariant feature with using the transform in Sec. 5.3.1.

Training was done in a similar manner to [53]. As a loss for training, the cross-entropy loss was used. Momentum SGD was used with an iteration of 200 epochs as the optimizer. The learning rate was initially set to 0.02 and was decayed by 0.5 at 60, 120 and 160 epochs. Momentum and weight decay were 0.9 and 0.0005, respectively. The mini-batch size was 128. For data-augmentation during training, the images were randomly flipped and shifted.

Results

Table 5.1 shows the error ratios on the CIFAR-10/CIFAR-100 datasets. This result shows that all the models with RIS outperform the models without RIS, demonstrating the effectiveness of RIS.

Next, the relationship between RIS switch weight and input category was investigated. Fig. 5.6 shows the switch weight distribution of each channel. As expected, channels with large weights tend to be found on the rotatable categories such as airplanes, cats and birds. For non-rotatable categories (*e.g.*, horses, ships, trucks), channels with large weights tend to be fewer, indicating that the switcher captures the rotatability of the input and that the rotatability can be observed through the switch weight.

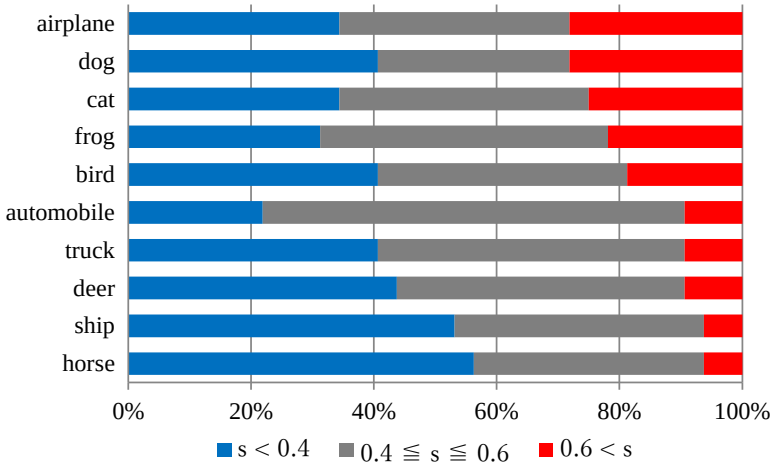


Figure 5.6: Distribution of weights of channels by the channel-wise switcher on the CIFAR-10 dataset. The channels are grouped by average switch weight for each channel.

5.4.2 Person/non-person classification

Setting

Next, RIS was evaluated in the context of a binary person classification task.

For this evaluation, a subset of the MS-COCO dataset [68] was used. First, images in all categories were cropped using the ground-truth bounding boxes and the object in each image was centered. Then the cropped images were resized to 64×64 pixels. Finally, the cropped and resized images were categorized into two classes, person class, and non-person class. The datasets for training and testing were constructed from the train-2014 set and val-2014 set, respectively.[1] For training, 3,950 images were used for person class and 38,280 images for non-person class. For testing, 1,704 and 20,010 images were used for person class and non-person class, respectively. For data-augmentation during training, the images were randomly flipped and shifted.

The network consists of one convolutional layer (*conv1*), four res-blocks (*res2, res3, res4, res5*), global average polling and full-connection. RIS was evaluated on the two architectures shown in Fig. 5.5. For comparison, two types of networks similar to the previous experiment were used.

Training was done like the previous experiment. However, since the dataset was unbalanced, a mini-batch was constructed by sampling equally for each class.

[1] All images used are licensed CC-BY-2.0, CC-BY-SA-2.0 or CC-BY-ND-2.0.

Table 5.2: Error ratios [%] on the MS-COCO person/non-person dataset. Rotation invariance is abbreviated as "RI." "Multi-RIS" column checks indicate models based on the multiple RIS architecture (Fig. 5.5b). Unchecked models are based on the single RIS architecture (Fig. 5.5a).

Model		Multi-RIS	Error
w/o RIS	non RI		1.021
	RI		0.954
		✓	1.359
w/ RIS	image-wise		1.003
		✓	**0.890**
	position-wise		0.963
		✓	**0.902**
	channel-wise		**0.948**
		✓	1.000
	pos/ch-wise		**0.922**
		✓	**0.936**

Results

Similar to the previous experiment, the use of RIS lead to an improvement in classification performance (Table 5.2). Performance of the image-wise and position-wise switchers improved through the use of multiple RIS modules, while the classification performance of the model without RIS decreased when multiple conventional rotation invariance modules were used. This result indicates that adaptive switching by RIS effectively works on each layer.

