2016

Scene categorization under geometric deformations

Xue Wei
University of Wollongong

Scene Categorization under Geometric Deformations

A thesis submitted in partial fulfilment of the requirements for the award of the degree

Doctor of Philosophy

by

Xue Wei

School of Electrical, Computer and Telecommunications Engineering

UNIVERSITY OF WOLLONGONG

March 2016

Statement of Originality

I, Xue Wei, declare that this thesis, submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, is wholly my own work unless otherwise referenced or acknowledged. The document has not been submitted for qualifications at any other academic institution.

Xue Wei

March, 2016

Contents

List of Figures

List of Tables

Acronyms

2-D	Two-dimensional
3-D	Three-dimensional
AUC	Area under ROC curve
BIF	Biologically-inspired features
BOP	Bag-of-parts
BoW	Bag-of-words
CENTRIST	Census transform histogram
CNN	Convolutional neural network
DeCAF	Deep convolutional activation feature
GIST	An abstract representation of the scene
HIK	Histogram intersection kernel
HMAX	Hierarchical model and X
HOG	Histogram of oriented gradients

HOG-SPM HOG with spatial pyramid matching

HSOG Histogram of the second-order gradients

HVS Human visual system

IT Inferior temporal

LBP Local binary pattern

LBP-HF LBP with Fourier histogram

LCS Local color statistic

LGN Lateral geniculate nucleus

LIP Lateral intraparietal

OB Object bank

PCA Principal components analysis

PFC Prefrontal cortex

PLBP LBP with pyramid representation

PPA Parahippocampal place area

RBF Radial basis function kernel

ROC Receiver operating characteristic

SEV Sequential edge vectors

SIFT Scale-invariant feature transform

SIFT-FV	SIFT with Fisher vector
SIFT-LLC	SIFT with locality-constrained linear coding
SIFT-ScSPM	SIFT with sparse coding based spatial pyramid matching
SIFT-SPM	SIFT with spatial pyramid matching
SURF	Speeded up robust features
SVM	Support vector machine

Abstract

Humans are endowed with the ability to grasp the overall meaning or the gist of a complex visual scene at a glance. We need only a fraction of a second to decide if a scene is indoors, outdoors, on a busy street, or on a clear beach. In recent years, computational gist recognition or scene categorization has been actively pursued, given its numerous applications in image and video search, surveillance, and assistive navigation. Many visual descriptors have been developed to address the challenges in scene categorization, including the large number of semantic categories and the tremendous variations caused by imaging conditions. However, the existing methods for scene categorization still have difficulties to recognize images undergone geometric deformations, such as translation, scaling, shearing, rotation, and projection.

A major goal of a visual system (natural or machine) is to recognize objects or scenes, regardless of their location or pose relative to the viewer. Furthermore, the geometric invariances are required not only for scene categorization, but also for many other computer vision applications, including handwritten digit recognition, texture recognition, face matching, and face recognition. Therefore, extracting geometric invariance is a key for efficient image recognition.

This thesis investigates a geometric-invariant visual system to determine the categories of images. The proposed visual system achieves the geometric invariance through image normalization and feature extraction. A novel image approach to normalize affine deformations is presented in this thesis. The proposed approach produces normalized images by solving a constrained optimization problem based on image moments. An image normalization approach for projective deformations is also proposed. The image normalization methods allow geometric-invariant features to be extracted, thereby reducing the complexity of scene classifiers and the cost of classifier training. Visual descriptors used for scene categorization are reviewed in this thesis, from both methodological and experimental perspectives. Different visual descriptors are also combined to improve the scene categorization performance under geometric deformations.

Acknowledgments

First, I want to thank my principal supervisor, Associate Professor Son Lam Phung, for all of his guidance, counsel, encouragement and technical support.

Special thanks also go to my co-supervisor Professor Abdesselam Bouzerdoum for all his time, assistance, and knowledge.

Moreover, I gratefully acknowledge the ongoing support of the staff of the School of Electrical, Computer and Telecommunications Engineering for giving me personal and professional support during my studies at the University of Wollongong.

Thanks to my fellow students and friends, who have helped me during my study at the University.

And finally, I would like to express my gratitude to my Parents and Zhuo Chen, who have supported me during my studies and research projects.

Chapter 1

Introduction

Chapter contents

1.1 Research objectives

Humans can grasp rapidly the overall meaning of a complex visual scene. With a single glance, they can determine whether they are looking at a room, a beach, or a forest [1]. Viewers need only a fraction of a second to associate a picture with an abstract concept such as girl clapping or busy street [2]. Furthermore, humans can recognize objects from different viewpoints and in different arrangements. This ability to understand the conceptual meaning of a scene at a glance, regardless of its visual complexity and without attention to details, is known as *gist recognition*.

In computer vision, gist recognition is also known as *scene categorization*, which aims to classify a scene into semantic categories [3, 4, 5, 6, 7]. The scene could be a static image or dynamic video, and the semantic categories could be indoor

versus outdoor, gas station versus restaurant, or slow traffic versus flowing traffic. Scene categorization can be used to provide cues about objects and actions, detect abnormal events in public places, sense dangerous situations, and search for images and video; therefore, it is highly useful for applications in surveillance [8, 9], navigation [10, 11, 12, 13], and multimedia [14, 15, 16, 17, 18].

A good computational scene categorization system must possess discriminative power to characterize different semantic categories, while remaining stable in the presence of inter- and intra-class variations caused by photometric and geometric image distortions. The overall goal of this project is to develop a vision system to automatically recognize images that have undergone geometric distortions. Our approach allows geometric-invariant features to be extracted after image normalization, thereby reducing the complexity of scene classifiers and the cost of classifier training. The novelty of this project is the use of image normalization to achieve geometric invariants. Two image normalization approaches based on new moment constraints are proposed. The project is a step towards developing a view-invariant scene categorization system.

This thesis addresses two topics: scene categorization and image normalization. Chapter 2 and 3 provide theoretical and experimental evaluations of scene categorizations. Chapters 4 and 5 addresses image normalization algorithms for geometric distortions. Chapter 6 applies image normalization algorithm for scene categorization. Chapter 7 is the conclusion.

The specific aims of the project are to:

- Evaluate and investigate approaches to extract visual features for scene categorization.

- Investigate algorithms that normalize images under affine-deformations.
- Develop algorithms that normalize images under projective-deformations.
- Analyze geometric normalization effects on image class separability.
- Analyze geometric normalization effects on scene categorization.

1.2 Thesis organization

This thesis consists of seven chapters:

- **Chapter 1** outlines the project background and objectives. It highlights the research contributions and publications.
- **Chapter 2** gives a literature review on the human visual system and computational feature extraction for scene categorization. The computational descriptors for scene categorization are grouped into three categories: biologically-inspired features, local features, and global feature formation.
- **Chapter 3** presents the experimental results of visual descriptors for scene categorization. The existing bench mark data sets and performance measures for scene categorization are also discussed.
- **Chapter 4** presents the proposed image normalization method for affine deformations based on new moment constraints. The proposed method computes the normalization matrix T by solving an optimization problem in one step.
- **Chapter 5** describes the proposed image normalization method for projective deformations. The proposed algorithm allows an image with arbitrary

projective distortions to be recognized efficiently. A two-stage approach is present to calculate the 8 parameters of the required projective transformation matrix using image moments.

- **Chapter 6** presents the scene categorization method under geometric deformations. We investigate the effects of the proposed image normalization methods on several state-of-the-art visual descriptors for scene categorization. We also combine different visual descriptors to improve the scene categorization performance.

- **Chapter 7** summaries the research activities and provides the concluding remarks.

1.3 Contributions

The principal contributions of this thesis are listed as follows.

- A literature review on visual descriptors for scene categorization is provided from both methodological and experimental perspectives. The human visual system is also studied to inspire the computational vision models. The existing computational approaches for visual feature extraction in scene categorization can be divided into three broad categories: biologically-inspired methods, local features, and global features.

- A novel moment-based image normalization method is proposed to achieve fully affine invariants. The proposed approach produces normalized images by solving an optimization problem based on image moments. The moment propositions used in our normalization method are presented and proved.

- A novel moment-based image normalization method is proposed to achieve projective invariants. We present a two-stage approach to calculate the projective transformation matrix. The proposed normalization method produces the same set of normalized images for projective distorted images.

- A scene categorization method that is invariant to geometric distortions is proposed. It contains three steps: image normalization, feature extraction, and classification. We combine different visual descriptors and investigate their scene categorization performance under geometric deformations.

1.4 Publications

The publications arising from this project (March 2012 - March 2016) are listed as follows.

- X. Wei, S. L. Phung, A. Bouzerdoum, A. Bermak, "Invariant Image Recognition under Projective Deformations: An Image Normalization Approach", *IEEE International Conference on Visual Communications and Image Processing*, pp. 1-4, 2015, Singapore.

- X. Wei, S. Phung, and A. Bouzerdoum, "Visual descriptors for scene categorization: experimental evaluation," *Artificial Intelligence Review*, vol. 45, pp. 333-368, 2015.

- X. Wei, S. Phung, and A. Bouzerdoum, "Affine-invariant scene categorization," *IEEE International Conference on Image Processing*, pp. 1031-1035, 2014, Paris.

- X. Wei, S. L. Phung, and A. Bouzerdoum, "Object segmentation and classification using 3-D range camera," *Journal of Visual Communication and Image Representation*, vol. 25, pp. 74-85, 2014.

Chapter 2

Review of visual descriptors for scene categorization

Chapter contents

2.1 Introduction

This chapter * reviews the descriptors used for scene categorization. Humans possess a remarkable ability of grasping rapidly the overall meaning or the gist

*Chapter 2 and 3 have been published in our paper "Visual descriptors for scene categorization: experimental evaluation," *Artificial Intelligence Review*, vol. 45, no.3, pp. 333-368, 2016.

of a complex visual scene. This perception occurs even before time is sufficient for recognizing individual objects in the scene. Viewers need only about 32ms to judge if a scene is indoor or outdoor [1], or 250ms to associate a picture with an abstract concept such as a clapping girl, a busy street, or a clear beach [2]. Gist recognition is defined as the ability to understand the conceptual meaning of a scene at a glance, regardless of its visual complexity and without attention to details. The *gist* of a scene plays many roles in visual perception. It guides the viewer's attention, aids object recognition, and affects the viewer's recollection of the scene [19].

Behavioral experiments confirm that the accuracy of gist recognition increases with exposure time. For example, in Potter's experiment on human subjects, the recognition accuracy was about 79% after 125 ms exposure, and it increased to 90% after 333 ms exposure [20]. Renninger and Malik conducted an experiment that involved 48 subjects, 2500 images and 10 semantic categories: beach, mountain, forest, city, farm, street, bathroom, bedroom kitchen, and living-room. After one fixation of about 70ms, the subjects could identify the scene categories with an accuracy of over 90%. When exposure duration was 35ms, the recognition accuracy reduced to about 76% [21].

Not all visual cues are employed at the same time in gist recognition. Pavlopoulou and Yu [1] showed that at short exposure times, edge cues are predominantly used, but at a longer exposure time (more than 32 ms), texture and shade play a more significant role. Castelhano and Henderson found that color information is involved in gist recognition only after 80 ms of viewing [22]. Castelhano and Heaven investigated the top-down influence, e.g. specific target search, on scene

categorization [23]. By monitoring the eye movements, they found that scene context and target features improve early attentional guidance and the speed of target recognition.

Understanding gist recognition in humans and replicating this ability in computers have been ongoing research goals [24, 25, 26, 27, 28, 29, 30]. In the early psychovisual studies, gist recognition is shown to happen in the first 30 to 300ms of viewing a scene [2]. This perception occurs much earlier than the time required for recognizing individual objects in the scene. Consequently, gist recognition is considered to rely significantly more on holistic and low-level properties than on the detection of individual objects [25]. These low-level properties include edges [1], color [22, 31], and texture [21]. It was even found that object shape and identity are not necessary for the rapid perception of scenes [32].

In computational scene categorization, visual descriptors play a central role in recognition performance. A good visual descriptor is invariant in the presence of inter- and intra-class variations caused by photometric and geometric image distortions. Note that previously Douze *et al.* [33] compared different visual descriptors for image search, but the compared descriptors were restricted to only GIST descriptor [32] and its variants. Mikolajczyk and Schmid evaluated the local descriptors, such as SIFT-based features, steerable filters, complex filters, and moment invariants [34]. Van de Sande *et al.* analyzed mainly SIFT-based color descriptors on the PASCAL VOC Challenge 2007 [35]. Xiao *et al.* compared fourteen descriptors on the SUN397 data set [4]. Their comparison also focused on the local features, such as *scale-invariant feature transform* (SIFT) [36], *histogram of oriented gradients* (HOG) [37], and *local binary pattern* (LBP) [38]. This chapter

aims to assess the state-of-the-art visual descriptors for scene categorization of static images, from both methodological perspective. The compared descriptors range from biologically-inspired features, local features to global features.

The chapter is structured as follows. Section 2.2 discusses gist recognition in humans. Section 2.3 reviews visual descriptors used for scene categorization.

2.2 Gist recognition in humans

Gist recognition takes places in the early stage of *human visual system* (HVS). It relies on features such as line orientations, edges, colors, depths, and movements [19]. The transduction of light signals into information that is understood by humans is a complex task of the brain, which is far beyond the capabilities of current computers. In visual perception, light signals are captured by the eyes, focused onto the retina, and processed by two types of photo-receptors, rods (sensitive in low illumination) and cones (sensitive to color in bright illumination). The pre-processed information reaching the retinal layer is encoded by the retinal ganglion cells into features such as edges, colors, and changes in contrast. After retinal processing, the features are transmitted by *lateral geniculate nucleus* (LGN) to the visual cortex in the brain, where complex tasks, such as motion detection, object search, and face analysis, are performed.

Neuroscientists have investigated the areas of the human brain that might be involved in visual perception [39]. They have shown that primary visual cortex area V1, *inferior temporal* (IT) cortex, *prefrontal cortex* (PFC), and *lateral intraparietal* (LIP) area are all important in encoding visual features and classifying scenes [2]. Snowden *et al.* showed that V1 and V2 are involved in basic visual

features, V3 forms perception, V3/VP is for shape perception, V4 is for color perception, and V5/MT is used for motion detection, spatial localization, hand and eye movements [19]. Epstein *et al.* suggested that *parahippocampal place area* (PPA) within the medial temporal lobes is involved in place recognition, route planning, and perceptual encoding [40].

Psychophysical demonstrations, such as tile after effect and the simultaneous size illusion [19, 41], show the images that enter our eyes are filtered by the visual system into discrete channels of features, such as orientations, colors, and motions at each point on the scene [19]. In the following subsections, three key aspects of the human visual system for gist recognition are reviewed: perception of space, perception of color/luminosity, and perception of motion.

2.2.1 Perception of space

The human visual system analyzes the orientation, size, and depth of our environment, and forms the perception of space in the brain. The tilt after effect reported in [41] provided strong evidence that humans have the orientation-selective neurons that give vigorous responses to line stimuli at different orientations. The oriented neurons are considered as a series of band-pass filters [19]. The tilt after effect shows that after the neuron for a particular orientation is excited, it will adapt to this orientation. When a new stimulus appears near the neuron, its response will be lowered, and the response distribution will shift to the opposite direction of the neuron.

Oriented neurons are tuned with the receptive fields of different sizes, when the bars with different sizes are used as stimuli. The large receptive fields are

activated for large bars [19]. In fact, the large thick bars have low spatial frequency, whereas the thin bars have high spatial frequency. For gist recognition, the detailed information, e.g. the texture of the leaves and the number plate of a car at a far distance, contains high spatial frequency, and is not essential. However, detecting pedestrians that are crossing the road, and analyzing the stream of people in the train station are important for surveillance, and these visual tasks involve lower spatial frequencies. In addition, the real size of object is different with its size at the retina. Even the real size of an object remains the same, its size at the retina will change according to the distance between the object and our eyes. However, the size constancy of HVS allows us to perceive accurately the real size of objects, regardless of their distance from the eyes.

Humans have several ways to infer the depth information, from the flat images at the two retinas. First, the depth is formed from the disparity between the left and right retinal images; this is known as stereopsis. Second, the motion parallax gives us a powerful depth cue. Objects that are close to us move faster, while the objects that are far away move slowly. Third, humans can still perceive depth from a flat image, even without stereopsis and motion information. This is achieved by using pictorial cues: the size of nearby objects is larger, objects block other objects behind them, and objects with shadows lead to different interpretations of depth. The pictorial cues are learned from the order of the nature. For example, the human brain assumes that light comes from above, and decodes shading and shadows to infer depth relationships [19]. When humans are not in the normal condition, e.g. astronauts on the international space station, their visions can be altered temporarily by the weightlessness and lighting conditions [42, 43].

2.2.2 Perception of color/luminosity

Color perception in humans is developed for tasks that are vital for our survival, such as avoiding dangers and finding food. We use color to follow traffic signals or select our favorite fruits. HVS is a trichromatic system that is sensitive to the wavelengths of visible light, ranging from 350 to 770nm. The trichromatic system provides humans three main signals from blue, green, and red cones. These visual signals from retina and LGN are then conveyed to area $V1$, $V4$, and $V8$ in the visual cortex [19]. Area $V1$ changes as the wavelength of the illumination changes, area $V4$ exhibits color constancy, and area $V8$ is activated more by colored patterns than by luminance patterns.

2.2.3 Perception of motion

Motion perception is important to humans, especially for daily tasks like driving cars or filling a glass with water. There are two ways to perceive movements. One is by detecting movements across the retina (retinal movement system); these movements are caused by the objects moving in the world. The other is by detecting movements of the eyes (eye and head movement system). Humans can see motions, tell what is moving, and calculate the direction and speed of an object, based on the complex movement patterns that mix retina motions and eye-head motions. The area involved in motion perception of human brain is $V5/MT$. This area receives signals from LGN and extracts directions and speed of moving objects [19]. A basic model of the motion sensor in humans is the delay-and-compare detector. It samples two different points in space with a time delay and compares the receptive fields corresponding the two points. If an object

moves from point A to point B, it will excite the receptor with a receptive field at *A* before the one with a receptive field at *B*.

Studies of the human visual system have shown that there are so many different cells and layers in our brain, and it is hard to find a universal model that captures the complexity of the visual cortex. However, Mountcastle proposed that the neocortex is uniform in appearance and structure [44]. He believed that the cortex uses similar computational functions to accomplish all tasks, such as sight, hearing, touch, smell, and taste. This discovery inspired Hawkins and Blakeslee to build the fundamental theory of neocortex and create brain-inspired data processing tool called Grok [45, 46].

Findings from these biological studies have inspired several computational vision models. In the next section, we will review existing computational approaches for gist recognition and scene categorization.

2.3 Visual descriptors

Many descriptors have been developed for scene categorization from static images. For convenience, the list of the most widely used acronyms in this section is given in Table 2.1. In scene categorization, visual features are first extracted from the input image, and then classified into semantic categories using a trained classifier, e.g., support vector machines. Hence, feature extraction plays a vital role in scene categorization. The existing approaches for visual feature extraction in scene categorization can be divided into three broad categories: biologically-inspired methods, local features, and global features. Table 2.2 summarizes the three categories and gives the representative visual descriptors of each category.

Table 2.1: List of acronyms used in this section.

Acronym	*Definition*
BIF	Biologically-inspired features [47]
BOP	Bag-of-parts [48]
BoW	Bag-of-words [49]
CENTRIST	Census transform histogram [50]
CNN	Convolutional neural network [51, 52]
DeCAF	Deep convolutional activation feature [53]
GIST	An abstract representation of the scene [32]
HIK	Histogram intersection kernel [54]
HMAX	Hierarchical model and X [55]
HOG	Histogram of oriented gradients [37]
HOG-SPM	HOG with spatial pyramid matching [3]
HSOG	Histogram of the Second-Order Gradients [56]
LBP	Local binary pattern [38]
LBP-HF	LBP with Fourier histogram [57]
LCS	Local color statistic [58]
OB	Object bank [59]
PLBP	LBP with pyramid representation [60]
RBF	Radial basis function kernel [61]
SEV	Sequential edge vectors [62]
SIFT	Scale-invariant feature transform [36]
SIFT-FV	SIFT with Fisher Vector [63]
SIFT-LLC	SIFT with locality-constrained linear coding [64]
SIFT-ScSPM	SIFT with sparse coding based spatial pyramid matching [65]
SIFT-SPM	SIFT with spatial pyramid matching [3]
SURF	Speeded up robust features [66]
SVM	Support vector machine [67]

2.3.1 Biologically-inspired feature extraction models

To mimic the gist recognition capability of the human vision, researchers have developed a variety of computational algorithms for scene categorization from static images. For example, Lee and Mumford developed a hierarchical Bayesian inference for scene reconstruction based on the *early visual neurons* [76]. Song and Tao suggested a gist recognition model, where intensity, color, and C1 visual features are extracted [47]. Grossberg and Huang proposed the ARTSCENE system based on gist and texture features for natural scene classification [77]. In the following subsections, we discuss three major biologically-inspired feature extraction approaches: the HMAX model, the GIST model, and the deep learning model.

Table 2.2: Classification of scene categorization descriptors

Approaches	*Representative works*
Biologically-inspired feature extraction	
- Visual cell model	HMAX [68]
- Layout properties	GIST [32]
- Deep learning features	Convolutional Neural Networks [69], OverFeat [70], DeCAF [53]
Local feature extraction	
- Patch-based features	SIFT [36], HOG [37], LBP [38], CENTRIST [50] HSOG [56]
- Object-based model	Object bank [59]
- Region-based model	Edge vectors [62] Bag of parts [48]
- Keypoint-based features	SURF [66], FREAK [71], BRISK [72]
Global feature formation	
- Principal component analysis	SIFT features with PCA [73]
- Histogram	Multi-resolution histogram [74]
- Feature encoding	SPM [3], ScSPM [65], LLC [64] FV [63, 75]

2.3.1.1 HMAX model

Riesenhuber and Poggio proposed a feed-forward architecture, called HMAX, which is inspired by the hierarchical nature of the primate visual cortex [55]. The HMAX model has four layers: two simple layers ($S1$ and $S2$) and two complex layers ($C1$ and $C2$). In layer $S1$, the input image is densely filtered with Gaussian filters at several orientations and scales. The feature maps generated from layer $S1$ are then arranged into filter bands that contain neighboring scale maps with different orientations. After layer $S1$, the spatial maximum values in each filter band are computed at the same orientation to form layer $C1$. Let P_i, $i = 1, \ldots, K$, be a dictionary that is learned from samples of layer $C1$. P_i is used as a prototype to represent intermediate-level feature $S2$. The maximum values of $S2$ are computed over all positions and scales, and are used as the output features $C2$, which are

shift- and scale-invariant.

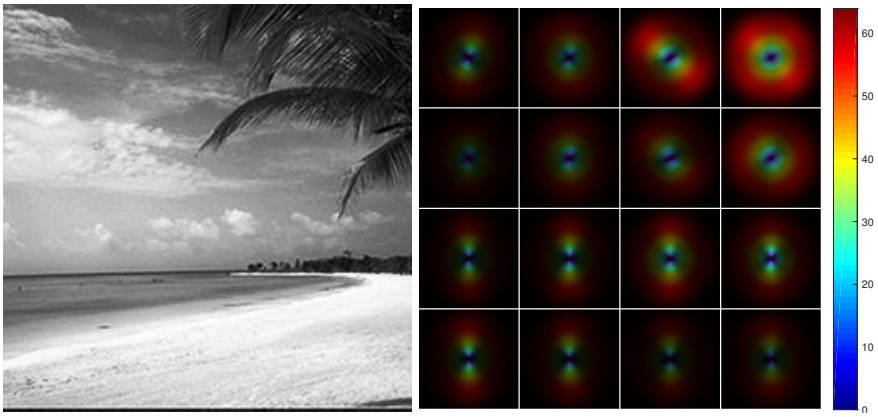
The HMAX architecture has the advantages of both template-based features [78] and histogram-based features [36, 37]. The HMAX features preserve object geometry similarly to template-based features, and they are robust to small distortions in objects, like histogram-based features. Serre *et al.* later used the HMAX features for object recognition and scene understanding [79]. In object recognition tasks, the HMAX model outperforms HOG [37], the part-based model [80], and the local patch correlation [81]. However, HMAX has a longer processing time than other algorithms, such as HOG and SIFT.

Several extensions to HMAX have been proposed. Serre and Riesenhuber introduced a new HMAX that uses Gabor filters to model simple cell receptive field instead of Gaussian filters [82]. Mutch and Lowe extended HMAX by introducing lateral inhibition and feature localization [83]. The extended HMAX achieved an improvement of 14% in the classification rate compared to the original HMAX, in object categorization on the Caltech-101 data set. Brumby *et al.* developed a large-scale functional model based on HMAX and applied it to detect vehicles in remote-sensing images [84]. Inspired by HMAX, to describe the scene, Jiang *et al.* [85] proposed a new approach that combines features from simple cells and complex cells with *sparse coding based spatial pyramid matching* (ScSPM) [65].

2.3.1.2 GIST model

Oliva and Torralba proposed a computational model, known as GIST, for scene categorization [32]. They suggested that images in a scene category possess a similar spatial structure that can be extracted without segmenting the image. The

GIST features are the statistical summary of the scene spatial layout. That is, they capture the dominant perceptual properties, such as naturalness, openness, roughness, expansion, and ruggedness, of a scene.



(a) input image (b) GIST features in polar form

Figure 2.1: Visual illustration of GIST feature extraction.

The GIST model can be described as follows. An input image I is first padded, whitened, and normalized to reduce the blocking artifact. Next, the image is processed with a set of multi-scale oriented Gabor filters. The impulse response of a Gabor filter is a Gaussian modulated by a harmonic function:

$$g(x,y) = \cos(2\pi\frac{x'}{\lambda} + \Phi)\ \exp(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}), \tag{2.1}$$

where $x' = x\cos\theta + y\sin\theta$, $y' = -x\sin\theta + y\cos\theta$, θ is the rotation angle, Φ is the phase offset, λ is the wavelength of the harmonic function, σ is the standard deviation of the Gaussian function, and γ is the spatial aspect ratio. In the original GIST model, 32 filters at four scales and eight orientations are used. Each output filtered image is partitioned into 16 blocks, and the average value of each block is used as a feature. Overall, a GIST feature vector has 512 elements. Figure 2.1 (b) shows the GIST features extracted from an input image shown in Fig. 2.1 (a). Here, the input image is divided into 4×4 regions. From each region 32

gist features (4 scales and 8 orientations) are extracted and visualized in polar coordinates.

The GIST features were found to be more effective for recognizing outdoor scenes than the indoor scenes [50]. They have been combined with other features for scene categorization. Torralba *et al.* proposed a new GIST model that combines the local features, global features, bottom-up saliency, and top-down mechanisms to predict which image regions are likely to be fixated by human observers [86]. Han and Liu [87] developed a hierarchical GIST model for scene classification with two layers: i) a perceptual layer based on the GIST model proposed in [32]; and ii) a conceptual layer based on the kernel PCA [88].

2.3.1.3 Deep learning

In recent years, deep learning architectures have gained fervent research interest for image recognition. One of the major deep learning architectures is *convolutional neural networks* (CNN) developed by LeCun *et al.* [51]. CNNs are inspired by the discoveries of Hubel and Wiesel [89] about the receptive fields in mammal visual cortex. CNNs are based on three key architectural ideas: i) *local receptive fields* for extracting local features; i) *weight sharing* for reducing network complexity; iii) *sub-sampling* for handling local distortions and reducing feature dimensionality. An advantage of CNNs is that they can be trained to map raw pixels to image categories, thereby alleviating the need for hand-designed features.

CNN is a feed-forward architecture with three main types of layers: (i) 2-D convolution layers; (ii) 2-D sub-sampling layers; and (iii) 1-D output layers (see Fig. 2.2 for an example). A *convolution layer* consists of several adjustable 2-D

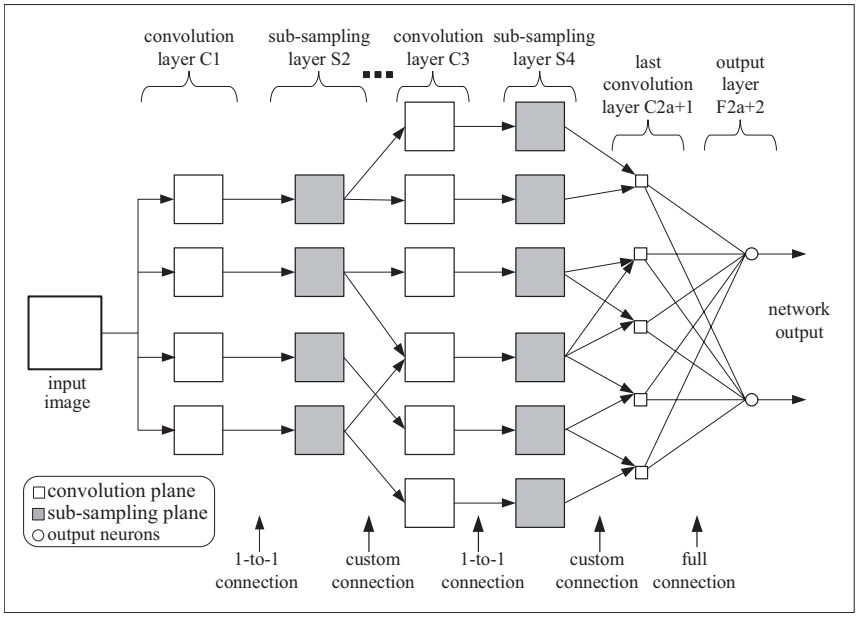


Figure 2.2: An example of layers in a convolutional neural network.

filters. The output of each filter is called a feature map, because it indicates the presence of a feature at a given pixel location. A *sub-sampling layer* follows each convolution layer, and reduces the size of each input feature map, via mean pooling or max pooling. The 1-D layers map the extracted 2-D features to the final network output.

Designing and training a CNN or deep learning architecture is a computation-intensive task that requires significant engineering efforts. Hinton *et al.* proposed a new approach for training deep networks [90]. Their main idea is to pre-train each layer of the network using an unsupervised learning algorithm, e.g. *restricted Boltzmann machine*, *denoising auto-encoder*, and *kernel principal component analysis*. Once the pre-training is completed, a supervised learning algorithm, e.g. error backpropagation, is employed to adjust the connection weights of the hidden layers and the output layer [91].

Krizhevsky *et al.* developed a CNN for image classification, which has 5 con-

volution layers, 650000 neurons and 60 million parameters, and produces 4096 features [52]. On the ImageNet benchmark, which comprises 1.2 million images with 1000 object categories, CNN achieved a top-1 classification rate of 62.5%, which was higher than the previous results obtained by other methods. Sermanet *et al.* later developed a CNN-based integrated framework, called OverFeat [70], to perform both localization and detection tasks; their system achieved very competitive results (1st in localization and 4th in classification) on the ImageNet benchmark. CNNs have also been applied for scene categorization by Zhou *et al.* [69] and Donahue *et al.* [53]. CNNs have been shown to be less effective for moderate-size data sets [92], however, they perform well when trained on large-scale data sets [52, 53].

2.3.2 Local feature extraction

Local descriptors capture low-level properties of the scene, whereas global descriptors represent the overall spatial information. Vogel *et al.* studied the use of local features and global features in the categorization of natural scenes [93]. Their results suggested that humans rely as much on local region information as on global configural information. Existing algorithms for local descriptors can be divided into three main categories: patch-based, object-based, and region-based.

2.3.2.1 Patch-based local features

Patch-based algorithms extract features from small patches of input images. For SIFT [36] and *speeded up robust features* (SURF) [66], patches are generated from local windows around interest points. For LBP [38], patches are formed from rectangular regions of each pixels. For HOG [37], patches are non-overlapping

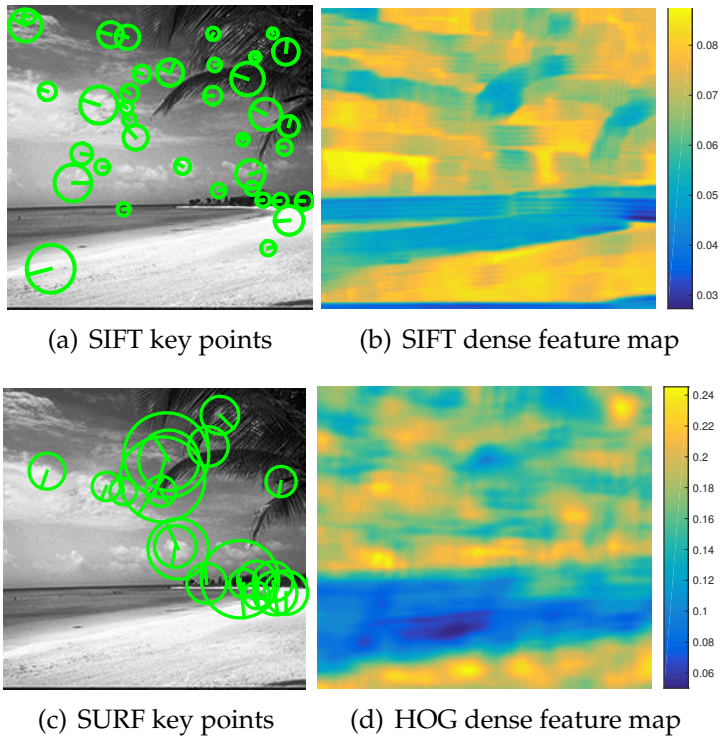
blocks where orientation voting and normalization are applied. For SIFT-ScSPM [65], patches are overlapping blocks formed from a regular grid at the same scale. For SIFT-LCS-FV [63], patches are overlapping blocks formed from a regular grid at five scales.

A) Scale-invariant feature transform

Lowe developed SIFT algorithm to extract image features that are invariant to image scale, rotation, and changing illumination [36]. Extracting SIFT local features involves four main steps. First, the difference-of-Gaussian filters are applied to identify the location and scale of interest points. Second, the interest points with high stability are selected as the key points. Third, dominant orientation is assigned to each key point based on local image gradient. Fourth, the SIFT features that are partially invariant to affine distortions and illumination changes are extracted from key-point regions. The SIFT features are computed from image gradient magnitude and orientation in a region centered at key point. An example of SIFT key points is shown in Fig. 2.3 (a). The centers of circles are the key points, the radiuses of circles are the average scales of the key points, and the arrows inside the circles are the average orientations of the key points.

To apply SIFT for scene categorization, Fei-Fei and Perona proposed to extract local features from dense patches [94]. A dictionary is formed from random local patches using k-means algorithm. Then, for each input image, a feature vector is generated using the trained dictionary. An example of a SIFT feature map that is extracted from dense patches is shown in Fig. 2.3 (b). In this figure, SIFT features are averaged at each pixel location and shown.

The original SIFT algorithm has been extended by several researchers. Brown



(a) SIFT key points (b) SIFT dense feature map

(c) SURF key points (d) HOG dense feature map

Figure 2.3: Visual illustration of SIFT, SURF, and HOG feature extraction of the input image in Fig. 2.1(a).

and Susstrunk proposed *multi-spectral SIFT* (MSIFT) on color and near-infrared images for scene categorization [95]. They showed that compared with SIFT, HMAX, and GIST on the 8-outdoor-scene data set [32], MSIFT reduced feature dimensionality and improved recognition accuracy. Liu *et al.* proposed *SIFT flow* to align images across scenes, and applied it for image alignment, video retrieval, face recognition, and motion prediction [96]. Bo *et al.* improved the low-level SIFT features using a kernel approach [97]. The kernel descriptors provide a principled tool to convert pixel attributes to patch-level features.

B) Speeded-up robust features Bay *et al.* proposed scale- and rotation-invariant visual descriptor, called SURF [66]. SURF detects interest points using determinant of Hessian matrix. The computation time of interest point detection is reduced by using integral images [98]. The key points are then selected from interest points

using non-maximum suppression in multi-scale space. The orientation of key point is assigned using sliding orientation windows on Haar wavelet response maps. The longest vector over all windows defines the orientation of the key point. The Haar wavelet response in the horizontal direction (d_x) and the vertical direction (d_y) are computed from a 4×4 sub-region over the key point. The feature vector for each sub-region is $v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$. SURF key points of an example image is shown in Fig. 2.3(c). The centers of circles are the key points, the radiuses of circles are the average scales of the key points, and the arrows inside the circles are the average orientations of the key points.

C) Histogram of oriented gradients Dalal and Triggs originally developed the HOG descriptor for pedestrian detection in gray-scale images [37]. The HOG features have since been applied for recognition of other image categories, such as cars [99], bicycles [100], and facial expressions [101]. The HOG feature extraction involves four main steps. First, an input image is normalized by the power-law, and the image gradients are computed along the horizontal and vertical directions. Next, the image is divided into cells; a cell can be a rectangular or circular region. In the third step, the histograms for multiple orientations are computed for each cell, where each pixel in the cell contributes a weighted score to a histogram. Finally, the cell histograms are normalized and grouped in blocks to form the HOG features. An example of HOG features is shown in Fig. 2.3(d), which illustrates the strength of averaged HOG features at each pixel location.

For scene categorization, the HOG features are useful for capturing the distribution of image gradients and edge directions in a regular grid. Xiao *et al.* compared HOG with other descriptors, such as GIST and SIFT, on the SUN397

data set [4]. HOG achieved a higher classification rate (CR) compared with other hand-designed descriptors.

D) Local binary pattern

Ojala *et al.* first developed the LBP algorithm for texture classification [38]. Since then, LBP has been applied to many computer vision tasks, including face recognition [102], pedestrian detection [103], and scene categorization [4]. The LBP algorithm analyzes the textures of a local patch by comparing each center pixel with the neighboring pixels.

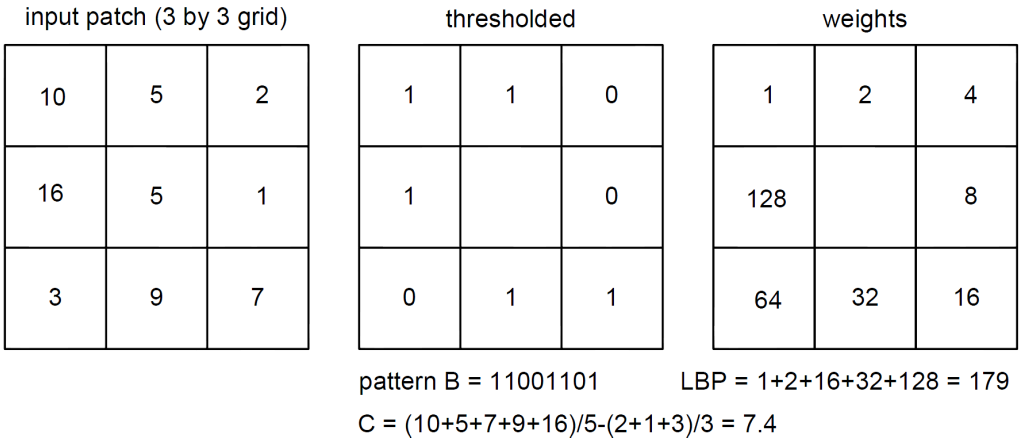


Figure 2.4: Illustration of the basic LBP algorithm.

