Invariant Encoding Schemes for Visual Recognition

Andrew J. Newell

Centre for Mathematics and Physics in the Life Sciences
and Experimental Biology (CoMPLEX)
University College London

A thesis submitted to University College London (UCL) in the
Faculty of Mathematical and Physical Sciences
for the degree of Doctor of Philosophy

December 2011

Declaration

I, Andrew Newell, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Andrew J. Newell
3rd December 2011

Abstract

Many encoding schemes, such as the Scale Invariant Feature Transform (SIFT) and Histograms of Oriented Gradients (HOG), make use of templates of histograms to enable a loose encoding of the spatial position of basic features such as oriented gradients. Whilst such schemes have been successfully applied, the use of a template may limit the potential as it forces the histograms to conform to a rigid spatial arrangement. In this work we look at developing novel schemes making use of histograms, without the need for a template, which offer good levels of performance in visual recognition tasks.

To do this, we look at the way the basic feature type changes across scale at individual locations. This gives rise to the notion of column features, which capture this change across scale by concatenating feature types at a given scale separation. As well as applying this idea to oriented gradients, we make wide use of Basic Image Features (BIFs) and oriented Basic Image Features (oBIFs) which encode local symmetry information. This resulted in a range of encoding schemes.

We then tested these schemes on problems of current interest in three application areas. First, the recognition of characters taken from natural images, where our system outperformed existing methods. For the second area we selected a texture problem, involving the discrimination of quartz grains using surface texture, where the system achieved near perfect performance on the first task, and a level of performance comparable to an expert human on the second. In the third area, writer identification, the system achieved a perfect score and outperformed other methods when tested using the Arabic handwriting dataset as part of the ICDAR 2011 Competition.

Acknowledgements

I would like to thank a number of people for their help and support during the course of my PhD.

My supervisor, Lewis Griffin, has provided great support in helping me gather my ideas and apply them in a way that has led to publication. My second supervisor, Alan Johnston, has provided very helpful input throughout the course of the PhD. Ruth Morgan provided invaluable help with the work on texture. In addition, many members of both CoMPLEX and the Department of Computer Science have helped along the way. I'd also like the thank the anonymous reviewers who have provided helpful feedback on the work that has been submitted for publication.

Published work

Sections of this work have appeared in the following publications:

A. J. Newell, L. D. Griffin, R. M. Morgan, and P. A. Bull. Texture-based estimation of physical characteristics of sand grains. In *Proceedings of the International Conference on Digital Image Computing: Techniques and Applications (DICTA)* Sydney, pages 504–509, 2010.

A. J. Newell and L.D. Griffin. Multiscale histogram of oriented gradient descriptors for robust character recognition. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, 2011.

A. D. F. Clarke, F. Halley, A. J. Newell, L. D. Griffin, and M. J. Chantler. Perceptual similarity: A texture challenge. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2011.

A. J. Newell and L.D. Griffin. Natural image character recognition using oriented Basic Image Features. In Proceedings of the International Conference on Digital Image Computing: Techniques and Applications (DICTA), 2011.

A. J. Newell, R. M. Morgan, L. D. Griffin, P. A. Bull, J.R. Marshall, and G. Graham. Automated texture recognition of quartz sand grains for forensic applications. *Journal of Forensic Sciences* 2012 (in press)

In addition, results from Chapter 8 have been published in:

A. Hassane, S. Al-Maadeed, J. M. Aljaam, A. Jaoua, and A. Bouridane. The ICDAR2011 Arabic writer identication contest. In *Proceedings of The International Conference on Document Analysis and Recognition (ICDAR)*, 2011.

Contents

List of Experiments

List of Figures

List of Algorithms

List of Tables

List of Abbreviations

Abbreviation	Details
BIFs	Basic Image Features
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart
DOLP	Difference of Low Pass transform
DoG	Difference of Gaussian
DtG	Derivative of Gaussian
ELF	Energy Level of Formation
GLOH	Gradient Location Orientation Histogram
HMAX	The HMAX model of object recognition [188].
HOG	Histograms of Oriented Gradients
ICDAR	The International Conference on Document Analysis and Recognition
kNN	k Nearest Neighbour
LBP	Local Binary Patterns
MKL	Multiple Kernel Learning
MNIST	The MNIST set of handwritten digits [131]
NN	Nearest Neighbour
NUP	No Upturned Plates
oBIFs	Oriented Basic Image Features
OCR	Optical Character Recognition
OGC	Oriented Gradient Columns
PCA	Principal Component Analysis
RIFT	Rotation Invariant Feature Transform
SEEMORE	An object recognition system proposed by Mel [150]
SEM	Scanning Electron Microscope
SIFT	Scale Invariant Feature Transform
SVM	Support Vector Machine
UP	Upturned Plates
VOC	Visual Object Classes

Chapter 1

Introduction

1.1 Background

In visual recognition tasks, an unknown entity is compared to a set of previously learnt classes. These classes, which have labels, may be individual objects, or object categories or textures or something more abstract. At the heart of this process lies a measure of similarity which enables us to make statements about the degree to which the unknown entity belongs to any of the previously learnt classes.

Measuring this similarity requires a computational process which takes training data, from which classes are learnt, and then compared to the unknown entities, or the test data. If there were unlimited training data and an infinite computational power then recognition could be done by comparing the test data with each example of the training data and finding the most suitable label. In this way we could consider the training data mapping out a region in some space for each class, and the process then is to see whether the test example falls within any of these regions.

In a real recognition task we have a limited amount of training data and finite computational power. Mapping out the regions of the space for each class then becomes far more difficult as we have to estimate where the boundaries lie between classes using a small number of training examples. We could view this simply as a learning task, in which some form of model is supplied and a computational process is applied to find the optimal parameters of the model.

However, visual recognition problems involve images that are taken, or simulate, the physical world. Objects within the physical world are subject to physical laws, and thus, for example, moving an object slightly does not change its identity. These rules carry over into the images, so that a small shift of an object within an image does not change its identity. Similarly, a rotation of a texture is unlikely to change its identity.

Thus, we can view visual recognition tasks as forming a subset of general recognition tasks, bound together by a common set of rules which ultimately come from the nature of physical objects and their interaction with light. We could then attempt to tackle visual recognition problems by finding a suitable representation of images that incorporates as much prior knowledge of these rules as possible which allows us to learn class boundaries efficiently using a computational process.

Finding such a representation is the wider goal of this work.

1.2 Finding a suitable representation

Finding a suitable representation can itself involve a learning process, where features at different stages of the computation of representation are learnt using large numbers of natural images. This can be done at the early stages [146], or for mid level features [69, 20, 215] or can be applied to an entire hierarchy [132, 119]. In this work we take a different approach, viewing the process of finding a suitable representation of images for visual recognition tasks as an investigative process.

This process is guided by the traditional need for a set of features which are discriminative enough to be able to distinguish between classes in visual recognition, yet consistent enough to ensure that they occur for instances of the same class.

The requirement for consistency gives rise to the need for a representation to be invariant to transformations that we know do not affect object identity. We can divide these transformations into two groups. The first of these is transformations that are common to all objects and where the effect can be predicted. For example a shift in the position of an object gives rises to a shift in the image, which can be predicted. Thus, it may be possible to build invariance to these transformations into a representation.

The second group of transformations are those that have an effect that cannot be predicted from a single instance of a particular class. For example, a class of objects may contain slightly different spatial arrangements of the same basic features, in which case we may wish to develop invariance to small adjustments in the positions of basic features. However, the range in which positions of basic features vary within instances of the same class will be specific to the class. We cannot therefore build invariance to this into a representation and the aim instead could be to develop representation that allows such invariance to be learnt for each class.

1.3 Levels of representation

In the search for a suitable representation for visual recognition, we can consider the process in stages where invariance to different transformations is introduced. As we are concerned with images, we begin with a measurement of the light intensity at each location. This representation is sensitive to changes in light level, both in terms of local changes due to light sources and to global changes. The first stage in a representation could therefore be expected to provide a set of features that is largely invariant to changes in lighting conditions.

Many such representations exist, such as edge features, oriented gradients and corner detectors. Each of these can be detected in many different ways, but the overall aim is to achieve a representation of consistent features. This first stage is not the focus of our work. Instead we focus on the next stage, where we look to combine these basic features into a representation that has sufficient power of discrimination as well as providing invariance as described in the previous section.

In combining these basic features we are looking to encode the relationship between them in some way. As suggested in the previous section, we might view a suitable way of encoding the relationship as a loose spatial structure. In the most general case, we could view a model of a visual class as an arrangement of basic features with approximate distances between them. So, for the example illustrated in Figure 1.1, one particular class could be an arrangement of five features, $(F_1, F_2, .., F_5)$ with certain distances, (d_1, d_2, d_3, d_4), and tolerances, $(\delta_1, \delta_2, \delta_3, \delta_4)$ between groups of features.

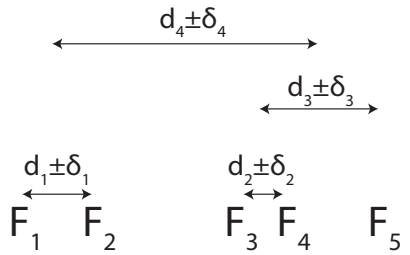


Figure 1.1: An example general model for a visual class.
Individual features are encoded along with the spatial relationship between them.

Our aim could then to be develop a representation that could learn arbitrary classes along these lines, with basic features connected by approximate distances, thus creating a structural description of a visual class. However, such structural representations pose

considerable computational difficulties[129]. As a result, it is often desirable to impose a predetermined structure on the representation.

In computer vision, this is often done through the use of complex features that possess their own structure. For example, with the Scale Invariant Feature Transform (SIFT) [143] and Histogram of Oriented Gradient (HOG) [53] features, this structure is a template of histograms of oriented gradients. An image is then encoded as an unordered set of such features.

In biologically motivated network models, such as the one proposed by Riesenhuber et al. [188], the structure arises from the architecture of the network. In this model, alternating layers of selective and pooling cells gradually introduce spatial and scale invariance as the receptive field of cells increases up the hierarchy. This leads to a representation that can be less spatially constrained than a template of histograms, but the size of the receptive fields and the number of layers still imposes some structure on the representation.

1.4 Multiscale encoding schemes

Features such as SIFT are encoded within a scale space environment, with each feature centred upon a particular location and scale. Thus, when an image is encoded with multiple SIFT features at different locations and scales, we can consider that, in some way, to be a multiscale representation. In the network model of Riesenhuber et al., the first layer comprises cells that are sensitive to local oriented features at different scales. Subsequent layers then serve to pool the outputs of these cells to produce a final representation that is scale invariant.

Both of these methods make use of scale for the purposes of scale invariance, so that they can be used to recognise objects that occur at different scales. In addition to providing the means for scale invariance, scale space also provides for the potential detect additional structure, referred to as deep structure [122, 15, 247, 138]. The essential aspect of deep structure is the linking of structure found at different scales, which does not occur in a set of SIFT features.

Representations that link structure at different scales have been used in the past. The most common of these, pyramid representations, [37, 36, 1, 49, 50] use a smoothing kernel to produce a stack of blurred images of decreasing size. Whilst such representations do link structure across different scales, this is achieved either using a rigid geometric structure, which limits the invariance properties of the representation.

Varma et al. have proposed a set of texton-based features that combine the outputs of filters at different scales [245] in a way that offers a far greater degree of invariance to pyramid representations. However, whilst these features have performed well for texture recognition, the features do not encode local orientation information, which has played a significant part in many modern object recognition methods.

1.5 The contributions of this work

In this work we propose a novel set of encoding schemes that combine structure at different scales to produce invariant representations that are of use in a range of applications . To do this, we first suggest a new set of multiscale features, called column features, which encode conjunctions of primitive features across scale. We explore how these features can be used in the context of simple histograms and demonstrate how they can outperform existing single scale schemes, such as SIFT and HOG, in the problem of natural image character recognition.

We also investigate multiscale extensions to the HOG encoding scheme and demonstrate that, by combining oriented gradient features in a particular manner, the multiscale scheme can offer an improvement in performance over the single scale version. We then demonstrate the usefulness of column features in two additional application areas, texture recognition and write identification. Our overall aim in this work is to demonstrate that, by combining structure at different scales within a single feature, we can produce encoding schemes that perform well in a range of applications.

The key novel aspects of the work are summarised below.

- **The oBIF Column scheme**

  The oriented Basic Image Feature (oBIF) Column scheme is a novel encoding scheme that encodes local orientation and symmetry type information at multiple scales. The scheme is fundamentally different from previous multiscale schemes in the way that information is combined. Whereas schemes such as pyramid representations involve a stack of blurred images, the oBIF Column scheme uses a histogram of conjunctions of primitive features at each location in the image. This allows the oBIF Column scheme to be shift invariant, both at the global level and, to a certain extent, at the local level.

  The scheme differs from single scale features, such as SIFT and HOG, which use a template of histograms. In these schemes a template is necessary, as a simple

histogram of oriented gradients is rarely descriptive enough, as demonstrated in chapter 4 of this work. However, as the oBIF Column features encode information at multiple scales, they can offer good levels of performance without the use of a template.

- **A novel multiscale HOG scheme**

  We propose a novel multiscale version of the HOG descriptor and demonstrate that it outperforms the single scale version on the problem of natural image character recognition. The novel aspect of this method is in the way in which local orientation information is combined across scales. In an investigation of multiscale versions of HOG we demonstrate that it is necessary to combine orientation information at each location in the image rather than at the level of the template.

- **A novel way of applying texture encoding to writer identification**

  In order to apply the oBIF Column scheme to the problem of writer identification we have proposed a novel way to use a texture-based encoding scheme to determine authorship of handwriting. This involves the extraction of a style vector for each author, by looking at the deviation from the mean encoding for a certain piece of text. This differs from other approaches to writer identification which rely on the detection of specific features for each author.

- **Discrimination of quartz grain types using surface texture**

  We propose the column scheme for use on the problem of quartz grain discrimination. As this problem has not previously been investigated using modern texture recognition methods, the application itself is novel. In this work we investigate different ways of combining multiple images from a single grain and demonstrate that individual classification, followed by a pooling stage, achieves the best results. The results indicate that the performance of the column scheme is comparable to expert human performance.

- **An evaluation of spatial binning schemes with multiple features**

  In order to be able to compare the novel multiscale encoding schemes with single scale schemes that use templates of histograms, we have performed an evaluation of oriented Basic Image Features. We investigate the use of oBIFs, and the orientation version Basic Image Features (BIFs), both within the framework of a simple histogram and a template of histograms. We demonstrate that , in the context of a simple histogram, oBIFs outperform oriented gradients. However,

when spatial binning is used, the performance of oBIFs is equivalent to oriented gradients. This work demonstrates the relative usefulness of local orientation and symmetry type information in character recognition.

1.6 The structure of this thesis

We begin by presenting a review of the relevant literature in Chapter 2. This begins with a brief review of the history of invariant recognition within computer vision and a summary of the current methods in popular use. This is followed by a discussion of a range of biologically motivated models of recognition, particularly their contribution to the computational understanding of problems in recognition. We then discuss different form of representation, including the difference between image- and object-based representations and view-based versus structural representations. We then discuss texture, in terms of perception, recognition and representation. Finally, we provide a discussion of scale space and mutiscale representations.

In the second section of Chapter 2, we describe and discuss four key methods in recognition. Particular attention is given to the way in which invariance is introduced in to the representation produced by each method.

In Chapter 3, we introduce the datasets which are used for assessing the performance of the various encoding schemes. These are given together with details of basic experiments performed to establish a benchmark performance.

In Chapter 4, we describe and evaluate basic histogram schemes, with and without spatial binning. The main purpose of the work in this chapter is to enable a comparison with the performance of the column schemes produced in the following chapter. In this chapter, the novel aspect of the work is the evaluation of Basic Image Features (BIFs) and oriented Basic Image Features (oBIFs).

The oBIF Column and BIF Column schemes are presented in Chapter 5. This chapter begins with a discussion of the multiscale features referred to as column features. We then assess the performance of the column schemes in a similar way as in Chapter 4. Various invariant versions of the column schemes are presented.

In Chapter 6, we test the column schemes on a current problem in character recognition, which involves characters taken from natural images. Here, the performance of column schemes is compared to existing methods. We also the present the novel multiscale HOG schemes in this chapter.

Chapter 7 presents the work on the texture recognition problem, involving the discrimination of quartz grains using surface texture. This chapter does not present theory, but serves both as a test of the BIF Column scheme, presented in Chapter 5, and a demonstration of the applicability of such systems to the previously unstudied problem of automatic grain type recognition.

In Chapter 8, we describe the extension of the oBIF Column scheme to the problem of writer identification. The performance of the new scheme is then assessed against other methods within the framework of the ICDAR 2011 Arabic Writer Identification contest.

We then examine the performance of the oBIF Column scheme in the presence of clutter, in Chapter 9, and propose a way of extending the scheme to make it more robust. Finally, in Chapter 10, we provide a summary of set of conclusions from the work.

Chapter 2

Literature Review

In this chapter we present a discussion of the relevant literature as well as a description of specific methods. This begins with a discussion of the history of invariant recognition within computer vision, describing key stages in the development of current techniques. Next we discuss a range of biologically motivated models, the majority of which are network models inspired by neurophysiology. We then discuss different forms of representation, in terms of image- versus object-based representations and view-based versus structural representations. The next section covers texture, in terms of perception, recognition and representation. We then describe various approaches to scale space representation, with particular emphasis on multiscale representations. In the second half of this chapter we describe four key methods currently used in computer vision. The methods are described in detail along with a discussion of the invariance properties of each.

2.1 A brief history of invariant recognition in computer vision

2.1.1 Geometric invariant approaches

Many early approaches to recognition in computer vision centred on the search for an appropriate representation [159], the aim of which was to make object identity information more explicit. Where the problem was classification, which involved determining a property that all members of a group share [197], the required information was seen as the invariant of the class. Thus, for object classification, the general approach was to seek a representation that was invariant to any transformation that did not alter membership of the class.

The choice of invariant in such early methods was generally a geometrical one. Mundy suggests three reasons for this [160]. First, capturing the geometry of an object

offered invariance to viewpoint, which was seen as necessary for three-dimensional objects. Second, the objects' geometry was invariant to illumination changes. Third, there was a large body of theory available from mathematics which could be applied to the problem of object recognition.

Whereas this approach worked for very simple classes of objects [61, 194, 35], it was less successful when applied to more complicated classes. In particular, it was shown that invariants did not exist for general three-dimensional shapes [34]. Additionally, geometric invariant approaches were found to have susceptibility to noise and occlusion [160].

2.1.2 Early appearance-based methods

Whilst problems with the geometric invariant approaches were becoming apparent, progress was being made with appearance based methods. In the field of facial recognition, Eigenfaces [236] achieved good levels of performance by creating a low dimensional representation for faces. This was achieved using Principal Component Analysis (PCA) on a set of face images, which had a common alignment and illumination.

For general object categories, appearance manifolds [161] offered a similar way of creating a low dimensional representation by creating an eigenspace. However, it was necessary to have a set of images from different viewpoints and different illuminations to create the eigenspace.

Early appearance based methods also required objects to be well segmented. As this was difficult for objects in complex images, several methods proposed a sliding window [27, 201]. This involved applying a recognition technique to one small section of an image at a time, and was particularly effective in applications such as pedestrian detection [248].

Whilst sliding window methods performed well on certain tasks, the performance came at a considerable computational cost, as a classifier had to be applied to each window. In addition, a suitable size and shape had to be chosen for the window, which limited the success of the approach.

2.1.3 Image patches and feature-based methods

These early appearance-based methods had demonstrated good performance on certain tasks, particularly facial recognition and pedestrian detection. However, for the problem

of object categorisation a different set of methods showed promise. Sometimes referred to as feature-based methods, they combined elements of the early global invariant methods with the sliding window approach to encode patches of an image.

In these methods, patches of an image were encoded to create a set of features that were more complex than the traditional edge or corner features. Patches were encoded to be invariant to changes in illumination [169] and affine transformations [200].

Other methods, inspired by biological systems, utilised local orientation encoding to achieve invariance to illumination changes. Most notably amongst these is the SIFT descriptor [143, 144], which is discussed in greater detail later in this chapter.

2.1.4 Bag of words methods

Many of the feature-based methods, such as SIFT, encode each image patch and then compare it to a set of reference patches. Whilst effective, such a process involves making many comparisons between patches as each individual SIFT feature is stored. In an attempt to limit the number of possible comparisons necessary, a set of related methods called bag-of-words methods have been proposed for categorisation. Whilst these methods are very similar to feature-based methods, an additional quantisation step is used, meaning that individual exact features are not stored.

The standard bag-of-words approach took its inspiration from document processing, where it had been established that the content of a document could be well represented using an orderless set of words. It was thought that the same could be true of images, in that the contents of an image could be well represented by an orderless collection of its parts.

In the application of bag-of-words approach to images, it was necessary to establish the visual equivalent of a word. This enabled an image to be represented not by a set of encoded patches, but by a histogram of visual words. Thus, images no longer had to be compared by classifying individual patches, but rather by comparing two histograms.

Early bag-of-words methods generally used a clustering method to create a codebook of visual words. Since then more complex methods have been used for codebook construction such as vocabulary trees [167] and ensembles of trees [153].

2.1.5 Parts and structure methods

Together, the bag-of-words and feature-based methods have offered high levels of performance in tasks such as categorisation. However, despite the somewhat surprising levels of performance that have been achieved with orderless representations, there still seems a likely upper bound to the level of performance that can be achieved without considering the spatial relationship between features. Here we discuss methods that have attempted encode both the parts of object, such as features or patches, and the structure, meaning the spatial relationship between the parts.

The basic principle is demonstrated in Figure 2.1, where a face is represented by individual parts and the spatial relationship is encoded in the distance between parts. Early parts and structure models tended to detect the presence and location of parts first and then calculated distance between them [33]. However, such methods required hand labelling of parts in training images. In addition, such methods may be highly susceptible to errors in the initial parts detection phase.

Figure 2.1: A simple parts and structure model. (From [73])
Individual parts are encoded along with approximate spatial relations between them.
This is in contrast to the parts only models.

Later methods have taken a probabilistic approach. Generative parts models [72, 70] have demonstrated good levels performance, but this has come at a great computational expense meaning that the maximum number of parts that can be included in an object model is generally low.

Other methods have attempted a computationally simpler approach by extending the orderless appearance based methods to include distance between pairs of features [217, 127]. However, this approach has not demonstrated significantly improved performance with current implementations.

An alternative approach which has shown improved levels of performance without such computational complexity is pyramid matching [129], which constructs a spatial pyramid of local histograms of features. However, this method involves dividing the image using a fixed geometrical structure and is therefore not invariant to affine transformations.

2.1.6 The current state of invariant recognition

The early geometric invariant methods have largely been superseded by the appearance-based orderless methods, which have shown a surprising level of performance on tasks such as object and scene categorisation. However, methods that attempt to encode the structure of objects as well as the parts have either come at a very large computational cost or have imposed a fixed geometrical structure on the image, at the cost of invariance.

Many recent approaches have looked to combine features from the orderless methods with machine learning methods. However, the search for a representation that capture both parts and structure remains active.

2.2 Biologically motivated models

Alongside the methods for invariant recognition found in computer vision there are a set of methods that can be described as biologically motivated computational models. These models combine experimental findings with a computational approach to recognition and are generally assessed not only on their performance but also on their biological plausibility. They play an important role in the wider picture of invariant recognition, not only as an insight into potential computational methods but also, with their generic nature, as an important contrast to the more application-based computer vision methods.

2.2.1 Network models

The majority of models presented in this section are network models of some form or another. These models have their roots in the work of Hubel et al. [105, 106], which laid the foundations for hierarchical models. In this early work, it was demonstrated how visual processing could occur in layers of cells connected through simple transformations. The effect of the hierarchy could therefore be viewed as transforming the representation from the input, in the form of an image, through each layer.

In hierarchical models, two key aspects of layers are selectivity and invariance. In the experiments of Hubel et al., it was found that each of these functions were met by different cells. These two fundamental cell types, the simple cell and the complex cell, are found in some form in almost all network models for visual recognition.

Fukushima extended the simple and complex cell idea into a cascade of alternating selective and invariant layers, labelled S and C cells [81]. In this model, the neocognitron, the S cells were responsible for detecting local features whilst their outputs were spatially pooled by the C cells. In the original scheme, the neocognitron used an unsupervised learning regime to learn. Later versions moved to a supervised learning regime [82], which improved performance but arguably made it less biologically plausible [142].

Convolutional neural networks [131] use a similar structure to the neocognitron. However, constraints are placed upon the connectivity to enforce a local pattern between layers. In addition, weight sharing, where the weights of the feature detectors are copied across the layer, provide position invariance. This allows convolutional networks to perform particularly well with images without significant preprocessing [162].

Other models have used preselected feature detectors for the first layer. The HMAX model [188], which is described in detail later in this chapter, uses Gaussian or Gabor filters at the first stage with a MAX function to produce invariance over position and scale. The SEEMORE system proposed by Mel [150] uses filters sensitive to contour, texture and colour cues.

A more complex model, proposed by Oram et al. [173] , uses four stages of computation inspired by neurobiological data. In the first of these, boundaries between regions are computed and local features are grouped, effectively segregating features by object. The second stage involves selectivity for features of mid range complexity, such as T junctions and concentric rings. The third stage then selects for viewpoint dependent object attributes with the final level.

Whilst many of the network models are feedforward only, some models have been proposed that include a feedback based attention mechanism. For example, Amit et al. proposed a network model that has feedback across all layers, with units that can be primed to focus on a particular location within the image or on a particular feature [4]. Deco et al. [58] have similarly proposed a network model with feedback that allows both object- and space-based visual search.

Some authors have proposed models that contain the extra biological detail of expressing the outputs of units in terms of firing rates. Of particular amongst this group

of models is proposed by Thorpe et al., where information is encoded in the order in which cells fire [231]. Using evidence from ultra rapid scene categorisation the authors argue that there is insufficient time for responses to be encoded explicitly in the firing rates of cells. This is of interest from a computational point of view as network models are often concerned with the relative strength of features and thus encoding orders may be beneficial.

2.2.2 The relationship between network models and computer vision models

In many ways, the feedforward network models are very similar to bag-of-words methods used in computer vision. In both cases, the final representation is an unordered set of complex features. Whilst the two may employ different pooling functions, with bag-of-words methods typically summing over the image and network models taking the maximum response, the underlying structure is the same. Indeed many methods which utilise local histograms, including SIFT and HOG, are effectively utilising a layer of selective units followed by a set of pooling units much in the same way as network models do.

This means that the network models suffer from the same limitations of bag-of-words models, in that they are invariant to different spatial arrangements of the same features. A solution to this has been proposed by Ullman et al. using a fragment based approach [241, 240]. In a similar way to many methods in computer vision, Ullman et al. propose encoded an image as a set of patches or fragments. In order to encode the spatial arrangement of these fragments, the model proposes encoding the conjunction of overlapping fragments.

The use of overlapping fragments offers an interesting solution to encoding spatial structure without the need to explicitly encode spatial relationships between parts. Whilst some of the feature-based methods from computer vision use some loose form of colocation (e.g [144]), the use of conjunctions of overlapping features is not common. This may be because, in general, we are concerned with distinctive features which may be sufficiently sparsely distributed that overlapping occurrences are rare. Alternatively, it could be because the explicit encoding of conjunctions between features in a standard bag-of-words model would significantly increase the size of representation, as conjunctions would have to be counted for each pair of features.

This highlights the major difference between bag-of-words models and network models, which is the way invariance is introduced into the representation. Whereas bag-of-words methods tend to encode each location and then discard spatial information,

network models discard spatial information in stages.

The process of gradual discarding of spatial information presents the opportunity to encode more abstract representations. However, for a given size of representation it is not obvious whether this opportunity is used to best advantage in current network models.

2.2.3 Perceptual models

In addition to the many network models, which take their inspiration from neurophysiology, there are models which take their inspiration from perceptual studies. From the point of view of computer vision, perhaps the most influential of these has been the recognition-by-components system presented by Biederman [17].

Recognition-by-components proposes that objects can be represented by a small number of shape primitives called geons, which are shapes such as bricks, wedges and cylinders. Biederman suggest that a set of less than 36 geons is required for recognition and that for any particular object, the recognition of three geons and their relationships would be sufficient for entry level recognition [17].

In some ways, the recognition-by-components system has much in common with the parts and structure models described in the first section. However, whereas in other parts and structure models the parts are generally selected on the basis of recognition performance, the part in the recognition-by-components system also relate to the structure of objects themselves. This has the potential advantage of providing not only information about object identity but also about object function [189].

However, the disadvantage of using geons is that they in themselves may be difficult to detect. As the geons are three-dimensional viewpoint invariant parts, an effective implementation would require the recognition of all possible two-dimensional projections. Given the difficulties encountered with this form of recognition in computer vision, as opposed to modern feature-based methods, the accurate detection of geons within general images is likely to represent a significant obstacle. This raises the possibility that the geons that comprise an object may be more difficult to detect than the object itself, in which case their role in the first stage of recognition is questionable.

2.3 Image-based versus object-based methods

When considering suitable representations for recognition, we can make a distinction between image-based representations and object-based representations. In image-based

representations, there may be no prior knowledge of the location or scale of objects within the image. Therefore, the entire contents of the image are encoded. Object-based representations, on the other hand, are spatially localised and may encode features particular to the form of a specific object.

2.3.1 Characteristics of image-based and object-based representations

The majority of the recent methods used in invariant recognition in computer vision, as discussed in the first section of the literature review, involve image-based representations. Whether this is through a global histogram, such as in the bag-of-words methods, or in terms of an unordered list of encoded patches there is no explicit attempt to locate or represent individual objects within the image.

Methods from the first section that could be described as having an object-based representation include Eigenfaces, where the representation is a facial image projected onto to the set of principal components found from a set of face images. Object-based representations are found in biological systems. For example, the attributes of individual objects are found in single cell recordings in monkeys [171].

In considering the difference between image-based and object-based representations, a key aspect is the degree of prior knowledge. Image-based representations generally assume no prior knowledge as to the content or location of objects within the image. Object-based representations, however, must involve prior knowledge as to location, scale and possibly object identity.

2.3.2 Multiple problems in recognition

In the discussion so far, we have largely referred to recognition as one problem. However, problems within recognition can vary widely depending on the nature of the information sought and the nature of the prior information. For example, in a categorisation problem the desired information may be the probability that an instance of each of a set of pre-learnt object categories is present in an image. Alternatively, a recognition problem might be to estimate the pose of a certain object, such as a pedestrian.

Given the different nature of recognition problems, it is likely that different forms of representation will be suited to these different problems. For problems that involve determining what is present in the image, it may be necessary to encode the entire contents of the image. For problems that involve recognising an attribute of a particular

object, such as the pose of an object, it may be preferable to use an object-based representation.

In this work we are primarily concerned with the problem of determining what is present in an image, with little or no prior information. This can be thought of as the first phase of recognition. Sometimes referred to as immediate recognition [207, 206], this recognition problem is generally tackled with feedforward methods. This aspect of recognition is also found in biological systems, for example rapid classification [230, 231].

For the problem of immediate recognition, it may be that an image-based representation is preferable to gain information about the contents of an image or a scene. Object-based representations could then be used for additional recognition tasks.

2.3.3 Segmentation

If a representation encodes the contents of an image, then not only the object but the additional structure in the image will be encoded. In contrast, object-based representations will usually only encode structure related to the object. This raises the question of whether some form of segmentation is advantageous, necessary and possible in the context of immediate recognition.

Modern segmentation methods, such as superpixels [187] or normalised cuts[213] have been shown to be effective at assigning class labels to pixels in an image. More recently, image parsing [234, 99, 232] has attempted to combine elements of recognition and segmentation to produce a form of image understanding. These methods would clearly be of benefit in creating a representation for immediate recognition, providing they are accurate. However, if errors are made at the segmentation stage then the representation will be less effective.

Many feature based methods do not use segmentation(e.g. [143]) but instead rely on the detection of features specific enough to differentitate an object from the background. In addition, the positions and scales of individual features may be considered to link clusters of features to objects [144]. However, these methods are only successful when an object can be identified from an orderless features, in other words from its parts. Where object identity can only be established from considering the structure of a set of common parts, segmentation may be necessary.

2.3.4 View-based and structural representations

Image-based representations by their very nature do not encode aspects of individual objects within the scene and thus only encode the view of objects presented in the image. This raises the question of whether image-based representations can be effective in the recognition of alternative views of objects or whether, for a general purpose recognition system, it is necessary to have an object-centred representation. The debate over view-based versus object-centred, or structural, representations has ranged to different extents in the computer vision and biological modelling literature.

In computer vision, the argument is linked to the arguments between bag-of-words versus parts and structure models. The dominance of bag-of-word and feature-based methods, which are by their nature view-based, and the difficulties encountered with parts and structure models has led to a general preference for view-based representations.

For a feature-based representation to work when presented with novel views of an object, it must be possible to identify a sufficient number of features in the novel view to perform recognition. This can only happen if the features themselves are sufficiently invariant to depth rotation, so that they appear sufficiently similar in the novel view. Moreels et al. explored this characteristic of various features and demonstrated that features such as SIFT are reasonably invariant to viewpoint changes [154].

Whilst it could be argued that the emphasis on view-based methods in the computer vision community has been driven by relative performance, in the biological modelling literature view-based approaches have been driven by evidence that biological systems, especially the human visual system, are sensitive to viewpoint [227, 228, 62, 63, 182, 227, 238, 239, 32].

These biologically motivated models that employ view-based representations all require some means of comparing a novel view of an object to previously seen views. For example, in the model proposed by Poggio et al. [182], a novel view is transformed to a canonical view by first forming a hypothesis about the viewpoint. An appearance model is then computed which allows the novel view of the object to be transformed to a standard view.Likewise, Edelman et al. proposed a two layer network that was capable of developing multi-view representations in an unsupervised manner [63].

A strong argument in favour of these approaches is that they avoid the computationally difficult task of developing a structural representation of objects [182, 63]. However, whilst the majority of authors agree on the difficulty of developing such representations, some have claimed that they are likely to be far more powerful representations for recognition [18, 108, 107].

Hummel argues that the discussion over view-based versus structural descriptions should not just be about novel views [107] and that there are two key reasons why structural representations are necessary in recognition. First, a structural representation allows the evaluation of entities and their relations independently, whereas a view-based representation is holistic and offers no means of evaluating relations between parts. Thus, if recognition requires knowledge about these relations, a structural representation is necessary. The second reason provided by Hummel is that the representation of objects as combinations of parts is more efficient than the holistic view-based representations, as different combinations can be used to represent different objects whereas each view has to be stored separately.

This appears to be contradicted by Tarr et al. who investigated the effect of depth rotation using objects with differing numbers of discernible parts [228]. Their results showed that recognition performance dropped off with depth rotation for all objects. However, they found that the impact of rotation was less for objects of one part than for those with a greater number of parts. In addition they found that additional parts produce strong viewpoint dependency that was equivalent to objects with no distinct parts. However, Biederman has argued that the stimuli used in such experiments do not afford unique structural representations [18].

More recent work has suggested the possibility of both forms of representations being used simultaneously. For example Foster et al. used computer generated three-dimensional objects to test the effects of depth rotation [78]. With these stimuli they found that the results could best be explained by the action of two independent systems, one of which was view-based and the other structural.

Given the theoretical advantages of structural representations and the computational difficulties involved in computing them, it is tempting to suggest that they may employed where possible, with view-based representations being used in all other circumstances. However, it is not possible to establish whether this difficulty arises because sufficient structural representations have not yet been discovered or because they do not exist for all objects.

The relative absence of structural representations in computer vision could also be explained using arguments concerning computationally difficulty. However, it may also be due to the differing respective means of evaluation for computer vision methods and biological models. The standard datasets used to assess recognition performance in computer vision (e.g. [71]) may not require structural representations for high levels of performance.

2.4 Texture perception, recognition and representation

Many recent methods that have shown success in computer vision can be described as texture-based methods. This term has generally come to refer to the nature of the representation involved, rather than the task at hand. As a result there are texture-based methods for scene recognition and object recognition. In this section we provide a brief review of texture as a visual entity, its representation and perception.

2.4.1 What is texture?

A universal definition of texture has been elusive [48, 21], meaning that often texture can just be considered as 'stuff in the image' [2]. However, there are general aspects of texture that find agreement.

First, texture is considered to be made up of repeatable parts. This is central to the various forms of representation suggested for texture both in terms of modelling perception and computer vision, which model texture as a statistical distribution.

Second, texture has a repeated structure but with local variation. This is perhaps best expressed in terms of scale. At a certain scale, texture has obvious differences between locations whereas at a coarser scale texture appears to have a homogenous structure. Or alternatively, texture can be viewed as 'giving different interpretations at different distances and at different degrees of visual attention' [41].

This homogenous nature of texture at a certain scale leads to the idea of texture as defining regions in an image. In this view of texture, the boundaries between regions of different texture are important features that can indicate such properties as three-dimensional shape [84].

2.4.2 Texture perception

Much of the early work on texture perception involved investigating the ability of observers to discriminate pairs of textures. In these experiments investigators tended to use artificial textures, which gave them control over the characteristics that they considered important in texture discrimination [113, 114, 115, 9, 10].

Julesz investigated texture pair discrimination, or segregation, using textures made from dot patterns. First concentrating on characteristics such as the brightness and density of dot patterns, Julesz explained the texture pair discrimination results in terms

of clusters or lines of dots [113]. Similarly, Beck hypothesised that textural segregation could be explained by simple properties such as brightness, colour and the slopes of contours formed by simple features [9].

In further experiments by Julesz, this work was extended to a consideration as to whether texture discrimination was due to first, second or higher order statistics. In this context, first order statistics involved characteristics such as the density of dots. Second order characteristics involved pairs of dots, and so reflected traits such as the local orientation of dot pairs. Results from observers indicated that only first and second order statistics were utilised in texture discrimination and, in a model proposed by Julesz, it was suggested that texture discrimination involved comparison between the outputs of pairs of simple feature extractors [114].

Julesz then went on to suggest that texture discrimination was due to a small number of locally conspicuous features called textons [115]. This view was supported by Beck who argued that specific stimulus features were responsible for texture discrimination rather than second order statistics [10].

Texton theory has been highly influential in the representation of texture, however, whilst the term itself remains in use, its precise meaning has been more fluid, possibly because it has lacked a precise mathematical definition [272]. In the years since texton theory was first presented, many models have used the outputs of filter banks to represent the elements of texture.

Voorhees et al. used banks of Laplacian of Gaussian filters to detect blobs in images as textons [249]. Tuceryan et al. used Difference of Gaussian filters to create a Voronoi tessellation to model texture segmentation [235]. Many other models of texture segmentation have employed linear filters at the first stage, followed by a nonlinear stage, such as rectification, followed by an additional linear filtering stage. (See [125] for a comprehensive review).

2.4.3 A representation for texture

The notion of the basic element of texture is central to constructing a suitable representation for texture as a statistical distribution of parts. However, this still leaves the question of exactly what form the parts should take. In devising a representation for texture, the approaches can broadly divided depending on whether the goal is to model texture perception or to perform texture discrimination.

When attemtping to model texture perception, the general approach has been to use

human observers to group textures to produce some form of similarity matrix. This can then be used to determine a representation, usually with a very low number of dimensions. For example Rao et al [186] used multi-dimensional scaling on a set of 54 textures to produce a three-dimensional representation. With this approach, we might expect the dimensions of the representation to be meaningful in some way and, in the work of Rao et al., it was suggested that the three dimensions correlated with orientation, complexity and repetitiveness.

Whilst this approach is advantageous in that it produces meaningful dimensions, there are difficulties with establishing a representation using perceptual data. Perhaps the main difficulty is the consistency of the similarity measurements, in that context may affect the perceptual judgements [101].

A different appraoch has been taken in computer vision, where the goal has generally been to demonstrate the discriminative power of a representation using standard datasets. Whilst some of these have also been used in perceptual experiments [25], others are generally confined to testing in computer vision [55, 128, 98].

In designing a representation for use on these datasets, the focus of attention has tended to be on the invariance properties, pariticularly rotational and luminance invariance. Many different features have been proposed. For example, local binary patterns (LBP) [169, 170], which encode binarized local gradients. Other methods have used filters and a quantisation step [246], in the same way as the standard bag-of-words methods, as described previously.

Impressive performance has been achieved on the standard datasets, seemingly indicating the success of existing representations for texture discrimination. However, the power of a representation can only be measured in terms of the context of a particular task. It may be that the computer vision datasets only test fine similarities between textures, in that they are testing the ability of a method to match very similar textures but failing to test the dissimilarity of other texture pairs.

In order to produce a texture representation that reflects the wider similarity structure found in human perception, it is likely to require perceptual data. This issue is raised by Petrou et al. [181], who suggest that perceptual data should be used to select from a large number of computationally developed texture features.

2.5 Scale space and multiscale representations

Scale space, both in terms of the heuristic application of smoothing kernels for image processing and the mathematically formalised scale space theory, has provided the means to develop many different forms of representation. In this section we discuss the different ways of constructing scale space representations, their relationship to biological systems, and their application to problems in recognition.

2.5.1 Pyramid representations

The motivation for creating scale space representations comes from the notion that objects make sense at a particular scale. If an object is observed at too coarse a scale then the details that signal its identity may not be visible, and at a finer scale, the detail may relate to aspects of the object that are not related to its identity. For example, at a coarser scale a tree may appear as a blob whereas at a finer scale the detail of the bark is apparent. It is only at the appropriate scale that the object appears as a tree.

