Article:

Chen, Jinhui, Lue, Zhaojie, Zhang, Zhihong et al. (4 more authors) (2019) Polar Transformation on Image Features for Orientation-Invariant Representations. IEEE Transactions on Multimedia. pp. 300-313. ISSN 1520-9210

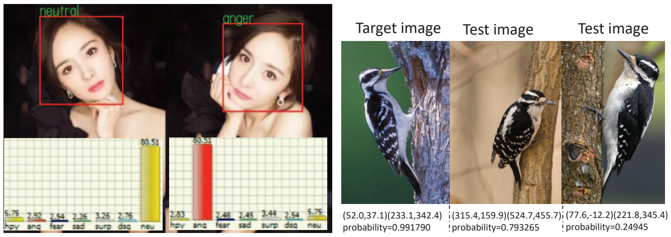
https://doi.org/10.1109/TMM.2018.2856121

Polar Transformation on Image Features for Orientation-Invariant Representations

Jinhui Chen[†], Zhaojie Luo[†], Zhihong Zhang[*], Faliang Huang, Zhiling Ye, Tetsuya Takiguchi, and Edwin R. Hancock, *Fellow, IEEE*

***Abstract*—The choice of image feature representation plays a crucial role in the analysis of visual information. Although vast numbers of alternative robust feature representation models have been proposed to improve the performance of different visual tasks, most existing feature representations (*e.g.* handcrafted features or Convolutional Neural Networks (CNN)) have a relatively limited capacity to capture the highly orientation-invariant (rotation/reversal) features. The net consequence is suboptimal visual performance. To address these problems, this study adopts a novel transformational approach, which investigates the potential of using polar feature representations. Our low level consists of a histogram of oriented gradient, which is then binned using annular spatial bin-type cells applied to the polar gradient. This gives gradient binning invariance for feature extraction. In this way, the descriptors have significantly enhanced orientation-invariant capabilities. The proposed feature representation, termed *orientation-invariant histograms of oriented gradients* (Oi-HOG), is capable of accurately processing visual tasks (*e.g.*, facial expression recognition). In the context of the CNN architecture, we propose two polar convolution operations, referred to as Full Polar Convolution (FPolarConv) and Local Polar Convolution (LPolarConv), and use these to develop polar architectures for the CNN orientation-invariant representation. Experimental results show that the proposed orientation-invariant image representation, based on polar models for both handcrafted features and deep learning features, is both competitive with state-of-the-art methods and maintains a compact representation on a set of challenging benchmark image datasets.**

***Index Terms*—Rotation-invariant and reversal-invariant representation, HOG, CNN.**



(a) FER based on handcrafted descriptors (HOG) **(b)** Image classification based on deep features (CNN)

Fig. 1: Sensitivity of the handcrafted features and CNN features to large transformation: **(a)** Error result of the FER caused by the head-rotated cases due to the handcrafted feature (HOG) is not rotation- invariant; **(b)** The feature extracted on CNN model lacks of reversal invariance, which leads to a non-correlation probability calculated by feature correlations between test and target (reversed case) images (containing same-class objects) for image classification.

I. Introduction

IMAGES dominate multimedia data. The manipulation of this data is therefore dependent on the availability of effective visual tools (*e.g.*, image classification/retrieval engines). One of the most significant challenges is to develop robust image feature representation models [1], which can used for a wide range of visual tasks, such as scene classification [2], image annotation [3], object recognition/localization [4] *etc.*

Existing feature representation methods can be roughly divided into two categories, namely a) conventional handcrafted descriptor based image features and b) deep neural network based image features. Specifically, handcrafted descriptors are one of the most popular feature representation models. They can be used as input to probabilistic algorithms for learning classifiers designed to produce discriminative visual words. Recently, with the availability of both large-scale publicly available image datasets and high-performance processors, deep feature models and in particular Convolutional Neural Network (CNN), are viewed as the state-of-the-art in numerous visual tasks [5], [6]. However, many papers in the literature reveal that both deep and handcrafted representation models are sensitive to orientation (reversal or rotation) deformations [5], [7]–[9]. This in turn leads to an overall limitation of their performance in visual processing tasks. Consequently, work aimed at solving these shortcomings of existing feature representation models is still an active area of research.

To provide a visual illustration, in Fig. 1 we consider an image together with its reversed version, which obviously convey the same visual concepts. We have conducted an experiment (without augmented training data) to test the sensitivity of both CNN features as well as handcrafted features (histograms of oriented gradients (HOG) [10]) to large transformations that result in a modified arrangement of the underlying objects. Both the HOG and CNN representations of the two images are totally different after their orientations are deformed. Such a change leads to an inferior performance on the subsequent visual processing tasks (see Fig. 1). The

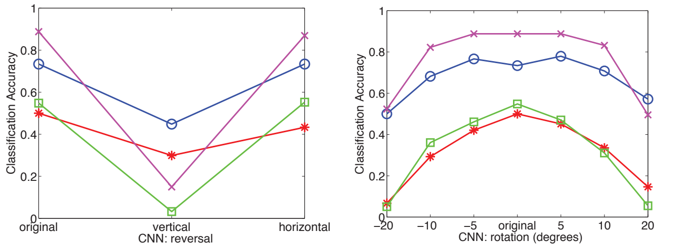
[†] Equal contribution. [*] Corresponding author: Zhihong Zhang (E-mail: zhihong@xmu.edu.cn).

J. Chen, Z. Luo and T. Takiguchi are with the Graduate School of System Informatics, Kobe University, Kobe, Japan.

Z. Zhang and Z. Ye are with the Xiamen University, Xiamen, China.

F. Huang is with College of Mathematics and Informatics, Fujian Normal University, Fuzhou, China.

E. R. Hancock is with the the Department of Computer Science, The University of York, York, UK.



(a) Reversal

(b) Rotation

Fig. 2: Accuracies for image classification on four classes (red-line: arrival gate; blue-line: florist shop; pink-line: volleyball court, green-line: ice skating) from the SUN dataset [7] as a function of different transformations of the test images: **(a)** Reversal; **(b)** Rotation.

reason is primarily that existing deep features together with the handcrafted descriptors are generally not orientation-invariant. As a result the descriptor structures are radically altered by reversal [1], [5], [7], [9]. Consequently, it is difficult to address tasks such as feature correspondence once the orientation (rotation/reversal) of objects is modified. This further leads to an overall limitation of the performance when classification is attempted using these descriptors.

As widely demonstrated, convolutional networks provide an effective architecture to analyze large-scale and high-dimensional data. In a manner similar to related work [7], we have evaluated the sensitivity of the CNN model on reversal transformation for the test image in Fig.2(a) and rotated version of the test image in Fig.2(b) respectively. It is clear that the final CNN representations are sensitive to both image reversal and large rotations. The results further indicate that the max-pooling operations of the CNN model have a limited capacity to handle large image rotations. In other words, there is still considerable scope for improving the demanding task of designing an effective image representation for learning discriminative models.

To address the above challenge we explore a novel transformation approach to enhancing the robustness of both deep features and handcrafted features. A successful feature representation method should exhibit two desirable properties, namely a) robustness, that is the model should be able to learn features independent of scaling, rotation, shifting, photometric deformations and noise, and b) it should be structure-preserving, that is the learned representation model should preserve the discriminative structure extracted from raw data samples.

In this paper, we propose an orientation-invariant (rotation/reversal) image representation based on polar models for both handcrafted features and deep features. In this way we can obtain identical representations for an image and its orientation-deformed version without training using augmentation data. For handcrafted feature representation, we have proposed a HOG-type polar feature. Here we transform gradients into polar form (see Fig. 3) and adopt the polar descriptor to design a local feature to achieve orientation-invariant representation. We refer to the proposed feature as orientation-invariant histograms of oriented gradients (Oi-HOG). Experiments on real-time classification tasks (*e.g.* facial expression recognition (**FER**)) verify the efficiency and effectiveness of our proposed Oi-HOG model.

Motivated by the advantages of the polar model-based handcrafted feature representation (Oi-HOG), in this paper, we generalize the idea of polar data structures so that they can be used in conjunction with the state-of-the-art CNN architecture. Specifically, we propose two polar convolution models, referred to as Full Polar Convolution (**FPolarConv**) and Local Polar Convolution (**FPolarConv**) respectively, to construct deep-polar architectures for deep image feature extraction. The proposed framework can be viewed as a polar-data CNN model (**P-CNN**). By using polar operations in the input layer, a new raw data structure based on polar data is generated that can replace the traditional RGB-channel input. For the convolutional data of the intermediate convolutional layers, we adopt FPolarConv (see Fig. 8(a)) and LPolarConv (see Fig. 8(b)) to transform the full feature sheets and convolution operations (kernels) into polar ones. We mathematically verify that the above two operations can preserve the orientation-invariant structure for CNN features. Compared with existing deep features [11]–[17], our proposed polar-data CNN (**P-CNN**) architecture has two advantages, namely i) unlike existing DNNs that rely on a very large number of augmented data to train a robust network with millions of parameters, the **P-CNN** excels at image classification on small to medium sized datasets by preserving orientation-invariant information for image representation in each layer, and ii) compared with prior-art deep models, polar input structure presents more candidate parameters for CNN learning.

In summary, the main aim of this study is to investigate how polar features affect image representation results. Both handcrafted and deep models are components of polar feature representation. The polar network approach is in fact an extension of studies concerning handcrafted ones. Our study makes the following contributions:

- We design polar-type architectures to represent image features, which structurally encapsulate orientation-invariant capabilities as well as preserve the discriminative power of the original features. Moreover, in this study, polar structure bridges both the handcrafted and deep feature representations, while inheriting the respective advantages of both representations for visual tasks.
- The **P-CNN** is the original deep representation model that adopts a polar architecture. It helps the comprehensive understanding of image the components utilized in CNN. The polar operations for convolutional networks proposed in the work reported in this paper are therefore likely to lead to further related research.
- Using a polar representation to replace the RGB input channels, we find that RGB data can be decomposed into primal information to gain more candidate representation parameters for learning.

The remainder of this paper is organized as follows. In Section II, we review the literature on related work, and discuss the relationship between our proposed model and a number of alternative methods. Sections IV and V respectively illustrate the orientation-invariant transformation on gradient descriptors for handcrafted features and provide details of their experimental evaluation. In section VI, we present the orientation-invariant transformation approaches for the CNN model. This is followed by an experimental evaluation in Section VII. Finally, conclusions are presented in Section VIII.

