Universität des Saarlandes
Max-Planck-Institut für Informatik

Encoding Spatial Context in Local Image Descriptors

Masterarbeit im Fach Informatik
Master's Thesis in Computer Science
von / by

Dushyant Mehta

angefertigt unter der Leitung von / supervised by

Dr. Roland Angst

betreut von / advised by

Dr. Roland Angst

begutachtet von / reviewers

Dr. Roland Angst
Prof. Dr. Joachim Weickert

Saarbrücken, February 28, 2016

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides Statt, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Statement in Lieu of an Oath

I hereby confirm that I have written this thesis on my own and that I have not used any other media or materials than the ones referred to in this thesis.

Einverständniserklärung

Ich bin damit einverstanden, dass meine (bestandene) Arbeit in beiden Versionen in die Bibliothek der Informatik aufgenommen und damit veröffentlicht wird.

Declaration of Consent

I agree to make both versions of my thesis (with a passing grade) accessible to the public by having them added to the library of the Computer Science Department.

Saarbrücken, February 28, 2016 Dushyant Mehta

Abstract

This work has two underlying themes. The main theme of this work is exploiting the relative orientations of local image descriptors as a means of increasing the discriminativeness of descriptors while retaining in-plane rotation invariance. We use the shallow image classification pipeline as the pertinent application setting to develop and examine the effect of this contextual information. We then port it to Convolutional Neural Networks. The secondary theme concerns understanding the mechanisms of capture and encoding of contextual information in both, shallow and deep, classification pipelines. Understanding the nature of feature hierarchies in both pipelines opens up the possibility of applying methods developed for one pipeline onto the other.

We take a close look at Dense SIFT and deduce that implicit relative orientation information is the key to its efficacy. We then propose methods to explicitly capture this relative orientation context information from local descriptor neighbourhoods, while retaining in-plane rotation invariance. Towards this we propose a 2D Histogram approach for context and appearance capture. We additionally propose a directional pooling mechanism for context extracted from feature neighbourhoods that strives to reject clutter.

We also discuss the capture of weak spatial co-occurrence relationship between descriptors as a means to further improve discriminativeness, while exploring various ways of encoding this information.

We then propose modifications to Convolutional Networks to make a special allowance for feature orientation information. This coaxes the network to decouple feature appearance and rotational invariance learning, resulting in a reduction in the number of learnable parameters. Substituting relative orientation information in the proposed Convolutional Layers in place of absolute orientation information leads to the additional benefit of inherently in-rotation invariant representations.

Acknowledgments

This work was made possible through the direct and indirect contribution of many. I am grateful to Dr. Roland Angst for allowing me the freedom to explore various seemingly tangential ideas, while helping me shape and mold them into a coherent form.

It goes without saying that I am grateful for my family's support in this endeavor.

I would also like to thank Dr. Tobias Ritschel and his research group at MPII for being a stable anchor during my academic pursuits, as well as fellow dwellers of the *student cave* for lending a patient ear to my academic and non-academic whining, in particular Elena, Rui and Junaid. I am also thankful to Eldar and Nik for help debugging Caffe code, as well as their suggestions towards the content of the thesis.

Contents

1 Introduction **1**
1.1 A Brief Glimpse Into Computer Vision: Applications & Issues 1
1.2 Overview Of The Thesis 3

2 Background and Related Work **5**
2.1 Machine Learning Perspective on Image Classification 5
2.1.1 Support Vector Machines 6
2.1.2 Decision Trees and Forests 7
2.2 Image Classification Pipeline 7
2.2.1 Image Features 8
2.2.2 Encoding Image Descriptors 9
2.3 Datasets 10

3 A Detailed Look At Contextual Cues In Image Representation **13**
3.1 Broad Categorization of Context 13
3.2 Utilizing Contextual Cues 14
3.2.1 Contextual Cues As Features 14
3.2.2 Augment Local Features With Cues From The Spatial Neighbourhood 14
3.2.3 Global Feature Distribution Statistics 15
3.2.4 Voting or Constraints in Post Processing 15
3.2.5 Part Based Models 16
3.3 Feature Hierarchy, Rotation Invariance and Geometric Context: A Case for Disctinctiveness Of Visual Words in BoW 16
3.4 Aspects Of Context Explored In This Work 17