The same can be said for features that make an object. For example, an object may exhibit edges at many different scales. The outline of an object may be revealed by considering edges at a coarse scale, with other details being found at finer scales. Given this range of edges across scale, we can ask which of these edges are meaningful for recognition. In general, it is not possible to answer this question and thus we require a representation that encodes edges at all scales. This led to the first major form of multiscale representations called pyramid representations.

Burt [37] proposed an algorithm for simultaneously convolving an image with a family of smoothing kernels of a single parameter. This parameter, the scale parameter, designated the size of the smoothing kernel and thus the representation consisted of a stack of convolutions of the image with different size kernels. A subsampling procedure was then used to reduce the size of representation as the size of the smoothing kernel increased to create a pyramid structure.

The purpose of the pyramid was to produce a compact representation that could be used to detect image features across all scales. Burt proposed that simple feature detectors, such as spot, edge and bar detectors, could be applied across the structure with a computational cost not much greater than required for the original image.

In the early work on pyramid representations two different forms were proposed, the choice of which depended on the features of image to be enhance. The low pass pyramid used an approximate Gaussian smoothing kernel and a subsampling factor of

2 to produce the pyramid. It was later suggested that, for many applications, it was only necessary to store the difference between layers in the pyramid, which led to the Laplacian or bandpass pyramid [36].

Adelson argued that the Laplacian pyramid offered a superior representation to Fourier-based representations, such as those based on the power spectrum, as it enabled localisation in both the frequency and spatial domains [1]. However, this localisation in the spatial domain meant that pyramid representation were not shift invariant, meaning that pattern matching required convolution.

Crowley used a bandpass pyramid, termed the Difference of Low Pass (DOLP) Transform, for shape representation [49]. Rather than using the whole pyramid, Crowley proposed extracting salient features, such as local peaks and ridges, at each scale. These were then linked together to form a multiscale graph representation which offered a more compact encoding than the standard bandpass pyramid. However, an effective means of comparing such representations for recognition was not provided.

A key consideration in any pyramid representation is the choice of smoothing kernel. For low pass pyramids, some authors argued that the kernel should be Gaussian-like [37, 36]. Meer argued that the optimal choice of filter should be as close to the idealised low pass filter as possible [149]. In practice, the most common choice was the binomial kernel which served as a good approximation to the Gaussian filter [37, 36, 49, 50].

Many variations of pyramid representations have been proposed. Adelson et al. [3] investigated using local orientation tuning within the context of pyramids. This led to the proposal of two new form of pyramids, Quadrature Mirror Filter pyramids and steerable pyramids. Gluckman et al. explored higher order image pyramids as a nonlinear generalisation of the Laplacian pyramid [85].

In recent work, Lazebnik et al. [129] have presented a pyramid representation that has much in common with modern feature-based methods used for object categorisation in computer vision. Whereas traditional pyramid representations use a smoothing kernel to create a stack of images, Lazebnik et al. use varying apertures to create a stack of histograms of local features. Thus, in contrast to traditional methods where the smoothing kernel acts on intensity values, this can be seen as a kernel operating on the outputs of local feature detectors. The representation has been shown to be particularly effective because it makes use of the pyramid match kernel [87], which allows the representation to be used in a support vector machine.

However, the pyramid matching scheme proposed by Lazebnik et al. imposes a

fixed geometrical structure on the image. This means that, as with other pyramid-based methods, the representation is a way of encoding both the contents and structure of an image, as opposed to bag-of-words methods which only encode the contents. This has implications for the use of such representations in invariant recognition, where looser spatial structures may be required.

2.5.2 Scale space theory

Pyramid representations were appealing for their simplicity and their compact nature, which made their use popular for a period of time. However, as Lindeberg states [136], the algorithmic nature of the pyramid formation made analysis difficult, and the representation depended upon the particular subsampling regime employed.

In contrast to the mainly heuristic approaches to scale space used in pyramids, Gaussian scale space theory aimed to develop an axiomatic scale space representation which was independent from the sampling regime. At the core of this formulation was the notion that the visual front end should be uncommitted [191], in that there is no model involved.

First presented by Witkin [257] and developed by Koenderink [122], the requirements of scale space were laid down as a set of axioms. First, the creation of scale space should not involve the creation of any additional structure, which means that no new maxima or minima should be created along the scale dimension and that maxima and minima in the image must not be enhanced. Second, the construction should be shift invariant, as an uncommitted system should show no preference to any particular location. Third, the construction should be isotropic, thus showing no preference for orientation. Fourth, no preference should be shown to any particular scale. Finally, as the system assumes no particular model, the construction should be linear. Witkin and Koednerink showed that these axioms lead to the unique solution of the Gaussian kernel as the kernel for scale space construction, a result which can be reached through various routes [257, 122, 166, 190].

Gaussian scale space theory provides a representation that is three-dimensional and continuous in the scale dimension. In order to use this representation for recognition, it is necessary to have a means of detecting salient structure. This can be done through some form of keypoint detection method, such as searching for extrema of the Difference-of-Gaussians [143]. Lindeberg has also provided a method of automatic scale detection and demonstrated its use for the detection of features such as edges, blobs and ridges [138].

Whilst scale space is often used for its scale invariance properties, in that it can be

used to detect single scale structures found at any scale, there is another form of structure in scale space. Referred to as deep structure [257, 122], this is structure that occurs across multiple scales, and thus can be considered as a three-dimensional feature in scale space. The central idea behind deep structure is that, in certain circumstances, it is more informative to consider coarse and fine structure at a particular location together rather than separately, as in the case when simple scale invariance is sought.

Bergholm has described a method of using this deep structure to enhance or focus edges by linking structure across scales [15] . Perona et al. used form an alternative form of scale space, based upon anisotropic diffusion, to detect the exact locations of coarse edges for image segmentation [178]. Vincken, with the hyperstack algorithm, provides a way to create a tree like structure in scale space for use with segmentation [247].

2.5.3 Relationship to biological vision

Biological vision systems are capable of perceiving objects at different scales. For example, humans are able to perceive different objects within a scene that occur at different scales. Therefore, the human visual system must, in some way, be operating at multiple scales [265, 104, 74, 190, 191]. However, this does not necessarily imply that it uses Gaussian scale space.

Some authors have suggested, using the theoretical arguments on an uncommitted front end system, that we could expect the early stages of the visual system to use Gaussian scale space [136, 123]. This is supported by evidence of cells in the early visual system which can be modelled by Gaussian functions and their derivatives. For example, the Laplacian of Gaussian, or Mexican Hat, is often used as a model for the centre-surround receptive field [148]. Neurophysiological studies by Young et al. have shown that suppositions of Gaussian derivatives are a good model for cells in both the retina and cortex [263]. However, others have proposed alternatives, such as the Gabor filter, as more accurate models [56].

2.5.4 Scale space and current methods in invariant recognition

Many of the methods currently used in computer vision operate in scale space. For example, in the SIFT method [143], keypoints are found in scale space by locating the extrema of the Difference-of-Gaussians function. The SIFT features are then encoded at the scale of the keypoint. Amongst the network models, some models use filters of different sizes as the first layer. For example, in the model presented Riesenhuber [188], the first layer consists of Gabor filters at various scales.

However, in these representations, scale space is used to introduce scale invariance. Whilst this is a desirable trait at the object level, such representations also introduce invariance to the relative scale of individual features. Thus, there is no encoding of the deep structure in the scale space, which requires features at different scales to be linked.

For texture recognition, Varma et al. [246] have proposed a texton based representation that combines the outputs of oriented filters across 3 scales, thus producing a representation that links fine and coarser scales. However, orientation information at each scale is discarded, with only the maximum response being used.

Each of these three methods utilises oriented filters at a range of scales. The difference between them can be seen in terms of how links between the local orientations are preserved. However, none of these representations link orientation information at different scales.

2.6 Methods review

In the next section of this chapter we present a summary of four current methods that have shown strong performance in applications in visual recognition. This is not intended to be a comprehensive review of the literature available on the general problem of object recognition, but instead serves as a summary of the methods that guided the investigation presented in this work.

In choosing these methods we concentrated on those that are of current interest, both in terms of the development of the theory and in their continued relevance to cutting edge application. We also wanted to concentrate on those methods that were based upon a relatively simple basic idea, which had been applied in multiple application areas.

For each method we provide a summary of the architecture, then a more detailed account of the way in which the method introduces different forms of invariance. We then briefly discuss the way in which the method has been applied.

2.7 Scale Invariant Feature Transform

The Scale Invariant Feature Transform (SIFT) has been one of the most widely used methods in object recognition since it was first presented by Lowe in 1999 [143]. At the heart of the method is an encoding scheme of oriented gradients. Such basic features have

been used in computer vision before the arrival of SIFT (e.g. [79, 80]) and form part of a family of features encoding local orientation that have been of interest to the wider vision community since the experiments of Hubel and Wiesel [106]. However, it was the arrival of SIFT, and the encoding of local oriented gradients within a template of histograms, that produced an encoding scheme that led to the widespread use of such schemes.

The various steps of the SIFT descriptor are explained below.

2.7.1 Architecture

2.7.1.1 Keypoint Localisation

The SIFT method operates in scale space [122, 136, 137], where an image is convolved with a Gaussian kernel. The scale space image is:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

where:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$

In the SIFT method, as presented by Lowe, the whole scale space is not encoded. Instead, keypoints are first detected, and the region around these is then encoded. To detect keypoints the Difference of Gaussians is used:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$$

The keypoints are positioned at the local maxima and minima of this function. This is done by comparing each location to its eight neighbours. In some implementations keypoints are further localised by interpolation [26, 144]. The result of this process is a set of keypoints with a three-dimensional position for each.

2.7.1.2 Orientation Assignment

The next step is to assign an orientation to each keypoint. This is done by computing an orientation histogram in the region around the keypoint, to determine the dominant orientation. Using the scale of the keypoint, L, the orientation, θ, then the magnitude, m, of the gradient is computed using:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$$
$$\theta(x, y) = \tan^{-1} \frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}$$

The gradient is then weighted using a Gaussian window centred on the keypoint and a histogram of the weighted gradients is then formed by summing for each of 36 orientations. The largest bin is then selected as the orientation for that keypoint. If other bins are within 80% of this magnitude then they are also selected and another descriptor is calculated.

2.7.1.3 The Descriptor

To form the descriptor a grid is placed over the region surrounding the keypoint. For each subregion within the grid, a local histogram is calculated by summing the magnitudes of each of the orientations, as calculated in the previous step. In the standard implementation of SIFT [144], this grid consists of 4x4 subregions computed from a 16x16 array around the keypoint. Typically, for the computation of the descriptor, the number of orientations is reduced to 8, giving a total descriptor size of 128 bins.

In order to try and avoid the effects of the boundaries between the subregions, entries for each histogram are weighted by a linear function of their distance from the centre of the subregion. Finally the whole descriptor is thresholded, to ensure no bin is greater than 0.2, and then normalised.

2.7.1.4 Classification

In the SIFT method, an image is generally encoded into a large number of SIFT descriptors. The exact depends on the number of keypoints that are located within the image. Thus the encoding is a list of SIFT descriptors, with a position and scale for each. In order to classify an image into one of a set of previously learnt classes, each descriptor is compared to a reference set of descriptors computed from training data.

The matching of descriptors involves finding nearest neighbours in the 128 dimensional SIFT space, where the number of reference descriptors may number in their tens of thousands. In order to make this computationally efficient, approximate methods are used such as the Best Bin First algorithm [11].

The result of the matching process is a set of labels, from positive matches, and a position and scale within the test image for each. In order to convert this into a classification, a voting system is used to determine clusters of positive matches with a common position and scale.

2.7.2 Invariance

- **Translation** When considering invariance to translation, we have to differentiate between the SIFT descriptor and the SIFT method. The descriptor itself, which consists of the template of histograms, will be invariant to the shifting or oriented gradient features within subregions, but will not be invariant to movement of features from one subregion to another.

  The SIFT method, as with other bag-of-words approaches is invariant to shifting of individual descriptors within the image. Thus, if the spatial arrangement of descriptors is reorganised the method will not detect this.

  Overall then, we have different degrees of invariance at different levels. A small tolerance to shifting of basic features, then a rigid spatial structure of local histograms followed by complete invariance to the spatial arrangement of descriptors.

- **Rotation** The SIFT descriptor develops rotational invariance by aligning itself to the dominant orientation. Thus, a rotated version of the same image would lead to the same computed descriptor. However, because the dominant orientation of the keypoint is calculated by taking the maximum value of the histogram bins, there is the potential for slightly different images to produce very different descriptors.

  If, for example, we consider the SIFT encoding of an 'L' shape where there are two possible dominant orientations. Depending on the relative length of each of the line segments, we may get a different dominant orientation, and thus a very different descriptor. This is countered to a certain extent in the SIFT method by computing multiple descriptors where there is an ambiguous dominant orientation.

  Within the descriptor itself, there is no rotational invariance, as both the gradient features and the grid structure contain orientation information. However, rotational invariant forms have been developed, such as RIFT[126], where the local histograms are arranged in disk segments and orientations are calculated relative to each subregion.

- **Scale**

  The SIFT descriptor is calculated at a single scale, which is the scale of the keypoint. Thus, invariance to scale can only come about by the keypoint selection step.

- **Clutter**

The SIFT method attempts to handle clutter at the level of descriptor matching, so that descriptors belonging to the target will be positively matched and descriptors arising from clutter will not. There is the potential for descriptors from the clutter to be incorrectly positively matched, however Lowe proposes handling this by looking for positive matches with a common position and scale [144].

2.7.3 Application

Whilst SIFT was originally proposed with the keypoint localisation stage, it has subsequently been shown that superior performance can be gained from dense encoding [168]. Thus, it is the SIFT feature descriptor which forms the enduring aspect of the method, many variations of which exist [86, 51, 218]. These includes descriptors that reduce the dimensionality of the descriptor for more efficient matching. Such methods include PCA-SIFT, which applies a Principal Components Analysis stage to the descriptor has been proposed for image retrieval [120], Speeded Up Robust features (SURF) [8] and the Gradient Location Orientation Histogram (GLOH), which introduces granularity to histograms [151].

2.7.4 Discussion

The various different implementations of SIFT have helped to demonstrate which aspects of the method contribute to performance. The demonstration that keypoint localisation is unnecessary[168] and the use of SIFT descriptors within multiple classification frameworks demonstrates that the enduring component of the SIFT method is the SIFT descriptor itself. This is also the most interesting aspect of the method in the current consideration for our work, where we are investigating ways of combining basic features into a suitable representation for visual recognition.

In the case of the SIFT descriptor these basic features are oriented gradients. The first question then that we might want to ask, is whether this first level of representation contains sufficient information for recognition or whether, for some tasks, we may have already lost crucial information at this stage.

This is a difficult question to answer in general, in the absence of perfectly performing recognition systems. However, in certain recognition problems, such as with the digits and letters used in the investigation in this work, the identity of the digit can still clearly be recognised from the encoded image which implies that we have not discarded crucial information at this stage. This is illustrated in Figure 2.2 where the identity of the character encoded in oriented gradients is still clearly visible. However, in other applications this may not be the case and we may need higher order features[264]. This

is explored in this work to a certain extent, through the use of alternative sets of basic features.

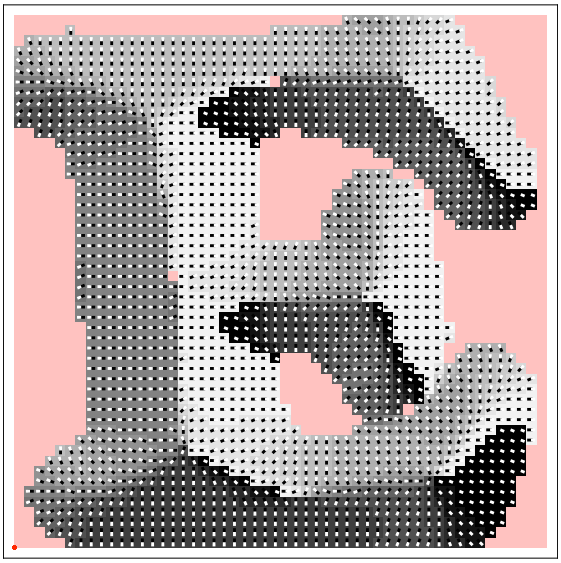


Figure 2.2: An image encoded in oriented gradients.
The orientation is represented by shades of grey with flat regions shown in pink. From visual inspection, the shape of the character is still easily identifiable.

Assuming that we are dealing with an application where oriented gradients provide a sufficient basic feature set, the next question to ask is, to what extent does the SIFT descriptor capture useful information about class identity? To answer this we need to look at the information that is encoded in the SIFT descriptor.

The first thing that the SIFT descriptor encodes is the distribution of orientations across the whole region. Although this is not explicitly included in the descriptor, it could be reasonably well estimated from the individual histograms. Secondly, SIFT encodes the spatial relationship between these orientations, which is captured by use of the grid of histograms.

In order for a SIFT descriptor to be matched to another both these types of information must be similar for the two descriptors. If we consider the effect of a change in the distribution of orientations across the whole descriptor then in the SIFT encoding space we would expect the distance between the two descriptors to gradually increase.

However, this is not the case with the spatial information. As oriented gradients move from one local histogram region to another we would expect to see a sharp jump in the distance between the two descriptors. Thus if we imagine an object within the region encoded by a descriptor, as this object begins to deform the descriptor moves relatively

large distances in SIFT encoding space.

This is likely to mean that SIFT descriptors will be matched providing that the spatial arrangement of orientations within the descriptor region is very similar. In this case SIFT descriptors would be effective in recognition tasks where instances of the same class contain distinctive small patches.

There are two occasions when SIFT is likely to fail. First, when instances of the same class contain distinctive features that do not fit the rigid spatial structure of the SIFT descriptor. This may be the case when the distinctive feature of the class is more associated with its shape. The second occasion is when distinctive features that fit the SIFT structure are present, but the distinctive feature of the class is in the spatial arrangement of the descriptors themselves. In this case SIFT descriptors could be used in conjunction with a spatialisation scheme, such as pyramid matching. [129].

A common approach in recent attempts at object recognition has been to use SIFT descriptors in conjunction with other sets of features and a suitable method for selecting the features that are appropriate for each class. (e.g. [219]). In this way we can see the SIFT descriptor as forming part of suitable representation for visual recognition tasks, but not a complete encoding scheme that can be expected to be of use in all applications.

2.8 Histograms of Oriented Gradients

The Histograms of Oriented Gradients (HOG) scheme [53] has many similarities with SIFT. However, it is useful to consider it as a separate method because there are key differences between the two, both in their construction and application, that may help shed light on which aspects of each method are contributing to their performance.

As evident from the name, HOG uses oriented gradients as the underlying feature, though generally the calculation of these is different from SIFT. These are then grouped into a template of histograms, as with SIFT. The details are given in the next section.

2.8.1 Architecture

The architecture of the HOG method consists of three key steps. The first is gradient computation, where the underlying local gradients features are established. The second is the grouping of these together to form a template of histograms. The third is the normalisation of each histogram to form the final HOG encoding. These steps are illustrated in Figure 2.3.

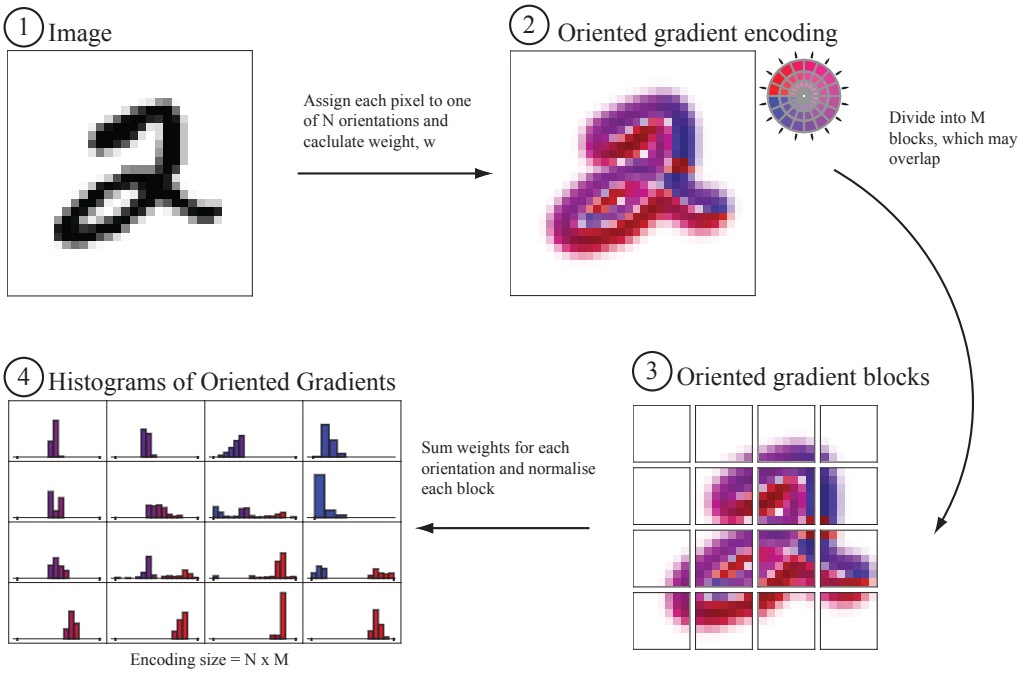


Figure 2.3: The different stages of the HOG encoding.
The image is first encoded into oriented gradients, then divided into locally normalised blocks.

2.8.1.1 Gradient Computation

The first stage in the HOG scheme is the calculation of oriented gradients at each point in an image. In the method as presented by Dalal et al. [53, 54], this is done only at the finest possible scale, and thus the HOG scheme does not encode a scale space representation of an image.

At each location, the gradient is calculated using simple 1-D centred masks, which gives an orientation for the maximum gradient and a corresponding magnitude for the gradient. Typically, the masks have a scale of 0.

2.8.1.2 Spatial Binning

Once a gradient orientation and magnitude has been assigned to each location, histograms are computed over local regions, or cells. Typically, the histograms are calculated by summing the magnitudes for each orientation bin across the cell, though in some cases the square root or the square of magnitude is used.

When computing the histogram, orientations can be signed or unsigned. The signed gradient is an arrow-like feature, which indicates the direction of the gradient across the full $360°$. The unsigned gradient indicates the direction across a $180°$ range and is

invariant to whether the gradient is from light to dark or dark to light. In the version of HOG presented by Dalal and Triggs, nine unsigned orientation bins are used.[53]

2.8.1.3 The Descriptor

To compute the final descriptor, the histogram for each cell is normalised so that the total gradient magnitude is the same for every cell. This step is seen as being critical in dealing with differences in contrast the image. The histograms are then concatenated to make the final descriptor.

2.8.2 Classification

As with SIFT, the classification of an image is performed by first classifying each HOG descriptor using a set of labelled reference descriptors. In the first presentation of the method, when it was tested on pedestrian detection, this was done using a Support Vector Machine (SVM) to determine whether a particular HOG descriptor came from a pedestrian scene or not.

2.8.3 Invariance

- **Translation**

  As with SIFT, we can consider invariance to translation of features within the descriptor invariance to shifts in the position of the descriptors. In the first case, the descriptor is invariant to small changes in the position of oriented gradients within cells. However, the concatenation of the cell histograms leads to a fixed structure, meaning that movement of oriented gradients across cell boundaries will lead to a very different descriptor.

  When HOG descriptors are compared, position information is generally not used. Therefore HOG can be seen as being completely invariant to different spatial arrangements of the same HOG descriptors.

- **Rotation**

  Unlike SIFT, there is no mechanism within the HOG descriptor to develop invariance to rotation. However, variations have been developed which introduce invariance, such as RIFF [226] or polar-HOG [252].

- **Scale**

  The HOG descriptor is calculated at a single scale, which is usually zero. However, as the oriented gradients are calculated using filters, which can be of any size, it would be possible to compute the HOG descriptor at a scale selected by some form of keypoint selection process which would introduce invariance to scale in the same way as SIFT.

- **Clutter**

  The HOG descriptor itself will be sensitive to clutter. However, clutter within an image can be handled by computing multiple descriptors in the same way was with SIFT, where providing there are a sufficient number of true positive matches the descriptors arising from clutter can be ignored.

2.8.4 Application

HOG has been most commonly used for detecting and localising people in images [53, 271, 112, 83, 254]. In addition it has been used in many object recognition schemes in conjunction with other features. For example the top performing scheme in the VOC 2011 challenge used HOG descriptors at three scales in combination with other feature sets, including SIFT [219].

Variants of HOG have been produced, such as the rotationally schemes mentioned above, or schemes that compress the HOG descriptor. [39]

2.8.5 Discussion

The HOG descriptor can be thought of as a generalisation and a simpler version of the SIFT descriptor. It does not involve many of the steps that were included in the original SIFT method, which suggests that it is the template of histograms underlying both methods that is providing the performance.

Like SIFT, the HOG descriptor imposes a rigid spatial structure in the form of the template of histograms. This makes it well suited to applications where classes consist of relatively rigid shapes such as pedestrian detection, which also involve objects at a common orientation.

The use of HOG descriptors at multiple scales, as in [219], raises the possibility of representing a spatial arrangement of HOG descriptors by using another descriptor at a coarser scale, which would allow both fine detail and the overall shape of objects to be captured. However, this would only work for classes that conform to the rigid spatial

structure at both scales, which may further limit the range of classes that can be well represented using HOG.

2.9 Shape Context

The Shape Context descriptor [12, 156, 13, 157], differs from the methods previosuly described in that it encodes vectors between features, rather than features that occur within a certain subregion. In the scheme, orientation information is encoded, but not at the level of the basic feature. Instead, the orientation of vectors between unoriented edge features is encoded. This makes it an interesting method to compare to SIFT and HOG.

2.9.1 Architecture

There are three key stages in forming the descriptor. Firstly, edge features are detected within the image or region. Second, for a certain feature within the image, vectors to all other edge features are calculated. Finally, a histogram of these features is calculated using a log polar grid. Each of these steps is described in detail below.

2.9.1.1 Point selection

The first stage is to find edge features in the image. There are many edge detectors available but typically this step is performed using a Canny edge detector[38]. This gives rise a to a large number of locations within the image and so, for ease of calculation, a further sampling step may be employed where a subset of the edge points is randomly selected.

2.9.1.2 Vector calculation

The next stage is to calculate vectors from the location of a given edge feature within the image to all other points selected in the first stage. Each vector is typically calculated both in terms of distance in the image plane and the orientation of the vector. As the descriptor is calculated from a given point, the possible orientations cover the full 360°.

2.9.1.3 The Descriptor

The final stage in calculating the descriptor is to capture the distribution of vectors using a histogram. This is a two-dimensional histogram with axes for orientation and the log polar distance. Typically, the histogram will be divided into 12 orientations and 5 log

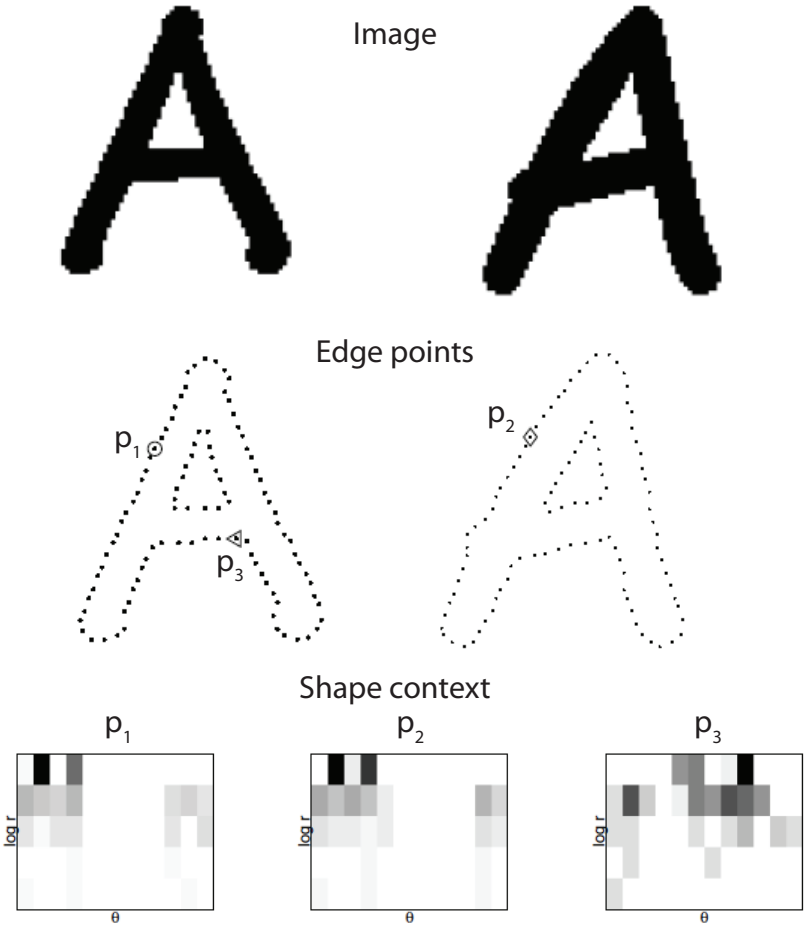


Figure 2.4: Shape contexts for two different letters (adapted from [12]).
The encoding for the two corresponding points, p_1 and p_2, are very similar whereas at point p_3 the encoding is very different.

distance regions [156]. Finally, the distances in the histogram are normalised according to the mean distance, to give a descriptor that is invariant to scaling of the image. The process is illustrated in Figure 2.4.

2.9.1.4 Matching

As shape context descriptors are computed at a certain location within the image, rather than over a region, they require a correspondence matching process which determines the closest pair matches out of a set of descriptors for two objects. This is generally split into two different processes. Firstly, *fast pruning* [156] is used to select a small subset of descriptors as a potential match. A more detailed matching process is then performed to calculate the distance between descriptors. This distance, sometimes referred to as the cost, is calculated using:

$$C_{i,j} = \tfrac{1}{2} \sum_{k=1}^{K} \frac{(g(k) - h(k))^2}{g(k) + h(k)}$$

where $g(k)$ and $h(k)$ are the K bin histograms at point i and j respectively.

2.9.2 Invariance

- **Translation**

  As the descriptor is calculated at a specific location, invariance to translation can only be achieved by using correspondence matching in the standard bag-of-words sense. Within the descriptor itself, a small degree of translational invariance is allowed as individual edge features can move within the bins of the histogram.

- **Rotation**

  The descriptor as it comes is not rotationally invariant. However, invariant versions have been proposed [13, 260].

- **Scale**

  The scale of the descriptor depends on the scale at which the edge features are calculated and the size of the grid which is used for the histogram.

- **Clutter**

  The way in which the descriptor can handle clutter is through the standard bag-of-words approach, in that recognition proceeds through positive matching. Therefore, shape contexts that are calculated from clutter will only cause confusion if they are erroneously positively matched with labelled shape contexts.

2.9.3 Application

When first presented Shape Context was first tested using the MNIST dataset, achieving leading performance [12]. Along the same lines Mori have shown success in recognition of text within CAPTCHA [158].

Shape context has also been widely used for other applications involving shape recognition, outperforming other methods in pedestrian detection [204]. Hand shape recognition [172] and pose estimation [5] and pose estimation in video [94] have also successfully been tackled with Shape Context.

2.9.4 Discussion

Shape Context differs from both SIFT and HOG in that it does not encode oriented gradients, but the oriented vectors between unoriented edge features. The first difference between the schemes therefore comes from the different set of basic features that are used.

The second difference is that, whilst SIFT and HOG involve a template of histograms, Shape Context employs a histogram of vectors. This might first seem to overcome the issues of the rigidity of the spatial relationships encoded in the SIFT and HOG descriptors. However, in Shape Context, the histogram is not a count of the number of occurrences of a set of features within a certain region, but a measure of the distribution of vectors between edge features. The histogram therefore encodes the spatial relationships between the features.

As with SIFT and HOG, the way in which these spatial relationships are encoded makes it tolerant of small changes in the position of edge features but when the edge features move a sufficient amount to make the vectors cross the bin boundaries in the histogram the result is a very different encoding. Whilst the use of the log distance in the histogram will make the descriptor more tolerant of movement of features in its periphery, this brings the extra problem of requiring the centre of two descriptors to be well matched.

2.10 HMAX

The HMAX model [188, 208, 205, 210, 207, 209] is one of a family of models that take their inspiration from the characteristics of cells in the primary visual cortex as first shown by Hubel and Wiesel [106]. Beginning with the neocognitron [81], many variations have been suggested using the same basic architecture. The concensus between these models has given rise to the label of the 'Standard Model' of object recognition [205] as a benchmark biologically plausible system recognition system. These models are of interest in our work because it combines basic features in a very different way to SIFT, HOG and Shape Context.

2.10.1 Architecure

Within this family of models, HMAX has gained prominence in recent years as a biologically plausible model for visual recognition. Drawing inspiration on further physiological data [140, 180, 179, 173], the HMAX hierarchy consist of five layers, as shown in Figure 1.

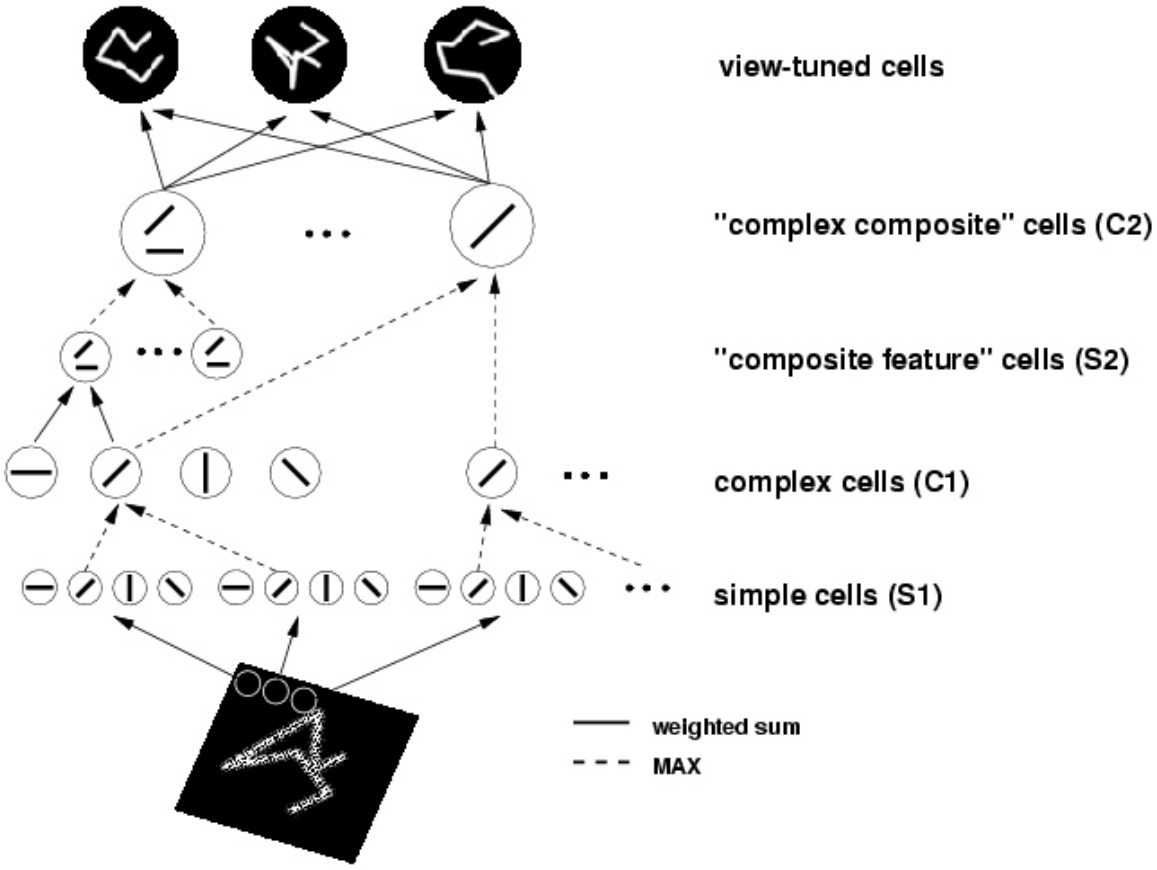


Figure 2.5: The five layers of the HMAX architecture, from [205]
The model consist of alternate layers of simple cells, which are tuned to a particular feature, and complex cells, which pool responses across banks of simple cells.

- **S1 Layer** The first input layer in the model is equivalent to the simple cells drawn from Hubel and Wiesel, which are sensitive to the local orientation. Typically a bank of Gabor filters is used, as in [188] with a typical implementation using four orientations and eight different sizes of filters. A bank of such filters is applied at every point in the image. This layer is the first stage of selectivity, as each cell is tuned to a particular feature of the image. The particular parameters of the Gabor filters are taken from physiological data relating to the 'simple' cells found in the primary visual cortex. In some implementations, alternative oriented filters have also been used, such as Gaussian Derivative filters as in [147].

- **C1 Layer** The C1 layer is the first pooling layer, equivalent to the complex cells in Hubel and Wiesel, achieving invariance to position and scale within a small section of the image. Several inputs of S1 units, which are neighbours in position and scale, act as inputs to a single C1 unit. Invariance is achieved by applying a MAX function over these inputs.

- **S2 Layer** The second tuned layer takes inputs from the C1 Layer and gives a response dependent on the similarity of the image patch to the preferred stimulus of

the S2 unit. This preferred stimulus can vary in its complexity, according to the particular implementation, as discussed further in the section on *Learning* below. There is one S2 unit for each preferred stimulus at each point in the image.

- **C2 Layer** The second pooling layer consists of units that take inputs from S2 units with the same preferred stimulus across all positions and scale. The MAX function is then applied, as in the C1 layer, giving the maximum response of that preferred stimulus across the whole image. The total number of C2 units is determined by the particular implementation and is independent of the size of the original image.

- **VTU Layer** At the top is a level of view tuned units (VTUs), or view tuned cells, which take their input from the C2 layer. These are tuned units where the optimal stimulus usually corresponds to a particular view or partial view of an object.

Learning Learning can occur at different stages in the HMAX model. In all implementations the tuning of the VTUs must be learnt, so as to recognise the objects in any particular task. This is always done through a supervised learning process, where labelled training examples are used to determine the weight matrix on each VTU. In this case each VTU represents one object.

In addition learning can occur at the S2 level. The original implementation sets the optimal stimulus of each S2 unit as a simple combination of four C1 input units, and then takes all possible combinations to form the S2 layer [188]. Alternatively, more complex combinations of C1 units can be selected at random, as in [210], or an alphabet of S2 features can be learnt [209]. In this case, learning occurs in an unsupervised manner by exposing the system to a large number of unlabelled images, and then performing a clustering operation on combinations of C1 features. Typically, implementations employing this will select around 1000 unique S2 features.

2.10.2 Invariance

- **Translation** Each unit in the C2 layer pools inputs from S2 units across all positions and should therefore detect the presence of its optimal stimulus anywhere in the image. For an object comprised of such features, the model should also be completely translationally invariant, providing all such features are not dissected. This is tested in [206] and, subject to minor errors appears to be correct, agreeing with equivalent findings from translational invariance in physiological experiments. [109]

- **Rotation** There is no mechanism built into the HMAX model to achieve rotational invariance. However, if each S1 unit partially responds to oriented features within the image that are slightly off the preferred orientation of the unit, a rotation of the image will not result in a sudden falloff in performance. We would therefore not

expect the HMAX model to be rotationally invariant, but would expect it to be able to handle small rotations of an image. This, however, has not been shown.

- **Scale** As with position, each C2 unit pools inputs from S2 units across all scales and therefore each C2 response should be absolutely invariant to scale, providing all relevant features can be detected using the scales of filters selected. This was tested, as before, and found to be approximately correct. [206]

- **Clutter** Units in the C2 layer respond when their optimal stimulus is present and as there is no competition between units in each layer, they should respond regardless of any other features that are present. Providing that each VTU is tuned to presences of C2 features only, and not absences, we would expect that the HMAX model should be relatively robust to clutter. This is dependent on using a suitable learning method to ensure that the tuning for each VTU only select relevant C2 features and ignores all others. However, this only ensures a positive response when the object is present. As the area of clutter increases the chance of each of the C2 units being present increases as well, increasing the chances of falsely reporting the presence of an object. Thus we might expect recognition performance to fall in the presence of clutter. This is indeed the case, as reported in [209] where classification performance dropped by approximately half as the relative area of clutter to object was doubled.

- **Classification** In addition to being invariant to translation and scaling at an object level, the HMAX model is invariant to a change in the position of individual features. Each C1 unit pools input across several positions and multiple scales, and therefore small changes in the locations of each S1 feature will still produce the same output of the C1 unit. Similarly, changes in the location of S2 features will leave the output of C2 units unchanged. If some intra-class variation of objects is down to small changes in position of individual features then HMAX should use this to enable greater classification performance.

2.10.3 Discussion

The HMAX model performs well on the five class recognition problem, and it clearly demonstrates a high degree of invariance to translation and rescaling at the object level. It also possesses an appealingly simple architecture, taking its inspiration from biological vision and repeating the simple two stage tune and pool process.

The performance on the two class problems using objects taken from the Caltech101 dataset show very good results. Performance on the full 101 class problem does not compare favourably with other methods, for example in [129] an overall score of 64 ± 0.7 % was achieved on the same task. However, it should be noted that the Caltech101 dataset contains objects that occupy a large proportion of the image and generally have a common

view, orientation and size. Such a dataset may not be a particularly good test of fully invariant recognition, as considered in this thesis. Methods, such as in [129], are not designed to be invariant in this manner and may capitalise on the nature of the Caltech101 images.