The basic LBP operates on 3-by-3 blocks. Each pixel in the block is compared to the center pixel, and a binary value of 1 or 0 is returned (see Fig. 2.4). A pattern B is formed by concatenating the binary values from the neighboring pixels. The decimal LBP code is obtained by summing the thresholded differences weighted by powers of two. Furthermore, a contrast measure C is obtained by subtracting the average of pixel values smaller than the center pixel value p from the average of pixel values larger than or equal to p. An example of LBP code map is shown in Fig. 2.6. In the example, the pattern B is 11001101; the LBP code is 179; the contrast C is 7.4. The histogram of local contrast (LBP/C) is used as a feature

vector. The computational simplicity of the LBP algorithm makes it suitable for real-time image analysis.

Several variants of LBP have been developed. Ojala *et al.* proposed a generic form of LBP that supports arbitrary neighborhood sizes [104]. By contrast to the basic LBP that uses 8 neighboring pixels in a 3-by-3 block, the generic operator $LBP_{P,R}$ is circularly symmetric, see Fig. 2.5.

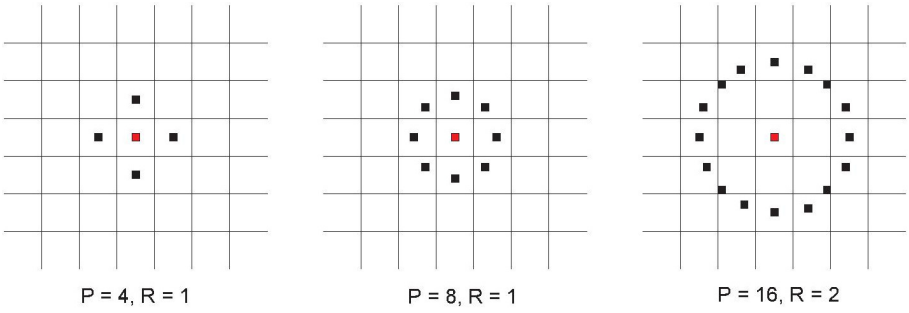
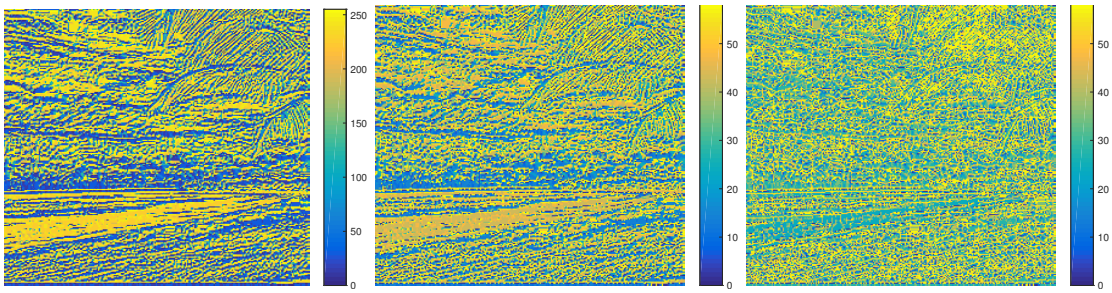


Figure 2.5: The circular regions in a generic form of LBP. Here, P is the number of neighboring pixels, and R is the circle radius. When $P = 8$ and $R = 1$, the basic LBP operator is obtained.

Ojala *et al.* also developed the uniform patterns, denoted as $LBP^{u}_{P,R}$ [104]. Here, u is the number of transitions (0 to 1, or 1 to 0) in an LBP pattern. A local binary pattern is called uniform if $u \leq 2$. Different output labels are assigned to uniform LBP codes, and a single label is assigned to non-uniform patterns. Experiments in [104] indicated that uniform patterns can be considered as fundamental textures because they represent the vast majority of local texture patterns.

Ahonen *et al.* proposed an algorithm, called LBP-HF, that combines uniform LBP and Fourier coefficients [57]. They showed that LBP-HF has better rotation invariance than uniform LBP. In the LBP-HF algorithm, uniform pattern $LBP^{u}_{P,R}$ at pixel location(x, y) is replaced by $LBP^{u}_{P,(R+\theta) \bmod P}$, where θ is a rotation angle: $\theta = 0, \frac{2\pi}{P}, \frac{2\times 2\pi}{P}, \ldots, \frac{(P-1)\times 2\pi}{P}$. Then, the histograms h_θ of $LBP^{u}_{P,(R+\theta) \bmod P}$ are computed.

Finally, Discrete Fourier Transform is applied on h_θ to form LBP-HF features.



(a) LBP (b) Uniform LBP (c) LBP-HF

Figure 2.6: Visual illustration of LBP-based feature extraction of the input image in Fig. 2.1(a).

Guo *et al.* proposed a *completed local binary pattern* (CLBP) to extract image features for texture classification [105]. The original LBP only encodes the signs of differences between center pixel and its neighbors (see Fig. 2.4). CLBP encodes both the signs (CLBP-S) and magnitudes (CLBP-M) of the differences. Furthermore, the intensity of center pixel (CLBP-C) is encoded as the third part of CLBP. Guo *et al.* showed that the texture classification accuracy of CLBP was better than the original LBP algorithm. Li *et al.* proposed a scale- and rotation-invariant LBP descriptor [106]. The scale-invariance of this method is achieved by searching for the maximum response over scale spaces. The rotation invariance is achieved by locating the dominant orientations of the uniform-LBP patterns. The scale- and rotation-invariant LBP outperforms the classical uniform LBP in texture classification. In another approach, Qian *et al.* proposed an LBP descriptor with pyramid representation (PLBP) [60]. By cascading the LBP features obtained from hierarchical spatial pyramids, the PLBP descriptor extracts texture resolution information. The pyramid representation for LBP is more efficient than the multi-resolution representation for LBP proposed by [107].

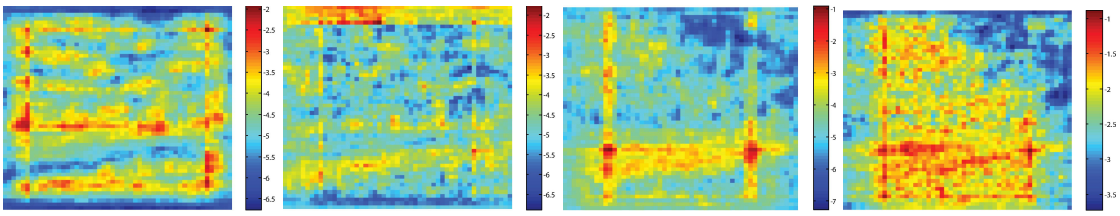
E) Census transform histogram Wu and Rehg proposed a visual descriptor called CENTRIST, which is a holistic representation of structural and geometrical properties of images [50]. In CENTRIST, feature maps are calculated by census transform (CT), which is equivalent to the local binary pattern $LBP_{8,1}$. Wu and Regh presented an experiment showing that CT values encode shape information. The authors identified 6 CT values with highest counts (31, 248 240, 232, 15, and 23) in the 15-scene data set [3, 94]. The 6 CT values correspond to local 3-by-3 neighborhoods that have horizontal or close-to-diagonal edge structures. To encode the global structure of an image, the CT values are processed by the spatial pyramid algorithm, described in [3]. The features extracted from the pyramid feature maps are then reduced using the spatial PCA or BoW methods.

Wu and Rehg showed experimentally that CENTRIST with the spatial PCA outperforms the state-of-the-art algorithms, such as SIFT and GIST, on several scene categorization data sets. However, CENTRIST has a number of limitations. First, CENTRIST is not invariant to rotations. Second, it is not designed for extracting color information. Third, CENTRIST still has difficulty in learning semantic concepts from images with varied viewing angles and scales. Based on CENTRIST, a multi-channel feature generation mechanism was proposed in [108]. The CENTRIST features with multi-channel information (RGB channels and infrared channel) improves grayscale CENTRIST's performance on scene categorization.

2.3.2.2 Object-based local features

Object-based algorithms rely on landmark objects to classify scenes. They have been applied for scene perception in robotic navigation systems [109, 110, 111]. Bao *et al.* proposed a scene layout reconstruction algorithm by determining the 3-D locations and the support planes of objects [112]. In these methods, the scene is classified based on landmark objects and their configuration. A challenge of object-based methods is to detect small objects, especially in outdoor conditions. Another challenge is to select a small set of landmark objects to represent a scene [113].

An example of object-based algorithms for visual categorization is *object bank* proposed by Li *et al.* [59]. The OB algorithm was designed to decrease the gap between low-level visual features of objects and high-level semantic information of the scene.



(a) 'tree' (b) 'sky' (c) 'beach' (d) 'coast'

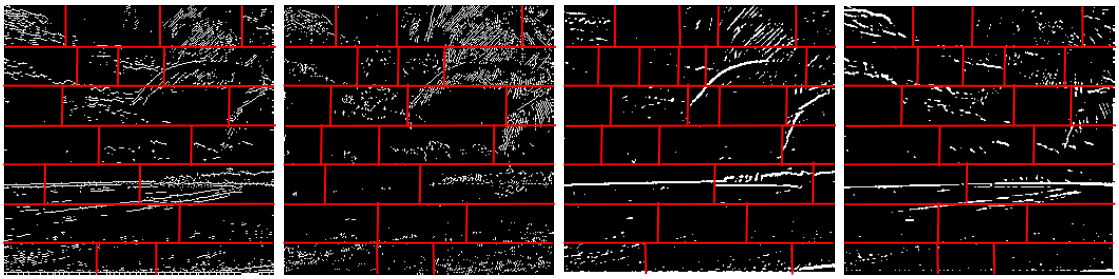
Figure 2.7: Object bank feature maps of input image in Fig. 2.1(a).

In the OB approach, an image is represented by scale-invariant response maps produced by pre-trained object detectors. Objects are classified by two types of detectors: i) the SVM detector, proposed by Felzenszwalb *et al.* [100], for objects such as humans, cars, and tables; ii) the texture detector, proposed by Hoiem *et al.* [114], for objects such as sky, road, and sea. Li *et al.* analyzed common object

types in four data sets ESP [115], LabelMe [116], ImageNet [117], and Flickr [118], and selected 200 object detectors that were trained with 12 detection scales and 3 spatial pyramid levels [119]. The example feature maps produced by four object detectors are shown in Fig. 2.7 (a) to (d). Li *et al.* compared the OB algorithm with SIFT, GIST, and *spatial pyramid matching* (SPM) [3] on several scene data sets. The results showed that OB outperforms the other algorithms on the UIUC data set [120], the 67-indoor-scene data set [121], and the 15-scene data set [3].

2.3.2.3 Region-based local features

Region-based algorithms segment images and extract features from different regions. Boutell *et al.* proposed an algorithm to classify scenes using the identities of regions and the spatial relations between regions [122]. Gokalp and Aksoy proposed a bag-of-regions algorithm for scene classification [123]. In their algorithm, an image is partitioned into regions, and the structure of the image is represented by a bag of individual regions and a bag of region pairs. Juneja *et al.* used distinctive parts for scene categorization [48]. First, the distinctive parts in each category are detected and learned based on HOG features. Then, the features of parts are extracted and encoded using bag of words or Fisher Vector.



(a) (b) (c) (d)

Figure 2.8: Illustration of SEV regions of input image in Fig. 2.1(a) and the computed edge maps along the (a) horizontal direction, (b) vertical direction, (c) +45 degree direction, and (d) −45 degree direction.

An example of region-based algorithms is *sequential edge vectors* (SEV) proposed by Morikawa and Shibata [62]. SEV reduces the ambiguity of features caused by the inter-class variations among scene categories. Unlike other region-based algorithms that segment the entire image, SEV segments only local regions. SEV method considers an image as a *document* consisting of *sentences*, in which the separated regions play the role of *words*, and oriented edges play the role of *letters*. SEV identifies each word from the letters, forms the document vector from words, and finally determines the topic vector.

The main steps of the SEV method can be described as follows. First, oriented edges of input image are detected by horizontal, vertical, and diagonal edge filters. Second, the horizontal or vertical edge map is used to generate fixed sentences (also called local threads). All local threads are scanned from one end to the other, producing edge distribution vectors in four directions. Next, the sum of absolute differences (SAD) between neighboring local windows is calculated. The boundaries of meaningful sequences are determined by locating peaks in the SAD histograms. Figure 2.8 shows the boundaries in four edge maps. Then, the meaningful sequences form words that describe images. Finally, probabilistic latent semantic analysis (PLSA) [124] is applied on meaningful sequences (words) to generate topic vectors.

Using the 8-outdoor-scene data set, Morikawa and Shibata evaluated the SEV algorithm on seven scene categories: coast, open country, forest, highway, mountain, street, and tall building. SEV method with a 16-by-16 scan window achieved a higher F-measure (65%) than a model that uses SIFT and PLSA (55%).

2.3.3 Global feature formation

For scene categorization, global features are often extracted without image segmentation or object detection to summarize the statistics of the image. For example, Renninger and Malik computed histograms of features generated by a bank of Gaussian derivative filters to represent scenes [21]. Serrano *et al.* used quantized color histograms and wavelet texture as global features for scene categorization [125]. Furthermore, Mikolajczyk and Schmid showed that the performance of visual descriptors depends not only on local regions but also global information [34]. Therefore, several scene categorization algorithms focus on first extracting suitable local features and then forming global features from regular grids. In the following subsections, four representative algorithms for global feature formation in scene categorization are described: PCA, histograms, BoW, and Fisher Vector.

2.3.3.1 Principal component analysis

Principal component analysis (PCA) represents data by a small number of principal components. It has been used for dimensionality reduction in a wide range of computer vision tasks, such as image categorization [126], face recognition [127, 128], and feature selection [129, 130, 131].

Ke and Sukthankar proposed a visual descriptor, known as PCA-SIFT, that combines SIFT and PCA [73]. In their approach, a projection matrix P is calculated from a large number of image patches. For each training image patch, a feature vector is generated from the horizontal and vertical gradient maps. The covariance matrix C of all feature vectors is calculated. Eigen-analysis is applied to C, and the n most-significant eigenvectors of C are selected to form the projec-

tion matrix P. In Ke and Sukthankar's experiments, n was selected to be 20, which is significantly smaller than the number of features (128) in the standard SIFT algorithm. Compared to SIFT features, PCA-SIFT features lead to an improvement in image matching, when evaluated on the INRIA database [132].

2.3.3.2 Histogram

Histogram is a method to represent the statistical distribution of data. It is efficient and robust to noise compared to other feature formation methods [74]. Therefore, it has been used in many image processing tasks, including image and video retrieval [133, 134, 135, 136], image structure analysis [137], image filtering [138, 139], and color indexing [140, 141, 142, 143]. In several scene categorization algorithms, such as LBP and its variants, global features are formed by calculating the histogram of local features [38, 50, 144].

A weakness of the histogram approach is that it does not capture the spatial information. To encode spatial information, Hadjidemetriou *et al.* proposed multi-resolution histograms that extract shape and texture features at several resolution levels [74]. Given an image $I(x, y)$ with n gray-levels, the spatial resolution of the image is decreased by convolving it with a Gaussian kernel $G(x, y; \sigma)$. The multi-resolution histograms are calculated as $\mathbf{h}[I * G(\sigma)]$. Then, the cumulative histograms corresponding to each image resolution is computed. Next, the differences between the cumulative histograms of consecutive levels are calculated. The difference histograms are sub-sampled and normalized to make them independent of the sub-sampling factor. Finally, the normalized difference histograms are concatenated to form a feature vector.

2.3.3.3 Bag-of-words

The *bag-of-word* (BoW) algorithms were first used in document classification to simplify the representation of natural language. Recently, BoW has been used to classify images based on the appearance of image patches. In the BoW algorithms, features are extracted from regular grids (*feature extraction*) and quantized into discrete visual words (*encoding*). The coding strategy aims to generate similar codes for similar features. A compact representation of visual words is built for each image based on a trained dictionary (*codebook*).

Qin and Yung proposed an algorithm based on contextual visual words [145]. To train the visual words, SIFT features are calculated from both *region of interest* (ROI) and the regions surrounding ROI. Fei-Fei and Perona [94] proposed a BoW algorithm based on *latent Dirichlet allocation* [146]. They compared different local patch detectors, such as regular grids, random sampling, saliency detector [147], and difference-of-Gaussian detector [36]. Their experiment results showed that regular grids perform better than random sampling, saliency detector, and difference-of-Gaussian detector for scene categorization.

Next, we describe three representative BoW algorithms for training visual words on regular grids: SPM [3], ScSPM [65], and LLC [64].

Spatial pyramid matching (SPM) for recognizing natural scenes was proposed by Lazebnik *et al.* [3]. Unlike the traditional BoW algorithms that extract orderless features, the SPM algorithm retains the global geometric correspondence of images. It divides the input image into regular grids and computes local features, such as SIFT and HOG in each grid. The visual vocabulary is formed by k-means clustering, and then all features are formed using vector quantization

(VQ). Based on the trained dictionary, local features are represented. Finally, the spatial histograms (average pooling) of coded features are used as feature vectors. To recognize multiple scene categories, a support vector machine with the one-versus-all strategy is used.

Sparse coding based spatial pyramid matching (ScSPM) was suggested by Yang *et al.* [65] to improve the efficiency of SPM. The original SPM uses *k*-means vector quantization, whereas ScSPM uses sparse coding to quantize the local features. Furthermore, for spatial pooling, the original SPM uses histograms, whereas ScSPM applies the *max* operator, which is more robust to local spatial translations. For ScSPM, the linear-SVM classifier is used to reduce the computation cost. The experiments by Yang *et al.* show that the sparse coding of SIFT descriptors with the linear-SVM outperforms several methods, including kernel codebooks [148], SVM-K-Nearest-Neighbor [149], and naive Bayes nearest-neighbor (NBNN) [150].

Locality-constrained linear coding (LLC) was proposed by Wang *et al.* [64] to reduce the computational cost of SPM. Because of the importance of locality, as demonstrated by Yu *et al.* [151], LLC replaces the sparsity constraint used in ScSPM with the locality constraint to select similar bases for local features from the trained visual words. In Wang et al.'s approach, a linear weighted combination of these bases is learned to represent local features. Their experiments on the Catech-101 data set [49] show that LLC achieved higher classification rates than ScSPM, NBNN, and kernel codebooks [148]. Recently, Goh *et al.* improved ScSPM and LLC by using a deep architecture [92]. The deep architecture merges the strengths of BoW framework and the deep learning method to encode the SIFT

features.

2.3.3.4 Fisher Vector

Fisher Vector (FV) is a feature encoding algorithm proposed by Sanchez *et al.* [63]. The local features extracted from dense multi-scale grids are represented by their deviation from a *Gaussian mixture model* (GMM). The local features are mapped to a higher-dimensional space which is more amenable to linear classification.

In the FV algorithm, a GMM is first computed from a training set of local features using the Expectation-Maximization algorithm. The parameters of a GMM are denoted by $\lambda = \{(w_k, \boldsymbol{\mu}_k, \boldsymbol{\sigma}_k), k = 1, ..., K\}$, where w_k is the mixture weight, $\boldsymbol{\mu}_k$ is the mean vector, and $\boldsymbol{\sigma}_k$ is the covariance matrix of the k-th Gaussian component.

Let $X = \{\mathbf{x}_t, t = 1, ..., T\}$ be the set of local descriptors extracted from an input image. For local descriptor $\mathbf{x}_t$, let $p_\lambda(\mathbf{x}_t)$ be the probability density function of local descriptor, as computed by GMM model. Let L_λ be the square-root of the inverse of the Fisher information matrix. The normalized gradient statistics are computed as

$$\varphi(\mathbf{x}_t) = L_\lambda \nabla_\lambda \log p_\lambda(\mathbf{x}_t). \tag{2.2}$$

A Fisher Vector is the sum of normalized gradient statistics:

$$\mathcal{G}_\lambda^X = \sum_{t=1}^{T} \varphi(\mathbf{x}_t) \tag{2.3}$$

The final Fisher Vector is the concatenation of the gradients $\mathcal{G}_{w_k}^X$, $\mathcal{G}_{\boldsymbol{\mu}_k}^X$, and $\mathcal{G}_{\boldsymbol{\sigma}_k}^X$. To improve the classification accuracy, two normalization steps, l_2-normalization and power normalization [152], can be applied on Fisher Vector.

Compared with other bag-of-words algorithms, Fisher Vector has many advantages. First, it provides a generalized method to define a kernel from a generative

process of the data. Second, Fisher Vector can be computed from small vocabularies with a lower computational cost. However, Fisher Vector is dense, which leads to storage issues for large-scale applications.

2.3.3.5 Composite global features

Global features are also formed by combining different global features. For example, CENTRIST features are generated by combining histograms and PCA [50]. The HOG-SPM and SIFT-SPM features are first extracted by BoW algorithms, and then accumulated by spatial histograms [3]. SIFT-ScSPM [65] and SIFT-LLC [64] represent local features by the BoW algorithm and the max pooling. The image categorization method proposed by Krapac *et al.* [5] combines SPM and the Fisher kernel to encode spatial layout of images.

2.4 Chapter summary

This chapter presented a survey of recent work on visual gist recognition and scene categorization, from theoretical perspective. After describing gist recognition in humans, we reviewed the computational approaches for scene categorization in three categories: biologically-inspired features, local features, and global feature formation.

Chapter 3

Experimental evaluation of visual descriptors for scene categorization

Chapter contents

In this chapter *, we present an extensive experimental evaluation of the state-of-the-art descriptors, which include biologically-inspired, local, and global feature extraction methods. The selected descriptors are evaluated on four benchmark data sets with respect to scene categorization accuracy and class separability

*Chapter 2 and 3 have been published in our paper "Visual descriptors for scene categorization: experimental evaluation," *Artificial Intelligence Review*, vol. 45, no.3, pp. 333-368, 2016.

of feature vectors. For scene categorization, five measures are used to evaluate the classification accuracy, namely *classification rate* (CR), *precision* (P), *recall* (R), *F-measure* (F), and *area under ROC curve* (AUC). By using the same classification protocol described in [52], we are able to compare descriptors on the SUN397 data set with several recent methods: ImageNet-CNN [52], BOP-FV [48], discriminative patches [153], OverFeat [70], Places-CNN [69], and DeCAF [53]. For class separability, the visual descriptors are compared using Fisher discriminant analysis.

The rest of the chapter is structured as follows. Section 3.1 and 3.2 describe the image data sets and performance measures used for scene categorization. Section 3.3 describes the implementation of the descriptors and classifiers. Section 3.4 presents results of a comparative study using different classifiers with four data sets. Finally, Section 3.5 evaluates the class separability and stability of feature vectors.

3.1 Data sets for scene categorization

Progress in image recognition is due in part to the existence of comprehensive data sets on which new or existing algorithms can be rigorously evaluated [154, 155]. In this section, we review the publicly available data sets for scene categorization algorithms, and discuss their characteristics.

Because of the difficulty of finding one representative data set, it is not sufficient to evaluate a scene categorization algorithm on only one data set. For example, on the Caltech-101 data set, the LLC algorithm is found to have a higher classification rate than the ScSPM algorithm [64]. However, on the 15-scene data set, the ScSPM

algorithm has a higher classification rate than the LLC algorithm (see Section 3.4). One possible reason is that built-in biases are present when collecting image data for a recognition task, e.g. the viewing angle, the type of background scene, and the composition of objects. These intrinsic biases cause every data set to represent the physical world differently. Therefore, scene categorization algorithms should be evaluated on multiple large-scale data sets that exhibit more diversity and less bias.

Table 3.1 summarizes the major data sets used for scene categorization. For each data set, the source, the number of images, and the number of image categories are given. In the following, we describe the major data sets. Three benchmark data sets, 8-outdoor-scene [32], 13-natural-scene [94], and 15-scene [3], have been used by many researchers [62, 77, 85, 92, 145, 156, 157, 158, 159]. The 8-outdoor-scene data set has eight categories of only-outdoor scenes (coast, forest, highway, inside city, mountain, open country, street, and tall buildings), whereas the 13-natural-scene data set contains the same eight categories of outdoor scenes and five additional categories of indoor scenes (bedroom, kitchen, living-room, office, and store). The 15-scene data set includes all images from the 13-natural-scene, and two additional outdoor scenes of man-made structures (suburb and industry). The images in each category are from different sources, such as COREL data set, Google image search, and personal photographs.

Large data sets have been developed to increase the number of semantic categories and the diversity of images. The 67-indoor-scene data set [121] divides five indoor scenes (working places, home, leisure, store, and public spaces) into 67 sub-categories. Common objects appear in multiple categories; therefore, it

Table 3.1: Data sets for scene categorization.

Data set	*Data set size (images)*	*Image categories*
8-outdoor-scene [32]	2,600	8 outdoor scenes
13-natural-scene [94]	3,759	13 natural scenes
15-scene [3]	4,485	15 indoor/outdoor scenes
67-indoor-scene [121]	15,620	67 indoor scenes
SUN397 [4]	108,754	397 general scenes
Places205 [69]	2,448,873	205 general scenes

is harder to distinguish between the image categories. The SUN397 data set [4] has 397 scene categories, from abbey, bedroom, and castle to highway, theater, and yard. There are at least 100 images in each category. The Places205 data set [69] has 205 scene categories; each category has at least 5000 images. Among the benchmark data sets listed in Table 3.1, SUN397 has the most number of categories and Places205 has the most number of images.

The experiments in this chapter were conducted on four data sets: the 8-outdoor-scene data set, the 15-scene data set, the 67-indoor-scene data set, and the SUN397 data set. The 8-outdoor-scene data set and the 15-scene data set have been used as benchmark for scene categorization by many researchers [50, 59, 65, 85, 94, 121, 156, 160]. The 67-indoor-scene data set contains 67 indoor scene categories. There are at least 100 images per category and all images have a size of at least 200×200 pixels. The 67-indoor-scene data set has been used to evaluate scene categorization in [48, 50, 64, 75, 121, 153, 161, 162]. The SUN397 data set contains 397 scene categories and 108,754 images. It has been used as a benchmark for scene categorization by [163], [63], [164], [161], [69], [53], and [75].

3.2 Performance measures

To evaluate a scene categorization system, a data set is typically partitioned into three separate subsets: training, validation, and test. The training subset is used to determine the system's adjustable parameters, the validation set is used to prevent over-training, and the test set is used to estimate the system's generalization capability. The generalization capability of a system is commonly measured using classification rate (CR), which is the percentage of test images that are correctly classified. For example, CR has been used for scene categorization in [50, 156, 158, 165].

To prevent bias in partitioning the data set and to estimate more reliably the generalization capability, an alternative technique known as n-fold *cross-validation* is usually adopted. The image data set is divided into n subsets of equal size. For each validation fold, one subset is used for testing, and the remaining $(n-1)$ subsets are used for training and validation. This is repeated n times, each time a different subset fold is used for testing. Finally, the n classification rates are averaged to give the overall CR.

Scene categorization is a multi-class recognition problem. Many scene categorization algorithms are also evaluated using the *confusion matrix* [50, 62, 145, 157, 158, 166]. For a problem involving K categories, the confusion matrix has K rows and K columns. Each row represents an actual category, and each column represents a predicted category. The entry at row r, column c is the number of category r samples that are classified as category c. Clearly, the correct classification for individual categories are the diagonal entries, whereas the miss-classification are

the non-diagonal entries.

Evaluation measures for two-class problems are also applied for scene categorization. Consider a scene category r. The positive class consists of all samples belonging to category r, whereas the negative class consists of all samples belonging to the remaining categories. Four measures can be computed:

- *True positives* (tp) is the number of test samples in the positive class that are correctly classified.
- *False positives* (fp) is the number of test samples in the negative class that are incorrectly classified.
- *False negatives* (fn) is the number of test samples in the positive class that are incorrectly classified.
- *True negatives* (tn) is the number of test samples in the negative class that are correctly classified.

The precision rate P and the recall rate R are then defined as

$$P = \frac{tp}{tp + fp}, \text{ and } R = \frac{tp}{tp + fn}. \tag{3.1}$$

A good scene categorization system should have a high precision rate and a high recall rate. These two requirements can be reflected in a single measure called the *F-measure*, which is the harmonic mean of the precision rate and recall rate:

$$F = \frac{2\,P\,R}{P + R}. \tag{3.2}$$

The plot of the true positive rate versus the false positive rate is called *receiver operating characteristic* (*ROC*) curve [167]. It is a useful tool for visualizing the

scene categorization performance, when a system parameter is varied. Another performance measure is *area-under-the-ROC-curve* or *AUC*. *ROC* and *AUC* have been used for scene categorization in [117] and [4].

The measures described above are suitable for evaluating the performance of a complete scene categorization system that includes both a feature extractor and a classifier. To evaluate the performance of the feature extraction independently of the classifier, we can use the class separability of the extracted features. *Fisher's discriminant analysis* (FDA) is a tool for analyzing the separability of features. For example, Tao *et al.* [168] and Chin *et al.* [169] used FDA to analyze multi-class image classification. Consider a scene categorization problem that involves K classes. The within-class covariance matrix C_w is calculated as

$$C_w = \sum_{k=1}^{K} \sum_{\mathbf{x} \in \omega_k} (\mathbf{x} - \bar{\mathbf{x}}_k)(\mathbf{x} - \bar{\mathbf{x}}_k)^T, \tag{3.3}$$

where $\bar{\mathbf{x}}_k$ is the mean vector of class ω_k. The between-class covariance matrix C_b is given by

$$C_b = \sum_{k=1}^{K} N_k (\bar{\mathbf{x}}_k - \bar{\mathbf{x}})(\bar{\mathbf{x}}_k - \bar{\mathbf{x}})^T, \tag{3.4}$$

where $\bar{\mathbf{x}}$ is the mean vector of all classes and N_k is the number of samples in class ω_k. The *S score* of the feature vector is given by

$$S = \frac{|\text{trace}(C_b)|}{|\text{trace}(C_w)|}. \tag{3.5}$$

A high S score means there is a high separability between the K classes using the given feature vector.

3.3 Implementation of visual descriptors and classifiers

This section describes the implementation of the visual descriptors compared in our experiments. There are two biologically-inspired descriptors (GIST and HMAX), four SIFT-based descriptors (SIFT-SPM, SIFT-ScSPM, SIFT-LLC, and SIFT-FV), two other BOW-based descriptors (HOG-SPM and SURF-ScSPM), five LBP-based descriptors (LBP, Uniform LBP, LBP-HF, PLBP, CENTRIST), and one object-based descriptor (OB). Most of the 14 descriptors combine local and global features.

The biologically-inspired descriptors implemented in this chapter are GIST and HMAX. The *GIST descriptor* is a low dimensional representation of an image. In our experiment, the normalized input image was convolved with Gabor filters at 4 scales and 8 orientations. Each filtered output was down-sampled to a 4 by 4 patch and reshaped to a 16 element vector. The GIST descriptor assembled all outputs from the 32 filters to form a feature vector with 512 elements. The *HMAX descriptor* contains two simple layer $S1$ and $S2$, and two complex layer $C1$ and $C2$. In our experiment, the $S1$ layer was formed from the outputs of Gaussian filters with 4 orientations and 12 scales. In layer $C1$, the $S1$ feature maps were grouped into 4 filter bands of a certain scale range. The max pooling operation was applied to each filter bank. Only $S1$ units with the same preferred orientation fed into a given C1 unit. The $S2$ features were formed from $C1$ features with 256 visual words (learned from $C1$). The $C2$ features were generated from $S2$ features using max pooling. The final HMAX descriptor had 4069 features.

The SIFT-based descriptors were compared to evaluate the encoding capabil-

ity of SPM, ScSPM, LLC, and FV. For the four SIFT-based descriptors (SIFT-SPM, SIFT-ScSPM, SIFT-LLC, and SIFT-FV), their local features were calculated from overlapping patches (16×16) on a dense grid every 8 pixels. The local patch was first filtered with Gaussian filters to generate 8 orientation maps. The histograms were then generated from the orientation maps and further weighted by a Gaussian function. The SIFT features were formed by concatenating the orientation histograms. The *SIFT-SPM descriptor* extracted global features from dense SIFT features by the SPM algorithm. In the experiments, k-means clustering and PCA were used to train and extract 200 visual words from random samples of local features. The local features were quantized by the trained visual words. Finally, histograms of quantized features were formed with 1000 bins. The *SIFT-ScSPM descriptor* formed global features from dense-SIFT features using the ScSPM algorithm. A feature dictionary including 1024 visual words was obtained by applying k-means clustering to the local features. Finally, the ScSPM algorithm was employed to convert the local SIFT features to global features. The *SIFT-LLC descriptor* formed global features from dense-SIFT features using the LLC algorithm. The SIFT-LLC descriptor converted the local feature maps to global features based on the n nearest neighbors of the feature dictionary. In the experiment, the number of neighbors was set to 5. The *SIFT-FV descriptor* extracted global features from dense-SIFT features using the Fisher Vector encoding. The dictionary was trained using a Gaussian mixture model with 256 Gaussians.

Other BOW-based descriptors that are similar to the SIFT-based descriptors include HOG-SMP and SURF-ScSPM. In our experiment, the *HOG-SPM descriptor* extracted local features from overlapping patches (16×16) on a dense grid every

8 pixels. The input image was first normalized globally by the power-law. Then, image gradients were computed along the horizontal and vertical directions. For each patch, histograms for multiple orientations were computed to form HOG features. The global features were then generated by spatial pyramid matching (SPM). The *SURF-ScSPM descriptor* used SURF for local feature extraction. The ScSPM algorithm was applied for global feature formation. Different from the SIFT-based descriptors and HOG-SMP that extracted local features from dense patches, the SURF-ScSPM descriptor extracted local features from interest-points (sparse patches). In our experiment, at least 100 interest points were found in each image.

The LBP-based descriptors compared in the experiment include LBP, uniform LBP, LBP-HF, PLBP, and CENTRIST. The *LBP descriptor* is a histogram of the LBP feature map. In our experiment, the feature map was generated from 3-by-3 blocks of the entire image. The number of histogram bins was selected as 256. When the number of histogram bins was reduced, our preliminary experiments indicated that the classification rate dropped by about 10%. The *uniform LBP descriptor* is similar to the LBP descriptor, but it only encodes uniform patterns of LBP. For uniform LBP, we represented each input image with 59 uniform patterns. The histograms of uniform LBP were calculated with 59 bins. The *LBP-HF descriptor* is an extension of uniform LBP. In our experiment, the histograms of uniform LBP were first calculated on the input image and its 90-degree rotated version. Then, Fourier transform was applied on the histograms. The magnitudes of Fourier coefficients were calculated as the LBP-HF features. The final LBP-HF vector had 76 elements: half of the elements were generated from the original image, and the

other half from the rotated image. The *PLBP descriptor* is an LBP descriptor with spatial pyramid representation. To calculate PLBP features, each input image was decomposed into 5 Gaussian pyramid images with dyadic scales. The histograms of LBP in each pyramid image were combined to form the PLBP features. The *CENTRIST descriptor* first converted input image into the CENTRIST feature map (similar to LBP feature map). Then, spatial histogram with 3 spatial levels and PCA with 40 eigenvectors were employed to form the CENTRIST feature map. The dimension of CENTRIST was 1240.

The object-based descriptor OB extracts features from a large number of pre-trained object detectors. It has the longest feature-extraction stage among the compared descriptors. In our experiment, 176 object detectors with 12 detection scales and 3 spatial pyramid levels were used. An OB descriptor with 44604 features was formed by max pooling.

Support Vector Machines (SVM) with linear, RBF and HIK kernels were used to classify the different descriptors. The linear kernel is given by

$$K(\mathbf{f}_i, \mathbf{f}_j) = \mathbf{f}_i \cdot \mathbf{f}_j, \tag{3.6}$$

where $\mathbf{f}_i$ and $\mathbf{f}_j$ are two feature vectors. The RBF kernel is given by

$$K(\mathbf{f}_i, \mathbf{f}_j) = \exp\{-\gamma ||\mathbf{f}_i - \mathbf{f}_j||^2\}, \tag{3.7}$$

where γ is a positive scalar. The HIK kernel is computed as

$$K(\mathbf{f}_i, \mathbf{f}_j) = \sum_{n=0}^{N} \min[\mathbf{h}_i(n), \mathbf{h}_j(n)], \tag{3.8}$$

where $\mathbf{h}_i(n)$ and $\mathbf{h}_j(n)$ are, respectively, the N-bin histograms of $\mathbf{f}_i$ and $\mathbf{f}_j$.

Five-fold cross validation was used to evaluate the performance of the SVM classifiers with different visual descriptors. In each fold, four subsets were used

Table 3.2: Scene categorization performance on the 15-scene data set using linear-SVM.

ID	*Algorithms*	*CR (%)*	*Precision (%)*	*Recall (%)*	*F-measure (%)*	*AUC (%)*
1	SIFT-ScSPM	**84.5 ± 1.5**	**84.8 ± 1.7**	**84.0 ± 1.4**	**84.1 ± 1.4**	**98.9 ± 0.1**
2	SIFT-LLC	83.0 ± 1.3	83.0 ± 1.5	82.2 ± 1.5	82.3 ± 1.4	98.8 ± 0.2
3	SIFT-FV	80.2 ± 1.8	79.6 ± 1.9	79.2 ± 1.9	79.0 ± 2.0	98.4 ± 0.1
4	HOG-SPM	70.6 ± 2.7	71.7 ± 2.7	68.7 ± 2.6	70.0 ± 2.4	96.3 ± 0.3
5	OB	79.9 ± 1.6	80.0 ± 1.6	79.1 ± 2.0	79.0 ± 1.9	97.7 ± 0.3
6	SIFT-SPM	60.9 ± 4.0	62.3 ± 5.6	57.8 ± 4.3	57.0 ± 4.7	94.4 ± 0.5
7	SURF-ScSPM	73.3 ± 2.1	72.9 ± 2.1	72.3 ± 2.3	72.3 ± 2.4	97.3 ± 0.4
8	GIST	71.5 ± 1.2	71.1 ± 1.4	70.6 ± 1.5	71.2 ± 1.0	95.7 ± 0.4
9	CENTRIST	72.7 ± 1.4	72.8 ± 1.1	72.2 ± 1.9	71.9 ± 1.7	95.9 ± 0.3
10	LBP	71.1 ± 2.9	69.8 ± 4.0	70.3 ± 3.4	69.3 ± 3.9	94.9 ± 1.2
11	Uniform LBP	56.2 ± 3.0	55.6 ± 5.8	54.7 ± 3.2	52.6 ± 3.6	92.8 ± 1.0
12	LBP-HF	64.9 ± 3.4	63.8 ± 4.6	63.9 ± 3.8	62.5 ± 4.3	92.8 ± 1.0
13	PLBP	53.8 ± 3.6	52.7 ± 4.7	52.4 ± 3.6	51.5 ± 4.1	92.1 ± 1.5
14	HMAX	61.1 ± 4.1	60.8 ± 4.0	59.8 ± 4.5	59.9 ± 4.3	79.7 ± 6.9

for training and validation, and the remaining subset was used for testing. The parameters of the SVM classifiers were determined using a validation set in each fold. The average values of *CR*, *P*, *R*, *F*, and *AUC* were calculated over the five folds. The standard deviations of *CR*, *P*, *R*, *F*, and *AUC* over the five folds are used as a measure of variation in the classification performance.