The switch weight distributions on the image-wise architecture are shown in Fig. 5.7. Distributions of persons and the non-persons for res2-4 are sharp and similar to each other. However, the distribution for res5 is wider and distinctly different from the other classes, indicating that the rotatability appears in a deep layer close to the output, not a shallow layer close to the input and depends on the input's contents such as category.

The relationship between the input image and switch weight for res5 are shown in Fig. 5.8. Non-person images which contained rotatable objects such as kites and fruits (Fig. 5.8c) were weighted with a large switch weight. Non-person images with a small switch weight (Fig. 5.8d), contained non-rotatable objects such as cars and horses. Interestingly, while the person images were expected to have a large switch weight (Fig. 5.8b), images containing persons lying down or sitting were found to have a small switch weight (Fig. 5.8a), suggesting that the switch weight is pose-sensitive. These results suggest that RIS depends on not only the

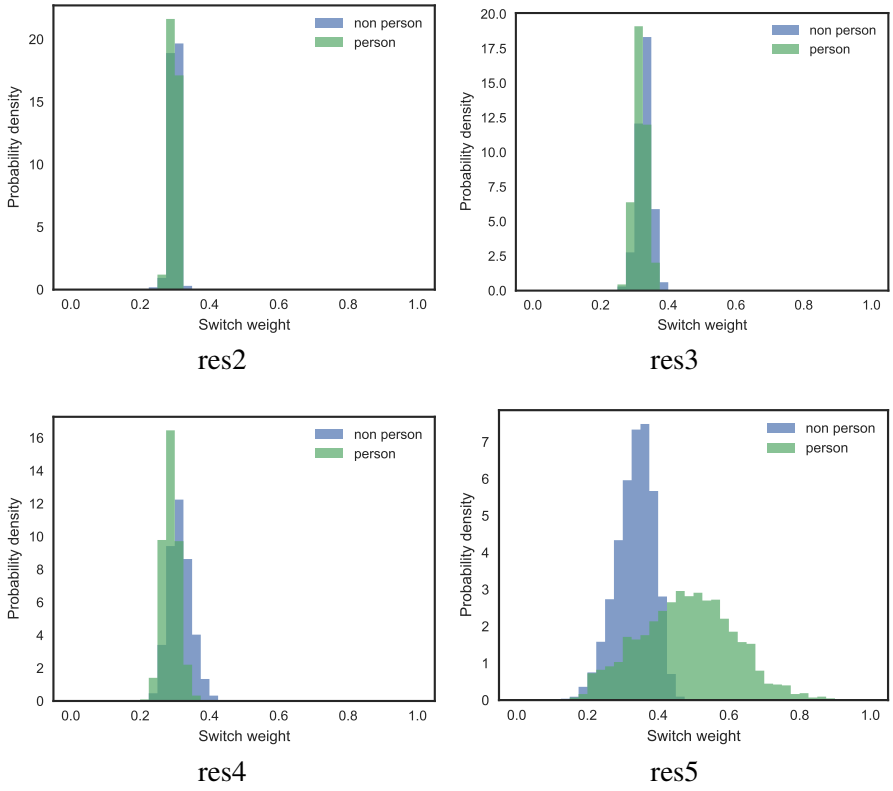


Figure 5.7: Switch weight distributions of the image-wise switcher on the MS-COCO person/non-person dataset.

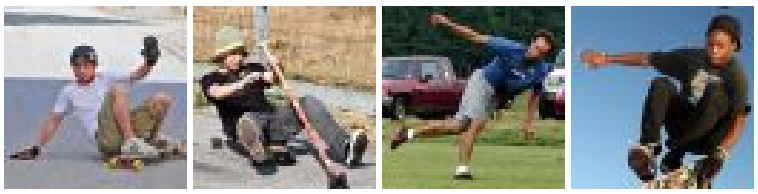
explicit target categories but also implicit properties such as the pose of a person and the content of the input image.

A heatmap of the switch weight of the position-wise switcher is shown in Fig. 5.9. For res2-3, the weights on the object edges are large, irrespective of class. The weights for res4-5 are dependent on class, with large switch weights for persons and small switch weights for non-persons. Interestingly, the weights around a person's neck for res4 are large and can be concluded that RIS responses to parts of a person and that rotatability may depend on object parts.