Despite this, there is clearly a large drop in performance in the shift from the 2 class problem to the 101 class problem. This could be for two reasons. Firstly, the C2 features used in the HMAX model may be consistent enough to distinguish certain classes of objects in the 2 class problem, but not others. The five classes that are chosen, Cars, Faces, Airplanes, Motorbikes and Leaves may be classes with relatively little intra-class variation. If the other classes in Caltech101 contain a far higher degree of intra-class variation then this would clearly drive down performance in the full 101 class problem. In this case the C2 features may not be consistent enough descriptors of all objects in Caltech101.

An alternative explanation is that the C2 features by themselves are not discriminative enough to perform well in a problem with a high number of classes, even though they are consistent within each object class. In this case it may be possible to improve performance by creating more complex features which are combinations of C2 features, by introducing a new tuned and pooling layer, S3 and C3. However, as the number of layers is increased the learning task becomes far more difficult and it is far from certain how this could be overcome.

As discussed above, there is no explicit mechanism with the HMAX model to achieve rotational invariance although it may be relatively insensitive to small changes in orientation. The objects in Caltech101 tend to appear at a common orientation, and so HMAX may be able to handle the small variations that exist, however as there is no measure either of how orientation varies in Caltech101 nor how sensitive HMAX is to small changes in orientation it is impossible to determine whether rotation has an impact on the performance.

An additional consideration is whether the HMAX model suffers by being invariant to different spatial combinations of the same C2 features. As it is only the presence of these features that is considered, any rearrangement of the same features will produce the same encoding. Thus if two object classes were, broadly speaking, different arrangements of the same underlying features the HMAX model might not be able to distinguish between them. This may be solved by adding an additional layer of tuned and pooling units, but, as discussed previously, this presents problems for the learning system.

Alternatively, if enough C2 features are used they may sufficiently overlap each other to 'lock in' one spatial configuration, as in [241]. However, there is no evidence that performance can be improved by simply increasing the number of C2 features.

Overall the HMAX model provides some encouraging results, but there appear to be clear limits on its performance. its unclear whether these can be overcome within the framework of the model or, as suggested in [206], this represents the limit in performance

possible with a purely feedforward architecture.

We also have to question exactly which aspects of the HMAX model contribute to the results when they are strong. For example, how important is the use of the MAX function? How critical is the choice of the S1 features? We could use other oriented features at this layer, or expand into basic second order features. Without a clear understanding of these issues, it is difficult to establish how far the HMAX framework can be taken.

[241]

2.11 Basic Image Features and oriented Basic Image Features

2.11.1 Basic Image Features

Basic Image Features (BIFs) [92, 93, 88, 90, 91, 89] are different from the other schemes presented so far in this chapter, as they are a feature alphabet and thus offer a level of representation that is comparable to just the first stage of the schemes previously discussed.

Whereas schemes like SIFT and HOG make use of oriented gradients, the BIF system does not encode local orientation information, but instead classifies locations according to local symmetry type.

In the BIF system, each pixel in an image is classified into one of seven types based upon the type of the approximate local symmetry. These approximate types are *flat*, *dark* and *light rotational*, *dark* and *light line*, *slope* and *saddle-like*. The classification is determined from the output of a bank of six derivative-of-Gaussian filters, one 0th order, two 1st order and three 2nd order. The algorithm has two tunable parameters. A filter scale parameter, σ, and a threshold, ϵ, which is influential in deciding whether a locality should be classified as *flat*, or as one of the other six articulated symmetry types. Larger values of ϵ result in a greater proportion of an image being classified as *flat*. The BIF calculation is given in Algorithm 2.1. In the BIF calculation, it is first necessary to adjust the filter responses to ensure the final output is dimensionless. As the calculation involves ratios of filter responses, it is sufficient to ensure that the filter outputs are all of the same dimension. This is done by multiplying the output of the derivative filters by the scale raised to the power of the order of the filter.

The information encoded by BIFs is very different from oriented gradients for two reasons. First, symmetry type is orientation invariant and second, BIFs do not encode a magnitude or *strength* of feature type. This results in a very different encoding to oriented gradients, as can be seen from Figure 2.6, where we show an image encoded

Algorithm 2.1 The BIF calculation

1. Measure filter responses c_{ij} to an (i, j)-order derivative-of-Gaussian filter, and from these calculate the scale normalised filter responses $s_{ij} = \sigma^{i+j} c_{ij}$
2. Compute $\lambda = s_{20} + s_{02}, \gamma = \sqrt{(s_{20} - s_{02})^2 + 4s_{11}^2}$
3. Classify according to the largest of: $\{\epsilon s_{00}, 2\sqrt{s_{10}^2 + s_{01}^2}, \pm\lambda, (\gamma \pm \lambda)/\sqrt{2}, \gamma\}$

into oriented gradients with a magnitude, thresholded oriented gradients and BIFs.

2.11.2 Oriented Basic Image Features

As well as the BIF system, which encoded only local symmetry type information, we consider oriented Basic Image Features (oBIFs). In the oBIF system, both local symmetry type and local orientation are encoded into single features. This can be considered as a natural second order extension to oriented gradients, which would be the *slope* BIF type combined with an orientation.

Assigning an orientation only makes sense for certain BIF types. As the *dark rotational*, *light rotational* and *flat* classes are rotationally invariant, no orientation is assigned to them. The *dark line*, *light line* and *saddle* types can be assigned an orientation, but as these types have reflective symmetry the orientation is unsigned. The *slope* class is equivalent to an oriented gradient and thus can possess a signed orientation. This means that the *slope* class has twice the number of possible orientations as the *dark line*, *light line* and *saddle* classes. The oBIF calculation is given in Algorithm 2.2.

When BIF and oBIF encoded images are shown, we use a colour coding for the BIF type and an arrow for the orientation where it is assigned. This is illustrated in Figure 2.7.

2.11.3 Application

As BIFs and oBIFs are relatively new feature alphabets, they are comparatively untested in terms of performance on common recognition problems. However, when used in SIFT like features, Lillholm et al have shown that oBIFs offer better performance than oriented gradients[134] when tested on the VOC2007 object class challenge[66].

2.11.4 Discussion

As our main aim in this work involves developing ways of combining basic features, it is very useful to consider more than one set of such features. This is so that we can try and gauge whether the method of combining features is contributing to performance, rather

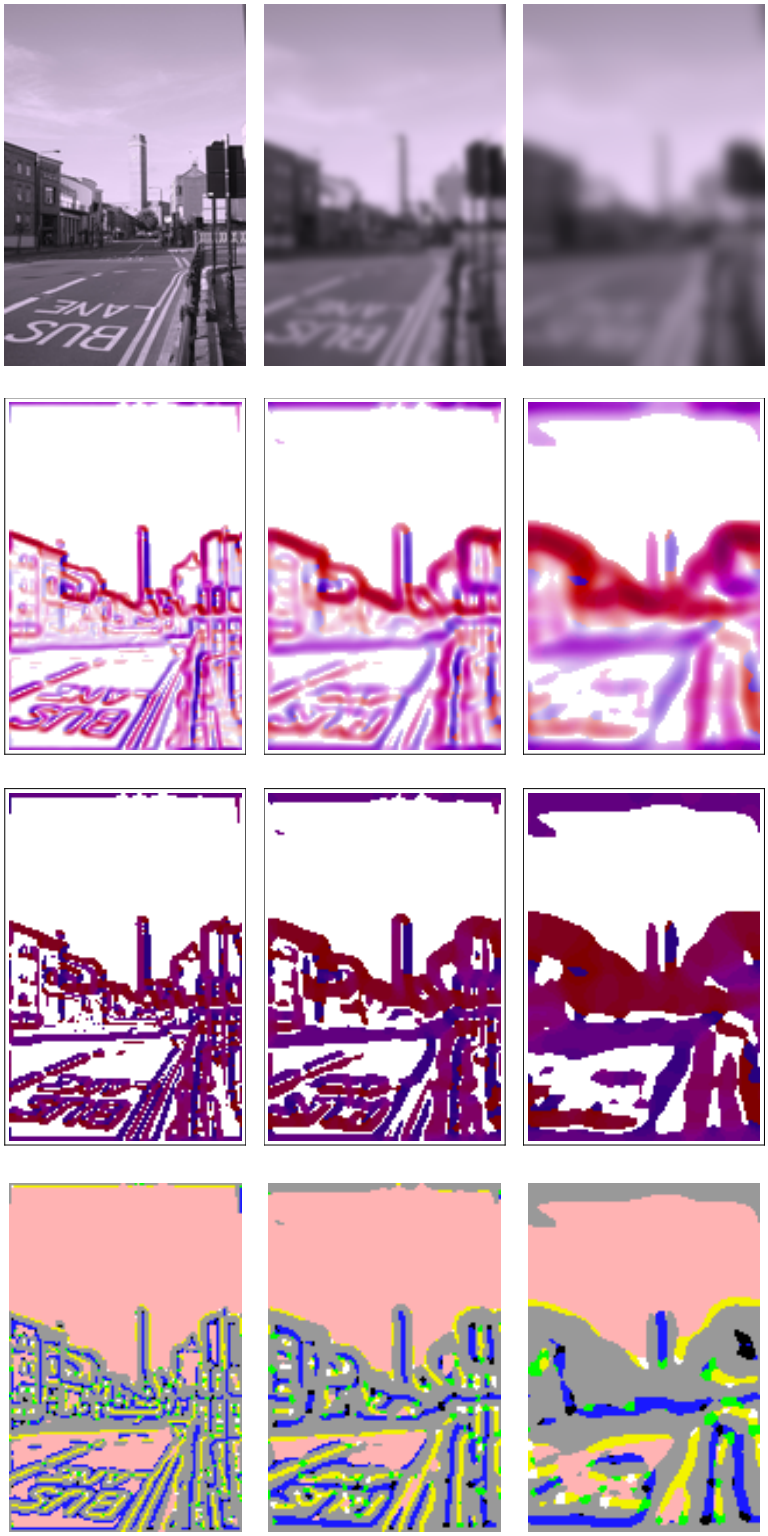


Figure 2.6: Oriented gradients and BIFs
The original image convolved with three different Gaussain kernels (top), along with the oriented gradients, thresholded oriented gradients and BIFs at the same scales. In the oriented gradient encoding, both orientation and gradient are encoded. In the thresholded oriented gradients, only orientation is encoded and in BIFs only local symmetry type is encoded.

Algorithm 2.2 The oBIF calculation

1. Measure filter responses c_{ij} to an (i, j)-order derivative-of-Gaussian filter, and from these calculate the scale normalised filter responses $s_{ij} = \sigma^{i+j} c_{ij}$
2. Compute $\lambda = s_{20} + s_{02}, \gamma = \sqrt{(s_{20} - s_{02})^2 + 4s_{11}^2}$
3. Assign BIF type according to which *Expression* is largest, then calculate orientation where appropriate:

Expression	*BIF type*	*Quantisable orientation*
ϵs_{00}	*flat*	*No orientation*
$2\sqrt{s_{10}^2 + s_{01}^2}$	*slope*	$\arctan\left(\frac{s_{01}}{s_{10}}\right)$ $s_{10} > 0$ $\arctan\left(\frac{s_{01}}{s_{10}}\right) + \pi$ $s_{01} \geq 0, s_{10} < 0$ $\arctan\left(\frac{s_{01}}{s_{10}}\right) - \pi$ $s_{01} < 0, s_{10} < 0$
λ	*dark rotational*	*No orientation*
$-\lambda$	*light rotational*	*No orientation*
$(\gamma + \lambda)/\sqrt{2}$	*dark line*	$\arctan \frac{2s_{11}}{(s_{02} - s_{20} + \gamma)}$
$(\gamma - \lambda)/\sqrt{2}$	*light line*	$\arctan \frac{2s_{11}}{(s_{02} - s_{20} + \gamma)}$
γ	*saddle-like*	$\arctan \frac{2s_{11}}{(s_{02} - s_{20} + \gamma)}$

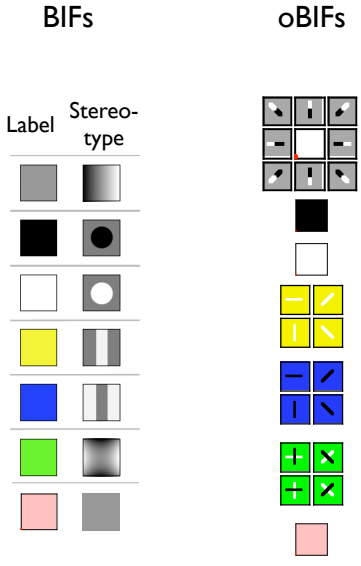


Figure 2.7: The colour coding used for BIFs and oBIFs
Different colours are used to represent each BIF type. For oBIF features, the orientation is indicated using a single line segment, for the line and slope BIF types, and a pair of lines for the saddle BIF type.

than it being a property of the features themselves.

The use of BIFs also provides an interesting contrast to oriented gradients, as the two feature alphabets encode very different information. By using oBIFs we can also try and assess whether these two types of information complement one another in recognition tasks.

2.12 Summary and conclusions

In this chapter we have discussed various sections of the literature that are relevant to the work presented in this thesis. This began with a brief history of invariant recognition in computer vision, where we described how bag-of-words and related methods have come to dominate recent approaches to recognition. We then discussed biologically motivated models, with particular emphasis on network models and how they provide important insight into the computational aspects of recognition.

In the discussion on different forms of representation we proposed that an image-based representation was suitable for the first stage of recognition. Despite the advantages of structural representations, we described how there are significant computational difficulties with these, which has contributed to the wise use of view-based representations.

We then briefly described texture perception and different forms of texture representation. This was followed by a discussion of scale space and multiscale representations. In this section, we described methods such as pyramid representations which use a smoothing kernel to produce a stack of blurred images. However, we concluded that the use of such representations for invariant recognition is limited due to the rigid geometrical structure.

In the methods section, we described the SIFT, HOG, HMAX and Shape Context methods in detail, highlighting how each introduces invariance into the final representation. We then described Basic Image Features and oriented Basic Image Features as a means of encoding local symmetry type.

In the following chapters we shall use elements of these methods to produce a novel multiscale encoding scheme that is an image-based bag-of-words representation. In chapter 6, we provide a comparison of the performance of SIFT, HOG and Shape Context against the novel encoding schemes.

Chapter 3

Datasets

3.1 Introduction

This chapter introduces the datasets that were used for our investigation into encoding schemes. The construction of each set of images is explained along with a description of our evaluation methodology and the computation of a benchmark level of performance.

3.2 MNIST

In order to explore potential invariant encoding schemes, we needed datasets where different aspects of variation can be controlled so that performance can be assessed against the degree of variation. We also required a certain degree of intra-class variation to ensure that the ability of each encoding scheme to categorise was tested at each stage.

To meet these requirements we started with the MNIST dataset [131], which consists of handwritten digits. The small size of the images, at 28 x 28 pixels, and the relatively low number of classes made them ideal for exploring different schemes due to the relatively short computation time of evaluation.

Each image in the MNIST set contains exactly one digit, which is centred, scaled and aligned to a common orientation. The background in each image is blank and there is no noise or clutter present in any of the images.

In order to create an initial benchmark for the performance of an encoding scheme, we first established a classifier to be used in the testing regime. As the emphasis of the investigation was on representation, we were keen to ensure that the classifier used was as simple as possible whilst still offering reasonable performance. For this reason, we decided to use a Nearest Neighbour (NN) classifier. Whilst it was understood that other classifiers, such as Support Vector Machines, may have offered superior levels of absolute performance it was considered unnecessary to use them for this study of relative performance.

The basic evaluation procedure is described in Experiment 3.1 on page 70.

3.3 Shifted MNIST

The first aspect of variation to be considered was translation of the object. In order to create a dataset which could test the robustness of an encoding scheme against translation we took the same MNIST dataset set as before and randomly shifted each digit within a certain range. This was referred to as the Shifted MNIST dataset. In order to create a benchmark performance, images were classified using the normalised intensity encoding and an NN classifier as before. Details are given in Experiment 3.2 on page 71.

3.4 Rotated MNIST

Next, we constructed a set of rotated digits in order to explore rotational invariance. As real objects may often be presented close to a typical orientation, we were interested in exploring how sensitive encoding schemes were to different levels of variation in orientation. Therefore, as with the shifted MNIST images, we constructed a dataset that possessed different ranges of rotation, referred to as the Rotated MNIST set of images. Examples are shown in Figure 3.1.

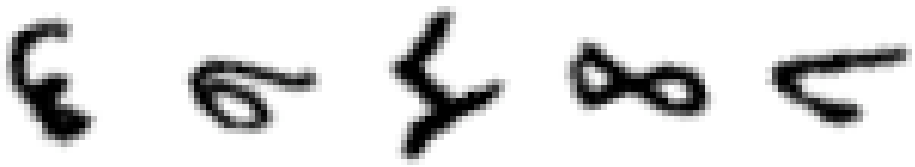


Figure 3.1: Examples from the rotated MNIST dataset.
The set contains figures from MNIST that have been rotated across the full range.

Performance with the normalised intensity encoding and NN classifier was assessed, as with the shifted MNIST dataset. Details are given in Experiment 3.3 on page 72.

3.5 Scaled MNIST

As with the shifted and rotated sets, a set of scaled MNIST digits were created. The details are given in Experiment 3.4 on page 73. Example images are shown in Figure 3.2.

3.6 Cluttered MNIST

The final aspect of variation to be investigated was clutter. In order to create a realistic model of clutter we used blocks of MNIST digits which had not been used in previous

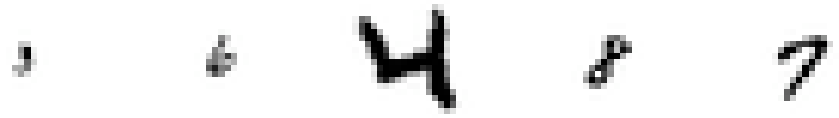


Figure 3.2: Examples from the scaled MNIST dataset.
The set contains figures from MNIST that have been padded with a white background and then resized.

experiments, each of was large enough to be similar to parts of digits but small enough to ensure no block contained a whole digit. The blocks were then placed at random positions around whole digits to create the set referred to as cluttered MNIST. Examples are shown in Figure 3.3.

Figure 3.3: Examples from the cluttered MNIST dataset.
The border is made up of small blocks randomly selected from other digits to create clutter that contains similar local features to the digits.

3.7 Summary and conclusions

In this chapter we have presented several datasets, created from the MNIST set of handwritten digits, that each display a different aspect of variation. For each dataset a benchmark performance has been established using a Nearest Neighbour classifier and the intensity vales of the images. The shifted, rotated and scaled sets shall be used in chapters 4 and 5 to assess the invariance properties of the encoding schemes presented in these chapters. The cluttered set shall be used in chapter 10, where we investigate the effect of clutter on the encoding schemes.

Experiment 3.1 MNIST with Nearest Neighbour

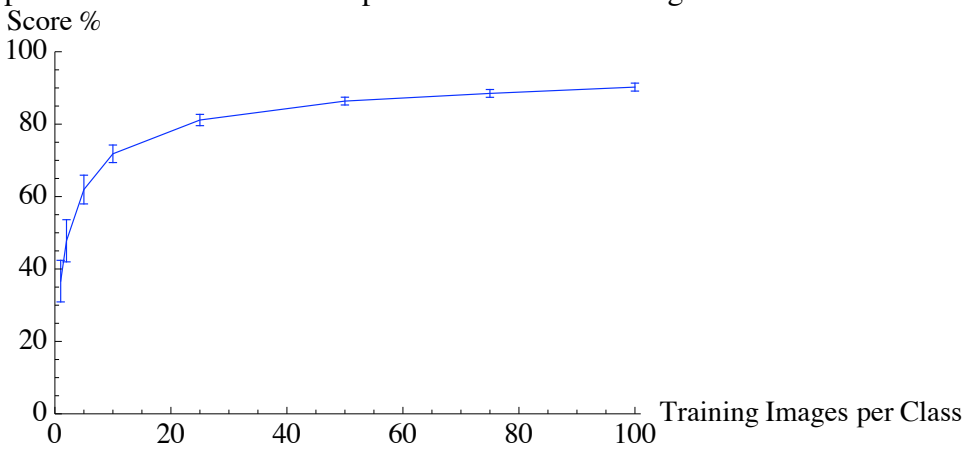
Methods

First, a subset of the MNIST dataset was created by randomly drawing 150 images per class. This was to be used for selecting multiple training and testing sets for evaluation purposes and is referred to as the main set. We then drew another 150 images per class from the remaining MNIST set to use for parameter tuning. This second set was referred to as the tuning set.

Each image was then normalised so that the total intensity across the image was 1. A group of images from each class was then randomly selected from the main set of images as a training set, and the rest were set aside as a test set. Training images were then used to build a Nearest Neighbour classifier using the Euclidean distance. Subsequent training and test sets were randomly drawn from the main set with a total of 50 runs. The mean and standard deviation over all 50 runs was then reported as the performance.

Results

The results are plotted in graph (a), where the bars indicate the standard deviation of the performance scores for that particular size of training set.



(a) The performance of intensity matching using a Nearest Neighbour classifier

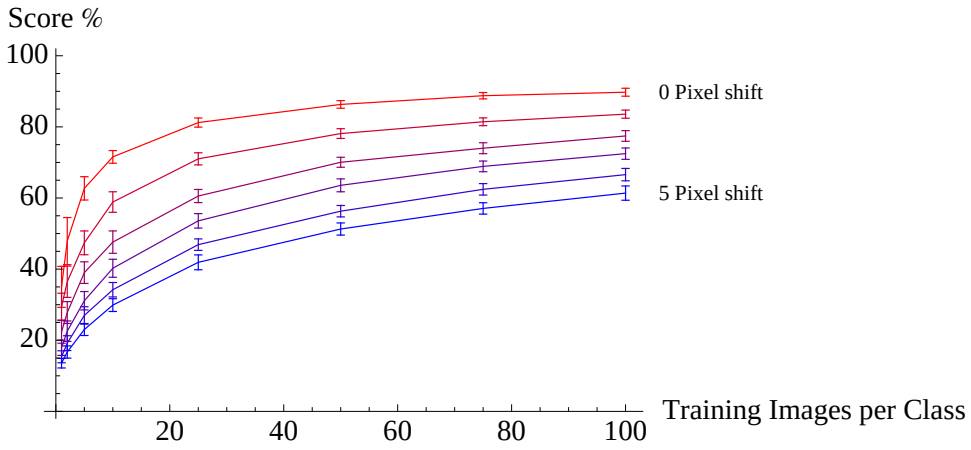
Experiment 3.2 Shifted MNIST with Nearest Neighbour

Methods

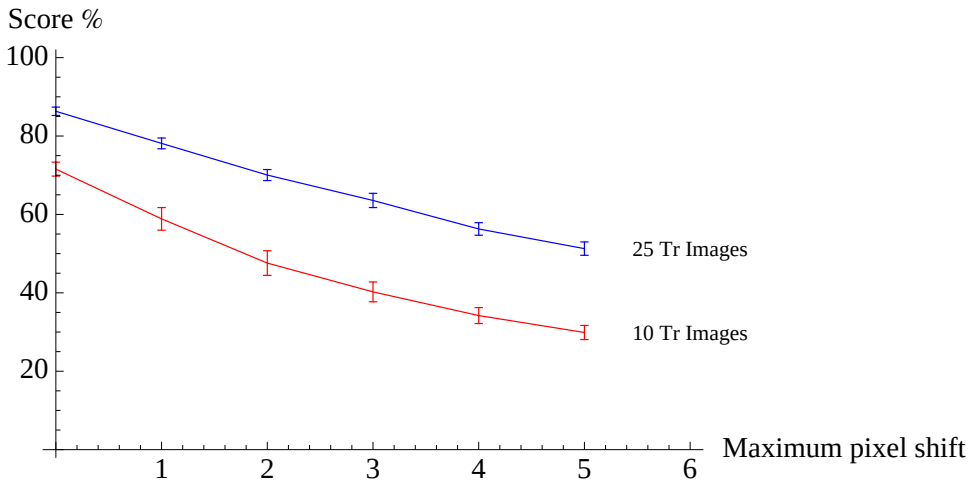
The same subset of 150 images per class from Experiment 3.1 was used from which to draw training and test sets. First each image was normalised as before. Then each image was randomly shifted, within a set range, in both the horizontal and vertical directions. A training set was then randomly selected as before, and the results were calculated over 50 runs.

Results

The performance for different sizes of training set are given in Figure (a), with curves for different ranges of shifting up to 5 pixels. The curve with 0 pixel shift is equivalent to the curve shown in Experiment 3.1. In addition, two curves are given in Figure (b) that show how performance falls as the maximum pixel shift increases. These curves are for 10 and 25 training images per class.



(a) The performance of intensity matching using a Nearest Neighbour classifier for different sizes of training set and different ranges of shifting



(b) The performance of intensity matching using a Nearest Neighbour classifier for different ranges of shifting

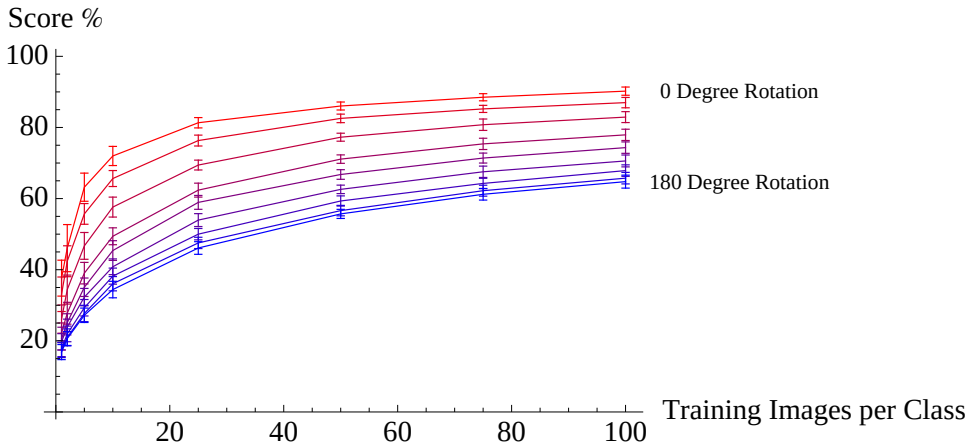
Experiment 3.3 Rotated MNIST with Nearest Neighbour

Methods

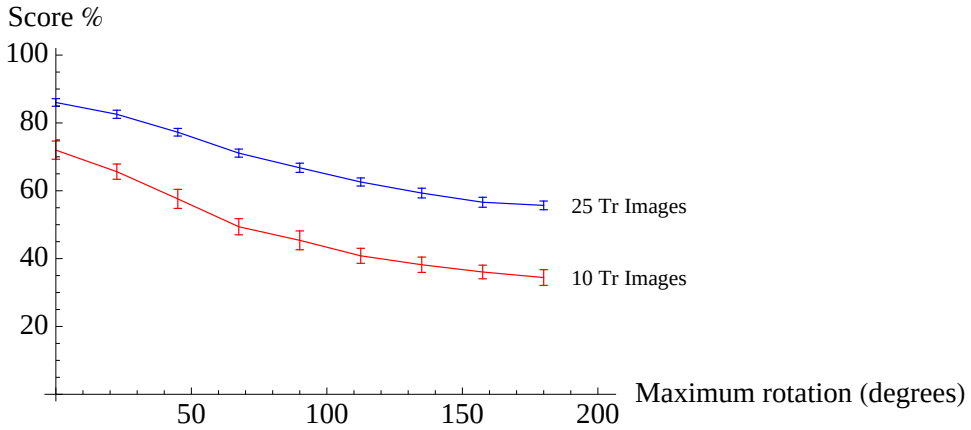
Using the same 150 images per class set as before, all images were rotated by a random angle within a given range. This was done using the built in image rotation function in Mathematica, which used a Gaussian function when resampling the rotated image. Images were then cropped to ensure that they were 28 x 28 pixels, and then normalised as before. The same process was used as in Experiment 3.2 to measure the performance for different sizes of training sets and different ranges of variation.

Results

The performance for different size of training sets is given in Figure (a), with each curve showing a different range of rotation angle. The performance against rotation angle range is shown in Figure (b) for training set sizes of 10 and 25 images per class.



(a) The performance of intensity matching using a Nearest Neighbour classifier for different sizes of training set and different ranges of rotation



(b) The performance of intensity matching using a Nearest Neighbour classifier for different ranges of rotation

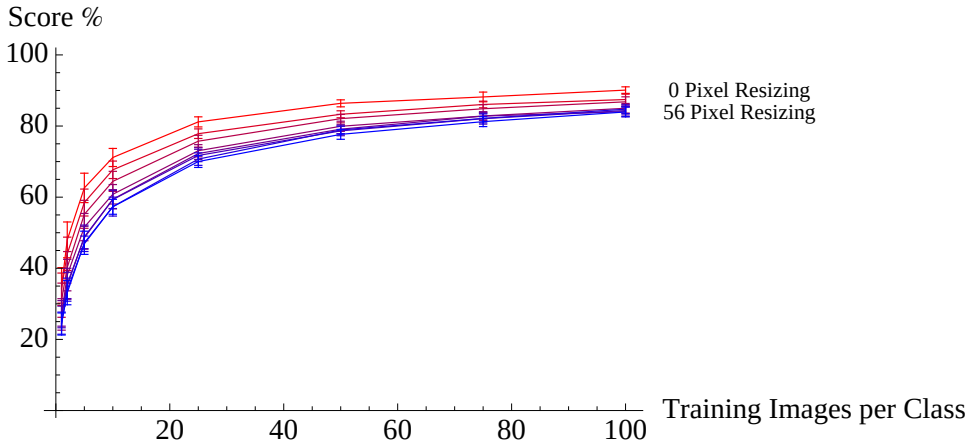
Experiment 3.4 Scaled MNIST with Nearest Neighbour

Methods

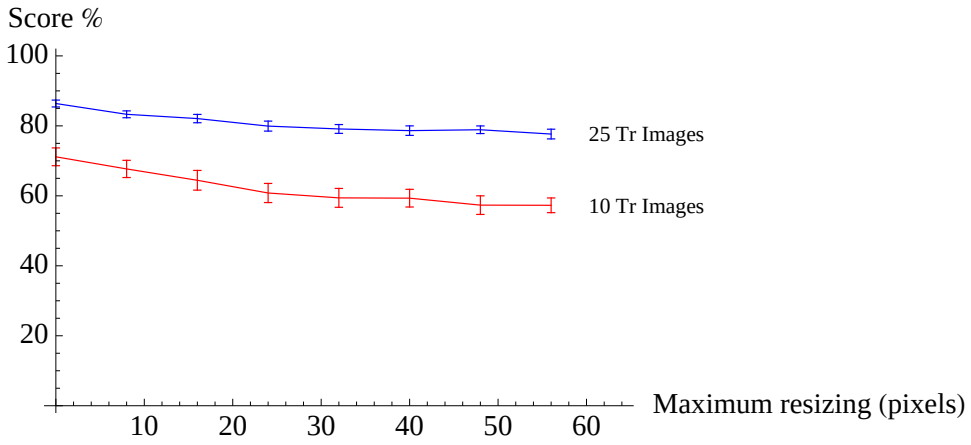
The same subset of images was used as in previous experiments. Each image was first padded with a blank border of a width randomly chosen within a certain range. Images were then resized to 28 x 28 pixels and normalised as before.
The same testing regime was then used to produce graphs for the performance of a simple NN classifier for different sizes of training sets and different ranges of variation.

Results

The performance for different sizes of training set is shown in Figure (a). Each curve indicates the performance for a particular range of border added, with the maximum value being 56 pixels. Performance against maximum border size is shown for training set sizes of 10 and 25 images per class in Figure (b).



(a) The performance of intensity matching using a Nearest Neighbour classifier for different sizes of training set and different ranges of resizing



(b) The performance of intensity matching using a Nearest Neighbour classifier for different ranges of resizing

Experiment 3.5 Cluttered MNIST with Nearest Neighbour

Methods

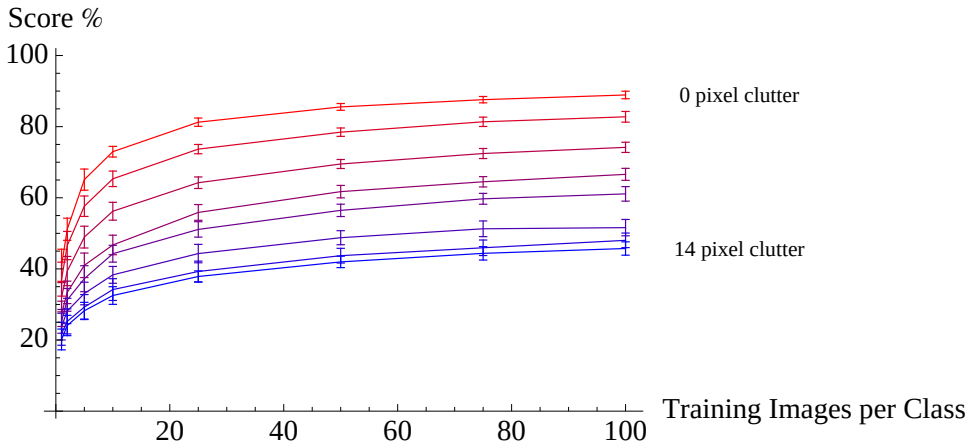
The same subset of 150 images per class were used as before, referred to as the evaluation set for the purposes of this experiment. All images in the evaluation set were first normalised. A further subset of MNIST images was then randomly selected, with equal numbers of images per class, and divided into blocks of 7x7 pixels. For each image in the evaluation set randomly selected blocks were added at random positions around the digit to construct the clutter border. A region of 22 x 22 pixels at the centre of each image in the evaluation set was left untouched to ensure the digit was still present. Images were then cropped so that the clutter border fell within a certain range.

The performance of the NN classifier with the images was then investigated for different sizes of training set and clutter border.

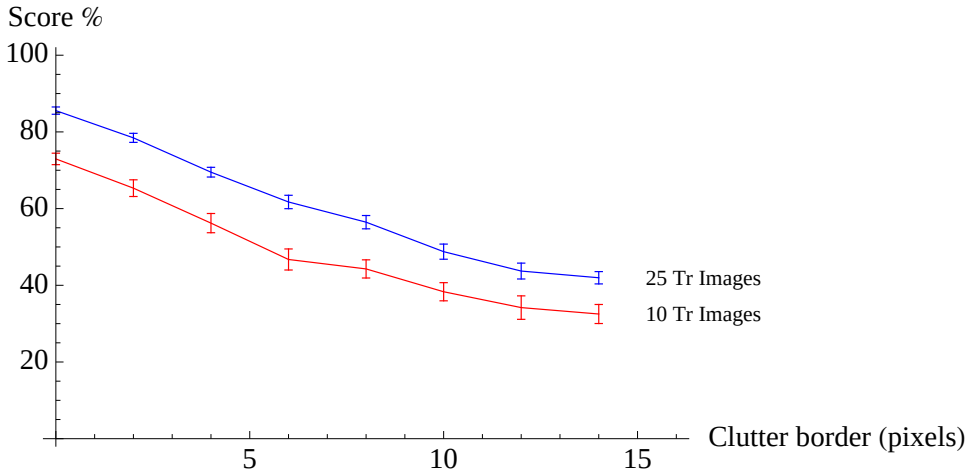
Results

The performance of the NN classifier for different sizes of training set is shown in Figure (a). Each curve represents a different size of clutter border up to a maximum of 14 pixels, which approximates to a clutter area to object area ratio of 3:1.

Graphs are also given for the performance against clutter border size for training sets of 10 and 25 images per class, as shown in (b).



(a) The performance of intensity matching using a Nearest Neighbour classifier for different sizes of training set and different ranges of clutter



(b) The performance of intensity matching using a Nearest Neighbour classifier for different ranges of clutter

Chapter 4

Histograms of Features

At the heart of several of the schemes discussed in Chapter 2 are histograms of basic features. In descriptors such as SIFT and HOG these histograms are calculated within subregions and then combined within a template to form the descriptor. To begin our investigation we wanted to see how histograms of basic features performed, both when using the whole image and when calculating local histograms of subregions. Whilst this has been previously well investigated for oriented gradients, we wanted to understand how simple histogram schemes performed when using BIFs and oBIFs.

This chapter begins with a brief description of the use of histogram schemes. We then investigate how simple histograms schemes perform when using each of three basic feature types, oriented gradients, BIFs and oBIFs. Finally, we look at the effect of spatial binning for each of the three feature types, where local histograms are calculated in a similar way to SIFT and HOG.

4.1 Introduction

Histogram representations involve the loss of spatial information. In its simplest form an image can be converted into a histogram by taking the intensity values across the entire image and discarding all spatial information. In order to allow simple comparison, the intensity values are then placed into bins by counting the number of occurrences within certain ranges. Thus, we can view a histogram as an approximation of a distribution of a set of features as the contents of an image.

In general, histogram representations are defined by two characteristics. First, the degree to which spatial information is discarded. This can lead to a global histogram, where all spatial information is discarded, or local histograms (or the scale space equivalent locally orderless image [124, 244]), where coarse spatial information is preserved.

The second defining characteristic is the set of features to be counted. The choice of features can depend on the application but common sets include colour [225, 96, 40, 95], orientation [79, 143, 269, 151, 271] and local binary patterns (LBP)[170].

As histogram representations involve the loss of spatial information, they have been widely used in texture applications in the form of global histograms[242, 170, 139, 245, 196, 195]. In other applications, such as object recognition, local histograms tend to be used [258, 198, 199, 13, 211, 212].

4.2 Oriented gradient histograms

We began the investigation by looking at histograms of oriented gradients. These form the basis of many schemes that have been used for object recognition, such as HOG and SIFT, and therefore we wanted to use such a scheme to form a base level of performance. The simplest of oriented gradient schemes is a single histogram at a single scale and this was tried first. The details are described in Experiment 4.1 on page 81.

The results show that the performance is significantly lower than when using the intensity levels for the MNIST dataset, meaning that, in this simple form, a histogram of oriented gradients is unlikely to be of much use in applications such as this. However, despite the low performance, the results do also indicate that the encoding is highly invariant to shifting, as would be expected of a global histogram representation. In addition, the performance appears stable in relation to small changes in the parameter values.

4.3 Histograms of Basic Image Features

Next, we looked at Basic Image Features (BIFs). As each BIF contained information about the local symmetry type but discarded orientation information we were interested to see how the performance compared to oriented gradients. The experiment details are given in Experiment 4.2 on page 82.

The results show that the performance of BIFs is considerably worse than with oriented gradients. If we look at the encoded images, as shown in Figure 4.1, we see the most prominent aspect is the encoding of the line segments (shown by the blue and grey BIFs.) Counting the occurrence of these types is likely to be similar to estimating the length of line segment within the image, which in itself is unlikely to be a useful descriptor for digit recognition.

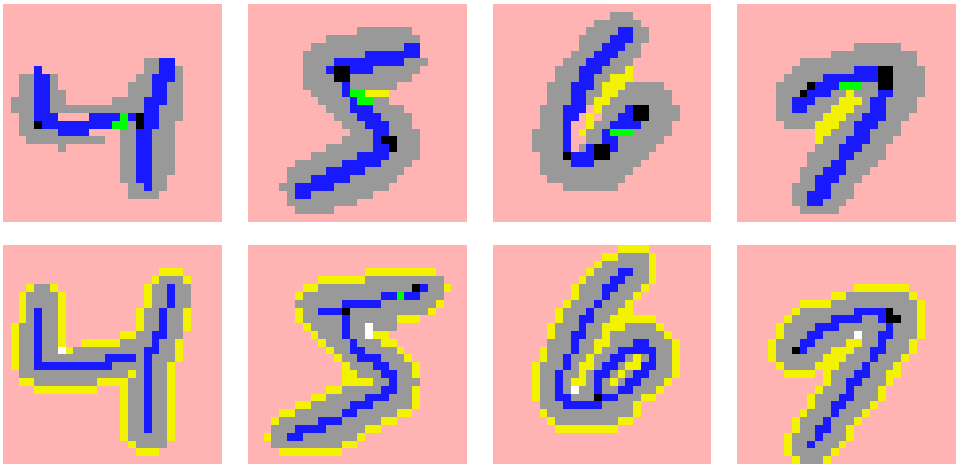


Figure 4.1: Optimal parameter values for BIFs
The BIFs at the optimal parameter values (top) and at the initial parameter values. At the optimal parameter values the yellow BIF type tends to appear in between line segments, thus capturing the interaction between segments. At other parameter values, the yellow BIF type merely traces the outline of the digit meaning that it is unlikely to contain information not captured by the blue BIF type.

The optimal value for the threshold was far higher than with oriented gradients, meaning that only locations with strong symmetry are not encoded as *flat*. If we look at the images encoded at the optimal parameter values we can see that the yellow BIF type tends to appear in between line segments. It may be the case that, at these parameter values, the BIF encoding is best able to capture the interaction between line segments.

4.4 Histograms of Oriented Basic Image Features

Whilst the performance of BIFs was poor compared to oriented gradients, the information encoded by the two schemes was very different. We were interested to see how combining the two sources of information, in the form of oriented Basic Image Features (oBIFs) would perform. To do this, we repeated the previous experiment using oBIFs, as explained in Experiment 4.3 on page 83.

By combining orientation and symmetry type, in the form of oBIFs, we see an improved performance over the oriented gradient histogram. However, the performance was still lower than that which could be achieved using the intensity values.

4.5 Spatial Binning

In order to improve the performance of the histogram schemes we decided to first look at the effect of spatial binning, where basic features are counted in spatial regions of the image. This is used both in HOG and in SIFT and, as discussed in Chapter 2, but it had not been tested with BIFs and oBIFs before.