II. Related Works

Based on both shallow and deep models, a large number of related works have proposed various robust feature representation models to improve the performance of visual tasks. In contrast, this paper presents a general and basic approach for both handcrafted features and deep features aimed at achievement addressing orientation invariance (rotation/reversal). In this section, we will discuss and detail the differences between the proposed method and the prior-art in handcrafted features and deep features, respectively.

A. Handcrafted Descriptors

Conventional image descriptors represent salient image regions by using a set of handcrafted filters designed using prior knowledge. Examples include the Weber's Law descriptor [18], Gabor features [19], [20], scale invariant feature transform (SIFT) [21], histogram of oriented gradients (HOG) [10], local binary patterns (LBP) [22] and speeded up robust features (SURF) [23].

These local features are based on handcrafted descriptors such as texture (filters), histograms, *etc*., which are usually robust to some types of image transformation such as rotation or translation. For example, the SIFT descriptor [21] achieves invariance to rotation and robustness to a moderate degree of perspective transformation [24], [25]. Motivated by the SIFT descriptor for computing distinctive invariant local features, SURF [23] uses the integral image representation to speed up computation. Local binary patterns (LBP) [22] has the characteristics of being invariant against any monotonic transformation of the gray scale [22]. Rotation invariance is achieved by minimizing the LBP code value using the bit cyclic shift. The HOG feature [10] was first proposed to represent objects using the distribution of gradient intensities and orientations over spatially distributed regions. It has been widely acknowledged as one of the best features to capture edge or local shape information of objects. However, many related works have proved that the HOG is neither rotation-invariant nor reversal-invariant [8], [26]. Therefore, dealing with the above issues for HOG appears to be a research problem of both significance and urgency.

There are many existing histogram-based feature representations that claim to be invariant. Currently, two of the most popular and representative ones are 2D HOG [27] and HOG 3D [28], which offers interesting solutions to the problems caused by rotations. However, these methods also suffer from bottlenecks. The 2D HOG descriptor was motivated by Jhuang *et al.*'s approaches [29] that uses 2D Gabor-filter responses combined with optical flow. Their dense representations avoid some of the rotational problems but cannot completely solve them. Moreover, it brings further time complexity because 2D HOG requires a region of interest (ROI) around the task region, which is usually obtained by using either a separate detector or background subtraction followed by blob detection. Motivated by the SIFT descriptor [21], HOG 3D constructs a platonic solids system using auxiliary coordinates to achieve invariant feature representation. It is an interesting solution, but unfortunately comes at a high cost in terms of computational time and memory requirements. Although the task images (faces) are distributed over the 2D polar coordinates and all task images are congruent in order to reduce memory requirements, computational speed remains a bottleneck. Furthermore, HOG 3D is reliant on integral videos [28], which limits its applications. Therefore, neither of these approaches can be considered as ideal solutions.

In our method, we subdivide the local patch into annular spatial bins to achieve spatial binning invariance. Motivated by Takacs *et al.*'s rotation-invariant image features [30], we apply a polar gradient to achieve gradient binning invariance. A preliminary publication [8] describes a rotation-invariant descriptor based on the distance metric for content-based image retrieval. In this paper we provide a more in-depth development and analysis of our earlier work, while presenting a number of new improvements in efficiency and the discriminative power of the learning framework needed to adapt the proposed feature representation for classification tasks.

B. Deep Convolutional Neural Networks

Even though the aforementioned feature representations have shown impressive success on a variety of visual tasks, they are either handcrafted or restricted to shallow representations. Furthermore, they also have three limitations, namely a) most of the handcrafted design of the features is domain-specific and can only tackle specific types of transformation variance, b) the design process leading to these handcrafted features is time consuming and requires prior knowledge, c) they all adopt shallow models, which have relatively limited capacity to represent highly non-linear structures in the underlying data. Deep features can successfully avoid the above problems, but still suffer from the common issues *i.e.*, lacking orientation-invariance. In this paper, to demonstrate the more general usefulness of the proposed polar features. We therefore expand the method to encompass deep features.

Although some deep net operations, such as max-pooling within each feature map, can alleviate the effect of small-scale rotations of patterns, it is difficult to effectively handle the problem of large orientation deformations. Recently, there have been several related attempts to address these problems, including data augmentation and the spatial transform network (STN). Due to the limited size of the training dataset, data augmentation is an effective method to expand the training dataset with transformed versions of the original images, resulting in new samples as additional training data. It has been shown that data augmentation by learning all the possible transformations enforces robustness of a learning model to variations of the input [12], [15], [31], [32]. Despite the effectiveness of data augmentation, its main drawback lies in the fact that it is computationally expensive to learn a large

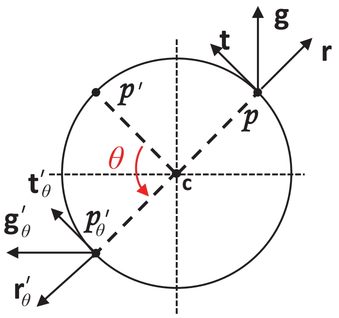


Fig. 3: Illustration of local polar coordinate system.

set of possible transformations of augmented data. Moreover, it significantly increases the number of network parameters and the risk of network over-fitting. As a representative example, the spatial transformer network (STN) [33] introduces an additional network module that can transform the input according to a defined class of transform parameters estimated using a localization sub-CNN. In this way STN contributes a general framework which can be applied to any existing CNN architecture for spatial transform. However, the problem of how to estimate the complex transform parameters by CNN remains unsolved [34], [35]. Furthermore, in the cases of big data or complex networks, the data augmentation approach performs much better than the ST-based approach [36], [37].

III. Definition and Theorem

Algorithm 1 Orientation-invariant Transformation Operation

```
Require:
    Raw picture $I \in \mathbb{R}^{H \times H}$;
Ensure:
    Orientation-invariant Vector field $\mathbb{G} = \mathbf{P}(I)$;
    Turn $I$ into pixel gradient map $\hat{I} \in \mathbb{R}^{H \times H} \times \mathbb{R}^{H \times H}$
    Set $j = 0$, $k = 0$ and set $\mathbf{c}$ as centre point of $\hat{I}$;
    for $j : H$ do
        for $k : H$ do
            $\mathbf{g} = \hat{I}[j, k]$;
            to yield a local polar coordinate system for the current point $p(j, k)$ : $\mathbf{r} = \frac{\mathbf{p}-\mathbf{c}}{\|\mathbf{p}-\mathbf{c}\|}$; $\mathbf{t} = \mathbf{r} \times R_{\frac{\pi}{2}}$;
            $\mathbb{G}[j, k] = (\mathbf{g}^T\mathbf{r}, \mathbf{g}^T\mathbf{t})$; // i.e., the encapsulation of $(\mathbf{P_r}(\cdot),\ \mathbf{P_t}(\cdot))$.
        end for
    end for
    Output $\mathbb{G}$;
```

In this section we describe the theoretical basis for our proposed approaches. We commence by proving the theorem, which underpins the work reported in this paper.

Definition Given an image $I \in \mathbb{R}^{H \times H}$, space sets of polar-gradient descriptor extracted from the original, reversed, rotated, and reversal-rotation/rotation-reversal copies are defined as $\mathbb{G} = P(I)$, $\mathbb{G}_M = P(I_M)$, $\mathbb{G}_R = P(I_R)$, $\mathbb{G}_{MR} = P(I_{MR})$, respectively. In this paper, we present a general algorithm (Algorithm 1) to implement the polar transformation operation for input data, which is denoted as $P(\cdot) = (P_r(\cdot), P_t(\cdot)) : \mathbb{R}^{n \times n} \to \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times n}$.

Theorem 1 Given a $G = (\mathbf{g}^T\mathbf{r}, \mathbf{g}^T\mathbf{t}) \in \mathbb{G}$, and the corresponding reversed, rotated, and reversal-rotation/rotation-reversal counterparts are denoted as $G_M \in \mathbb{G}_M$, given $G_R \in \mathbb{G}_R$ and given $G_{MR} \in \mathbb{G}_{MR}$, respectively; then $G = G_M = G_R = G_{MR}$, namely $P(I) = P(I_M) = P(I_R) = P(I_{MR})$.

Proof In theory, we need to verify three situations, including the reversed, the rotated and the reversal-rotation/rotation-reversal. Succinctly, we adopt the most complex one as the representative, *i.e.*, we verify the reversal-rotation/rotation-reversal one as the representative.

Now, we assume that the patch has been reversed and rotated against its center by a given angle θ, as shown in Fig. 3. This yields a new local coordinate system and gradient: $\mathbf{g}'_\theta = MR_\theta\mathbf{g}$, $\mathbf{r}'_\theta = MR_\theta\mathbf{r}$, $\mathbf{t}'_\theta = MR_\theta\mathbf{t}$. In addition, M is a diagonal matrix with diagonal elements 1 or -1 and R_θ is a rotation matrix. Obviously, M and R_θ both are orthogonal matrix. The coordinates of the gradient in the local frame are invariant to reversal as well as rotation, which is verified by:

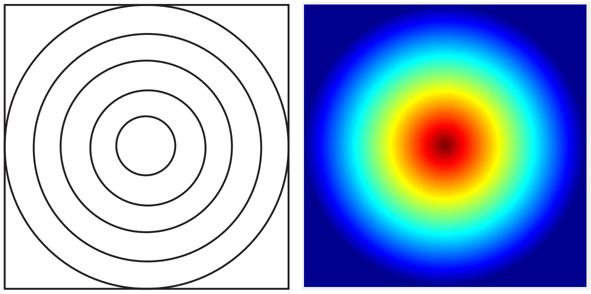
$$
\begin{aligned}
&\mathbf{g}'^T_\theta\mathbf{r}'_\theta \\
&= (MR_\theta\mathbf{g})^T MR_\theta\mathbf{r} \\
&= \mathbf{g}^T R_\theta^T M^T MR_\theta\mathbf{r} \\
&= \mathbf{g}^T\mathbf{r} \\
\Rightarrow &(\mathbf{g}'^T_\theta\mathbf{r}'_\theta, \mathbf{g}'^T_\theta\mathbf{t}'_\theta) = (\mathbf{g}^T\mathbf{r}, \mathbf{g}^T\mathbf{t}).
\end{aligned}
\tag{1}
$$

Obviously, the invariant availability under the rotated and reversed conditions, would be respectively verified in the same way.