4 Detour: Why does Dense SIFT work so well? **19**
4.1 Factoring Out the Number of Samples and Scale Information 20
4.2 Quantifying the Effect of Number of Samples 20
4.3 Discriminativeness and Geometric Invariance Redux 22

5 Intentionally Capturing Relative Orientation Context **23**
5.1 Explicitly Encoding Relative Orientation Context: First Steps 23
5.1.1 Orthogonal Clustering of Appearance and Context 24
5.2 Re-incorporating Relative Orientation Information into SIFT 26
5.3 Directional Context Capture 27
5.3.1 Discriminative Feature Selection 28
5.3.2 Directional Context Results 29

6 Going Beyond Relative Orientation Context **31**
6.1 Coarse Keypoint SIFT in Keypoint Neighbourhood as Context 31
6.1.1 Coarse Keypoint SIFT and Re-incorporated Relative Orientation as Context . . . 32
6.2 Combining Coarse Keypoint SIFT & Relative Orientation Histograms in Keypoint Neighbourhood 33
6.2.1 Coarse Keypoint SIFT in Keypoint Neigbourhood Without Clustering 34
6.3 Coarse Keypoint SIFT Histogram in Keypoint Neighbourhood as Feature 34
6.4 Directional Capture of Coarse Keypoint SIFT Context 35

7 Orientation Context in Convolutional Neural Networks 37
7.1 Deep Learning Background and Motivation 37
7.1.1 (Short) Introduction To Deep Learning 37
7.1.2 Invariance and Context In Deep Networks 38
7.1.3 Modifying Deep Architectures Through Traditional Computer Vision Methods . . 40
7.2 Orientation Aware Convolutions 41
7.2.1 Gradient Oriented Convolution Layer 42
7.2.2 Kernel Orientation Max Pool Convolution Layer 42
7.3 Incorporating Relative Orientation In CNNs 43

8 Conclusion and Implementation Notes 45
8.1 Summary 45
8.2 Implementation Notes 45
8.3 Future Work 46

List of Figures 47

List of Tables 49

Bibliography 51

Chapter 1

Introduction

Applications of *Computer Vision* span from object detection and image search to activity recognition and pose estimation, bolstered in no small part by ever more capable processing hardware, cleverer data storage and processing, and advances in *machine learning.* The past decade has also witnessed the resurgence of *Neural Networks* as a one stop solution for various Computer Vision tasks that previously required hand-engineered multi-stage solutions. While Computer Vision has made significant strides with *Deep Neural Networks*, partly made possible through vast amounts of labeled data, there still remain plenty of problems to tackle, and plenty more that come with the use of Deep Neural architectures.

This work has two broad underlying themes.
- First, we examine Dense SIFT [Lowe, 2004] closely and uncover its secret sauce, which will turn out to be a form of spatial context implicitly incorporated in the method. We then use this gleaned information to come up with new methods that capture similar information while trying not to compromise on in-plane rotation invariance. We then extend the developed framework to capture weak co-occurrence relationships between features.
- Second, we provide further examples towards the assertion that deep learning architectures can benefit from the lessons learned during the past few decades of feature engineering in Computer Vision [Chatfield et al., 2014]. We have already seen this work the other way around, with methods like data augmentation that came into the fore with deep learning yielding similar gains in traditional shallow architectures. Lessons from traditional approaches can be considered guiding principles to come up with novel approaches and neural architectures. We exemplify this through modified versions of *Convolutional Layer*, which is the key building block of a class of Neural Networks called *Convolutional Neural Networks.*

If you are well versed in Computer Vision terminology, you can proceed on directly to Section 1.2 for an overview of the Thesis and then onto Chapter 4 for original contribution of this work.

1.1 A Brief Glimpse Into Computer Vision: Applications & Issues

We begin by describing common computer vision tasks through illustrative examples, and discuss the challenges and demands placed on image features. We primarily motivate *image classification* as the application setting for the later chapters, and the desire for immunity of the features to specific *geometric variations.*

Applications

Having computers make sense of images and image sequences finds use in multiple scenarios, some of which are enumerated below in a generally increasing order of implementation and computational complexity. The list is by no means exhaustive.

- Image Classification: Given an image and n categories, predict the category that the image belongs to.
  - Coarse: Given an image of a sparrow, predict if it is a bird or a plane or Superman. Large dissimilitude between categories is a characteristic of coarse classification tasks, making it easier

to extract representative features unaffected by intra-class variations or pose and orientation variations.