We began by looking at spatial binning using oriented gradients. This was very similar to the HOG scheme, except that we used orientations that had been thresholded (so that each location was either classed as having an orientation or being *flat*) whereas the HOG scheme uses the weights of gradient. In addition, the standard HOG scheme uses unsigned gradients, whereas our scheme used signed gradients. However, with a dataset such as MNIST, where all the digits are dark lines on light backgrounds, the effect of this is likely to be minimal. The experimental details are given in 4.4.

The performance with spatial binning is considerably better than with a simple histogram of oriented gradients. This result is not unexpected, as the success of both HOG and SIFT have demonstrated its usefulness in other applications. We might also expect the digits in the MNIST dataset, which have been centred, scaled and oriented, to fit well to a spatial template.

We then applied the same process to BIFs, dividing the encoded image into local histograms of a certain size. The details are given in Experiment 4.5 on page 85.

As with oriented gradients, we observed a big increase in performance with the introduction of spatial binning. The tuning graph shows that the performance dropped off more rapidly than with oriented gradients as the box size increased.

It is interesting to note that the tuned threshold parameter was 0, meaning that no location in the image is classed as *flat*. As can be seen in the lower half of Figure 4.1, BIFs with a lower threshold value tend to encode images with a BIF type that correlates with distance from the line segment, with the blue BIF type falling on the line, then the grey BIF and then the yellow BIF. This may mean that, with spatial binning, BIFs are acting simply as a blurred template matching system which would mean we would expect slightly better performance than when using the intensity values with MNIST.

Finally, we looked at applying spatial binning to oBIFs, in a similar way as the previous two sections. The details are provided in Experiment 4.6.

Whereas with the simple histogram scheme, the use of local symmetry information in addition to local orientation information improved performance, when spatial binning

is used the same effect is not observed. This is a somewhat surprising result, given the relatively large increases in performance observed when spatial binning is used with both oriented gradients and BIFs.

However, as much as the MNIST digits are centred, scaled and aligned, the use of a template (in the form of the grid of blocks) will place an upper limit on the performance. This upper limit comes from the degree to which the digits comply with the template. It is possible that this upper limit is reached when using spatially binned oriented gradients and that the addition of local symmetry information can therefore not increase performance. In this case we might also expect the use of an additional source of information to reduce performance as a tighter constraint is being used, in that both local orientation and local symmetry have to match the template.

4.6 Comparison of Results

The results of the different schemes tested in this chapter, as summarized in Table 4.1 with computational performance given in Table 4.2, have shown the relative performance of the three sets of features. When a simple histogram scheme is used, the use of local orientation and local symmetry produce the best results. However, the performance of this scheme is still relatively poor and in order to gain an increase in performance we have to introduce spatial binning. When this is done, performance using all three feature sets improves but the highest level of performance is achieved when using oriented gradients without local symmetry information.

Table 4.1: Comparison of performance (in % correct) for the histogram schemes

Scheme	Training images per class		
	2	10	100
Oriented gradients (OGs)	37.2±3.6%	56.0 ±2.2%	71.0 ±1.4%
BIFs	27.2±3.0%	33.8 ±1.5%	39.9 ±1.1%
oBIFs	41.5±2.9%	62.3 ±2.2%	82.1 ±1.5%
OGs with spatial binning	**62.3±4.9%**	**83.4 ±1.5%**	**93.4 ±0.9%**
BIFs with spatial binning	50.7±4.2%	71.2 ±1.9%	86.3 ±1.3%
oBIFs with spatial binning	60.0±4.3%	80.9 ±1.6%	92.5 ±1.0%

In order to further test the performance of the histograms, we also tested each on an additional dataset. This set, chars74k, consists of letters and digits and is discussed in greater detail in Chapter 6. The performance for each of the schemes is given in Table 4.3. As the number of images per class in this dataset is less than 100, the performance figures are only given for 2 and 10 training images per class.

Table 4.2: The computational performance for each of the histogram schemes

The encoding time is given for each image, along with the classification time for a Nearest Neighbour classifier using 10 images per class. The computation times given are based upon an implementation of the system in Mathematica 7.

Scheme	Size	Computation time (ms)	
		Encoding	Classifier
Oriented gradients (OGs)	24	12	0.60
BIFs	7	10	0.52
oBIFs	43	13	1.3
OGs with spatial binning	4056	130	7.7
BIFs with spatial binning	1183	450	2.3
oBIFs with spatial binning	1548	170	2.2

Table 4.3: Comparison of performance (in % correct) for the histogram schemes on the additional chars74k dataset.

Scheme	Training images per class	
	2	10
Oriented gradients (OGs)	20.0±2.2%	31.1±1.3%
BIFs	12.7±1.7%	19.1±1.5%
oBIFs	25.6±2.5%	41.2±1.5%
OGs with spatial binning	37.2±3.7%	**52.7±1.5%**
BIFs with spatial binning	31.5±2.9%	45.2±1.2%
oBIFs with spatial binning	**37.7±3.2%**	52.3±1.4%

4.7 Summary and conclusions

In this chapter we have described the histogram schemes for three different feature sets, oriented gradients (OGs), Basic Image Features (BIFs) and oriented Basic Image Features (oBIFs). For each, we have established the recognition performance using the datasets from Chapter 3, and confirmed that each scheme is invariant to translation. Out of three different feature sets, oBIFs have achieved the highest recognition rate.

We have then looked at the effect of spatial binning for each of the three feature sets. The results have shown that, for each of the three different feature sets, performance has improved considerably with spatial binning. However, in these experiments oriented gradients marginally outperformed oBIFs.

In the next chapter we look to demonstrate similar levels of performance using histograms of features, without spatial binning, by using multsicale features referred to as column features.

Experiment 4.1 Oriented gradient histograms with the MNIST datasets

Methods

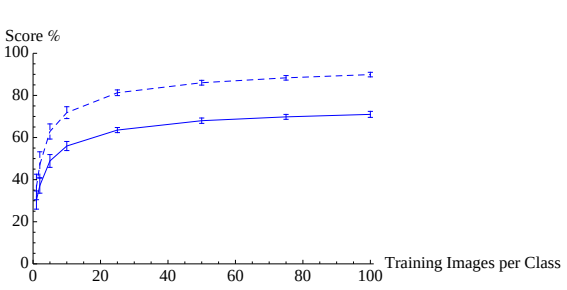
We used the same subsets of the MNIST dataset described in Experiment 3.1, which provided 150 images of each class for the purposes of parameter tuning and a further 150 images per class from which to draw multiple training and test sets for evaluation purposes. Each image was encoded in oriented gradients using a set of 1st order Derivative of Gaussian (DtG) filters. These were used in order to ensure a fair comparison against other features considered in this work, namely BIFs and oBIFS, which both used DtG filters as a first step. For consistency we also used a threshold rather than a continuous weight variable, so that if, at a certain location, the weight of the oriented gradient was above the threshold is was assigned that orientation and if it was below, the location was assigned as being *flat* as in the BIF scheme explained in Chapter 2.

There were three parameters to be tuned. These were the orientation quantisation, which determined the number of bins in the orientated gradient histogram, the scale parameter, which determined the scale at which gradients were calculated, and the threshold parameter described above. These parameters were tuned using the tuning set of images, with a single sweep for each parameter. Tuning was done for the orientation quantisation first, then the scale parameter and then finally the threshold parameter.

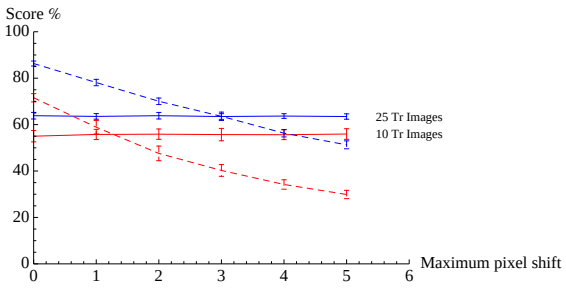
Using the tuned parameters, images were then encoded into oriented gradient histograms by counting the number of occurrences of each orientation in the image. For a given training set size, a set of images was randomly selected. These were then used to train a Nearest Neighbour (NN) classifier using the Bhattacharyya distance [117]. This was then repeated for 50 different randomly sampled training sets, with the mean score being given as the performance. The experiment was then repeated for the shifted MNIST set.

Results

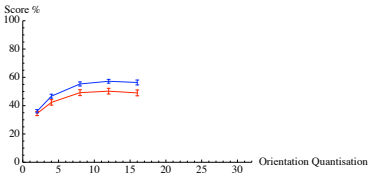
The performance on the MNIST set is shown in graph (a), where the performance was 37.2±3.6%, 56.0 ±2.2% and 71.0 ±1.4% for 2, 10 and 100 training images per class respectively. The results for the shifted MNIST set are given in graph (b). For each graph the benchmark performance obtained from Chapter 3 is shown as a dotted line. In addition the tuning curves for the three parameters are given where the two curves are for 10 and 25 tuning images per class.



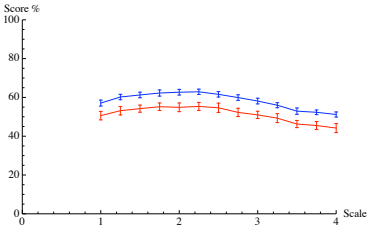
(a) Oriented Gradients



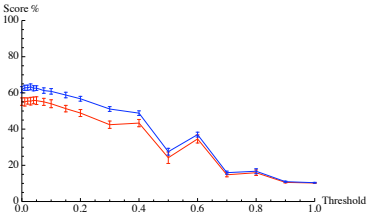
(b) OGs Shifted



(c) Orientation quantisation tuning



(d) Scale tuning



(e) Threshold tuning

Experiment 4.2 Histograms of BIFs with the MNIST datasets

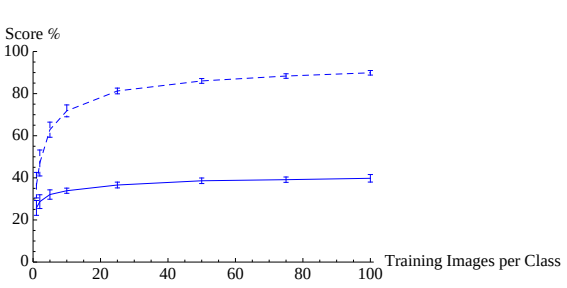
Methods

The same subsets of the MNIST set were used as in Experiment 3.1. Images were encoded into BIFs which, as described in Chapter 2, use a bank of DtG filters to assign each location within an image into one of seven types based upon local symmetry. The algorithm takes two parameters, the scale, which determines the size of filters, and the threshold, which determines the likelihood of any location being classified as *flat*.
In order to tune the parameters we followed the same procedure as in the previous experiment, where performance was examined for a sweep through each parameter in turn. For this experiment the scale was tuned first.

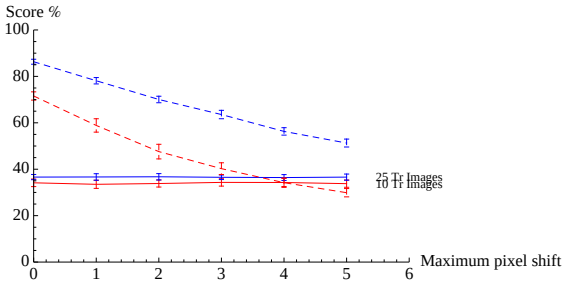
Histograms of BIFs were then calculated by counting the number of occurrences of each BIF type, then normalising so that the final encoding sums to one. Classification was performed using a Nearest Neighbour classifier with the Bhattacharyya distance.

Results

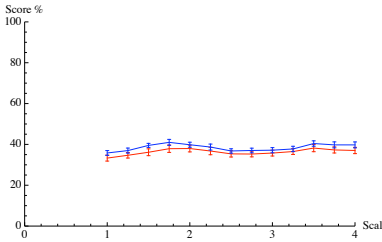
The results are shown below with the graph on the left giving the performance for different training set sizes, where the performance was 27.2±3.0%, 33.8 ±1.5% and 39.9 ±1.1% for 2, 10 and 100 training images per class respectively, and the graph on the right demonstrating the shift invariance of the encoding. The parameter tuning gave an optimal scale of 1.75 and and an optimal threshold of 0.2. The tuning curves are shown below.



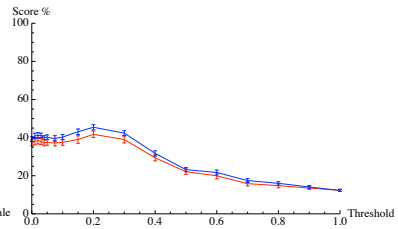
(a) BIFs with MNIST



(b) BIFs with shifted MNIST



(c) BIFs scale tuning



(d) BIFs threshold tuning

Experiment 4.3 Histograms of oBIFs with the MNIST datasets

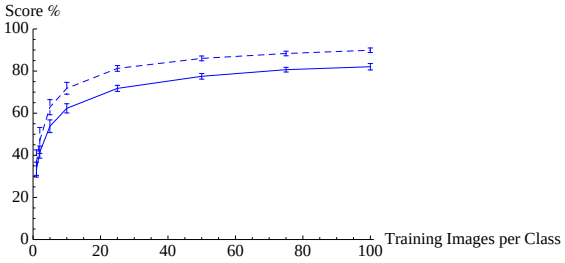
Methods

Using the same sets as before, images were encoded into oriented Basic Image Features, which assigns both a local symmetry type and a local orientation to each location. This takes an additional parameter to the BIF system, the orientation quantisation, which determines the number of possible orientations for the four symmetry types that allow orientation. (Locations classed as *dark rotational*, *light rotational* and *flat* are not assigned an orientation.) Parameters were tuned using a single sweep for each, in the order orientation quantisation followed by scale then the threshold.

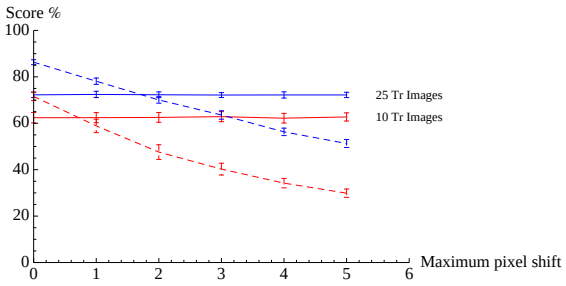
Histograms were calculated as before, by counting the occurrences of each oBIF type and then normalising. Classification was done using a Nearest Neighbour classifier and the Bhattachrarryya distance.

Results

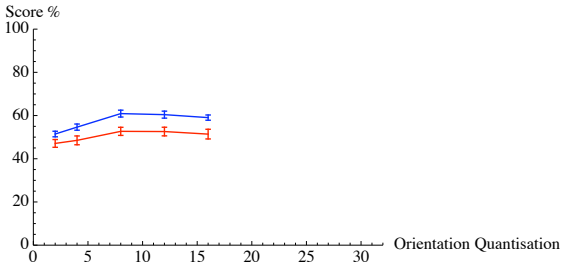
The performance for different sizes of training set is shown in graph (a), where the performance was 41.5±2.9%, 62.3 ±2.2% and 82.1 ±1.5% for 2, 10 and 100 training images per class respectively. The performance using the shifted MNIST set is shown in graph (b). The tuned orientation quantisation was 8 for oBIFs, which meant there was a set of 43 oBIF features. The tuned scale was 2 and the threshold value was 0.01. The tuning graphs are shown below.



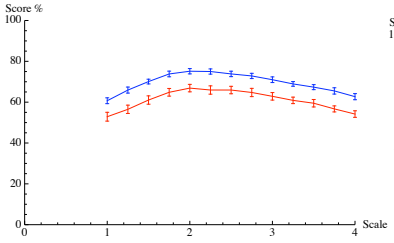
(a) oBIFs



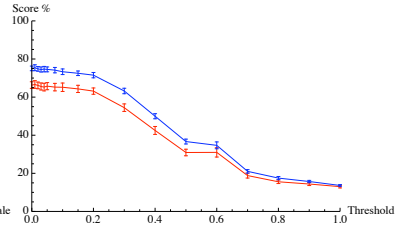
(b) oBIFs shifted



(c) oBIFs orientation tuning



(d) oBIFs scale tuning



(e) oBIFs threshold tuning

Experiment 4.4 Oriented gradient histograms with spatial binning

Methods

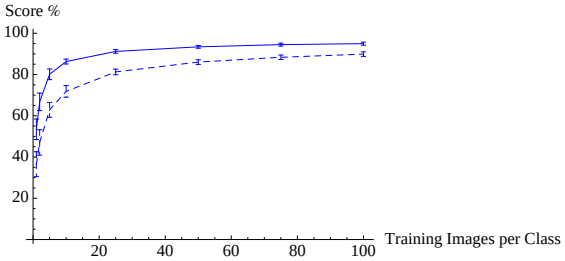
Using the MNIST and shifted MNIST subsets, images were encoded into oriented gradients as described in Experiment 4.1. Each encoded image was then divided into blocks of a given size. For each block, a histogram was calculated by counting the occurrences of each possible orientation and then normalising.

As the blocks were overlapping there were two additional parameters to tune, the block size and the overlap between blocks. However, for simplicity we set the overlap at half the block size and used this as single parameter to tune. This, and the orientation quantisation, scale and threshold parameters, were tuned with a single parameter sweep beginning with the block size.

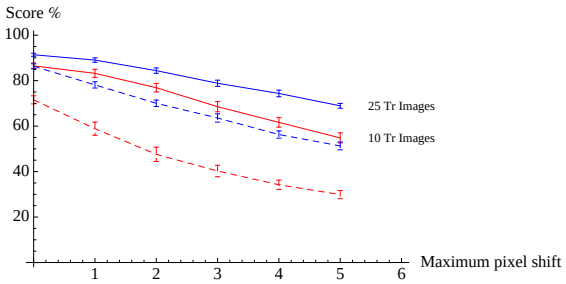
Encoded images were classified using a Nearest Neighbour classifier and the Bhattacharyya distance.

Results

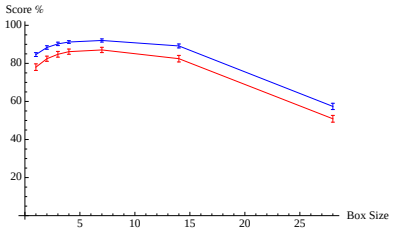
The performance for different sizes of training set is shown in the graph on the left, where the performance was 62.3±4.9%, 83.4 ±1.5% and 93.4 ±0.9% for 2, 10 and 100 training images per class respectively, with the performance with the shifted MNIST set being shown on the right. The tuning graph for the block size is also given below.



(a) Oriented Gradients



(b) OGs Shifted



(c) Box size for OG

Experiment 4.5 BIF histograms with spatial binning

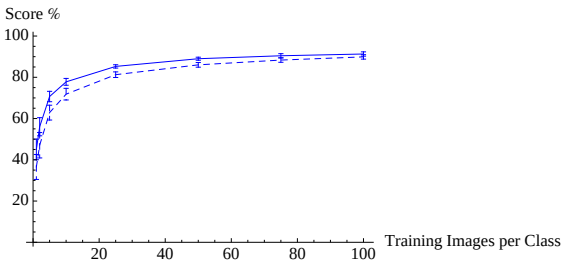
Methods

Using the same MNIST subsets, images were encoded into BIFs. Each encoded image was then divided up into blocks of a given size. As in Experiment 4.4, we set the overlap between blocks equal to half the block size, meaning that there was a single parameter associated with the blocks.

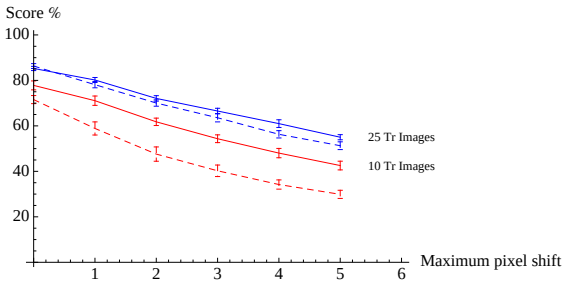
Parameters were tuned using a single sweep for each, beginning with the block size. Images were then classified using a Nearest Neighbour classifier and the Bhattacharyya distance.

Results

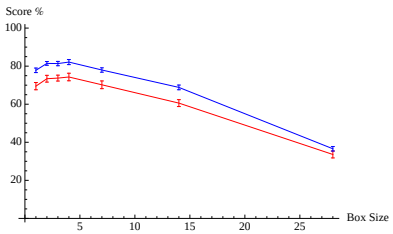
The performance for different sizes of training set is given in graph (a), where the scores were 50.7±4.2%, 71.2 ±1.9% and 86.3 ±1.3% for 2, 10 and 100 training images per class respectively. The performance with the shifted MNIST set is given in graph (b).



(a) BIFs



(b) BIFs shifted



(c) Box size for BIFs

Experiment 4.6 oBIF histograms with spatial binning

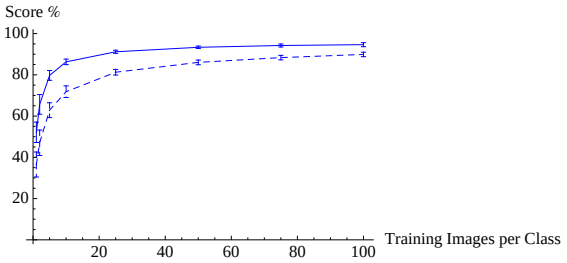
Methods

Using the same MNIST subsets, images were encoded into oBIFs. Encoded images were then divided into blocks. As in Experiment 4.4, we set the overlap between blocks equal to half the block size, meaning that there was a single parameter associated with the blocks.

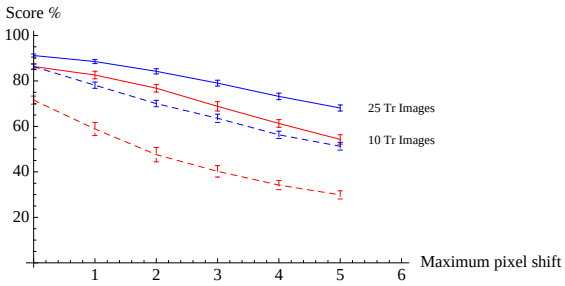
Parameters were tuned using a single sweep for each, beginning with the block size. Images were then classified using a Nearest Neighbour classifier and the Bhattacharyya distance.

Results

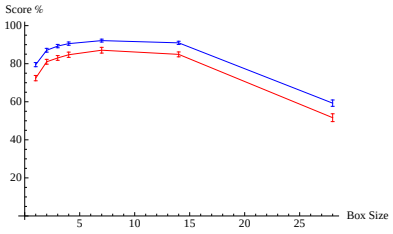
The performance for different training is shown in graph (a), where the scores were 60.0±4.3%, 80.9 ±1.6% and 92.5 ±1.0% for 2, 10 and 100 training images per class respectively, with the performance against the shifted MNIST dataset in graph (b). The tuning graph for the block size is shown in graph (c).



(a) oBIFs



(b) oBIFs shifted



(c) Box size for oBIFs

Chapter 5

Column Features

The work in the previous chapter demonstrated that histograms of features can perform well when used with spatial binning to create a template of histograms. This works well with datasets, such as MNIST, where the objects within each image are well aligned, scaled and positioned. In addition the spatial arrangement of local features has to be consistent in order to correctly match individual histograms within the template.

However, a key aim of the investigation is to move away from the grid structure that is involved with templates of histograms. We therefore turned our attention away from schemes that involved spatial binning and concentrated on schemes that used a single histogram.

5.1 Introduction

In order to achieve good performance, whilst using a single histogram, we needed to develop a different set of features to be used with the histogram. Our approach to this was based upon a simple observation, first described in [164], that the oriented gradient, BIF and oBIF type at certain locations can change over scale, as shown in Figure 5.1.

We wondered whether, by considering the occurrence of these different features along the dimension of scale, this would capture useful information about the structure of the image. Our hope was that patches of a certain feature type at the coarser scale would take the place of the regions of a grid, thus removing the need to place a predetermined structure upon the descriptor as in other schemes.

Using features at different scales is not a new idea. Mutliscale versions of both SIFT and HOG have been proposed [19]. In the HMAX model, features are calculated at multiple scales and then a MAX function is applied to select the greatest response over scale. Other schemes have used gradients at multiple scales [135], multiscale Gabor features [103] and texture schemes have used features at multiple scales [270, 245].

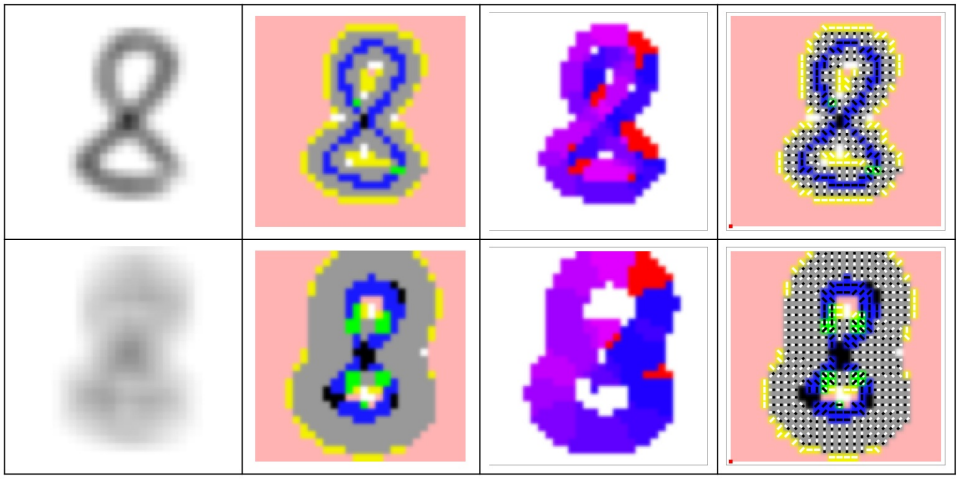


Figure 5.1: Multiscale images along with BIFs, oriented gradients and oBIFs. At certain locations within the image, the BIF, oBIF or oriented gradient feature type changes over scale.

Our approach differed from previous schemes in that we were interested in the combination of features at individual locations within the image. We thought this was best captured by considering concatenations of features across scale, which we referred to as column features.

The simplest form of column feature consisted of a pair of basic features, either oriented gradients, BIFs or oBIFs, separated by a scale ratio, r. Such features could be used within the histogram framework, as by using pairs of features the scheme would not reach the point of combinatorial explosion. However, we would end up with a histogram that contained the square of the number of bins as in the basic feature set. For example, when using oBIF columns, the original set of 23 features would turn into a set of 23^2 oBIF column features, as would the corresponding histogram. This idea is illustrated for oBIFs in Figure 5.2.

In order to evaluate this idea we applied the column idea to each of the three feature types that were used in Chapter 4.

5.2 Oriented Gradient Columns

For oriented gradient columns, the first step was the same as in the previous chapter, where oriented gradients are calculated using the output of Derivative of Gaussian filters. For the purposes of comparison with BIFs and oBIFs we kept the threshold, beneath which locations would be classed as *flat* and above which locations would be classed as having an orientation with unity weight.

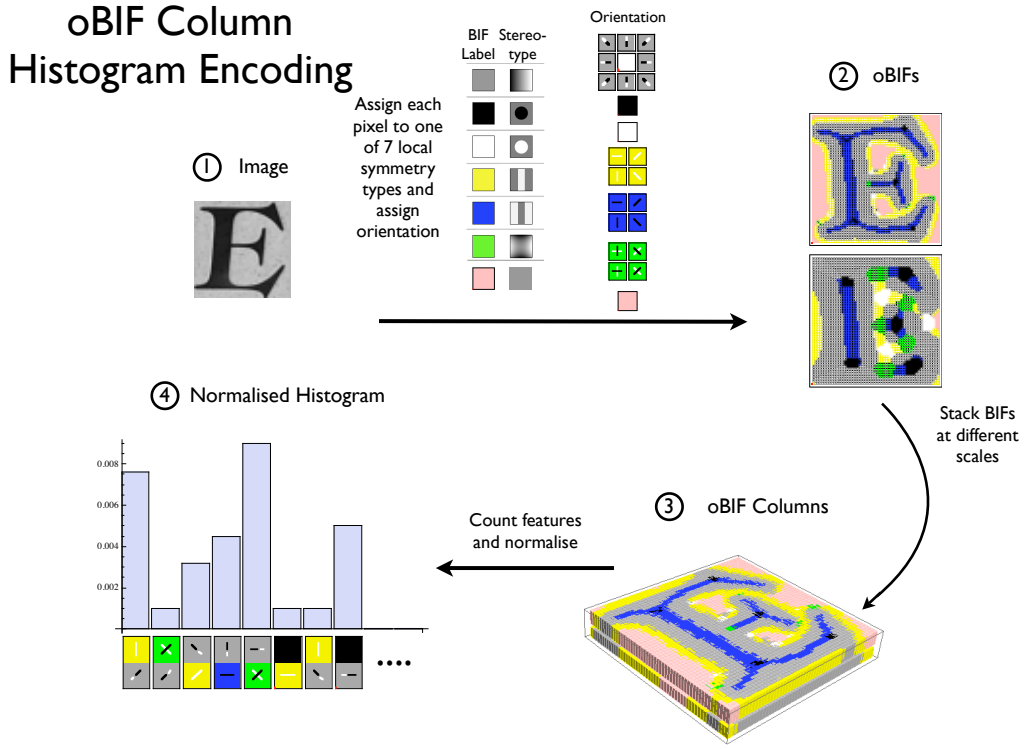


Figure 5.2: The oBIF Column encoding scheme.
An image is first encoded into oBIFs at two scales. The oBIF column features are then calculated by concatenating features at each location. Finally, oBIF column types are counted and normalised to give the final histogram.

As the scheme involved calculating oriented gradient columns we had one more tunable parameter than with the oriented gradient histograms. We characterised this by using a base scale, which was the finer of the two scales at which oriented gradients were calculated, and the scale ratio, which was the ratio between the base scale and the coarser scale. The oriented gradient column process is given in Algorithm 5.1.

We tested the scheme with the datasets as used in the previous chapter. The details are given in Experiment 5.1 on page 100.

The results show that the oriented gradient columns perform much better than oriented gradients at a single scale. Even when using only 4 orientations (giving a column histogram of 16 bins), the performance of the column scheme greatly exceeds that with 16 oriented gradients at a single scale. Importantly, the tuning curve for the scale ratio, which is the new parameter in the column scheme, appears to the stable around the optimal value. When the scheme is extended to three scales, there is a very marginal improvement. However as this increases the size of the histogram to $(n+1)^3$, where n is the orientation quantisation, we considered the two scale scheme to be preferable.

Algorithm 5.1 The Oriented Gradient Column scheme

1. For a given σ_{BASE} and scale ratio, r, measure filter responses $s_{ij} = \sigma^{i+j} c_{ij}$ of 1st order derivative-of-Gaussian filters at scales σ_{BASE} and $r\sigma_{BASE}$, and from these calculate the scale normalised filter responses $s_{ij} = \sigma^{i+j} c_{ij}$

2. For each location at each scale if $2\sqrt{s_{10}^2 + s_{01}^2} > \epsilon$ calculate orientation as:

   $Round(\frac{\arctan(\frac{s_{01}}{s_{10}})}{2\pi})$ $\quad s_{10} > 0$

   $Round(\frac{\arctan(\frac{s_{01}}{s_{10}})+\pi}{2\pi})$ $\quad s_{01} \geq 0, s_{10} < 0$

   $Round(\frac{\arctan(\frac{s_{01}}{s_{10}})-\pi}{2\pi})$ $\quad s_{01} < 0, s_{10} < 0$

   where n is the orientation quantisation, otherwise classify as *flat*

3. Count pairs of feature combinations across all locations to give histogram of size $(n+1)^2$, where n is the orientation quantisation

4. Normalise histogram by dividing by total number of locations

Whilst the results from Experiment 5.1 were promising, we wanted to ensure that the column features were contributing to this, rather than just the use of features at two scales across the image. To test this we used a simple multiscale scheme, where a histogram of oriented gradients was computed at two scales and then the histograms concatenated to form the encoding. Thus, in this scheme, there was no link between features at different scales at each location. The details are given in Experiment 5.2 on page 101.

From the results of this experiment, it was clear that just combining histograms at different scales did not have the same effect as the using the column features.

5.3 BIF Columns

We then applied the same process for BIFs, combining them into features referred to as BIF columns. The experimental details are given in Experiment 5.3 on page 102.

Whilst BIF columns showed an improvement over the simple BIF histograms, the performance was still well below that of other schemes, including the intensity values. However, it was interesting to note that BIF columns performed best whilst using 3 scales. Thus we have the BIF column algorithm, as given in Algorithm 5.2.

Algorithm 5.2 The BIF Column scheme

1. For a given σ_{BASE} and scale ratio, r, calculate BIFs as described in Algorithm 2.1 at scales σ_{BASE}, $r\sigma_{BASE}$ and $r^2\sigma_{BASE}$.
2. Count pairs of BIF combinations across all locations to give a histogram of 343 bins
3. Normalise histogram by dividing by total number of locations

5.4 oBIF Columns

Finally we tried the column scheme with oBIFs. The details are given in Experiment 5.4 on page 103.

Overall the performance of oBIF columns was very similar to oriented gradient columns, with a large increase in performance over a single scale oBIF histogram. The encoding is still strongly invariant to shifting, as shown in graph (b). The scale ratio parameter also appears to be stable around the optimal value. As with the oriented gradient columns, the difference in performance between the two and three scale schemes was very small so, given the comparative size of the encodings, we concluded that the two scheme was preferable.

Whilst it was tempting to conclude that, given the similar performance between oriented gradients and oBIFS, that local symmetry information does not contribute to performance, there was an interesting difference. As can be seen from the tuning curves for the orientation quantisation, the performance for oBIF columns with a low orientation quantisation is substantially higher than for oriented gradient columns. This implies that, under certain circumstances, the local symmetry type information does aid performance. When compared to the performance for BIF columns, where no orientation is used, there appeared to be a large jump in performance when orientation is first introduced. This is demonstrated by the change in performance from BIF columns to oBIF columns with an orientation quantisation of just 2, which changes from 55% to almost 75% when using 10 training images per class.

Algorithm 5.3 The oBIF Column scheme

1. For a given σ_{BASE} and scale ratio, r, calculate oBIFs as described in Algorithm 2.2 using orientation quantisation n at each scale.
2. Count pairs of oBIF combinations across all locations to give a histogram of $(5n + 3)^2$ bins, where n is the orientation quantisation
3. Normalise histogram by dividing by total number of locations

5.5 Weighted schemes

Next we wanted to investigate whether it was better to use the thresholded system, where each location in the image is assigned to a class only, or a weighted system, where each location is assigned a class and a weight. The motivation for this was that many other schemes, including HOG and SIFT use weights for the strength of gradient.

One advantage of moving to a system of weighted features is that we lose the need for the threshold parameter as locations which would have previously been classed as *flat* would have a low or zero weight. However, perhaps the main advantage would be that the system effectively selects locations that contain high contrast features. If such features are more relevant for establishing object identity, this is likely to be of benefit in a recognition task. The same effect would occur to a certain extent when using a threshold, but in a weighted system this effect is continuous.

A key consideration when using column features is in how to determine the weight of the column, given the weights of its constituent features. In single scale schemes, the use of a weight gives prominence to features with a higher contrast compared to a thresholded system. When using two scales the choice was whether to combine the individual weights through addition or through multiplication. If addition was used, emphasis would be given to columns that have a higher contrast feature at either scale whereas multiplication would give emphasis to columns with higher contrast features at both scales. Given that the main argument for using weights over a threshold was that it automatically selected *stronger* features, it seemed natural to see *stronger* columns as being those with high contrast features at both scales. Therefore we combined the weights through multiplication. This gave us the weighted oriented gradient column scheme, as detailed in Algorithm 5.4.

A similar argument was used to develop the weighted oBIF column scheme. The main difference here was that we had to use the strengths for each of the different BIF types, rather than simply the gradient strength. The algorithm for weighted oBIF columns is given in Algorithm5.5.

We tested both weighted schemes in the same way as the standard column schemes, the details of which are given in Experiment 5.5 on page 104.

For both oriented gradients and oBIFs there is a slight performance improvement in using the weighted schemes over a threshold. Combined with the removal of the need for the threshold parameter, this would appear to make these schemes preferable over the thresholded versions.

Algorithm 5.4 The Weighted Oriented Gradient Column scheme

1. For a given σ_{BASE} and scale ratio, r, measure filter responses $c_{1,0}$ and $c_{0,1}$ of 1st order derivative-of-Gaussian filters at scales σ_{BASE} and $r\sigma_{BASE}$, and from these calculate the scale normalised filter responses $s_{ij} = \sigma^{i+j} c_{ij}$

2. Calculate quantised orientations, θ_1 and θ_2, at both scales according to:

   $Round(\frac{\arctan(\frac{s_{01}}{s_{10}})}{2\pi})$ $\quad s_{10} > 0$

   $Round(\frac{\arctan(\frac{s_{01}}{s_{10}})+\pi}{2\pi})$ $\quad s_{01} \geq 0, s_{10} < 0$

   $Round(\frac{\arctan(\frac{s_{01}}{s_{10}})-\pi}{2\pi})$ $\quad s_{01} < 0, s_{10} < 0$

3. Compute weight, w_1 and w_2, according to $\sqrt{s_{10}^2 + s_{01}^2}$ at both scales

4. Combine across scales to form a column feature with orientation (θ_1, θ_2) and weight $w_1 w_2$

5. Sum weights across each possible orientation combination to create a histogram of size n^2, where n is the orientation quantisation

6. Normalise by dividing by sum of all column weights

Algorithm 5.5 The Weighted oBIF Column scheme

1. For a given σ_{BASE} and scale ratio, r, measure filter responses $c_{1,0}$ and $c_{0,1}$ of 1st order derivative-of-Gaussian filters at scales σ_{BASE} and $r\sigma_{BASE}$, and from these calculate the scale normalised filter responses $s_{ij} = \sigma^{i+j} c_{ij}$

2. Compute $\lambda = s_{20} + s_{02}, \gamma = \sqrt{(s_{20} - s_{02})^2 + 4s_{11}^2}$ at each location and scale

3. Assign BIF type and weight, w, according to which *Expression* is largest, then calculate orientation where appropriate:

Expression	*BIF type*	*Quantisable orientation*	*Orientations*
$2\sqrt{s_{10}^2 + s_{01}^2}$	*slope*	$\arctan(\frac{s_{01}}{s_{10}})$ $\quad s_{10} > 0$ $\arctan(\frac{s_{01}}{s_{10}}) + \pi$ $\quad s_{01} \geq 0, s_{10} < 0$ $\arctan(\frac{s_{01}}{s_{10}}) - \pi$ $\quad s_{01} < 0, s_{10} < 0$	$2n$
λ	*dark rotational*	*No orientation*	0
$-\lambda$	*light rotational*	*No orientation*	0
$(\gamma + \lambda)/\sqrt{2}$	*dark line*	$\arctan(2s_{1,1}/(s_{0,2} - s_{2,0} + \gamma)$	n
$(\gamma - \lambda)/\sqrt{2}$	*light line*	$\arctan(2s_{1,1}/(s_{0,2} - s_{2,0} + \gamma)$	n
γ	*saddle-like*	$\arctan(2s_{1,1}/(s_{0,2} - s_{2,0} + \gamma)$	n

4. Combine across scales to form a column feature with class $(oBIF_1, oBIF_2)$ and weight $w_1 w_2$

5. Sum weights across each possible oBIF combination to create a histogram of size $(5n + 2)^2$, where n is the orientation quantisation

6. Normalise by dividing by sum of all column weights

However, a key downside of using weights is that we may lose invariance to contrast changes across an image. This is unlikely to pose a problem when using datasets such as MNIST but may be issue when dealing with other sets. In schemes such as HOG, this is overcome by normalising individual histograms within the grid, which showed a strong improvement in performance in pedestrian detection [53]. Whilst normalisation is not as straightforward within the column schemes, because we do not use local histograms, it would still be possible to introduce local normalisation of weights without having to introduce a grid into the final encoding.

5.6 Rotational Invariance

The next issue we considered was how to make the histograms invariant to rotations of the image. As both BIFs and the column process were invariant to rotations, BIF columns themselves should be automatically rotationally invariant. In order to test this we used the rotated MNIST set. The details are given in Experiment 5.6 on page 105.

As oriented gradients and oBIFs contain local orientation information, they are not naturally rotationally invariant. In order to produce a rotationally invariant version of these schemes there were two main options. Either we could create rotationally invariant features (as in schemes such as [200]), by encoding the relative orientation between features at the two scales, or we could create a rotationally invariant histogram, by rearranging the bins so that they are aligned to a dominant orientation, an approach more similar to SIFT [144]. As our aim was to produce a rotationally invariant encoding of the image, we concentrated on the rotationally invariant histogram schemes.

In order to rearrange the histogram we needed to determine a dominant orientation for the image, as is done in schemes such as SIFT. For oriented gradients, we decided to do this by considering only the orientations that occurred at the coarser scale and selecting the most commonly occurring orientation as the dominant one.

Once the dominant orientation had been established we rearranged the bins of the histogram according to the deviation of the dominant orientation from the vertical. This ensure that the resulted encoding was equivalent to that which would be obtained for an identical image aligned to the vertical. This process was referred to as histogram rotation.

The oriented gradient column scheme with histogram rotation was tested using the MNIST and rotated MNIST sets, as described in Experiment 5.7 on page 106.

We used a similar process to create rotationally invariant oBIF Column histograms,

which was tested in Experiment 5.8 on page 107.