5.5 Summary

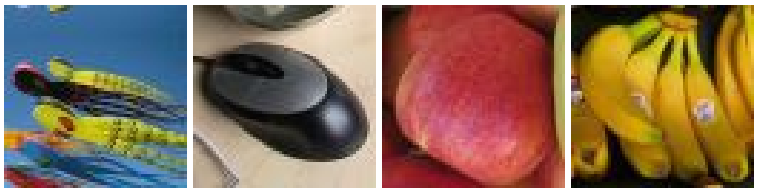
In this chapter, we have proposed Rotation Invariance Switcher (RIS), which estimates the rotatability of an input image and adaptively switches between using rotation invariance or variance. Four RIS modules have been developed that switch on different axes, and multiple RIS modules can be applied at different scales in

a network. The CIFAR-10 and CIFAR-100 datasets were used to evaluate object classification, and a subset of the MS-COCO dataset was used for person/non-person classification. We have demonstrated that RIS improves recognition performance and enables us to observe rotatability from various viewpoints. We hope that the observations via RIS will promote understanding of the relationship between object rotation and image recognition.



(a) Person images / Switch weight: Large

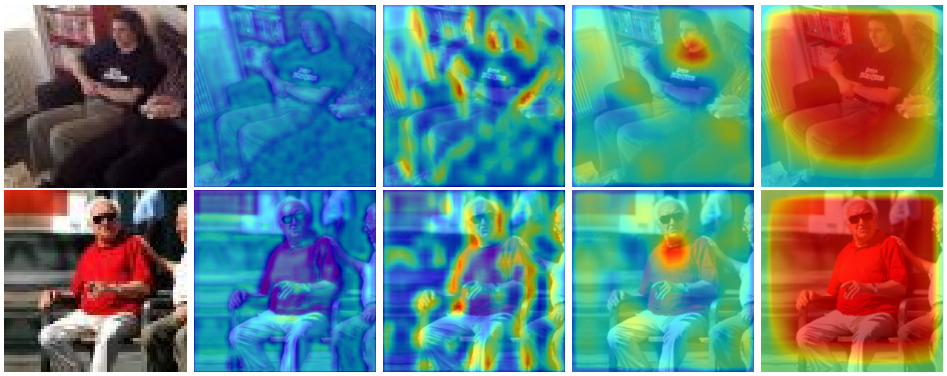
(b) Person images / Switch weight: Small



(c) Non-person images / Switch weight: Large

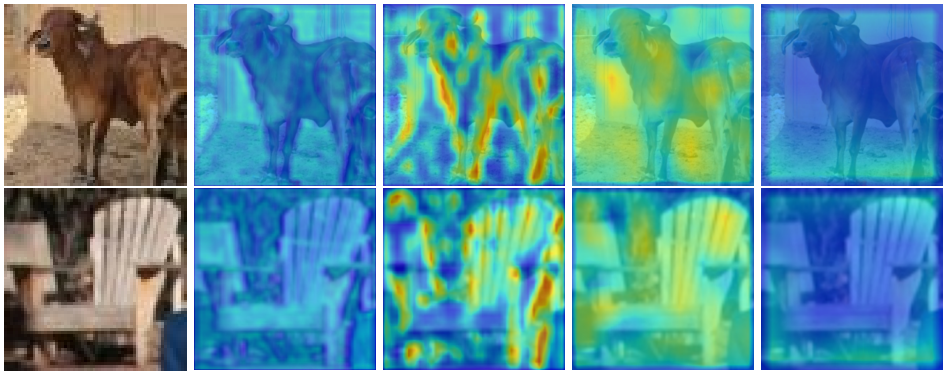
(d) Non-person images / Switch weight: Small

Figure 5.8: Example images and switch weights of the image-wise switcher on the MS-COCO person/non-person dataset. The image IDs in the MS-COCO dataset [68] for each image are following (from left to right): (a) 278095, 465820, 209162, 436317. (b) 359115, 576085, 209763, 516800. (c) 75806, 248314, 424174, 10432. (d) 301634, 395046, 198943, 421109.



input res2 res3 res4 res5

(a) person



input res2 res3 res4 res5

(b) non-person

Figure 5.9: Switch weight heatmap of the position-wise switcher on the MS-COCO person/non-person dataset. The image IDs in the MS-COCO dataset [68] for each image are following (from top to bottom): (a) 34657, 427113. (b) 360986, 33044.