3.4 Classification results

The following four subsections present and discuss the results of image classification on the 15-scene, 8-scene, 67-indoor-scene, and SUN397 data sets.

3.4.1 Classification results on the 15-scene data set

In the first experiment, the 14 selected descriptors were evaluated on the 15-scene data set. Tables 3.2 to 3.4 present the classification performance measures of the different visual descriptors using linear-SVM, RBF-SVM, and HIK-SVM. In these tables (and also Tables 3.5, 3.6, and 3.7), the best performance measure is indicated in bold font. The results in these tables show that each descriptor achieves its best classification rate using a different SVM classifier. For example, SIFT-ScSPM,

Table 3.3: Scene categorization performance on the 15-scene data set using RBF-SVM.

ID	*Algorithms*	*CR (%)*	*Precision (%)*	*Recall (%)*	*F-measure (%)*	*AUC (%)*
1	SIFT-ScSPM	**83.8 ± 1.7**	**84.2 ± 1.6**	**83.2 ± 1.6**	**83.5 ± 1.3**	**98.3 ± 0.0**
2	SIFT-LLC	82.2 ± 1.1	82.5 ± 1.2	81.5 ± 1.1	81.5 ± 1.2	98.2 ± 0.1
3	SIFT-FV	79.4 ± 1.7	78.9 ± 2.1	78.1 ± 2.1	77.6 ± 2.3	97.5 ± 0.2
4	HOG-SPM	78.9 ± 1.2	76.8 ± 3.2	76.3 ± 2.9	76.1 ± 3.1	96.5 ± 0.5
5	OB	73.2 ± 2.0	73.5 ± 1.9	72.0 ± 2.4	72.1 ± 2.3	95.5 ± 0.4
6	SIFT-SPM	72.9 ± 1.3	72.0 ± 1.6	71.6 ± 1.4	71.3 ± 1.4	95.2 ± 0.4
7	SURF-ScSPM	72.4 ± 1.6	71.6 ± 1.8	71.1 ± 1.7	70.6 ± 1.8	95.8 ± 0.4
8	GIST	72.8 ± 1.2	72.2 ± 1.0	71.8 ± 1.4	71.5 ± 1.2	95.2 ± 0.2
9	CENTRIST	72.6 ± 1.8	72.7 ± 1.1	72.0 ± 2.1	71.8 ± 1.8	91.2 ± 1.0
10	LBP	70.9 ± 4.2	69.6 ± 5.2	70.1 ± 4.8	69.2 ± 5.3	96.2 ± 0.7
11	Uniform LBP	67.8 ± 3.9	66.8 ± 5.0	66.8 ± 4.4	65.6 ± 4.9	94.0 ± 0.8
12	LBP-HF	67.3 ± 4.4	66.0 ± 5.4	66.5 ± 4.9	66.3 ± 4.4	94.0 ± 0.8
13	PLBP	69.4 ± 3.5	69.2 ± 4.9	68.2 ± 4.1	68.1 ± 4.4	95.4 ± 1.0
14	HMAX	62.4 ± 3.7	61.6 ± 3.4	61.0 ± 4.0	60.8 ± 3.7	79.5 ± 6.0

Table 3.4: Scene categorization performance on the 15-scene data set using HIK-SVM.

ID	*Algorithms*	*CR (%)*	*Precision (%)*	*Recall (%)*	*F-measure (%)*	*AUC (%)*
1	SIFT-ScSPM	**83.6 ± 1.6**	**84.2 ± 1.7**	**83.2 ± 1.4**	**83.2 ± 1.4**	**98.2 ± 0.1**
2	SIFT-LLC	82.6 ± 1.5	82.8 ± 1.4	82.1 ± 1.3	82.1 ± 1.3	98.2 ± 0.0
3	SIFT-FV	80.0 ± 1.7	80.1 ± 2.0	79.7 ± 1.4	79.3 ± 1.7	97.7 ± 0.2
4	HOG-SPM	80.0 ± 2.6	79.5 ± 2.5	79.7 ± 2.5	79.6 ± 2.4	97.8 ± 0.2
5	OB	76.9 ± 2.0	76.9 ± 1.8	76.0 ± 2.3	76.0 ± 2.2	97.4 ± 0.2
6	SIFT-SPM	77.9 ± 1.2	77.5 ± 1.3	76.8 ± 1.0	76.8 ± 1.0	97.0 ± 0.3
7	SURF-ScSPM	72.3 ± 2.4	71.4 ± 2.5	71.2 ± 2.5	70.6 ± 2.6	95.6 ± 0.4
8	GIST	72.1 ± 0.7	71.8 ± 0.7	71.4 ± 0.9	72.1 ± 0.9	95.2 ± 0.2
9	CENTRIST	70.9 ± 2.4	71.3 ± 1.8	70.1 ± 2.6	70.0 ± 2.3	95.9 ± 0.4
10	LBP	71.9 ± 2.5	71.0 ± 3.2	71.1 ± 3.2	70.5 ± 3.4	95.7 ± 0.8
11	Uniform LBP	70.6 ± 3.5	70.4 ± 4.4	69.8 ± 4.0	69.3 ± 4.5	95.7 ± 1.0
12	LBP-HF	66.4 ± 3.3	65.6 ± 4.4	65.5 ± 3.7	64.8 ± 4.2	95.7 ± 1.0
13	PLBP	73.0 ± 3.4	72.3 ± 5.1	72.9 ± 4.2	72.6 ± 4.7	96.6 ± 1.0
14	HMAX	63.9 ± 3.4	63.5 ± 3.6	62.5 ± 3.7	62.6 ± 3.6	82.9 ± 8.3

SIFT-LLC, SIFT-FV, SURF-ScSPM, CENTRIST, and OB achieve higher classification rates with linear-SVM than with RBF- or HIK-SVM. LBP-HF and GIST have their highest classification rates when using RBF-SVM. HOG-SPM, SIFT-SPM, LBP, uniform LBP, PLBP, and HMAX achieve their highest classification rates with HIK-SVM. The highest classification rates on the 15-scene data set for individual descriptors are (in a descending order) SIFT-ScSPM (84.5%), SIFT-LLC (83.0%), SIFT-FV (80.2%), HOG-SPM (80.0%), OB (79.9%), SIFT-SPM (77.9%), SURF-ScSPM (73.3%), PLBP (73.0%), GIST (72.8%), CENTRIST (72.7%), LBP (71.9%), uniform

LBP (70.6%), LBP-HF (67.3%), and HMAX (63.9%).

The biologically-inspired HMAX algorithm has the lowest *CRs* among the compared algorithms. As shown in Tables 3.2 to 3.4, the *CRs* of HMAX using linear-SVM, RBF-SVM, and HIK-SVM are 61.1%, 62.4%, and 63.9%, respectively. The other biologically-inspired descriptor, namely GIST, has *CRs* of 71.5%, 72.8%, and 72.1%, respectively. Furthermore, the GIST algorithm performed better than the LBP-based and HMAX algorithms with all three SVM kernels.

The SIFT-ScSPM outperforms all other 13 descriptors on the 15-scene data set; its *CRs* is 84.5% for the linear-SVM, 83.8% for RBF-SVM, and 83.6% for HIK-SVM. SIFT-ScSPM algorithm also has higher values of precision, recall, F-measure, and *AUC* than other algorithms.

SIFT-LLC and SIFT-FV perform better than HOG-SPM and SIFT-SPM. SIFT-LLC uses the locality-constrained linear coding and SIFT-FV uses Fisher kernel coding, whereas HOG-SPM and SIFT-SPM uses vector quantization for global feature formation. The result indicates that a better encoding algorithm like ScSPM and Fisher Vector improves the classification performance. Note that among the top-seven algorithms, SIFT-ScSPM, SIFT-LLC, SIFT-FV, HOG-SPM, SIFT-SPM, and SURF-ScSPM all encode local features using BoW.

It is interesting to note that the classification rates of HOG-SPM and SIFT-SPM improve by 8.3% and 12.0%, respectively, when using RBF-SVM compared to using linear-SVM. In addition, using HIK-SVM increases the classification rates of HOG-SPM and SIFT-SPM by 9.4% and 17.0%, respectively. However, for SIFT-LLC, SIFT-ScSPM, and SIFT-FV, the classification performance is not improved by using RBF-SVM and HIK-SVM. Note that previous tests [64, 65] on

several benchmark data sets also indicate that the choice of the SVM kernel does not affect the performance of LLC and ScSPM significantly.

SURF-ScSPM is also a BOW-based descriptor. However, it extracts local features from the interest point patches. *CRs* of SURF-ScSPM is lower than *CRs* of SIFT-ScSPM, SIFT-LLC, and HOG-SPM, which extract local features from the dense patches. The result indicates that sparse local features can not carry enough information for scene categorization compared to dense local features.

LBP, uniform LBP, LBP-HF, PLBP, and CENTRIST have higher classification rates than HMAX on the 15-scene data set. PLBP has the highest classification rate (73.0%) among the LBP-based descriptors. For the original LBP, uniform LBP, and PLBP, scene categorization performance is better using HIK-SVM than linear-SVM and RBF-SVM. The highest classification rates of LBP, uniform LBP, PLBP are 71.9%, 70.6%, and 73.0%, respectively. For the LBP-HF algorithm, a higher classification rate (67.3%) is achieved using RBF-SVM, compared to linear-SVM (64.9%) and HIK-SVM (66.4%). The highest *CR* of CENTRIST is 72.7%, achieved with linear-SVM. In fact, CENTRIST is also a LBP-based algorithm because it extracts local features using the LBP feature map. The difference is that CENTRIST forms the global feature vector using the spatial PCA, whereas LBP, uniform LBP, and LBP-HF form the global feature vector using histograms. PCA with 3 spatial levels accounts for the higher performance of CENTRIST over the original LBP algorithm.

OB has a higher classification rate (79.9%) than SIFT-SPM, LBP-based algorithms, and the biologically-inspired algorithms on the 15-scene data set. The OB algorithm achieves its highest classification rate of 79.9% when using linear-SVM.

A similar observation was reported in [59]; OB performs better than SIFT-SPM on the UIUC-sport-event data set, the 15-scene data set, and the 67-indoor scene data set. However, in our experiment, OB has a lower *CR* than SIFT-ScSPM and SIFT-LLC on the 15-scene data set. This result indicates that the global feature formation used in the OB is not as good as in SIFT-ScSPM and SIFT-LLC.

Based on this experiment, we determined a suitable SVM kernel for each of the descriptors. The selected SVM classifiers were used in the subsequent experiments, where we evaluated the visual descriptors on three other data sets: the 8-outdoor-scene, the 67-indoor-scene, and the SUN397 data sets. The aim of the subsequent experiments is to identify the algorithms that have consistent performance on multiple data sets.

3.4.2 Classification results on the 8-outdoor-scene data set

Table 3.5 shows scene categorization results of the 14 selected descriptors on the 8-outdoor-scene data set. Among the 14 descriptors, SIFT-ScSPM has the highest *CR* (89.8%). SIFT-ScSPM, SIFT-LLC, SIFT-FV, HOG-SPM, OB, SIFT-SPM, and SURF-ScSPM are the top-7 algorithms, listed in a descending order of *CR*. For the last 7 descriptors, the classification rates of GIST, CENTRIST and PLBP are higher than 80.0%. Note that HMAX achieves a higher *CR* (79.8%) on the outdoor scene categorization than three LBP-based algorithms (76.6% for LBP, 75.2% for uniform LBP, and 71.9% for LBP-HF). This result indicates that the biologically-inspired features are useful for outdoor scene categorization.

Table 3.5: Scene categorization performance on the 8-nature-outdoor-scene data set.

ID	*Algorithms*	*CR (%)*	*Precision (%)*	*Recall (%)*	*F-measure (%)*	*AUC (%)*
1	SIFT-ScSPM (linear)	**89.8 ± 1.8**	**90.4 ± 1.8**	**90.0 ± 2.2**	**90.0 ± 2.1**	**98.4 ± 0.4**
2	SIFT-LLC (linear)	88.1 ± 2.1	88.6 ± 2.0	88.3 ± 2.6	88.3 ± 2.4	98.2 ± 0.6
3	SIFT-FV (linear)	88.1 ± 3.3	88.2 ± 3.4	88.0 ± 3.9	87.9 ± 3.8	98.7 ± 0.4
4	HOG-SPM(HIK)	88.1 ± 2.5	89.0 ± 2.1	88.2 ± 3.2	88.3 ± 2.9	98.3 ± 0.6
5	OB (linear)	87.5 ± 2.5	88.0 ± 2.3	87.6 ± 3.1	87.5 ± 2.8	98.4 ± 0.5
6	SIFT-SPM(HIK)	87.4 ± 2.3	87.9 ± 2.2	87.8 ± 2.5	87.6 ± 2.4	98.1 ± 0.4
7	SURF-ScSPM (linear)	85.9 ± 2.6	85.4 ± 2.8	85.0 ± 3.4	85.9 ± 3.2	98.0 ± 0.5
8	GIST (RBF)	85.3 ± 2.5	85.9 ± 2.4	85.4 ± 3.1	85.4 ± 2.9	98.0 ± 0.8
9	CENTRIST (linear)	83.5 ± 4.9	84.3 ± 4.8	83.6 ± 5.8	83.5 ± 5.5	97.2 ± 1.1
10	LBP (HIK)	76.6 ± 4.3	77.0 ± 4.6	76.8 ± 5.1	76.5 ± 4.9	95.8 ± 1.2
11	Uniform LBP (HIK)	75.2 ± 4.6	76.0 ± 4.6	75.2 ± 5.8	74.9 ± 5.5	95.0 ± 1.6
12	LBP-HF (RBF)	71.9 ± 3.7	72.1 ± 4.5	71.9 ± 4.7	71.4 ± 4.6	92.5 ± 2.6
13	PLBP (HIK)	81.4 ± 2.9	81.6 ± 3.6	81.3 ± 3.7	81.2 ± 3.8	96.7 ± 1.1
14	HMAX (HIK)	79.8 ± 0.8	80.1 ± 0.7	80.2 ± 0.7	80.0 ± 0.7	96.5 ± 0.3

3.4.3 Classification results on the 67-indoor-scene data set

Table 3.6 shows scene categorization results of the 14 selected descriptors on the 67-indoor-scene data set. The SIFT-ScSPM still has the highest *CR* (45.6%) compared with the other 13 descriptors. The top-7 descriptors are SIFT-ScSPM, OB, SIFT-LLC, SIFT-FV, SURF-ScSPM, HOG-SPM, and SIFT-SPM, listed in a descending order of *CR*. These seven algorithms (except for OB) use BoW methods for global feature formation. These results indicate that using BoW methods is more robust than using histograms, PCA and down-sampling, especially when the complexity of images is increased. Note that the OB descriptor outperformed most of the BoW methods on the 67-indoor-scene data set. This result indicates that the object information is useful for indoor scene categorization.

From Tables 3.2 to 3.6, we can see that the ranking based on different measures were consistent for the top-seven algorithms. A higher *CR* was also accompanied by a higher precision, recall, F-measure, and *AUC* values.

Table 3.6: Scene categorization performance on the 67-indoor data set.

ID	*Algorithms*	*CR (%)*	*Precision (%)*	*Recall (%)*	*F-measure (%)*	*AUC (%)*
1	SIFT-ScSPM (linear)	**45.6 ± 1.0**	**45.4 ± 2.5**	**35.8 ± 1.5**	**36.9 ± 1.7**	**91.7 ± 0.4**
2	SIFT-LLC (linear)	43.9 ± 0.4	44.7 ± 1.2	35.2 ± 0.8	36.3 ± 0.9	91.3 ± 0.5
3	SIFT-FV (linear)	41.5 ± 1.2	46.0 ± 1.3	31.4 ± 1.0	32.5 ± 1.0	90.4 ± 0.6
4	HOG-SPM (HIK)	31.0 ± 0.5	27.5 ± 0.8	25.5 ± 0.6	25.8 ± 0.6	85.1 ± 0.6
5	OB (linear)	45.3 ± 0.5	43.7 ± 1.0	39.4 ± 0.7	39.2 ± 0.7	91.7 ± 0.4
6	SIFT-SPM (HIK)	31.4 ± 0.8	28.4 ± 1.1	25.9 ± 0.7	26.4 ± 0.8	85.3 ± 0.5
7	SURF-ScSPM (linear)	34.5 ± 0.7	35.5 ± 0.9	25.4 ± 0.4	26.2 ± 0.2	87.6 ± 0.3
8	GIST (RBF)	30.9 ± 0.9	28.0 ± 0.1	25.2 ± 0.3	25.7 ± 0.4	84.4 ± 0.4
9	CENTRIST (linear)	12.2 ± 11.0	15.4 ± 8.5	10.8 ± 9.6	10.1 ± 9.6	78.4 ± 3.8
10	LBP (HIK)	22.9 ± 0.6	21.0 ± 1.0	17.7 ± 0.9	18.4 ± 0.9	81.7 ± 0.5
11	Uniform LBP (HIK)	22.0 ± 1.1	18.5 ± 0.4	16.2 ± 1.0	16.6 ± 0.9	80.0 ± 0.6
12	LBP-HF (RBF)	15.4 ± 1.2	12.7 ± 0.8	11.9 ± 0.7	11.8 ± 0.7	76.3 ± 1.0
13	PLBP (HIK)	27.2 ± 1.0	24.1 ± 1.3	21.6 ± 1.0	22.2 ± 1.0	84.5 ± 0.3
14	HMAX (HIK)	11.6 ± 2.3	10.5 ± 2.7	9.4 ± 1.9	9.5 ± 2.1	72.9 ± 3.0

3.4.4 Classification results on the SUN397 data set

Using the SUN397 data set, we compared the 14 visual descriptors and 4 recent methods based on deep learning: OverFeat [70], DeCAF [53], ImageNet-CNN [52], and Places-CNN [69]. Note that training a CNN for scene categorization on a large data set requires significant engineering efforts for parameter tuning. To achieve a fair comparison, we evaluated the 14 visual descriptors using the same evaluation protocol described in [4] for the SUN397 data set. This data set is divided into fixed partitions. In each partition, 50 training images and 50 test images per class are used for evaluation. The classification rate, averaged over the fixed partitions, is used for comparison. The four deep learning methods had been evaluated using the same protocol, and their results have been reported in [4, 53, 69].

Figure 3.1 presents *CRs* and their standard deviation of the 14 visual descriptors and the four deep-learning methods (OverFeat, DeCAF, ImageNet-CNN, and Places-CNN) on the SUN397 data sets. All the four deep features outperform the other visual descriptors. However, even the best algorithm (Places-CNN) had a

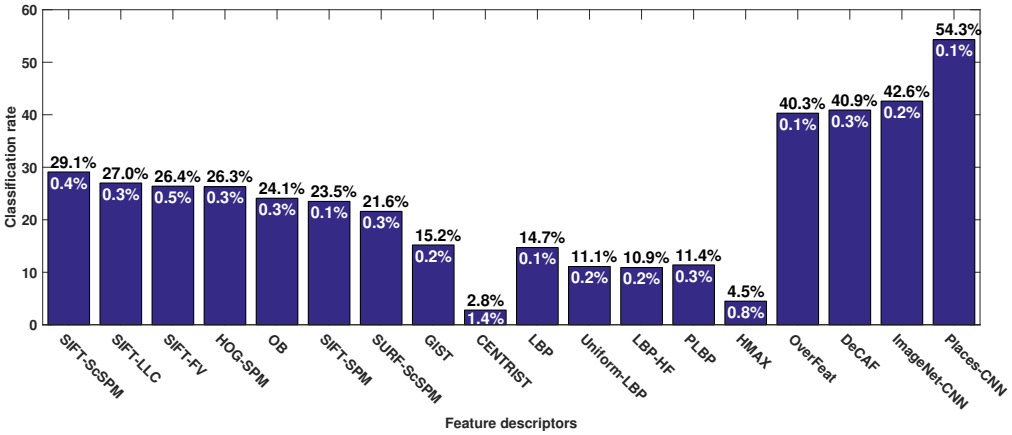


Figure 3.1: Comparison of scene categorization methods on the SUN397 data set. For each method, the top number in black is the classification rate, and the bottom number in white is the standard deviation.

CR of only 54.3%, which is still significantly lower than human performance of 68.0% [4].

To rank the data sets in terms of their degree of difficulty, we compared the average classification rates of the top 7 descriptors on each of the four data set. The highest average *CR* of 87.8% is obtained with the 8-outdoor-scene data set, compared to 79.8% for the 15-scene data set, 39.0% for the 67-indoor-scene data set, and 25.4% for the SUN397 data set. These results indicate that among the four data sets, the 8-outdoor-scene data set is the easiest and the SUN397 data set is the hardest to classify. Similar ranking is obtained if we use the median *CR* on each data set. Apart from the difference in the number of images and scene categories, the 8-outdoor-scene data set consists of only outdoor images, whereas the SUN397 data set contains not only outdoor scenes but also indoor and man-made scenes.

3.5 Class separability and stability of feature vectors

We evaluated the class separability of the feature vectors using the Fisher score S (see Section 3.2). Note that this evaluation is independent of the classifier used.

Table 3.7 presents the class separability scores (S) for the compared features on the four data sets. Among the biologically-inspired features, HMAX has a low S score on the four data sets. The HMAX features are formed by using max pooling in layer C1, BoW in layer S2, and max pooling in layer C2. The GIST has a higher S score than the HMAX, CENTRIST, OB, and the LBP-based features. As shown in Tables 3.2 to 3.6, the GIST algorithm also has higher classification rates than OB, HMAX, LBP, uniform LBP, and LBP-HF. However, the GIST algorithm has a lower class separability score than BoW algorithms (SIFT-ScSPM, SURF-ScSPM, SIFT-LLC, SIFT-SPM, and HOG-SPM). Note that GIST forms global features using only down-sampling and averaging. This result indicates that biologically-inspired features can benefit from better schemes for forming global features.

Among the 14 descriptors, SIFT-FV (which calculates SIFT features on dense-grids, and forms global features using the Fisher kernel coding) has the highest S score on the 8-outdoor-scene data set (2.5470), the 15-outdoor-scene data set (2.3111), the 67-indoor-scene data set (1.1484), and the SUN397 data set (1.0132). The result indicates the Fisher Vector extracts discriminative global features. Note that in our experiment, SIFT-FV has lower *CRs* than SIFT-ScSPM on the four data sets. However, in the paper of [63], *CR* of SIFT-FV on the SUN397 data set is 43.3%, which is higher than other hand-designed features compared in our experiments. The reason is SIFT-FV in [63] extracts local SIFT features from overlapping patches

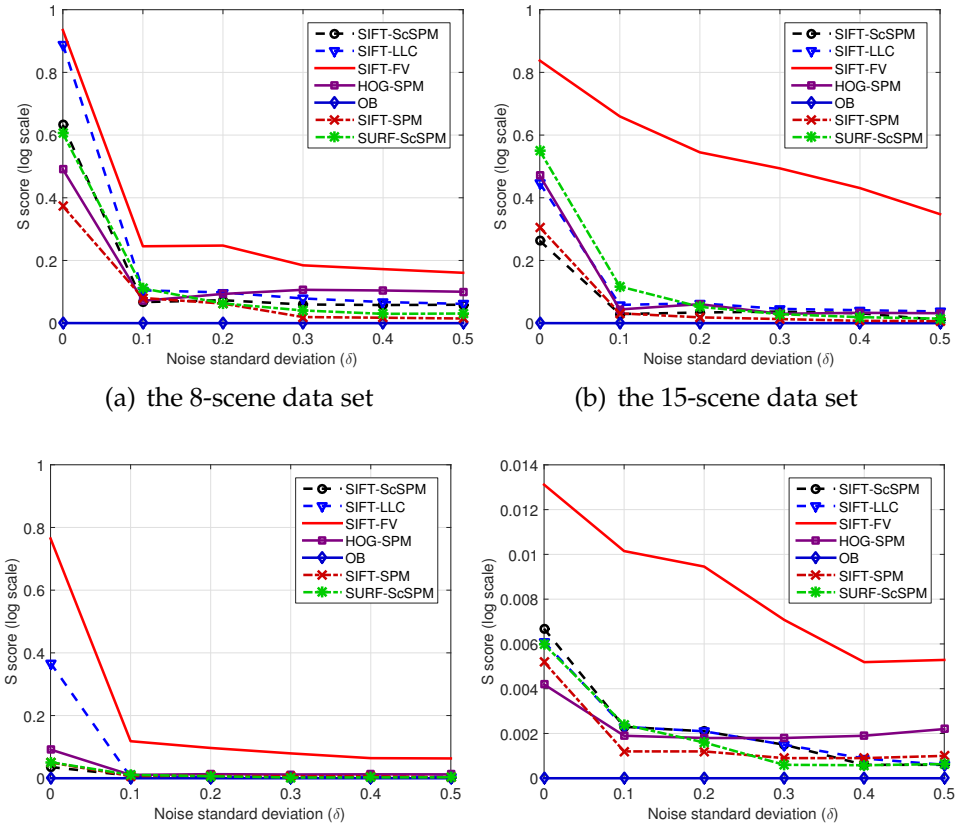
Table 3.7: The S score for class separability of feature vectors. A high value of S means the extracted scene categories are highly separable using the given feature vector.

ID	*Algorithms*	*8-outdoor-scene*	*15-scene*	*67-indoor-scene*	*SUN397*
1	SIFT-ScSPM	1.8827	1.3005	1.0364	1.0064
2	SIFT-LLC	2.4339	1.5626	1.1441	1.0061
3	SIFT-FV	**2.5470**	**2.3111**	**1.1484**	**1.0132**
4	HOG-SPM	1.6340	1.6010	1.0959	1.0042
5	OB	1.0003	1.0001	1.0001	1.0000
6	SIFT-SPM	1.4521	1.3573	1.0516	1.0052
7	SURF-ScSPM	1.8323	1.7330	1.0525	1.0060
8	CENTRIST	1.0045	1.0039	1.0000	1.0000
9	GIST	1.0604	1.0283	1.0047	1.0008
10	LBP	1.0434	1.0136	1.0021	1.0004
11	Uniform LBP	1.0187	1.0070	1.0015	1.0003
12	LBP-HF	1.0146	1.0091	1.0012	1.0002
13	PLBP	1.0630	1.0255	1.0034	1.0002
14	HMAX	1.0006	1.0003	1.0004	1.0001

(24×24) on a regular grid every 4 pixels at 5 scales. SIFT-FV in our experiment extracts SIFT features from overlapping patches (16×16) on a regular grid every 8 pixels at one scale. The difference between *CRs* of SIFT-FV in [63] and in this chapter indicates that informative local features and discriminative global feature formation methods can improve the classification performance.

The other BoW algorithms (SIFT-LLC, SIFT-ScSPM, SIFT-SPM, HOG-SPM, and SURF-ScSPM) also has higher S scores than the LBP-based algorithms (LBP, uniform LBP, LBP-HF, PLBP, and CENTRIST) on the four data sets. Note that to form global features, SIFT-ScSPM and SIFT-LLC combine the BoW algorithms with spatial histograms and max pooling, whereas the LBP-based methods only use histograms.

SURF-ScSPM has lower S score than most of the BOW-based descriptors, such as SIFT-ScSPM, SIFT-LLC, and SIFT-FV, but it has higher S score than OB, SIFT-SPM, LBP-based, and biologically-inspired descriptors. This shows that compared to SIFT-ScSPM, the discriminative power of SURF-ScSPM is reduced



(a) the 8-scene data set (b) the 15-scene data set

(c) the 67-indoor-scene data set (d) the SUN397 data set

Figure 3.2: Stability of features under the presence of Gaussian noise of varying standard deviation, on the four data sets.

by extracting the SURF features from key points. The S score of SURF-ScSPM is 1.8323 on the 8-outdoor-scene data set, 1.7330 on the 15-outdoor-scene data set, 1.0525 on the 67-indoor-scene data set, and 1.0060 on the SUN397 data set.

The LBP-based features (LBP, uniform LBP, PLBP, LBP-HF, and CENTRIST) have lower S scores than the BoW features. However, the LBP-based features achieve higher S scores than OB and HMAX. Note that the LBP-based features are more efficient to compute than the BoW features. Among the LBP-based algorithms, PLBP achieved the highest class separability score. This is also reflected in the higher CR of PLBP, compared to LBP, uniform LPB, LBP-HF, and CENTRIST (see Section 3.4). Uniform LBP and LBP-HF reduce the dimension

and also the class separability of features. This result indicates that the class separability and classification accuracy of the LBP-based algorithms can be improved by using a better global feature formation, instead of the histograms.

OB has the lowest S score among the compared features. Its S score is 1.0003 on the 8-outdoor-scene data set, 1.0001 on the 15-outdoor-scene data set, 1.0001 on the 67-indoor-scene data set, and 1.0000 on the SUN397 data set. The reason may be that many similar objects appear in different scene categories. Note that OB uses only max pooling to form global features, and it extracts a large number of features (44604 per image).

Next, we evaluated the stability of the scene categorization algorithms in the presence of noise. In this experiment, Gaussian noise with varied standard deviation was added to the original images. Then, the class separability scores were computed for the noisy images. Figure 3.2 shows the results for the top-7 algorithms: SIFT-ScSPM, SIFT-LLC, SIFT-FV, HOG-SPM, OB, SIFT-SPM, and SURF-ScSPM. These algorithms (based on the bag-of-words) are identified as having high classification accuracy in Section 3.4. When noise is added, the class separability (S score) of all features reduces. At all noise levels, the SIFT-FV descriptors has a higher S score than all other descriptors.

The results presented in this section indicate that the method for forming global features affects the class separability significantly. Using BoW algorithms before applying histograms, PCA or max pooling (as in SIFT-FV, SIFT-ScSPM, and HOG-SPM) produces feature vectors with more discriminative power. Using histograms as the first step of the global feature formation (as in CENTRIST) decreases the class separability of features. Using only one method for global feature formation

(as in the LBP-based algorithms) does not yield high separability scores and nor high classification rates.

3.6 Chapter summary

This chapter presented an experimental evaluation of existing visual descriptors for scene categorization. The existing benchmark data sets and performance measures for scene categorization were also discussed.

The experimental results indicate that SIFT-ScSPM outperforms all other tested descriptors on the 15-scene data set. SIFT-ScSPM uses SIFT as its local descriptor and ScSPM as its global feature formation. Local descriptors, SIFT, HOG, and SURF achieve higher classification rates than LBP, CENTRIST, and HMAX. The global feature formation methods affect the class separability of feature vectors significantly. Using ScSPM, LLC, and FV for global feature formation leads to higher class separability than using histograms and PCA. Using BoW before histograms, PCA, and max pooling makes feature vectors more distinguishable. The results on 67-indoor-scene data set show that the mid-level features like objects, bag-of-parts, and the efficient patch encoding algorithm like Fisher Vector improve the classification rates for indoor scenes. The results on the SUN397 data set indicate that SIFT-ScSPM outperforms all other hand-designed descriptors. The learned features produced by deep learning establish the new state-of-the art performance in scene categorization. However, there is still a large performance gap between the best computational algorithm and humans.

Based on this survey and evaluation, several promising research directions can be highlighted. First, local feature descriptors can be built that combine

the properties of SIFT, HOG, SURF, GIST, or the early stages of deep learning architecture. A good local descriptor leads to a high classification rate. Second, global feature formation algorithms can be developed based on ScSPM and FV. Third, most existing studies on gist recognition have been concerned with static scenes, which is the focus of this study. In recent years, gist recognition of dynamic scenes has attracted the attention of researchers [170, 171, 172, 173], and therefore, the extension of this study to dynamic scenes would be invaluable.

Chapter 4

Image normalization for affine deformations

Chapter contents

4.1 Introduction

This chapter * describes a new image normalization algorithm for affine deformations. A major goal of a visual system (natural or machine) is to recognize objects that are visible in the scene, regardless of their location or pose relative to the viewer. Humans can recognize objects from different viewpoints and in different arrangements. However, machines have difficulties to recognize objects undergone affine deformations, such as translation, scaling, shearing, and rotation. The affine invariants are required for many computer vision applications, including handwritten digit recognition [174, 175], texture recognition [176, 177, 178], face matching [179], and face recognition [180, 181]. Therefore, extracting affine invariance is a key for efficient image recognition.

The existing approaches to achieve affine invariants include training classifiers using samples with affine deformations [182, 183, 184], extracting affine-invariant features [36, 185, 186, 187, 188], and normalizing affine distorted images [189, 190, 191, 192, 193]. The invariance by brute-force training techniques can easily be applied to image recognition. However, it is time consuming. If the training set is not carefully designed, the classifier may not learn the desired invariance.

The invariance by feature techniques, like SIFT [36], SURF [66], and BRISK [72] has been broadly used to extract some forms of affine invariance. In these descriptors, scale-invariance is achieved by scale-space keypoint detection, whereas rotation-invariance is achieved by orientation assignment. However, the extrac-

*Parts of Chapter 4 have been published in our paper "Affine-invariant scene categorization," *IEEE International Conference on Image Processing*, pp. 1031-1035, 2014. Chapter 4 has been submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

tion of fully affine-invariant features is still a challenge.

The invariance by image normalization techniques normalize input images geometrically before image features are extracted and classified. The image normalization is a pre-processing strategy that transforms original and distorted images into their normalized form. At the same time, the normalized images retain all the relevant information of the original images. Affine normalization methods have been used for image watermarking [194], pattern matching [195, 196], and handwritten character recognition [174, 175, 197].

Affine-invariant image normalization has been attempted by different approaches. Yasein and Agathoklis developed an affine normalization method using feature points [198]. The normalization matrix is estimated from three points that have the highest responses during a feature-detection stage. A disadvantage of the point-based method is that point matching is needed between original images and distorted images. Recently, Zhang *et al.* recovered the affine and projective deformations by minimizing low-rank matrix of images [192]. This method works well for regular and near-regular patterns or objects (e.g. building facades, printed text, and human faces). However, for most non-regular patterns, the normalization performance still needs to be improved.

Several existing affine normalization methods are proposed based on image moments. For example, Pei and Lin used the covariance matrix of moments to normalize images [199]. Rothe *et al.* used moment constraints to normalize images via a sequence of transformations [190]. Sheng and Ip developed a moment-based normalization method to handle shaped planar images [200]. Suk and Flusser decomposed affine distortions and formed normalized images by low-

order moments [191]. Zhang *et al.* studied the ambiguities of the moment-based normalization methods, and introduced a strategy to choose a consistent result [201]. This method produces a consistent output for the same pattern under artificial affine distortions.

In this chapter, we propose a novel moment-based image normalization method to achieve fully affine invariants. We present experimental results to compare the image normalization accuracies and to study how the image normalization affects class separability on several benchmark data sets. We also analyze the effects of image noise and image cropping on the image normalization. The rest of the chapter is structured as follows. Section 4.2 describes the proposed image normalization approach to achieve affine invariance. Section 4.3 analyzes the results of image normalization on several benchmark data sets, and Section 4.4 gives the concluding remarks.

4.2 Image normalization for affine distortions

In this section, we present a new image normalization approach based on new moment constraints. Let $\mathcal{I}$ denote the set of all images generated by arbitrary

Table 4.1: Major types of affine transformations.

Type	*Transformation matrix*	*Comment*
Translation	$T_{\text{tr}} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix}$	t_x and t_y are the shift parameters along the x and y axes.
Scale	$T_{\text{sc}} = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$	s_x and s_y are the scaling factors in the x and y directions, respectively.
Rotation	$T_{\text{ro}} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$	θ is the rotation angle in the clockwise direction about the origin $(0,0)$.
Shear	$T_{\text{sh}} = \begin{pmatrix} 1 & h_x & 0 \\ h_y & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	h_x and h_y are the shear parameters in the x and y directions, respectively.
Affine	$T = \begin{pmatrix} t_1 & t_2 & t_3 \\ t_4 & t_5 & t_6 \\ 0 & 0 & 1 \end{pmatrix}$	t_1 to t_6 are the six parameters for a general affine transformation.

affine transformations of an image I. The proposed method aims to map the set $\mathcal{I}$ to a small and finite set $\hat{\mathcal{I}}$ using moment-based image normalization. This section is organized as follows. Subsection 4.2.1 gives a brief introduction to image moments and describes the key proposition, which establishes a relationship between the moments of a source image and an affine-transformed image. Subsection 4.2.2 shows the proposed image normalization approach as an optimization problem involving low-order image moments, whereas Subsection 4.2.3 describes analytical solutions of the moment equations. Subsection 4.2.4 describes the affine-normalization algorithm and illustrates its effects on several image data sets. Subsection 4.2.5 compares rotation invariants calculated by the proposed method and the principal axes method. Subsection 4.2.6 describes a method to produce a consistent order of normalized images. Subsection 4.2.7 discusses the relation between the proposed normalization algorithm and the existing moment-based normalization algorithms.

4.2.1 Image moments and moment propositions

An affine transformation is characterized by a transformation matrix T with 6 real parameters:

$$T = \begin{pmatrix} t_1 & t_2 & t_3 \\ t_4 & t_5 & t_6 \\ 0 & 0 & 1 \end{pmatrix}. \tag{4.1}$$

A pixel coordinate (x, y) in the input image I is mapped to a pixel coordinate (x', y') in the output image I' as

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = T \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}. \tag{4.2}$$

The common affine transformations are listed in Table 4.1. Note that applying an affine transformation T_1 followed by an affine transformation T_2 is equivalent to

applying an affine transformation $T = T_1 T_2$. Furthermore, an affine transformation can be decomposed into a sequence of translation, scaling, shearing, and rotation.