Both rotationally invariant schemes showed a significant drop in performance compared to the standard schemes. This was to be expected, as with objects such as digits, the orientation of the object is useful in establishing object identity. For example, certain '1's and '7's can be very similar in shape but the orientation of the digit indicates which label it should have.

5.7 Scale Averaging

All the features considered so far had used two scale parameters, the base scale and the scale ratio. Whilst the scale ratio had shown a consistent optimal value of 2 or 2.25 with stable performance around this value, the optimal base scale seemed to show more variation. We were interested to see whether we could mitigate the effect of the choice of base scale.

To do this we looked at calculating the column histograms at a range of scales and then combining them by taking the mean histogram. We tested this scheme with oriented gradient columns, as described in Experiment 5.9 on page 108, and oBIF columns in Experiment 5.10 on page 109.

Finally we applied the scale averaging to weighted oriented gradient columns and oriented gradient columns, described in Experiment 5.11 on page 109.

5.8 Discussion

5.8.1 Comparison of Results

The performance of the difference column schemes is summarised in Table 5.1 with the computational performance being given in Table 5.2. Here we see that the oriented gradient and oBIF column schemes both outperform BIF columns. With both of these schemes, performance is marginally better when using weighted features, rather than a threshold. A more significant increase in performance is achieved through scale averaging.

Table 5.1: Comparison of performance (in % correct) for the column schemes

Scheme	Training images per class		
	2	10	100
Oriented gradients columns(OGCs)	65.9±3.9%	85.0 ±1.5%	94.1 ±0.9%
Simple multiscale OG	44.2±3.8%	65.7 ±1.8%	82.6 ±1.4%
BIF columns (3 scale)	39.4±3.2%	55.3 ±2.6%	69.9 ±1.4%
oBIF columns	61.8±4.1%	83.1 ±1.5%	93.9 ±1.0%
Weighted OGCs	66.4±3.6%	86.0 ±1.3%	94.9 ±0.9%
Weighted oBIF columns	65.0±4.6%	86.2 ±1.5%	95.4 ±0.8%
OGCs with histogram rotation	38.6±3.2%	59.9 ±2.1%	81.4 ±1.6%
oBIF columns with histogram rotation	29.2±5.1%	52.1 ±1.9%	78.2 ±1.4%
OGCs with scale averaging	69.5±4.4%	87.5 ±1.4%	95.6 ±0.7%
oBIF columns with scale averaging	69.6±4.4%	88.0 ±1.2%	95.2 ±0.8%
Weighted OG columns with scale averaging	72.1±3.8%	89.9 ±1.0%	96.3 ±0.8%
Weighted oBIF columns with scale averaging	71.1±4.9%	90.4 ±1.0%	96.8 ±0.5%

Table 5.2: The computational performance for each of the column schemes

The encoding time is given for each image, along with the classification time for a Nearest Neighbour classifier using 10 images per class. The times are based upon an implementation in Mathematica 7.

Scheme	Size	Computation time (ms)	
		Encoding	Classifier
Oriented gradients columns(OGCs)	64	0.068	1.6
Simple multiscale OG	16	0.025	0.49
BIF columns (3 scale)	49	0.011	1.1
oBIF columns	529	0.089	1.6
Weighted OGCs	64	0.59	1.1
Weighted oBIF columns	529	0.68	1.6
OGCs with histogram rotation	64	0.068	1.1
oBIF columns with histogram rotation	529	0.12	1.6
OGCs with scale averaging	64	0.52	1.1
oBIF columns with scale averaging	529	0.65	1.6
Weighted OG columns with scale averaging	64	4.5	1.1
Weighted oBIF columns with scale averaging	529	5.0	1.6

5.8.2 Feature changes across scale

For each of the feature types there appears to be a large improvement in performance when using columns rather than single scale features. The key difference must come from the fact that the feature types changes over scale, otherwise the column histograms would contain exactly the same information as the single scale histograms. However, if the key information is in the changes of feature type across scale, it is unclear as the best way to capture this information. As our feature types are quantised, the location of the changes themselves will move in scale space according to where the boundaries are placed. For example, if we look at Figure 5.3, we see that the majority of changes in the example figure are of a single class change in orientation.

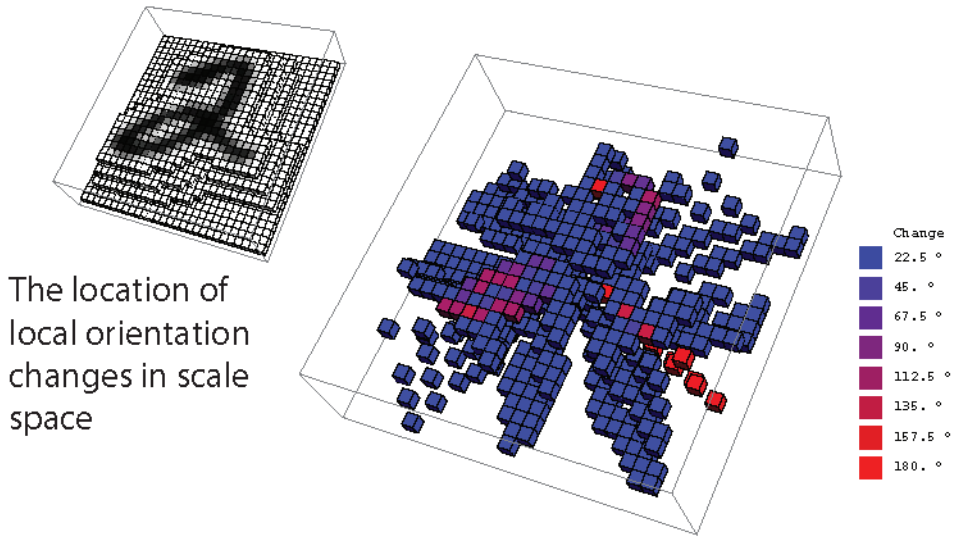


Figure 5.3: The changes in orientation across locations in scale space.
The majority of changes reflect a small change in local orientation. However, at certain locations, along the outline of the digit, the local orientation change across scale is far greater. This suggests that particular column features capture different elements of the scale space structure of the digit.

Thus, if the exact location is dependent on the position of the quantisation boundaries, perhaps the best way is to see whether a change occurs within a certain region. This is exactly what happens with the column features, where the scale ratio specifies the range in the scale dimension over which we will capture a change. This may explain why the scale averaging has a positive effect, since it enable us to capture feature changes over a wider portion of scale space, whilst still looking for changes of a certain size.

A further question that we have to consider is how much feature change occurs in scale space. In Figure 5.4, we show the number of feature changes for each location in the image for oriented gradients and oBIFs.

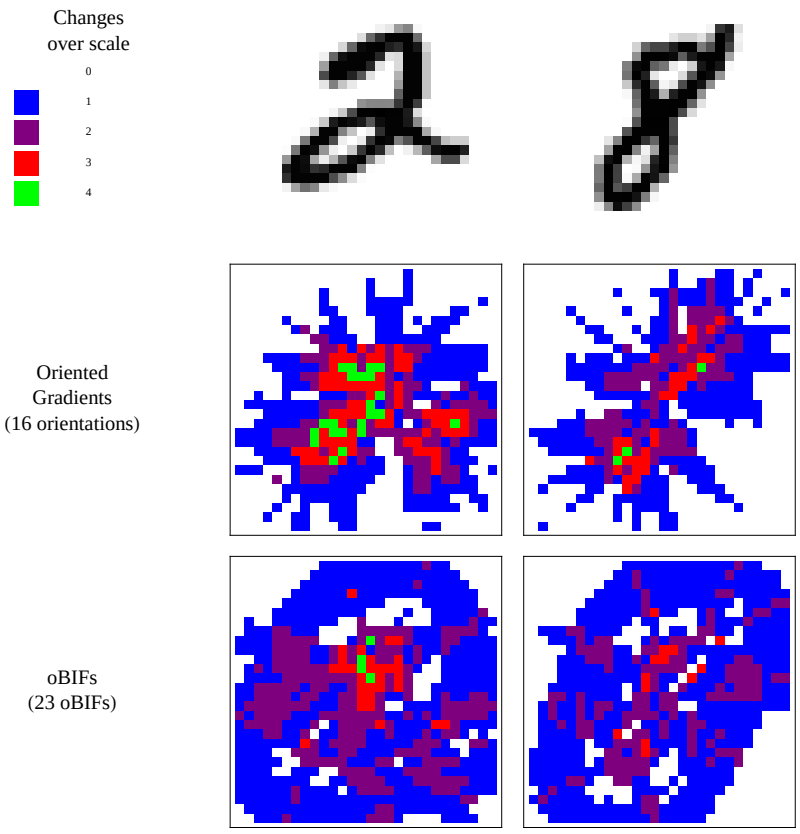


Figure 5.4: The number of feature changes across scale for each location in the image. Certain elements of the structure of the digit, such as junctions, appear to produce a greater number of feature changes across scale than others.

Here we see that many locations have a single change. A single scale histogram would not contain any information about these changes. A two scale column system may capture a large amount of the information about locations with a single change. However, the difference between a two and a three column system may largely depend upon capturing locations with more than one feature change. As we can see in the example images, whilst these are present, they are in a minority. This may mean that the difference between a 2 scale and a 3 scale histogram is very small for the MNIST dataset. If this were true it would explain the similar levels of performance for 2 and 3 scale systems for both oriented gradients and oBIFs.

Given the nature of the feature changes across scale it is interesting to try and relate these to the underlying physical structure of the digit. With oriented gradients, we would expect the orientation to point away from the dominant dark patch at the given scale. At a very coarse scale, every digit will effectively appear as a dark blob, meaning that the orientations should all point towards the border of the image. As we move to a finer scale, patches of light appear in certain regions of the digit, such as in the middle of the round sections of the '8'. As these areas appear we would expect the local orientation to change so that it points away from the light area. Thus we might expect the greatest number of

orientation changes to occur in the middle of the areas of light at different scales.This appears to be the case foe both the '2' and the '8' in Figure 5.4, with the greatest number of local orientation changes found in between different regions of light.

With oBIFs, the situation appears to be similar as with oriented gradients although the overall number of feature changes across scale is lower. In particular, the number of feature changes appears to be far lower along line sections of the digit. This may be because, along such sections of the digit, the assignment of orientation is unstable as it changes rapidly across the middle of the line segment. This is not so much of an issue with oBIFs, where line segments are assigned an orientation along the line as opposed to in the direction of the gradient. Thus, along line segments, the oBIF classification is typically stable across scale and we observe very few feature changes at these locations.

5.9 Summary and conclusions

In this chapter we have introduced the idea of column features, which concatenate lower level features across scale. We have presented these using three different feature sets, oriented gradient columns (OG Columns), Basic Image Feature Columns (BIF Columns) and oriented Basic Image Feature Columns (oBIF Columns).

When tested using the datasets presented in Chapter 3, the oriented gradient column and oBIF Column schemes have outperformed the histograms and spatial binning schemes from Chapter 4. We have proposed that these schemes have an advantage over the spatial binning schemes in that they require a single parameter, the scale ratio, whereas the spatial binning schemes involve a grid structure.

We have also investigated the performance of the schemes under various aspects of variation and proposed rotationally invariant and scale averaged versions of the column schemes. In addition we have shown that a version of the scheme which uses weighted gradient strengths outperforms the standard column schemes.

In the next three chapters we will describe the evaluation of the schemes presented in this chapter, using problems in three different application areas. The first of these, presented in Chapter 6, involves the recognition of characters taken from natural images.

Experiment 5.1 Oriented gradient columns with the MNIST datasets

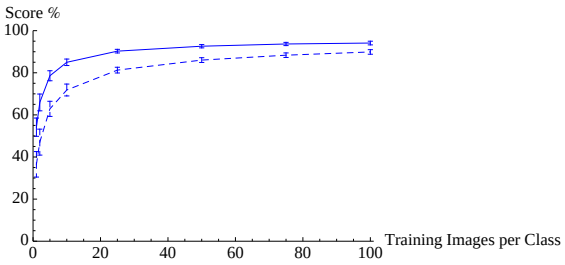
Methods

The same MNIST sets as in previous experiments were used. Images were encoded into oriented gradients columns, as described in Algorithm 5.1. The algorithm takes four parameters, which are the orientation quantisation, the base scale, the scale ratio and the threshold. The parameter values were tuned using a single sweep through each parameter using the tuning set of images, as described in previous experiments. Parameters were tuned in the order stated above.

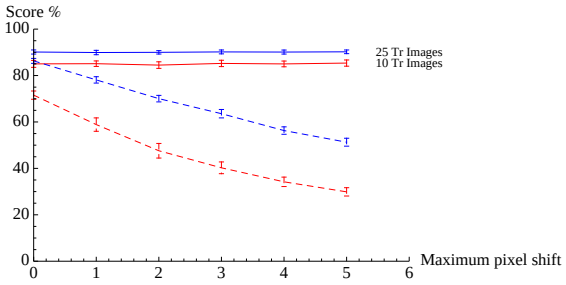
Using the tuned parameter values, the main set of images was encoded into oriented gradient columns. For a given training set size, images were randomly selected from each class for training purposes, with the rest being used as a test set. This process was repeated fifty times, with the mean and standard deviation of scores being reported. Classification was done using a Nearest neighbour classifier with the Bhattacharyya distance.The process was then repeated for the shifted MNIST dataset. Finally the performance was tested using column features with 3 scales per column.

Results

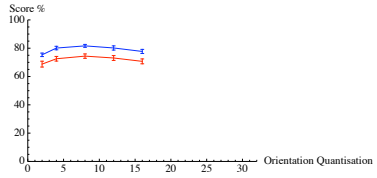
The performance for different sizes of training set is given in graph (a), where the score was $\mathbf{65.9 \pm 3.9\%}$, $\mathbf{85.0 \pm 1.5\%}$ and $\mathbf{94.1 \pm 0.9\%}$ for 2, 10 and 100 training images per class respectively. The performance on the shifted MNIST dataset being given in graph (b). The tuning curves for each of the four parameters are given in graphs (c) to (f). The performance for the 3 scale scheme, using 100 training images per class, is given in graph (g), along with the score from Experiment 4.1. Dashed lines give benchmark performance (from Chapter 3) and in the tuning graphs, curves are given for 10 images per class (red) and 25 (blue).



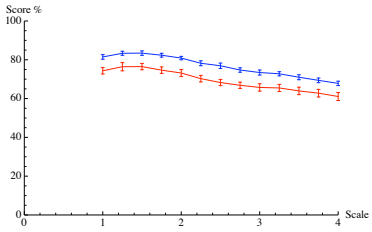
(a) Oriented Gradient Columns with the MNIST set



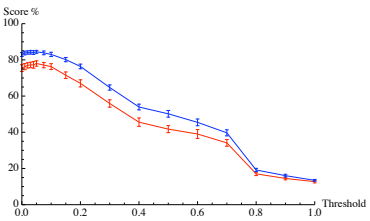
(b) Oriented Gradient Columns with the shifted MNIST set



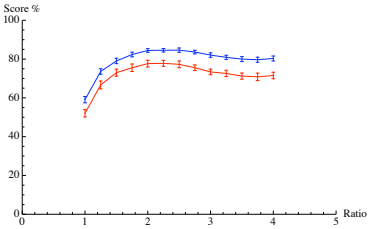
(c) Orientation quantisation tuning



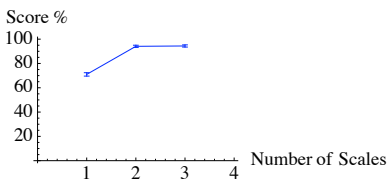
(d) Scale tuning



(e) Threshold tuning



(f) Ratio tuning



(g) Number of scales

Experiment 5.2 Simple multiscale oriented gradient scheme

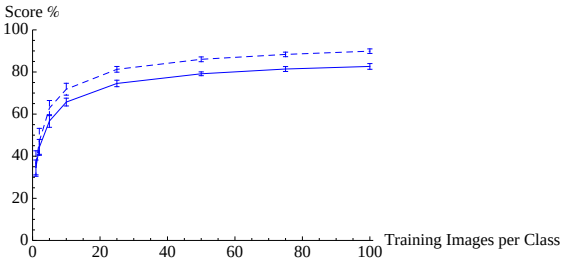
Methods

Using the same datasets as in Experiment 5.1 images were encoded into oriented gradients at two scales. Histograms were then created at each scale by counting the number of occurrences each orientation at each scale, and then normalised by dividing by the total number of locations in the image. The two histograms were then concatenated to create the final encoding for each image.

There were four parameters to tune, which were the orientation quantisation, the base scale, the scale ratio and the threshold. Parameters were tuned with a single sweep for each in the order listed. Classification was performed using a Nearest Neighbour classifier with the Bhattacharyya distance. Training and test set selection was done as in Experiment 5.1.

Results

The performance for different sizes of training set is shown in graph (a), with the performance for 2, 10 and 100 training images per class respectively being $\mathbf{44.2 \pm 3.8\%}$, $\mathbf{65.7 \pm 1.8\%}$ and $\mathbf{82.6 \pm 1.4\%}$. Dashed lines give benchmark performance (from Chapter 3).



(a) Simple 2 scale oriented gradients

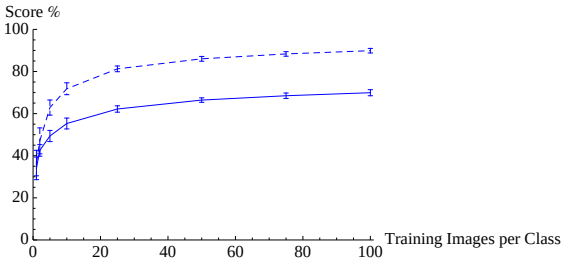
Experiment 5.3 BIF Columns with the MNIST datasets

Methods

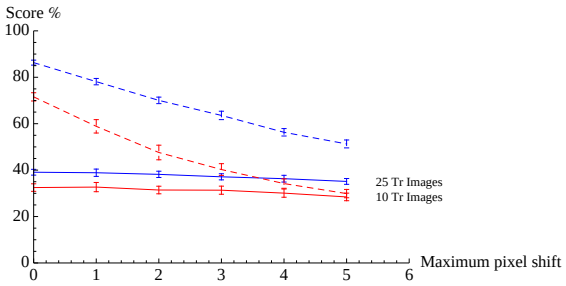
Using the same datasets as in Experiment 5.1, images were encoded as BIF columns. For the three parameters, the base scale, scale ratio and threshold, values were selected using a single parameter sweep for each in the order listed. The tuned values were then used to encode the main set of images. Multiple training and test sets were then drawn with the mean and standard deviation being reported. This process was then repeated for the shifted MNIST dataset. Finally, we repeated the process for 3 and 4 scale columns.

Results

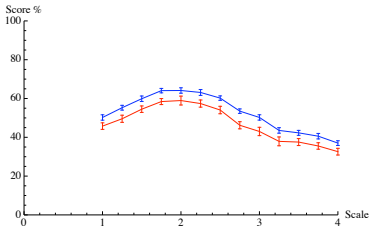
The performance for different sizes of training set with the MNIST dataset is shown in graph (a). The performance on the shifted MNIST set is given in graph (b) with the three tuning graphs given in (c) - (e). The performance for the 3 and 4 scale schemes are shown in graph (f), where the 3 column scheme was the best performing where the scores for 2, 10 and 100 training images per class were $\mathbf{39.4 \pm 3.2\%}$, $\mathbf{55.3 \pm 2.6\%}$ and $\mathbf{69.9 \pm 1.4\%}$ respectively. Dashed lines give benchmark performance (from Chapter 3) and in the tuning graphs, curves are given for 10 images per class (red) and 25 (blue).



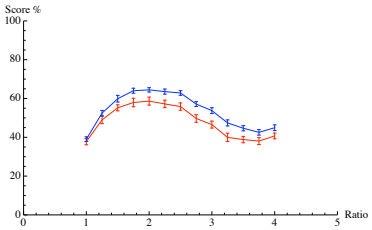
(a) BIF Columns with the MNIST set



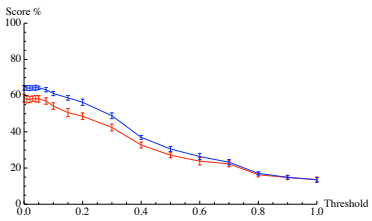
(b) BIF Columns with the shifted MNIST set



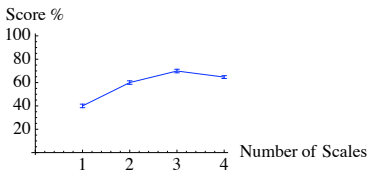
(c) Scale tuning



(d) Scale ratio tuning



(e) Threshold tuning



(f) Number of scales

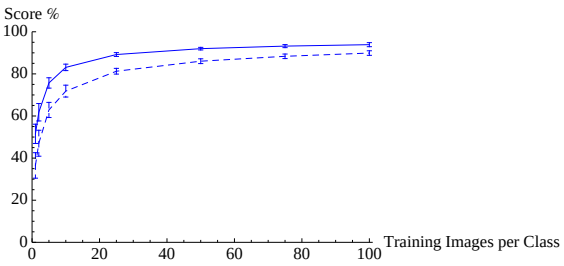
Experiment 5.4 oBIF Columns with the MNIST datasets

Methods

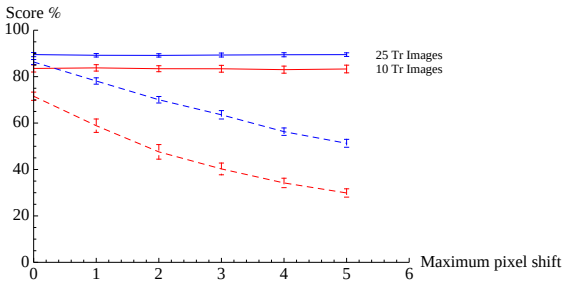
The same datasets were used as in Experiment 5.1. As in that experiment, there were four parameters to tune which were the orientation quantisation, the base scale, the scale ratio and the threshold. These parameters were tuned, using the tuning set of images, with a single parameter sweep for each in the order listed. Images from the test set were then encoded in oBIF columns, as detailed in Algorithm 5.3. Multiple training and test sets were then randomly selected, as in Experiment 5.1, and the mean and standard deviation are reported. Classification was done using a Nearest Neighbour classifier with the Bhattacharyya distance. This process was then repeated for a 3 scale scheme.

Results

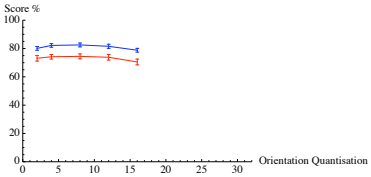
The results for different sizes of training set are given in graph (a), where the performance for 2, 10 and 100 training images per class was $\mathbf{61.8 \pm 4.1\%}$, $\mathbf{83.1 \pm 1.5\%}$ and $\mathbf{93.9 \pm 1.0\%}$ respectively. The results for training sets of 10 and 25 images per class using the shifted MNIST are given in graph (b). The four parameter graphs are given in (c) - (f). The comparison of performance, using 100 training images per class, for different number of scales is given in graph (g). Dashed lines give benchmark performance (from Chapter 3) and in the tuning graphs, curves are given for 10 images per class (red) and 25 (blue).



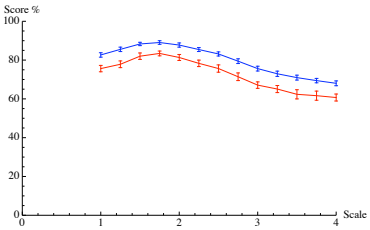
(a) oBIF Columns with the MNIST set



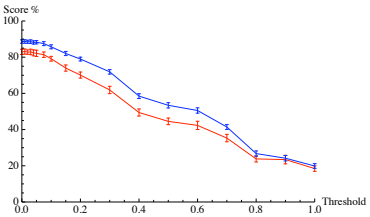
(b) oBIF Columns with the shifted MNIST set



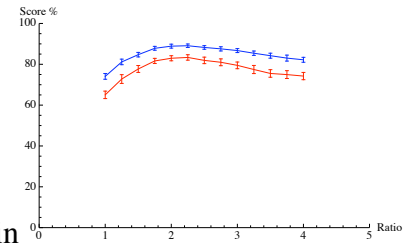
(c) Orientation quantisation tuning



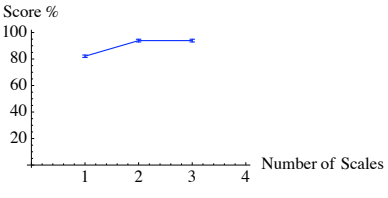
(d) Scale tuning



(e) Threshold tuning



(f) Ratio tuning



(g) Number of scales

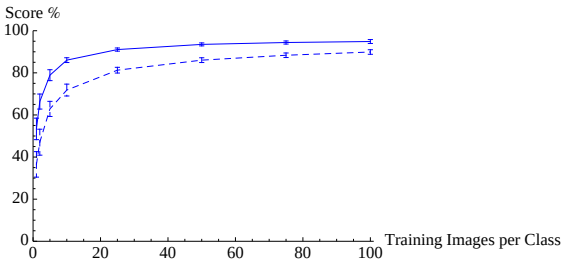
Experiment 5.5 Weighted feature column histograms

Methods

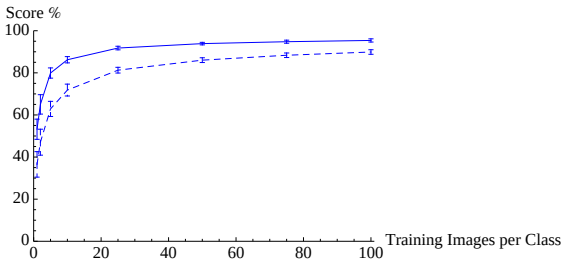
Images were encoded into weighted oriented gradient histograms, as detailed in Algorithm 5.4. There were three parameters to tune, which were the orientation quantisation, the base scale and the scale ratio. These were tuned with a single parameter sweep in the order given. We used the same experimental scheme as in Experiment 5.1 for classification and sampling of multiple training and test sets. This process was then repeated for Weighted oBIF columns, using Algorithm 5.5.

Results

The performance for different sizes of training set for weighted orient gradient columns is given in graph (a), where the score for 2, 10 and 100 training images per class was $\mathbf{66.4 \pm 3.6\%}$, $\mathbf{86.0 \pm 1.3\%}$ and $\mathbf{94.9 \pm 0.9\%}$ respectively. The performance for weighted oBIF columns is given in graph (b), where the scheme scored $\mathbf{86.2 \pm 1.5\%}$ and $\mathbf{95.4 \pm 0.8\%}$ for training sizes of 10 and 100 training images per class. Dashed lines give benchmark performance (from Chapter 3).



(a) Weighted oriented gradients columns



(b) Weighted oBIF columns

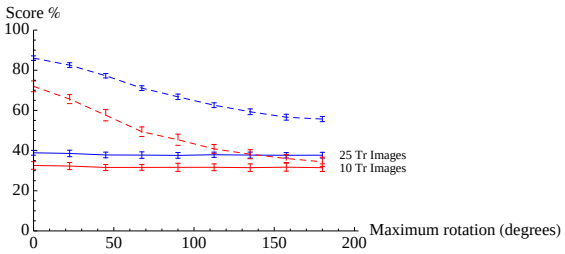
Experiment 5.6 BIF Colums and the Rotated MNIST set

Methods

Using the rotated MNIST set, images were encoded into 3 scale BIF columns using the tuned parameter values from Experiment 5.3. Classification, as before, was performed using a Nearest Neighbour classifier with the Bhattacharyya distance. Training and test selection was performed as in Experiment 5.1.

Results

The performance for training set sizes of 10 and 25 images per class is shown in graph (a). Dashed lines give benchmark performance (from Chapter 3) and in the tuning graphs.



(a) BIF Columns and the Rotated MNIST set

Experiment 5.7 Rotationally invariant oriented gradient columns

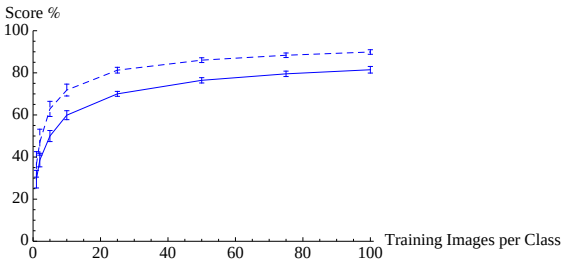
Methods

Images were encoded into oriented gradient columns using the parameter values from Experiment 5.1. Normalised histograms were then calculated and, for each image, the dominant orientation was established by finding the orientation with the greatest number of occurrences at the coarser scale. The bins of the histogram were then reordered, as described previously.

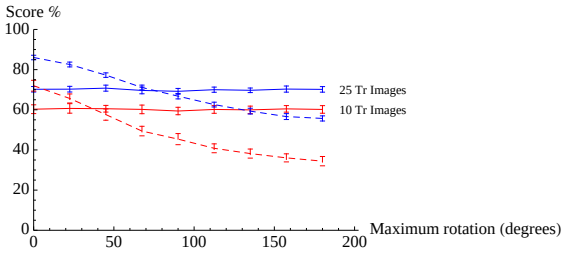
Classification and sampling multiple training and test sets followed the same procedure as previous experiments.

Results

The performance for different sizes of training set on the MNIST set is given in graph (a), where the score for 2, 10 and 100 training images per class was $\mathbf{38.6 \pm 3.2\%}$, $\mathbf{59.9 \pm 2.1\%}$ and $\mathbf{81.4 \pm 1.6\%}$. The performance on the rotated MNIST set is shown in graph (b). Dashed lines give benchmark performance (from Chapter 3).



(a) Rotationally invariant OGC



(b) Rotationally invariant OGC with Rotated MNIST

Experiment 5.8 Rotationally invariant oBIF columns

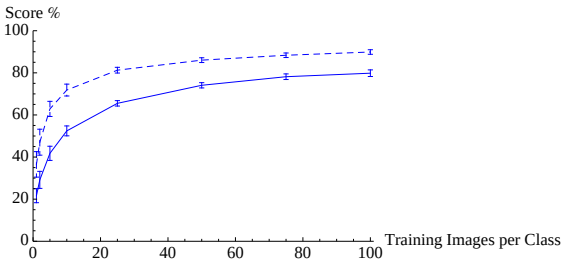
Methods

Images were encoded into oBIF columns using the parameter values from Experiment 5.4. Normalised histograms were then calculated and the dominant orientation, for each image, was established by finding the orientation of the grey oBIF with the greatest number of occurrences at the coarser scale. Histograms were then reordered, as described in the histogram rotation process previously.

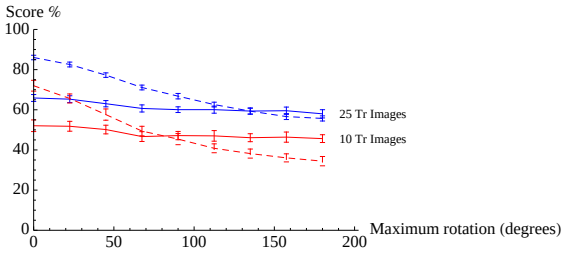
Classification and training and test set selection was then done as in Experiment 5.1.

Results

The performance for different sizes of training set is given in graph (a), where the score for 2, 10 and 100 training images per class was $\mathbf{29.2 \pm 5.1\%}$, $\mathbf{52.1 \pm 1.9\%}$ and $\mathbf{78.2 \pm 1.4\%}$ respectively. The performance on the rotated MNIST set, with 10 and 25 training images per class is shown in graph (b). Dashed lines give benchmark performance (from Chapter 3).



(a) Rotationally invariant oBIFC



(b) Rotationally invariant oBIFC with Rotated MNIST. The dashed line is the performance using intensity values from Chapter 3.

Experiment 5.9 Scale averaged oriented gradient columns

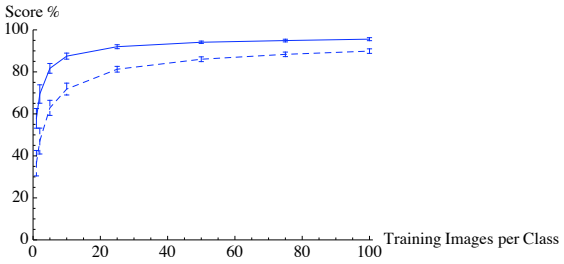
Methods

We used the MNIST set and the scaled MNIST set. Images were encoded into oriented gradient columns for a range of base scales. Histograms were then calculated for each base scale. Then, for each image, the scale averaged histogram was taken by calculating the mean of the histograms, as described previously. This process involved the four parameters from Experiment 5.1 and, in addition, the range of base scales to be used. For the first four parameters, the same values as in Experiment 5.1 were used. In order to determine the range of base scales, we first fixed the mid point of the range as the tuned base scale from Experiment 5.1. We then tuned for both the range of scales, and the number of scales within that range using the tuning set of images.

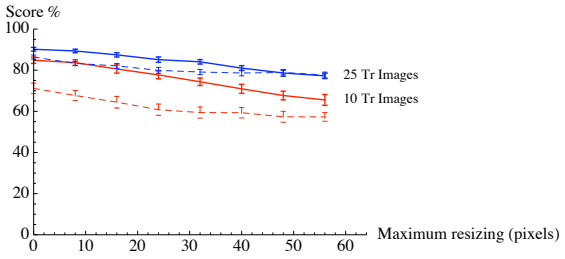
Classification and training and test set selection was performed as in previous experiments.

Results

The performance for different sizes of training set on the MNIST set is given in graph (a) where the performance for 2, 10 and 100 training images per class was $\mathbf{69.6 \pm 4.4\%}$, $\mathbf{87.5 \pm 1.4\%}$ and $\mathbf{95.6 \pm 0.7\%}$ respectively. The performance on the scaled MNIST set is given in graph (b). Dashed lines give benchmark performance (from Chapter 3).



(a) Scale averaged OGC



(b) Scale averaged OGC with Scaled MNIST

Experiment 5.10 Scale averaged oBIF columns

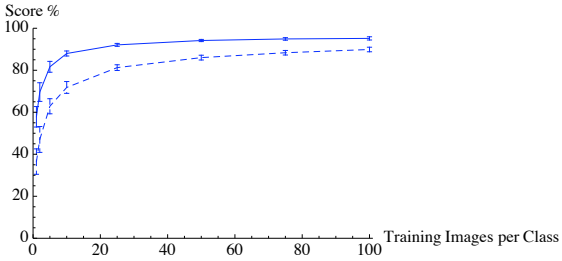
Methods

The MNIST and scaled MNIST sets were used. Images were encoded into oBIF columns at a range of base scales. Scale averaged histograms were then calculated, as given in 5.9. The values for the orientation quantisation, scale ratio and threshold were taken from the tuning process in 5.4. To determine the range of base scales, we first set the mid point of the range as the tuned base scale from Experiment 5.4. We then tuned for the range of scales, and then the number of scales calculated with that range, using a single paramter sweep for each.

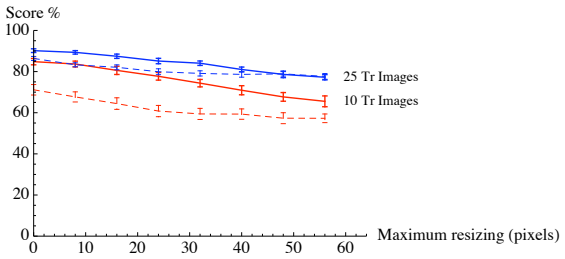
Classification and training and test set selection then proceeded as in Experiment 5.1.

Results

The performance for different sizes of training set is shown in graph (a), where the performance for 2, 10 and 100 training images per class was $\mathbf{69.6 \pm 4.4\%}$, $\mathbf{88.0 \pm 1.2\%}$ and $\mathbf{95.2 \pm 0.8\%}$ respectively. The performance on the scaled MNIST is given in graph (b). Dashed lines give benchmark performance (from Chapter 3).



(a) Scale averaged oBIF Column



(b) Scale averaged oBIF Column with Scaled MNIST

Experiment 5.11 Scale averaged weighted columns

Methods

Using the parameters for the weighted schemes from Experiment 5.5 and the scale averaged schemes from Experiments 5.9 and 5.10, images were encoded into weighted scale averaged weighted orient gradient columns and oBIF columns. Classification and training and test set selection then proceeded as in previous experiments.

Results

For scale averaged weighted oriented gradient columns the performance for 2, 10 and 100 training images per class was $\mathbf{72.1 \pm 3.8\%}$, $\mathbf{89.9 \pm 1.0\%}$ and $\mathbf{96.3 \pm 0.8\%}$ respectively. For scale averaged weighted oBIF columns the performance for 2, 10 and 100 training images per class was $\mathbf{71.1 \pm 4.9\%}$, $\mathbf{90.4 \pm 1.0\%}$ and $\mathbf{96.8 \pm 0.5\%}$ respectively. Dashed lines give benchmark performance (from Chapter 3).

Chapter 6

Natural Image Character Recognition

6.1 Introduction

The exploration of encoding schemes, presented in the previous two chapters, was intended to be of use to a range of applications. As the methods were tested using handwritten digits, it seemed natural to first test the resulting methods on current problems involving character recognition.

For this purpose we selected the problem of character recognition in the context of images and graphics, where current state of the art methods indicate there is still room for improvement[57, 250]. Whilst the problem is of current interest, it is mature enough that there were existing datasetes which had been tested on a variety of methods, enabling comparison with our methods.

This chapter begins with a brief introduction to the problem of natural image character recognition, and its relationship with the form of character recognition used in our investigation. We then test the column features, along with single scale histograms, on the two most commonly used datasets. Finally we use the column features to propose extensions to the HOG scheme, which had previously shown leading performance on this problem.

6.2 From MNIST to chars74k

The MNIST dataset, which we used in our investigation, contained images that each contained a single handwritten digit. The images had been sufficiently preprocessed to ensure that each digit was centred, scaled and oriented which made recognition feasible with a very simple scheme such as matching intensity values with a Nearest Neighbour classifier, as shown in Experiment 3.1 on page 70. When used with more advanced learning techniques, very high levels of performance can be achieved, with scores over

99% when using the full MNIST set of 60000 training images and 10000 test images [185, 184].

Such high levels of performance would seem to indicate that, with these problems, we do not need to convert the images into a different representation to perform recognition. In a similar way, we might expect recognition of individual printed characters to be possible using purely learning techniques.

However, with these problems we are dealing with characters that have been produced with the intent to convey information in a specific format. In other cases we may have to recognise text and characters that are presented in a less convenient format. In particular, the rise in the number of devices with cameras has led to a great number of images in which the text may provide rich sources of information for understanding the contents of the scene. For example, cameras in motor vehicles may capture road signs or cameras on mobile telephones may capture shop names or place identifiers. In these cases of camera-based recognition we are facing a different set of challenges. [152, 133].

The first issue is detecting whether an image contains text and, if so, locating the region of text so that it can be interpreted. Many methods exist for this step both in single images [44, 224, 251, 43, 42, 176, 261] and in video [253]. Once text has been detected it can be either recognised using context [255, 68] or individual characters can be put through a recognition process. As we were using this problem to evaluate our system, this individual character recognition was the aspect of the problem which we focused on.

Individual characters, once extracted from the images, still possess many differences from images used in standard OCR problems.

- **Range of fonts** Printed text may contain a limited number of fonts. OCR methods may attempt to learn models of each font so that characters can be recognised, though font free OCR systems do exist [116]. When dealing with characters taken from natural images, the range of fonts is almost unlimited with text created in a vast number of different ways. Thus, it may not be possible to form any model of fonts before recognition, and instead any recognition system must be able to generalise from printed and handwritten styles in the training set to previously unseen styles.
- **Perspective** Text from documents is typically presented front on, meaning that the relative dimensions of characters should demonstrate some consistency. However, in images the viewpoint may not have been selected for convenient reading of the text and thus characters may appear at a whole range of different perspectives. Whilst there are methods to tackle this [45, 237, 175], we view this aspect of the problem as a useful test of the column system in its ability to generalise.

- **Clutter** As text found in images may not simply be there to convey information, but also for aesthetic appeal, characters may well appear on background textures or with additional elements intersecting them. Thus, we may have to deal with clutter within the image, even if the detection and segmentation stages are accurate.

6.3 Work related to the specific problem

Whilst many previous problems in character recognition have been tackled by well established OCR methods, problems in natural image character recognition have also been attempted with techniques more associated with object recognition [57], which is of great use in our work where we are interested in generic schemes.

Whilst SIFT has been used successfully in the recognition of Chinese characters[268], there is evidence that it performs poorly when dealing with perspective distortions of characters[121]. Given this, perhaps it is unsurprising that SIFT has not performed well when tested against other methods [57] on natural image character recognition.

In contrast, Shape Context [13] and Geometric Blur [14], have performed better than SIFT and other methods when used in conjunction with a Nearest Neighbour (NN) classifier [57]. However, in the same study it was shown that the best performing method was to combine all these features within a Multiple Kernel Learning framework.

Most recently Wang et al. [250] have shown that Histograms of Oriented Gradients (HOG) could be used in conjunction with a Nearest Neighbour classifier to produce better performance than all previous methods.

6.4 Datasets

In order to evaluate the column schemes we used the *chars74k* [57] and ICDAR03-CH [145] datasets. The first of these has 62 classes of images, made up of digits with upper and lower case letters. Images generally contained a single character from a natural image. However, high levels of clutter meant that some images contained small subsidiary characters, in which case only the main character is labelled. Examples from the chars74k are shown in Figure 6.1.