Chapter 6

Conclusion

In this study, we have addressed how to handle the rotation of an object effectively in feature extraction for image recognition from three perspectives, (1) the rotation invariant feature, (2) the rotation variant feature, and (3) both combination. Then, we have proposed the following four novel methods based on them.

First, we have enhanced LBP histogram, which is commonly used in many types of image recognition tasks, by considering the spatial co-occurrence among local LBPs. One method was named "Co-occurrence among Adjacent LBPs" (CoALBP). In this method, the spatial co-occurrence among LBPs is represented by a histogram of LBP pairs. The performance of CoALBP was evaluated on multiple tasks of face recognition under the various illumination and texture classification to confirm the effectiveness of the combination of the spatial co-occurrence and LBP representation. CoALBP demonstrated high and practical performance on the HEp-2 cells image classification 2012.

Next, we have incorporated the rotation invariance into CoALBP to construct "Rotation Invariant Co-occurrence among LBPs" (RIC-LBP). In this method, the rotation invariance was realized by using histograms of LBP pairs that can be regarded to be rotation equivalent with each other. RIC-LBP inherits the advantages of CoALBP such as the high descriptive ability and low computational cost while retaining the robustness against rotation of an object. RIC-LBP was also evaluated on texture classification and HEp-2 cells classification to confirm the effectiveness.

Then, we considered rotation variant feature by CNN and rotation covariance which means the situation that an object and an oriented bounding box can rotate together. Under this situation, we have proposed a novel bounding box regression named "Orientation-Aware Regression" (OAR), in which the object's orientation is propagated to the output bounding box in 2D-vector representation. The proposed

method can be adapted to not only applications with existing detectors but also to other applications such as polygon mask estimation and bounding box estimation on 3D-images. We applied OAR to person detection and confirmed experimentally that a bounding box estimated by OAR is very accurate and robust against the rotation of an object, improving the detection performance.

Finally, we have considered the combination of rotation invariant feature and the rotation variant feature, and we have proposed "Rotation Invariance Switcher" (RIS). This method estimates the "rotatability" of an input image, which means whether a rotated input image can make sense, and adaptively switches between rotation invariant features or rotation variant features. As RIS can be used as a kind of module for CNN, it can be adapted to various types of CNN architectures. We applied RIS to object classification tasks and confirmed that RIS could achieve high recognition performance by observing the rotatability from various viewpoints.

Feature works are as follow. Our study has focused on only 2D rotation of an object in an image. However, 3D rotation of 3D object in an image remains as a more challenging task, because 3D rotation can induce significant change in appearance of an object. To overcome this problem, some implicit 3D representation is required. For example, recognition processing is performed using pre-estimated 3D information such as the attitude of an object and the positions of an object's parts in 3D. Besides, hierarchical rotation information is also useful. Object's rotation can be found in multi-scale (*e.g.*, rotation of an entire object as global rotation and rotation of each part of an object as local rotation). It should be studied that how to describe or estimate the relations between rotations in different scales. We expect that these considerations would make the estimation of the object's rotation more accurate and efficient.

Bibliography

[1] P Foggia, G Percannella, P Soda, and M Vento. Early experiences in mitotic cells recognition on HEp-2 slides. In *IEEE International Symposium on Computer-Based Medical Systems*, pages 38–43, 2010.

[2] Timo Ojala, Topi Mäenpää, Matti Pietikäinen, Jaakko Viertola, Juhaönen Kyll, and Sami Huovinen. Outex - New framework for empirical evaluation of texture analysis algorithms. *Pattern Recognition*, pages 701–706, 2002.

[3] Timo Ojala, Matti Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59, 1996.

[4] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893, 2005.

[5] Paul Viola and Michael Jones. Rapid Object Detection Using a Boosted Cascade of Simple Features. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[6] A Oliva and A Torralba. Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[7] David G Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[8] Michael Calonder, Vincent Lepetit, C. Strecha, and Pascal Fua. BRIEF: Binary Robust Independent Elementary Features. In *European Conference on Computer Vision*, pages 778–792, 2010.

[9] Stefan Leutenegger, Margarita Chli, and R.Y. Siegwart. BRISK: Binary Robust Invariant Scalable Keypoints. In *IEEE International Conference on Computer Vision*, pages 2548–2555, 2011.

[10] Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2037–2041, 2006.