Since the point $p(x, y)$ as well as the angle θ are arbitrary, and all the gradients are transformed in the same way, *i.e.*, they are a one-to-one mapping. Thus, the set of gradients at any given point around the feature patch is invariant to reversal as well as rotation. Therefore, both $G = G_M = G_R = G_{MR}$ and $\mathbb{G} = \mathbb{G}_M = \mathbb{G}_R = \mathbb{G}_{MR}$ can be established.

IV. Orientation-invariant HOG

We illustrate the procedure for calculating HOG descriptors using Fig. 3. In Fig. 3, given a point p on the circle, the task is to compute the polar gradient magnitude of point $p\ (x, y)$. Polar data can either be represented in Cartesian basis $\eta = [\alpha, \beta]^T \in \mathbb{R}^2$ or in a radial basis $[n, \varphi]$ by using norm n and angle φ of η. The radial representation can be used conveniently for Fourier analysis. However, in so doing, we require more complex transformational approaches. In fact, most current commonly available computers are sufficiently powerful to undertake most of the calculations required for HOG. Therefore, we adopt a Cartesian representation in this study. We decompose the corresponding gradient vector $\mathbf{g}$ into its local coordinate system as $(\mathbf{g}^T\mathbf{r}, \mathbf{g}^T\mathbf{t})$, by projecting $\mathbf{g}$ into the $\mathbf{r}$ and $\mathbf{t}$ orientations as shown in Fig. 3. Since the component vectors of $\mathbf{g}$ in $\mathbf{r}$ and $\mathbf{t}$ orientations can be quickly obtained by $\mathbf{r} = \frac{\mathbf{p}-\mathbf{c}}{\|\mathbf{p}-\mathbf{c}\|}$, $\mathbf{t} = \mathbf{r} \times R_{\frac{\pi}{2}}$, where $\mathbf{c}$ is the gradient vector of the central point and, R_θ is the rotation matrix by angle θ. In addition, we can obtain the gradient g easily using the gradient filter. We use the method outlined in Algorithm 2 to extract the **Oi-HOG** descriptors.



(a) Annular spatial cells **(b)** Polar gradient template

Fig. 4: Illustration of annular spatial cells and a polar gradient template image over the annular spatial patch (100 × 100 patch with 5 cells).

Alternatively, since some computational platforms have vector-processing resources, the matrix can sometimes be computed easily. We can therefore also represent these bases by adopting the angle of φ of vector $p-c$ to calculate the polar gradient η as follows,

$$\eta = \left[\begin{array}{c} r^{\mathrm{T}} \\ t^{\mathrm{T}} \end{array} \right] g, \tag{2}$$

where $r=(\cos\varphi, \sin\varphi)$, $t=(-\cos\varphi, \sin\varphi)$ and φ denotes the angle of vector $p-c$.

In this paper, we use Algorithm 1 to pre-calculate a batch of polar templates (Fig. 4(b)), which contain the full set of 1-norm Descartes vector elements. The maximum patch size of these templates consists of 100 × 100 different cells. We set the patch sizes to range from 50 × 50 pixels to 100 × 100 units by sliding the patch over the template with five pixels forward to ensure an adequate template-level difference. In addition, we allow different aspect ratios for each template patch (i.e. the ratio of width to height). In so doing, we can efficiently calculate polar gradients on local patches by computing the dot product between the entries of the template and the input, and then producing a 2-dimensional activation polar map for that template. In the training stage, this approach will cut out much of the repetition involved. The learning framework can thus quickly decide the best number of partitions required using the template pool to construct the classifier.

V. Experiments for Oi-HOG

In this section, we detail our experimental setup and evaluation results. We have implemented all the training and detection programs in C++ on a Win 10 (64-bit) OS on a PC with a Core i7-6700K 4.01-GHz CPU and 32 GB of RAM.

A. Dataset and Experimental Implementation

We now report the results of using our representation for facial expression recognition. To thoroughly evaluate the out-of-plane head orientation cases, we have evaluated the proposed method on two reference facial expression databases, *i.e.* MMI, and AFEW, which are respectively a lab-based database collected under controlled conditions and a database acquired in the wild under uncontrolled conditions.

Algorithm 2 Oi-HOG feature

```
Require:
    a data set $\Gamma$ over the feature local patch;
Ensure:
    Oi-HOG feature $\mathbf{F}(\Gamma)$;
Initialization: Set feature descriptor set $\mathbf{F} = \varnothing$ and i=0;
Orientation-invariant transformation: $\mathbb{G}_\Gamma = \mathbf{P}(\Gamma)$;
Subdivide the local patch into annular spatial cells, as shown in Fig. 4(a);
for all elements of $\mathbb{G}_\Gamma$ do
    Normalize the polar gradient using L2 normalization followed by clipping;
    Calculate the gradient magnitudes and orientations of the polar gradients using Eq. 3:
        $M_{GRT}(x,y) = \sqrt{(\mathbf{g}^T\mathbf{r})^2 + (\mathbf{g}^T\mathbf{t})^2},$
        $\theta(x,y) = arctan\frac{\mathbf{g}^T\mathbf{t}}{\mathbf{g}^T\mathbf{r}};$      (3)
    Determine the polar gradient magnitude of each pixel from the annular spatial cells and sort them into nine bins, according to their polar gradient orientations;
end for
for all spatial cells do
    for i : size(bins) do
        if Angles $\theta$ of current gradients belong to the angle range of i-th bin then
            Add max $(\mathbb{G}_\Gamma)$ to F; // max $(\cdot)$ denotes the operation function of getting the maximum element from an input set.
        end if
        i=i+1;
    end for
end for
Output F;
```

MMI DB The MMI DB [38] is a public database that includes more than 30 subjects, in which the female-male ratio is roughly 11:15. The subjects have ages ranging from 19 to 62 years, and they are of European, Asian or South American origins. This database is considered to be more challenging than CK+ [39], because there are many side-view images and some subjects have worn accessories such as glasses. We use the MMI database to evaluate out-of-plane head rotation under controlled conditions. In our experiments, we used all 205 image sequences of the six expressions in the MMI dataset. According to the person-independent levels, these images were categorized into sub-sets, which were made into videos for test.

AFEW DB The AFEW DB [40], is a much more challenging database. All of the AFEW image sets were collected from movies depicting so-called “in-the -wild scenarios” . In experiments, the videos are separated into images for training and validation. We have trained our models on the training

TABLE I: Training speed by adopting different features.

Method	Oi-HOG	HOG	SURF	SIFT	Haar
Time cost (min)	304	369	87	640	836

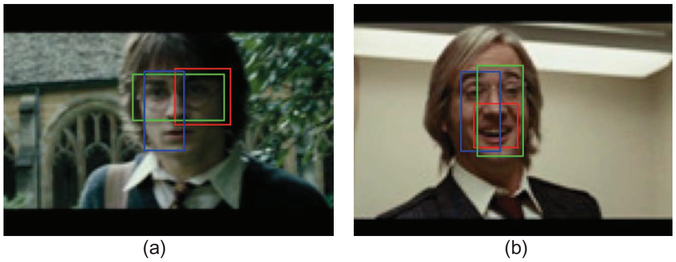


Fig. 5: Top-3 local patches picked by training procedure in the green-red-blue order on AFEW database.

sets and the evaluation results are reported on recognizing its validation set.

We used all of the training samples in the AFEW training set and collected training samples according to the person-independent 10-fold cross-validation rule. We normalized all of the training samples to 100 × 100-pixel facial patches. To enhance the generalization ability of the learning process, we performed a variety of transformations (illumination and scale) on the training samples, ultimately increasing the original number of samples by a factor of 64. We did not normalize the testing sample sequences. In the training stages, we used the training data for the current processing expression as our positive sample data and the remainder as our negative data.

B. Results Comparison

To learn the Oi-HOG features, we use a multithreading cascade algorithm, which appears in our previous publication [41] and the training procedure also follows the setting reported in our previous publication. The novel classification framework in the study reported in this paper is referred to as multithreading cascade of orientation-invariant histograms for oriented gradients (McOiHOG). Adopting the pre-calculated polar templates, the convergence rate for feature learning was improved by 73 minutes. To make fair comparison, we used the same learning framework and the same data (multithreading cascade [41]) for learning **Oi-HOG**, HOG, SURF, SIFT, and Haar features. The results are shown in Table I. The proposed method took 304 minutes to converge. The learning procedure only needed to evaluate 1.5 **Oi-HOG** per window for all categories.

After training, we observed that the top three local patches selected for FER were in the two eye and mouth regions. This situation is similar to Haar-based classifiers [42], as shown in the examples processed by the proposed framework in Fig. 5.

In this study, we have used the ground-truth labels provided for the expression categories given in the original databases. We based all of our recognition experiments on videos and evaluated their accuracies frame by frame.

To explore their feature orientation invariance, we have evaluated and compared several existing state-of-the-art learning models, adopting **Oi-HOG** as features. The results are listed in Table II, where the best one (MC) was used to construct the proposed feature learning framework for comparison with the state-of-the-art in Tables V and VI. The top performers are BinBoost [43], joint cascade (JC) [44], soft cascade (SC) [45] and multithreading cascade (MC) [41].

TABLE II: Average precision using different classifiers.

Database	Precision of classifiers (%)			
	BinBoost [43]	JC [44]	SC [45]	MC [41]
MMI	62.6	55.9	50.2	**72.4**
AFEW	43.9	40.6	26.8	**56.8**

TABLE III: Average precision using different features.

Database	Precision of feature (%)				
	SIFT	SURF	Haar	HOG	Oi-HOG
MMI	65.4	46.0	42.2	58.8	**72.4**
AFEW	41.5	35.8	17.3	32.4	**56.8**

To demonstrate the effectiveness of **Oi-HOG**, we further evaluated the selected learning framework for some popular existing local features, but without augmentation of the training data. Experimental results are shown in Table III. Under the same experimental conditions, we found that the **Oi-HOG** features outperform the alternatives. Moreover, the recognition accuracies of the proposed method in each facial expression category are relatively stable. However, for the alternative features, such as SURF, only the accuracies of the surprise and happiness images are acceptable, while for the other categories they were close zero. The SIFT results were slightly better than SURF, but its application is limited by its speed. These results show that the recognition results for the proposed method are more reliable than many existing local features. Moreover, the stability of the proposed method is better than many of the prior-art handcrafted features. We also considered using an off-line training process that does not have an impact on the real-time testing. Thus, when balanced with the accuracy, it is worth making some low-cost concessions in the implementation.