From a given input image $I(x, y)$, the moment-based normalization method calculates the required transformation matrices T by setting low-order moments to constants. The geometric moment $m_{p,q}$ of order (p, q) for image $I(x, y)$ is defined as

$$m_{p,q} = \iint_{\Gamma} x^p y^q I(x, y) dx dy, \tag{4.3}$$

where Γ denotes the support of the image. The normalized geometric moment of order (p, q) is given as

$$\nu_{p,q} = \frac{m_{p,q}}{m_{0,0}}. \tag{4.4}$$

Other types of image moments are also formulated to improve invariance to translation. The central moment $\mu_{p,q}$ of image $I(x, y)$ is

$$\mu_{p,q} = \iint_{\Gamma} (x - \overline{x})^p (y - \overline{y})^q I(x, y) dx dy, \tag{4.5}$$

where $\overline{x} = \nu_{1,0}$ and $\overline{y} = \nu_{0,1}$. Similarly to (4.4), the normalized central moment is defined as

$$\eta_{p,q} = \frac{\mu_{p,q}}{\mu_{0,0}}. \tag{4.6}$$

Under an affine transformation, the transformed moments of the output image are related to the moments of the input image according to *Proposition* 1.

Remark: A proof of *Proposition* 1 is provided in Appendix 8.1. Equation (4.9) with four parameters t_1, t_2, t_4, and t_5 has been reported in [202]. In this section, we present and prove more generalized equations involving six affine transformation parameters.

Proposition 1. *Under the affine transformation represented by matrix T with six free parameters (t_1, t_2, t_3, t_4, t_5, and t_6), the moments $m'_{p,q}$, $\nu'_{p,q}$, $\mu'_{p,q}$, and $\eta'_{p,q}$ of the output image $I'(x', y')$ are related to the moments of the input image $I(x, y)$ as*

$$m'_{p,q} = \det(J) \sum_{(i,j)\in S_p} \sum_{(k,l)\in S_q} \binom{p}{i,j,p-i-j} \binom{q}{k,l,q-k-l} \times t_1^i\, t_2^j\, t_4^k\, t_5^l\, t_3^{p-i-j}\, t_6^{q-k-l}\, m_{i+k,j+l}, \tag{4.7}$$

$$\nu'_{p,q} = \sum_{(i,j)\in S_p} \sum_{(k,l)\in S_q} \binom{p}{i,j,p-i-j} \binom{q}{k,l,q-k-l} \times t_1^i\, t_2^j\, t_4^k\, t_5^l\, t_3^{p-i-j}\, t_6^{q-k-l}\, \nu_{i+k,j+l}, \tag{4.8}$$

$$\mu'_{p,q} = \det(J) \sum_{i=0}^{p} \sum_{j=0}^{q} \binom{p}{i}\binom{q}{j} t_1^i\, t_2^{p-i}\, t_4^j\, t_5^{q-j}\, \mu_{i+j,p+q-i-j}, \tag{4.9}$$

$$\eta'_{p,q} = \sum_{i=0}^{p} \sum_{j=0}^{q} \binom{p}{i}\binom{q}{j} t_1^i\, t_2^{p-i}\, t_4^j\, t_5^{q-j}\, \eta_{i+j,p+q-i-j}, \tag{4.10}$$

where $J = \begin{pmatrix} t_1 & t_2 \\ t_4 & t_5 \end{pmatrix}$, *and* $S_r = \{(u, v) \in \mathbb{N}^2 \mid u + v \le r\}$.

Applying *Proposition* 1, several examples of the transformed moments are obtained from the input moments as follows:

$$\begin{cases} \nu'_{1,0} = t_3 + t_1\nu_{1,0} + t_2\nu_{0,1}, \\ \nu'_{0,1} = t_6 + t_4\nu_{1,0} + t_5\nu_{0,1}, \\ \eta'_{2,0} = t_2^2\eta_{0,2} + 2t_1t_2\eta_{1,1} + t_1^2\eta_{2,0}, \\ \eta'_{0,2} = t_5^2\eta_{0,2} + 2t_4t_5\eta_{1,1} + t_4^2\eta_{2,0}, \\ \eta'_{1,1} = t_2t_5\eta_{0,2} + (t_2t_4 + t_1t_5)\eta_{1,1} + t_1t_4\eta_{2,0}, \\ \eta'_{1,2} = t_2t_5^2\eta_{0,3} + (t_1t_5^2 + 2t_2t_4t_5)\eta_{1,2} + t_1t_4^2\eta_{3,0} + (t_2t_4^2 + 2t_1t_4t_5)\eta_{2,1}, \\ \eta'_{2,1} = t_2^2t_5\eta_{0,3} + (t_2^2t_4 + 2t_1t_2t_5)\eta_{1,2} + t_1^2t_4\eta_{3,0} + (t_1^2t_5 + 2t_1t_2t_4)\eta_{2,1}, \\ \eta'_{2,2} = t_1^2t_4^2\eta_{4,0} + (2t_1t_2t_5^2 + 2t_2^2t_4t_5)\eta_{1,3} + t_2^2t_5^2\eta_{0,4} + (2t_1t_2t_4^2 + 2t_1^2t_4t_5)\eta_{3,1} + \\ \qquad (t_1^2t_5^2 + 4t_1t_2t_4t_5 + t_2^2t_4^2)\eta_{2,2}. \end{cases} \tag{4.11}$$

4.2.2 Formulation of the proposed moment constraints

In the proposed affine-normalization approach, transformation matrices T are found by solving the following constrained optimization problem:

$$\underset{T}{\text{maximize}} \{\eta'_{2,2}\} \text{ subject to } \begin{cases} \nu'_{1,0} & = c_1, \\ \nu'_{0,1} & = c_2, \\ \eta'_{2,0} & = c_3, \\ \eta'_{0,2} & = c_4, \\ \eta'_{1,1} & = c_5, \end{cases} \tag{4.12}$$

where c_1, c_2, c_3, c_4, and c_5 are five fixed parameters.

We explain the inspirations for the constraints in (4.12) by analyzing special cases of affine transformations (see Table 4.1). First, consider the case where there is only translation along the x and y direction. The transformation parameters are $t_1 = t_5 = 1$, $t_2 = t_4 = 0$, $t_3 = t_x$, and $t_6 = t_y$. From (4.11), we obtain $\nu'_{1,0} = t_3 + \nu_{1,0}$, and $\nu'_{0,1} = t_6 + \nu_{0,1}$. Therefore, setting $\nu'_{1,0}$ and $\nu'_{0,1}$ to a fixed value will determine t_3 and t_6 for the inverse translation, and thereby normalizing the image against translation.

Next, consider the case where there is only scaling along the x and y direction. The transformation parameters are $t_1 = s_x$, $t_5 = s_y$, and $t_2 = t_3 = t_4 = t_6 = 0$. From (4.11), we obtain $\eta'_{2,0} = t_1^2\, \eta_{2,0}$ and $\eta'_{0,2} = t_5^2\, \eta_{0,2}$. Therefore, setting $\eta'_{2,0}$ and $\eta'_{0,2}$ to a fixed value will normalize the image against scaling. Note that four pairs (t_1, t_5) of alternating signs will be produced.

Then, consider the case where there is only shearing along the x direction. The transformation parameters are $t_2 = h_x$, $t_1 = t_5 = 1$, and $t_3 = t_4 = t_6 = 0$. From (4.11), we obtain $\eta'_{1,1} = t_2\eta_{0,2} + \eta_{1,1}$. Consider the case where there is only shearing along the y direction. The transformation parameters are $t_4 = h_y$, $t_1 = t_5 = 1$, and $t_2 = t_3 = t_6 = 0$. From (4.11), we obtain $\eta'_{1,1} = t_4\eta_{2,0} + \eta_{1,1}$. Therefore, setting $\eta'_{1,1}$ to

a fixed value will normalize the image against shearing.

Furthermore, because $\eta'_{1,1} = t_2 t_5 \eta_{0,2} + (t_2 t_4 + t_1 t_5)\eta_{1,1} + t_1 t_4 \eta_{2,0}$, combining the constraint $\eta'_{1,1} = c_5$ and the other two constraints ($\eta'_{2,0} = c_3$ and $\eta'_{0,2} = c_4$) will produce the required parameters t_1, t_2, t_4, and t_5. Subsequently, parameters t_3 and t_6 can be determined using the constraints $\nu'_{1,0} = c_1$ and $\nu'_{0,1} = c_2$.

Next, consider the case where there is only rotation by an angle θ in the counter-clockwise direction. The transformation parameters are $t_1 = \cos\theta$, $t_2 = -\sin\theta$, $t_4 = \sin\theta$, $t_5 = \cos\theta$, and $t_3 = t_6 = 0$. From (4.11), we can show that

$$\eta'_{2,2} = \frac{1}{8}(6\eta_{2,2} - \eta_{0,4} - \eta_{4,0})\cos 4\theta + \frac{1}{2}(\eta_{3,1} - \eta_{1,3})\sin 4\theta + \frac{1}{8}(2\eta_{2,2} + \eta_{4,0} + \eta_{0,4}). \tag{4.13}$$

Therefore, maximizing $\eta'_{2,2}$ will produce invariant angle and normalize the image against rotation. In Section 4.2.5, the relation between moment $\eta'_{2,2}$ and principal axes are discussed.

Several factors are considered in formulating the equations in (4.12). First, it should involve image moment of low-order to reduce the computation cost. Second, the fixed values c_1, c_2, c_3, c_4, and c_5 should be selected so that a real solution of (4.12) can be computed efficiently. In this method, we use the values: $c_1 = 0$, $c_2 = 0$, $c_3 = c^2$, $c_4 = c^2$, and $c_5 = 0$. The required transformation matrix for affine normalization is found as follows:

$$\hat{T} = \arg\max_{T} \{\eta'_{2,2}\} \text{ subject to } \begin{cases} \nu'_{1,0} & = 0, \\ \nu'_{0,1} & = 0, \\ \eta'_{2,0} & = c^2, \\ \eta'_{0,2} & = c^2, \\ \eta'_{1,1} & = 0, \end{cases} \tag{4.14}$$

where c is fixed positive parameter to control the size of the normalized image. The next subsection will describe how the optimization problem in (4.14) is solved.

4.2.3 Solutions of the moment constraints

First of all, using the Cauchy-Schwarz inequality, we can show that $\eta_{2,0}\eta_{0,2} - \eta_{1,1}^2 \geq 0$. Therefore, for simplicity we can denote $\eta_{2,0}\eta_{0,2} - \eta_{1,1}^2 = D^2$, where $D \geq 0$.

Definition 1. *An image I is called moment-normalizable if it satisfies the following two conditions:*

$$A \neq 0 \text{ and } B \neq 0, \tag{4.15}$$

where

$$\begin{aligned} A = {} & 16\eta_{2,0}\eta_{3,1}\eta_{1,1}^3 + 8\eta_{0,2}\eta_{2,0}\eta_{4,0}\eta_{1,1}^2 + 6\eta_{0,2}\eta_{2,2}\eta_{2,0}^3 \\ & + 4\eta_{1,1}\eta_{1,3}\eta_{2,0}^3 - 12\eta_{2,2}\eta_{1,1}^2\eta_{2,0}^2 - 12\eta_{0,2}\eta_{1,1}\eta_{3,1}\eta_{2,0}^2 \\ & - 8\eta_{4,0}\eta_{1,1}^4 - \eta_{0,4}\eta_{2,0}^4 - \eta_{4,0}\eta_{0,2}^2\eta_{2,0}^2 \end{aligned} \tag{4.16}$$

$$\begin{aligned} B = {} & 4D\,[3\eta_{2,0}\eta_{3,1}\eta_{1,1}^2 + \eta_{1,1}\eta_{4,0}D^2 + \eta_{1,3}\eta_{2,0}^3 \\ & - 3\eta_{1,1}\eta_{2,2}\eta_{2,0}^2 - \eta_{2,0}\eta_{3,1}D^2 - \eta_{4,0}\eta_{1,1}^3]. \end{aligned} \tag{4.17}$$

We can now present and prove Proposition 2 about the solutions of the equations in (4.14).

Proposition 2. *If an image I is moment-normalizable, the optimization problem stated in (4.14) has exactly 8 distinct solutions.*

Proof. From the formulas of $\eta'_{2,0}$, $\eta'_{0,2}$, and $\eta'_{1,1}$ in (4.11), and using the three constraints $\eta'_{2,0} = c^2$, $\eta'_{0,2} = c^2$, and $\eta'_{1,1} = 0$, we obtain:

$$\begin{cases} t_2^2\ \eta_{0,2} + 2t_1\ t_2\ \eta_{1,1} + t_1^2\ \eta_{2,0} = c^2, \\ t_5^2\ \eta_{0,2} + 2t_4\ t_5\ \eta_{1,1} + t_4^2\ \eta_{2,0} = c^2, \\ t_2\ t_5\ \eta_{0,2} + (t_2\ t_4 + t_1\ t_5)\ \eta_{1,1} + t_1\ t_4\ \eta_{2,0} = 0. \end{cases} \tag{4.18}$$

With $D > 0$ (note that D is always non-negative), the constraints in (4.18) lead to two parameterized solutions:

$$\begin{cases} t_1 = \frac{c}{D\sqrt{\eta_{2,0}}}(-\eta_{1,1}\sin\theta + D\cos\theta), \\ t_2 = \frac{c\sqrt{\eta_{2,0}}}{D}\sin\theta, \\ t_4 = \frac{c}{D\sqrt{\eta_{2,0}}}(-\eta_{1,1}\cos\theta - D\sin\theta), \\ t_5 = \frac{c\sqrt{\eta_{2,0}}}{D}\cos\theta, \end{cases} \tag{4.19}$$

or

$$\begin{cases} t_1 = \frac{c}{D\sqrt{\eta_{2,0}}}(-\eta_{1,1}\sin\theta - D\cos\theta), \\ t_2 = \frac{c\sqrt{\eta_{2,0}}}{D}\sin\theta, \\ t_4 = \frac{c}{D\sqrt{\eta_{2,0}}}(-\eta_{1,1}\cos\theta + D\sin\theta), \\ t_5 = \frac{c\sqrt{\eta_{2,0}}}{D}\cos\theta, \end{cases} \tag{4.20}$$

where θ is an arbitrary angle in $[0, 2\pi)$.

For parameterized solution 1 in (4.19), the term $\eta'_{2,2}$ in (4.14) can be expressed as

$$\eta'_{2,2} = f_1(\theta) = \frac{c^4}{8D^4\eta_{2,0}^2}(A\cos 4\theta + B\sin 4\theta + C), \tag{4.21}$$

where A is given in (4.16), B is given in (4.17), and

$$\begin{aligned} C = {} & 4\eta_{2,2}\eta_{2,0}^2\eta_{1,1}^2 + 2\eta_{0,2}\eta_{2,2}\eta_{2,0}^3 + \eta_{0,4}\eta_{2,0}^4 \\ & + \eta_{4,0}\eta_{0,2}^2\eta_{2,0}^2 - 4\eta_{1,1}\eta_{1,3}\eta_{2,0}^3 - 4\eta_{0,2}\eta_{1,1}\eta_{3,1}\eta_{2,0}^2. \end{aligned} \tag{4.22}$$

Let Φ be an angle in $[0, 2\pi)$ so that $\cos\Phi = \frac{A}{\sqrt{A^2+B^2}}$ and $\sin\Phi = \frac{B}{\sqrt{A^2+B^2}}$. The constraint $\eta'_{2,2}$ becomes

$$\eta'_{2,2} = f_1(\theta) = \frac{c^4}{8D^4\eta_{2,0}^2}\,(\sqrt{A^2+B^2}\cos(4\theta - \Phi) + C). \tag{4.23}$$

Clearly, in interval $[0, 2\pi)$, function $f_1(\theta)$ has four maximum points at

$$\hat{\theta} = \frac{\Phi}{4} + \frac{k\pi}{2}, \text{where } k = 0, 1, 2, 3. \tag{4.24}$$

For parameterized solution 2 in (4.20), the term $\eta'_{2,2}$ in (4.14) can be expressed as

$$\begin{aligned} \eta'_{2,2} = f_2(\theta) &= \frac{c^4}{8D^4\eta_{2,0}^2}(A\cos 4\theta - B\sin 4\theta + C), \\ &= \frac{c^4}{8D^4\eta_{2,0}^2}\,(\sqrt{A^2+B^2}\,\cos(4\theta + \Phi) + C), \end{aligned} \tag{4.25}$$

where A, B, and C are the same as in (4.16), (4.17), and (4.22). Then, the function $f_2(\theta)$ has four maximum points at

$$\hat{\theta} = -\frac{\Phi}{4} + \frac{k\pi}{2}, \text{where } k = 1, 2, 3, 4. \tag{4.26}$$

Figure 4.1 illustrates the functions $f_1(\theta)$ and $f_2(\theta)$ for an example input image for θ in the range from 0 to 2π.

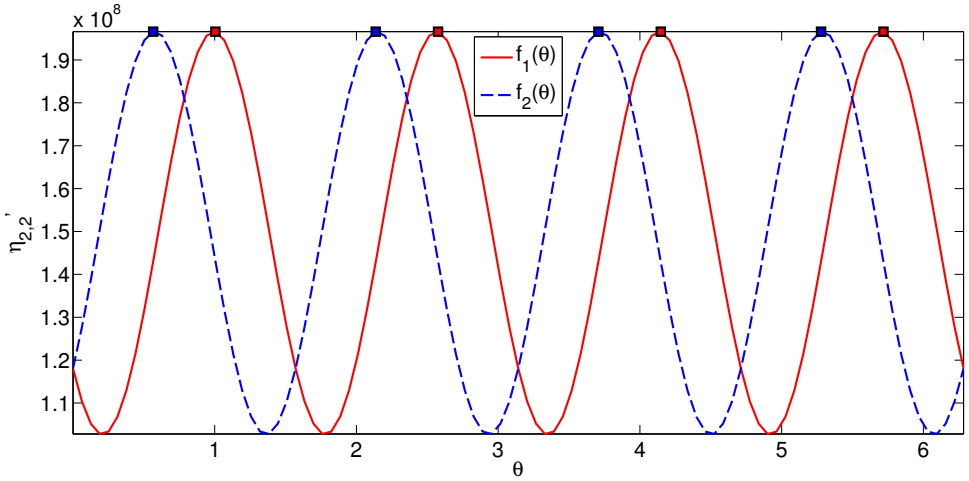


Figure 4.1: The functions $f_1(\theta)$ and $f_2(\theta)$ for an example input image for θ in the range from 0 to 2π. The locations of the 8 maximum points are also shown.

Once parameters t_1, t_2, t_4, and t_5 are found, parameters t_3 and t_6 can be determined as

$$\begin{cases} t_3 &= -t_1\nu_{1,0} - t_2\nu_{0,1}, \\ t_6 &= -t_4\nu_{1,0} - t_5\nu_{0,1}. \end{cases} \tag{4.27}$$

Because $A \neq 0$ and $B \neq 0$, the 8 angles $\hat{\theta}$ computed in (4.24) and (4.26) are distinct. Hence, the constrained optimization problem in (4.14) has exactly 8 distinct solutions. □

4.2.4 Affine-normalization algorithm

Based on the derivations in Subsection 4.2.3, we can now describe the steps of the proposed affine-normalization (see Table 4.2). The uniqueness of the set of normalized images is stated in Proposition 3. The proof of this proposition is given in Appendix 8.2.

Next, we explore the effects of the proposed algorithm on sample images. Figure 4.2 shows four examples. In each example, the input image is shown on

Table 4.2: Proposed affine-normalization algorithm.

Input: Image $I(x,y)$
Output: A set $\mathcal{I}$ of 8 normalized images $\hat{I}(x,y)$
Steps:

1. Compute image moments: $\nu_{1,0}$, $\nu_{0,1}$, $\eta_{1,1}$, $\eta_{2,0}$, $\eta_{0,2}$, $\eta_{2,2}$, $\eta_{3,1}$, $\eta_{1,3}$, $\eta_{4,0}$, and $\eta_{0,4}$.
2. Compute parameters A using (4.16), and B using (4.17).
3. Compute an angle Φ in $[0,2\pi)$ so that $\cos\Phi = \frac{A}{\sqrt{A^2+B^2}}$ and $\sin\Phi = \frac{B}{\sqrt{A^2+B^2}}$.
4. Construct 4 matrices $\hat{T}$ with t_1, t_2, t_4, and t_5 given in (4.19), where $\hat{\theta} = \frac{\Phi}{4} + \frac{k\pi}{2}$ and $k = 0,1,2,3$. Parameters t_3 and t_6 are computed as in (4.27).
5. Construct 4 other matrices $\hat{T}$ with t_1, t_2, t_4, and t_5 given in (4.20), where $\hat{\theta} = -\frac{\Phi}{4} + \frac{k\pi}{2}$ and $k = 1,2,3,4$. Parameters t_3 and t_6 are computed as (4.27).
6. Generate a set $\mathcal{I}$ of 8 normalized images $\hat{I}$ by applying the affine transformation matrices $\hat{T}$ on input image $I(x,y)$.

Proposition 3. *Let $I(x,y)$ be a moment-normalizable image, and $I_{\mathrm{d}}(x,y)$ be a distorted image that is obtained by applying an arbitrary affine transformation matrix T_{d} on I. The affine-normalization algorithm shown in Table 4.2 will produce the same set of normalized output images for image I and distorted image I_{d}.*

the left, whereas the 8 normalized output images are shown on the right.

In the first example (Fig. 4.2(a)), the input image is an original (non-distorted) image. The first 4 normalized images are generated from solution 1 in (4.19). These images are equivalent via a rotation of $k\times 90^{\mathrm{o}}$, where $k = 0,1,2,3$. The other 4 normalized images are generated from solution 2 in (4.20). These images are equivalent to the first 4 normalized images from solution 1 via horizontal flipping (this is because $f_1(\theta) = f_2(-\theta)$).

In the second example (Fig. 4.2(b)), the input image is an affine-distorted image of the original image in Fig. 4.2(a). As can be seen, the proposed algorithm

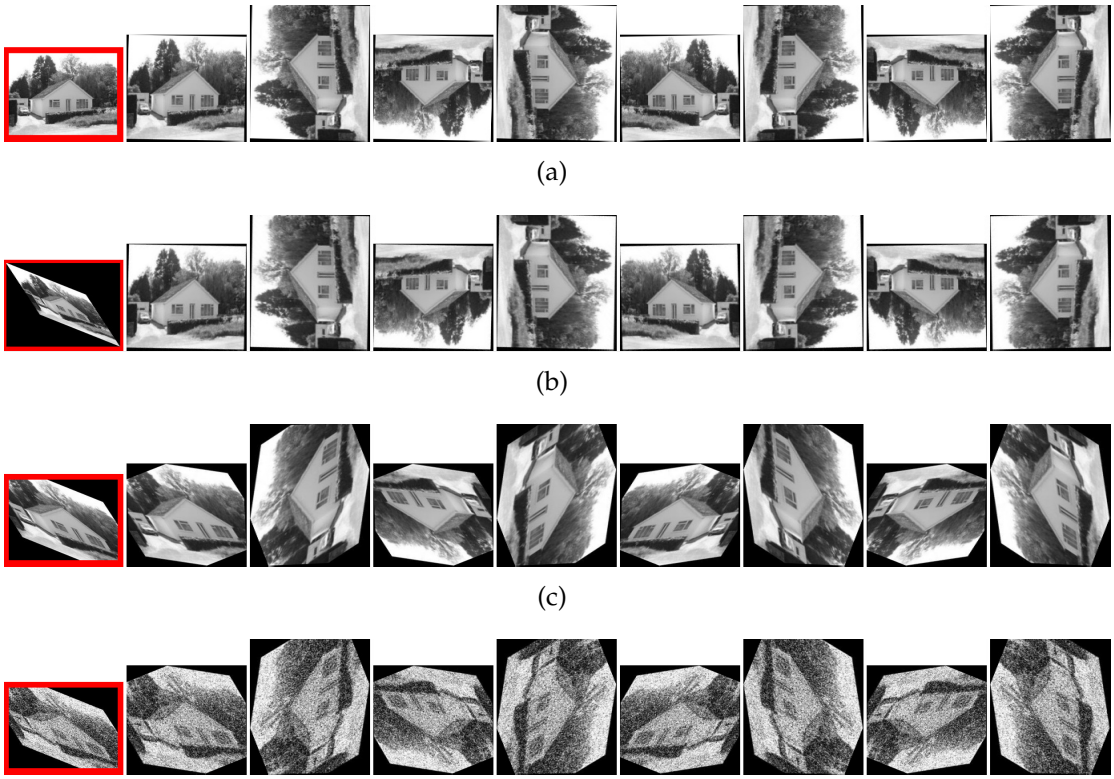


Figure 4.2: Examples of the proposed affine normalization. Column 1 is an input image, whereas Columns 2 to 9 are the 8 normalized images. The input image is: (a) an original non-distorted image, (b) an affine-distorted image, (c) an affine-distorted image with image cropping (40% of the image is removed), (d) an affine-distorted image with image cropping and noise (noise density = 0.1).

produces the same set of normalized output images for the inputs in Fig. 4.2(a) or 4.2(b). This result is consistent with Proposition 3.

In the third example (Fig. 4.2(c)), the input image is distorted by both affine transformation and image cropping. The proposed algorithm produces a set of normalized images that are similar to the normalized images in Fig. 4.2(a) and 4.2(b).

In the fourth example (Fig. 4.2(d)), the input image contains affine distortions, image cropping, and random noise. The proposed algorithm again produces a set of normalized images that are similar to those for the original image in Fig. 4.2(a). The results in Fig. 4.2 indicate that the proposed affine normalization

algorithm can handle affine distortion, image cropping, and image noise. Note that the complexity of the proposed affine normalization algorithm is $O(N)$ where N is the number of image pixels.

Figure 4.3 also illustrates some examples of the normalized images generated by the proposed method. In each row, the first input image is the original image. Then, the input image is distorted by scaling, shearing, rotation, and combined affine transformations, separately. Their normalized outputs are shown next to each input image. From the first row to the second row of Fig. 4.3, the original objects are two handwritten digits (3 and 5) in the MNIST data set [51]. The normalized images for the same digit are similar. From the third row to the fourth row of Fig. 4.3, the original objects are two types of cars in the COIL-100 data set [203]. Under different affine distortions (scaling, shearing, rotation, and affine), the normalized outputs are similar for different types of cars. From the fifth row to the sixth row of Fig. 4.3, the original objects are faces of two subjects from the ORL data set [204]. The normalized outputs of distorted faces are normalized to the similar form. From the seventh row to the eighth row of Fig. 4.3, the two scene images (*house*) in the SUN397 data set [4] are normalized under several affine distortions. The outputs also have similar forms for different images.

Another experiment is performed to further illustrate the effects of the proposed algorithm on affine-distorted images. We construct a scatter plot, where the x-axis is normalized central moment $\eta_{p,q}$, and the y-axis is normalized central moment $\eta_{q,p}$. The plot based on an example original image is shown in Fig. 4.4 for $p = 3$ and $q = 2$. The moments of 50 affine-distorted images (square markers) are scattered on the moment plane. The moments of 8 normalized images (triangle

Figure 4.3: Examples of the proposed normalization on different affine distortions. All input images are highlighted by the red border. The input image in (b) is distorted by scaling parameters $s_x = 3$ and $s_y = 1.5$; The input in (c) is distorted by shearing parameters $h_x = -0.5$ and $h_y = 1.5$; The input in (d) is distorted by the rotation parameter $\theta = 120°$; The input in (e) is distorted by the combining parameters from (b) to (d). The normalized images that have the highest correlation score with the original image in (a) are shown next to each input image.

markers) are clustered near the moments of the original image (circle marker).

This experiment shows that the proposed normalization reduces the variations

caused by affine distortions.

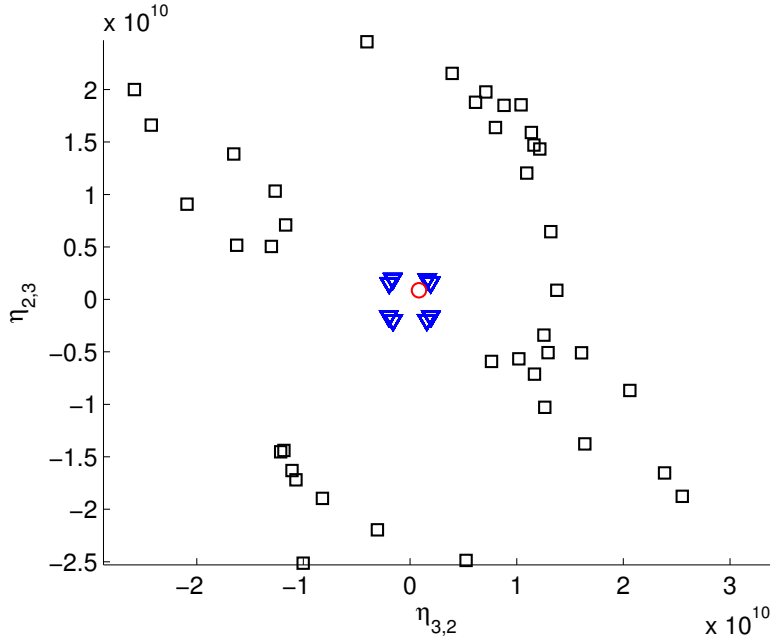


Figure 4.4: The scatter plot of normalized central moments for an original image (red circle ○), affine-distorted images (black square □), and normalized images (blue triangle ▽).

4.2.5 Relationship between moment $\eta'_{2,2}$ and principal axis

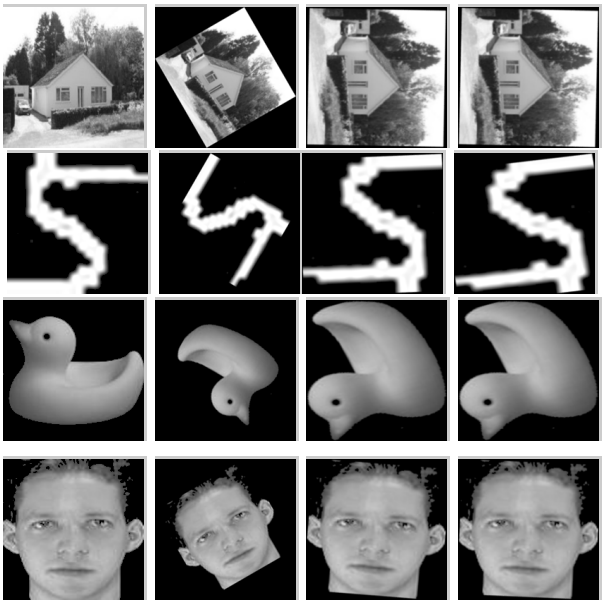
This section discusses the relationship between moment $\eta'_{2,2}$ and principal axis. According to (4.13) in the proposed method, the invariant angle for the first normalized image is

$$\theta = \frac{1}{4}\arctan\frac{B}{A}. \tag{4.28}$$

According to [205], for a 2-D image, the angle between the principal axis and the x-axis is

$$\alpha = \frac{1}{2}\arctan\frac{2\mu_{1,1}}{\mu_{0,2}-\mu_{2,0}}. \tag{4.29}$$

The difference between angle θ and α is a fixed value depending on the moments of input image.



(a) (b) (c) (d)

Figure 4.5: Examples of the normalized images using the moment $\eta'_{2,2}$ and the principal axis: (a) original image, (b) input image with rotation, (c) normalized image for (b) using the moment $\eta'_{2,2}$, and (d) normalized image for (b) using the principal axis. Only the first normalized image is shown for each example. The orientation of the first normalized image depends on the orientation of the input image.

To compare the angle α with the proposed invariant angle, the angles $\hat{\theta}$ in (4.24) and (4.26) are replaced by the angle α as follows. For the first parameterized solution in (4.24),

$$\hat{\theta} = \alpha + \frac{k\pi}{2}, \text{where } k = 0,1,2,3. \tag{4.30}$$

For the second parameterized solution in (4.26),

$$\hat{\theta} = -\alpha - \frac{k\pi}{2}, \text{where } k = 0,1,2,3. \tag{4.31}$$

Figure 4.5 shows some examples of image normalization using the moment $\eta'_{2,2}$ and the principal axis, respectively. The input images are only distorted by rotations. The normalized images using angles calculated by the moment $\eta'_{2,2}$ have slightly difference with the normalized images using angles calculated by the principal axis. In the proposed method, the moment $\eta'_{2,2}$ is used to determine

the invariant angle for rotation normalization. In the principal axis method, the angle between principal axis of the image and x-axis is invariant to rotation.

4.2.6 Sorting the normalized images

In the proposed normalization method, 8 output images form a set of normalized images. As shown in Fig. 4.5, the orientation of the first normalized image depends on the orientation of the input image. Once the first normalized image is determined, the other 7 normalized images are also calculated. The first 4 normalized images are equivalent via a rotation of 90°. The other 4 normalized images are equivalent to the first 4 normalized images via horizontal flipping (see Fig. 4.2). For an input image with different rotations, the order of the 8 normalized images is different. In this section, we provide a scheme to eliminate the ambiguity of orientation and sort the 8 normalized images.

As show in (4.11), the value of moment $\eta'_{1,2}$ and $\eta'_{2,1}$ depends on the values of t_1, t_2, t_4, t_5, and the moments of input image. The moment $\eta'_{1,2}$ for the first solution in (4.19) is

$$U = \frac{c^3}{4D^3\sqrt{\eta_{2,0}^3}}(E\sin\hat{\theta} + F\cos\hat{\theta} + G\sin 3\hat{\theta} + H\cos 3\hat{\theta}). \tag{4.32}$$

The moment $\eta'_{2,1}$ for the first solution in (4.19) is

$$V = \frac{c^3}{4D^3\sqrt{\eta_{2,0}^3}}(E\cos\hat{\theta} - F\sin\hat{\theta} - G\cos 3\hat{\theta} + H\sin 3\hat{\theta}), \tag{4.33}$$

Table 4.3: Moment $\eta'_{1,2}$ and $\eta'_{2,1}$ for the 8 normalized images without sorting.

$\hat{\theta}$	Matrix T	*Moment* $\eta'_{1,2}$	*Moment* $\eta'_{2,1}$	$\hat{\theta}$	Matrix T	*Moment* $\eta'_{1,2}$	*Moment* $\eta'_{2,1}$
$\frac{\Phi}{4}$	$\hat{T}=\begin{pmatrix}\hat{t}_1 & \hat{t}_2 & \hat{t}_3\\ \hat{t}_4 & \hat{t}_5 & \hat{t}_6\\ 0 & 0 & 1\end{pmatrix}$	U	V	$-\frac{\Phi}{4}+2\pi$	$\hat{T}=\begin{pmatrix}-\hat{t}_1 & -\hat{t}_2 & -\hat{t}_3\\ \hat{t}_4 & \hat{t}_5 & \hat{t}_6\\ 0 & 0 & 1\end{pmatrix}$	$-U$	V
$\frac{\Phi}{4}+\frac{\pi}{2}$	$\hat{T}=\begin{pmatrix}\hat{t}_4 & \hat{t}_5 & \hat{t}_6\\ -\hat{t}_1 & -\hat{t}_2 & -\hat{t}_3\\ 0 & 0 & 1\end{pmatrix}$	V	$-U$	$-\frac{\Phi}{4}+\frac{3\pi}{2}$	$\hat{T}=\begin{pmatrix}-\hat{t}_4 & -\hat{t}_5 & -\hat{t}_6\\ -\hat{t}_1 & -\hat{t}_2 & -\hat{t}_3\\ 0 & 0 & 1\end{pmatrix}$	$-V$	$-U$
$\frac{\Phi}{4}+\pi$	$\hat{T}=\begin{pmatrix}-\hat{t}_1 & -\hat{t}_2 & -\hat{t}_3\\ -\hat{t}_4 & -\hat{t}_5 & -\hat{t}_6\\ 0 & 0 & 1\end{pmatrix}$	$-U$	$-V$	$-\frac{\Phi}{4}+\pi$	$\hat{T}=\begin{pmatrix}\hat{t}_1 & \hat{t}_2 & \hat{t}_3\\ -\hat{t}_4 & -\hat{t}_5 & -\hat{t}_6\\ 0 & 0 & 1\end{pmatrix}$	U	$-V$
$\frac{\Phi}{4}+\frac{3\pi}{2}$	$\hat{T}=\begin{pmatrix}-\hat{t}_4 & -\hat{t}_5 & -\hat{t}_6\\ \hat{t}_1 & \hat{t}_2 & \hat{t}_3\\ 0 & 0 & 1\end{pmatrix}$	$-V$	U	$-\frac{\Phi}{4}+\frac{\pi}{2}$	$\hat{T}=\begin{pmatrix}\hat{t}_4 & \hat{t}_5 & \hat{t}_6\\ \hat{t}_1 & \hat{t}_2 & \hat{t}_3\\ 0 & 0 & 1\end{pmatrix}$	V	U

where

$$
\begin{aligned}
E =& \eta_{0,3}\eta_{2,0}^3+\eta_{0,2}\eta_{2,0}^2\eta_{2,1}-\eta_{1,1}\eta_{1,2}\eta_{2,0}^2+2\eta_{1,1}^2\eta_{2,0}\eta_{2,1}-\eta_{0,2}\eta_{1,1}\eta_{2,0}\eta_{3,0},\\
F =& D(\eta_{1,2}\eta_{2,0}^2+\eta_{0,2}\eta_{2,0}\eta_{3,0}-2\eta_{1,1}\eta_{2,0}\eta_{2,1}),\\
G =& \eta_{0,3}\eta_{2,0}^3-4\eta_{1,1}^3\eta_{3,0}-3\eta_{0,2}\eta_{2,0}^2\eta_{2,1}-3\eta_{1,1}\eta_{1,2}\eta_{2,0}^2+6\eta_{1,1}^2\eta_{2,0}\eta_{2,1}+3\eta_{0,2}\eta_{1,1}\eta_{2,0}\eta_{3,0},
\end{aligned}
$$

and

$$
H = D(3\eta_{1,2}\eta_{2,0}^2+4\eta_{1,1}^2\eta_{3,0}-\eta_{0,2}\eta_{2,0}\eta_{3,0}-6\eta_{1,1}\eta_{2,0}\eta_{2,1}). \tag{4.34}
$$

Among all 8 output images, we first select images that have the maximum $\eta'_{2,1}$. Next, among the two short-listed images, we select the image that has the maximum $\eta'_{1,2}$ to be the first normalized image in the sorted outputs. Then, the second to the fourth images are equivalent to the first image via a rotation of 90°. The other 4 normalized images are equivalent to the first 4 normalized images via horizontal flipping.

Table 4.3 shows the patterns of $\eta'_{1,2}$ and $\eta'_{2,1}$ for 8 output images. Based on this table, we can show that there is a unique image selected according to the above scheme, provided that $U \neq V \neq 0$.