[11] G Zhao and M Pietikäinen. Dynamic Texture Recognition Using Local Binary Patterns with an Application to Facial Expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):915–928, 2007.

[12] G. Zhao and Matti Pietikäinen. Local binary pattern descriptors for dynamic texture recognition. In *International Conference on Pattern Recognition*, pages 211–214, 2006.

[13] Matti Pietikäinen, Timo Ojala, and Z. Xu. Rotation-invariant texture classification using feature distributions. *Pattern Recognition*, 33(1):43–52, 2000.

[14] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[15] Timo Ahonen, J. Matas, C. He, and Matti Pietikäinen. Rotation invariant image description with local binary pattern histogram fourier features. *Image Analysis*, 5575:61–70, 2009.

[16] Xiaoyang Tan and Bill Triggs. Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions. *IEEE Transactions on Image Processing*, 19(6):1635–1650, 2007.

[17] Ville Ojansivu and Janne Heikkilä. Blur Insensitive Texture Classification Using Local Phase Quantization. In *International Conference on Image and Signal Processing*, pages 236–243, 2008.

[18] Daniel Maturana, Domingo Mery, and Alvaro Soto. Learning discriminative local binary patterns for face recognition. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 470–475, 2011.

[19] Liu Li, Paul Fieguth, and Gangyao Kuang. Generalized Local Binary Patterns for Texture Classification. In *British Machine Vision Conference*, pages 123.1–123.11, 2011.

[20] Li Liu, Lingjun Zhao, Yunli Long, Gangyao Kuang, and Paul Fieguth. Extended local binary patterns for texture classification. *Image and Vision Computing*, 30(2):86–99, 2012.

[21] T Watanabe, S Ito, and K Yokoi. Co-occurrence Histograms of Oriented Gradients for Pedestrian Detection. *Pacific-Rim Symposium on Image and Video Technology*, pages 37–47, 2009.

[22] Takumi Kobayashi and Nobuyuki Otsu. Image feature extraction using gradient Local Auto-Correlations. In *European Conference on Computer Vision*, pages 346–358, 2008.

[23] T Mita, T Kaneko, B Stenger, and O Hori. Discriminative Feature Co-Occurrence Selection for Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(7):1257–1269, 2008.

[24] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.

[25] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng Yang Fu, and Alexander C. Berg. SSD: Single shot multibox detector. In *European conference on computer vision*, pages 21–37, 2016.

[26] Zikun Liu, Jingao Hu, Lubin Weng, and Yiping Yang. Rotated region based CNN for ship detection. In *IEEE International Conference on Image Processing*, pages 900–904, 2017.

[27] Lars Wilko Sommer, Tobias Schuchert, and J. Beyerer. Fast deep vehicle detection in aerial images. In *IEEE Winter Conference on Applications of Computer Vision*, pages 311–319, 2017.

[28] Minghui Liao, Baoguang Shi, and Xiang Bai. TextBoxes++: A Single-Shot Oriented Scene Text Detector. *IEEE Transactions on Image Processing*, 27(8):3676–3690, 2018.

[29] Haibo Wang, Angel Cruz-Roa, Ajay Basavanhally, Hannah Gilmore, Natalie Shih, Mike Feldman, John Tomaszewski, Fabio Gonzalez, and Anant Madabhushi. Cascaded Ensemble of Convolutional Neural Networks and Handcrafted Features for Mitosis Detection. In *Medical Imaging 2014: Digital Pathology*, page 90410B, 2014.

[30] Shangxuan Wu, Ying-Cong Chen, Xiang Li, An-Cong Wu, Jin-Jie You, and Wei-Shi Zheng. An Enhanced Deep Feature Representation for Person Re-identification. In *IEEE Winter Conference on Applications of Computer Vision*, pages 1–8, 2016.

[31] Loris Nanni and Stefano Ghidoni. How could a subcellular image, or a painting by Van Gogh, be similar to a great white shark or to a pizza? *Pattern Recognition Letters*, 85(1):1–7, 2017.

[32] Loris Nanni, Stefano Ghidoni, and Sheryl Brahnam. Handcrafted vs. non-handcrafted features for computer vision classification. *Pattern Recognition*, 71:158–172, 2017.

[33] Loris Nanni, Sheryl Brahnam, Stefano Ghidoni, and Alessandra Lumini. Bioimage Classification with Handcrafted and Learned Features. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2018.