For comparison, we have also selected a number of state-of-the-art methods from this field, including those that have been proposed for improving spatiotemporal descriptors. These include LBP-TOP [46], HOE [47], and HOG 3D [28]. CLM [48] is a typical approach used to process facial action units. These methods are very popular for FER, whereas 3DCNN-DAP [49] and STM-ExpLet [50] are the most recently developed methods. We have also compared the classification frameworks for those methods that focus on enhancing the robustness of the classification approach. These include ITBN [51], 3D LUT [42] and LSH-CORF [52].

To ensure a fair comparison, we used the same databases and evaluated them via standardized items. Table V and VI compare our method (**McOiHOG**) with these state-of-the-art methods, most of which were conducted by using their released code and with their parameters tuned to give best performance in our experiments. However, we could not obtain the source codes of some methods including STM-ExpLet [50] and 3DCNN-DAP [49]), so we simply cite the results

TABLE IV: Comparisons of the proposed method with the state-of-the-art on PASCAL VOC 2007 dataset.

	Accuracy of different object-category items (%)																				
	airplane	*bike*	*bird*	*boat*	*bottle*	*bus*	*car*	*cat*	*chair*	*cow*	*table*	*dog*	*horse*	*mbike*	*person*	*plant*	*sheep*	*sofa*	*train*	*TV set*	Ave.
UCI [53]	28.8	56.2	3.2	14.2	**29.4**	38.7	48.7	12.4	16.0	17.7	24.0	11.7	45.0	39.4	35.5	15.2	16.1	20.1	34.2	35.4	27.1
DPM [54]	33.2	60.3	10.2	16.1	27.3	54.3	58.2	23.0	20.0	24.1	26.7	12.7	58.1	48.2	43.2	12.0	21.1	36.1	46.0	43.5	33.7
LEO [55]	29.4	55.8	9.4	14.3	28.6	44.0	51.3	21.3	20.0	19.3	25.2	12.5	50.4	38.4	36.6	15.1	19.7	25.1	36.8	39.3	29.6
DSO [56]	32.5	60.1	11.1	16.0	31.0	50.9	59.0	26.1	21.2	**26.5**	25.4	16.4	61.7	48.3	42.2	16.1	**28.2**	30.1	44.6	46.3	34.7
CA [57]	34.5	61.1	11.5	**19.0**	22.2	46.5	58.9	24.7	**21.7**	25.1	27.1	13.0	59.7	51.6	44.0	**19.2**	24.4	33.1	48.4	49.7	34.8
HoPS [58]	37.0	60.7	11.2	18.6	27.8	54.5	59.1	26.9	20.5	25.8	**29.0**	15.3	**59.9**	49.8	43.0	13.4	23.2	**38.4**	48.8	45.1	35.4
Ours	**39.2**	**66.9**	**19.6**	18.0	28.1	**58.4**	**64.7**	**30.6**	21.2	24.6	35.8	**21.7**	59.5	**55.2**	**51.6**	15.1	25.6	35.3	**49.5**	**51.9**	**38.6**

TABLE V: Recognition results on MMI.

Method	Accuracy on MMI (%)						
	An	Di	Fe	Ha	Sa	Su	Ave.
HOE [47]	46.4	58.3	33.2	62.6	60.8	65.1	55.5
LBP-TOP [46]	58.1	56.3	53.6	78.6	46.9	50.0	57.2
HOG 3D [28]	61.3	53.1	39.3	78.6	43.8	55.0	55.2
ITBN [51]	46.9	54.8	57.1	71.4	65.6	62.5	59.7
LSH [52]	59.6	**71.4**	62.3	68.9	**70.3**	75.1	61.8
3D LUT [42]	43.3	55.3	56.8	71.4	28.2	77.5	47.2
3DCNN-DAP [49]	64.5	62.5	50.0	**85.7**	53.1	57.5	62.2
STM [50]	–	–	–	–	–	–	65.4
Ours	**70.2**	60.4	**76.5**	81.2	62.1	**84.2**	**72.4**

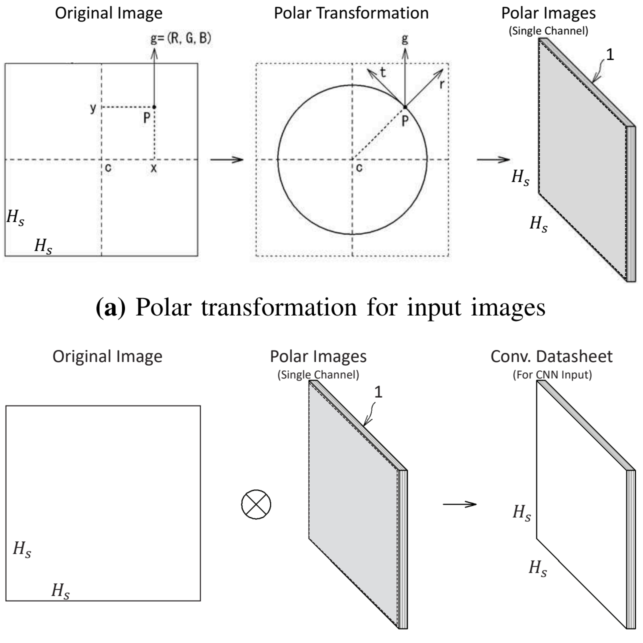
TABLE VI: Recognition results on AFEW.

Method	Accuracy on AFEW(%)						
	An	Di	Fe	Ha	Sa	Su	Ave.
HOE [47]	11.2	16.5	9.0	33.5	15.3	28.3	19.0
LBP-TOP [46]	11.7	**19.6**	17.9	42.3	**23.8**	33.6	24.8
HOG 3D [28]	–	–	–	–	–	–	26.9
LSH [52]	23.1	12.8	**38.6**	9.7	21.1	10.9	19.4
3D LUT [42]	45.7	0	0	**62.0**	13.2	48.6	28.2
STM [50]	–	–	–	–	–	–	31.7
Baseline [40]	50.0	25.0	15.2	57.1	16.4	21.7	33.2
Ours	**56.2**	36.3	**48.5**	**74.6**	**36.0**	**89.1**	**56.8**

reported in related studies of these methods. The average precision values (Ave.) of our framework (Ri-HOG cascade) were 72.4% and 56.8% when using MMI and AFEW. These levels represent the performances on MMI and AFEW datasets have been improved by the proposed framework, compared to the results obtained using the state-of-the-art methods.

To further evaluate the discriminative power of the proposed method, we also have executed experiments using PASCAL VOC 2007 dataset [59], which includes 9,963 images of 20 different object categories, containing 5,011 training images and 4,952 testing images. PASCAL VOC 2007 dataset is one of the most popular datasets in object detection/ recognition tasks and many evaluation experiments of state-of-the-art methods were evaluated on it. But please note that it is so difficult to improve the results on this dataset. The latest top results are only slightly different.

Experimental results in Table. IV were carried out on comparisons of our approach and state-of-the-art methods (UCI [53], DPM [54], LEO [55], DSO [56], CA [57], HoPS [58]). Our method obtained the average accuracy of 38.6%, which is highly competitive to these comparison methods (UCI: 27.1% [53], DPM: 33.7% [54], LEO: 29.6% [55], DSO: 34.7% [56], CA: 34.8% [57], HoPS: 35.4% [58]). Therefore, the proposed framework achieved the state-of-the-art performance.



(**a**) Polar transformation for input images

(**b**) Polar convolution for input images

Fig. 6: Polar transformation for convolutional operations of input layer: (**a**) Polar transformation for input images; (**b**) Polar convolution for input images.

In summary for the **Oi-HOG**, we present a novel variant of HOG, which has a simple feature extraction system and robust feature descriptors for rotation and reversal-invariant local feature representation. We have tested our proposed method with respect to visual tasks and validated its use on widely used and representative public databases. Our overall results show that this framework outperforms other state-of-the-art methods.

VI. The Orientation-invariant Representation for CNN Models

Motivated by the advantages of deep architecture and the success of polar model based orientation-invariant feature representations, we explore how to use polar structures on the convolutional layers and how this affects the performance of convolutional networks. Hence, we generalize the idea of polar data structure to state-of-the-art CNN models in sections VI and VII. In section VI, we propose polar convolution operations for CNN models, so that the neuron responses on prior-art CNN layers can be transformed into polar processing units. We aim to improve the orientation-invariant abilities for CNN models by embedding these polar operations in the convolutional architecture. We report the results of qualitative evaluation of the effect of incorporating polar structure into the CNN in VII.

It has been pointed out in [5], [7], [9] that the CNN model is sensitive to reversal. Meanwhile, Gong *et al.* [7] have proved that large-amplitude rotation could also lead to significant performance degradation in CNN models. To address these issues, one route to improving the recognition accuracy is by using data augmentation approaches [12], [60] in both the training and testing stages. However, the extended samples can not fully cover the complete space of variability spanned by both reversals and rotations situations and the choice of parameters. Meanwhile, preprocessing the data and learning from extended samples also require significant time and computational resources. As demonstrated in subsection III, since orientation-invariant capabilities are inherent in our polar descriptors, investigating similar architectures built upon the CNN model may lead to a successful solution to the aforementioned problems.

A. Polar Convolution Operation

We start with the polar transformation for each training sample. Before input into the CNN model, RGB channels of each sample are transformed into polar data using the local coordinate system that is detailed in Algorithm 1. After they are decomposed into the r and t orientations as shown in Fig. 6(a), the training samples can generate a set of images with a single channel of polar color data (six-type polar color data in total, see Fig.7). These polar color data are used as the CNN input after convolutional calculation with the original sample, in a similar manner as performed by the prior-art CNN model (R, G, B channels) doing. These are used as the CNN input after the convolutional computations with the original sample, in a similar manner to the prior-art CNN model (as shown in in Fig. 6(b). In addition, the polar transformation (color/convolutional) and data reading/writing instructions are executed in parallel in the same loop, and as a result the time complexity will not be increased.