The ICDAR03-CH dataset comes from the robust OCR challenge section of the ICDAR03 challenges and contains 75 classes, made up of digits, upper and lower case letters and symbols. When referring to specific results we use the convention of suffixing the dataset name with the number of training images per class, so, for example,

chars74k-05 refers to the chars74k dataset when tested with 5 training images per class. Examples from the ICDAR03-CH dataset are shown in Figure 6.2.

Figure 6.1: Examples from the chars74k dataset.
The dataset consists of 62 classes, made up of upper and lower case letters and digits, with a wide variation of font types, backgrounds and orientations.

6.4.1 Preprocessing

As a first step, all images were resized and padded where necessary to ensure they were all the same size. The datasets contain both light letters on dark background and dark letters on a light background. In order to be invariant to this difference we performed a simple test on each image by looking at the relative strength of oBIF type at a coarse level, which essentially considered whether the image tended to a dark patch on light or a light patch on dark. If this oBIF type was a *light rotational* images were inverted, whereas if it was a *dark rotational* images were left as they were.

If this step were not performed, then the effective training set size would be halved as dark letters would only serve as training examples for other dark letters and not light letters. This could be overcome by introducing invariance to polarity into the oBIF column scheme, by summing each pair of bins representing opposing polarity column features. However, as we considered polarity invariance to be required only for specific tasks, we decided to use the simple step described above.

Figure 6.2: Examples from the ICDAR03-CH dataset.
The dataset differs slightly from the chars74k set in that it also contains punctuation marks. In addition, the orientation of characters appears to be more consistent than in the chars74k set.

6.4.2 Dataset splits

In order to make a comparison to previous work we focused on using training sets of 5 or 15 images per class. With such low numbers of images per class available for training the results can vary substantially from run to run. Therefore, we wanted to ensure the performance measures were based upon a significant number of trials. To do this we first selected 30 images from each class in the chars74k dataset from which to draw subsequent training and test sets. These images, referred to as the main set, contained the majority of available images for most classes.

We used the remaining images from chars74k to tune the parameter values for the four schemes. For this process none of the images from the main set were used with the consequence that the number of images available per class varied considerably in the tuning process. The same parameter values were used for both datasets.

6.5 Column Features

We tested the oriented gradient column scheme and the oBIF column scheme on the two datasets. For the purposes of comparison we also tested single scale histograms of oriented gradients and oBIFs. The details are given in Experiment 6.1 on page 123. As the performance of BIFs and BIF columns had been very poor compared to the other schemes when tested on the MNIST dataset, we did not believe that they would offer top performance on this application and so we chose not to test them explicitly. However, the performance of BIF columns is provided as part of our parameter investigation where we looked at the performance as the orientation quantisation reached zero.

6.5.1 Comparison of Performance

The results for the four schemes are shown in Table 6.1 alongside previously published results including SIFT, HOG, Shape Context and the OCR software ABBYY. Results are also given for an implementation of Multiple Kernel Learning[57], which uses multiple sets of features and a more advanced learning framework than Nearest Neighbour. The three columns indicate the dataset and the number of training images per class. The performance measure given is the mean score over all runs, which was 50 for the chars74k dataset and 20 on the ICDAR03 dataset, along with the standard deviation of these scores.

From this table it can be seen that, on the chars74k dataset, both oBIF columns and oriented gradient columns outperformed previous methods when using either 5 or 15 training images per class. However, on the ICDAR03 dataset only the oBIF column scheme outperforms previous methods.

Table 6.1: Comparison of performance (in % correct) on the chars74k and ICDAR03 datasets.

Scheme	Chars74k-5	Chars74k-15		ICDAR03-CH-5
SIFT [57]	-	20.8		-
ABBYY [250]	18.7	18.7		21.2
Multiple Kernel Learning [57]	-	55.3		-
Shape Context [57]	26.1±1.7	34.4		18.3
Geometric Blur [57]	36.9±1.0	47.1		27.8
MKL [57]		55.3		
HOG Features [250]	45.3±1.0	57.5		51.5
Oriented gradients	27.0±1.2	36.7±0.8		28.4±1.1
oBIFs	35.3±1.2	46.4±1.0		30.8±1.2
Oriented Gradient columns	50.8±1.1	60.2±1.1		46.1±1.3
oBIF columns	**53.4**±1.4	**64.3**±1.3		**52.7**±1.2

6.5.2 Confusion Matrix

As part of the evaluation process we looked at the pairs of classes that were often confused. For the oBIF column scheme and the chars74k dataset this is shown in the confusion matrix in Figure 6.3. The classes are given in the same order as in Figure 6.1, that is digits followed by upper case letters then lower case letters.

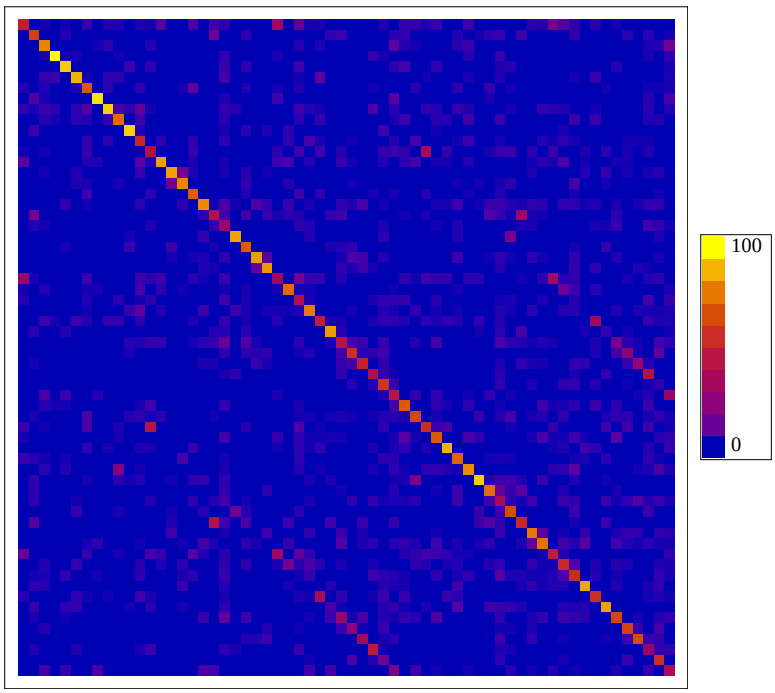


Figure 6.3: The confusion matrix for oBIF columns on the chars74k-15 task. The array is ordered in the same way as in Figure 6.1, with upper case letters first, followed by lower case letters and then digits. A notable feature of the confusion matrix is the pair of lines running parallel to the main diagonal, which indicate confusion between upper and lower case examples of the same letter.

A notable feature of the confusion matrix is the pair of lines running parallel to the main diagonal. These indicate a relatively high level of confusion in between upper and lower case examples of the same letter. If we allow confusion between upper and lower cases examples, thus reducing the problem to 36 classes the overall score becomes $73.0 \pm 1.3\%$ on chars74k-15.

A similar pattern was seen in the confusion matrices for the other schemes, as well as for the ICDAR03 dataset.

6.5.3 Parameter Investigation

As with the MNIST set in our investigation, we wanted to see how performance varied with each of the parameter values to ensure that the scheme was stable around optimal values. To do this we used chars74k-15 and varied each of the parameters in turn. The

details are given in Experiment 6.2 on page 124.

6.6 Discussion

6.6.1 Is it possible to achieve 100% performance?

The oBIF columns method tested within this work has shown an improvement over previous methods using the chars74k and ICDAR03 datasets. However, the performance is still far from perfect and, even with 29 training images per class we see performance flattening out at approximately 70% (as shown in graph (a) of Experiment 6.1).

In order to put this improvement into context it would be helpful to have some idea of the upper bound on performance. Ideally this would come from human level performance, but such an estimate is very difficult to make as it would require subjects who had no prior experience of the characters used in these datasets. We can, however, detect apparent ambiguities between classes that occur because of the nature of the testing regime where each character is presented out of context. Certain pairs of classes, for example an upper and lower case 'x' or a 'one' and a lower case 'l', may have visually identical instances meaning that the true class can only be identified by the context. Upper case letters may be larger than surrounding letters and found at the start of words, whereas digits may be found next to other digits. Examples of ambiguous pairs of classes are shown in Figure 6.4.

Figure 6.4: Examples of ambiguous images in the chars74k dataset.
All images in the top row are from a different class than the corresponding image in the bottom row.

The absence of these context cues will likely place an upper bound on performance. It is difficult to determine exactly what this upper bound might be, but if, for example, there were 10 pairs of visually ambiguous classes, as implied by the confusion matrices, we would have an upper bound of just under 84%.

6.6.2 How do oBIF columns compare to other methods?

The methods tested in this work are perhaps most comparable to the Shape Context, Geometric Blur and HOG features. All these methods involve the extraction of local

features, followed by some form of pooling step followed by a nearest neighbour classifier. There are however important differences between the methods. Both Shape Context and Geometric Blur combine local features in a way that will produce an encoding that is largely invariant to position and at least partially invariant to the size of the object. Therefore, when testing these methods on a set of upright characters, we are essentially evaluating their ability to categorise images.

However, when using HOG features, as local histograms are essentially concatenated to make the overall image encoding, we would not expect this to be particularly invariant to changes in position and size. When testing this method then, performance may vary according to how much size and position variation there is in the dataset, as well as with intraclass variation. Thus, given two datasets with different levels of variation in size and position, we might expect the relative performance between the datasets to differ between HOG and Geomtric Blur or Shape Context.

In the previously published results it is interesting to note the relative performance between Geomtric Blur, with 36.9% on chars74k-05 and 27.8% on the ICADR03-CH-5 test, and HOG features, which scored 45.3% on chars74k-05 and 51.5% on ICDAR03-CH-5. Whilst HOG features outperform Geometric Blur on both sets, it performs far better on ICDAR03 as opposed to chars74k whereas the opposite is true with Geomteric Blur. This may be down to differing levels of variation in position and size between the datasets. If the ICDAR03 set contained characters of relatively uniform position and size, but with a greater degree of intraclass variation, then this might explain the relative differences in performances between the methods.

As all the schemes tried in these experiments have used global histograms, and should therefore be highly position invariant and at least partially invariant to size changes, we might expect a relative performance more similar to Geometric Blur than to HOG features. This is roughly what is seen in the results, where oBIF columns outperform HOG features by a margin of 8.1% on chars74k-5 but only 1.2% on ICDAR03-CH-5.

6.7 Multiscale HOG

As an additional investigation, we wanted to establish whether column features could work in conjunction with the standard HOG scheme. The purpose of this was both to see whether performance could be improved and to see whether the column features were encoding the same information as schemes with spatial binning.

Using the HOG scheme, as described in Chapter 2, we considered two schemes that

extend the descriptor. The first scheme simply extends the histograms across scale space, in a way related to previous multiscale methods. The second scheme incorporates column features, where pairs of oriented gradients across scale are combined.

For the first scheme, oriented gradients are calculated using Derivative-of-Gaussian (DtG) filters. For each location in the image, at a given scale, a single orientation is assigned along with a weight, which is calculated from the response of the DtG filters. Next, for a given block size, we calculate the total strength for each orientation across the block and across all scales. This is repeated across multiple overlapping blocks within the image. Each histogram is then normalised so that the total weight across all orientations sums to one. All histograms are then concatenated to make a single descriptor for the image. The algorithm is given in Algorithm 6.1.

Algorithm 6.1 The simple multiscale HOG encoding

1. For a given scale, σ, measure filter responses c_{10} and c_{01} of 1st order derivative-of-Gaussian filters, and from these calculate the scale normalised filter responses $s_{ij} = \sigma^{i+j} c_{ij}$
2. Assign orientation by quantising:
   $\arctan\left(\frac{s_{01}}{s_{10}}\right)$ $\quad s_{10} > 0$
   $\arctan\left(\frac{s_{01}}{s_{10}}\right) + \pi$ $\quad s_{01} \geq 0, s_{10} < 0$
   $\arctan\left(\frac{s_{01}}{s_{10}}\right) - \pi$ $\quad s_{01} < 0, s_{10} < 0$
3. Calculate weight according to $\sqrt{s_{10}^2 + s_{01}^2}$
4. Repeat for range of σ
5. For each block in the image sum weights across all positions and all σ for each orientation and normalise
6. Concatenate all blocks in the image to make overall encoding

In the second scheme, oriented gradients are calculated at two scales, the base scale σ_{BASE} and a coarser scale, $r\sigma_{BASE}$, where r is the scale ratio. Then, for each location in the image, an orientation vector is assigned comprising the orientations at each scale, and a weight equal to the product of the weight at each scale. These features are referred to as oriented gradient columns. Histograms of these oriented gradient columns are then calculated across multiple blocks and base scales, and then normalised as before. The scheme is therefore equivalent to our weighted oriented gradient column system with spatial binning and scale averging. The algorithm is described in Algorithm 6.2.

We tested these schemes on both datasets, as described in Experiment 6.3 on page 125.

Algorithm 6.2 The HOG Column encoding

1. For a given σ_{BASE} and scale ratio, r, measure filter responses c_{10} and c_{01} of 1st order derivative-of-Gaussian filters at scales σ_{BASE} and $r\sigma_{BASE}$, and from these calculate the scale normalised filter responses $s_{ij} = \sigma^{i+j} c_{ij}$
2. Calculate quantised orientations, θ_1 and θ_2, at both scales according to:

   $Round(\frac{\arctan(\frac{s_{01}}{s_{10}})}{2\pi})$ $\quad s_{10} > 0$

   $Round(\frac{\arctan(\frac{s_{01}}{s_{10}})+\pi}{2\pi})$ $\quad s_{01} \geq 0, s_{10} < 0$

   $Round(\frac{\arctan(\frac{s_{01}}{s_{10}})-\pi}{2\pi})$ $\quad s_{01} < 0, s_{10} < 0$

3. Compute weight, w_1 and w_2, according to $2\sqrt{s_{10}^2 + s_{01}^2}$ at both scales
4. Combine across scales to form a feature with orientation (θ_1, θ_2) and weight $w_1 w_2$
5. Repeat for range of σ_{BASE}
6. For each block sum weights across all positions and σ_{BASE} for each orientation vector and normalise
7. Concatenate all blocks in the image to make overall encoding

6.7.1 Parameter Sensitivity

As with the standard column methods, we wanted to see how performance varied with the parameter values. The details are given in Experiment 6.4 on page 126.

6.7.2 Comparison of Performance

The performance of the two schemes for each testing regime is given in Table 6.2, with the computational performance for the novel schemes in Table 6.3. This is given alongside the previously published results and the results from the standard column features schemes. Each score is the mean performance over 50 runs for the chars74k dataset and 10 runs for the ICDAR03-CH-5.

From the table it can be seen that the first scheme offers a small improvement over the single scale HOG on the chars74k dataset, but a decrease in performance on the ICDAR03-CH dataset. The second scheme, using oriented gradient columns, shows an improvement in performance on both datasets.

The HOG columns scheme also shows an improvement over the standard column schemes, which implies that the spatial binning step is capturing information useful to

Table 6.2: Comparison of performance for the multiscale HOG schemes

Scheme	Chars74k-5	Chars74k-15	ICDAR03-CH-5
Shape Context [57]	26.1±1.7	34.4	18.3
Geometric Blur [57]	36.9±1.0	47.1	27.8
Multiple Kernel Learning [57]	-	55.3	-
ABBYY [250]	18.7	18.7	21.2
SIFT [57]	-	20.8	-
HOG Features [250]	45.3±1.0	57.5	51.5
Oriented Gradient columns	50.8±1.1	60.2±1.1	46.1±1.3
oBIF columns	53.4±1.4	64.3±1.3	52.7±1.2
HOG multiscale	49.1±1.3	58.8±1.2	48.3±1.2
HOG columns	**57.7±1.1**	**66.5±1.2**	**57.1±0.9**

Table 6.3: The computational performance for each of the schemes

The encoding time is given for each image, along with the classification time for a Nearest Neighbour classifier using 10 images per class. The times are based upon an implementation in Mathematica 7.

Scheme	Size	Computation time (ms)	
		Encoding	Classifier
Oriented Gradient columns	144	72	1.1
oBIF columns	1849	81	1.1
HOG multiscale	1944	510	1.4
HOG columns	7056	530	6.7

the recognitions task. it is interesting to note the shape of graph of performance against box size for the HOG column scheme. Whereas for the standard HOG scheme there is a peak around the optimal box size, with the HOG column scheme we see the peak but we also see a rise in performance again as the box size becomes the same size as the image.

This suggests that there are two effects present. First the gain in performance that comes from capturing as much of the spatial structure of the characters as possible, through the spatial binning step. This should lead to a performance graph with a peak around the optimal box size with decreasing performance either side. The second effect is the gain in performance due to the invariance of using a global histogram. If this was the case, then both effects combined would lead to a trough in performance where the box size is too large to capture useful spatial information but yet still sufficient to remove the invariance aspect. This is what we see in graph (a) of Experiment 6.4 on page 126.

6.8 Summary and Conclusions

In this chapter we have presented an evaluation of the oBIF Column and oriented gradient column schemes using a character recognition problem of current interest, which is the recognition of characters taken from natural images. The results have been presented alongside previously published results of leading schemes, including HOG, SIFT and Shape Context. The results indicate that, on this problem, the oBIF Column scheme outperforms both the oriented gradient column scheme and previously published methods.

We have also presented a novel multiscale HOG scheme, using the column features presented in Chapter 5. This has been evaluated using the same problem, with the results showing that the novel multiscale HOG scheme outperforms all other methods including oBIF Columns. However, the increase in performance comes at the cost of an increase in the size of the encoding.

Despite the leading performance of the two novel schemes, oBIF Columns and HOG Columns, the levels of performance are significantly below perfect recognition. We have suggested that this is partly down to the context-free nature of the problem, meaning that certain pairs of classes are indistinguishable without context. However, we still believe there is significant room for improvement.

In the next chapter we will apply the column system to a texture problem, which involves the discrimination of grain types using their surface texture as revealed through electron microscopy.

Experiment 6.1 Chars74k using Column Features

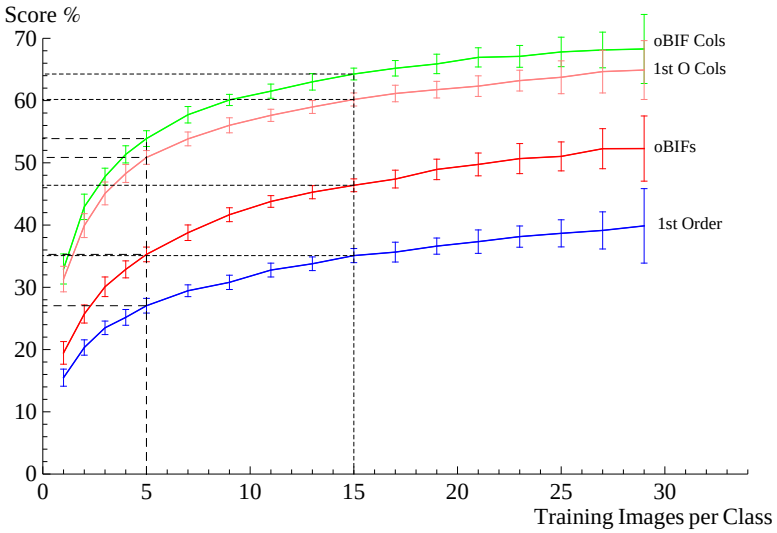
Methods

The parameter values were tuned using the images set aside for tuning, as described previously. Images from the main set were encoded into oriented gradient columns, as outlined in the previous chapter, using the tuned parameter values. A set of training images was randomly selected, which was used to build a Nearest Neighbour classifier using the Bhattacharyya distance. Subsequent training and test sets were drawn, with a total of 50 runs. The mean and standard deviation of the scores are reported.

This was then repeated for the ICDAR2003-CH dataset. We then performed the same procedure for single scale oriented gradients, oBIFs and oBIF columns.

Results

The results for the four schemes are given in graph (a). The performance for oriented gradient columns was 50.8 ± 1.1 and 60.2 ± 1.1 for the chars74k with 5 and 15 training images per class, and 46.1 ± 1.3 for the ICDAR03 set with 5 training images per class. For oBIF columns the corresponding scores were 53.4 ± 1.4, 64.3 ± 1.3 and52.7 ± 1.2.
For the oBIF column scheme, the tuning process gave ϵ as 0.03 and an orientation quantisation of 8, giving 43 oBIF features. The optimal ratio between the scales in the oBIF features was 3. For the single scale oBIF scheme, the tuning process gave an optimal value of ϵ of 0.05 and an optimal orientation quantisation set of 12, giving an oBIF set of 63 features.
For the oriented gradient column and the single scale oriented gradient schemes, a set of 24 orientations was optimal for both. The ϵ values were 0.02 for the single scale oriented gradient scheme and 0.05 for oriented gradient columns and the optimal scale ratio was 2.5.

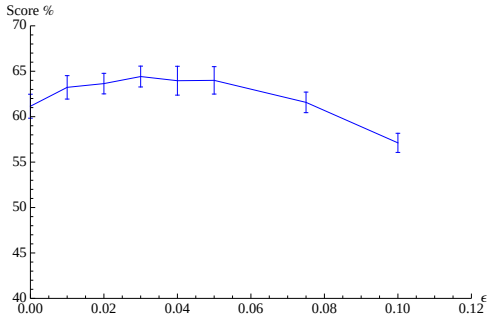


(a) The performance for each of the four schemes

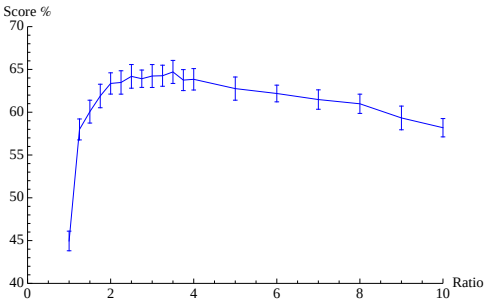
Experiment 6.2 Parameter Investigation of oBIF Columns

First, we looked at the sensitivity of the value of ϵ. From the graph (a), it can be seen that the performance is robust with respect to small changes in ϵ with at least 95% optimum performance being achieved in the range 0 to 0.075, which is 2.5 times the optimal value of ϵ.

We then looked at the influence of the ratio between the two scales. For this we looked at how the level of performance changed as the ratio between the two scales varied between 1 and 10, with results shown in graph (b). As seen from this graph, when the ratio between the scales is 1, the system is equivalent to using single scale oBIFs and we therefore get the same level of performance. There is a sharp increase in performance as the ratio increases to a value of 2, with marginal increases in performance thereafter. The true optimal value of ratio is 3.5, as opposed to the value of 3 obtained from the tuning process. However, performance within 95% of the optimum is achieved with a ratio in the range 1.75 to 7.



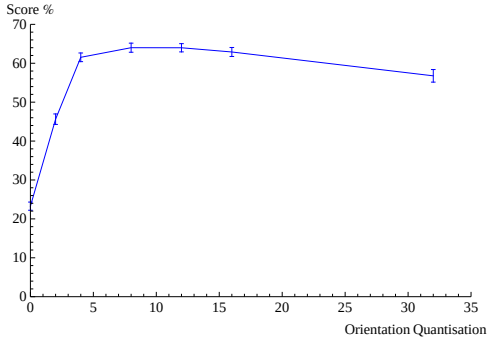
(a) Epsilon (threshold)



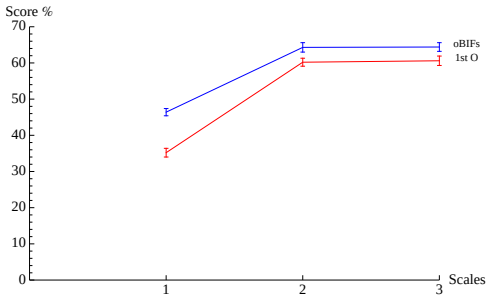
(b) Scale ratio

Next, we looked the importance of the level of orientation quantisation in the oBIF set, shown in graph (c). Here, the point on the far left represents the BIF column system without orientation. We see a rapid increase in performance as orientation is introduced, with a significant increase up to a quantisation level of 8 with a slow decline in performance thereafter.

Finally we wanted to see whether any further increase in performance could be achieved by increasing the complexity of the oBIF column features to triplets of oBIFs across 3 scales. The oBIF features were calculated as before and performance for both oBIFs and first order features are plotted against the single scale and 2 scale schemes in graph (d). This shows a marginal, though not significant, improvement in both schemes. It should also be noted that the 3 scale oBIF columns produce a total encoding size of $(5n+3)^3$ as opposed to the 2 scale oBIF column encoding size of $(5n+3)$, where n is the orientation quantisation.



(c) Orientation quantisation



(d) Number of scales

Experiment 6.3 Evaluating Multiscale HOG with chars74k and ICDAR03-CH

Methods

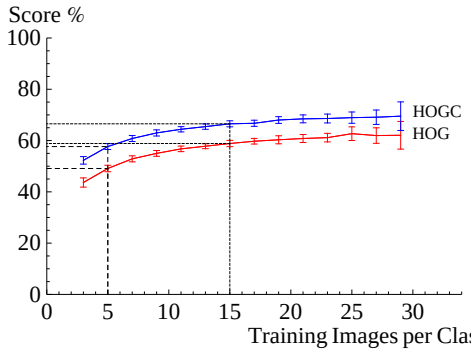
For both schemes, we tuned the parameters using the tuning set of images described previously in this chapter. Images in the main set were then encoded using the multiscale HOG schemes described in Algorithms 6.1and 6.2. As the multiscale HOG schemes involved scale averaging we needed to select a suitable range of scales to use. To do this, we made a visual inspection of the encoded images and observed that at a scale of 7 pixels, images had no recognisable structure. Therefore we selected an arithmetically spaced range of scales between 1 and 7.

Multiple training and test sets were then selected as in Experiment 6.1, with the mean and standard deviation over 50 runs being reported. Classification, as before, was done using a Nearest Neighbour classifier with the Bhattacharyya distance.

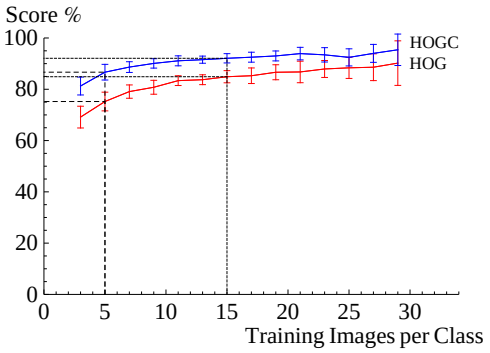
Results

The results for both schemes are given in graph (a), where the scores for the simple multiscale HOG were 49.1±1.3%, 58.8±1.2% and 48.3±1.2 % for the chars74k-05, chars74k-15 and ICDARCH-03-5 respectively. For the HOG Columns scheme the corresponding scores were 57.7±1.1%, 66.5±1.2% and 57.1±0.9%.

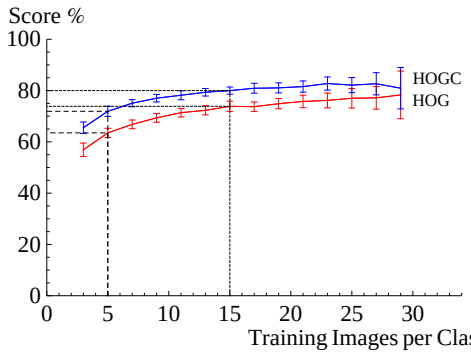
For both schemes, 16 orientations were used and the block size was set to 20 pixels, which was approximately half the object size, with an overlap of 15 pixels between neighbouring blocks. For the second scheme, the tuning process gave an optimal scale ratio of 3.



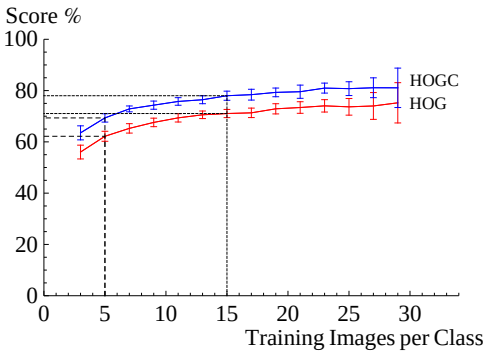
(a) All characters



(b) Digits only



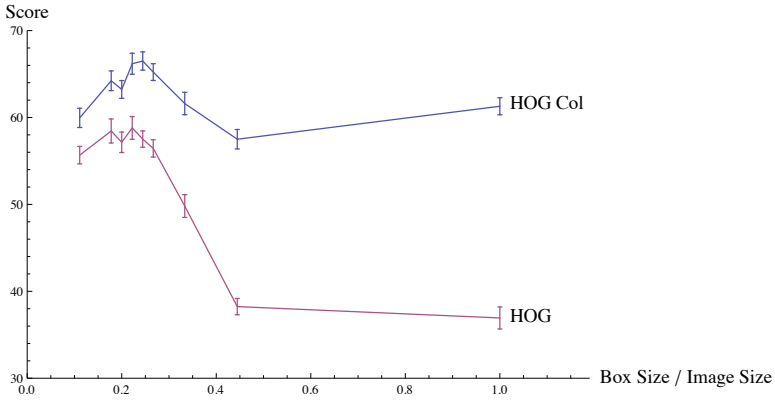
(c) Upper case letters



(d) Lower case letters

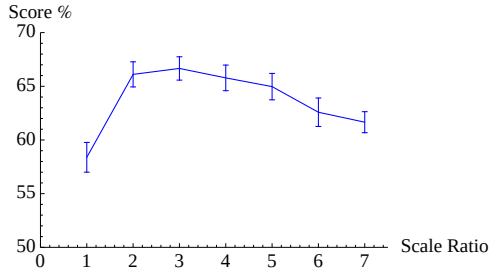
Experiment 6.4 Parameter Investigation for Multiscale HOG

We wanted to see how the performance of both size varied with the block size. To do this, we fixed the overlap between blocks at half the width of the box so we had a single parameter to vary. The performance for both schemes for different block sizes is shown in graph (a).



(a) Block Size

As our better performing scheme, using oriented gradient columns, used the scale ratio parameter that is not found in other implementations of HOG we were keen to see how it affected performance. To do this we used the chars74k-15 test and looked at how performance changed as we varied the scale ratio from 1, which is equivalent to the first of our schemes, up to 7. The results are shown in graph (b). As the graph shows, there is a sharp increase in performance as the ratio increases above 1, with a peak at a scale ratio of 3, followed by a slow drop off.



(b) Scale Ratio

Chapter 7

A Texture Problem: Differentiation of Quartz Grains

After the work on character recognition we wanted to test the column schemes on a recognition problem in a different application area. Given the histogram nature of the schemes, a natural choice for this was texture recognition.

During the course of this work, the BIF Column scheme was evaluated by Crosier et al. [47] and shown to outperform other leading methods with state of the art performance on two of the most commonly used texture datasets, UIUCTex[128], KTH-TIPS[98] and near state-of-the-art on the CUReT dataset[52]. Following on from these results, supported by [46], we wanted to test the BIF Column scheme on a real world problem to test the ability of the system to categorise texture.

For this purpose we selected the discrimination of quartz grains using surface texture as our recognition problem.

7.1 Introduction

Quartz sand grains have potential importance as a trace evidence for forensic investigations. This potential is based on two key features. First, the grains are ubiquitous in the environment and thus their occurrence in sources of evidence is common. Second, they have variable yet distinctive surface characteristics determined by their mode of formation and subsequent erosion, weathering and transportation[31]. The ability to reach exclusionary conclusions based on the provenance of quartz grains in forensic samples with the quartz grains identified in known samples is of value for forensic investigations [155, 31].

The surface characteristics of grains are visible using a Scanning Electron Microscope

(SEM), as shown in the images in Figures 7.2 and 7.3. With expert knowledge it is possible to use features from such images to place a grain within a classification tree [30], that can be used to designate grain types. However, such expert knowledge is rare and the manual identification of grains in this manner is time-intensive. Automatic identification of grains would therefore provide significant advantages in terms of making classification more widely available, as well as time efficiency and offering a standardisation of performance.

The earliest attempt[65] at a mathematical characterization of the physical characteristics of grains used Fourier methods to describe their shape. Since then other authors have proposed further methods based on shape[229, 130, 22], distribution of shape and size[183], and surface texture[256]. However, despite recent advances in texture recognition systems[245, 246, 24, 267, 47] there appear to be very few examples of these being applied to problems in the earth sciences. As far as we are aware, this is the first attempt to bring any of these techniques to grain analysis for applications in forensic analysis.

7.2 Applying column features to texture recognition

Using the results from Crosier et al., we concentrated on evaluating the BIF column scheme. The work using the three artificial datasets had shown that the use of four scales within each column was optimal for texture recognition and that the optimal value for ϵ was 0.[47] We used these same values in this work, which meant that in the texture BIF column scheme the histograms had 6^4(=1296) bins. This is illustrated for a texture image in Figure 7.1.

To compare histograms we used the Bhattacharyya distance[117] as a metric, as in previous work. However, for classification we used a slightly different method. To classify an unseen image, we computed its histogram and use the metric to find the k Nearest Neighbouring (kNN) histograms from all histograms in a training set. If the number of the k having some label is above a threshold, we infer that the unseen image should have the label. Where possible the two parameters, k and the threshold, of our classifiers are determined using a validation set. However, in cases where the number of images per class is very small, these values were set in advance.

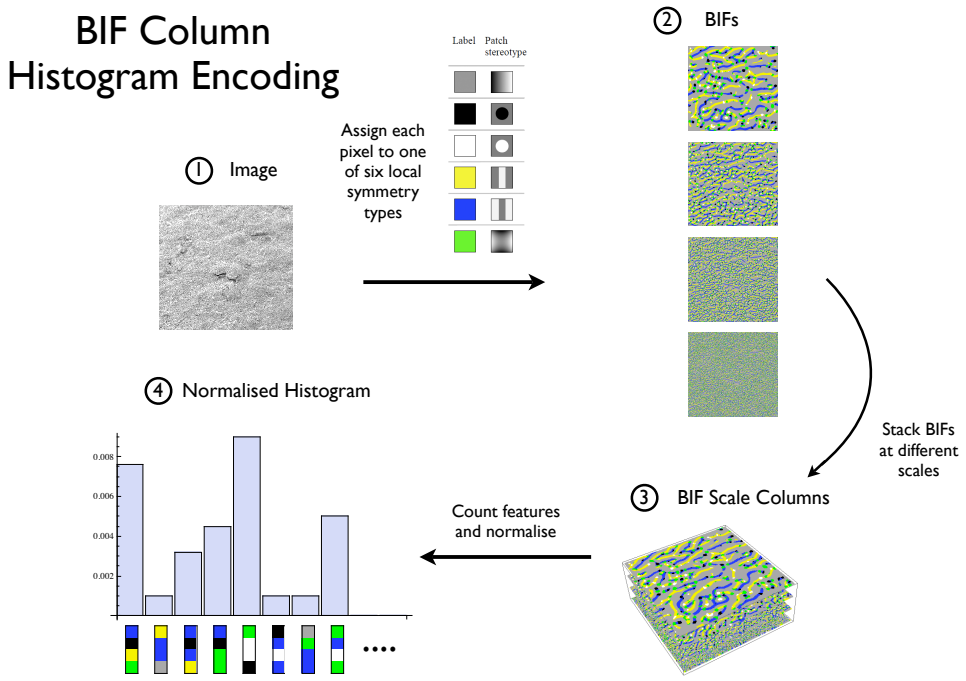


Figure 7.1: The BIF Column scheme applied to quartz grain discrimination. The scheme is similar to the oBIF columns used in the previous chapter. However, two keys differences are the lack of orientation and the use of four scales rather than two.

7.3 Datasets

We chose two problems that are representative of the classification structure in forensic analysis, and assembled a dataset suitable for testing performance on them. The first of these was a two class problem, which involved detecting the presence of a geological feature called *Upturned Plates* [165].

Individual quartz grains were impacted against each other under aeolian conditions of known velocities under controlled laboratory conditions. Forty seven grains were subsequently imaged using scanning electron microscopy (SEM) and 266 images were taken of distinct areas of each grain which exhibited the *Upturned Plates* feature. This set of images was created using expert geological knowledge to ensure that the *Upturned Plates* features were present in each image and is referred to as the UP set. In order to create counterexamples to the UP set, 41 grains were selected by a geological expert from a library of quartz grains which displayed a range of alternative textures. From this set, 237 images were acquired using an SEM, none of which contained *Upturned Plates*. This set is referred to as the NUP set. Examples from UP and NUP are shown in Figure 7.2.

For the second problem the images in the UP set were further divided, by controlling

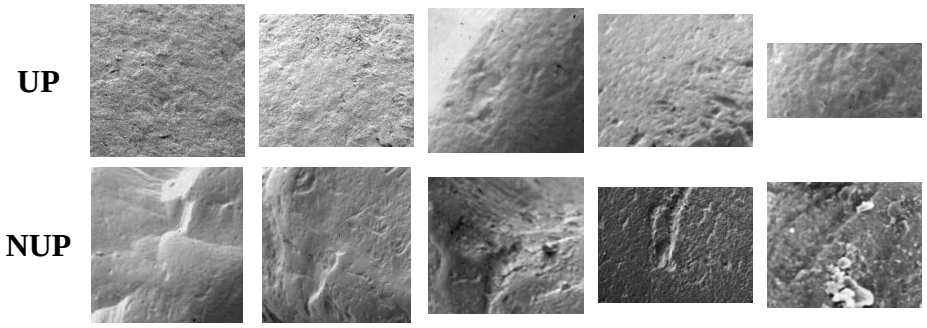


Figure 7.2: Example images from UP and NUP.
In the UP set each image contains examples of the *Upturned Plates* feature whereas these are absent from each image in the NUP set.

the conditions of their formation, into six classes according to the *Energy Level of Formation* (ELF). The 266 images were labelled as 4mps, 8mps, 11mps, 14mps, 17mps or 20mps, which relates to the wind speed to which the grains have been exposed. There was a minimum of 21 images from 6 grains per class. Examples from the different classes are shown in Figure 7.3.

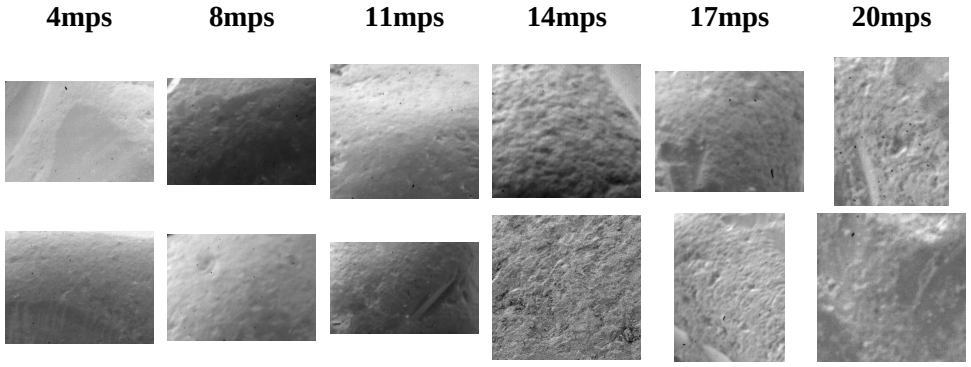


Figure 7.3: Example images from the set used in the *Energy Level of Formation* task.
The different classes correspond to the speed at which the grains have been impacted with each other. Grains impacted at the lower speeds tend to exhibit a smoother surface structure than those at the higher speeds.

Across the dataset, in addition to the textural differences that we sought to characterise, there were grosser types of difference that were due to imaging effects. Examples of global differences included variation in the overall lightness of the image, difference in the apparent focus of the image as well as difference in the image dimensions and overall area. Within images themselves there were variations in the lightness of individual regions, giving the appearance of shadows, although as the images came from the SEM these were not actual optical shadow effects. Each aspect of variation appeared within each class and there was no apparent correlation between each non-textural aspect of variation and the class labels. We expected that the invariance properties of BIFs would

mean that these types of variation would be ignored by our encoding as so not to lessen classification performance.

7.4 Experiments

We first tested the BIF Column scheme on the *Uptruned Plates* problem. The details are given in Experiment 7.1 on page 133.

7.5 Energy Level of Formation

In order to determine the general discrimination power of the method in the ELF task, we first looked at the performance in discriminating each possible pair of classes. The same three schemes for grain classification were used as for *Upturned Plates*: based on a single image; based on pooling the classifications of individual images; and based on classifying the global histogram. The details are given in Experiment 7.2 on page 134.

From these results it was apparent that the system was performing poorly in discriminating between the 14mps, 17mps and 20mps classes, with results across all three scheme being consistent with chance level performance, as indicated in the grey boxes of Table (a). We therefore decided to combine all images from these classes into one new class, labelled 14mps+. We then looked at classification using the revised set of four classes, with details being given in Experiment 7.3 on page 135.

As the number of images available for each class in the ELF task was relatively low, we also wanted to investigate whether we might expect performance to improve if more images were made available. To do this we looked at how performance changed when using a subset of grains from the 47 made available for the ELF task, using the pooling method as, out of the three, this had produced the best results. The details are given in Experiment 7.4 on page 136.

7.6 Summary and Conclusions

In this chapter we have tested the BIF Column scheme on the problem of quartz grain discrimination using surface texture. We have tested three variations of the scheme, each of whcih combines histograms from multiple images in different ways. In the *Upturned Plates* task, where expert human level performance is expected to be perfect, the best performing scheme within the BIF system achieves 98.8%, which equates to classifying

all but one grain correctly.

There is a clear improvement in the performance of the system when multiple images from a single grain are combined in some way. However, in this task there is no apparent advantage of one method of combining images over the other.