[34] Dat Tien Nguyen, Tuyen Danh Pham, Na Rae Baek, Kang Ryoung Park, Dat Tien Nguyen, Tuyen Danh Pham, Na Rae Baek, and Kang Ryoung Park. Combining Deep and Handcrafted Image Features for Presentation Attack Detection in Face Recognition Systems Using Visible-Light Camera Sensors. *Sensors*, 18(3):699, 2018.

[35] Kuang-Chih Lee, David J. Kriegman, and Jeffrey Ho. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):684–698, 2005.

[36] C Liu and H Wechsler. Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *IEEE Transactions on Image Processing*, 11(4):467–476, 2002.

[37] Michael J. Swain and Dana H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[38] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[39] O Yamaguchi, K Fukui, and K Maeda. Face recognition using temporal image sequence. *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 318–323, 1998.

[40] Tomokazu Kawahara, Masashi Nishiyama, Tatsuo Kozakaya, and Osamu Yamaguchi. Face Recognition based on Whitening Transformation of Distribution of Subspaces. *Asian Conference on Computer Vision Workshop on Subspace*, pages 97–103, 2007.

[41] Pasquale Foggia, Gennaro Percannella, Paolo Soda, and Mario Vento. Benchmarking HEp-2 Cells Classification Methods. *IEEE Transactions on Medical Imaging*, 32(10):1878–1889, 2013.

[42] S Lazebnik, C Schmid, and J Ponce. A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1265–1278, 2005.

[43] Pasquale Foggia, Gennaro Percannella, Paolo Soda, and Mario Vento. Special issue on the analysis and recognition of indirect immuno-fluorescence images. *Pattern Recognition*, 47(7), 2013.

[44] Gennaro Percannella, Pasquale Foggia, and Paolo Soda. HEp-2 Cells Classification Contest. http://mivia.unisa.it/hep2contest/, 2012.

[45] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

[46] Zhenhua Guo, Lei Zhang, and D Zhang. A Completed Modeling of Local Binary Pattern Operator for Texture Classification. *IEEE Transactions on Image Processing*, 19(6):1657–1663, 2010.

[47] Ryusuke Nosaka, Yasuhiro Ohkawa, and Kazuhiro Fukui. Feature Extraction Based on Co-occurrence of Adjacent Local Binary Patterns. *Pacific-Rim Symposium on Image and Video Technology*, 2:82–91, 2011.

[48] Goran Konjevod and T Nathan Mundhenk. Deep Multi-Modal Vehicle Detection in Aerial ISR Imagery. In *IEEE Winter Conference on Applications of Computer Vision*, pages 916–923, 2017.

[49] Oded Krams and Nahum Kiryati. People detection in top-view fisheye imaging. In *IEEE International Conference on Advanced Video and Signal-based Surveillance*, pages 1–6, 2017.

[50] Xinshuo Weng, Shangxuan Wu, Fares Beainy, and Kris M Kitani. Rotational Rectification Network: Enabling Pedestrian Detection for Mobile Vision. *IEEE Winter Conference on Applications of Computer Vision*, pages 1084–1092, 2018.

[51] Gong Cheng, Peicheng Zhou, and Junwei Han. RIFD-CNN: Rotation-Invariant and Fisher Discriminative Convolutional Neural Networks for Object Detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2884–2893, 2016.

[52] Minghui Liao, Zhen Zhu, Baoguang Shi, Gui-Song Xia, and Xiang Bai. Rotation-Sensitive Regression for Oriented Scene Text Detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5909–5918, 2018.

[53] Yanzhao Zhou, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Oriented Response Networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4961–4970, 2017.

[54] P F Felzenszwalb, R B Girshick, D McAllester, and D Ramanan. Object Detection with Discriminatively Trained Part-Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, sep 2010.

[55] Dmitry Laptev, Nikolay Savinov, Joachim M. Buhmann, and Marc Pollefeys. TI-POOLING: transformation-invariant pooling for feature learning in Convolutional Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 289–297, 2016.

[56] Diego Marcos, Michele Volpi, Nikos Komodakis, and Devis Tuia. Rotation Equivariant Vector Field Networks. In *IEEE International Conference on Computer Vision*, pages 5058–5067, 2017.

[57] Taco S. Cohen and Max Welling. Group Equivariant Convolutional Networks. In *International Conference on Machine Learning*, volume 48, pages 2990–2999, 2016.

[58] Taco S. Cohen and Max Welling. Steerable CNNs. In *International Conference on Learning Representations*, 2017.