B. The FPolarConv and LPolarConv

To generate an orientation-invariant CNN architecture, we propose two alternatives for the intermediate layers. The first is a full polar convolution (**FPolarConv**), while the second is a local polar convolution (**LPolarConv**). These two possibilities are illustrated in Figs. 8(a) and (b), respectively. As shown in Fig. 8(a), the **FPolarConv** is embedded into layers for global transformation to achieve an orientation-invariant representation. In contrast, the **LPolarConv** (Fig. 8(b)) is embedded into the local convolution operations (kernels) in each layer to produce orientation-invariant responses. In this way the prior-art CNN model can generate orientation-invariant convolutional features through adopting either the **FPolarConv** or **LPolarConv**.

We denote the input convolutional datasheets in the intermediate layer and the size of the sheets/ convolutional kernels in

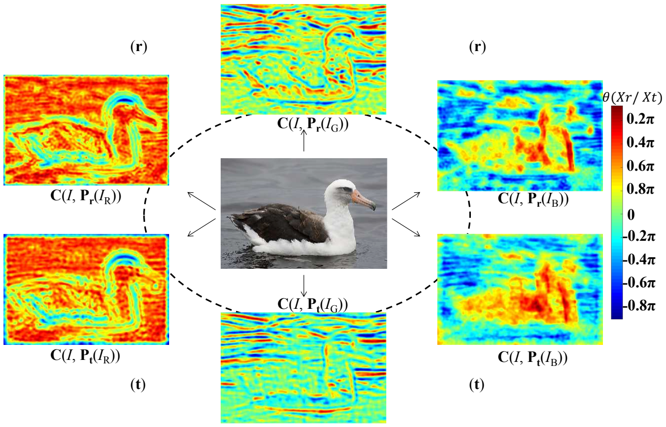


Fig. 7: Illustration of generated six-type convolutional images (single polar channel datasheets) for the CNN input. I_R, I_G, I_B correspond to red, green and yellow channel of I, respectively.

the layer as f and H, respectively. We also use label subscripts to distinguish the layers, and the layer number is indexed by l ($l > 1$) (see Fig 8).

FPolarConv After the feature vector is extracted from the input datasheet $f_{FP}(l-1)$ with conventional operations (kernels), we can obtain a set of $H_l \times H_l$, the intermediate conventional outputs, which are transformed into polar data with the local coordinate system (similar to Fig 3(b)) by a one-to-one mapping, as shown in Fig 8(a). The output function of the l-th **FPolarConv** layer is:

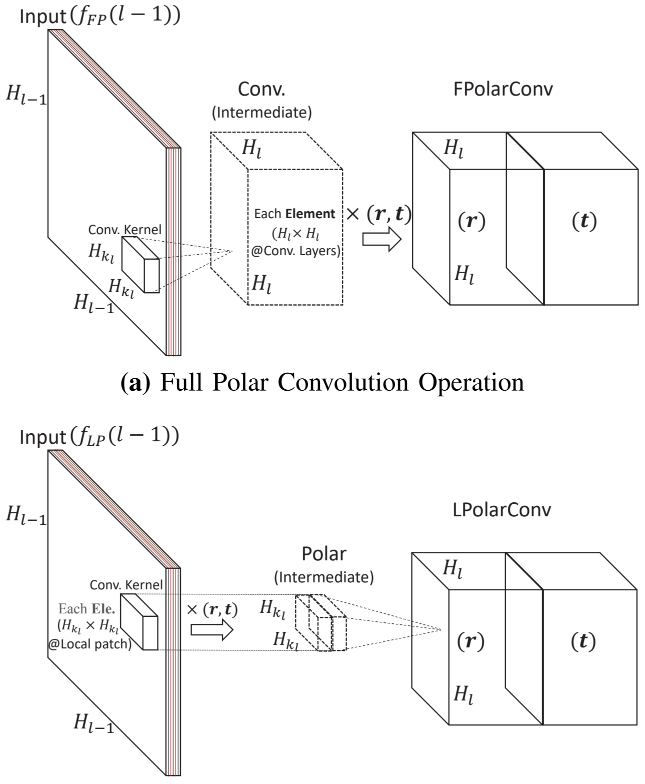
$$f_{FP}(l) = P(C(f_{FP}(l-1), K_{l,n})), \tag{4}$$

where $C(\cdot, K)$ indicates the conventional calculation with kernel K and $P(\cdot)$ denotes the polar transformation operation, which is detailed in Algorithm 1. Here $n = 1, \cdots N_l$ is the index number for the conventional kernel on the l-th layer. When the output of a given layer was reversed and rotated against its center of coordinates by a given angle θ, from **Theorem 1**, we can verify the output orientation invariance as follows,

$$\begin{aligned} \widehat{f}_{FP}(l) =& P(C(MR_\theta f_{FP}(l-1), K_{l,n})), \\ =& P(MR_\theta C(f_{FP}(l-1), K_{l,n})), \\ =& P(C(f_{FP}(l-1), K_{l,n})), \\ =& f_{FP}(l). \end{aligned} \tag{5}$$

In this paper, we adopt the deep features extracted from the first fully connected layer. Since the **FPolarConv** is embedded in the output for each layer, the CNN model, we can therefore produce orientation-invariant features.

LPolarConv In a manner different from the FPolarConv, LPolarConv transforms the elements in the local patch of conventional operations (obtained via kernels) into polar ones so as to generate polar kernels. These kernels are used to extract feature vectors from $l-1$-th layer, as shown in Fig 8(b). Consequently, the output-function in the l-th **LPolarConv** layer is:



(a) Full Polar Convolution Operation

(b) Local Polar Convolution Operation

Fig. 8: Two orientation-invariant solutions for CNN: **(a)** Full Polar Convolution (FPolarConv) Operation; **(b)** Local Polar Convolution (FPolarConv) Operation.

$$f_{LP}(l) = C(f_{LP}(l-1), P(K_{l,n})). \tag{6}$$

In the CNN model, each layer consists of a set of learnable kernels which have a small receptive field. This extends through the full depth of the input volume. The network learns kernel filters that can activate when it responds to some specific type of feature at some specific spatial position in the input. Each kernel is convolved across the width and height of the input volume, computing the dot product between the entries of the filter and the input-producing a two-dimensional activation map for that kernel. Consequently, the output of a given element (neuron response) η_{LP} in the l-th **LPolarConv** layer can be expressed as follows:

$$\begin{aligned}\eta_{LP} =&(\sum_{i=1}^{H_{K_l}}\sum_{j=1}^{H_{K_l}} \mathbf{g}_{i,j}^T k_{i,j}\mathbf{r}_{i,j}, \sum_{i=1}^{H_{K_l}}\sum_{j=1}^{H_{K_l}} \mathbf{g}_{i,j}^T k_{i,j}\mathbf{t}_{i,j})\\ =&(\sum_{i=1}^{H_{K_l}}\sum_{j=1}^{H_{K_l}} k_{i,j}\mathbf{g}_{i,j}^T\mathbf{r}_{i,j}, \sum_{i=1}^{H_{K_l}}\sum_{j=1}^{H_{K_l}} k_{i,j}\mathbf{g}_{i,j}^T\mathbf{t}_{i,j}),\end{aligned} \tag{7}$$

where $k \in K_{l,n}$ and $\mathbf{g}$ denotes the 1-norm Descartes vector, which is used to polarize the element at local point (i,j) of convolution kernel patch. Supposing the output feature in the l-th layer is orientation-invariant, its given neuron response η_{LP} can be calculated by

$$\eta_{FP} = (\mathbf{g'}^T\mathbf{r}', \mathbf{g'}^T\mathbf{t}'). \tag{8}$$

As a conventional kernel with the determined size will generate the same neuron responses for an image, but a conventional kernel with different sizes will generate different neuron responses, we need to prove that Eq. 8 holds with a given conventional kernel covering all kernel sizes. To do these we use mathematical induction as follows:

a)When $H_{K_l} = 1$, $\eta_{LP} = (k_{1,1}\mathbf{g}_{1,1}^T\mathbf{r}_{1,1}, k_{1,1}\mathbf{g}_{1,1}^T\mathbf{t}_{1,1})$. Obviously, the given neuron response η_{LP} can be rewritten in the same form as Eq. 8.

b)When $H_{K_l} = m-1$, $\eta_{FP} = (\mathbf{g}'^T_{m-1}\mathbf{r}', \mathbf{g}'^T_{m-1}\mathbf{t}')$ is established, namely, $\mathbf{g}'^T_{m-1}\mathbf{r}' = \sum_{i=1}^{m-1}\sum_{j=1}^{m-1} k_{i,j}\mathbf{g}_{i,j}^T\mathbf{r}_{i,j}$, $\mathbf{g}'^T_{m-1}\mathbf{t}' = \sum_{i=1}^{m-1}\sum_{j=1}^{m-1} k_{i,j}\mathbf{g}_{i,j}^T\mathbf{t}_{i,j}$ and $\mathbf{t}' = \mathbf{r}' \times R_{\frac{\pi}{2}}$.

c)When $H_{K_l} = m$, *i.e.*, padding the width of kernel $K_{(m-1)\times(m-1)}$ by one unit, η_{FP} would be calculated as follows:

$$\begin{aligned}\eta_{LP} =&(\sum_{i=1}^{m}\sum_{j=1}^{m} k_{i,j}\mathbf{g}_{i,j}^T\mathbf{r}_{i,j}, \sum_{i=1}^{m}\sum_{j=1}^{m} k_{i,j}\mathbf{g}_{i,j}^T\mathbf{t}_{i,j})\\ =&(\mathbf{g}'^T_{m-1}\mathbf{r}' + k_{i,j}\mathbf{g}_{m,m}^T\mathbf{r}_{m,m}, \mathbf{g}'^T_{m-1}\mathbf{t}' + k_{m,m}\mathbf{g}_{m,m}^T\mathbf{t}_{m,m}).\end{aligned} \tag{9}$$

Considering $\mathbf{t}' = \mathbf{r}' \times R_{\frac{\pi}{2}}$ and setting $\mathbf{g}''^T\mathbf{r}'' = \mathbf{g}'^T_{m-1}\mathbf{r}' + k_{m,m}\mathbf{g}_{m,m}^T\mathbf{r}_{m,m}$, we can simply derive Eqs.10 and 11 by

$$\begin{aligned}&\mathbf{g}'^T_{m-1}\mathbf{t}' + k_{i,j}\mathbf{g}_{m,m}^T\mathbf{t}_{m,m}\\ =&\mathbf{g}'^T_{m-1}\mathbf{r}'R_{\frac{\pi}{2}} + k_{m,m}\mathbf{g}_{m,m}^T\mathbf{r}_{m,m}R_{\frac{\pi}{2}}\\ =&(\mathbf{g}'^T_{m-1}\mathbf{r}' + k_{m,m}\mathbf{g}_{m,m}^T\mathbf{r}_{m,m})R_{\frac{\pi}{2}}\\ =&\mathbf{g}''^T\mathbf{r}''R_{\frac{\pi}{2}}\\ =&\mathbf{g}''^T\mathbf{t}'',\end{aligned} \tag{10}$$

i.e., η_{LP} can be rewritten as:

$$\eta_{LP} = (\mathbf{g}''^T\mathbf{r}'', \mathbf{g}''^T\mathbf{t}''). \tag{11}$$

In CNN training, each of the conventional calculations are executed by each kernel and the convolutional layer is the core building block of a CNN. Hence, according to **Theorem 1**, the generated polar data set on the output datasheet (feature vector set) in **LPolarConv** is invariant to orientation.