Figure 4.6 shows two examples of the sorted images using moments $\eta'_{2,1}$ and $\eta'_{1,2}$. The first normalized image has the highest values of $\eta'_{2,1}$ and $\eta'_{1,2}$. Under

(a)

(b)

Figure 4.6: Examples of normalized images sorted by image moments $\eta'_{2,1}$ and $\eta'_{1,2}$

different affine distortions, the output images are not only the same set of images, but also in the same order.

4.2.7 Relationship between moment-based normalization algorithms

In order to compare the proposed normalization method with existing methods, the representative normalization algorithms using image moments are described in this section.

Existing image normalization methods transform the original image and its distorted versions into a canonical form so that the image moments of the canon-

ical form are independent of affine deformations. The moment constraints play a key role in the existing methods. They affect the uniqueness of the normalization matrix and the simplicity of computation.

To avoid solving complex systems of non-linear equations, existing algorithms decomposed the normalization matrix T into several simplified transformation matrices. However, the proposed method computes the normalization matrix T by solving an optimization problem in one step. Table 4.4 compares five moment-based normalization methods with the proposed normalization method.

In the existing algorithms, the centroid of input image has been translated to the origin of the coordinate system. Therefore, t_3 and t_6 for translation are not considered in the five existing methods. Reiss decomposed the affine matrix T into an x-shearing matrix, a scaling matrix, and a rotation matrix (XSR) [202]:

$$\begin{pmatrix} t_1 & t_2 \\ t_4 & t_5 \end{pmatrix} = \begin{pmatrix} 1 & h_x \\ 0 & 1 \end{pmatrix} \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}. \tag{4.35}$$

A condition that $t_1 t_5 - t_2 t_4 \neq 0$ is required to ensure the uniqueness of this decomposition. Based on the XSR decomposition, moment constraints

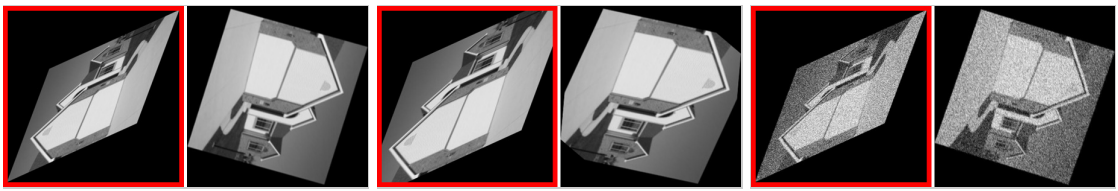
$$\mu'_{1,1} = 0,\ \mu''_{2,0} = 1,\ \mu''_{0,2} = 1,\ \mu'''_{3,0} + \mu'''_{1,2} = 0, \tag{4.36}$$

are applied sequentially on an input image as an x-shearing normalization ($\mu'_{1,1} = 0$), a scaling normalization ($\mu''_{2,0} = 1$ and $\mu''_{0,2} = 1$), and a rotation normalization ($\mu'''_{3,0} + \mu'''_{1,2} = 0$). The output images are invariant to affine distortions after the three

Table 4.4: Comparison between moment-based affine normalization methods.

Algorithm	*Moment constraints*	*Number of decomposed transformations*	*Sum of moment orders*
XSR-Reiss [202]	$\mu'_{1,1} = 0,\ \mu''_{2,0} = 1,\ \mu''_{0,2} = 1,\ \mu'''_{3,0} + \mu'''_{1,2} = 0.$	3	12
XYS-Rothe [190]	$\mu'_{3,0} = 0,\ \mu''_{1,1} = 0,\ \mu'''_{2,0} = 1,\ \mu'''_{0,2} = 1.$	3	9
XYS-Zhang [206]	$\mu'_{3,0} = 0,\ \mu''_{0,3} = 0,\ \mu'''_{2,1} = 1,\ \mu'''_{1,2} = 1.$	3	12
XYS-Dong [194]	$\mu'_{3,0} = 0,\ \mu''_{1,1} = 0,\ \mu'''_{5,0} > 0,\ \mu'''_{0,5} > 0.$	3	15
RSR-Pei [199]	$\mu_{2,0}, \mu_{1,1}, \mu_{0,2}, \mu_{1,2}, \mu_{3,0}, \mu_{0,3}, \mu_{2,1}$	3	18
Proposed method	$\max\{\eta'_{2,2}\}$ s.t. $\nu'_{1,0} = 0,\ \nu'_{0,1} = 0,\ \eta'_{2,0} = c^2,\ \eta'_{0,2} = c^2,\ \eta'_{1,1} = 0.$	0	12

sequential transformations. In the experimental section, this method is named as XSR-Reiss normalization. Figure 4.7 shows example results of XSR-Reiss. The output images are consistent for the same input image under different affine-distortions. However, the normalized images have angle differences with the original non-distorted image. The black border generated by the normalization will increase the difficulty to classify images.



(a) (b) (c)

Figure 4.7: Examples of normalized images for XSR-Reiss normalization. Input images are highlighted by the red border: (a) affine-distorted image, (b) affine distorted and cropped image, and (c) affine distorted and noisy image.

Note that moment $\mu'_{p,q}$ is calculated from the output image of the first transformation; moment $\mu''_{p,q}$ is calculated from the output image of the second transformation, and moment $\mu'''_{p,q}$ is calculated from the output image of the third transformation.

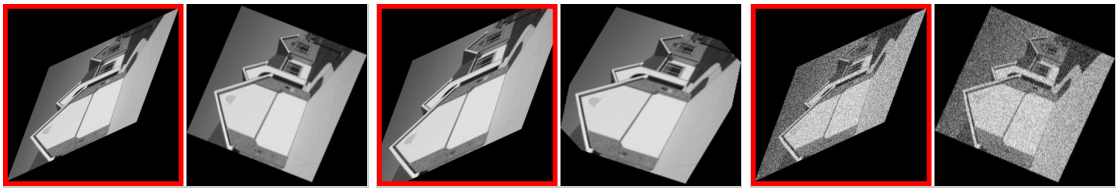
Rothe *et al.* decomposed the affine matrix T into an x-shearing matrix, a y-shearing matrix, and a scaling matrix (XYS) [190]:

$$\begin{pmatrix} t_1 & t_2 \\ t_4 & t_5 \end{pmatrix} = \begin{pmatrix} 1 & h_x \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ h_y & 1 \end{pmatrix} \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix}. \tag{4.37}$$

Based on the XYS decomposition, different constraints were used to normalize input images.

Rothe *et al.* used the moment constraints as follows [190]:

$$\mu'_{3,0} = 0,\ \mu''_{1,1} = 0,\ \mu'''_{2,0} = 1,\ \mu'''_{0,2} = 1. \tag{4.38}$$



(a) (b) (c)

Figure 4.8: Examples of normalized images for XYS-Rothe normalization. Input images are highlighted by the red border: (a) affine-distorted image, (b) affine distorted and cropped image, and (c) affine distorted and noisy image.

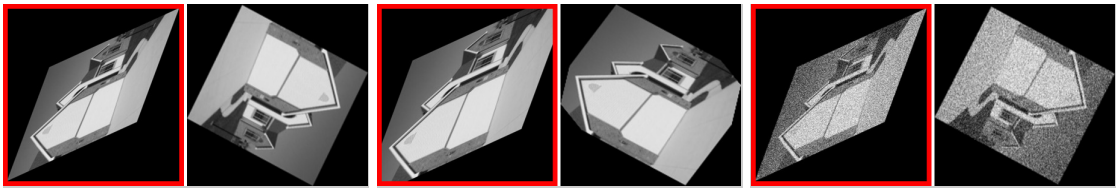
These constraints are applied sequentially as an x-shearing normalization ($\mu'_{3,0} = 0$), a y-shearing normalization ($\mu''_{1,1} = 0$), and a scaling normalization ($\mu'''_{2,0} = 1$ and $\mu'''_{0,2} = 1$). In the experimental section, this method is named as XYS-Rothe normalization. Figure 4.8 shows example results of XYS-Rothe. Same as XSR-Reiss, the normalized images have angle differences with the original non-distorted image.

Zhang *et al.* also decomposed the affine matrix T into an x-shearing matrix, a y-shearing matrix, and a scaling matrix (XYS). They proposed an affine normalization method using the following moment constraints [206]:

$$\mu'_{3,0} = 0,\ \mu''_{0,3} = 0,\ \mu'''_{2,1} = 1,\ \mu'''_{1,2} = 1. \tag{4.39}$$

The four constraints were applied sequentially as an x-shearing normalization ($\mu'_{3,0} = 0$), a y-shearing normalization ($\mu''_{0,3} = 0$), and a scale normalization ($\mu'''_{2,1} = 1$ and $\mu'''_{1,2} = 1$). In the experimental section, this method is named as XYS-Zhang normalization. Figure 4.9 shows example results of XYS-Zhang. The output image under affine-distortion (Fig. 4.9(a)) is different with the output image under affine-distortion and cropping (Fig. 4.9(b)). The XYS-Zhang normalization is not stable for image cropping.

Dong *et al.* decomposed the affine matrix T into an x-shearing matrix, a y-shearing matrix, and a scaling matrix (XYS). They proposed an affine normal-



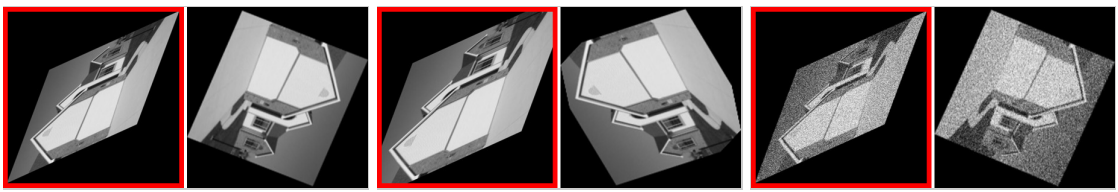
(a) (b) (c)

Figure 4.9: Examples of normalized images for XYS-Zhang normalization. Input images are highlighted by the red border: (a) affine-distorted image, (b) affine distorted and cropped image, and (c) affine distorted and noisy image.

ization method using the following moment constraints [194]:

$$\mu'_{3,0} = 0,\ \mu''_{1,1} = 0,\ \mu'''_{5,0} > 0,\ \mu'''_{0,5} > 0. \tag{4.40}$$

First, the moment $\mu'_{3,0}$ is set to 0 for an x-shearing normalization. Next, the $\mu''_{1,1}$ is set to 0 for a y-shearing normalization. At last, the image is transformed to a standard size for a scaling normalization. The signs of parameters for scaling are determined so that the $\mu'''_{5,0}$ and $\mu'''_{0,5}$ are positive. In the experimental section, this method is named as XYS-Dong normalization. Figure 4.10 shows example results of XYS-Dong. The output image under affine-distortion (Fig. 4.10(a)) is different with the output image under affine-distortion and cropping (Fig. 4.10(b)). The orientation of the normalized image is uncertain for the XYS-Dong normalization, especially when images are cropped.



(a) (b) (c)

Figure 4.10: Examples of normalized images for XYS-Dong normalization. Input images are highlighted by the red border: (a) affine-distorted image, (b) affine distorted and cropped image, and (c) affine distorted and noisy image.

Pei and Lin decomposed the affine matrix T into a rotation matrix (R_1), a

scaling matrix (S), and another rotation matrix (R_2) [199]:

$$\begin{pmatrix} t_1 & t_2 \\ t_4 & t_5 \end{pmatrix} = R_1 \, S \, R_2 \tag{4.41}$$

The first rotation matrix R_1 and the scaling matrix S are computed based on the covariance-matrix M:

$$M = \begin{pmatrix} \mu_{2,0} & \mu_{1,1} \\ \mu_{1,1} & \mu_{0,2} \end{pmatrix}. \tag{4.42}$$

The scaling matrix S is computed from the eigenvalues of M:

$$S = \begin{pmatrix} \frac{c}{\sqrt{\lambda_1}} & 0 \\ 0 & \frac{c}{\sqrt{\lambda_2}} \end{pmatrix}, \tag{4.43}$$

where (λ_1, λ_2) are eigenvalues of M and $c^2 = \sqrt{\lambda_1}\sqrt{\lambda_2}$. The rotation matrix R_1 is computed from the eigenvectors of M:

$$R_1 = \begin{pmatrix} e_{1,1} & e_{1,2} \\ e_{2,1} & e_{2,2} \end{pmatrix}, \tag{4.44}$$

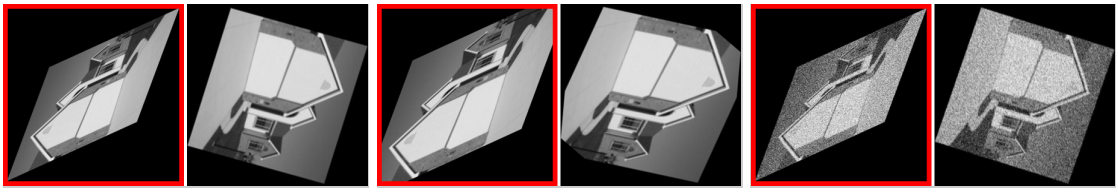
where $(e_{1,1}, e_{1,2})$ is the eigenvector corresponding to λ_1 and $(e_{2,1}, e_{2,2})$ is the eigenvector corresponding to λ_2.

Since the matrix M is real and symmetric, both eigenvectors are orthonormal to each other. Hence $e_{1,1} = e_{2,2}$ and $e_{1,2} = -e_{2,1}$. With the matrix R_1, the image becomes uncorrelated to the transformed coordinate system. With the matrix S, the image is rescaled according to the eigenvalues of M.

To make output image invariant to rotation, an angle β of matrix $R_2 = \begin{pmatrix} \cos\beta & \sin\beta \\ -\sin\beta & \cos\beta \end{pmatrix}$ is determined from

$$\tan\beta = -\frac{\mu''_{1,2} + \mu''_{3,0}}{\mu''_{0,3} + \mu''_{2,1}}. \tag{4.45}$$

If $-(\mu''_{1,2} + \mu''_{3,0})\sin\beta + (\mu''_{0,3} + \mu''_{2,1})\cos\beta < 0$, then $\beta = \beta + \pi$. In the experimental section, this method is named as RSR-Pei normalization. Figure 4.11 shows



(a) (b) (c)

Figure 4.11: Examples of normalized images for RSR-Pei normalization. Input images are highlighted by the red border: (a) affine-distorted image, (b) affine distorted and cropped image, and (c) affine distorted and noisy image.

example results of RSR-Pei. The normalized images have angle differences with the original non-distorted image.

We highlight two improvements of the proposed image normalization algorithm compared to the existing methods. First, the proposed algorithm estimates the affine transformation parameters directly from the moments of the input image, without decomposing the transformation matrix into a series of simplified matrices (see Table 4.4). This strategy improves the efficiency of the normalization and avoids re-sampling errors. Second, a complete set of 8 normalized images are generated, which avoids the image-reflection ambiguity and orientation uncertainty in the existing methods.

4.3 Experimental evaluation and results

In this section, the proposed image normalization method is compared with five existing moment-based normalization methods. The effects of image normalization on class separability were also evaluated. Section 4.3.1 describes the data sets used in the experiments and Section 4.3.2 defines the performance measures for image normalization. Section 4.3.3 compares the proposed image normalization with five existing affine normalization algorithms. Section 4.3.4 analyzes effects

of the proposed normalization on class-separability.

4.3.1 Image data sets

Evaluation was conducted using four public data sets: the SUN397 data set of scenes [4], the MNIST data set of handwritten digits [51], the COIL-100 data set of objects [203], and the ORL data set of faces [204].

The SUN397 data set has 397 scene categories, from abbey, bedroom, and castle to highway, theater, and yard. There are 108,754 images in total and at least 100 images in each category. Images in the SUN397 data set have various object arrangement and complex background. The MNIST data set contains 10 types of digits (0,1,...,9) written by 500 different writers. There are 70,000 digits in total and at least 6,000 digits per class. The COIL-100 data set contains 7,200 images of 100 objects. There are 72 images per object captured from different viewing angles. The ORL data set consists of 400 face images in total. There are 10 images per subject captured from different viewing angles. All patterns in the ORL and COIL-100 data sets have arbitrary shape and diverse intensity.

The affine-distorted images for the four data sets were formed by applying random affine transformations on the original images. The translation parameters t_x and t_y were between -100 to 100. The scaling parameters s_x and s_y were non-zero values between -2 to 2. The shearing parameters h_x and h_y were between -4 to 4. The rotation parameter θ was between 0 to 2π. The distorted images contained at least 30% non-zero pixels of the original images. The image normalization was then applied on these original and distorted images to obtain the normalized images.

4.3.2 Performance measures for image normalization

The normalization accuracy was measured using the peak signal-to-noise ratio (*PSNR*). *PSNR* between the normalized image $\hat{I}$ of the original image I and the normalized image $\hat{I}_d$ of the distorted image I_d is computed as

$$PSNR = 10 \times log_{10}(\frac{255^2}{MSE}). \tag{4.46}$$

The mean square error (*MSE*) is defined as

$$MSE = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} [\hat{I}(i,j) - \hat{I}_d(i,j)]^2, \tag{4.47}$$

where the size of the normalized images $\hat{I}$ and $\hat{I}_d$ is $M \times N$. A larger *PSNR* value implies a more precise normalization.

For each input image, the proposed normalization algorithm produces 8 outputs, which are equivalent via $k \times 90^\circ$ rotation or horizontal flipping. Among the 8 images, the one having the highest correlation score with the original image I is used to compute the *PSNR* value. The analysis of normalization accuracy using *PSNR* will be described in Subsection 4.3.3.

The effects of image normalization on class separability were evaluated using the correlation coefficients within the same class (ρ_w) and correlation coefficients between different classes (ρ_b). The similarity between two images I_1 and I_2 was calculated by the correlation coefficients:

$$\rho(I_1, I_2) = \frac{cov(I_1, I_2)}{\sigma(I_1)\sigma(I_2)}. \tag{4.48}$$

Here, *cov*() is the covariance function and $\sigma()$ is the standard deviation function.

For multi-class data sets, the class separability was measured by the intra-clas correlation coefficients (ρ_w) and the inter-class correlation coefficients (ρ_b). The

intra-class similarity for the image I_i in class n is calculated as

$$\rho_w(I_i) = \max \rho(I_i, I_j), \tag{4.49}$$

where $I_j \in$ class n, and $i \neq j$.

The inter-class similarity for image I_i in class n is calculated as

$$\rho_b(I_i) = \max \rho(I_i, I_j), \tag{4.50}$$

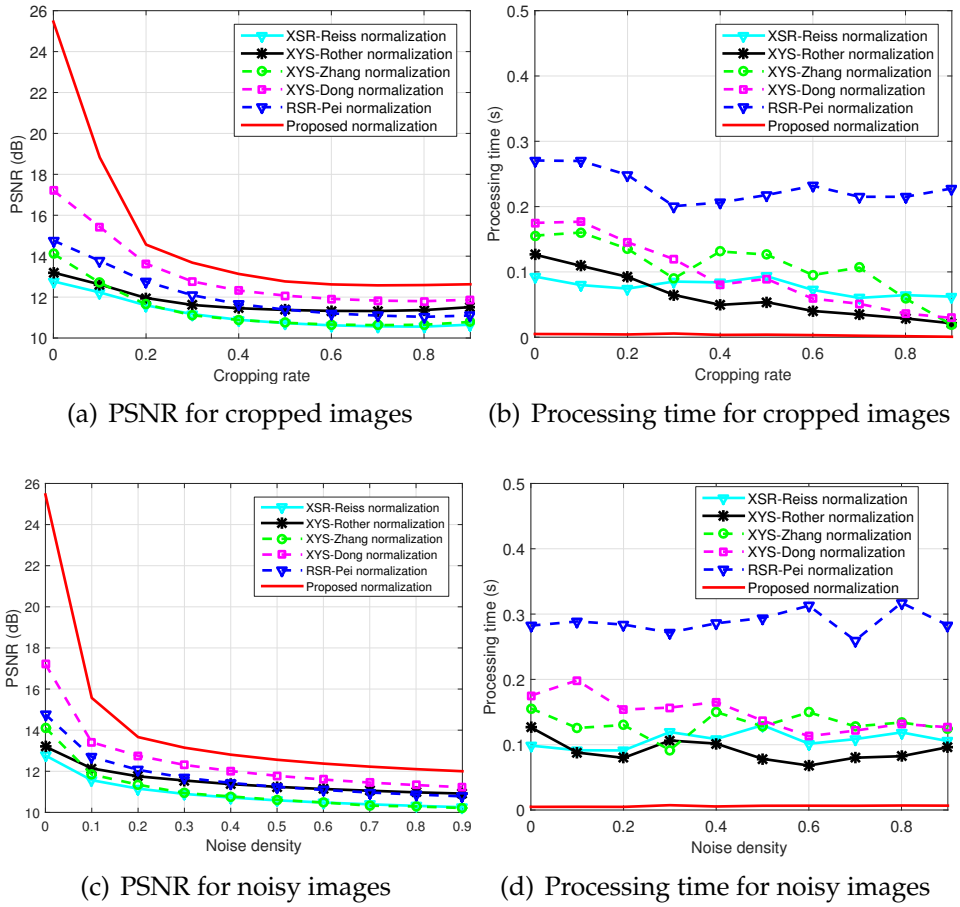
where $I_j \in$ class m, and $m \neq n$.

To measure the effects of the normalization algorithm on class-separability, the probability density function (PDF) and the cumulative distribution function (CDF) of ρ_w and ρ_b were computed for original images, affine-distorted images, and affine-normalized images. The receiver operating characteristic (ROC) curve computed from CDF was also used to compare the class separability. The detection rate of the image normalization is defined as $(1 - F_w)$, where the F_w is the CDF of ρ_w. The false-alarm rate of the image normalization is defined as $(1 - F_b)$, where the F_b is the CDF of ρ_b. A larger AUC value implies a better class separability. The analysis of class separability using PDF, CDF, ROC, and AUC will be shown in Subsection 4.3.4.

4.3.3 Analysis of affine normalization performance

In this section, the proposed image normalization is compared on the SUN397, MNIST, COIL-100, and ORL data sets with five existing normalization algorithms: XSR-Reiss normalization [202], XYS-Rothe normalization [190], XYS-Zhang [206], XYS-Dong normalization [194], and RSR-Pei normalization [199].

First, the six normalization algorithms are compared on affine-distorted and cropped images. The cropping rates vary from 0.0 to 0.9. A cropping rate of 0.0



(a) PSNR for cropped images

(b) Processing time for cropped images

(c) PSNR for noisy images

(d) Processing time for noisy images

Figure 4.12: Image normalization performance on the SUN397 data set with image cropping or noise.

means the image is only distorted by the affine transformations. A cropping rate of 0.4 means 40% of the image is removed.

Figure 4.12(a) shows the *PSNR* of the six normalization algorithms on the distorted SUN397 data set with different cropping rates. The *PSNR* values are always higher when cropping rate is zero (no cropping) than when cropping rate is non-zero (with cropping). The *PSNR* values also decrease when the cropping rate increases. For example, for the proposed normalization, the *PSNR* value is 25.5 dB when the cropping rate is 0.0, and 13.7 dB when the cropping rate is 0.3. Among the six normalization algorithms, the proposed algorithm has the highest *PSNR* value at all cropping rates.

Figure 4.12(b) shows the average processing time of the six normalization algorithms on the distorted SUN397 data set with different cropping rates. The processing-time for the proposed normalization algorithm is less than 0.01s for all cropping rates. The processing-time for the five existing normalization algorithms are higher than 0.01s for all cropping rates. With the increasing of the cropping rates, the processing-time is reduced for all algorithms. Furthermore, the processing-time variations for the proposed algorithm is much lower than the five existing algorithms. This means the proposed normalization is more stable for different affine distortions and image size.

Next, we evaluate the six normalization algorithms on affine-distorted images with random speckle noise. The noise density changes from 0.0 to 0.9. A noise density of 0.0 means no noise is added to the affine-distorted images.

Figure 4.12(c) shows the *PSNR* of the six normalization algorithms on the distorted SUN397 data set with different noise density. When the noise density increases from 0.0 to 0.9, the *PSNR* of the proposed normalization reduces from 25.5 dB to 12.0 dB. The average *PSNR* for the proposed algorithm is 14.2 dB. In comparison, the average *PSNR* values of the five existing normalization algorithms are 10.9 dB for the XSR-Reiss normalization, 11.1 dB for the XYS-Zhang normalization, 11.5 dB for the XYS-Rother normalization, 12.5 dB for the XYS-Dong normalization, and 11.7 dB for the RSR-Pei normalization. At all noise density, the proposed algorithm has higher *PSNR* values compared to the five existing algorithms.

Figure 4.12(d) shows the processing time of the six normalization algorithms on the distorted SUN397 data set with different noise density. At all noise density,

Table 4.5: Image normalization performance on the SUN397 data set with affine distortions.

Data set	*PSNR(dB)*	*Processing time(s)*
XSR-Reiss	12.8	0.0928
XYS-Rother	13.2	0.1265
XYS-Zhang	14.1	0.1556
XYS-Dong	17.2	0.1751
RSR-Pei	14.7	0.2706
Proposed method	25.5	0.0049

Table 4.6: Image normalization performance on the MNIST data set with affine distortions.

Data set	*PSNR(dB)*	*Processing time(s)*
XSR-Reiss	8.3	0.0233
XYS-Rother	6.6	0.0178
XYS-Zhang	8.2	0.0326
XYS-Dong	8.9	0.0329
RSR-Pei	8.3	0.0233
Proposed method	15.0	0.0004

the proposed normalization has a shorter processing time than the existing normalization algorithms. The processing-time variations are small for the proposed normalization and are large for the existing normalization algorithms, especially when the noise density increases.

Table 4.5 to 4.8 show the PSNR rates and processing time for the compared normalization algorithms on the SUN397, MNIST, COIL-100, and ORL data sets when the cropping rate and noise density are zero. The proposed normalization method has the highest *PSNR* rate and lowest processing time on the four data sets. All image normalization algorithms have better *PSNR* values on the COIL-100 data set. The reason is that images in the COIL-100 data set contain artificial objects that have regular shapes. All algorithms have lower *PSNR* values on the MNIST data set. The hand written digits in the original MNIST data set contain geometric distortions. It is difficult to normalize general geometric distortions using algorithms for affine-distortions.

Table 4.7: Image normalization performance on the COIL-100 data set with affine distortions.

Data set	*PSNR(dB)*	*Processing time(s)*
XSR-Reiss	18.9	0.0604
XYS-Rother	12.5	0.0267
XYS-Zhang	20.2	0.0536
XYS-Dong	26.0	0.0448
RSR-Pei	21.9	0.1443
Proposed method	35.3	0.0053

Table 4.8: Image normalization performance on the ORL data set with affine distortions.

Data set	*PSNR(dB)*	*Processing time(s)*
XSR-Reiss	19.4	0.0390
XYS-Rother	12.2	0.0566
XYS-Zhang	21.7	0.0628
XYS-Dong	22.3	0.0390
RSR-Pei	23.2	0.1168
Proposed method	29.4	0.0062

4.3.4 Analysis of normalization effects on class separability

In this section, the class separability of the four data sets (SUN397 of scene, MNIST of digits, COIL-100 of objects, and ORL of faces) was analyzed by the correlation coefficients ρ_w and ρ_b.

First, the intra-class similarity (ρ_w) and the inter-class similarity (ρ_b) were computed on the original images of the four data sets. Then, the PDF and CDF of ρ_w and ρ_b on the original images were calculated. Similarly, the PDF and CDF of ρ_w and ρ_b on the affine-distorted images and on the affine-normalized images were also computed.

Figure 4.13 shows the PDF and CDF of the intra-class similarity (ρ_w) and the inter-class similarity (ρ_b) of the COIL-100 data set. In the original data set, there are less affine variations within the same class. The average value of intra-class similarity ρ_w is higher than the average value of inter-class similarity ρ_b. In the distorted data set, all images have affine distortions. The mean value of ρ_w and ρ_b

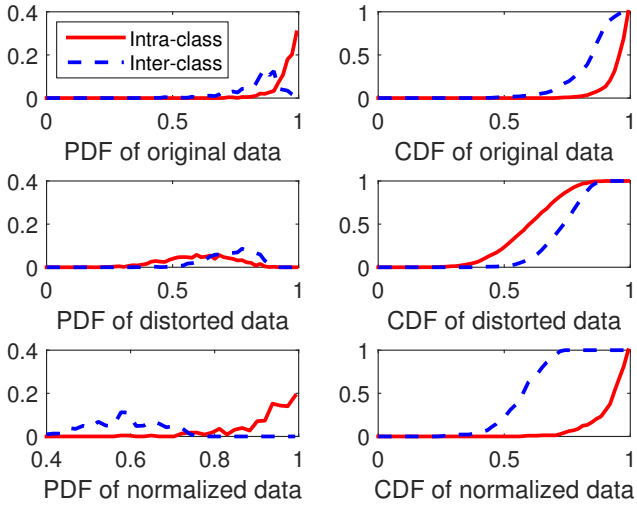


Figure 4.13: The PDF and CDF of the correlation coefficients for the COIL-100 data set of objects.

Table 4.9: Class separability as measured by AUC for original images, distorted images, and normalized images.

Data set	*Original data*	*Distorted data*	*Normalized data*
SUN397	0.2004	0.0619	0.5582
MNIST	0.9209	0.7201	0.9324
COIL-100	0.9125	0.2094	0.9941
ORL	0.9304	0.2634	0.9925

are both reduced. In the normalized data set, the proposed image normalization method increases the intra-class similarity ρ_w and reduce the inter-class similarity ρ_b. Figure 4.14 shows the ROC curves of the COIL-100 data set. The AUC value of normalized data is higher than the AUC value of the original data, and the AUC value of the original data is higher than the AUC value of distorted data. That indicates the image normalization increases the class separability on the COIL-100 data set.

Table 4.9 shows the area under ROC curves (AUC) of original data, distorted data, and normalized data for the four data sets. The AUC values for the normalized data on the four data sets are always higher than the AUC values for the distorted data and the original data.

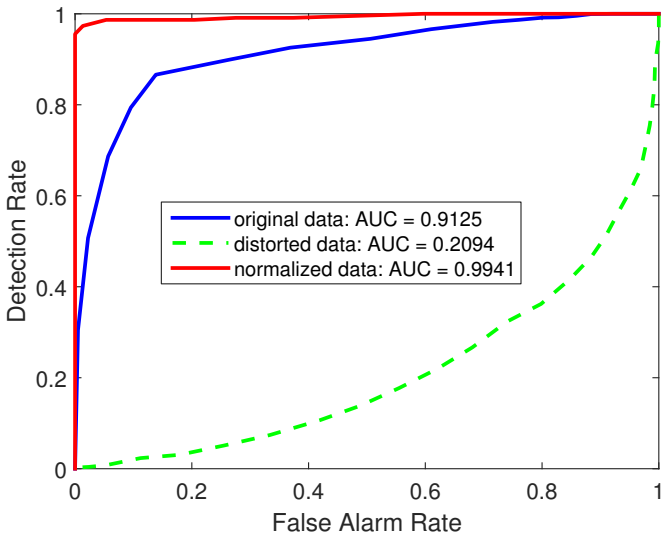


Figure 4.14: The ROC of the correlation coefficients for the COIL-100 data set of objects.

4.4 Conclusion

In this chapter, a new image approach to normalize affine distortions is presented. The proposed approach produces normalized images by solving an optimization problem based on image moments. The moment propositions used in our normalization method are presented and proved. In our experiments, the proposed method is compared with five existing normalization methods in terms of *PSNR* and processing time. The results show that the proposed image normalization is more robust to affine distortions, image cropping, and image noise. The class separability of images is also increased by applying the proposed normalization method.

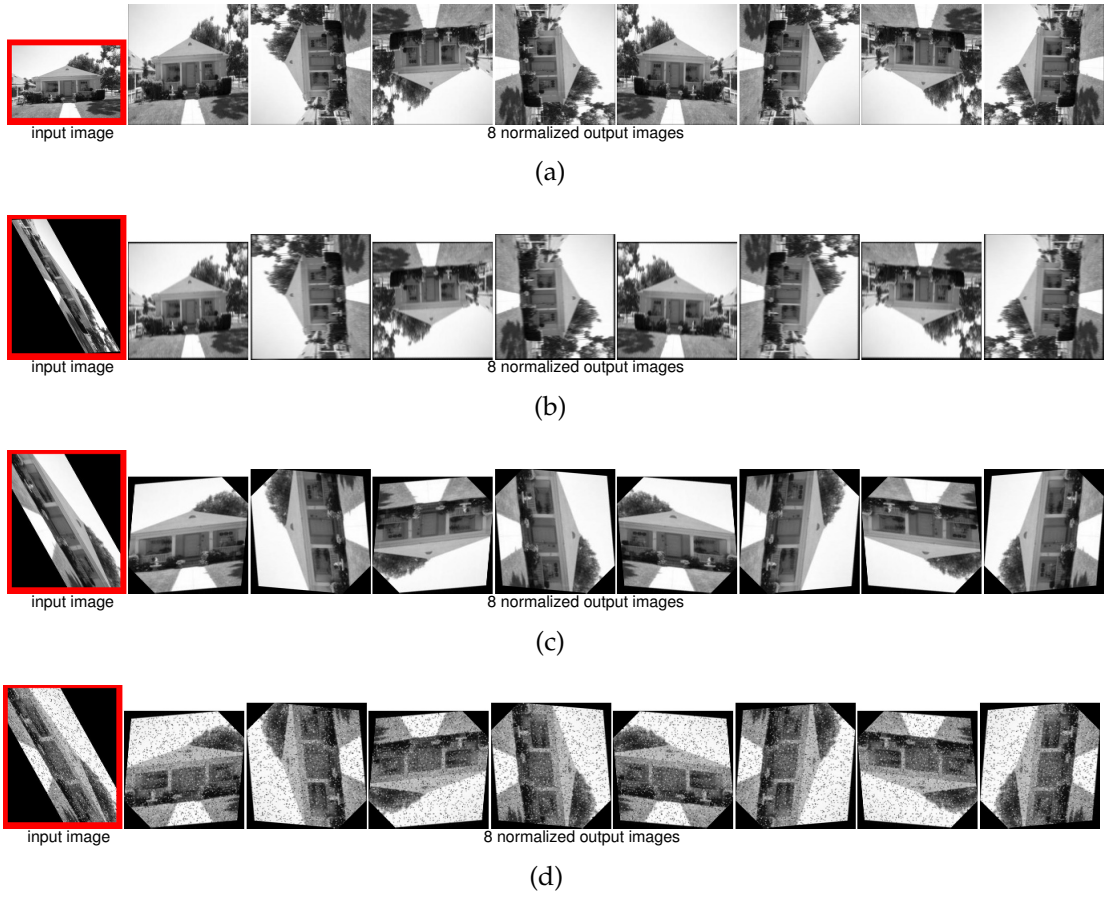


Figure 4.15: Examples of the proposed affine normalization. Column 1 is an input image, whereas Columns 2 to 9 are the 8 normalized images. The input image is: (a) an original non-distorted image, (b) an affine-distorted image, (c) an affine-distorted image with image cropping, (d) an affine-distorted image with image cropping and noise (noise density = 0.1).

Chapter 5

Image normalization for projective deformations

Chapter contents

5.1 Introduction

Robustness in image recognition refers to the ability to perceive an image pattern regardless of factors including camera views and locations. This chapter * presents a new algorithm that allows an image with arbitrary projective distortions to be

*Parts of Chapter 5 have been published in our paper "Invariant image recognition under projective deformations: an image normalization approach", *IEEE International Conference on Visual Communications and Image Processing*, 2015.

recognized efficiently. For an input image, the proposed algorithm generates a set of output images that are independent of the projective deformations, such as rotation, scaling, shearing, translation, and perspective projection. By producing projective-invariant images, our approach allows a system designed on a small set of normalized images to generalize well to an infinite number of projective deformations. In addition, it also reduces significantly the complexity and the cost of classifier training in image recognition tasks. We present a two-stage approach to calculate the 8 parameters of the required projective transformation matrix using image moments. The proposed algorithm is evaluated on two benchmark data sets.

The rest of the chapter is structured as follows. Section 5.2 describes existing methods for removing projective deformations. Section 5.3 presents the proposed image normalization approach to achieve projective invariance. Section 5.4 analyzes the results of image normalization on benchmark data sets.

5.2 Existing image normalization for projective deformations

Projective deformation is a more general type of geometric deformation. A number of approaches have been proposed to address projective deformations. In [198] and [207], the feature points of images were used to estimate normalization parameters and recover projective deformations. The normalization accuracy of these methods depends on the stability of feature-point detection.

Weiss proposed the differential invariants for the recognition of planar curves under projective deformations [208]. Given a curve and its first four derivatives

with respect to transformation parameter t, one can always find a canonical form that is independent of the original form. The canonical form is invariant to projective deformations. However, the differential invariants have problems in estimating high-order derivatives.

Suk and Flusser used image moments to normalize a shape with projective deformations [209]. They have proven that projective moment invariants have a form of infinite series containing moments with positive and negative indices. An advantage of image moments is that the integral quantities are less sensitive to noise. However, the moments used in this method have to be calculated from the whole object. This method is sensitive to partial occlusion. Note that image moments have been used for affine deformations in [190, 194, 210].

Zhang *et al.* proposed a rank minimization method to correct projective distortions on image texture, such as building facades, printed texts, and human faces [192]. They aim to extract invariant structures in 2-D images by undoing the domain transformations (affine or projective). In their method, the 2-D image contains regular patterns, whose appearance can be modelled as a low-rank matrix. By utilizing advanced convex optimization tools from matrix rank minimization, a low-rank texture is recovered from the associated deformations.

Given a deformed and corrupted image $I = (I^0 + E) \circ \tau^{-1}$ that contains a low-rank matrix I^0 and some error matrix E, the TILT algorithm recovers the low-rank matrix and finds the domain transformation τ, where $\circ$ is the image transformation operator. This formulation leads to the following optimization problem:

$$\min_{I^0, E, \tau} \text{rank}(I^0) + \lambda \|E\|_0 \quad \text{s.t.} \quad I \circ \tau = I^0 + E, \tag{5.1}$$

where $||E||_0$ is the number of non-zero entries in E.

However, the optimization problem in (5.1) is difficult to optimize. Under fairly broad conditions and by linearizing the constraint, (5.1) can be replaced by

$$\min_{I^0,E,\tau} ||I^0||_* + \lambda||E||_1 \quad \text{s.t.} \quad I \circ \tau + \nabla I \Delta\tau = I^0 + E, \tag{5.2}$$

where the nuclear norm ($||I^0||_*$) of a matrix is the sum of all its singular values and the l^1-norm of a matrix ($||E||_1$) is the sum of the absolute values of its entries. In [192], the optimization problem (5.2) is solved by the Augmented Lagrange Multiplier (ALM) method.