[59] Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical CNNs. In *International Conference on Learning Representations*, 2018.

[60] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial Transformer Networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.

[61] Daniel Worrall, Stephan J. Garbin, Daniyar Turmukhambetov, and Gabriel J. Brostow. Harmonic Networks: Deep Translation and Rotation Equivariance. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7168–7177, 2017.

[62] Yann LeCun. Learning Invariant Feature Hierarchies. In *European Conference on Computer Vision*, pages 496–505, 2012.

[63] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks For Large-Scale Image Recognition. In *International Conference on Learning Representations*, 2015.

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

[65] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-Excitation Networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018.

[66] Sergey Zagoruyko and Nikos Komodakis. Wide Residual Networks. In *British Machine Vision Conference*, pages 87.1–87.12, 2016.

[67] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, pages 3856–3866, 2017.

[68] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO : Common Objects in Context. In *European Conference on Computer Vision*, pages 740–755, 2014.

[69] Diego Marcos, Michele Volpi, and Devis Tuia. Learning rotation invariant convolutional filters for texture classification. In *International Conference on Pattern Recognition*, pages 2012–2017, 2017.

[70] Xuepeng Shi, Shiguang Shan, Meina Kan, Shuzhe Wu, and Xilin Chen. Real-Time Rotation-Invariant Face Detection with Progressive Calibration Networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2295–2303, 2018.

[71] Erik J Bekkers, Maxime W Lafarge, Mitko Veta, Koen A J Eppenhof, Josien P W Pluim, and Remco Duits. Roto-Translation Covariant Convolutional Networks for Medical Image Analysis. *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 440–448, 2018.

[72] Rupesh Kumer Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training Very Deep Networks. In *Advances in Neural Information Processing Systems*, pages 2377–2385, 2015.

[73] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual Attention Network for Image Classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6450–6458, 2017.

[74] Saumya Jetley, Nicholas A Lord, Namhoon Lee, and Philip H S Torr. Learn To Pay Attention. In *International Conference on Learning Representations*, 2018.

[75] Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L Yuille. Attention to Scale : Scale-aware Semantic Image Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3640–3649, 2016.

[76] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images, 2009.

Acknowledgments

This work has been done while I was at University of Tsukuba and SECOM Co., Ltd., and it was impossible without the support many people.

First of all, I offer my sincerest gratitude to my supervisor, Professor Kazuhiro Fukui, who has supported me during for many years. He introduced me to the research field of computer vision when I was undergraduate, and gave me much advice and discussions.

For organizing this thesis, Professor Keisuke Kameyama, Professor Yutaka Satoh, Associate Professor Hotaka Takizawa, and Associate Professor Itaru Kitahara also provided helpful comments, and I would like to thank them.

I would like to thank the support from people at Intelligent Systems Laboratory, SECOM Co., Ltd. Among them, Mr. Hidenori Ujiie and Mr. Takaharu Kurokawa gave me much support and discussions for this work. I also appreciate their great effort towards my advance to the doctoral course.

Finally, I thank my family for supporting my life. In particular, my wife Haruna and my daughter Minori allowed me to spend much time researching and writing and gave me a lot of loving support and warm encouragement.

List of Publications

International journal

1. Ryusuke Nosaka, Kazuhiro Fukui, "HEp-2 cell classification using rotation invariant co-occurrence among local binary patterns", *Pattern Recognition*, Vol. 47, Issue 7, pp. 2428–2436, 2014.

International conference

1. Ryusuke Nosaka, Hidenori Ujiie, Takaharu Kurokawa, "Orientation-Aware Regression for Oriented Bounding Box Estimation", *IEEE International Conference on Advanced Video and Signal-based Surveillance (AVSS)*, 2018.

2. Ryusuke Nosaka, Chendra Hadi Suryanto, Kazuhiro Fukui, "Rotation invariant co-occurrence among adjacent LBPs", *International Workshop on Computer Vision With Local Binary Pattern Variants (LBP)*, Part I, Vol. 7728, pp. 15–25, 2012.

3. Ryusuke Nosaka, Yasuhiro Ohkawa, Kazuhiro Fukui, "Feature Extraction Based on Co-occurrence of Adjacent Local Binary Patterns", *Pacific-Rim Symposium on Image and Video Technology (PSIVT)*, Part II, Vol. 7088, pp. 82–91, 2011.