Deep features are extracted by a pre-trained CNN model in the application (test) stage. Hence, the deep features extracted from a test image can be considered as a set of neuron responses of a pre-trained model. Both the **FPolarConv** and **LPolarConv** approaches can generate orientation-invariant responses. Theoretically, we may conclude that the deep features extracted by the CNN model based on **FPolarConv** or **LPolarConv** will be invariant to reversal and rotation. The practical performance obtained with these two architectures, will be evaluated and compared experimentally in the next section.

VII. Experiments for P-CNN

In this section we detail the experimental setup and evaluation results for the new CNN models developed using polar representations. In order to compare with the latest and most relevant reversal-invariant representation model based on the CNN [5], we applied the proposed method to (fine-grained)

TABLE VII: The results on Aircraft dataset.

Method	Mean Accuracy (%)
Xie *et al.* [5] (BoF)	78.92
Vedaldi *et al.* [65]	54.80
Sanchez *et al.* [66]	73.26
Gosselin *et al.* [67]	**80.74**
Maji *et al.* [61]	48.69
AlexNet (no-Augm) [12]	44.68
Ri-Deep [5] (AVG, AlexNET (no-Augm))	52.74
Ri-Deep [5] (MAX, AlexNET (no-Augm))	53.06
AlexNet [12]	49.34
VGG-16 [15]	67.18
VGG-19 [15]	68.21
Ri-Deep [5] (AVG, VGG-19)	69.31
Ri-Deep [5] **(MAX, VGG-16)**	**69.44**
P-CNN (FPolarConv, AlexNet)	54.42
P-CNN (LPolarConv, AlexNet)	54.60
P-CNN (FPolarConv, VGG-16)	**72.66**
P-CNN (LPolarConv, VGG-16)	69.31
P-CNN (FPolarConv, VGG-19)	71.20
P-CNN (LPolarConv, VGG-19)	68.42

TABLE VIII: The results on Pet dataset.

Method	Mean Accuracy (%)
Xie *et al.* [5] **(BoF)**	**63.49**
Gosselin *et al.* [67]	49.82
Murray *et al.* [68]	56.80
Angelova *et al.* [69]	54.30
Sanchez *et al.* [66]	59.63
Wang *et al.* [70]	57.77
AlexNet (no-Augm) [12]	76.95
Ri-Deep [5] (AVG, AlexNET (no-Augm))	79.60
Ri-Deep [5] (MAX, AlexNET (no-Augm))	79.40
AlexNet [12]	80.85
VGG-16 [15]	93.09
VGG-19 [15]	93.10
Ri-Deep [5] **(AVG, VGG-16)**	**93.31**
Ri-Deep [5] (MAX, VGG-16)	93.25
P-CNN (FPolarConv, AlexNet)	81.93
P-CNN (LPolarConv, AlexNet)	82.60
P-CNN (FPolarConv, VGG-16)	93.21
P-CNN (LPolarConv, VGG-16)	93.55
P-CNN (FPolarConv, VGG-19)	93.29
P-CNN (LPolarConv, VGG-19)	**93.81**

TABLE IX: The results on Bird dataset.

Method	Mean Accuracy (%)
Xie *et al.* [5] (BoF)	50.81
Gosselin *et al.* [67]	45.71
Sanchez *et al.* [66]	47.63
Zhang *et al.* [71]	**50.98**
Girshick *et al.* [72]	51.05
Murray *et al.* [68]	33.30
AlexNet (no-Augm) [12]	43.50
Ri-Deep [5] (AVG, AlexNET (no-Augm))	47.98
Ri-Deep [5] (MAX, AlexNET (no-Augm))	47.82
AlexNet [12]	50.20
VGG-16 [15]	71.62
VGG-19 [15]	71.70
Ri-Deep [5] **(AVG, VGG-16)**	**72.66**
Ri-Deep [5] (MAX, VGG-19)	72.59
P-CNN (FPolarConv, AlexNet)	52.36
P-CNN (LPolarConv, AlexNet)	57.06
P-CNN (FPolarConv, VGG-16)	72.77
P-CNN (LPolarConv, VGG-16)	75.20
P-CNN (FPolarConv, VGG-19)	73.74
P-CNN (LPolarConv, VGG-19)	**76.93**

image classification following their reported procedures. For the results reported in this section, we have implemented all the training and detection programs on powerful GPU cards with 12 GB of RAM. We report the experimental results on the FGVC-Aircraft dataset (Aircraft) [61], the Oxford-IIIT Pet dataset (Pet) [62], and the Caltech-UCSD Birds-200-2011 dataset (Bird) [63].

A. Experimental Dataset and Implementation

The Aircraft dataset contains 10,200 images of aircraft with 100 images for each of 102 different aircraft model variants. The Pet dataset includes 7,349 images, covering 37 different breeds of cats and dogs. The Bird dataset has 11,788 images, containing 200 categories of birds. These referenced databases are widely used for evaluating classification models on fine-grained object recognition.

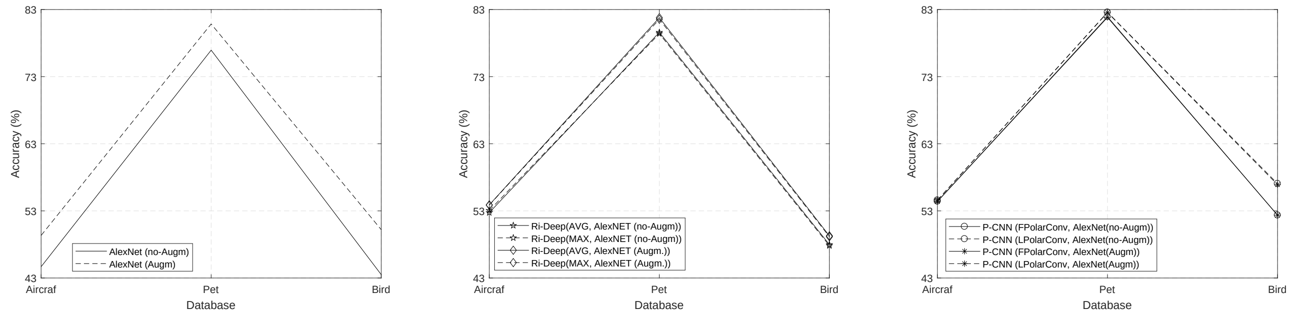
The basic experimental protocol follows that used for the evaluation of the latest proposed reversal-invariant representation model [5]. In all of our experiments, we use FPolarConv and LPolarConv to transform the data structure into polar layers for evaluation. The resulting polar CNN models are implemented based on the state-of-the-art CNN modes AlexNet [12], VGG-16 and VGG-19 nets [15], which are provided in the MatConvNet toolbox [64]. For comparison, we use the pre-trained models of AlexNet, VGG-16 and VGG-19, which are downloaded from the library of MatConvNet(pre-trained models) [64] trained by adding augmented data. For pre-training the proposed polar CNN models, we do not use any augmentation data.

B. Experimental Results of Image Classification

In this study, the input image is resized to the resolution of 256×256, which is the same as the original procedure for testing stage [12]. Then the resized image is used for polar transformations. However, we do not use sub images that are cropped from different positions for calculating the average response. For VGG-16 and VGG-19, the input image is resized to 512×512 following the original strategy so that it can provide enough and reliable responses for deep down-sampling (declining layer by layer) and achieve better performance. After the neuron responses are calculated, we extracted deep features from each layer by mean-pooling and max-pooling for **FPolarConv** and **LPolarConv**, respectively. Finally, we obtained the features extracted from the first fully connected layer activated by the rectified linear units (ReLu) [12]. The extracted feature vectors are normalized with l_2 normalization and fed into support vector machines (SVMs) for learning.

The results are summarized in Tables VII, VIII, and IX for Aircraft, Pet, and Bird datasets, respectively. These tables are divided into three blocks which report a) the prior-art Bag-of-features-based (BoF-based) image representation models, b) state-of-the-art deep-feature models and c) the proposed models, respectively.

Through analyzing the results, we find that the deep features outperform BoF-based models in almost all tasks except the Aircraft dataset. The reason is that the pre-trained CNN models are trained by using the ILSVR2012 dataset [73] which



(a) The AlexNet

(b) Ri-Deep

(C) The P-CNN

Fig. 9: Investigation about how augmentation data affect each CNN model in the accuracy: **(a)** The original AlexNet: the recognition rate is improved by augmented data; **(b)** The reversal-invariant AlexNet transformed by the Ri-Deep: the recognition rate is slightly improved by augmented data; **(b)** The orientation-invariant AlexNet transformed by the proposed P-CNN: the recognition rate is almost not improved according to augmented data

does not include the aircraft images. Hence the model pre-trained on the ILSVR2012 fails to generate the best response for the aircraft features. This in turn limits the discriminative power of the deep features extracted from the aircraft images. In contrast, for the Pet and Bird databases, *i.e.*, the ILSVR2012 dataset covers many examples of pet images and a certain number of bird images. As a result the accuracy obtained on the Pet database > on the Bird database > that on the Aircraft database.