5.3 Image normalization for projective deformations

The aim of projective image normalization is to produce the same set of output images for any input image, which has been derived from an original undistorted image via an arbitrary projective transformation. A projective transformation is characterized by a transformation matrix T with 8 real parameters:

$$T = \begin{pmatrix} t_1 & t_2 & t_3 \\ t_4 & t_5 & t_6 \\ t_7 & t_8 & 1 \end{pmatrix}. \tag{5.3}$$

A pixel coordinate (x, y) in the input image I is mapped to a pixel coordinate (x^*, y^*) in the output image I^* as

$$\begin{pmatrix} sx^* \\ sy^* \\ s \end{pmatrix} = T \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}. \tag{5.4}$$

or

$$\begin{cases} x^* & = \frac{t_1 x + t_2 y + t_3}{t_7 x + t_8 y + 1} \\ y^* & = \frac{t_4 x + t_5 y + t_6}{t_7 x + t_8 y + 1} \end{cases}. \tag{5.5}$$

First, we observe that a projective transformation can be decomposed into an affine transformation followed by a simplified projective transformation:

$$I(x,y) \xrightarrow{\text{affine}} I'(x',y') \xrightarrow{\text{simplified projective}} I^*(x^*,y^*), \tag{5.6}$$

where

$$\begin{cases} x' &= t_1x + t_2y + t_3 \ , \\ y' &= t_4x + t_5y + t_6 \ . \end{cases} \tag{5.7}$$

and

$$\begin{cases} x* &= \frac{x'}{\alpha x' + \beta y' + \gamma} \ , \\ y* &= \frac{y'}{\alpha x' + \beta y' + \gamma} \ . \end{cases} \tag{5.8}$$

The parameters α, β, and γ in (5.8) are:

$$\begin{cases} \alpha = \frac{t_5t_7 - t_4t_8}{t_1t_5 - t_2t_4} \ , \\ \beta = \frac{t_1t_8 - t_2t_7}{t_1t_5 - t_2t_4} \ , \\ \gamma = 1 - \frac{t_3(t_5t_7 - t_4t_8)}{t_1t_5 - t_2t_4} - \frac{t_6(t_1t_8 - t_2t_7)}{t_1t_5 - t_2t_4} \ . \end{cases} \tag{5.9}$$

The affine transformation in (5.7) is represented by transformation matrix T_a, and the simplified projective transformation in (5.8) is represented by transformation matrix T_p:

$$T_a = \begin{pmatrix} t_1 & t_2 & t_3 \\ t_4 & t_5 & t_6 \\ 0 & 0 & 1 \end{pmatrix} \text{ and } T_p = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \alpha & \beta & \gamma \end{pmatrix}. \tag{5.10}$$

With this observation, we propose a method to find projective transformation, described by matrix T, to normalize an input image I. The proposed method has two stages. In the first stage, affine-transformation parameters t_1 to t_6 of matrix T_a are determined to map input image $I(x,y)$ to an affine-normalized image $I'(x',y')$. In the second stage, projective-transformation parameters t_7 and t_8 of matrix T_p are determined to generate the projective-normalized image $I^*(x^*,y^*)$. Note that once t_7 and t_8 are determined, the α, β, and γ are also calculated correspondingly.

5.3.1 Stage 1: Finding affine-transformation parameters t_1 to t_6

Transformation matrix T_a is found by solving the following constrained optimization problem:

$$T_a = \text{maximize } \{\eta'_{2,2}\} \text{ subject to } \begin{cases} \nu'_{1,0} & = 0 \\ \nu'_{0,1} & = 0 \\ \eta'_{2,0} & = c^2 \\ \eta'_{0,2} & = c^2 \\ \eta'_{1,1} & = 0 \end{cases}, \tag{5.11}$$

where c is a positive parameter to control the size of the output image. A larger value of c will produce a larger normalized image. Figure 5.1 shows example results of affine normalization (Stage 1). The input image in Stage 2 is the output image of Stage 1.

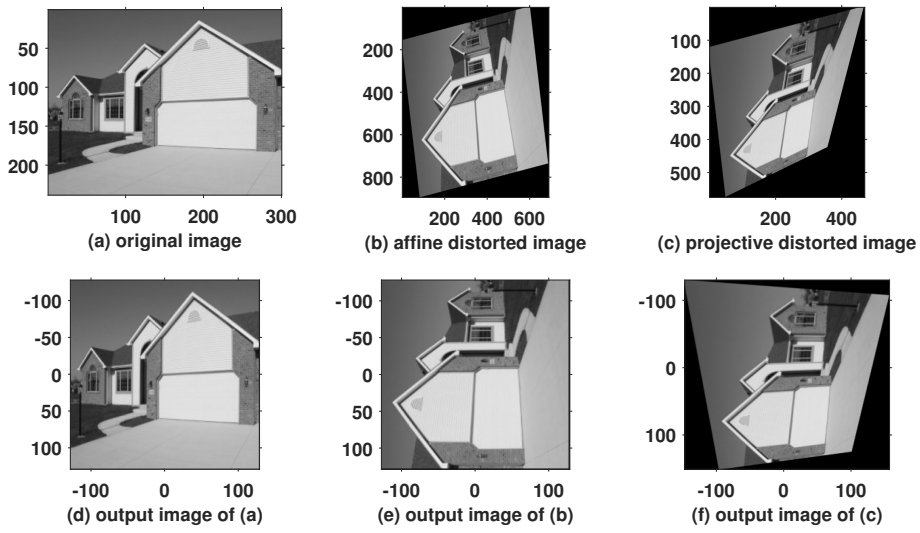


Figure 5.1: Examples of input images and their affine-normalized output images (Stage 1). The image in (a) is from the SUN397 data set.

5.3.2 Stage 2: Finding projective-transformation parameters t_7 and t_8

Consider the Cartesian coordinates shown in Fig. 5.2. Image rotations on the x-y plane are equivalent to image rotations around the z-axis in the 3-D space.

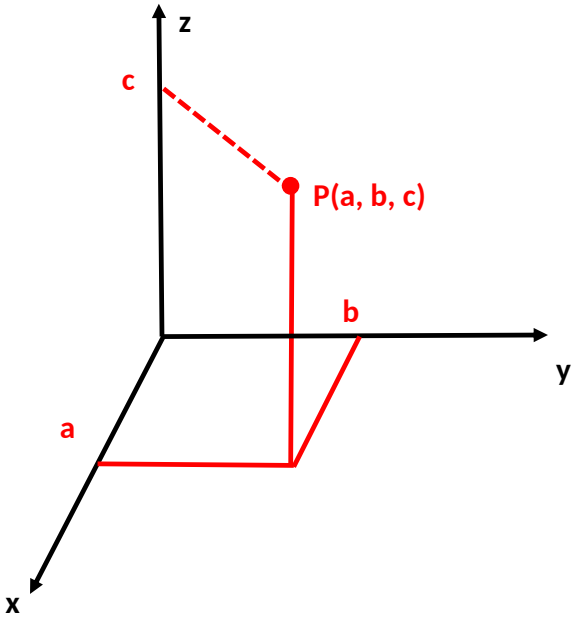


Figure 5.2: An example point in the 3-D Cartesian space.

Rotation counter-clockwise around the z-axis by an angle θ is represented as

$$[x',\ y',\ z'] = [x,\ y,\ z]\begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{5.12}$$

Similarly, rotation counter-clockwise around the y-axis by an angle θ is represented as

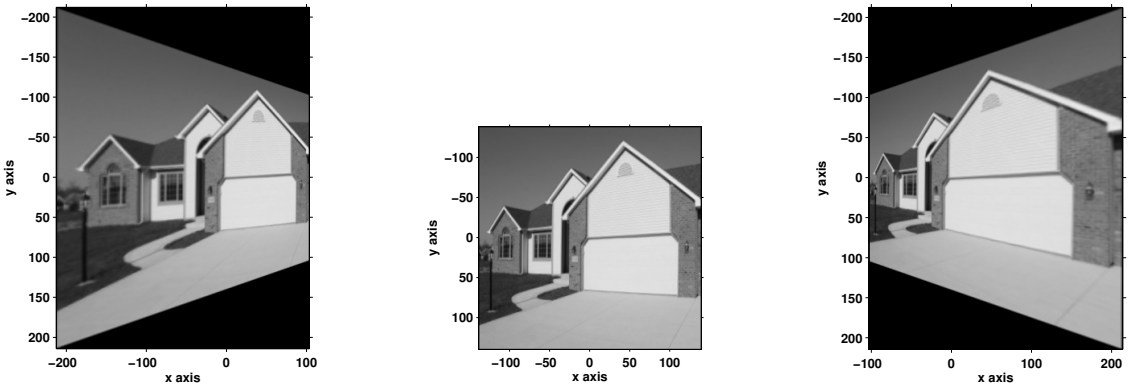
$$[x',\ y',\ z'] = [x,\ y,\ z]\begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix}. \tag{5.13}$$

Rotation counter-clockwise around the x-axis by an angle θ is represented as

$$[x',\ y',\ z'] = [x,\ y,\ z]\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix}. \tag{5.14}$$

By analyzing the relationship between parameter t_7, t_8 and the outputs of projective transformations, we find that the projective transformations for different values of parameter t_7 correspond to the image rotations around the y-axis in the 3-D space. The projective transformations for different values of parameter t_8 correspond to the image rotations around the x-axis in the 3-D space.

For parameter t_7, we project image rotations in the 3-D space to the x-y plane.

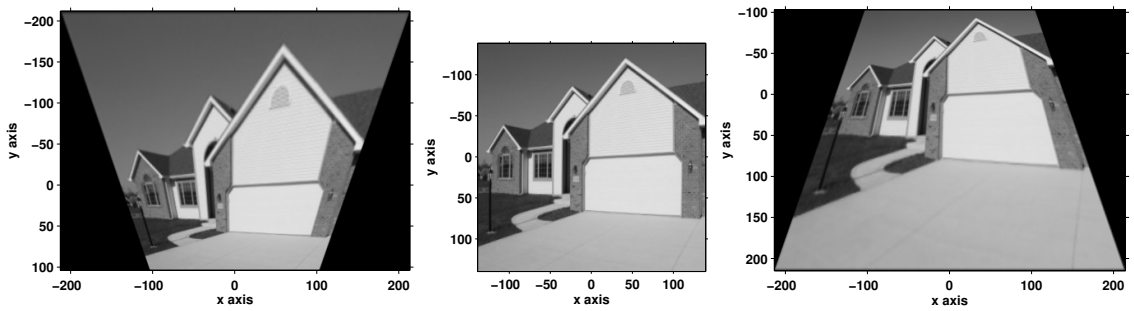


(a) $t_7 = 0.0025$ (b) $t_7 = 0.0000$ (c) $t_7 = -0.0025$

Figure 5.3: Image transformations with different values of t_7 on the x-y plane. They correspond to the image rotations around the y-axis in the 3-D space.

The projective transformations with different values of t_7 are shown in Fig. 5.3. After the affine-normalization step in Stage 1, the image is translated to the centroid $(\nu_{1,0}, \nu_{0,1})$. Then, with different values of t_7, the image is rotated around the line $x = 0$. In Fig. 5.3(a), the image is not normalized and the long side of the image is on the left of the rotation axis. In Fig. 5.3(b), the image is normalized. In Fig. 5.3(c), the image is not normalized and the long side of the image is on the right of the rotation axis. Therefore, we can find the value of t_7 at which the long side of the image switches from left to right.

For parameter t_8, we project the rotations in the 3-D space to the x-y plane. The projective transformations with different values of t_8 are shown in Fig. 5.4. After the affine-normalization in Stage 1, the image is translated to the centroid $(\nu_{1,0}, \nu_{0,1})$. Then, with different values of t_8, the image is rotated around the line $y = 0$. In Fig. 5.4(a), the image is not normalized and the long side of the image is on the top of the rotation axis. In Fig. 5.4(b), the image normalized. In Fig. 5.4(c), the image is not normalized and the long side of the image is on the bottom of the rotation axis. Therefore, we can find the value of t_8 at which the long side of the



(a) $t_8 = 0.0025$ (b) $t_8 = 0.0000$ (c) $t_8 = -0.0025$

Figure 5.4: Image transformations with different values of t_8 on the x-y plane. They correspond to the image rotations around the x-axis in the 3-D space.

image switches from top to bottom.

From (4.3), we know that the geometric moment on the x-y plane is

$$m_{p,q} = \iint_{\Gamma} x^p \; y^q \; I(x,y) \; dx \; dy, \tag{5.15}$$

where Γ denotes the support of the image.

The moments $m_{0,1}$ and $m_{1,0}$ calculated in Stage 2 are used to detect when the long side of the image switches its location. For example, the pixel with the minimum y value is located on the left border in Fig. 5.3(a), while the pixel with the minimum y value is located on the right border in Fig. 5.3(c). We can find the value of t_7 at which the distribution of minimum y value switches its location (see Fig. 5.3(b)). Similarly, the pixel with the minimum x value is located on the top border in Fig. 5.4(a), while the pixel with the minimum x value is located on the bottom border in Fig. 5.4(c). We can find the value of t_8 at which the distribution of minimum x switches its location (see Fig. 5.4(b)).

Parameters t_7 and t_8 are determined based on $m_{0,1}$ and $m_{1,0}$:

$$t_7 = \arg\max_{t_7} \{\frac{\partial^n m_{0,1}}{\partial t_7^n}\}, \tag{5.16}$$

$$t_8 = \arg\max_{t_8} \{\frac{\partial^n m_{1,0}}{\partial t_8^n}\}. \tag{5.17}$$

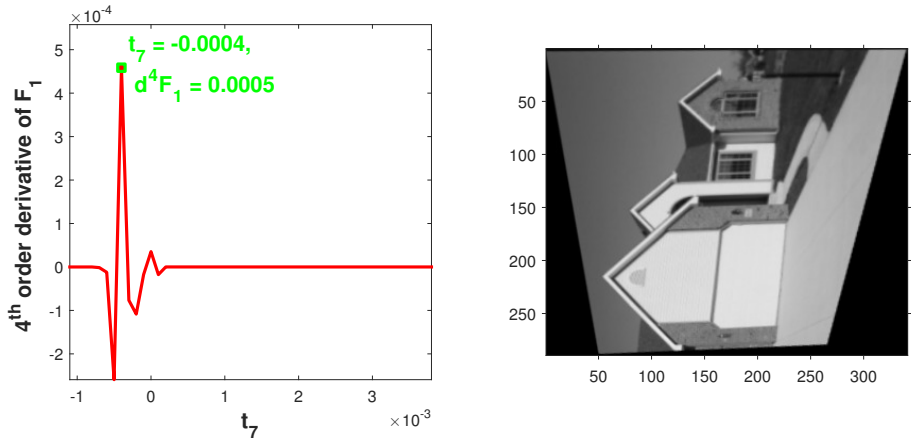


Figure 5.5: Finding t_7 of the projective transformation matrix T. *Left*: the 4^{th} derivative of $m_{0,1}$. *Right*: the normalized image using computed values of t_1 to t_7.

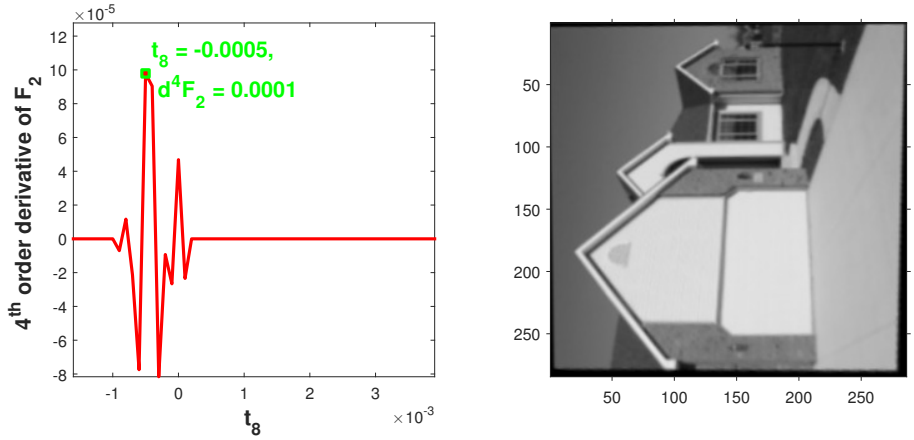


Figure 5.6: Finding t_8 of the projective transformation matrix T. *Left*: the 4^{th} derivative of $m_{1,0}$. *Right*: the normalized image using computed values of t_1 to t_8.

We first fix $t_8 = 0$, and use (5.16) to find t_7. Figure 5.5 shows an example of estimating t_7 for the input image in Fig. 5.1(c).

Once t_7 is found, we use (5.17) to estimate t_8. Figure 5.6 shows an example of estimating t_8 for the input image in Fig. 5.1(c). The normalization accuracy can be improved by increasing the numbers of iterations in Stage 2.

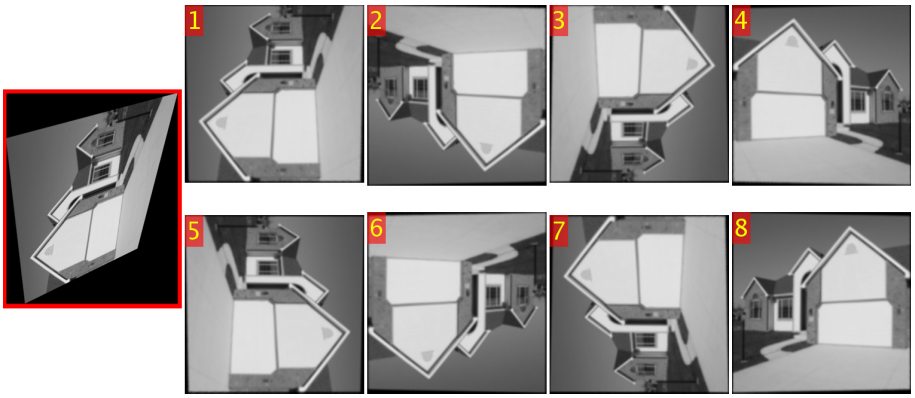


Figure 5.7: An example input image and its 8 projective-normalized images for the SUN397 data set.

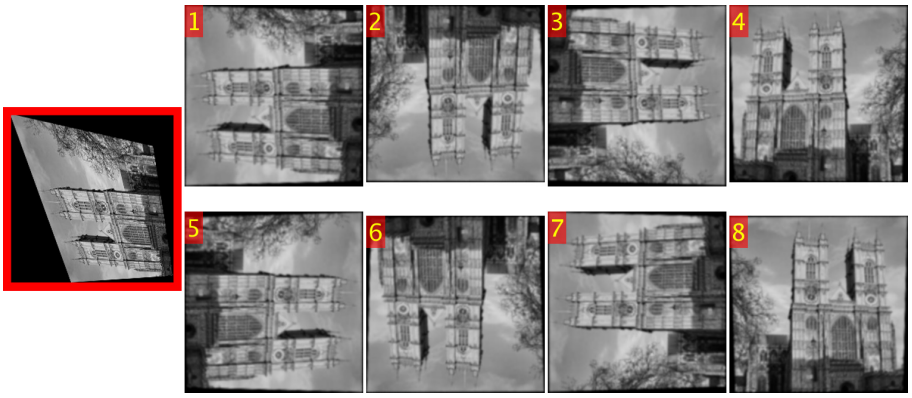


Figure 5.8: An example input image and its 8 projective-normalized images for the SUN397 data set.

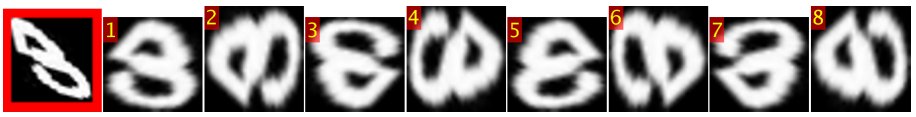


Figure 5.9: An example input image and its 8 projective-normalized images for digit 3 in the MNIST data set.

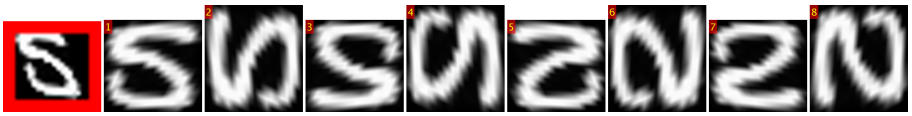


Figure 5.10: An example input image and its 8 projective-normalized images for digit 5 in the MNIST data set.

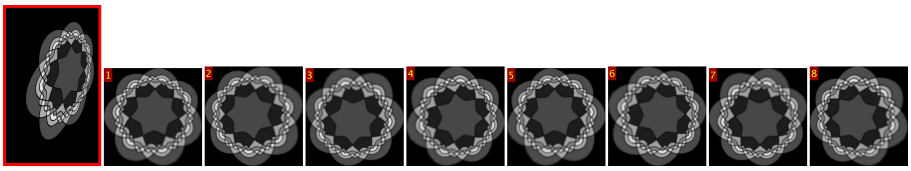


Figure 5.11: An example input image and its 8 projective-normalized images for a symmetric pattern.

Finally, the output image I^* is rotated by 90°, 180°, 270°, 90° and flipped vertically to generate 8 normalized images. The orientation of the first normalized image I^* depends on the orientation of the input image. Examples of input projective-distorted images and their 8 output normalized images are shown in Fig. 5.7 to 5.11.

5.4 Experimental evaluation and results

In this section, the proposed image normalization method is compared with three existing image normalization methods. Section 5.4.1 describes the data sets used in the experiments and defines the performance measures for image normalization. Section 5.4.2 presents the results of projective normalization.

5.4.1 Experimental methods

We evaluated the performance of the proposed normalization algorithm on two data sets: the MNIST handwritten digit data set [51] and the SUN397 data set [211]. The MNIST data set contains 70,000 handwritten digits from 500 different writers. The SUN397 data set has 397 scene categories and at least 100 images at various scales in each category.

For these two data sets, the distorted images were generated by applying random projective transformations on the original images. The size of distorted

images for the MNIST data set are between 5×5 and 100×100 pixels. The size of distorted images for the SUN397 data sets are between 50×50 and 2000×2000 pixels. After applying the proposed normalization algorithm on the original or distorted images, we obtained the normalized images. The normalization parameter c in Eq. (5.11) is set to 8 for the MNIST data set, and 80 for the SUN397 data sets.

Similarly to [198], we measured the accuracy of image normalization using the difference-image. Let I^* be the normalized image of original image I, and I_d^* be the normalized image of distorted image I_d. The difference-image is defined as $I^* - I_d^*$. The normalization score (N_s) is calculated as

$$N_s = (1 - \frac{\|I^* - I_d^*\|_1}{\|I\|_1}) \times 100\%, \tag{5.18}$$

where $\|\cdot\|_1$ denotes the ℓ_1-norm. A larger N_s value implies a more precise normalization. Among the 8 normalized images, only the image with the highest correlation score with the original image I is used to compute the normalization score in (5.18).

We compared the proposed normalization method with two affine-normalization methods based on image moments [190, 194], and a projective-normalization method based on rank minimization [192]. The two affine-normalization methods decompose an affine transformation into a series of simplified transformations, like scaling, shearing, and rotation. The constraints used in [190] is $\mu'_{3,0} = 0, \mu'_{1,1} = 0, \mu'_{2,0} = 1$, and $\mu'_{0,2} = 1$. The constraints used in [194] is $\mu'_{3,0} = 0, \mu'_{1,1} = 0$, $\mu'_{5,0} > 0$, and $\mu'_{0,5} > 0$. The rank minimization method corrects projective deformations for regular and near-regular patterns or objects (e.g. building facades, printed text,

and human faces). In this method, input image is considered as a matrix, and a geometric transformation is determined to minimize the rank of the output image.

5.4.2 Experimental results

First, we analyzed the number of iterations for Stage 2 to achieve good normalization results. As shown in Fig. 5.12, with the increasing of iterations, the average normalization scores on sample images are increased from around 64% (affine normalization without projective normalization) to around 80% (the third iteration). In the rest of our experiments, the Stage 2 are repeated 3 times to get the normalized images.

Table 5.1: Comparison of image normalization algorithms on MNIST.

Method	*Processing time (s)*	N_s (%)
Proposed projective normalization	0.47 ± 0.10	75.6 ± 8.9
Rank minimization [192]	0.81 ± 0.85	51.9 ± 20.0
Affine normalization-Rothe [190]	0.02 ± 0.01	60.5 ± 27.6
Affine normalization-Dong [194]	0.02 ± 0.01	50.6 ± 22.4

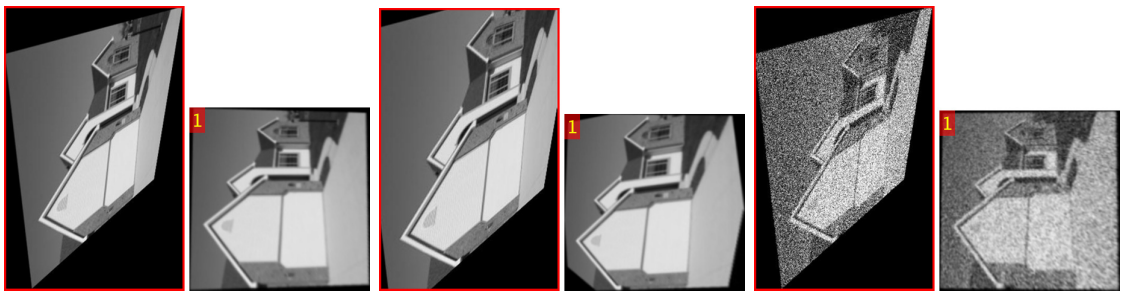
Table 5.2: Comparison of image normalization algorithm on SUN397.

Method	*Processing time (s)*	N_s (%)
Proposed projective normalization	1.16 ± 0.25	82.4 ± 10.0
Rank minimization [192]	1.73 ± 1.21	53.6 ± 25.2
Affine normalization-Rothe [190]	0.03 ± 0.01	63.2 ± 17.3
Affine normalization-Dong [194]	0.04 ± 0.01	70.0 ± 13.6

Then, we analyzed the speed and accuracy of normalization on the MNIST and SUN397 data sets. Table 5.1 and 5.2 present the processing time and normalization accuracy for the MNIST and SUN397 data sets, respectively. These tables show that the proposed normalization algorithm is faster than the rank minimization algorithm. Moreover, the processing-time variation for the rank minimization algorithm is about 7 times higher than the proposed algorithm. The

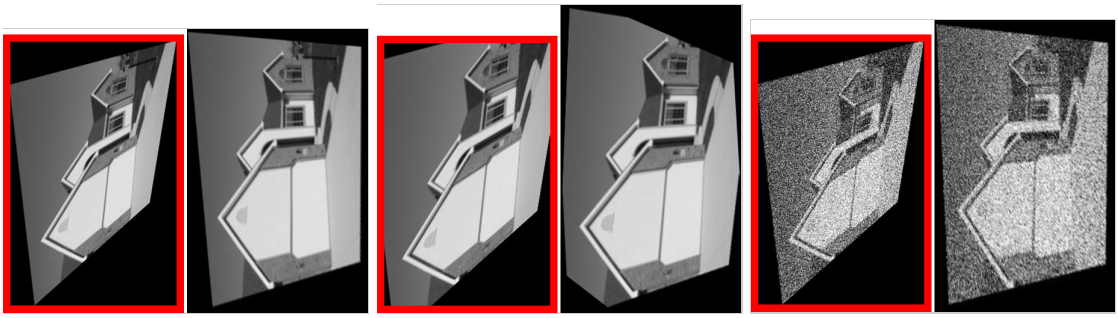
normalization scores N_s of the proposed method (75.6% and 82.4%) are higher than that of the rank minimization method (51.9% and 53.6%), affine normalization method proposed by Rothe *et al.* [190] (60.5% and 63.2%), and affine normalization method proposed by Dong *et al.* [194] (50.6% and 70.0%). These results indicate the proposed normalization is more efficient and accurate than the existing normalization methods, when tested on a wide range of images and projective deformations.

We also tested the four compared normalization methods on projective-distorted images, projective-distorted and cropped images, and projective-distorted and noisy images. Figure 5.13 to 5.16 show examples of the normalization results. As shown in Fig. 5.13, the proposed method corrects projective distortions consistently well on different images. As shown in Fig. 5.14, the rank minimization method can reduce geometric distortions. However, the projective distortion still exists in the output images. As shown in Fig. 5.15 and 5.16, the algorithms used for affine-distorted images do not work for projective-distorted images.



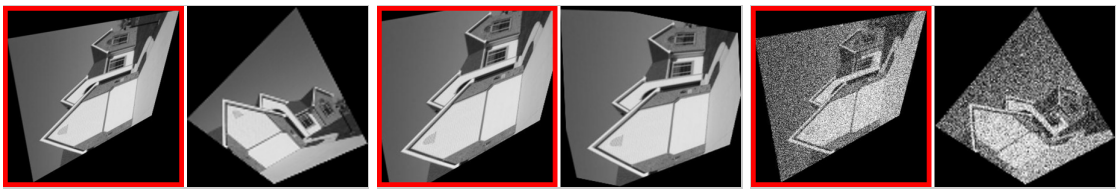
(a) (b) (c)

Figure 5.13: Examples of normalized images for the proposed method. Only the first normalized image is shown in this example. Input images are highlighted by the red border: (a) projective-distorted image, (b) projective distorted and cropped image, and (c) projective distorted and noisy image.



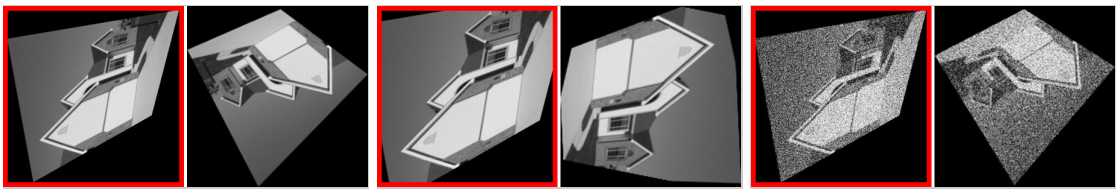
(a) (b) (c)

Figure 5.14: Examples of normalized images for rank minimization method. Input images are highlighted by the red border: (a) projective-distorted image, (b) projective distorted and cropped image, and (c) projective distorted and noisy image.



(a) (b) (c)

Figure 5.15: Examples of normalized images for XYS-Rothe normalization. Input images are highlighted by the red border: (a) projective-distorted image, (b) projective distorted and cropped image, and (c) projective distorted and noisy image.



(a) (b) (c)

Figure 5.16: Examples of normalized images for XYS-Dong normalization. Input images are highlighted by the red border: (a) projective-distorted image, (b) projective distorted and cropped image, and (c) projective distorted and noisy image.

5.5 Chapter summary

In this chapter, a two-stage normalization method for projective deformations is presented. In the first stage, affine-transformation parameters t_1 to t_6 are

determined. In the second stage, projective-transformation parameters t_7 and t_8 are determined. The proposed normalization method produces the same set of normalized images for projective distorted images. Our experiments show that the proposed method is more accurate than the existing normalization methods.

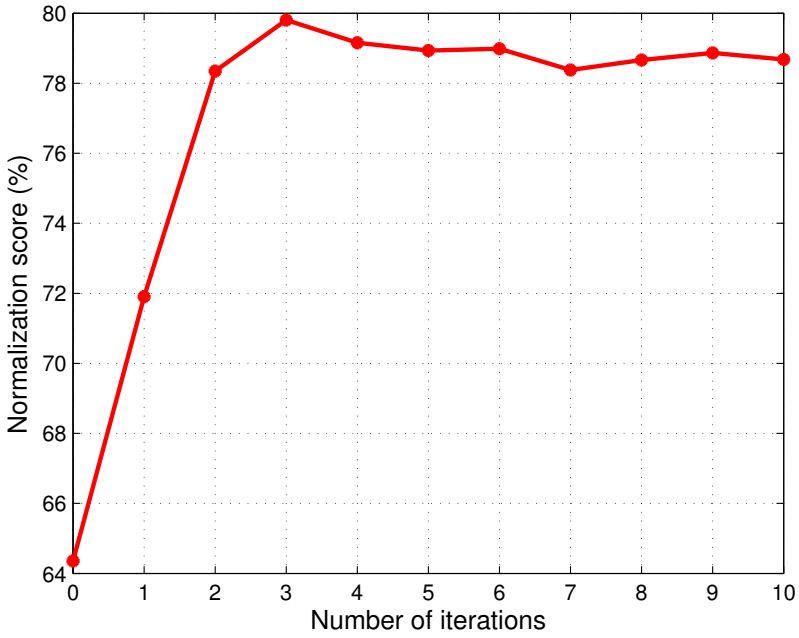


Figure 5.12: The normalization scores with different number of iterations for Stage 2 on the SUN397 data set.

Chapter 6

Scene categorization under geometric deformations

Chapter contents

This chapter * presents a scene categorization approach that is invariant to geometric distortions. We apply the image normalization algorithms proposed in Chapter 4 and Chapter 5 to generate an image, which is independent of the position, scale, shear, rotation, and projection of the input image. The proposed

*Parts of Chapter 6 have been published in our papers "Affine-invariant scene categorization," *IEEE International Conference on Image Processing*, pp. 1031-1035, 2014 and "Invariant image recognition under projective deformations: an image normalization approach", *IEEE International Conference on Visual Communications and Image Processing*, 2015.

approach produces normalized images before visual descriptors are extracted for scene categorization. We investigate the effects of the two proposed image normalization methods on several state-of-the-art visual descriptors for scene categorization. We also combine different visual descriptors to improve the scene categorization performance. The experimental results on several benchmark data sets show that the proposed image normalization methods are robust to affine distortion, image cropping, and image noise. Furthermore, the proposed normalization method improves significantly scene categorization of geometric-distorted images. Under affine distortions, the Places-CNN features combined with GIST features achieve the best classification performance on several benchmark data sets. Under projective distortions, the Places-CNN features achieve better classification performance on the 15-scene data set.

6.1 Introduction

Recognition of objects that are deformed geometrically has been a goal of recent research. The existing approaches for affine invariance can be classified into three categories: invariance by training, invariance by feature extraction, and invariance by image normalization.

In *invariance by training*, images used for training contain not only the original images but also their rotated, scaled, and deformed versions. Decoste and Scholkopf trained invariant support vector machines for handwritten digit recognition by augmenting training samples with different scales, rotations, and line thicknesses [182]. Tivive and Bouzerdoum developed a rotation-invariant face versus non-face classifier by training a convolutional neural network on a

large number of rotated face patterns [183]. Pereira *et al.* proposed a multi-pose face detection approach by training a classification tree using rotated face images [184]. These invariance-by-training techniques can easily be applied to scene categorization. However, brute-force training is time consuming, and if the training set is not carefully designed, the classifier may not learn the desired invariance.

In *invariance by feature extraction*, objects are described by features that are insensitive to a particular deformation. Significant work has been reported on affine-invariant feature extraction. Hu used moment invariants to develop visual features that are independent of position, size, and orientation of the object [185]. Flusser presented a general scheme for deriving affine-invariant features based on image moments [186]; these features were later adopted for handwritten digit recognition in [212]. Recently, researchers have proposed scale-invariant and rotation-invariant descriptors, such as SIFT [36], SURF [66], and BRISK [72]. In these descriptors, scale-invariance is achieved by scale-space keypoint detection, whereas rotation-invariance is achieved by orientation assignment. Morel and Yu proposed an affine invariant extension of SIFT for image matching [213]. Their algorithm achieves affine invariance by rotating and tilting input images to a finite number of deformed images. The deformed images are then compared by the original SIFT algorithm. Bruna and Mallat proposed a translation-invariant descriptor: scattering transform [214]. The scattering transform has been used for texture recognition in [188]. Besides local visual descriptors, global feature formation methods, such as histograms [37] and down-sampling [32] also reduce the sensitivity of features to geometric transformations. Oliva and Torralba built the GIST descriptor by averaging features in sub-regions [32]. Ojala *et al.* proposed

the rotation-invariant LBP algorithm using histograms and Fourier coefficients [104]. Recently, global formation methods have been improved via feature coding with sparse modelling [65] and low-rank property [215].

In *invariance by image normalization*, an input image is normalized before it is classified. Existing affine normalization methods have been used for applications such as image watermarking [194], handwritten character recognition [174, 197], and face recognition [216]. Several affine normalization methods were implemented on image moments. For example, Rothe *et al.* proposed a moment-based normalization method that decomposes the unknown affine transformation into skew, non-uniform scaling, and rotation [190]. Zhang *et al.* studied the ambiguities of the moment-based affine normalization, and proposed a method to choose a consistent normalized image [201]. Suk and Flusser decomposed the affine transformations and formed normalized images by means of low-order moments [191]. Different from the moment-based methods, Pei and Lin proposed an image normalization method based on the covariance matrix and tensor theory [199]. Yasein and Agathoklis developed an affine normalization method using image feature points [198]. The normalization parameters are estimated from the three key points that have the highest responses during the feature-detection stage.

In this chapter, we extract different visual features after applying the proposed image normalization methods on input images, to produce geometric-invariant scene categorization. Our approach is inspired by psychovisual evidences that humans have orientation preference when recognizing visual patterns [89, 217]. For example, people recognize an upright-frontal face easily, but if the face is inverted or rotated, recognition speed and accuracy reduce significantly [218, 219]. Our

approach allows affine-invariant features to be extracted after image normalization, thereby reducing the complexity of scene classifiers and the cost of classifier training. Furthermore, because deformations caused by camera viewpoints can be locally approximated by affine transformations [177, 213], this method is a step towards developing a view-invariant scene categorization system. In scene categorization, once the input image is normalized against geometric transformations, any features can be extracted and used for classification. In this chapter, we apply the proposed normalization methods to several state-of-the-art visual descriptors, and investigate how it affects the scene categorization performance of the descriptors.

The rest of the chapter is structured as follows. Section 6.2 describes the features used for scene categorization. Section 6.3 analyzes the results of image normalization and scene categorization on several benchmark data sets under geometric distortions.

6.2 Feature extraction and combination for scene categorization

In most scene categorization methods, visual features are first extracted from images and then classified into semantic categories. Hence, the feature extraction plays a central role in scene categorization. Low-level visual features, such as color, shape, and textures have been successfully utilized to classify indoor or outdoor scenes [220, 221, 222]. However, they are sensitive to geometric distortions and illumination changes. Therefore, local descriptors, such as LBP [104] and HOG [37], which are robust to lighting, scaling, or orientation

changes, are applied to scene categorization.