The *Energy Level of Formation* task represents a greater challenge. Expert human level performance, although not yet quantified, is expected to be significantly less than perfect, especially in discriminating the grains with higher ELFs. This is reflected in the results, where the BIF Column system is incapable of separating the 14mps, 17mps and 20mps classes from each other.

When these three classes are combined into one, discrimination performance between the four new classes is encouraging with the top performing scheme achieving a rate 81% exact classification and 96% within an error of one class . As before, there is a clear advantage in combining the images from a single grain. In this task our results indicate a slight superiority of pooling image classifications over using the global histogram, but our dataset is not large enough to claim this is significant.

Our results indicate that improved performance could be expected with a larger training dataset, in particular for exact-class classification of grains in the three lower speed ELF classes. For the faster speed classes, and for within-one-class classification a dataset of 47 grains seems sufficient.

From the results as a whole it is evident that the BIF Column system can be used to provide an effective solution to the problem of grain discrimination using surface texture. We note that since the BIF column approach for encoding texture was applied without alteration or tuning from its previously presented formulation, its prospects for successful application to similar problems in forensic analysis are good.

Experiment 7.1 Upturned Plates discrimination using BIF Columns

Methods

The images were encoded as 1296 bin histograms representing the frequency with which each BIF scale column occurred, as described previously. In order to make best use of the relatively small dataset without over-estimating performance by overlapping train and test sets, we used a nested leave-one-out method for both the test set and a validation set. To do this we first selected one grain as a test grain, with all images from this grain being extracted from the set. We then selected a single image from those left to act as a validation set, with images from the same grain being removed and the remainder used to build a *k*NN classifier. The validation image was then classified for different values of *k* and threshold. We then repeated this process for all images, except for those from the test grain, and found the optimal value of k and threshold. Using these optimal parameter values we then classified the images from the test grain. This was then repeated with each image in the dataset in turn acting as the test grain.

As the problem being tackled was correct classification of a *grain*, and we had multiple *images* from each grain, we needed a scheme for making use of the multiple images. We evaluated three schemes for this to gain an understanding of the problem. First, we simply looked at how well the system performed with single images. Second, we used a simple pooling scheme where each image from a grain was classified individually and then a single choice was made for the grain by taking the more common classification across all images for that grain. Finally, we made an estimate of the global histogram encoding for the grain by taking the mean of the individual histograms for the different images from a grain.

Results

Results are shown in Table (a) for all three schemes, where they are quantified as the average of the classification rate for UP grains and for NUP grains. Associated computational performance is given in Table (b). For the three schemes the optimal values for the classifier parameters were determined individually for each different validation set and thus there were no single values that applied to the whole dataset. However, the median value of *k* for the classifier in the first two schemes was 15 and for the global histogram scheme it was 5.

Scheme	Discrimination Performance Score
Without pooling	95.0%
With pooling	98.8%
Global Histogram	98.8%

(a) Performance for the *Upturned Plates* discrimination task

Scheme	Size	Computation time (ms)	
		Encoding	Classifier
Without pooling	1296	1100	1.5
With pooling	1296	1100	1.5
Global Histogram	1296	1100	1.5

(b) Computational performance for the *Upturned Plates* discrimination task

Experiment 7.2 ELF pair discrimination using BIF Columns

Methods

Images were encoded as described previously. As the *Energy Level of Formation* task used only those images from the UP set, which were then divided into six classes, the number of images in each class were far fewer than in the *Upturned Plates* task. As a result, there were not enough images to provide a stable validation process to determine the optimal classifer parameters, so the value of k was set at 3 in advance and the threshold set at the midway point.

Results

The performance for each pair is given in Table (a).

WITHOUT POOLING

	8 mps	11 mps	14 mps	17 mps	20 mps	
4 mps	67 %	79 %	88 %	98 %	99 %	4 mps
8 mps		57 %	79 %	94 %	90 %	8 mps
11 mps			71 %	79 %	77 %	11 mps
14 mps				47 %	46 %	14 mps
17 mps					55 %	17 mps

WITH POOLING

	8 mps	11 mps	14 mps	17 mps	20 mps	
4 mps	83%	87%	86%	100%	100%	4 mps
8 mps		80%	86%	100%	100%	8 mps
11 mps			76%	89%	83%	11 mps
14 mps				41%	71%	14 mps
17 mps					56%	17 mps

GLOBAL HISTOGRAM

	8 mps	11 mps	14 mps	17 mps	20 mps	
4 mps	67 %	73 %	93 %	100 %	100 %	4 mps
8 mps		53 %	93 %	100 %	100 %	8 mps
11 mps			76 %	89 %	83 %	11 mps
14 mps				53 %	59 %	14 mps
17 mps					56 %	17 mps

(a) Performance of the BIF system in the fifteen pairwise problems

Experiment 7.3 ELF discrimination using BIF Columns

Methods

Using the same three schemes again, and a *k*NN classifier with k set at 3, results are reported as the mean performance per class. We report results in terms of exact classification (i.e. the correct one of the four classes is identified), classification to within one class, and classification to within two classes.

Results

The results are shown in Table (a), with the computational performance in Table (b) and the associated confusion matrices given in Table (c).

Scheme	Exact	Within 1 Class	Within 2 Classes
Without pooling	69 %	90 %	100 %
With pooling	81 %	96 %	100 %
Global Histogram	78 %	92 %	96 %

(a) Performance for the Energy Level of Formation task

Scheme	Size	Computation time (ms)	
		Encoding	Classifier
Without pooling	1296	1100	1.5
With pooling	1296	1100	1.5
Global Histogram	1296	1100	1.5

(b) Computational performance for the Energy Level of Formation task

WITHOUT POOLING

		Classified As			
		4mps	8mps	11mps	14mps+
Label	4mps	71 %	0 %	29 %	0 %
	8mps	12 %	55 %	24 %	9 %
	11mps	3 %	12 %	55 %	30 %
	14mps+	1 %	0 %	6 %	94 %

WITH POOLING

		Classified As			
		4mps	8mps	11mps	14mps+
Label	4mps	83 %	0 %	17 %	0 %
	8mps	0 %	67 %	33 %	0 %
	11mps	0 %	0 %	78 %	22 %
	14mps+	0 %	0 %	4 %	96 %

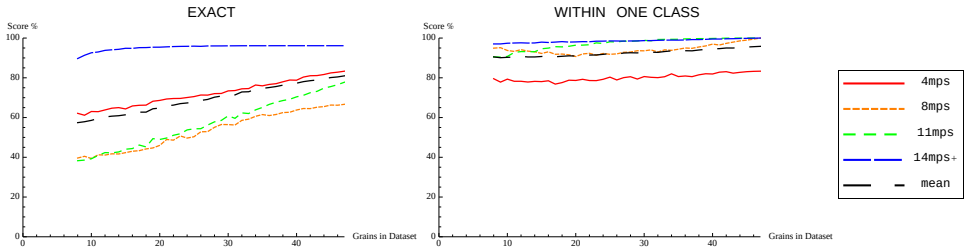
GLOBAL HISTOGRAM

		Classified As			
		4mps	8mps	11mps	14mps+
Label	4mps	81 %	0 %	19 %	0 %
	8mps	11 %	68 %	21 %	0 %
	11mps	0 %	0 %	88 %	12 %
	14mps+	15 %	0 %	0 %	85 %

(c) The confusion matrices for the ELF task

Experiment 7.4 ELF performance for different training set sizes

Two grains were first randomly selected from each class to ensure the minimum size necessary to get results for the classification performance for each class. Then for each increment between this minimum of 8 grains and the full set of 47 grains, further grains were randomly selected and the classification performance determined for both exact and within one class classification. This process was repeated 400 times for each possible number of grains and the mean performance for each class over all trials is plotted in graph (a). The mean performance over all classes is plotted as the black line, showing the overall performance as 81% when all grains are used, as previously shown in Table (a) of Experiment 7.2 on page 134. This line has a steady gradient of 0.5% increase in performance for each grain added.



(a) Performance with varying dataset sizes for each test class

Chapter 8

Writer identification

The third application we tested the column system on was writer identification, where the aim is to establish authorship of a piece of handwritten script given suitable examples from which to train. This is a problem of current interest [97, 141, 75] and one that was particularly interesting in the context of our work as it contained aspects similar to the work on character recognition but also our approach essentially viewed handwriting as a texture, and therefore there were similarities to the work presented in the previous chapter.

The chapter begins with a brief description of the current work on writer identification as well as related problems. We then describe how we attempted to apply the oBIF column method to the problem. Then we present the results from an experiment to test the method against a publicly available dataset, which was used in the 2011 ICDAR Arabic Writer Identification Contest. Finally we give a discussion of how our results compare to other methods and how this could guide future work.

8.1 Related work

Writer identification is not a new problem, with schemes such as run length features [6] being proposed a long time ago. The basis of any solution is that handwriting is individual enough to distinguish authorship [222, 223, 266, 233] and that a suitable quantity of handwriting is available [23] to extract some measure of the style of the handwriting.

As well as the Roman script, such distinctiveness appears in Chinese characters [100, 59], Arabic script [28, 221], Farsi [163], Kannada [118] and even handwritten music scores [77, 76, 75]. This distinctiveness contained within a whole range of handwritten entities has many applications, providing the style can be accurately recognised. Perhaps one of the main applications is in signature verification, where much work has been done (see [174] for a review). Other common applications include the analysis of historical documents [177, 64, 7, 203] and profiling [262, 67].

Writer identification using contemporary handwriting is generally split into offline, which consists of images of handwriting only, and online, where additional stroke path information is available. Our emphasis here is on offline recognition. Offline schemes have generally involved extracting features from samples of handwriting. Many different feature sets have been tried such as Hermite features [110], Lexeme features [16], Chain code features [214], adjacent segments [111], edge based directional features [29] and connected components [202]. Amongst these are multiscale techniques such as [64], who used multiscale Hermite and Gabor features. Alternatively certain schemes have taken a texture-based appraoch is [102].

8.2 Methods

In the work on character recognition, presented in Chapter 6, we showed that the oBIF Column encoding scheme performed relatively well when combined with a Nearest Neighbour classifier. This suggested that in oBIF Column space, examples of the same character are relatively well grouped into clusters. However, it was not possible to tell simply from the results whether different examples of the same character form a single cluster, or multiple smaller clusters.

Providing the distribution of each character can at least be approximated by a single cluster, then we can consider the mean encoding of each character as an approximation of the centre of the cluster. We can then consider what is represented by the vector between the mean of the cluster and the particular instance of the character. If this vector were to encode the style of the character, then we thought this would be useful for determining authorship.

However, if this approach was to be used letter by letter then, when attempting to determine authorship of a block text, we would have to segment each letter, then perform a process of character recognition and finally determine the deviation of each character from the character mean. Given the likely errors in segmentation and recognition steps, the cumulative effect of such a process would likely result in poor performance.

Given the nature of the oBIF Column encoding scheme, we thought it should be possible to avoid the need for segmentation and character recognition. As the encoding is a normalised histogram, the encoding for a block of text is equal to the mean histogram for each of the individual characters, weighted for relative size. If we were to be able to determine the mean histogram for the block of text, then we could use the deviation from that as our style vector.

For example, if we have a section of text A, written by three different authors to give blocks (A_1, A_2, A_3), then we map our samples of text from oBIF column space to the difference space, referred to as Δ space, as shown in Figure 8.1, using the mean encoding for that block of text. When we do the same with another block of text, B, we hope that the corresponding style vectors, $(\Delta B_1, \Delta B_2, \Delta B_3)$, are well grouped in Δ space. If this is the case then we can use a Nearest Neighbour, as in our previous schemes.

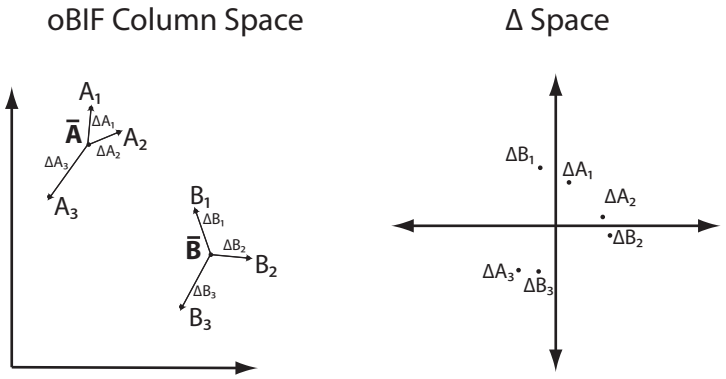


Figure 8.1: oBIF Column space and Δ space.
The oBIF column encoding of an image will be affected by both the text and the handwriting style of the author. In order to be able to compare the style, and become invariant to the underlying text, we need to consider the deviation from the mean encoding for a certain block of text.

Whilst we might expect this style vector to vary for different blocks of text, according to the different occurrences of each character within the block, the variation may be sufficiently small to allow style vectors to be compared across different blocks of text.

In previous schemes we have used the Bhattacharyya distance with the Nearest Neighbour classifier as this is a suitable way for encoding distributions. However, when dealing with vectors in Δ space we are not dealing with distributions and we have to use a distance such as the Euclidean instead. A potential danger with this is that distances become dominated by the most commonly occurring oBIF column types. In an attempt to counter this we used the square root of all oBIF Column histograms.

8.3 Experiments

In order to test the modified oBIF Column method worked for author identification we used the dataset provided for the 2011 ICDAR Arabic Write Identification Competition

[97]. This consisted of handwritten Arabic script from 54 different authors, each of whom had written the same three passages. Examples are shown in Figure 8.2, with a section showing the oBIF encoding in Figure 8.3.

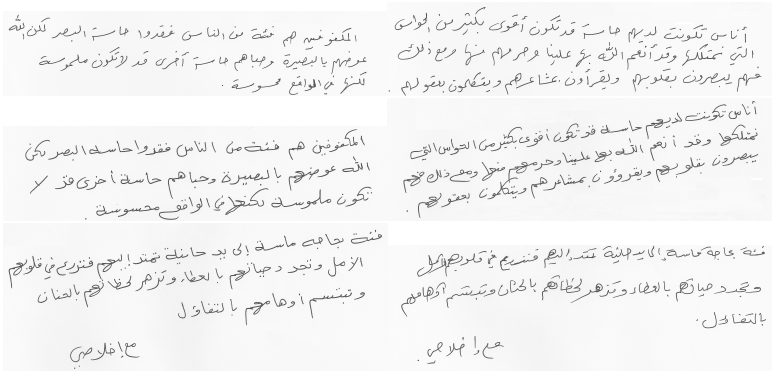


Figure 8.2: Example images from the Arabic handwritten dataset.
The training portion of the dataset consists of the same two paragraphs written by 54 different authors. The testing section consists of a single paragraph written by 54 authors, not all of which featured in the training section.

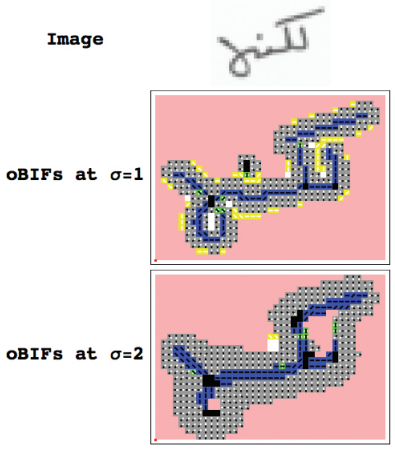


Figure 8.3: Arabic handwriting encoded using oBIFs.
The encoding at the optimal parameter values appears to be free from interaction between words, meaning that arrangement of words within a text should not affect the overall encoding.

As the dataset was provided for the purposes of the competition, the images had been split into training and test sets with only the training labels provided. This meant that the first two passages for each author were labelled, and formed part of the training set. The third passage for each author was unlabelled and these formed the test set. In order to test the ability of systems to detect unknown authors a number of authors had also been removed from the training set, though this number was not provided by the competition organisers.

Scheme	Identification Rate
oBIF Columns	100%
Team Shasta	89.19%
Wride	81.08%
Eu Jin Lok	78.38%
Intelligentia	78.38%
Wifahd	75.68%
Robin	5.45%

Table 8.1: The performance of oBIF Columns against other teams in the ICDAR 2011 Arabic Writer Identification Competition.

In order to be able to deal with unknown authors we devised an adapted nearest neighbour classifier, the details of which are explained in Experiment 8.1 on page 144.

8.4 Discussion

8.4.1 Comparison of Results

Over 30 teams submitted entries for Kaggle[259], though only 7 of these went on to provide a methods description for the ICDAR competition. The performance for each team is given in Table 8.1.

Four of the other teams, (*Intelligentsia, Team Shasta, Robin* and *Eu Jin Lok*) used features provided by the organiser of the competition. These features included 'connected components, number of holes, moments, projections, distributions, position of barycenter, number of branches in the skeleton, Fourier descriptors, tortuosities, directions, curvatures and chain codes' [97]. The difference in the teams' approaches came from different methods of feature selection and classification.

The other two teams used a different set of features, with *Wifahd* using run length codes [60] and *Wride* using edge hinge features and graphemes [243]. Both sets of features have been designed for the purposes of author identification.

8.4.2 Validity of the evaluation procedure

Whilst the evaluation presented shows very encouraging results for the oBIF Column method, as applied to author identification, there are several issues with the competition framework which may have influenced the outcome.

First, the size of the dataset is relatively small. Whilst it is difficult to estimate the

diversity of handwriting styles, it seems at least possible that 54 authors is not enough to adequately capture the full range. This is partly demonstrated by the achievement of a perfect identification rate, which gives us no upper bound on the error of the oBIF Column method. Perhaps more significantly, the number of passages per author is also too low and we are left with little indication as to how style may vary within the writing of a single author.

Second, the entries to the competition are only published in a summary format meaning that it is difficult to establish whether each method has been optimised. This makes it difficult to confirm the superiority of one method over another.

Third, the structure of the dataset, where the same underlying three blocks of text have been used for each author, suited the adapted oBIF Column method well as the mean encoding for each block of test could easily be calculated. In terms of the general problem of writer identification, it may not usually be possible to have examples of the same blocks of text from each author, and the performance of the previous evaluation may only hold for a small subset of the general problem.

8.4.3 Extensions to other work

The method, as presented in this chapter, can be applied to any application of writer identification where the mean encoding can be estimated. For example, it may be expected that the method could be applied to signature verification, where suitable numbers of labelled training data would be available.

However, in general, it may not be possible to estimate the mean encoding for a given block of text in a cost effective manner. In this case it may be necessary to identify sections within the text for which the mean encoding can be estimated. This could consist of common words or even pairs of letters, which have been shown to be effective in writer identification [220]. The difficulty is that this involves the additional steps of recognising the appropriate section of next and then being able to extract it from the rest of the text through a segmentation process. Errors in both these stages may bring down the overall performance. However, such a method would likely produce multiple sections of text for classification, whereas the method presented in this chapter has effectively used just one. If a suitable pooling method is used, using multiple sections may significantly reduce the errors.

8.5 Summary and Conclusions

In this chapter we have tested the oBIF Column scheme on the problem of writer identification. As the oBIF Column scheme has previously been shown, in Chapter 6, to perform effectively at recognising the underlying content of handwriting, we have adapted the scheme for the problem of writer identification. To do this we have introduced the novel step of creating a style vector, which is invariant to the underlying text but varies with the style of the author.

We have tested this scheme on the problem of identifying authorship of handwritten Arabic script, which has been provided as part of the 2011 ICDAR Arabic Writer Identification Competition. In the evaluation, the adapted oBIF Column scheme outperformed all other methods, achieving a recognition rate of 100%.

Whilst these results are very encouraging, we have outlined several limiting factors to the evaluation process and thus present the adapted oBIF Column scheme as a starting point for future work.

In the next chapter we will suggest another extension to the oBIF Column scheme, which arises from the effect of clutter on the scheme.

Experiment 8.1 Arabic handwriting author identification using oBIF columns

Methods

We used the images provided for the 2011 ICDAR Arabic Writer Identification Competition, and supplied through the competition website Kaggle [259]. This consisted of 108 training images and 53 test images. In the training set, it was known that there were exactly two blocks of handwritten Arabic text for each author. The test set was known to contain examples from both the authors featured in the training set as well as an unknown number of other authors.

In order to tune values for the scale, pink threshold, orientation quantisation and scale ratio we divided the training set into two, with one passage from each author in each half. Each image was then encoded into 2 scale oBIF columns with each parameter being varied in turn, using the order scale, pink threshold, scale ration and finally the orientation quantisation. Using the square root of the oBIF Column histograms, the mean was then calculated for each block of text. The style vectors for each author were then calculated by calculating at the deviations from these means. At each stage classification was performed using a Nearest Neighbour classifier, using the Euclidean distance, and the best performing parameter value was selected.

Using these values we then encoded all the images into 2 scale oBIF columns. In order to account for the unknown authors in the test set, we used an adapted Nearest Neighbour classifier. As a first step, the two training images for each author were combined by taking the mean of the two histograms. This effectively considered the two blocks of text as a single block for each author. Then, distances were calculated between each encoded image in the test set and each merged pair of encoded images in the training set.

We then checked to see if there were multiple examples by the same author in the test set. As each image in the test set contained the same block of text, we expected that if any author appeared more than once, the distance between the two occurrences in oBIF Column space would be very short. Therefore, we looked at the distances between the encoded test set to see whether any pair was closer than 3 standard deviations from the mean distance. The results indicated that no pair was so close, and therefore we concluded that no training author appeared in the test set more than once.

Using this knowledge we then assigned each test image to its closest training author. Where training authors were assigned to more than one test author, the pair with the shortest distance were assigned as correct and the others, of which there were three, were classed as unknown. Of the remaining unassigned training authors and test authors, we looked at the next ten nearest neighbours to see if a pair could be matched. Any remaining test authors were then assigned as unknown.

Results

The tuning process gave a base scale of 1, a pink threshold value of 0.05, a scale ratio of 2 and an orientation quantisation of 4 (23 oBIF features), which are very similar to the values used in the other applications involving oBIF columns. This gives an encoding size of 529, an encoding time of 0.51s per image and a comparison time of 0.016ms per pair of encoded images.

The adapted classifier labelled 51 of the 53 test images as having come from an author in the training set and the remaining 2 images were labelled as unknown authors. When compared to the correct labels provided through the Kaggle website, which had been unavailable until submission of the final method, the identification rate was 100% correct.

Chapter 9

Extending the Column Scheme

The results from the three application areas demonstrate that the column schemes can perform well compared to other methods. However, the performance on the character recognition tasks presented in Chapter 6, shows that there is still considerable room for improvement. In order to increase the performance of the column features schemes we used the cluttered MNIST set to investigate a means of extending the scheme.

9.1 The Effect of Clutter

Various measure of clutter exist. Feature Congestion [192] uses the notion that salient features tend to be unusual features in the local context. If these features are easier to detect as unusual then the task is simpler, and thus the level of clutter is deemed to be lower. As the local variability of features increases, it becomes more difficult to detect salient features and thus the level of clutter is deemed to be higher.

Subband Entropy [193] offers a measure of clutter in terms of the number of bits required to encode regions of an image whilst preserving perceptual image quality. As, in the general case, it is difficult to establish exactly how efficiently complex structure can be encoded, the measure relies on a simplified image coding using steerable pyramids [216]. Typically, both luminance and chrominance are encoded and thus regions with greater colour variation will result in a higher measure of clutter.

An alternative measure, Edge Density, calculates the level of clutter as a proportion of edge pixels within a region of an image. This is different from Feature Congestion and Subband Entropy in that regular structure could still result in a relatively high measure of clutter. For example, grid-like structure may result in a high proportion of edge pixels, yet these edge pixels have a regular pattern and thus can be encoded efficiently.

The chars74k set appeared, from visual inspection, to contain a relatively high level

of clutter. In order to assess the performance of the oBIF column system in the presence of clutter it may be possible to measure the clutter in each of the chars74k images and then calculate how well it is correlated with performance. However, as the chars74k set contains relatively few images per class, and the levels of clutter vary between classes, we did not think it would provide a good guide to the relationship between performance and clutter. Instead we decided to use the MNIST set to create a set of images with a controllable amount of clutter. The details are given in Experiment 9.1 on page 150.

As can be seen from the results, the oBIF Column system appears to be sensitive to even small levels of clutter. It should be noted that, in our experiments, clutter may not be used in the normal way. Schemes such as SIFT and HOG would generally look to handle clutter at the level of matching descriptors whereas we are looking to include clutter within the descriptor itself without any attempt at segmentation.

9.2 A Spatialisation Scheme

In order to overcome this problem, whilst staying within the histogram framework, we needed to find features in the digits that are rare enough to make it very unlikely that they appear in the clutter. One possible approach is to extend the scale column features, increasing the number of individual oBIFs in each column, increasing the number of histogram bins from 23^2 to 23^3 or 23^4. However, as was shown in Chapter 5, such features did not significantly improve performance in the standard MNIST recognition task, possibly because they capture very little information not already included in the two scale system.

An alternative to extending the features through scale is to look at the spatial relations of features within one scale. For example, if we could measure the approximate distance between pairs of oBIF scale columns, this could provide a feature that is both translationally and rotationally invariant.

Calculating these distances explicitly is computationally intensive and so instead we can employ a system that encodes spatial information is a less intensive manner. For this purpose we begin by considering local histograms of a certain box size. Each local histogram within the image captures the features that occur within the same box and if we could capture all these co-occurrences across then this would include the spatial information desired. Simply summing the local histograms across the image gives us the global histogram, which gives us the same method as applied in Experiment 5.4 on page 103. However, if we take the outer product of each local histogram, giving us a new encoding of dimensionality 23^4, and sum this across the image the individual spatial

relations are preserved.

The new method consists of the following steps:

1. Calculate oBIF 2 scale columns for the image, as in the previous method
2. Select box size, d
3. For each dxd window within the image calculate the local histogram of 2 scale columns
4. For each local histogram take the outer product of the histogram with itself
5. Sum all outer products across the image, giving an encoding scheme of 23^4 dimensions

In the case when the box size is 1, the encoding is exactly the same as the global histogram, and therefore performance should be equivalent to that in Experiment 5.4 on page 103. In the case that the box size is the size of the image, the encoding again simply catches the information of the global histogram and the performance should be the same as before. In between these two extremes, the encoding captures not only the information in the global histogram but also information relating to the local spatial relations.

9.3 Performance with MNIST

In order to see whether the extra information in the new encoding scheme was beneficial or not we wanted to see how recognition performance changed as the size of the box varied between the two extremes.

As our goal was to test the ability of the system to recognise a previously learnt object in the presence of clutter, we adopted a slightly different experimental procedure. Rather than using one set of images from which to draw subsequent training and test sets, we used the standard MNIST images as a training set and the cluttered MNIST set as a test set.

We also wanted to explore the effect of clutter separately from the problem of categorisation so we performed two experiments. In the first, the same underlying digits were used for the training and test sets, which would guarantee a perfect score in the absence of clutter. In the second experiment we used different underlying digits in the training and test sets. The details are given in Experiment 9.2 on page 151.

We can see from the results that, when the same digits are used in the train and test sets, the new system shows a high level of tolerance to clutter within the descriptor. When the clutter border size is 5 pixels, corresponding to a clutter area to object area ratio of

approximately 1:1, the system achieves a near perfect recognition rate with a box size of 15 pixels. This shows that when there is a very good model of the object, we can detect its present robustly in the presence of clutter.

When we use novel digits in the test images, so that the system has to generalise as well, performance with a clutter border of 5 pixels reaches the same level as the standard system working on clean digits, when using a box size of 15 pixels.

9.4 A General System of Features

The results from Experiment 9.2 on page 151 show that recognition performance is improved by using the new spatial relations, and thus there is useful information in this encoding. However, the outer product method means that we have to calculate all 23^4 values of the encoding. Instead, we can consider another implementation of the same method that allows to calculate only certain bin values.

Starting with the image, the first stage, as before, is to calculate oBIFs at two different scales. For each of the 23 oBIFs and for each scale this produces a map, of the same size as the original image, giving the positions where that particular oBIF type is found. Each map is binary valued and there are 56 maps in total.

Then for each possible pair of maps, where one is taken from each scale, the product of these produces a scale column map, giving the locations at which a particular 2 scale oBIF column is found. There are 23^2 of these new maps, and the sum of each across the image would give the global histogram as used in Experiment 9.2 on page 151.

However, rather than summing up each map we first convolve them with a box function, the size of which is equivalent to the size of the box used for the local histogram in the outer product method. Then the inner product of any pair of these 23^2 blurred maps gives one value of the encoding in the outer product method, of which there are 23^4.

This is shown for a single feature in Figure 9.1.

9.5 Conclusions

In this chapter we have considered the effect of clutter on the oBIF Column scheme. Using our simple model of clutter, consisting of small sections of digits taken from the MNIST set, we have demonstrated that performance of the oBIF Column scheme can be substantially reduced.

We have suggested that this is caused by the occurrence of individual oBIF Column

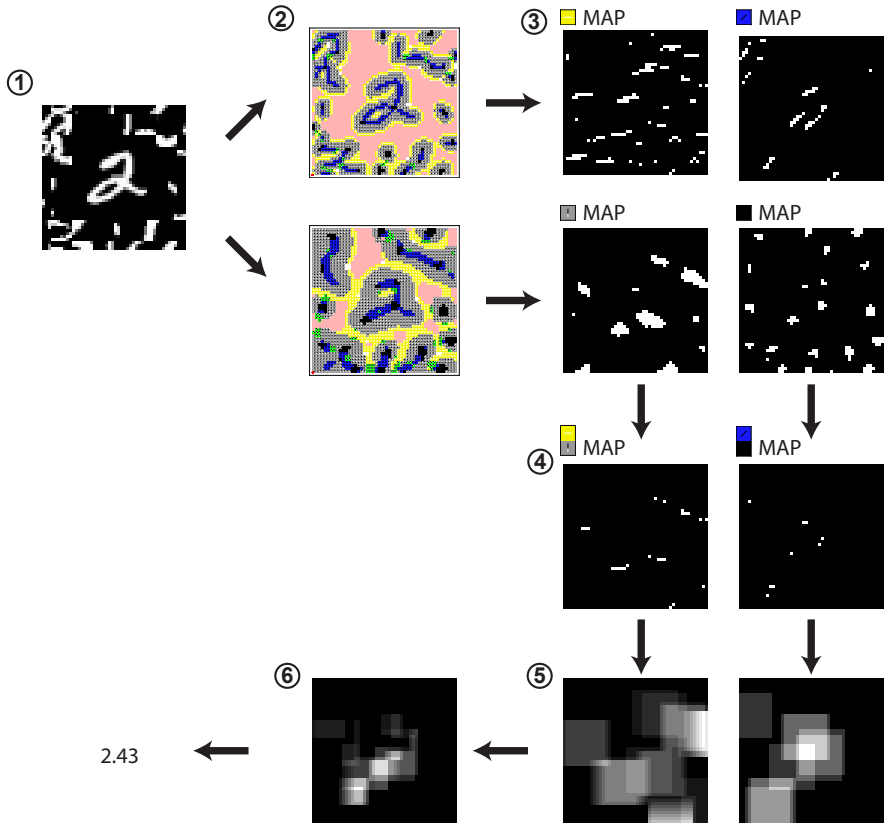


Figure 9.1: An example feature for spatialised oBIF columns.
In this scheme, an image is first encoded into maps of each oBIF type, giving the locations at which oBIF appears. Pairs of maps at two different scales are then multiplied together element-wise to give the map for a particular oBIF column map. Finally, the inner product of blurred oBIF column maps is calculated, to give a measure of the proximity of oBIF column features to each other within the image.

features in both the target digit and in the clutter. In order to overcome this problem we have proposed extending the oBIF Column scheme to include new features which consist of pairs of oBIF Columns occurring within a certain range. When this scheme is tested using the same levels of clutter, we have demonstrated that performance is far more resilient.

The improved performance of the proposed scheme, however, comes at a cost of an increased encoding size. We therefore suggest the new scheme as a starting point for future work, which would aim to find a more compact version of the new scheme.

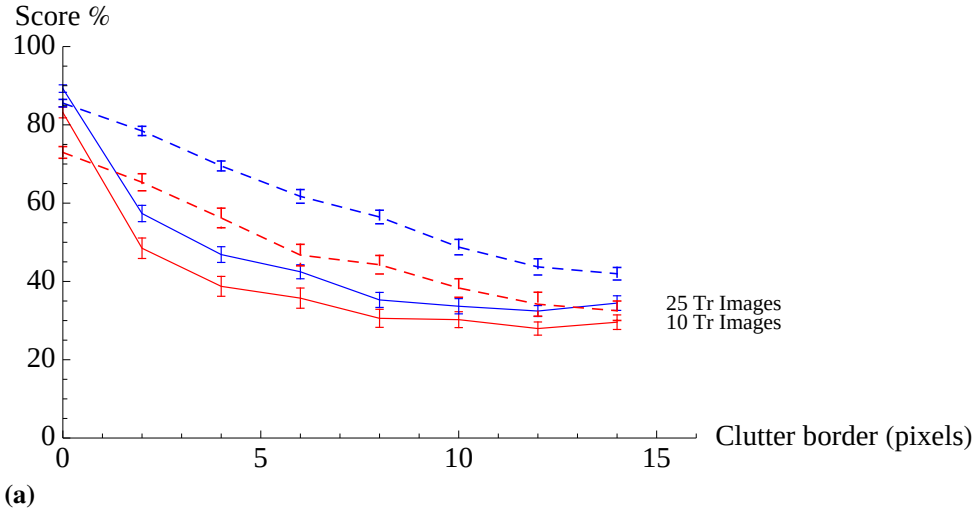
Experiment 9.1 oBIF Columns with clutter

Methods

We limited out investigation to oBIF Columns, as this was the best performing scheme when using the chars74k dataset. Using the parameter values from Experiment 5.4 on page 103, images from the cluttered MNIST set were encoded into oBIF Columns. Multiple training and test sets were then selected, as in Experiment 5.4, to produce the mean and standard deviation of scores over 50 runs. Classification was, as before, done using a Nearest Neighbour classifier with the Bhattacharyya distance.

Results

The results for different sizes of clutter border are shown in graph (a). The two lines are for 10 (red) and 25 (blue) training images per class. The dashed lines give the benchmark performance from Chapter 3.



(a)

Experiment 9.2 Spatialised oBIF Columns 1

Methods

We used the MNIST and cluttered MNIST sets of images. To form the training set, we randomly selected 10 images per class from the MNIST set. We then created two sets of test images. For the first of these, we selected the images from the cluttered MNIST set that contained the same digits as those in the training set. For the second test set we used the remaining images from the cluttered MNIST set.

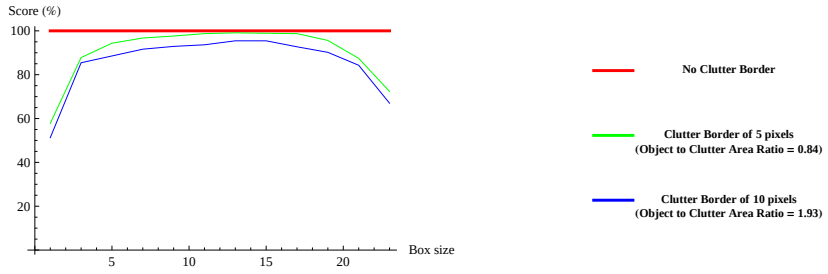
In order to limit the overall encoding size, we set the orientation quantisation to give an underlying set of 23 oBIF features. The scale was set at 1, the scale ratio at 2 and the threshold at 0.05.

For both training and test sets we then selected a particular box size, which corresponds to the size of the window used to calculate the local histogram, and determined all possible local histograms across the image. Each local histogram had 23^2 bins. We then took the outer product of each of these local histograms with themselves and summed the results across the image giving a global histogram with 23^4 bins.

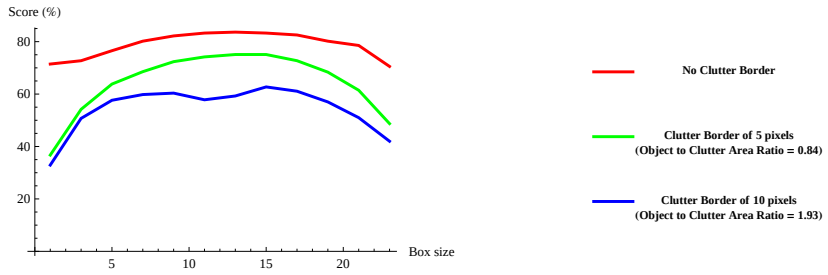
Classification was then performed with a Nearest Neighbour classifier using the Bhattacharyya distance.

Results

The results for the first test set, using the same underlying digits as the training set, are given i graph (a). The results for the second test, using novel digits, are given in graph (b). For both schemes the encoding size was 279 841, with an encoding time of 21s per image and a classification time of 8ms per pair of encoded images.



(a) Performance of the Outer Product Histogram method on the cluttered digits dataset using the same training and test digits



(b) Performance of the Outer Product Histogram method on the cluttered digits dataset using novel test digits

Chapter 10

Conclusions

In the final chapter of this work we summarise the work that has been presented and assess how far it has gone to meeting its aims. We then provide a critical account of the process that has been employed both in the investigation and the evaluation of the schemes.

10.1 Summary of the work

In this work we have attempted to develop new encoding schemes which are of use in visual recognition tasks. To do this, we have performed an investigation, guided by the ideas contained in several methods which have shown a generic usefulness over a period of time. One observation that was gathered from these methods was that they employed a grid, or template, at some stage and our investigation was guided by the search for an encoding scheme that could achieve good performance without this template aspect.

Our investigation, which used the MNIST dataset to estimate performance, resulted in looking to the scale dimension to create a new set of features which could fit within a histogram framework. Following from the observation that simple features, such as oriented gradients and BIFs, can change over scale at locations within the image, we developed a set of new features called column features. These features came in three forms depending on the basic feature used, oriented gradient columns, BIF columns and oBIF columns.

In our investigation we considered various different forms of these schemes, including weighted schemes, scale averaged schemes and rotationally invariant schemes, each of which may be suitable for different applications where the need for invariance arises.

Using the column feature schemes, we then evaluated our system on three different application areas. First, we looked at recognition of characters taken from natural images. In this we tested both oriented gradient columns and oBIF columns, with oBIF Columns showing superior performance to other methods, including SIFT, HOG and

Shape Context. We have also presented an extension of the HOG scheme using column features, which performs favourably.

Next we tested the scheme on a texture problem. This was the discrimination of quartz grains using surface texture, a problem which is thought to be challenging even for a human expert. In this problem, the BIF column system performed well, achieving near perfect performance in one task and a level of performance comparable to expert human in the other.

Finally we tested the system on writer identification, using oBIF columns to tackle the identification of authorship of Arabic handwriting as part of the ICDAR 2011 competitions. On this problem, the system achieved perfect performance, beating all other entries.

In addition to the basic column feature schemes, we have suggested a way in which performance may be improved in the presence of clutter within the descriptor, by including information on the spatial arrangement of column features. This has been suggested as a direction for future research.

10.2 The contributions of this work

The major contributions of this work have been:

- **The oBIF Column scheme**

  The oBIF Column scheme is a novel encoding scheme that combines local orientation and symmetry type information across different scales. The features are fundamentally different from other schemes that use local orientation, such as SIFT and HOG. Whereas in these other schemes, local orientation information is encoded in the form of a template of histograms, oBIF column features encode conjunctions of features at each location across scale. This means that the features can be used with a simple histogram, without the need for template.

  The oBIF Column scheme differs from previous multiscale representations. The most common of these, pyramid representations, uses a fixed structure to encode an image. In contrast to the histogram encoding used with column features, this fixed structure limits the invariance properties of the pyramid representations. In comparison to more recent schemes, such as the texture representation of Varma et al. [245], the oBIF Column scheme has two key differences. First, it encodes local orientation information, which is the basis of many recent object recognition schemes. Second, the column scheme does not require any feature quantisation

stage. It is proposed that these two factors make the scheme more applicable to a wider range of recognition tasks. In this work we have demonstrated its applicability to three application areas.

- **A novel multiscale HOG scheme**

  We have proposed a novel version of the HOG encoding scheme. Whereas the traditional HOG scheme encodes oriented gradients at a single scheme, the proposed version combines oriented gradients at different scales within a single descriptor. However, as we shave shown in chapter 6, the oriented gradient features have to be combined at each location in the image, rather than simply combining two templates of histograms. Thus, the essential aspect of the novel scheme is the way in which oriented gradient features are combined, as we have demonstrated that a simple multiscale extension offers no performance gain on the application tested.

- **A novel way of applying texture encoding to writer identification**

  In applying the oBIF Column scheme to the problem of writer identification we have proposed a novel way of using texture to determine the authorship of handwritten text. Whereas in the problem of character recognition, the oBIF Column scheme has been used as a simple histogram of features, for writer identification we have used it to produce a style vector which encodes the deviation of the text from the mean encoding.

- **Discrimination of quartz grain types using surface texture**

  We have applied the column scheme to the problem of quartz grain discrimination. This problem has not previously been investigated using modern texture recognition methods and thus the application represents new work. In our approach we have considered the problem of how to combine multiple images that arise from a single grain. For this we have considered three different approaches and concluded that the best performance can be obtained when images are first classified individually and then classifications are pooled for the grain.

- **An evaluation of spatial binning schemes with multiple features**

  We have assessed the performance of single scale oBIFs both in the context of simple histograms and templates of histograms. For this we have used two datasets, and demonstrated that, in the context of simple histograms, oBIFs outperform oriented gradients. This has demonstrated that local symmetry information is of

use, in addition to local orientation information. However, when used with a template of histograms, we have found that oBIFs do not outperform oriented gradients.