To compare with the prior-art CNN model, we list the results for AlexNet, VGG-16, and VGG-19 obtained using the pre-trained models. To check how the augmentation data affect the accuracies obtained in the CNN models, we also show the results obtained with the pre-trained model without augmented data for AlexNet (AlexNet (no-Augm)), which are provided in Xie *et al.*'s publication [5]. For training the proposed polar models, we also use the ILSVRC2012 dataset. This is a subset of the ImageNet database covering 1000 object categories. Results [top 1 error, top 5 error] (without augmented data) of **FPolarConv** and **LPolarConv** are respectively [41.31%, 18.68%] and [41.26%, 18.61%]. These are slightly reduced with respect to the standard error results provided by MatConvNet (with augmented data) [64] by [0.49%, 0.52%] and [0.54%, 0.59%]. Since ILSVRC2012 is a large dataset and the original results are obtained by using augmented data, these improvements are not so small. We can see that the proposed P-CNN models outperform the prior-art deep models. Moreover, the state-of-the-art invariant-transformation method for CNN (Ri-Deep) [5] more or less gives improvements in all cases by using the extended data (reversal) shown in Fig 9. One of the most important reasons for achieving the improved performance is that the reversed augmentation data creates many large-amplitude samples for training. In contrast, the performances of the proposed models are rarely improved by augmented data. This confirms the effectiveness of the proposed methods and proves that the orientation-invariant capabilities of P-CNN are better than both Ri-Deep together with its original model.

We also find that the performance for the LPolarConv model is better than that for the FPolarConv model, particularly in the cases of the Bird and Pet data. Analysis of the images in the Bird and Pet databases, reveal that there are many local image reversals. In other words, there are many images containing body-parts (*e.g.*, head or tail) that are reversed or rotated versions. The explanation is that the polar local operations (kernels) can flexibly and successfully generate invariant responses for any small sub-parts. Therefore, the accuracy of LPolarConv in the Bird database is better than that obtained with FPolarConv. However, the aircraft seldom has reversed or rotated parts. So the results obtained with both models are nearly the same. These results also give us some insight into the reasons that explain why local features are more robust than global features in many applications of handcrafted descriptors.

In summary, we adopt the P-CNN approach to transform CNN model into polar one so as to produce orientation-invariant features. We propose two polar convolutional operations, namely **FPolarConv** and **LPolarConv**. These two operations can both be implemented with the state-of-the-art CNN frameworks (AlexNet, VGG-16, VGG-19), which show significant advantages in image feature representation in our evaluation experiments. The experimental results prove that our proposed solutions are able to improve performances of CNN models.

VIII. Conclusions

In this paper, we propose polar-transformation as a solution to improve orientation-invariant image representations. The technical contents are divided into two parts for a) handcrafted descriptors and b) deep features.

As for handcrafted descriptors, we use gradient descriptors for orientation-invariant transformation, which are used to extract HOG-type features (**Oi-HOG**) for recognition tasks. The expression results show that **Oi-HOG** succeeded in out-of-plane head orientation cases. For deep features, we adopt the AlexNet, VGG-16 and VGG-19 for orientation-invariant transformation. We apply the approach to transforming RGB images into polar images to form a CNN input layer. Furthermore, we use the **FPolarConv** and **LPolarConv** polar convolutions to construct polar-data structures in the convolutional layers. Here is seems that the performance of **LPolarConv** is slightly better than **FPolarConv**. Experimental results clearly demonstrate the superiority of our framework when compared with current state-of-the-art methods. From studies of convolutional networks of increasing depth we also confirm tha a significant improvement in the classification performance can be achieved by increasing the number of layers or constructing parallel networks. In this study, demonstrate that the data structure on the CNN layers is key to improving the image classification performance. Meanwhile, the raw input data have advantages for deep learning and we usually adopt RGB as input data for training. However, in this study, we found that there are some data (*e.g.*, polar data) that are better than RGB to gain more candidate representation parameters for deep learning. These would be important to those with closely-related research interests.

Acknowledgements

We would like to thank Dr. Yasuo Ariki (Emeritus Professor at Kobe University, Japan), who has given many helpful advices in this study. We also sincerely thank the reviewers who participated in the peer review process for this paper. Their constructive comments helped to improve this paper.

This work was supported in part by PRESTO, JSTA (Grant No. JPMJPR15D2), Japan Society for the Promotion of Science KAKENHI (Grant No. JP17H01995), National Natural Science Foundation of China (Grant No.61402389), the Natural Science Foundation of Fujian Province (Grant No. 2017J01497) and the Fundamental Research Funds for the Central Universities in China (No. 20720160073).

References

[1] W. L. Zhao, C. W. Ngo, and H. Wang, "Fast covariant vlad for image search," *IEEE Trans. Multimedia (TMM)*, vol. 18, no. 9, pp. 1843–1854, Sept 2016.

[2] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2. IEEE, 2005, pp. 524–531.

[3] G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos, "Supervised learning of semantic classes for image annotation and retrieval," *IEEE Trans. Patt. Analy. Mach. Intell. (TPAMI)*, vol. 29, no. 3, pp. 394–410, 2007.

[4] C. Galleguillos, A. Rabinovich, and S. Belongie, "Object categorization using co-occurrence, location and appearance," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*. IEEE, 2008, pp. 1–8.

[5] L. Xie, J. Wang, W. Lin, B. Zhang, and Q. Tian, "Towards reversal-invariant image representation," *Int. J. Comput. Vis. (IJCV)*, vol. 123, no. 2, pp. 226–250, Jun 2017.

[6] H. Mller and D. Unay, "Retrieval from and understanding of large-scale multi-modal medical datasets: A review," *IEEE Trans. Multimedia (TMM)*, vol. 19, no. 9, pp. 2093–2104, Sept 2017.

[7] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 392–407.

[8] J. Chen, T. Nakashika, T. Takiguchi, and Y. Ariki, "Content-based image retrieval using rotation-invariant histograms of oriented gradients," in *Proc. Int. Conf. on Multimedia Retrieval (ICMR)*, 2015, pp. 443–446.

[9] J. Lin, L. Y. Duan, S. Wang, Y. Bai, Y. Lou, V. Chandrasekhar, T. Huang, A. Kot, and W. Gao, "Hnip: Compact deep invariant representations for video matching, localization, and retrieval," *IEEE Trans. Multimedia (TMM)*, vol. 19, no. 9, pp. 1968–1983, Sept 2017.

[10] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Hetection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2005, pp. 886–893 vol. 1.

[11] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*. IEEE, 2010, pp. 2528–2535.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Proc. Syst. (NIPS)*. Curran Associates, Inc., 2012, pp. 1097–1105.

[13] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

[14] L. Sun, K. Jia, T.-H. Chan, Y. Fang, G. Wang, and S. Yan, "Dl-sfa: deeply-learned slow feature analysis for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2014, pp. 2625–2632.

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[16] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[17] Q. V. Le, A. Karpenko, J. Ngiam, and A. Y. Ng, "Ica with reconstruction cost for efficient overcomplete feature learning," in *Proc. Adv. Neural Inf. Proc. Syst. (NIPS)*, 2011, pp. 1017–1025.

[18] J. Chen, S. Shan, C. He, G. Zhao, M. Pietikainen, X. Chen, and W. Gao, "Wld: A robust local image descriptor," *IEEE Trans. Patt. Analy. Mach. Intell. (TPAMI)*, vol. 32, no. 9, pp. 1705–1720, 2010.

[19] G. M. Haley and B. Manjunath, "Rotation-invariant texture classification using modified gabor filters," in *Proc. IEEE Int. Conf. Image Proces. (ICIP)*, vol. 1. IEEE, 1995, pp. 262–265.

[20] J. Han and K.-K. Ma, "Rotation-invariant and scale-invariant gabor features for texture image retrieval," *Image and vision computing*, vol. 25, no. 9, pp. 1474–1481, 2007.

[21] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis. (IJCV)*, vol. 60, no. 2, pp. 91–110, Nov 2004.

[22] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Trans. Patt. Analy. Mach. Intell. (TPAMI)*, vol. 28, no. 12, pp. 2037–2041, 2006.

[23] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-Up Robust Features (SURF)," *Comput. Vis. Image Underst. (CVIU)*, vol. 110, no. 3, pp. 346 – 359, 2008.

[24] M. Bicego, A. Lagorio, E. Grosso, and M. Tistarelli, "On the use of sift features for face authentication," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshop*. IEEE, 2006, pp. 35–35.

[25] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz, "Multi-view stereo for community photo collections," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*. IEEE, 2007, pp. 1–8.

[26] K. Liu, H. Skibbe, T. Schmidt, T. Blein, K. Palme, T. Brox, and O. Ronneberger, "Rotation-invariant hog descriptors using fourier analysis in polar and spherical coordinates," *Int. J. Comput. Vis. (IJCV)*, vol. 106, no. 3, pp. 342–364, 2014.

[27] C. Thurau and V. Hlavac, "Pose Primitive Based Human Action Recognition in Videos or Still Images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2008, pp. 1–8.

[28] A. Klaeser, M. Marszalek, and C. Schmid, "A Spatio-temporal Descriptor Based on 3D-gradients," in *Proc. British Machine Vis. Conf. (BMVC)*, 2008, pp. 99.1–99.10.

[29] H. Jhuang, T. Serre, L. Wolf, and T. Poggio, "A Biologically Inspired System for Action Recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct 2007, pp. 1–8.

[30] G. Takacs, V. Chandrasekhar, S. Tsai, D. Chen, R. Grzeszczuk, and B. Girod, "Fast computation of rotation-invariant image features by an approximate radial gradient transform," *IEEE Trans. Image Proc.(TIP)*, vol. 22, no. 8, pp. 2970–2982, Aug. 2013.

[31] D. A. Van Dyk and X.-L. Meng, "The art of data augmentation," *Journal of Computational and Graphical Statistics*, vol. 10, no. 1, pp. 1–50, 2001.

[32] D. Laptev, N. Savinov, J. M. Buhmann, and M. Pollefeys, "Ti-pooling: transformation-invariant pooling for feature learning in convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 289–297.

[33] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Proc. Syst. (NIPS)*, 2015, pp. 2017–2025.

[34] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Proc. Syst. (NIPS)*, 2014, pp. 2672–2680.