One of the most popular local descriptors used in scene categorization is the SIFT proposed by Lowe [36]. The original SIFT features achieve scale-invariance by detecting local extrema in the scale space. For scene categorization, the dense SIFT features are extracted from overlapped patches of the input image [3, 65, 215, 223, 224].

The local descriptors ignore spatial information and produce a large number of features. Therefore, the global feature formation methods, such as spatial histograms [50, 144, 225], principal component analysis [73], and bag-of-words [3, 65], are proposed to summarize the spatial information of an image and reduce the dimension of features. The spatial pyramid matching (SPM) proposed by Lazebnik *et al.* is a global feature formation method, which encodes the dense SIFT features using vector quantization and spatial histograms [3]. This method combines the local features (SIFT) and global features (SPM) to classify images. The sparse-coding spatial pyramid matching (ScSPM) also forms global features from local features [65]. It uses sparse coding to quantize the local features. For spatial pooling, the original SPM uses histograms, whereas the ScSPM uses the max operator, which is more robust to local translations.

Inspired by the human visual system, researchers also developed computational vision models for scene categorization. Oliva and Torralba proposed the *GIST* descriptor, which is the statistical summary of the spatial layout of the scene [226]. Serre *et al.* introduced a scene categorization method that is inspired by the organization of human visual cortex [79]. The features used in their method are invariant to position and scale. Inspired by the discoveries of Hubel and

Wiesel [89] about the receptive fields in mammal visual cortex, LeCun *et al.* [51] developed the *convolutional neural networks* (CNN). In recent years, deep learning architectures have gained fervent research interest for image recognition.

The existing visual features and models are mostly proposed to classify images captured from limited views. In practical applications, the scene could be imaged from arbitrary views, causing image variations that must be addressed by the scene categorization system. We have already evaluated the existing descriptors for scene categorization without affine deformations in Chapter 3. In the next section, we will evaluate the existing visual descriptors (SIFT-ScSPM, SIFT-SPM, GIST, LBP, and Places-CNN) with affine deformations.

We also combine different visual descriptors and investigate their scene categorization performance under affine deformations. The moment-ScSPM descriptor combines affine-invariant features (image moments [186]) with a global feature formation method (ScSPM [65]). The scattering-ScSPM descriptor combines the translation-invariant descriptor (scattering transform [188]) with the ScSPM. We also combines the deep learning features (Places-CNN [52]) with hand-designed features (GIST [226]) for scene categorization. In the next section, we will compare the three combined descriptors with the existing visual descriptors for scene categorization.

6.3 Experimental evaluation and results

In this section, the proposed image normalization and scene categorization algorithms are evaluated on the 8-outdoor scene, 15-scene, 67-indoor-scene, and SUN397 data sets. Section 6.3.1 presents the implementation of the visual de-

scriptors and the classifier. Section 6.3.2 and 6.3.3 evaluate different descriptors with or without the image normalization under affine distortions. Section 6.3.4 evaluate different descriptors with or without the image normalization under projective distortions.

6.3.1 Feature extraction and classification

This section presents the implementation of the descriptors and the classifier compared in this chapter.

- The *Places-CNN* features are generated from a convolutional neural network that is trained on 205 scene categories of Places database with 2.5 million images [52]. The CNN has 5 convolution layers, 650000 neurons, and 60 million parameters. The dimension of the Places-CNN deep features is 4096.

- The *SIFT-ScSPM* descriptor calculates local features using the dense-SIFT algorithm and forms global features using the ScSPM algorithm [65]. In the experiment, the patch size to extract SIFT features was 16×16 pixels. The input patch was first filtered with Gaussian filters to generate 8 orientation maps. The histograms were then generated from the orientation maps and further weighted by a Gaussian function. The SIFT descriptor was formed from all entries of these histograms. After extracting the local features, we built a feature dictionary based on k-means clustering. The sparse-coding spatial pyramid matching (ScSPM) was then applied to convert the local SIFT features to global features.

- The *SIFT-SPM* descriptor calculates local features using SIFT [36]. The global

features are then generated by the spatial pyramid matching (SPM) [3]. In the experiments, the SIFT features were calculated from overlapped patches (16×16) on a dense grid. Then, the *k*-means clustering and PCA were used to train and extract 200 visual words. The SIFT features were quantized by the trained visual words. Finally, the histograms of quantized features were computed with 1000 bins.

- The *GIST* descriptor is the statistical summary of the spatial layout of the scene [226]. To extract GIST features, the image was first padded, whitened, and normalized to reduce the blocking artifact. Next, a set of multi-scale oriented Gabor filters was generated from one mother wavelet, through dilation and rotation. The input image was convolved with Gabor filters at 4 scales and 8 orientations. Each filtered output was down-sampled to a 4-by-4 matrix, and reshaped to a vector with 16 elements. The GIST descriptor was obtained by combining all outputs from the 32 filters and forming a feature vector with 512 elements.

- The *LBP* descriptor extracts histogram features from the LBP map [104]. In the experiment, the image was first divided into cells. Each pixel in the cell was compared with its 8 neighboring pixels to form an 8-bit LBP pattern. If the center pixel value was greater than the neighbor pixel, the corresponding LBP bit was set to 1. Otherwise, the LBP bit was set to 0. The LBP features were formed from the histogram of the LBP patterns. The LBP map was generated from image cells of size 3-by-3 pixels. The number of histogram bins of the LBP was 256.

- The *moment-ScSPM* descriptor adopts the same framework of SIFT-ScSPM. The only difference is the local features are extracted by the moment invariants [186]. In the experiment, the local moment invariants were calculated from overlapped image patches (16×16). Each patch formed 17 moment invariants. The global features were calculated by the ScSPM algorithm.

- The *scattering-ScSPM* calculates local features from overlapped patches on a dense grid. The local feaures are extracted by scattering transform [188]. In the experiments, the size of the overlapped patch was 16×16 pixels. The distance between adjacent patch centers was 8 pixels. The dictionary training and global feature formation of the scattering-ScSPM were the same as the SIFT-ScSPM and moment-ScSPM descriptor.

- The *Places-CNN-GIST* concatenates 4096 Places-CNN features and 512 GIST features. Places-CNN learns the high-level features and GIST features are the statistical summary of the spatial layout of the scene.

Many tools can be used to classify the extracted features. This visual system uses *support vector machine* (SVM) as the main classification tool for its excellent generalization ability. In an SVM, the decision boundary is obtained from the training data by finding a separating hyperplane that maximizes the margins between the two classes. For complex problems involving nonlinear decision boundaries, the SVM projects the data onto a high-dimensional space using kernel methods.

In the experiment, the *one-versus-all* SVM with the linear, RBF, or HIK kernel was applied to classify the extracted feature vectors. With image normalization,

each test image generates 8 normalized images. The first 4 normalized images of candidate 1 are equivalent to the 4 normalized images of candidate 2 via horizontal flipping. Hence, the classification result for an input image was determined from the average classification score of the 4 normalized images from candidate 1.

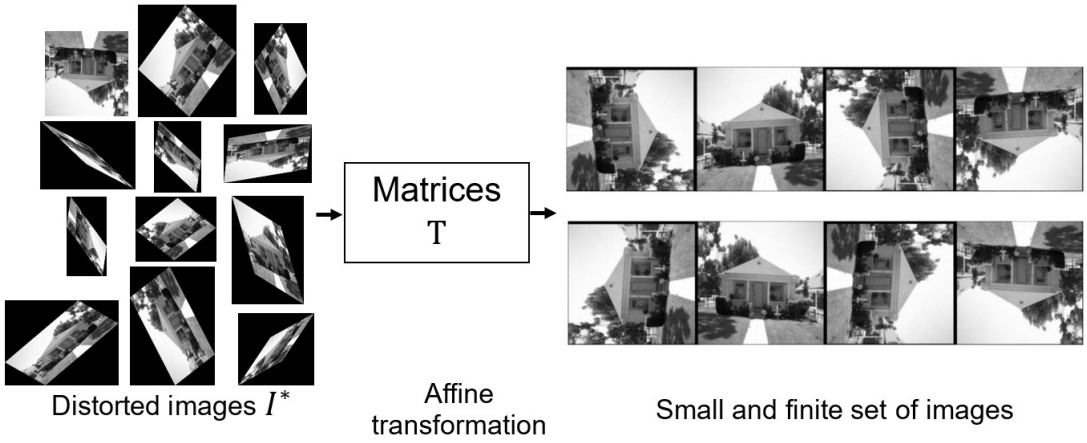


Figure 6.1: Image normalization for affine-distortions. Distorted images are mapped to a small set of normalized images using transformation matrices T.

6.3.2 Analysis of scene categorization on the 15-scene database under affine deformations

In this section, we evaluate how the image normalization method for affine distortions affects the performance of the state-of-the-art descriptors. Figure 6.1 shows how the affine normalization is performed on the affine-distorted images. The descriptors evaluated in our experiments are: Places-CNN [52], SIFT-ScSPM [65], SIFT-SPM [3], GIST [226], LBP [38], moment-ScSPM [186], scattering-ScSPM [188],

Table 6.1: Scene categorization performance on the 15-scene database using SVM.

Algorithms	*CR (%)*	*Precision (%)*	*Recall (%)*	*F-measure (%)*	*AUC (%)*
Places-CNN (linear)	91.8 ± 1.2	92.0 ± 1.2	91.8 ± 1.3	91.8 ± 1.3	99.5 ± 0.0
SIFT-ScSPM (linear)	84.5 ± 1.5	84.8 ± 1.7	84.0 ± 1.4	84.1 ± 1.4	98.9 ± 0.1
SIFT-SPM (HIK)	77.9 ± 1.2	77.5 ± 1.3	76.8 ± 1.0	76.8 ± 1.0	97.0 ± 0.3
GIST (RBF)	72.8 ± 1.2	72.2 ± 1.0	71.8 ± 1.4	71.5 ± 1.2	95.2 ± 0.2
LBP (HIK)	71.9 ± 2.5	71.0 ± 3.2	71.1 ± 3.2	70.5 ± 3.4	95.7 ± 0.8
Moment-ScSPM (linear)	61.6 ± 2.4	60.4 ± 2.3	60.2 ± 2.7	59.4 ± 2.4	93.0 ± 0.6
Scattering-ScSPM (linear)	84.8 ± 2.3	84.8 ± 3.0	84.6 ± 2.7	84.5 ± 3.0	99.3 ± 0.3
Places-CNN-GIST (linear)	92.4 ± 0.9	92.5 ± 1.1	92.2 ± 1.2	92.2 ± 1.1	99.5 0.0±

and Places-CNN-GIST. Among these 8 descriptors, moment-ScSPM, scattering-ScSPM, Places-CNN-GIST are analyzed for the first time for scene categorization. The evaluation measures are the classification rate (*CR*), precision (*P*), recall (*R*), F-measure (*F*), and area-under-the-ROC (*AUC*).

Table 6.1 shows the scene categorization performance of the compared descriptors on the original 15-scene database. The Places-CNN-GIST achieves the highest *CR* (92.4%) among all the compared descriptors. Combining the high-level features (Places-CNN) and the statistical summary of the scene (GIST) has better classification performance than other compared descriptors. Combining with the same global formation algorithm (ScSPM), the scattering features have a better classification performance (84.8%) than the SIFT features, which have been widely used for scene categorization. Furthermore, the descriptors that use ScSPM algorithm to form global features have higher *CRs* than the descriptors that use other global formation algorithms. For example, the *CR* of SIFT-ScSPM (84.2%) is higher than the *CR* of SIFT-SPM (77.9%).

Next, we evaluate the scene categorization algorithms on the affine-distorted 15-scene database. The results are shown in Table 6.2, and several observations can be made. First, without image normalization (Table 6.2, Column 2), the *CRs* of the 8 descriptors are much lower than the *CRs* on the non-distorted images (Table 6.1). This means affine distortions have severe effects on the existing scene categorization algorithms.

Second, on the distorted database, image normalization leads to higher *CRs* than without image normalization; this applies to all descriptors. For example, the GIST method has a *CR* of 47.8% without image normalization, and a *CR* of 70.5%

Table 6.2: Classification rates of scene categorization algorithms on distorted images of the 15-scene database.

Feature descriptor	*Without image normalization* (%)	*With proposed normalization* (%)
Places-CNN	48.9 ± 2.0	85.6 ± 1.1
SIFT-ScSPM	55.4 ± 1.8	75.9 ± 1.3
SIFT-SPM	39.7 ± 1.2	70.1 ± 4.0
GIST	47.8 ± 2.0	70.5 ± 1.7
LBP	18.1 ± 9.8	62.7 ± 1.0
Moment-ScSPM	30.0 ± 2.2	70.2 ± 1.4
Scattering-ScSPM	55.9 ± 3.2	80.2 ± 1.8
Places-CNN-GIST	49.0 ± 2.2	86.5 ± 1.1

with image normalization. The SIFT-ScSPM algorithm has a *CR* of 55.4% without image normalization, and a *CR* of 75.9% with image normalization. The moment-ScSPM algorithm has a *CR* of 30.0% without image normalization, and a *CR* of 70.2% with image normalization. It is worth noting that combining the moment-ScSPM algorithm and the proposed image normalization leads to a higher *CR* on the distorted database (70.2%) than on the original database (61.6%). The scattering-ScSPM algorithm achieves the higher *CR* (80.2%) on the normalized images than SIFT-ScSPM (75.9%). The Places-CNN-GIST algorithm achieves the highest *CR* (86.5%) on the normalized images among the compared descriptors.

Table 6.3: Classification rates of scene categorization algorithms on distorted+cropped images of the 15-scene database. The image cropping rate is 0.2.

Feature descriptor	*Without image normalization* (%)	*With proposed normalization* (%)
Places-CNN	56.2 ± 2.4	84.8 ± 1.1
SIFT-ScSPM	51.1 ± 3.2	72.3 ± 2.1
SIFT-SPM	51.1 ± 3.2	61.3 ± 3.5
GIST	34.1 ± 2.5	62.2 ± 3.0
LBP	18.2 ± 8.0	48.2 ± 5.0
Moment-ScSPM	41.4 ± 0.9	68.6 ± 2.1
Scattering-ScSPM	51.6 ± 0.9	74.0 ± 0.9
Places-CNN-GIST	57.5 ± 2.2	84.9 ± 1.7

Next, we evaluate the feature extractors on the distorted and cropped images.

The classification rates for this experiment are presented in Table 6.3. Without using the proposed normalization, the *CRs* of distorted and cropped images (Table 6.3, Column 2) are significantly lower than the *CRs* on the original images (Table 6.1). After applying the proposed normalization, the *CRs* on the distorted and cropped images (Table 6.3, Column 3) are increased for all features compared to the *CRs* without normalization (Table 6.3, Column 2). The Places-CNN-GIST has the highest classification rate (84.9%) among all descriptors on the affine-distorted and cropped images. This result means the Places-CNN-GIST is more robust to image cropping.

Table 6.4: Classification rates of scene categorization algorithms on distorted+noise images of the 15-scene database. The image noise density is 0.1.

Feature descriptor	*Without image normalization* (%)	*With proposed normalization* (%)
Places-CNN	40.4 ± 2.5	71.4 ± 1.9
SIFT-ScSPM	38.1 ± 1.4	63.1 ± 2.7
SIFT-SPM	38.1 ± 2.0	64.2 ± 2.4
GIST	26.6 ± 2.0	55.4 ± 4.2
LBP	14.2 ± 6.1	35.9 ± 10.9
Moment-ScSPM	33.3 ± 1.6	61.4 ± 2.2
Scattering-ScSPM	32.1 ± 2.0	55.5 ± 1.9
Places-CNN-GIST	39.7 ± 1.9	71.6 ± 1.9

Lastly, we evaluate the feature extractors on the distorted and noisy images. In this experiment, Gaussian noise with varied standard deviation was added to the affine distorted images. The classification rates for this experiment are presented in Table 6.4. Without using the proposed normalization, the *CRs* for distorted and noisy images (Table 6.4, Column 2) are lower than the *CRs* for original images (Table 6.1). For the distorted and noisy images, the *CRs* obtained with the proposed normalization (Table 6.4, Column 3) are higher than the *CRs* obtained without image normalization (Table 6.4, Column 2). Furthermore, the Places-CNN-GIST

Table 6.5: Classification rates of scene categorization algorithms on distorted images of the 8-scene database.

Feature descriptor	*Without image normalization* (%)	*With proposed normalization* (%)
Places-CNN	60.1 ± 1.8	91.2 ± 1.3
SIFT-ScSPM	71.6 ± 3.6	86.1 ± 3.0
SIFT-SPM	51.6 ± 2.1	85.1 ± 2.9
GIST	62.4 ± 3.7	82.7 ± 3.3
LBP	22.4 ± 8.3	76.7 ± 4.0
Moment-ScSPM	43.0 ± 3.3	84.2 ± 1.6
Scattering-ScSPM	72.8 ± 3.2	88.1 ± 2.5
Places-CNN-GIST	61.4 ± 3.7	91.6 ± 1.1

Table 6.6: Classification rates of scene categorization algorithms on distorted images of the 67-indoor-scene database.

Feature descriptor	*Without image normalization* (%)	*With proposed normalization* (%)
Places-CNN	20.2 ± 0.9	57.7 ± 2.7
SIFT-ScSPM	12.2 ± 0.5	33.9 ± 2.1
SIFT-SPM	6.5 ± 0.6	21.9 ± 1.8
GIST	8.2 ± 0.4	23.9 ± 2.8
LBP	5.9 ± 1.0	13.9 ± 2.6
Moment-ScSPM	7.6 ± 0.5	18.5 ± 1.0
Scattering-ScSPM	14.8 ± 0.7	37.3 ± 2.8
Places-CNN-GIST	20.5 ± 0.9	57.7 ± 2.9

Table 6.7: Classification rates of scene categorization algorithms on distorted images of the SUN397 database.

Feature descriptor	*Without image normalization* (%)	*With proposed normalization* (%)
Places-CNN	11.0 ± 0.2	39.0 ± 0.4
SIFT-ScSPM	3.1 ± 0.7	6.0 ± 0.4
SIFT-SPM	1.2 ± 0.2	2.0 ± 0.4
GIST	1.1 ± 0.2	2.7 ± 0.3
LBP	1.3 ± 0.3	1.7 ± 0.3
Moment-ScSPM	1.4 ± 0.2	2.4 ± 0.3
Scattering-ScSPM	3.3 ± 0.6	13.8 ± 0.7
Places-CNN-GIST	11.2 ± 0.2	39.9 ± 0.4

has the highest classification rate (71.6%) among the ten descriptors, which means the Places-CNN-GIST is more robust to image noise.

6.3.3 Analysis of scene categorization on multiple data sets under affine deformations

We also analyze the scene categorization on three other data sets: the 8-scene data set [227] and the 67-indoor-scene data set [121], and the SUN397 data set [211]. Table 6.5 to 6.7 present the *CRs* on the affine-distorted images of the three data sets.

On the affine-distorted 8-scene data set and without image normalization, the mean *CR*, averaged over all image features, is 55.7%. With the proposed normalization, the mean *CR* increases to 85.7%. For the Places-CNN-GIST feature, the *CR* reaches 91.6%.

On the affine-distorted 67-indoor-scene data set, the proposed image normalization algorithm improves the average *CRs* of the 8 descriptors from 12.0% to 33.1%. The Places-CNN achieves the highest classification rate among the 8 descriptors. Combining GIST features with Places-CNN does not improve the classification performance for indoor images.

On the affine-distorted SUN397 data set, the proposed image normalization algorithm improves the average *CRs* of the 8 descriptors from 4.2% to 13.4%. The Places-CNN-GIST still achieves the highest classification rate (39.9%) among the ten descriptors. Combining GIST features with Places-CNN improves the classification performance for the SUN397 data set. The *CRs* of the Scattering-ScSPM descriptor is also much higher than the *CRs* of other hand-designed features. The experimental results on the 8-scene data set, the 67-indoor-scene data set, and the SUN397 data set are consistent with results on the 15-scene data set, presented in Section 6.3.2.

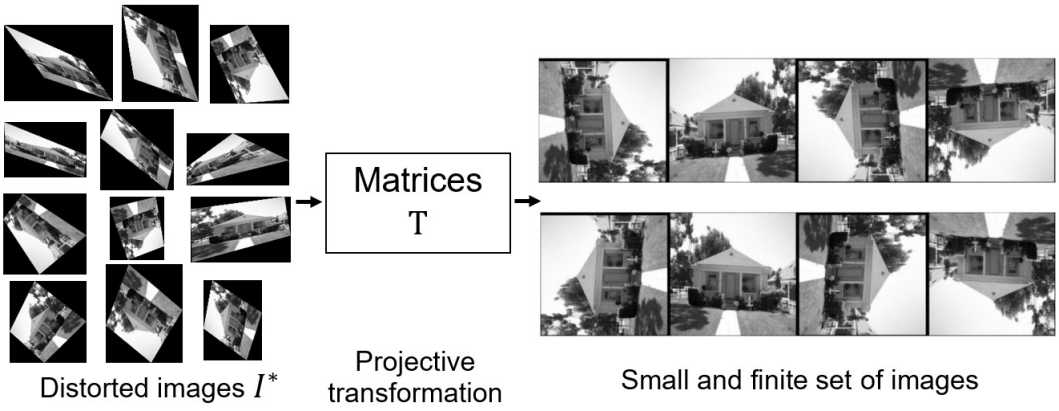


Figure 6.2: Image normalization for projective-distortions. Distorted images are mapped to a small set of normalized images using transformation matrices T.

6.3.4 Analysis of scene categorization under projective distortions

We also evaluated the effects of the image normalization for projective distortions on an image classification task. Figure 6.2 shows how the projective normalization is performed on the projective-distorted images. In this experiment, we evaluated four feature vectors: GIST [32], SIFT-ScSPM [65], Places-CNN [52], and Place-CNN-GIST on the 15-scene, 67-indoor-scene, and SUN397 data sets. The *one-versus-all* SVM with the RBF kernel was applied to classify the extracted feature vectors. For normalized images obtained from the original data sets, the classification rates were determined from the first normalized image that has the same orientation with the original image. For normalized images obtained from the projective-distorted data sets, the classification rates were determined from the average classification score of the 8 normalized images.

On the original 15-scene data set, results in Table 6.8 indicate that the proposed normalization improves the classification rate (*CR*) of the GIST descriptor. The GIST feature is robust to the re-sampling artifact and image blurring produced by image transformation. The *CR* of SIFT-ScSPM, Places-CNN, and Places-CNN-

GIST with image normalization is slightly lower than the results without image normalization on the original images. That means SIFT-ScSPM, Places-CNN, and Places-CNN-GIST are sensitive to the re-sampling artifact and image blurring

Table 6.8: Scene categorization results on the 15-scene data set under projective distortions.

Feature descriptor	*Images*	*Without image normalization* (%)	*With proposed normalization* (%)
GIST	original images	74.8 ± 1.8	75.4 ± 1.2
SIFT-ScSPM	original images	83.8 ± 1.4	82.3 ± 0.5
Places-CNN	original images	91.8 ± 1.2	90.8 ± 0.9
Places-CNN-GIST	original images	92.4 ± 0.9	91.9 ± 1.0
GIST	distorted images	47.8 ± 2.0	70.5 ± 1.7
SIFT-ScSPM	distorted images	55.4 ± 1.8	75.9 ± 1.3
Places-CNN	distorted images	48.8 ± 2.1	82.4 ± 0.8
Places-CNN-GIST	distorted images	49.0 ± 2.0	83.0 ± 1.1

Table 6.9: Scene categorization results on the 67-indoor-scene data set under projective distortions.

Feature descriptor	*Images*	*Without image normalization* (%)	*With proposed normalization* (%)
GIST	original images	30.9 ± 0.9	25.5 ± 1.7
SIFT-ScSPM	original images	45.6 ± 1.0	35.2 ± 1.0
Places-CNN	original images	68.2 ± 0.4	68.3 ± 1.8
Places-CNN-GIST	original images	70.2 ± 1.0	68.6 ± 1.6
GIST	distorted images	6.1 ± 0.3	15.4 ± 1.2
SIFT-ScSPM	distorted images	7.8 ± 0.6	17.0 ± 1.1
Places-CNN	distorted images	25.4 ± 1.7	43.2 ± 2.3
Places-CNN-GIST	distorted images	25.5 ± 1.9	42.9 ± 1.9

Table 6.10: Scene categorization results on the SUN397 data set under projective distortions.

Feature descriptor	*Images*	*Without image normalization* (%)	*With proposed normalization* (%)
GIST	original images	15.2 ± 0.2	14.7 ± 0.2
SIFT-ScSPM	original images	29.1 ± 0.4	20.6 ± 0.2
Places-CNN	original images	54.3 ± 0.1	54.7 ± 0.2
Places-CNN-GIST	original images	56.2 ± 0.2	55.0 ± 0.2
GIST	distorted images	0.9 ± 0.0	5.4 ± 0.4
SIFT-ScSPM	distorted images	1.5 ± 0.1	3.3 ± 0.1
Places-CNN	distorted images	10.7 ± 0.1	31.4 ± 0.4
Places-CNN-GIST	distorted images	10.7 ± 0.1	31.3 ± 0.5

produced by image transformation.

On the projective-distorted 15-scene data set, the results shown in the last two rows of Table 6.8 indicate that for all descriptors (GIST, SIFT-ScSPM, Places-CNN, and Places-CNN-GIST), image normalization leads to higher classification rates than without image normalization.

On the original 67-indoor-scene data set, results in Table 6.9 indicate that the proposed normalization improves the classification rate (*CR*) of the Places-CNN descriptor. The Places-CNN feature is robust to the indoor-scene images. The *CR* of GIST, SIFT-ScSPM, and Places-CNN-GIST with image normalization is slightly lower than the results without image normalization on the original images. That means SIFT-ScSPM, Places-CNN, and Places-CNN-GIST are not robust to extract indoor features. For indoor images, the details of objects carry more information for scene categorization.

On the projective-distorted 67-indoor-scene data set, the proposed image normalization algorithm improves the average *CRs* of the 4 descriptors from 16.2% to 29.6%. The Places-CNN achieves the highest classification rate among the 4 descriptors. Combining GIST features with Places-CNN does not improve the classification performance for indoor images.

On the original SUN397 data set, results in Table 6.10 indicate that the proposed normalization also improves the classification rate (*CR*) of the Places-CNN descriptor. The GIST feature is robust to large-scale image categories. The *CR* of GIST, SIFT-ScSPM, and Places-CNN-GIST with image normalization is slightly lower than the results without image normalization on the original images.

On the projective-distorted SUN397 data set, the proposed image normaliza-

tion algorithm improves the average *CRs* of the 4 descriptors from 6.0% to 17.9%. The Places-CNN descriptor achieves the highest classification rate (31.4%) among the 4 descriptors with image normalization. The classification rate of Places-CNN-GIST is 31.4%. Combining GIST features with Places-CNN does not improve the classification performance for the projective-distorted SUN397 data set.

6.4 Chapter summary

In this chapter, a new approach for scene categorization that is invariant to geometric transformations is presented. In our experiments, the effects of the proposed image normalization on the visual descriptors are analyzed using several public data sets. The experimental results indicate that even state-of-the-art descriptors suffer from image distortions caused by affine transformations, image cropping, and noise. The proposed image normalization methods increases the scene categorization accuracy of existing descriptors significantly. Among the 8 evaluated methods, the combination of the proposed image normalization and the Places-CNN-GIST descriptor achieves the highest scene categorization accuracy under affine distortions. The combination of the proposed image normalization and the Places-CNN descriptor achieves better scene categorization accuracy under projective distortions.

Chapter 7

Conclusion

Chapter contents

Scene categorization can be used to provide cues about objects and actions, detect abnormal events in public places, sense dangerous situations, and search for images and video; therefore, it is highly useful for applications in surveillance, navigation, and multimedia. The existing scene categorization methods are not fully-invariant to viewing angles. In this thesis, we propose image normalization methods for affine and projective deformations. Our approach allows geometric-invariant features to be extracted after image normalization, thereby reducing the complexity of scene classifiers and the cost of classifier training. It is a step towards developing a view-invariant scene categorization system.

This chapter is organized as follows: Section 7.1 summarizes the research contributions of the thesis; Section 7.2 outlines the future work and research directions; Section 7.3 draws conclusion for the thesis.

7.1 Research summary

The research activities have been documented in several chapters of the thesis. They are summarized as follows.

- We provided the literature review of human visual system and computational visual descriptors for scene categorization from both methodological and experimental perspectives. The state-of-the-art visual descriptors for scene categorization were analyzed. We also studied existing methods to achieve geometric-invariant features. We reviewed and compared the existing image normalization methods for both affine and projective deformations.

- We proposed an image normalization method for affine deformations. The normalization matrices T are found by solving a constrained optimization problem involving low-order image moments. We presented experimental results to compare the image normalization accuracies and to study the separability on several benchmark data sets. We also analyzed the effects of image noise and image cropping on the image normalization.

- We proposed an image normalization method for projective deformations. The proposed normalization method produces the same set of normalized images for projective distorted images. Our experiments showed that the proposed method is more accurate than the existing normalization methods. When projective deformations are present in the input image, our method allows invariant visual features to be extracted for image recognition.

- We developed a scene categorization system under geometric deformations. We extracted different visual features after applying the proposed image normalization methods on input images, to produce geometric-invariant scene categorization. This system includes three stages: image normalization, feature extraction, and classification. Among the evaluated methods, the combination of the proposed image normalization and the PlaceCNN+GIST descriptor achieves the highest scene categorization accuracy under affine deformations.

7.2 Future work

Possible research directions can be summarized as follows:

- Develop dynamic scene categorization system. Most existing studies on gist recognition have been concerned with static scenes, which is the focus of this thesis. In recent years, gist recognition of dynamic scenes has attracted the attention of researchers, and therefore, the extension of this thesis to dynamic scenes would be invaluable.

- Develop view-invariant descriptors based on the proposed image normalization algorithm. The proposed normalization for affine deformations achieves fully-affine invariants. The moment propositions used in our normalization method are proved. It is worth to develop a view-invariant descriptor based on the moment constraints proposed in this thesis.

- Improve the image normalization method for projective deformations. The proposed method for projective deformations is based on the experimen-

tal observation. Mathematical analysis of this algorithm warrants further investigation.

- Apply the proposed image normalization methods on other image classification tasks, such as face recognition, handwritten digit recognition, and texture recognition.

7.3 Conclusion

This thesis has presented a scene categorization system that is invariant to geometric deformations. The geometric-invariant scene categorization system consists with three main parts: image normalization, invariant feature extraction, and classification. The proposed image normalization method for affine deformations is compared with five existing normalization methods. The results show that the proposed image normalization is more robust to affine distortions, image cropping, and image noise. The class separability of images on several benchmark data sets is also increased by applying the proposed normalization method. The proposed image normalization method for projective deformations is also compared with two affine-normalization methods based on image moments and a projective-normalization method based on image rank minimization. The results indicate the proposed normalization is more efficient and accurate than the existing normalization methods for projective deformations. The state-of-the-art descriptors for scene categorization are reviewed and evaluated in this thesis. We also extract different visual features after applying the proposed image normalization method on input images, to produce geometric-invariant scene categorization. The combination of the proposed image normalization and the PlaceCNN+GIST

descriptor achieves the highest scene categorization accuracy under affine deformations. Furthermore, extracting PlaceCNN features after applying the proposed image normalization also achieves the highest scene categorization accuracy under projective deformations.

Chapter 8

Appendix

8.1 Proof of Proposition 1

Under the affine transformation T, the output image $I'(x', y')$ and the input image $I(x, y)$ are related as follows:

$$\begin{cases} x' & = xt_1 + yt_2 + t_3, \\ y' & = xt_4 + yt_5 + t_6, \\ I'(x', y') & = I(x, y). \end{cases} \tag{8.1}$$

First, we will prove Eq. (4.7). The geometric moment $m'_{p,q}$ of the output image is

$$\begin{aligned} m'_{p,q} &= \iint_{\Gamma'} (x')^p (y')^q I'(x', y')\, d(x')\, d(y') \\ &= \iint_{\Gamma} (xt_1 + yt_2 + t_3)^p\, (xt_4 + yt_5 + t_6)^q \times I(x, y) \det(J)\, dx\, dy. \end{aligned} \tag{8.2}$$

Here, J is the Jacobian matrix that is defined as

$$J = \begin{pmatrix} \frac{\partial(x')}{\partial x} & \frac{\partial(y')}{\partial x} \\ \frac{\partial(x')}{\partial y} & \frac{\partial(y')}{\partial y} \end{pmatrix} = \begin{pmatrix} t_1 & t_2 \\ t_4 & t_5 \end{pmatrix}. \tag{8.3}$$

Hence, $\det(J) = t_1 t_5 - t_2 t_4$. The output geometric moment can then be expressed as

$$
\begin{aligned}
& m'_{p,q} \\
&= \det(J) \iint_{\Gamma} (xt_1 + yt_2 + t_3)^p (xt_4 + yt_5 + t_6)^q I(x,y)\, dx\, dy \\
&= \det(J) \iint_{\Gamma} \left[\sum_{(i,j)\in S_p} \binom{p}{i,j,p-i-j} x^i t_1^i y^j t_2^j t_3^{p-i-j} \right] \times \left[\sum_{(k,l)\in S_q} \binom{q}{k,l,q-k-l} x^k t_4^k y^l t_5^l t_6^{q-k-l} \right] \times I(x,y)\, dx\, dy \\
&= \det(J) \iint_{\Gamma} \sum_{(i,j)\in S_p} \sum_{(k,l)\in S_q} \binom{p}{i,j,p-i-j} \binom{q}{k,l,q-k-l} \times t_1^i\, t_2^j\, t_4^k\, t_5^l\, t_3^{p-i-j}\, t_6^{q-k-l} \times x^{i+k} y^{j+l} I(x,y)\, dx\, dy \\
&= \det(J) \sum_{(i,j)\in S_p} \sum_{(k,l)\in S_q} \binom{p}{i,j,p-i-j} \binom{q}{k,l,q-k-l} \times t_1^i\, t_2^j\, t_4^k\, t_5^l\, t_3^{p-i-j}\, t_6^{q-k-l} \times \iint_{\Gamma} x^{i+k} y^{j+l} I(x,y)\, dx\, dy \\
&= \det(J) \sum_{(i,j)\in S_p} \sum_{(k,l)\in S_q} \binom{p}{i,j,p-i-j} \binom{q}{k,l,q-k-l} \times t_1^i\, t_2^j\, t_4^k\, t_5^l\, t_3^{p-i-j}\, t_6^{q-k-l} \times m_{i+k,j+l}.
\end{aligned}
\tag{8.4}
$$

Next, we will prove Eq. (4.8). Because $m'_{0,0} = \det(J) m_{0,0}$, the formula for the normalized geometric moment is obtained as

$$
\begin{aligned}
\nu'_{p,q} &= \frac{m'_{p,q}}{m'_{0,0}} \\
&= \sum_{(i,j)\in S_p} \sum_{(k,l)\in S_q} \binom{p}{i,j,p-i-j} \binom{q}{k,l,q-k-l} \times t_1^i\, t_2^j\, t_4^k\, t_5^l\, t_3^{p-i-j}\, t_6^{q-k-l} \times \nu_{i+k,j+l}
\end{aligned}
\tag{8.5}
$$

Now, we will prove Eq. (4.9). First, it can be shown that under the affine transformation, the centroids $(\overline{x'}, \overline{y'})$ of the output image are related to the centroids $(\overline{x}, \overline{y})$ of the input image as

$$
\overline{x'} = \overline{x} t_1 + \overline{y} t_2 + t_3,
$$

$$
\overline{y'} = \overline{x} t_4 + \overline{y} t_5 + t_6.
$$

Then, the central moment can be derived as

$$\begin{aligned}
\mu'_{p,q} &= \iint_{\Gamma'} (x'-\overline{x'})^p (y'-\overline{y'})^q I'(x',y')\, d(x')\, d(y') \\
&= \iint_{\Gamma} [(xt_1+yt_2+t_3)-(\overline{x}t_1+\overline{y}t_2+t_3)]^p \times [(xt_4+yt_5+t_6)-(\overline{x}t_4+\overline{y}t_5+t_6)]^q \times I(x,y)\det(J)\, dx\, dy \\
&= \iint_{\Gamma} [(x-\overline{x})t_1+(y-\overline{y})t_2]^p [(x-\overline{x})t_4+(y-\overline{y})t_5]^q \times I(x,y)\det(J)\, dx\, dy \\
&= \det(J) \iint_{\Gamma} \left[\sum_{i=0}^{p} \binom{p}{i} (x-\overline{x})^i t_1^i\, (y-\overline{y})^{p-i} t_2^{p-i}\right] \times \left[\sum_{j=0}^{q} \binom{q}{j} (x-\overline{x})^j t_4^j (y-\overline{y})^{q-j} t_5^{q-j}\right] \times I(x,y)\, dx\, dy \\
&= \det(J) \sum_{i=0}^{p}\sum_{j=0}^{q} \binom{p}{i}\binom{q}{j} t_1^i\, t_4^j\, t_2^{p-i}\, t_5^{q-j} \times \iint_{\Gamma} (x-\overline{x})^{i+j} (y-\overline{y})^{p+q-i-j}\, I(x,y)\, dx\, dy \\
&= \det(J) \sum_{i=0}^{p}\sum_{j=0}^{q} \binom{p}{i}\binom{q}{j} t_1^i\, t_2^{p-i}\, t_4^j\, t_5^{q-j}\, \mu_{i+j,p+q-i-j}
\end{aligned} \tag{8.6}$$

Finally, we will prove Eq. (4.10). Because $\mu'_{0,0} = \det(J)\mu_{0,0}$, the normalized central moment is obtained as

$$\begin{aligned}
\eta'_{p,q} &= \frac{\mu'_{p,q}}{\mu'_{0,0}} \\
&= \frac{\mu'_{p,q}}{\det(J)\mu_{0,0}} \\
&= \sum_{i=0}^{p}\sum_{j=0}^{q} \binom{p}{i}\binom{q}{j} t_1^i\, t_2^{p-i}\, t_4^j\, t_5^{q-j}\, \eta_{i+j,p+q-i-j}.
\end{aligned} \tag{8.7}$$

Proposition 1 is now proved.