10.3 Comments on the research process

This work has concentrated on producing novel encoding schemes that combine structure at different scales. The driving factor behind this was the view that, by devising such features, we could remove the need for a template, as found in features such as SIFT and HOG. Whilst we have demonstrated improved performance in the application area of character recognition, we feel it may have been advantageous to have selected slightly different application areas, where the use of a template could be shown to reduce performance in a more obvious manner.

It would also have been advantageous to select applications where a direct comparison could be made between the methods presented in Chapter 2 and the novel encoding schemes produced throughout this work. Many of these methods are not applicable to the range of application areas we have used. For example, SIFT and HOG are not generally proposed as useful for texture recognition. However, our aim was to demonstrate the usefulness of the novel schemes across a broad range of application areas.

In terms of the evaluation of BIFs and oBIFs, a more productive approach may have been to develop versions of existing schemes, such as SIFT and HOG, using BIFs and oBIFs, rather than attempting to combine the evaluation with the development of a novel scheme. This would have enabled a direct comparison to be made with a large body of published results.

However, despite the flaws in the evaluative process we do think that the column feature encoding schemes make a contribution to understanding the computational process of visual recognition and, with adequate further research, we think they will demonstrate wider usefulness.

Bibliography

[1] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden. 1984, Pyramid methods in image processing. *RCA Engineer*, 29(6):33–41, 1984.

[2] E. H. Adelson and J. R. Bergen. The plenoptic function and the elements of early vision. In *Computational Models of Visual Processing*, pages 3–20, 1991.

[3] E. H. Adelson, E. P. Simoncelli, and W. T. Freeman. Pyramids and multiscale representations. In *Proc European Conf on Visual Perception*, pages 3–16, Paris, August 1990.

[4] Y. Amit and M. Mascaro. An integrated network for invariant visual detection and recognition. *Vision Research*, 43:2073–2088, 2003.

[5] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1014 – 1021, 2009.

[6] B. Arazi. Handwriting identification by means of run-length measurements. *IEEE Trans. Syst., Man and Cybernetics*, 7:878–881, 1977.

[7] G.R. Ball, S.N. Srihari, and R. Stritmatter. Writer verification of historical documents among cohort writers. In *Proceedings of the International Conference on Frontiers in Handwriting Recognition (ICFHR) 2010*, pages 314–319, 2010.

[8] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110:346–359, June 2008.

[9] J. Beck. Similarity grouping and peripheral discriminability under uncertainty. *American Journal of Psychology*, 85:1–19, 1972.

[10] J. Beck. Textural segmentation, second-order statistics and textural elements. *Biological Cybernetics*, 48:125–130, 1983.

[11] J. S. Beis and D. G. Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1000–1012, 1997.

[12] S. Belongie, J. Malik, and J. Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *Advances in Neural Information Processing Systems (NIPS)*, pages 831–837, 2000.

[13] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, April 2002.

[14] A. C. Berg, T. L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondence. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 26–33, 2005.

[15] F. Bergholm. Edge focusing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(6):726–741, June 1987.

[16] A. Bhardwaj, A. Singh, H. Srinivasan, and S. Srihari. On the use of lexeme features for writer verification. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR) 2007*, pages 1088–1092, 2007.

[17] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–147, 1987.

[18] I. Biederman and P. C. Gerhardstein. Recognizing depth-rotated objects: evidence and conditions for three-dimensional viewpoint invariance. *J Exp Psychol Hum Percept Perform*, 19(6):1162–1182, December 1993.

[19] S.M. Bileschi. A multi-scale generalization of the HoG and HMAX image descriptors for object detection. In *CSAIL Technical Report MIT-CSAIL-TR-2008-019*, 2008.

[20] Y-L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2559–2566, 2010.

[21] A.C. Bovik, M. Clarke, and W.S. Geisler. Multichannel Texture Analysis Using Localized Spatial Filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:55–73, 1990.

[22] E. T. Bowman, K. Soga, and T. Drummond. Particle shape characterization using Fourier analysis. *Geotechnique*, 51:545–554, 2001.

[23] A.A. Brink, M. Bulacu, and L.R.B. Schomaker. How much handwritten text is needed for text-independent writer verification and identification. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 1–4, 2008.

[24] R. E. Broadhurst. Statistical estimation of histogram variation for texture classification. *Proceedings of the Fourth International Workshop on Texture Analysis and Synthesis. Beijing, China.*, pages 25–30, 2005.

[25] P. Brodatz. *Textures: A Photographic Album for Artists and Designers*. Dover Publications, August 1999.

[26] M. Brown and D. Lowe. Invariant features from interest point groups. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 656–665, 2002.

[27] R. Brunelli and T. Poggio. Face recognition: features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1042–1052, October 1993.

[28] M. Bulacu, L. Schomaker, and A. Brink. Text-Independent Writer Identification and Verification on Offline Arabic Handwriting. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR) 2011*, volume 2, pages 769–773, 2007.

[29] M. Bulacu, L. Schomaker, and L. Vuurpijl. Writer identification using edge-based directional features. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR) 2003*, pages 937–941, 2003.

[30] P. A. Bull and R. M. Morgan. Sediment fingerprints: A forensic technique using quartz sand grains. *Science & Justice*, 46(2):107 – 124, 2006.

[31] P. A. Bull, A. G. Parker, and R. M. Morgan. The forensic analysis of soils and sediment taken from the cast of a footprint. *Forensic Science International*, 162:6–12, 2006.

[32] H. H. Bulthoff and S. Edelman. Psychophysical Support For A 2-Dimensional View Interpolation Theory Of Object Recognition. *Proceedings Of The National Academy Of Sciences Of The United States Of America*, 89(1):60–64, January 1992.

[33] M.C. Burl and P. Perona. Recognition of planar object classes. pages 223–230, 1996.

[34] J. B. Burns, R. S. Weiss, and E. M. Riseman. Geometric invariance in computer vision. chapter The non-existence of general-case view-invariants, pages 120–131. MIT Press, Cambridge, MA, USA, 1992.

[35] J. B. Burns, R. S. Weiss, and E. M. Riseman. View variation of point-set and line-segment features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(1):51–68, 1993.

[36] P. J. Burt and E. H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31:532–540, 1983.

[37] P.J. Burt. Fast filter transforms for image processing. *Computer Graphics and Image Processing*, 16(1):20–51, May 1981.

[38] J. Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, November 1986.

[39] V. Chandrasekhar, G. Takacs, Y. Reznik, R. Grzeszczuk, and B. Girod. Compressed histogram of gradients: A low-bitrate descriptor. *International Journal of Computer Vision*, 05/2011 2011.

[40] P. Chang and J. Krumm. Object recognition with color co-occurrence histograms. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages II: 498–504, 1999.

[41] B. B. Chaudhuri, N. Sarkar, and P. Kundu. Improved Fractal Geometry Based Texture Segmentation Technique. *IEE Proceedings part E*, 140:233–241, 1993.

[42] D.T. Chen, J.M. Odobez, and H. Bourlard. Text detection and recognition in images and video frames. *Pattern Recognition*, 37(3):595–608, March 2004.

[43] X. Chen and A. L. Yuille. Detecting and reading text in natural scenes. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 366–373, 2004.

[44] P. Clark and M. Mirmehdi. Recognising text in real scenes. *International Journal on Document Analysis and Recognition*, 4(4):243–257, 2002.

[45] P. Clark and M. Mirmehdi. Rectifying perspective views of text in 3D scenes using vanishing points. *Pattern Recognition*, 36(11):2673–2686, 2003.

[46] A. D. F. Clarke, F. Halley, A. J. Newell, L. D. Griffin, and M. J. Chantler. Perceptual similarity: A texture challenge. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 120.1–120.0, 2011.

[47] M. Crosier and L. Griffin. Using basic image features for texture classification. *International Journal of Computer Vision*, 88(3):447–460, January 2010.

[48] G. R. Cross and A. K. Jain. Markov Random Field Texture Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5:25–39, 1983.

[49] J. L. Crowley and A. C. Parker. A Representation for Shape Based on Peaks and Ridges in the Difference of Low-Pass Transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(2):156–170, March 1984.

[50] J. L. Crowley, O. Riff, and J. H. Piater. Fast computation of characteristic scale using a half-octave pyramid. In *Proceedings of the 4th International Conference on Scale-Space theories in Computer Vision*, 2002.

[51] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Proceedings of the European Conference on Computer Vision Workshop*, pages 1–22, 2004.

[52] O. G. Cula and K. J. Dana. Compact representation of bidirectional texture functions. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1041–1047, 2001.

[53] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893, 2005.

[54] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *Proceedings of the European Conference on Computer Vision*, pages 428–441. Springer Berlin / Heidelberg, 2006.

[55] K. J. Dana, B. van Ginneken, S. K. Nayar, and J. J. Koenderink. Reflectance and texture of real-world surfaces. *ACM Trans. Graph.*, 18(1):1–34, January 1999.

[56] J. G. Daugman. Two-dimensional spectral analysis of cortical receptive field profiles. *Vision Research*, 20(10):847–856, 1980.

[57] T. E. de Campos, B. R. Babu, and M. Varma. Character recognition in natural images. In *Proceedings of the International Conference on Computer Vision Theory and Applications, Lisbon, Portugal*, 2009.

[58] G. Deco and E. T. Rolls. A neurodynamical cortical model of visual attention and invariant object recognition. *Vision Res*, 44(6):621–642, March 2004.

[59] W. Deng, Q. Chen, Y. Yan, and C. Wan. Off- line chinese writer identification based on character-level decision combination. *International Symposiums on Information Processing*, 0:762–765, 2008.

[60] C. Djeddi and L. Souici-Meslati. A texture based approach for arabic writer identification and verification. In *In Proc of International Conference on Machine and Web Intelligence (ICMWI)*, pages 115–120, 2010.

[61] R. O. Duda and P. E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, 1972.

[62] S. Edelman. Representation is representation of similarities. Technical report. Weizmann Science Press of Israel Jerusalem, Israel, Israel, 1996.

[63] S. Edelman and D. Weinshall. A self-organizing multiple-view representation of 3D objects. *Biological Cybernetics*, pages 209–219, 1991.

[64] V. Eglin, S. Bres, and C.J. Rivero Moreno. Hermite and gabor transforms for noise reduction and handwriting classification in ancient manuscripts. *International Journal on Document Analysis and Recognition*, 9(2-4):101–122, April 2007.

[65] R. Ehrlich and B. Weinberg. An exact method for characterization of sand shape. *Journal of Sedimentary Petrology 40*, 40:205–212, 1970.

[66] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results.

[67] M.C. Fairhurst and M.C.D. Abreu. An investigation of predictive profiling from handwritten signature data. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR) 2011*, pages 1305–1309, 2009.

[68] X. Fan and G.L. Fan. Graphical models for joint segmentation and recognition of license plate characters. *Signal processing Letters*, 16(1):10–13, January 2009.

[69] A. Fathi and G. Mori. Action recognition by learning mid-level motion features. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[70] L. Fei-Fei, R. Fergus, and P. Perona. A Bayesian approach to unsupervised one-shot learning of object categories. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 1134–1141, 2003.

[71] 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. In *Conference on Computer Vision and Pattern Recognition Workshop*, page 178, 2004.

[72] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 264–271, 2003.

[73] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Trans. Comput.*, 22:67–92, January 1973.

[74] L. Florack. A geometric model for cortical magnification. In *Proceedings of the First IEEE International Workshop on Biologically Motivated Computer Vision*, pages 574–583, 2000.

[75] A. Fornes, A. Dutta, A. Gordo, and J. Llados. The ICDAR 2011 Music Scores Competition: Staff Removal and Writer Identification. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR) 2011*, pages 1511–1515, 2011.

[76] A. Fornes and J. Llados. A symbol-dependent writer identification approach in old handwritten music scores. In *Proceedings of the International Conference on Frontiers in Handwriting Recognition (ICFHR) 2010*, pages 634–639, 2010.

[77] A. Forns, J. Llads, G. Snchez, and H. Bunke. On the use of textural features for writer identification in old handwritten music scores. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR) 2011*, pages 996–1000. IEEE Computer Society, 2009.

[78] D. H. Foster and S. J. Gilson. Recognizing novel three-dimensional objects by summing signals from parts and views. *Proceedings of the Royal Society: Biological Sciences*, 269(1503):1939–1947, September 2002.

[79] W. T. Freeman and M. Roth. Orientation histograms for hand gesture recognition. In *Proceedings of the International Workshop on Automatic Face and Gesture Recognition*, 1995.

[80] W.T. Freeman, K. Tanaka, J. Ohta, and K. Kyuma. Computer vision for computer games. In *Automatic Face and Gesture Recognition, 1996., Proceedings of the Second International Conference on*, pages 100 –105, oct 1996.

[81] K. Fukushima. Neocognitron - a self-organizing neural network model for a mechanism of pattern-recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202, 1980.

[82] K. Fukushima. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural Networks*, 1(2):119–130, 1988.

[83] J. Ge, Y. Luo, and G. Tei. Real-time pedestrian detection and tracking at nighttime for driver-assistance systems. *Trans. Intell. Transport. Sys.*, 10:283–298, June 2009.

[84] J. J. Gibson. *The Perception of the Visual World*. Houghton Mifflin, Boston, MA, 1950.

[85] J. Gluckman. Higher order image pyramids. In *Proceedings of the European Conference on Computer Vision*, pages 308–320, 2006.

[86] K. Grauman and T. Darrell. Efficient image matching with distributions of local invariant features. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 627–634, 2005.

[87] K. Grauman and T. Darrell. The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features. In *Proceedings of IEEE International Conference on Computer Vision*, volume 2, pages 1458–1465, Los Alamitos, CA, USA, October 2005. IEEE Computer Society.

[88] L. D. Griffin. The second order local-image-structure solid. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:1355–1366, 2007.

[89] L. D. Griffin and M. Lillholm. Classifying local image symmetry using a co-localised family of linear filters. *Perception 37 ECVP Abstract Supplement*, page 122, 2008.

[90] L. D. Griffin and M. Lillholm. Symmetry sensitivities of derivative-of-gaussian filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:1072–1083, 2010.

[91] L. G. Griffin and M. Lillholm. Feature category systems for 2nd order local image structure induced by natural image statistics and otherwise. In *Proceedings of the SPIE*, volume 6492, pages 1–11. SPIE, 2007.

[92] L.D. Griffin. Symmetries of 1-d images. *Journal of Mathematical Imaging & Vision*, 31(2-3):157–164, July 2008.

[93] L.D. Griffin, M. Lillholm, M.S. Crosier, and J. van Sande. Basic image features (BIFs) arising from approximate symmetry type. In *SSVM '09*, pages 343–355, 2009.

[94] M. Grundmann, F. Meier, and I. Essa. 3D shape context and distance transform for action recognition. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, number 1, pages 1–4, 2008.

[95] D. Guillamet and J. Vitria. A comparison of global versus local color histograms for object recognition. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, volume 2, pages 422–425, 2000.

[96] J.L. Hafner, H.S. Sawhney, W. Equitz, M.D. Flickner, and W. Niblack. Efficient color histogram indexing for quadratic form distance functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(7):729–736, July 1995.

[97] A. Hassane, S. Al-Maadeed, J. M. Aljaam, A. Jaoua, and A. Bouridane. The ICDAR2011 arabic writer identification contest. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR) 2011*, 2011.

[98] E. Hayman, B. Caputo, M. Fritz, and J. Eklundh. On the significance of real-world conditions for material classification. In *Proceedings of the European Conference on Computer Vision*, volume 4, pages 253–266, 2004.

[99] J. Hays and A. A. Efros. Scene completion using millions of photographs. *SIGGRAPH 2007*, 26(3), 2007.

[100] Z.Y. He, X. You, and Y.Y. Tang. Writer identification of chinese handwriting documents using hidden markov tree model. *Pattern Recognition*, 41(4):1295–1307, April 2008.

[101] C.M. Heaps. *Similarity and Features of Natural Textures*. University of Tennessee, Knoxville, 1996.

[102] P. S. Hiremath, S. Shivashankar, J. D. Pujari, and R. K. Kartik. Writer identification in a handwritten document image using texture features. In *Proceedings of the International Conference on Signal and Image Processing (ICSIP)*, pages 139–142, 2010.

[103] K. Hotta. Scene classification based on multi-resolution orientation histogram of gabor features. In *Proceedings of the 6th International Conference on Computer Vision Systems*, pages 291–301, 2008.

[104] D. H. Hubel. *Eye, Brain, and Vision (Scientific American Library, No 22)*. W. H. Freeman, 2nd edition, May 1995.

[105] D. H. Hubel and T. N. Wiesel. Receptive fields of single neurones in the cat's striate cortex. *The Journal of Physiology*, 148:574–591, October 1959.

[106] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in cats visual cortex. *Journal of Physiology*, 160(1):106–&, 1962.

[107] J. E. Hummel. Hummel, j. e. (2000). where view-based theories break down: The role of structure in shape perception and object recognition. in e. dietrich and a. markman (eds.). cognitive dynamics: Conceptual change in humans and machines (pp. 157- 185). hillsdale, nj:.

[108] J. E. Hummel and I. Biederman. Dynamic binding in a neural network for shape recognition. *Psychological review*, 99(3):480–517, July 1992.

[109] C. P. Hung, G. Kreiman, T. Poggio, and J. J. DiCarlo. Fast readout of object identity from macaque inferior temporal cortex. *Science*, 310(5749):863–866, 2005.

[110] A. Imdad, S. Bres, V. Eglin, C.J. Rivero Moreno, and H. Emptoz. Writer identification using steered hermite features and svm. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR) 2007*, pages 839–843, 2007.

[111] R. Jain and D. Doermann. Offline writer identification using k-adjacent segments. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR) 2011*, pages 769–773, 2011.

[112] H. Jia and Y. Zhang. Fast human detection by boosting histograms of oriented gradients. In *Proceedings of the Fourth International Conference on Image and Graphics*, pages 683–688, 2007.

[113] B. Julesz. Visual pattern discrimination. *IEEE Transactions on Information Theory*, 8(2):84–92, 1962.

[114] B. Julesz. Experiments in the visual perception of texture. *Scientific American*, (232):34–43, 1975.

[115] B. Julesz. Textons, the elements of texture perception, and their interactions. *Nature*, 290(5802):91–97, March 1981.

[116] A. Kae, D. A. Smith, and E. G. Learned-Miller. Learning on the fly: a font-free approach toward multilingual OCR. *International Journal on Document Analysis and Recognition*, 14(3):289–301, 2011.

[117] T. Kailath. The divergence and Bhattacharyya Distance measures in signal selection. *IEEE Transactions on Communications Technology*, 15(1):52–60, 1967.

[118] K. Karunakara and B. P. Mallikarjunaswamy. Writer identification based on offline handwritten document images in kannada language using empirical mode decomposition method. *International Journal of Computer Applications*, 30(6):31–36, 2011.

[119] K. Kavukcuoglu, P. Sermanet, Y-L. Boureau, K. Gregor, M. Mathieu, and Y. LeCun. Learning convolutional feature hierarchies for visual recognition. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1090–1098, 2010.

[120] Y. Ke and R. Sukthankar. PCA-SIFT: a more distinctive representation for local image descriptors. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 506–513, 2004.

[121] T. Kobayashi, M. Iwamura, and K. Kise. Recognition of affine distorted characters by using affine-invariantlocal descriptors. *Proceedings of the 2nd China-Japan-Korea Joint Workshop on Pattern Recognition (CJKPR2010)*, pages 74–77, 2010.

[122] J. J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.

[123] J. J. Koenderink. Operational significance of receptive field assemblies. *Biological Cybernetics*, 58:163–171, 1988.

[124] J. J. Koenderink and A. J. Van Doorn. The structure of locally orderless images. *International Journal of Computer Vision*, 31:159–168, April 1999.

[125] M. Landy and N. Graham. Visual Perception of Texture. *The Visual Neurosciences*, 2:1106–1118, 2003.

[126] S. Lazebnik, C. Schmid, and J. Ponce. Semi-local affine parts for object recognition. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 959–968, 2004.

[127] S. Lazebnik, C. Schmid, and J. Ponce. A maximum entropy framework for part-based texture and object recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, ICCV '05, pages 832–838, Washington, DC, USA, 2005. IEEE Computer Society.

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

[129] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2169–2178, 2006.

[130] V. F. Leavers. Use of the two-dimensional radon transform to generate a taxonomy of shape for the characterization of abrasive powder particles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1411–1423, 2000.

[131] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, volume 86, pages 2278–2324, 1998.

[132] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 609–616, New York, NY, USA, 2009. ACM.

[133] J. Liang, D. Doermann, and H.P. Li. Camera-based analysis of text and documents: a survey. *International Journal on Document Analysis and Recognition*, 7(2-3):84–104, July 2005.

[134] M. Lillholm and L. D. Griffin. Novel Image Feature Alphabets for Object Recognition. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 3181–3184, 2008.

[135] O. Linde and T. Lindeberg. Object recognition using composed receptive field histograms of higher dimensionality. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, volume 2, pages 1 – 6 Vol.2, 2004.

[136] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer, December 1993.

[137] T. Lindeberg. Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 21(2):224–270, 1994.

[138] T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30:79–116, 1998.

[139] X. Liu and D. Wang. Texture classification using spectral histograms. *IEEE Transactions on Image Processing*, 12(6):661 – 670, june 2003.

[140] N. K. Logothetis, J. Pauls, and T. Poggio. Shape representation in the inferior temporal cortex of monkeys. *Current Biology*, 5(5):552–563, 1995.

[141] G. Louloudis, N. Stamatopoulos, and B. Gatos. Icdar 2011 writer identification contest. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR) 2011*, pages 1475–1479, 2011.

[142] D. R. Lovell, T. Downs, and A. C. Tsoi. An evaluation of the neocognitron. *IEEE Transactions on Neural Networks*, 8:1098–1105, 1997.

[143] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 1150–1157, 1999.

[144] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

[145] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young. ICDAR 2003 robust reading competitions. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR) 2003*, pages 682–687, 2003.

[146] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 689–696, 2009.

[147] M. J. Marin-Jimenez and N. P. de la Blanca. Empirical study of multi-scale filter banks for object categorization. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 578–581, 2006.

[148] D. Marr. *Vision*. W. H. Freeman & Company, New York, 1982.

[149] P. Meer, E.S. Baugher, and A. Rosenfeld. Frequency domain analysis and synthesis of image pyramid generating kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4):512–522, July 1987.

[150] B. W. Mel. Seemore: Combining color, shape, and texture histogramming in a neurally inspired approach to visual object recognition, 1997.

[151] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615 – 1630, oct. 2005.

[152] M. Mirmehdi. Special issue on camera-based text and document recognition. *International Journal on Document Analysis and Recognition*, 7(2-3):83–83, 2005.

[153] F. Moosmann, B. Triggs, and F. Jurie. Fast discriminative visual codebooks using randomized clustering forests. In *Advances in Neural Information Processing Systems (NIPS)*, volume 2, 2007.

[154] P. Moreels and P. Perona. Evaluation of features detectors and descriptors based on 3d objects. *International Journal of Computer Vision*, 73(3):263–284, July 2007.

[155] R. M. Morgan, P. Wiltshire, A. G. Parker, and P. A. Bull. The role of forensic geoscience in wildlife crime detection. *Forensic Science International*, 162:152–162, 2006.

[156] G. Mori, S. Belongie, and J. Malik. Shape contexts enable efficient retrieval of similar shapes. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 723–730, 2001.

[157] G. Mori, S. Belongie, and J. Malik. Efficient shape matching using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1832–1837, 2005.

[158] G. Mori and J. Malik. Recognizing objects in adversarial clutter: Breaking a visual CAPTCHA. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 134, Los Alamitos, CA, USA, 2003. IEEE Computer Society.

[159] J. L. Mundy. Object recognition: The search for representation. In *Object Representation in Computer Vision*, pages 19–50, 1994.

[160] J. L. Mundy. Object recognition in the geometric era: A retrospective. In *Toward Category-Level Object Recognition*, pages 3–28, 2006.

[161] H. Murase and S. K. Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision*, 14(1):5–24, January 1995.

[162] C. Nebauer. Evaluation of convolutional neural networks for visual recognition. *IEEE Transactions on Neural Networks*, 9(4):685–696, 1998.

[163] F. Nejad and M. Rahmati. A new method for writer identification and verification based on farsi/arabic handwritten texts. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR) 2007*, pages 829–833, 2007.

[164] A. J. Newell. A biologically plausible model for handwritten digit recognition. Master's thesis, University College London, 2007.

[165] A. J. Newell, R. M. Morgan, L. D. Griffin, P. A. Bull, J.R. Marshall, and G. Graham. Automated texture recognition of quartz sand grains for forensic applications. *Journal of Forensic Sciences*.

[166] M. Nielsen, L. Florack, and R. Deriche. Regularization, scale-space, and edge detection filters. *J. Math. Imaging Vis.*, 7(4):291–307, October 1997.

[167] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2161–2168, June 2006.

[168] E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In *Proceedings of the European Conference on Computer Vision*, pages 490–503, 2006.

[169] T. Ojala, M. Pietikainen, and D. Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 582 –585 vol.1, oct 1994.

[170] T. Ojala, M. PietikŁinen, and T. MŁenpŁŁ. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[171] C. R. Olson. Object-based vision and attention in primates. *Current Opinion in Neurobiology*, 11(2):171 – 179, 2001.

[172] E. Ong and R. Bowden. A boosted classifier tree for hand shape detection. In *IEEE International Conference on Automatic Face and Gesture Recognition*, volume 1, page 889, 2004.

[173] M. W. Oram and D. I. Perrett. Modeling visual recognition from neurobiological constraints. *Neural Networks*, 7(6-7):945–972, 1994.

[174] S. Pal, M. Blumenstein, and U. Pal. Automatic off-line signature verification systems: A review. *IJCA Proceedings on International Conference and workshop on Emerging Trends in Technology (ICWET)*, (14):20–27, 2011.

[175] P. Pan, Y. Zhu, J. Sun, and S. Naoi. Recognizing characters with severe perspective distortion using hash tables and perspective invariants. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR) 2011*, pages 548–552, 2011.

[176] Y.F. Pan, X.W. Hou, and C.L. Liu. A hybrid approach to detect and localize texts in natural scene images. *IEEE Transactions on Image Processing*, 20(3):800–813, 2011.

[177] M. Panagopoulos, C. Papaodysseus, P. Rousopoulos, D. Dafi, and S. Tracy. Automatic writer identification of ancient greek inscriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(8):1404–1414, August 2009.

[178] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:629–639, 1990.

[179] D. I. Perrett, J. K. Hietanen, M. W. Oram, and P. J. Benson. Organization and functions of cells responsive to faces in the temporal cortex. *Philosophical Transactions of the Royal Society of London Series B*, 335(1273):23–30, JAN 29 1992.

[180] D. I. Perrett and M. W. Oram. Neurophysiology of shape processing. *Image and Vision Computing*, 11(6):317–333, JUL-AUG 1993.

[181] M. Petrou, A. Talebpour, and A. Kadyrov. Reverse engineering the way humans rank textures. *Pattern Anal. Appl.*, 10(2):101–114, April 2007.

[182] T. Poggio and S. Edelman. A network that learns to recognize three-dimensional objects. *Nature*, 343(6255):263–266, January 1990.

[183] N. Prakongkep, A. Suddhiprakarn, I. Kheoruenromne, and R. J. Gilkes. SEM image analysis for characterization of sand grains in Thai paddy soils. *Geoderma*, 156(1-2):20 – 31, 2010.

[184] M. A. Ranzato, Y-L. Boureau, and Y. LeCun. Sparse feature learning for deep belief networks. In *Advances in Neural Information Processing Systems (NIPS)*, page 11851192, 2007.

[185] M. A. Ranzato, C. Poultney, S. Chopra, and Y. Lecun. Efficient Learning of Sparse Representations with an Energy-Based Model. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1137–1144, 2006.

[186] A. R. Rao and G. L. Lohse. Identifying high level features of texture perception. *CVGIP: Graph. Models Image Process.*, 55(3):218–233, May 1993.

[187] X. Ren and J. Malik. Learning a classification model for segmentation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, volume 1, pages 10–17, 2003.

[188] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2(11):1019–1025, 1999.

[189] E. Rivlin, S. J. Dickinson, and A. Rosenfeld. Recognition by functional parts. *Journal of Computer Vision and Understanding*, 62(2):164–176, September 1995.

[190] B. M. Romeny and L. Florack. Front-end vision: A multiscale geometry engine. In *Proceedings of the First IEEE International Workshop on Biologically Motivated Computer Vision*, BMVC '00, pages 297–307, London, UK, UK, 2000. Springer-Verlag.

[191] B.M. Romeny. *Front-End Vision and Multi-Scale Image Analysis: Multi-scale Computer Vision Theory and Applications, written in Mathematica*. Springer Publishing Company, Incorporated, 1st edition, 2009.

[192] R. Rosenholtz, Y. Li, J. Mansfield, and Z. Jin. Feature congestion: a measure of display clutter. *Proceedings of the ACM Conference on Human Factors in Computing Systems*, page 761, 2005.

[193] R. Rosenholtz, Y. Li, and L. Nakano. Measuring Visual Clutter. *Journal of Vision*, 7(2):1–22, August 2007.

[194] C. Rothwell, A. Zisserman, D. A. Forsyth, and J. L. Mundy. Canonical frames for planar object recognition. In *Proceedings of the European Conference on Computer Vision*, pages 757–772, 1992.

[195] J. Rouco, A. Mosquera, M.G. Penedo, M. Ortega, and M. Penas. Texture description in local scale using texton histograms with quadrature filter universal dictionaries. 5(4):211–221, 2011.

[196] J. Rouco, M.G. Penedo, M. Ortega, and A. Mosquera. Texture description in local scale using texton histograms with universal dictionary. In *Proceedings of the International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 47–52, 2009.

[197] K.M. Sayre. *Recognition: a study in the philosophy of artificial intelligence*. University of Notre Dame Press, 1965.

[198] B. Schiele and J. L. Crowley. Object recognition using multidimensional receptive field histograms. In *Proceedings of the European Conference on Computer Vision*, pages 610–619, 1996.

[199] B. Schiele and J. L. Crowley. Recognition without correspondence using multidimensional receptive field histograms. *International Journal of Computer Vision*, 36:31–50, 2000.

[200] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:530–535, 1997.

[201] H. Schneiderman and T. Kanade. Object detection using the statistics of parts. *International Journal of Computer Vision*, 56(3):151–177, 2004.

[202] L.R.B. Schomaker and M. Bulacu. Automatic writer identification using connected-component contours and edge-based features of uppercase western script. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):787–798, June 2004.

[203] L.R.B. Schomaker, K. Franke, and M. Bulacu. Using codebooks of fragmented connected-component contours in forensic and historic writer identification. *Pattern Recognition Letters*, 28(6):719–727, April 2007.

[204] E. Seemann, B. Leibe, K. Mikolajczyk, and B. Schiele. An evaluation of local shape-based features for pedestrian detection. In *Proceedings of the British Machine Vision Conference (BMVC)*, page 1120, 2005.

[205] T. Serre, M. Kouh, C. Cadieu, U. Knoblich, G. Kreiman, and T. Poggio. A theory of object recognition: Computations and circuits in the feedforward path of the ventral stream in primate visual cortex. In *AI Memo*. Massachusetts Institute of Technology, 2005.

[206] T. Serre, G. Kreiman, M. Kouh, C. Cadieu, U. Knoblich, and T. Poggio. A quantitative theory of immediate visual recognition. *Progress in Brain Research*, 165:33–56, 2007.

[207] T. Serre, A. Oliva, and T. Poggio. A feedforward architecture accounts for rapid categorization. *Proceeding of the National Academy of Sciences*, 104(15):6424–6429, 2007.

[208] T. Serre, M. Riesenhuber, J. Louie, and T. Poggio. On the role of object-specific features for real world object recognition in biological vision. In *Proceedings of Biologically Motivated Computer Vision*, volume 2525, pages 387–397, 2002.

[209] 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*, 29:411–426, 2007.

[210] T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 994–1000, 2005.

[211] Y. Shan, H.S. Sawhney, B. Matei, and R. Kumar. Partial object matching with shapeme histograms. In *Proceedings of the European Conference on Computer Vision*, pages Vol III: 442–455, 2004.

[212] Y. Shan, H.S. Sawhney, B. Matei, and R. Kumar. Shapeme histogram projection and matching for partial object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):568–577, April 2006.

[213] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:888–905, 1997.

[214] I. Siddiqi and N. Vincent. A set of chain code based features for writer recognition. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR) 2011*, pages 981–985, 2009.

[215] R. Sigala, T. Serre, T. Poggio, and M. Giese. Learning features of intermediate complexity for the recognition of biological motion. In *Proceedings of the International Conference on Artificial Neural Networks*, 2005.

[216] E. P. Simoncelli and W. T. Freeman. The steerable pyramid: A flexible architecture for multi-scale derivative computation. pages 444–447, 1995.

[217] J. Sivic, B. Russell, A. A. Efros, A. Zisserman, and B. Freeman. Discovering objects and their location in images. In *Proceedings of the International Conference on Computer Vision (ICCV)*, October 2005.

[218] J. Sivic and A. Zisserman. Video Google: a text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision (ICCV)*, volume 2, pages 1470–1477, 2003.

[219] Z. Song, Q. Chen, Z. Y. Huang, Y. Hua, and S. Yan. Contextualizing object detection and classification. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1585–1592, 2011.

[220] S. N. Srihari. Evaluating the rarity of handwriting formations. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR) 2011*, 2011.

[221] S. N. Srihari and G. R. Ball. Writer verification of arabic handwriting. In *Document Analysis Systems*, pages 28–34, 2008.

[222] S.N. Srihari, S.H. Cha, H. Arora, and S.J. Lee. Individuality of handwriting: a validation study. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR) 2003*, pages 106–109, 2001.

[223] S.N. Srihari, S.H. Cha, and S.J. Lee. Establishing handwriting individuality using pattern recognition techniques. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR) 2003*, pages 1195–1204, 2001.

[224] C. Strouthopoulos, N. Papamarkos, and A.E. Atsalakis. Text extraction in complex color documents. *Pattern Recognition*, 35(8):1743–1758, August 2002.

[225] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7:11–32, 1991.

[226] G. Takacs, V. Chandrasekhar, S. Tsai, D. Chen, R. Grzeszczuk, and B. Girod. Unified real-time tracking and recognition with rotation-invariant fast features. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[227] M. J. Tarr. Rotating objects to recognize them: A case study on the role of viewpoint dependency in the recognition of three-dimensional objects. *Psychonomic Bulletin & Review*, 2:55–82, 1995.

[228] M. J. Tarr, H. H. Blthoff, M. Zabinski, and V. Blanz. To what extent do unique parts influence recognition across changes in viewpoint? *Psychological Science*, 8:282–289, 1997.

[229] M. C. Thomas, R. J. Wiltshire, and A. T. Williams. The use of Fourier descriptors in the classification of particle shape. *Sedimentology*, 42:635–645, 1996.

[230] S. J. Thorpe. How can the human visual system process a natural scene in under 150ms? experiments and neural network models. In *European Symposium on Artificial Neural Networks*, 1997.

[231] S. J. Thorpe. Ultra-rapid scene categorization with a wave of spikes. In *Biologically Motivated Computer Vision*, pages 1–15, 2002.

[232] J. Tighe and S. Lazebnik. Superparsing: Scalable nonparametric image parsing with superpixels. In *Proceedings of the European Conference on Computer Vision*, pages 352–365, 2010.

[233] C.I. Tomai, B. Zhang, and S.N. Srihari. Discriminatory power of handwritten words for writer recognition. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages II: 638–641, 2004.

[234] Z. Tu, X. Chen, A. L. Yuille, and S. C. Zhu. Image parsing: Unifying segmentation, detection, and recognition. In *Toward Category-Level Object Recognition*, pages 545–576, 2006.

[235] M. Tuceryan and A. K. Jain. Texture segmentation using voronoi polygons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:211–216, 1989.

[236] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–591. IEEE Comput. Sco. Press, 1991.

[237] S. Uchida, M. Iwamura, S. Omachi, and K. Kise. Ocr fonts revisited for camera-based character recognition. *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2:1134–1137, 2006.

[238] S. Ullman. Aligning pictorial descriptions: an approach to object recognition. *Cognition*, 32(3):193–254, August 1989.

[239] S. Ullman and R. Basri. Recognition by linear combination of models. Technical report, Cambridge, MA, USA, 1989.

[240] S. Ullman, E. Sali, and M. Vidal-Naquet. A fragment-based approach to object representation and classification. *Proceedings of the 4th International Workshop on Visual Form*, 2059:85–100, 2001.

[241] S Ullman and S Soloviev. Computation of pattern invariance in brain-like structures. *Neural Networks*, 12(7-8):1021–1036, OCT-NOV 1999.

[242] M. Unser. Sum and difference histograms for texture classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):118–125, January 1986.

[243] L. van der Maaten and E. Postma. Improving automatic writer identification. In *Proc. of 17th Belgium-Netherlands Conference on Artificial Intelligence*, page 260266, 2005.

[244] B. Van Ginneken and B. M. Ter Haar Romeny. Applications of locally orderless images. *Jounal of Vis. Comm. and Im. Repr.*, 11:196–208, 2000.

[245] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *International Journal of Computer Vision*, 62(1):61–81, 2005.

[246] M. Varma and A. Zisserman. A statistical approach to material classification using image patch exemplars. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):2032–2047, 2009.

[247] K.L. Vincken, A.S.E. Koster, and M.A. Viergever. Probabilistic multiscale image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):109–120, February 1997.

[248] P. A. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision*, 63(2):153–161, 2005.

[249] H. Voorhees and T. Poggio. Detecting blobs as textons in natural images. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 892–899, 1987.

[250] K. Wang and S. Belongie. Word spotting in the wild. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Proceedings of the European Conference on Computer Vision*, volume 6311, pages 591–604. Springer, 2010.

[251] K.Q. Wang and J.A. Kangas. Character location in scene images from digital camera. *Pattern Recognition*, 36(10):2287–2299, October 2003.

[252] L. Wang, C. Wu, D. Chen, and B. Lu. Rotation-invariant human detection scheme based on polar-HOGs feature and double scales direction estimation. *2011 Symposium on Photonics and Optoelectronics (SOPO)*, pages 1–4, 05 2011.

[253] X. Wang, T. X. Han, and S. Yan. An hog-lbp human detector with partial occlusion handling. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 32–39, 2009.

[254] T. Watanabe, S. Ito, and K. Yokoi. Co-occurrence histograms of oriented gradients for pedestrian detection. In *Advances in Image and Video Technology*, volume 5414, pages 37–47. Springer Berlin / Heidelberg, 2009.

[255] J. J. Weinman, E. Learned-Miller, and A. R. Hanson. Scene text recognition using similarity and a lexicon with sparse belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:1733–1746, 2009.

[256] A. T. Williams, R. J. Wiltshire, and M. C. Thomas. Sand grain analysis–image processing, textural algorithms and neural nets. *Computers & Geosciences*, 24(2):111 – 118, 1998.

[257] A. P. Witkin. Scale-space filtering. In *Proceedings of the Eighth international joint conference on Artificial intelligence - Volume 2*, IJCAI'83, pages 1019–1022, San Francisco, CA, USA, 1983. Morgan Kaufmann Publishers Inc.

[258] P. L. Worthington and E. R. Hancock. Histogram-based object recognition using shape-from-shading. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 643–648, 2000.

[259] www.kaggle.com.

[260] S. Yang and Y. Wang. Rotation invariant shape contexts based on feature-space fourier transformation. pages 575–579, 2007.

[261] C. Yi and Y. Tian. Text string detection from natural scenes by structure-based partition and grouping. *IEEE Transactions on Image Processing*, 20(9):2594–2605, September 2011.

[262] S.S. Yoon, S.S. Choi, S.H. Cha, and C.C. Tappert. Writer profiling using handwriting copybook styles. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR) 2005*, pages 600–604, 2005.

[263] R. A. Young. The Gaussian derivative model for spatial vision: I. Retinal mechanisms. *Spatial vision*, 2(4):273–293, 1987.

[264] R A Young and R M Lesperance. The gaussian derivative model for spatial-temporal vision: II. cortical data. *Spat Vis*, 14(3-4):321–89, 2001.

[265] S. Zeki. *A Vision of the Brain*. John Wiley & Sons, 1993.

[266] B. Zhang and S.N. Srihari. Analysis of handwriting individuality using word features. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR) 2003*, pages 1142–1146, 2003.

[267] J. Zhang, M. Marszalek, S. Lazebnik, and Schmid C. Local features and kernels for classification of texture and object categories: a comprehensive study. In *Conference on Computer Vision and Pattern Recognition Workshop*, page 13, 2006.

[268] Z. Zhang, L. Jin, K. Ding, and X. Gao. Character-SIFT: A Novel Feature for Offline Handwritten Chinese Character Recognition. *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR) 2011*, pages 763–767, 2009.

[269] H. Zhou, D.J. Lin, and T.S. Huang. Static hand gesture recognition based on local orientation histogram feature distribution model. In *Conference on Computer Vision and Pattern Recognition Workshop*, page 161, 2004.

[270] J. Zhou, L. Xin, and D. Zhang. Scale-orientation histogram for texture image retrieval. *Pattern Recognition*, 36(4):1061 – 1063, 2003.

[271] Q. Zhu, S. Avidan, M-c. Yeh, and K. Cheng. Fast human detection using a cascade of histograms of oriented gradients. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1491–1498, 2006.

[272] S.-C. Zhu, C.-E. Guo, Y. Wang, and Z. Xu. What are textons? *International Journal of Computer Vision*, 62(1-2):121–143, April 2005.