[35] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[36] Y. Peng, X. He, and J. Zhao, “Object-part attention model for fine-grained image classification,” *IEEE Trans. Image Proc.(TIP)*, vol. 27, no. 3, pp. 1487–1500, March 2018.

[37] Y. Zhou, Q. Ye, Q. Qiu, and J. Jiao, “Oriented response networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, July 2017, pp. 4961–4970.

[38] M. F. Valstar and M. Pantic, “Induced Disgust, Happiness and Surprise: an Addition to the MMI Facial Expression Database,” in *Proc. of Int. Conf. Language Resources and Evaluation, Workshop on EMOTION*, May 2010, pp. 65–70.

[39] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, “The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshops*, 2010, pp. 94–101.

[40] A. Dhall, R. Goecke, S. Lucey, and T. Gedeon, “Collecting Large, Richly Annotated Facial-Expression Databases from Movies,” *MultiMedia, IEEE*, vol. 19, no. 3, pp. 34–41, July 2012.

[41] J. Chen, Z. Luo, T. Takiguchi, and Y. Ariki, “Multithreading Cascade of SURF for Facial Expression Recognition,” *EURASIP Journal on Image and Video Processing*, vol. 2016, no. 1, p. 37, 2016.

[42] J. Chen, Y. Ariki, and T. Takiguchi, “Robust Facial Expressions Recognition Using 3 D Average Face and Ameliorated Adaboost,” in *Proc. ACM Multimedia Conf. (MM)*, 2013, pp. 661–664.

[43] T. Trzcinski, M. Christoudias, and V. Lepetit, “Learning Image Descriptors with Boosting,” *IEEE Trans. Patt. Analy. Mach. Intell. (TPAMI)*, vol. 37, no. 3, pp. 597–610, March 2015.

[44] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun, “Joint Cascade Face Detection and Alignment,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 109–122.

[45] L. Bourdev and J. Brandt, “Robust Object Detection via Soft Cascade,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, June 2005, pp. 236–243.

[46] G. Zhao and M. Pietikainen, “Dynamic Texture Recognition Using Local Binary Patterns with an Application to Facial Expressions,” *IEEE Trans. Patt. Analy. Mach. Intell. (TPAMI)*, vol. 29, no. 6, pp. 915–928, June 2007.

[47] L. Wang, Y. Qiao, and X. Tang, “Motionlets: Mid-level 3D Parts for Human Motion Recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2013, pp. 2674–2681.

[48] S. Chew, P. Lucey, S. Lucey, J. Saragih, J. Cohn, and S. Sridharan, “Person-independent Facial Expression Detection Using Constrained Local Models,” in *FG*, March 2011, pp. 915–920.

[49] M. Liu, S. Li, S. Shan, R. Wang, and X. Chen, “Deeply Learning Deformable Facial Action Parts Model for Dynamic Expression Analysis,” in *Proc. Asia Conf. Comput. Vis. (ACCV)*, vol. 9006, 2014, pp. 143–157.

[50] M. Liu, S. Shan, R. Wang, and X. Chen, “Learning Expressionlets on Spatio-temporal Manifold for Dynamic Facial Expression Recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2014, pp. 1749–1756.

[51] P. Scovanner, S. Ali, and M. Shah, “A 3-dimensional Sift Descriptor and Its Application to Action Recognition,” in *Proc. ACM Multimedia Conf. (MM)*, 2007, pp. 357–360.

[52] O. Rudovic, V. Pavlovic, and M. Pantic, “Multi-output Laplacian Dynamic Ordinal Regression for Facial Expression Recognition and Intensity Estimation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2012, pp. 2634–2641.

[53] C. Desai, D. Ramanan, and C. Fowlkes, “Discriminative models for multi-class object layout,” *Int. J. Comput. Vis. (IJCV)*, vol. 95, no. 1, pp. 1–12, 2011.

[54] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, “Object Detection with Discriminatively Trained Part-Based Models,” in *IEEE Trans. Patt. Analy. Mach. Intell. (TPAMI)*, vol. 32, no. 9, Sept 2010, pp. 1627–1645.

[55] L. Zhu, Y. Chen, A. Yuille, and W. Freeman, “Latent hierarchical structural learning for object detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2010, pp. 1062–1069.

[56] X. Wang, L. Lin, L. Huang, and S. Yan, “Incorporating Structural Alternatives and Sharing into Hierarchy for Multiclass Object Recognition and Detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2013, pp. 3334–3341.

[57] F. Shahbaz Khan, R. Anwer, J. van de Weijer, A. Bagdanov, M. Vanrell, and A. Lopez, “Color Attributes for Object Detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2012, pp. 3306–3313.

[58] W. Voravuthikunchai, B. Cremilleux, and F. Jurie, “Histograms of Pattern Sets for Image Classification and Object Recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2014, pp. 224–231.

[59] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman, “The Pascal Visual Object Classes (VOC) Challenge,” in *Int. J. Comput. Vis. (IJCV)*, vol. 88, no. 2, 2010, pp. 303–338.

[60] Y. Chai, V. Lempitsky, and A. Zisserman, “Symbiotic segmentation and part localization for fine-grained categorization,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec 2013, pp. 321–328.

[61] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi, “Fine-grained visual classification of aircraft,” Tech. Rep., 2013.

[62] O. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar, “Cats and dogs,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2012, pp. 3498–3505.

[63] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, “The Caltech-UCSD Birds-200-2011 Dataset,” Tech. Rep., 2011.

[64] A. Vedaldi and K. Lenc, “Matconvnet: Convolutional neural networks for matlab,” in *Proc. ACM Multimedia Conf. (ACM MM)*, 2015, pp. 689–692.

[65] A. Vedaldi, S. Mahendran, S. Tsogkas, S. Maji, R. Girshick, J. Kannala, E. Rahtu, I. Kokkinos, M. Blaschko, D. Weiss, B. Taskar, K. Simonyan, N. Saphra, and S. Mohamed, “Understanding objects in detail with fine-grained attributes,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2014, pp. 3622–3629.

[66] “Image classification with the fisher vector: Theory and practice,” *Int. J. Comput. Vis. (IJCV)*, vol. 105, no. 3, 2013.

[67] “Revisiting the fisher vector for fine-grained classification,” *Pattern Recognition Letters*, vol. 49, pp. 92 – 98, 2014.

[68] N. Murray and F. Perronnin, “Generalized max pooling,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2014, pp. 2473–2480.

[69] A. Angelova and S. Zhu, “Efficient object detection and segmentation for fine-grained recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2013, pp. 811–818.

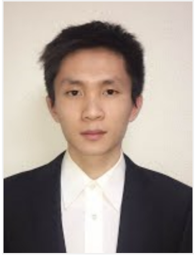
[70] “Collaborative linear coding for robust image classification,” *Int. J. Comput. Vis. (IJCV)*, vol. 114, no. 2-3, 2015.

[71] N. Zhang, R. Farrell, F. Iandola, and T. Darrell, “Deformable part descriptors for fine-grained recognition and attribute prediction,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2013, pp. 729–736.

[72] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2014, pp. 580–587.

[73] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *Int. J. Comput. Vis. (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

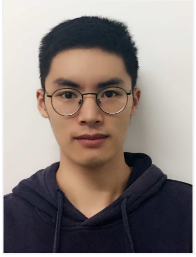
Jinhui Chen received his Ph.D degree in information science at Kobe University, Japan, in 2016. He is currently an assistant professor at Kobe University. His research interests include pattern recognition and machine learning. He is a member of IEEE, ACM, and IEICE. He has published more than 20 publications in major journals and international conferences, such as IEEE Trans. Multimedia, ACM MM, ACM ICMR, *etc.*



Zhaojie Luo Zhaojie Luo is a Ph.D. candidate in computer science at Kobe University, Kobe, Japan. He received his M.E.degree from the same university in 2017. His research interests are speech recognition, statistical signal processing and machine learning. He is a member of IEEE, ISCA and ASJ.

Zhihong Zhang received his BSc degree (1st class Hons.) in computer science from the University of Ulster, UK, in 2009 and the PhD degree in computer science from the University of York, UK, in 2013. He won the K. M. Stott prize for best thesis from the University of York in 2013. He is now an associate professor at the software school of Xiamen University, China. His research interests are wide-reaching but mainly involve the areas of pattern recognition and machine learning, particularly problems involving graphs and networks.

Faliang Huang received the Ph.D. degree in data mining from the South China University of Technology, Guangdong Sheng, China, in 2011. He is now an Associate Professor in the College of Mathematics and Informatics, Fujian Normal University, Fuzhou Shi, China. His research interests include data mining and natural computing.



Zhiling Ye received his B.S. degrees in the school of mathematical sciences from Qingdao University, China in 2015. He is currently a third-year graduated school student at the Xiamen University, China. His research interests are computer vision and machine learning.

Tetsuya Takiguchi received his B.S. degree in applied mathematics from Okayama University of Science, Okayama, Japan, in 1994, and his M.E. and Dr. Eng. degrees in information science from Nara Institute of Science and Technology, Nara, Japan, in 1996 and 1999, respectively. From 1999 to 2004, he was a researcher at IBM Research, Tokyo Research Laboratory, Kanagawa, Japan. He is currently a professor at Kobe University. His research interests include statistic signal processing and pattern recognition. He received the award from the Acoustical Society of Japan in 2002. He is a member of IEEE, IPSJ and ASJ.

Edwin R. Hancock holds a BSc degree in physics (1977), a PhD degree in high-energy physics (1981) and a D.Sc. degree (2008) from the University of Durham, and a doctorate Honoris Causa from the University of Alicante in 2015. He is Professor in the Department of Computer Science, where he leads a group of some faculty, research staff, and PhD students working in the areas of computer vision and pattern recognition. His main research interests are in the use of optimization and probabilistic methods for high and intermediate level vision. He is a fellow of the International Association for Pattern Recognition and the IEEE. He is currently Editor-in-Chief of the journal Pattern Recognition, and was founding Editor-in-Chief of IET Computer Vision from 2006 until 2012. He has also been a member of the editorial boards of the journals IEEE Transactions on Pattern Analysis and Machine Intelligence, Pattern Recognition, Computer Vision and Image Understanding, Image and Vision Computing, and the International Journal of Complex Networks. He is currently Vice President of the IAPR.