8.2 Proof of proposition 3

Consider an input image I and a distorted image I_{d}, which is obtained by applying an arbitrary affine transformation, represented by matrix T_{d}, on image I:

$$I \xrightarrow{T_{\mathrm{d}}} I_{\mathrm{d}}. \tag{8.8}$$

For image I, the proposed affine-normalization algorithm (Table 4.2) generates a set of 8 transformation matrices $\hat{\mathcal{T}} = \{\hat{T}_1, \hat{T}_2, ..., \hat{T}_8\}$. Applying these transformation matrices on image I will produce a set of 8 normalized images $\hat{\mathcal{I}} = \{\hat{I}_1, \hat{I}_2, ..., \hat{I}_8\}$:

$$I \xrightarrow{\hat{T}_i} \hat{I}_i, \text{ where } i = 1,2,...,8. \tag{8.9}$$

For image I_d, the proposed affine-normalization algorithm (Table 4.2) produces a set of 8 transformation matrices $\hat{\mathcal{T}}^* = \{\hat{T}_1^*, \hat{T}_2^*, ..., \hat{T}_8^*\}$. Applying these transformation matrices on image I_d will give a set of 8 normalized images $\hat{\mathcal{I}}^* = \{\hat{I}_1^*, \hat{I}_2^*, ..., \hat{I}_8^*\}$:

$$I_\mathrm{d} \xrightarrow{\hat{T}_i^*} \hat{I}_i^*, \text{ where } i = 1, 2, ..., 8. \tag{8.10}$$

Combining (8.8) and (8.10), we obtain:

$$I \xrightarrow{T_\mathrm{d}} I_\mathrm{d} \xrightarrow{\hat{T}_i^*} \hat{I}_i^*, \tag{8.11}$$

or

$$I \xrightarrow{T_\mathrm{d}\hat{T}_i^*} \hat{I}_i^*. \tag{8.12}$$

Equation (8.12) means that if transformation matrix $\hat{T}_i^*$ normalizes image I_d, then transformation matrix $T_\mathrm{d}\hat{T}_i^*$ normalizes image I, and vice versa.

According to Proposition 2, for *moment-normalizable* image I, there are exactly 8 transformation matrices that normalize I. Therefore, the two following sets of transformation matrices must be identical:

$$\begin{cases} \hat{\mathcal{T}} & = \{\hat{T}_1, \hat{T}_2, ..., \hat{T}_8\}, \\ T_\mathrm{d}\mathcal{T}^* & = \{T_\mathrm{d}\hat{T}_1^*, T_\mathrm{d}\hat{T}_2^*, ..., T_\mathrm{d}\hat{T}_8^*\}. \end{cases} \tag{8.13}$$

In other words, the set of normalized images for input image I and the set of normalized images for input image I_d are identical: $\hat{\mathcal{I}} \equiv \hat{\mathcal{I}}^*$. Proposition 3 is now proved.

References

[1] C. Pavlopoulou and S. X. Yu, "Indoor-outdoor classification with human accuracies: Image or edge gist?" in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2010, pp. 41–47.

[2] E. B. Goldstein, *Encyclopedia of Perception*. Thousand Oaks, California: SAGE Publications, 2010.

[3] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 2169–2178.

[4] J. Xiao, K. A. Ehinger, J. Hays, A. Torralba, and A. Oliva, "SUN database: Exploring a large collection of scene categories," *International Journal of Computer Vision*, pp. 1–20, 2014.

[5] J. Krapac, J. Verbeek, and F. Jurie, "Modeling spatial layout with fisher vectors for image categorization," in *Proceedings of IEEE International Conference on Computer Vision*, 2011, pp. 1487–1494.

[6] J. Yao, S. Fidler, and R. Urtasun, "Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 702–709.

[7] B. Zhao and E. Xing, "Hierarchical feature hashing for fast dimensionality reduction," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2051–2058.

[8] Z. Zhang, M. Li, K. Huang, and T. Tan, "Robust automated ground plane rectification based on moving vehicles for traffic scene surveillance," in *Proceedings of IEEE International Conference on Image Processing*, 2008, pp. 1364–1367.

[9] D. Gowsikhaa, S. Abirami, and R. Baskaran, "Automated human behavior analysis from surveillance videos: a survey," *Artificial Intelligence Review*, vol. 42, no. 4, pp. 747–765, 2014.

[10] C. K. Chang, C. Siagian, and L. Itti, "Mobile robot vision navigation and localization using gist and saliency," in *Proceedings of IEEE-RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 4147–4154.

[11] X. Jia, A. G. Schwing, and R. Urtasun, "Tell me what you see and i will show you where it is," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3190–3197.

[12] A. Khosla, A. Byoungkwon, J. J. Lim, and A. Torralba, "Looking beyond

the visible scene," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3710–3717.

[13] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendon-Mancha, "Visual simultaneous localization and mapping: a survey," *Artificial Intelligence Review*, vol. 43, no. 1, pp. 55–81, 2015.

[14] A. Chella, M. Frixione, and S. Gaglio, "Conceptual spaces for computer vision representations," *Artificial Intelligence Review*, vol. 16, no. 2, pp. 137–152, 2001.

[15] E. Maggio and A. Cavallaro, "Learning scene context for multiple object tracking," *IEEE Transactions on Image Processing*, vol. 18, no. 8, pp. 1873–1884, 2009.

[16] Z. Huang, H. Huang, W. Zhang, and L. Hou, "Face recognition using the global image features based on scene gist," *Journal of Information and Computational Science*, vol. 5, no. 2, pp. 919–928, 2008.

[17] H. Zhou, T. Hermans, A. V. Karandikar, and J. M. Rehg, "Movie genre classification via scene categorization," in *Proceedings of the International Conference on Multimedia*, 2010, pp. 747–750.

[18] Q. Dai, R. W. Zhao, Z. Wu, X. Wang, Z. Gu, W. Wu, and Y. G. Jiang, "Detecting violent scenes and affective impact in movies with deep learning," in *Proceedings of MediaEval Workshop*, 2015, pp. 1–3.

[19] R. Snowden, P. Thompson, and T. Troscianko, *Basic vision: An introduction to visual perception*. New York: Oxford University Press, 2004.

[20] M. C. Potter, "Meaning in visual search," *Science*, vol. 187, no. 4180, pp. 965–966, 1975.

[21] L. W. Renninger and J. Malik, "When is scene recognition just texture recognition?" *Vision Research*, vol. 44, pp. 2301–2311, 2003.

[22] M. S. Castelhano and J. M. Henderson, "The influence of color on the perception of scene gist," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 34, no. 3, pp. 660–675, 2008.

[23] M. S. Castelhano and C. Heaven, "The relative contribution of scene context and target features to visual search in scenes," *Attention, Perception, and Psychophysics*, vol. 72, no. 5, pp. 1283–1297, 2010.

[24] L. Fei-Fei, A. Lyer, C. Koch, and P. Perona, "What do we perceive in a glance of a real-world scene?" *Journal of Vision*, vol. 7, no. 1, pp. 1–29, 2007.

[25] L. C. Loschky, A. Sethi, D. J. Simons, T. N. Pydimarri, D. Ochs, and J. L. Corbeille, "The importance of information localization in scene gist recognition," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 33, no. 6, pp. 1431–1450, 2007.

[26] L. Wei, N. Sang, and Y. Wang, "A biologically inspired object-based visual attention model," *Artificial Intelligence Review*, vol. 34, no. 2, pp. 109–119, 2010.

[27] D. B. Walther, B. Chai, E. Caddigan, D. M. Beck, and L. Fei-Fei, "Simple line drawings suffice for functional MRI decoding of natural scene categories,"

Proceedings of the National Academy of Sciences, vol. 108, no. 23, pp. 9661–9666, 2011.

[28] K. W. Lee and H. Choo, "A critical review of selective attention: an interdisciplinary perspective," *Artificial Intelligence Review*, vol. 40, no. 1, pp. 27–50, 2013.

[29] D. Linsley and S. P. MacEvoy, "Evidence for participation by object-selective visual cortex in scene category judgments," *Journal of Vision*, vol. 14, no. 9, pp. 1–17, 2014.

[30] G. L. Malcolm, A. Nuthmann, and P. G. Schyns, "Beyond gist: Strategic and incremental information accumulation for scene categorization," *Psychological Science*, vol. 25, no. 5, pp. 1087–1097, 2014.

[31] A. Oliva, "Diagnostic colors mediate scene recognition," *Cognitive Psychology*, vol. 41, pp. 176–210, 2000.

[32] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.

[33] M. Douze, H. Jegou, H. Sandhawalia, L. Amsaleg, and C. Schmid, "Evaluation of gist descriptors for web-scale image search," in *Proceedings of International Conference on Image and Video Retrieval*, 2009, pp. 140–147.

[34] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[35] K. E. A. Van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1582–1596, 2010.

[36] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[37] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 886–893.

[38] T. Ojala, M. Pietikainen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[39] C. Rasche and C. Koch, "Recognizing the gist of a visual scene: possible perceptual and neural mechanisms," *Neurocomputing*, vol. 4446, pp. 979–984, 2002.

[40] R. Epstein, A. Harris, D. Stanley, and N. Kanwisher, "The parahippocampal place area: recognition, navigation, or encoding?" *Neuron*, vol. 23, pp. 115–125, 1999.

[41] J. J. Gibson, "Adaptation, after-effect, and contrast in the perception of tilted lines. II. simultaneous contrast and the areal restriction of the after-effect." *Journal of Experimental Psychology*, vol. 20, no. 6, pp. 533–569, 1937.

[42] NASA, "Astronaut vision changes offer opportunity for more research,"

2013. [Online]. Available: http://www.nasa.gov/mission_pages/station/research/news/Astronaut_Vision.html

[43] T. H. Mader, C. R. Gibson, A. F. Pass, L. A. Kramer, A. G. Lee, J. Fogarty, W. J. Tarver, J. P. Dervay, D. R. Hamilton, A. Sargsyan, J. L. Phillips, D. Tran, W. Lipsky, J. Choi, C. Stern, R. Kuyumjian, and J. D. Polk, "Optic disc edema, globe flattening, choroidal folds, and hyperopic shifts observed in astronauts after long-duration space flight," *Ophthalmology*, vol. 118, no. 10, pp. 2058 – 2069, 2011.

[44] V. Mountcastle, "An organizing principle for cerebral function: the unit model and the distributed system," in *The Mindful Brain*. MIT Press, 1978.

[45] J. Hawkins and S. Blakslee, *On intelligence*, 1st ed. New York: Times Books, 2004.

[46] Hawkins, J., "Grok solution," 2013. [Online]. Available: https://www.groksolutions.com

[47] D. Song and D. Tao, "Biologically inspired feature manifold for scene classification," *IEEE Transactions on Image Processing*, vol. 19, no. 1, pp. 174–184, 2010.

[48] M. Juneja, A. Vedaldi, C. Jawahar, and A. Zisserman, "Blocks that shout: Distinctive parts for scene classification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 923–930.

[49] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: an incremental Bayesian approach tested on

101 object categories," *Journal Computer Vision and Image Understanding*, vol. 106, no. 1, pp. 59–70, 2007.

[50] J. Wu and J. M. Rehg, "CENTRIST: A visual descriptor for scene categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1489–1501, 2011.

[51] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[52] A. Krizhevsky, I. Sutskever, and E. H. Geoffrey, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[53] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "DeCAF: A deep convolutional activation feature for generic visual recognition," in *Proceedings of International Conference on Machine Learning*, 2014, pp. 647–655.

[54] S. Maji, A. C. Berg, and J. Malik, "Classification using intersection kernel support vector machines is efficient," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[55] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nature Neuroscience*, vol. 2, pp. 1019–1025, 1999.

[56] D. Huang, C. Zhu, Y. Wang, and L. Chen, "HSOG: A novel local image descriptor based on histograms of the second-order gradients," *IEEE Transactions on Image Processing*, vol. 23, no. 11, pp. 4680–4695, 2014.

[57] T. Ahonen, J. Matas, C. He, and M. Pietikainen, "Rotation invariant image description with local binary pattern histogram Fourier features," in *Proceedings of Scandinavian Conference on Image Analysis*, 2009, pp. 61–70.

[58] S. Clinchant, G. Csurka, F. Perronnin, and J. Renders, "XRCEs participation to ImagEval," in *Proceedings of Workshop at Content Visualization and Intermedia Representations*, 2007, pp. 1–8.

[59] L. Li, H. Su, Y. Lim, and L. Fei-Fei, "Object bank: A high-level image representation for scene classification & semantic feature sparsification," in *Advances in Neural Information Processing Systems 23*, J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds. Curran Associates, Inc., 2010, pp. 1378–1386.

[60] X. Qian, X. S. Hua, P. Chen, and L. Ke, "PLBP: An effective local binary patterns texture descriptor with pyramid representation," *Pattern Recognition*, vol. 44, no. 1011, pp. 2502–2515, 2011.

[61] B. Scholkopf, K. K. Sung, C. J. C. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik, "Comparing support vector machines with gaussian kernels to radial basis function classifiers," *Computer Vision and Image Understanding*, vol. 45, no. 11, pp. 2758–2765, 1997.

[62] S. Morikawa and T. Shibata, "Scene image recognition based on the sequence

of local image vectors represented by oriented edges," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2012, pp. 1313 – 1316.

[63] J. Sanchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the fisher vector: Theory and practice," *International Journal of Computer Vision*, vol. 105, no. 3, pp. 222–245, 2013.

[64] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3360–3367.

[65] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1794–1801.

[66] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[67] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[68] T. Serre, L. Wolf, and T. Poggio, "Object recognition with features inspired by visual cortex," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 994–1000.

[69] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using Places database," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 487–495.

[70] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: Integrated recognition, localization and detection using convolutional networks," in *Proceedings of International Conference on Learning Representations*, 2014, pp. 1–15.

[71] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: Fast retina keypoint," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 510–517.

[72] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *Proceedings of IEEE International Conference on Computer Vision*, 2011, pp. 2548–2555.

[73] Y. Ke and R. Sukthankar, "PCA-SIFT: a more distinctive representation for local image descriptors," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2004, pp. 506–513.

[74] E. Hadjidemetriou, M. D. Grossberg, and S. K. Nayar, "Multiresolution histograms and their use for recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 7, pp. 831–847, 2004.

[75] M. Dixit, S. Chen, D. Gao, N. Rasiwasia, and N. Vasconcelos, "Scene clas-

sification with semantic fisher vectors," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2974–2983.

[76] T. S. Lee and D. Mumford, "Hierarchical Bayesian inference in the visual cortex," *Journal of optical society of America*, vol. 20, no. 7, pp. 1434–1448, 2003.

[77] S. Grossberg and T. Huang, "ARTSCENE: A neural system for natural scene classification," *Journal of Vision*, vol. 9, no. 4, pp. 1–19, 2009.

[78] P. Viola and M. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[79] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, "Robust object recognition with cortex-like mechanisms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 411–426, 2007.

[80] B. Leibe, A. Leonardis, and B. Schiele, "Combined object categorization and segmentation with an implicit shape model," in *Proceedings of European Conference on Computer Vision Workshop*, 2004, pp. 17–32.

[81] A. Torralba, K. P. Murphy, and W. T. Freeman, "Sharing features: efficient boosting procedures for multiclass object detection," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2004, pp. 762–769.

[82] T. Serre and M. Riesenhuber, "Realistic modeling of simple and complex cell tuning in the hmax model, and implications for invariant object recognition in cortex," Massachusetts Institute of Technology, Tech. Rep., 2004.

[83] J. Mutch and D. G. Lowe, "Multiclass object recognition with sparse, localized features," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 11–18.

[84] S. P. Brumby, G. Kenyon, W. Landecker, C. Rasmussen, S. Swaminarayan, and L. M. A. Bettencourt, "Large-scale functional models of visual cortex for remote sensing," in *Proceedings of IEEE Applied Imagery Pattern Recognition Workshop*, 2009, pp. 1–6.

[85] A. Jiang, C. Wang, B. Xiao, and R. Dai, "A new biologically inspired feature for scene image classification," in *Proceedings of International Conference on Pattern Recognition*, 2010, pp. 758–761.

[86] A. Torralba, A. Oliva, M. Castelhano, and J. Henderson, "Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search," *Psychological Review*, vol. 113, no. 4, pp. 766–786, 2006.

[87] Y. Han and G. Liu, "A hierarchical GIST model embedding multiple biological feasibilities for scene classification," in *Proceedings of International Conference on Pattern Recognition*, 2010, pp. 3109–3112.

[88] B. Scholkopf, A. Smola, and K. R. Muller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[89] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture

of monkey striate cortex," *The Journal of Physiology*, vol. 1, no. 195, pp. 215–243, 1968.

[90] 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.

[91] S. Theodoridis, *Machine learning: A Bayesian and optimization perspective*. London: Academic Press, 2015.

[92] H. Goh, N. Thome, M. Cord, and J. H. Lim, "Learning deep hierarchical visual feature coding," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 12, pp. 2212–2225, 2014.

[93] J. Vogel, A. Schwaninger, C. Wallraven, and H. H. Bülthoff, "Categorization of natural scenes: Local versus global information and the role of color," *ACM Transactions on Applied Perception*, vol. 4, no. 3, pp. 1–21, 2007.

[94] L. Fei-Fei and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 524–531.

[95] M. Brown and S. Susstrunk, "Multi-spectral SIFT for scene category recognition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 177–184.

[96] C. Liu, J. Yuen, and A. Torralba, "Sift flow: Dense correspondence across scenes and its applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 978–994, 2011.

[97] L. Bo, X. Ren, and D. Fox, "Kernel descriptors for visual recognition," in *Advances in Neural Information Processing Systems 23*, J. Lafferty, C. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Eds. Curran Associates, Inc., 2010, pp. 244–252.

[98] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001, pp. 511–518.

[99] P. Rybski, D. Huber, D. Morris, and R. Hoffman, "Visual classification of coarse vehicle orientation using histogram of oriented gradients features," in *Proceedings of IEEE Intelligent Vehicles Symposium*, 2010, pp. 921–928.

[100] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[101] Y. Bai, L. Guo, L. Jin, and Q. Huang, "A novel feature extraction method using pyramid histogram of orientation gradients for smile recognition," in *Proceedings of IEEE International Conference on Image Processing*, 2009, pp. 3305 –3308.

[102] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.

[103] Y. Zheng, C. Shen, R. Hartley, and X. Huang, "Pyramid center-symmetric lo-

cal binary/trinary patterns for effective pedestrian detection," in *Proceedings of Asian Conference on Computer Vision*, 2011, pp. 281–292.

[104] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[105] Z. Guo, D. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1657–1663, 2010.

[106] Z. Li, G. Liu, Y. Yang, and J. You, "Scale- and rotation-invariant local binary pattern using scale-adaptive texton and subuniform-based circular shift," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2130–2140, 2012.

[107] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution grayscale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, p. 2002, 2002.

[108] Y. Xiao, J. Wu, and J. Yuan, "mCENTRIST: A multi-channel feature generation mechanism for scene categorization," *IEEE Transactions on Image Processing*, vol. 23, no. 2, pp. 823–836, 2014.

[109] Y. Abe, M. Shikano, T. Fukuda, F. Arai, and Y. Tanaka, "Vision based navigation system for autonomous mobile robot with global matching," in

Proceedings of IEEE International Conference on Robotics and Automation, 1999, pp. 1299–1304.

[110] Y. Yu, G. K. I. Mann, and R. G. Gosine, "A novel robotic visual perception method using object-based attention," in *Proceedings of IEEE International Conference on Robotics and Biomimetics*, 2009, pp. 1467–1473.

[111] B. Schauerte, B. Kuhn, K. Kroschel, and R. Stiefelhagen, "Multimodal saliency-based attention for object-based scene analysis," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 1173–1179.

[112] S. Y. Bao, M. Sun, and S. Savarese, "Toward coherent object detection and scene layout understanding," *Image and Vision Computing*, vol. 29, no. 9, pp. 569–579, 2011.

[113] C. Siagian and L. Itti, "Biologically-inspired robotics vision monte-carlo localization in the outdoor environment," in *Proceedings of IEEE-RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 1723–1730.

[114] D. Hoiem, A. A. Efros, and M. Hebert, "Putting objects in perspective," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 2137– 2144.

[115] L. Ahn Von, "Games with a purpose," *Computer*, vol. 39, no. 6, pp. 96–98, 2006.

[116] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "LabelMe: a

database and web-based tool for image annotation," *International Journal of Computer Vision*, vol. 77, no. 1, pp. 157–173, 2008.

[117] D. Jia, D. Wei, R. Socher, L. Li-Jia, L. Kai, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[118] Yahoo, "Flickr," 2004. [Online]. Available: https://www.flickr.com/

[119] R. J. Peters and L. Itti, "Beyond bottom-up: Incorporating task-dependent influences into a computational model of spatial attention," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[120] L. Li and L. Fei-Fei, "What, where and who? classifying events by scene and object recognition," in *Proceedings of IEEE International Conference on Computer Vision*, 2007, pp. 1–8.

[121] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 413–420.

[122] M. R. Boutell, L. Jiebo, and C. M. Brown, "Factor graphs for region-based whole-scene classification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 2006, pp. 104–104.

[123] D. Gokalp and S. Aksoy, "Scene classification using bag-of-regions representations," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[124] P. Quelhas, F. Monay, J. M. Odobez, D. Gatica-Perez, T. Tuytelaars, and L. Van Gool, "Modeling scenes with local descriptors and latent aspects," in *Proceedings of IEEE International Conference on Computer Vision*, 2005, pp. 883–890.

[125] N. Serrano, A. Savakis, and J. Luo, "Improved scene classification using efficient low-level features and semantic cues," *Pattern Recognition*, vol. 37, no. 9, pp. 1773–1784, 2004.

[126] X. Han and Y. Chen, "Image categorization by learned PCA subspace of combined visual-words and low-level features," in *Proceedings of International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2009, pp. 1282–1285.

[127] S. Guangda, Z. Cuiping, D. Rong, and D. Cheng, "MMP-PCA face recognition method," *Electronics Letters*, vol. 38, no. 25, pp. 1654–1656, 2002.

[128] X. Xie and K. M. Lam, "Gabor-based kernel PCA with doubly nonlinear mapping for face recognition with a single face image," *IEEE Transactions on Image Processing*, vol. 15, no. 9, pp. 2481–2492, 2006.

[129] A. Malhi and R. X. Gao, "PCA-based feature selection scheme for machine defect classification," *IEEE Transactions on Instrumentation and Measurement*, vol. 53, no. 6, pp. 1517–1525, 2004.

[130] J. L. Yang and H. X. Li, "PCA based sequential feature space learning for gene selection," in *Proceedings of International Conference on Machine Learning and Cybernetics*, 2010, pp. 3079–3084.

[131] H. M. Ebied, "Feature extraction using PCA and kernel-PCA for face recognition," in *Proceedings of International Conference on Informatics and Systems*, 2012, pp. 72–77.

[132] INRIA Graffiti data set, "Viewpoint change sequences," 2004. [Online]. Available: http://kahlan.eps.surrey.ac.uk/featurespace/web/data.htm

[133] H. Y. Lee, H. K. Lee, and Y. H. Ha, "Spatial color descriptor for image retrieval and video segmentation," *IEEE Transactions on Multimedia*, vol. 5, no. 3, pp. 358–367, 2003.

[134] Y. J. Song, W. B. Park, D. W. Kim, and J. H. Ahn, "Content-based image retrieval using new color histogram," in *Proceedings of International Symposium on Intelligent Signal Processing and Communication Systems*, 2004, pp. 609–611.

[135] S. Jeong, C. S. Won, and R. M. Gray, "Histogram-based image retrieval using Gauss mixture vector quantization," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2003, pp. 677–680.

[136] G. H. Liu and J. Y. Yang, "Content-based image retrieval using color difference histogram," *Pattern Recognition*, vol. 46, no. 1, pp. 188 – 198, 2013.

[137] J. J. Koenderink and A. J. Van Doorn, "The structure of locally orderless images," *International Journal of Computer Vision*, vol. 31, no. 2, pp. 159–168, 1999.

[138] M. Kass and J. Solomon, "Smoothed local histogram filters," *ACM Transactions on Graphics*, vol. 29, no. 4, pp. 100:1–100:10, 2010.

[139] M. Igarashi, A. Mizuno, and M. Ikebe, "Accuracy improvement of histogram-based image filtering," in *Proceedings of IEEE International Conference on Image Processing*, 2013, pp. 1217–1221.

[140] M. J. Swain and D. H. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.

[141] M. Stricker and M. Orengo, "Similarity of color images," in *Proceedings of Storage and Retrieval for Image and Video Databases*, 1995, pp. 381–392.

[142] B. V. Funt and G. D. Finlayson, "Color constant color indexing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 5, pp. 522–529, 1995.

[143] H. Kikuchi, S. Kataoka, S. Muramatsu, and H. Huttunen, "Color-tone similarity of digital images," in *Proceedings of IEEE International Conference on Image Processing*, 2013, pp. 393–397.

[144] P. Gupta, S. S. Arrabolu, M. Brown, and S. Savarese, "Video scene categorization by 3D hierarchical histogram matching," in *Proceedings of IEEE International Conference on Computer Vision*, 2009, pp. 1655–1662.

[145] J. Qin and N. H. C. Yung, "Scene categorization via contextual visual words," *Pattern Recognition*, vol. 43, no. 5, pp. 1874–1888, 2010.

[146] D. Blei, A. Ng, and M. Jordan, "Latent Dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[147] T. Kadir and M. Brady, "Scale, saliency and image description," *International Journal of Computer Vision*, vol. 45, no. 2, pp. 83–105, 2001.

[148] J. C. V. Gemert, J. M. Geusebroek, C. J. Veenman, and A. W. M. Smeulders, "Kernel codebooks for scene categorization," in *Proceedings of European Conference on Computer vision*, 2008, pp. 696–709.

[149] H. Zhang, A. C. Berg, M. Maire, and J. Malik, "SVM-KNN: discriminative nearest neighbor classification for visual category recognition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 2126–2136.

[150] O. Boiman, E. Shechtman, and M. Irani, "In defense of nearest-neighbor based image classification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[151] K. Yu, T. Zhang, and Y. Gong, "Nonlinear learning using local coordinate coding," in *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, Eds. Curran Associates, Inc., 2009, pp. 2223–2231.

[152] F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier, "Large-scale image retrieval with compressed fisher vectors," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3384–3391.

[153] C. Doersch, A. Gupta, and A. A. Efros, "Mid-level visual element discovery as discriminative mode seeking," in *Advances in Neural Information Processing Systems 26*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 494–502.

[154] S. L. Salzberg, "On comparing classifiers: Pitfalls to avoid and a recom-

mended approach," *Data Mining and Knowledge Discovery*, vol. 1, no. 3, pp. 317–328, 1997.

[155] P. J. Phillips, M. Hyeonjoon, S. A. Rizvi, and P. J. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.

[156] H. Kazuhiro, "Local autocorrelation of similarities with subspaces for shift invariant scene classification," *Pattern Recognition*, vol. 44, no. 4, pp. 794–799, 2011.

[157] F. Cakir, U. Gudukbay, and O. Ulusoy, "Nearest-neighbor based metric functions for indoor scene recognition," *Computer Vision and Image Understanding*, vol. 115, no. 11, pp. 1483–1492, 2011.

[158] X. Meng, Z. Wang, and L. Wu, "Building global image features for scene recognition," *Pattern Recognition*, vol. 45, no. 1, pp. 373–380, 2012.

[159] L. Zhang, R. Ji, Y. Xia, Y. Zhang, and X. Li, "Learning a probabilistic topology discovering model for scene categorization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 8, pp. 1622–1634, 2015.

[160] S. Karayev, M. Fritz, and T. Darrell, "Anytime recognition of objects and scenes," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 572–579.

[161] A. Bergamo and L. Torresani, "Classemes and other classifier-based features for efficient object categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 10, pp. 1988–2001, 2014.

[162] G. Cheng, J. Han, L. Guo, and T. Liu, "Learning coarse-to-fine sparselets for efficient object detection and scene classification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1173–1181.

[163] Y. Huang, Z. Wu, L. Wang, and T. Tan, "Feature coding in image classification: A comprehensive study," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, pp. 493–506, 2014.

[164] M. Sun, W. Huang, and S. Savarese, "Find the best path: An efficient and accurate classifier for image hierarchies," in *Proceedings of IEEE International Conference on Computer Vision*, 2013, pp. 265–272.

[165] N. M. Elfiky, S. F. Khan, J. Weijer, and J. Gonzalez, "Discriminative compact pyramids for object and scene recognition," *Pattern Recognition*, vol. 45, no. 4, pp. 1627–1636, 2012.

[166] A. Perina, M. Cristani, and V. Murino, "Learning natural scene categories by selective multi-scale feature extraction," *Image and Vision Computing*, vol. 28, no. 6, pp. 927–939, 2010.

[167] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letter*, vol. 27, no. 8, pp. 861–874, 2006.

[168] T. Li, S. Zhu, and M. Ogihara, "Using discriminant analysis for multi-class classification," in *Proceedings of IEEE International Conference on Data Mining*, 2003, pp. 589–592.

[169] T. Chin, D. Suter, and H. Wang, "Boosting histograms of descriptor dis-

tances for scalable multiclass specific scene recognition," *Image and Vision Computing*, vol. 29, no. 4, pp. 241–250, 2011.

[170] N. Shroff, P. Turaga, and R. Chellappa, "Moving vistas: Exploiting motion for describing scenes," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1911–1918.

[171] K. G. Derpanis, M. Lecce, K. Daniilidis, and R. P. Wildes, "Dynamic scene understanding: The role of orientation features in space and time in scene classification," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1306–1313.

[172] R. De Geest and T. Tuytelaars, "Dense interest features for video processing," in *Proceedings of IEEE International Conference on Image Processing*, 2014, pp. 5771–5775.

[173] J. Y. H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4694–4702.

[174] T. Wakahara and K. Odaka, "Adaptive normalization of handwritten characters using global/local affine transformation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1332–1341, 1998.

[175] C. L. Liu, K. Nakashima, H. Sako, and H. Fujisawa, "Handwritten digit recognition: investigation of normalization and feature extraction techniques," *Pattern Recognition*, vol. 37, no. 2, pp. 265–279, 2004.

[176] M. Varma and A. Zisserman, *Classifying Images of Materials: Achieving Viewpoint and Illumination Independence*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2002, vol. 2352, ch. 17, pp. 255–271.

[177] S. Lazebnik, C. Schmid, and J. Ponce, "A sparse texture representation using local affine regions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1265–1278, 2005.

[178] M. Mellor, B. W. Hong, and M. Brady, "Locally rotation, contrast, and scale invariant descriptors for texture analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 1, pp. 52–61, 2008.

[179] S. R. Arashloo and J. Kittler, "Energy normalization for pose-invariant face recognition based on mrf model image matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 6, pp. 1274–1280, 2011.

[180] L. Ding, X. Ding, and C. Fang, "Continuous pose normalization for pose-robust face recognition," *IEEE Signal Processing Letters*, vol. 19, no. 11, pp. 721–724, 2012.

[181] M. Tistarelli, S. Yunlian, and N. Poh, "On the use of discriminative cohort score normalization for unconstrained face recognition," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 12, pp. 2063–2075, 2014.

[182] D. Decoste and B. Schlkopf, "Training invariant support vector machines," *Machine Learning*, vol. 46, no. 1-3, pp. 161–190, 2002.

[183] F. H. C. Tivive and A. Bouzerdoum, "A hierarchical learning network for

face detection with in-plane rotation," *Neurocomputing*, vol. 71, pp. 3253–3263, 2008.

[184] E. T. Pereira, H. M. Gomes, and J. de Carvalho, "An approach for multi-pose face detection exploring invariance by training," *Pattern Recognition*, vol. 8495, pp. 182–191, 2014.

[185] M. K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962.

[186] J. Flusser, "On the independence of rotation moment invariants," *Pattern Recognition*, vol. 33, no. 9, pp. 1405–1410, 2000.

[187] E. Rahtu, M. Salo, and J. Heikkila, "Affine invariant pattern recognition using multiscale autoconvolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 908–918, 2005.

[188] J. Bruna and S. Mallat, "Invariant scattering convolution networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1872–1886, 2013.

[189] D. Sinclair and A. Blake, "Isoperimetric normalization of planar curves," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 8, pp. 769–777, 1994.

[190] I. Rothe, H. Susse, and K. Voss, "The method of normalization to determine invariants," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 4, pp. 366–376, 1996.

[191] T. Suk and J. Flusser, "Affine normalization of symmetric objects," in *Advanced Concepts for Intelligent Vision Systems*, ser. Lecture Notes in Computer Science. Springer, 2005, vol. 3708, pp. 100–107.

[192] Z. Zhang, A. Ganesh, X. Liang, and Y. Ma, "TILT: Transform invariant low-rank textures," *International Journal of Computer Vision*, vol. 99, no. 1, pp. 1–24, 2012.

[193] X. Dai, H. Zhang, T. Liu, H. Shu, and L. Luo, "Legendre moment invariants to blur and affine transformation and their use in image recognition," *Pattern Analysis and Applications*, vol. 17, no. 2, pp. 311–326, 2014.

[194] P. Dong, J. G. Brankov, N. P. Galatsanos, Y. Yang, and F. Davoine, "Digital watermarking robust to geometric distortions," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2140–2150, 2005.

[195] H. Dirilten and T. G. Newman, "Pattern matching under affine transformations," *IEEE Transactions on Computers*, vol. C-26, no. 3, pp. 314–317, 1977.

[196] Y. Lamdan, J. T. Schwartz, and H. J. Wolfson, "Object recognition by affine invariant matching," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1988, pp. 335–344.

[197] S. Espana-Boquera, M. J. Castro-Bleda, J. Gorbe-Moya, and F. Zamora-Martinez, "Improving offline handwritten text recognition with hybrid HMM/ANN models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 767–779, 2011.

[198] M. S. Yasein and P. Agathoklis, "An image normalization technique based on geometric properties of image feature points," in *IEEE International Symposium on Signal Processing and Information Technology*, 2007, pp. 116–121.

[199] S. C. Pei and C. N. Lin, "Image normalization for pattern recognition," *Image and Vision Computing*, vol. 13, no. 10, pp. 711–723, 1995.

[200] D. Shen and H. H. S. Ip, "Generalized affine invariant image normalization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 5, pp. 431–440, 1997.

[201] Y. Zhang, C. Wen, Y. Zhang, and Y. C. Soh, "On the choice of consistent canonical form during moment normalization," *Pattern Recognition Letters*, vol. 24, no. 16, pp. 3205–3215, 2003.

[202] T. H. Reiss, *Recognizing Planar Objects Using Invariant Image Features*. Berlin: Springer, 1993.

[203] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library (COIL100)," Columbia University, Tech. Rep., 1996.

[204] F. Samaria and A. Harter, "Parameterisation of a stochastic model for human face identification," in *Proceedings of IEEE Workshop on Applications of Computer Vision*, 1994, pp. 138–142.

[205] J. Flusser, T. Suk, and B. Zitov, *Moments and moment invariants in pattern recognition*. John Wiley & Sons, Ltd, 2009.

[206] Y. Zhang, C. Wen, Y. Zhang, and Y. C. Soh, "Determination of blur and affine

combined invariants by normalization," *Pattern Recognition*, vol. 35, no. 1, pp. 211–221, 2002.

[207] C. A. Rothwell, A. Zisserman, D. A. Forsyth, and J. L. Mundy, "Canonical frames for planar object recognition," in *Proceedings of European Conference on Computer Vision*, vol. 588, 1992, pp. 757–772.

[208] I. Weiss, "Noise-resistant invariants of curves," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 943–948, 1993.

[209] T. Suk and J. Flusser, "Projective moment invariants," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 10, pp. 1364–1367, 2004.

[210] X. Wei, S. L. Phung, and A. Bouzerdoum, "Affine-invariant scene categorization," in *Proceedings of IEEE International Conference on Image Processing*, 2014, pp. 1031–1035.

[211] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3485–3492.

[212] A. L. C. Barczak, M. J. Johnson, and C. H. Messom, "Revisiting moment invariants: Rapid feature extraction and classification for handwritten digits," in *Proceedings of Image and Vision Computing New Zealand*, 2007, pp. 137–142.

[213] J. M. Morel and G. Yu, "ASIFT: A New Framework for Fully Affine Invariant

Image Comparison," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 438–469, 2009.

[214] J. Bruna and S. Mallat, "Classification with scattering operators," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1561–1566.

[215] T. Zhang, B. Ghanem, S. Liu, C. Xu, and N. Ahuja, "Low-rank sparse coding for image classification," in *Proceedings of IEEE International Conference on Computer Vision*, 2013, pp. 281–288.

[216] X. Chai, S. Shan, and W. Gao, "Pose normalization for robust face recognition based on statistical affine transformation," in *Proceedings of the Joint Conference of the Fourth International Conference on Information, Communications and Signal Processing and the Fourth Pacific Rim Conference on Multimedia*, vol. 3, 2003, pp. 1413–1417.

[217] E. Yacoub, N. Harel, and K. Uurbil, "High-field fMRI unveils orientation columns in humans," in *Proceedings of the National Academy of Sciences*, vol. 105, 2008, pp. 10 607–10 612.

[218] C. Jacques, C. Schiltz, and V. Goffaux, "Face perception is tuned to horizontal orientation in the n170 thiem window," *Journal of Vision*, vol. 14, no. 2, pp. 1–18, 2014.

[219] B. Rossion, "Picture-plane inversion leads to qualitative changes of face perception," *Acta Psychologica*, vol. 128, no. 2, pp. 274–289, 2008.

[220] J. Luo and A. Savakis, "Indoor vs outdoor classification of consumer photographs using low-level and semantic features," in *Proceedings of IEEE International Conference on Image Processing*, vol. 2, 2001, pp. 745–748.

[221] M. R. Boutell, C. B. Brown, and J. Luo, "Review of the state of the art in semantic scene classification," University of Rochester, Rochester, NY, USA, Tech. Rep., 2002.

[222] N. Serrano, A. Savakis, and J. Luo, "A computationally efficient approach to indoor/outdoor scene classification," in *Proceedings of IEEE International Conference on Pattern Recognition*, vol. 4, 2002, pp. 146–149.

[223] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2005, pp. 524–531.

[224] A. Bosch, A. Zisserman, and X. Muoz, "Scene classification via pLSA," in *Proceedings of the 9th European conference on Computer Vision*, vol. Part IV, 2006, pp. 517–530.

[225] B. Fernando, E. Fromont, and T. Tuytelaars, "Effective use of frequent itemset mining for image classification," pp. 214–227, 2012.

[226] A. Oliva and A. Torralba, "Building the gist of a scene: the role of global image features in recognition," in *Progress in Brain Research*, S. L. M. L. M. M. J. M. A. S. Martinez-Conde and P. U. Tse, Eds. Elsevier, 2006, vol. 155, pp. 23–36.

[227] ——, "Eight scene categories dataset and gist code," 2012. [Online]. Available: http://people.csail.mit.edu/torralba/code/spatialenvelope