  - Fine Grained: Given an image of a sparrow, predict what species of sparrow it is. Fine Grained categories tend to be very similar, making it harder to extract unique identifying characteristics without undoing the effect of pose and orientation, so as to pinpoint precisely the relevant variations that dictate class boundaries.

- Image Retrieval: Given a text or an image query, retrieve all images similar in content or style to the query.
  - By Type: Given the text "sparrow" or an image of a sparrow, retrieve all images of sparrows in the dataset.
  - By Style: Given a Jackson Pollock, retrieve all images of accidental or deliberate paint splatter in the dataset.
  - By Instance: Given the text "Chuck Jones" or an image of Chuck Jones, retrieve all images of Chuck Jones in the dataset and not of other people, i. e., Chuck Jones is an instance of the class *Person*.
- Object Detection: Given an image, identify and localize semantic parts in the image.
  Example: Given an image of a busy street, draw bounding boxes around all the people in the scene, all the cars in the scene, the sign posts and such, and correctly identify each as belonging to the respective category.
- Human Pose Estimation: Given a single image of a person or images from multiple angles, estimate the pose of the person. Image data may also be augmented by other modalities, such as depth information.
- Activity or Gesture Recognition: Given an image or an image sequence, recognize the activity that one or more people in the image are engaging in. This typically makes use of the preceding techniques as building blocks.
  Example: Differentiate between different hand gestures for the purposes of interacting with an interface, or track and log suspicious behaviour in security camera feeds.

Of note here is that some tasks, such as image classification and object detection, are end applications in their own right, and also contribute to other applications as building blocks.

Furthermore, applications may share the underlying methodology and representation. Take image classification and image retrieval for instance. Going from an image to a representation that the system can reason about takes the same route for both tasks. The difference being that image classification passes on the image representation to a classifier, while image retrieval relies on efficient matching, indexing and storage of the representations. We elaborate on this in Section 2.2.

Issues

There are nuances to the aforementioned applications, and challenges that must be dealt with. Elaborating on what we mentioned in the case of Fine Grained classification, a good image representation would be comprised of elements that are representative, i. e., occur frequently for some or all classes, and discriminative/distinctive, i. e., can be used to make a distinction between categories.

Depending on the application and the subject matter of the image, these image representations need to deal with variations in scene lighting, global transformations such as in-plane translation and rotation, change of camera viewpoint, local deformations such as pose change, occlusions etc. The nature of the application places specific demands on the image representation. For instance, representations for human detection need to be agnostic of pose whereas representations that may be used for activity tracking need to representative of the pose. In practice, there is often a trade off seen between invariance to the sundry transformations described above and discriminativeness. There is an eternal struggle to devise image descriptors that retain discriminativeness while being immune to various transformations. Briefly elaborating on the case for 'representativeness' made earlier, image descriptors must also be wary of encoding spurious but frequently occurring elements, i. e., clutter which may hamper the 'discriminativeness' of the representation.

In the end, the objective is to develop descriptors that are representative and discriminative, while being immune to or cognizant of various transformations as per application, and agnostic of clutter in the scene. In this work, the focus is on geometric transformation invariance. Additionally, storage and computational resources are of essence, and the descriptors proposed must be quick to compute and allow rapid storage and access.

Deep Learning architectures alleviate the explicit representativeness and discriminativeness requirements to a large extent, through massive amounts of labeled data and complex models backed by better computing resources. Dealing with image transformations involves specific tricks such as data augmentation [Krizhevsky et al., 2012], but the lack of an elegant solution is conspicuous. There have been attempts recently to remedy that [Jaderberg et al., 2015] [Gens, 2014].

1.2 Overview Of The Thesis

As delineated earlier, this thesis uses image classification as the pertinent application setting. Specifically *scene classification* on the Fifteen Scene Categories dataset [Lazebnik, 2004] and *digit classification* on MNIST variants [LeCun et al., 1998].

The chapter that follows (Chapter 2) goes a bit more in depth regarding the image classification pipeline, discusses popular image descriptors (including SIFT [Lowe, 2004]) and establishes much of the terminology used, including Bag-of-Words.

Chapter 3 defines *Context* and elaborates on the nuances of image cues that lead to more descriptive features. Particular emphasis is on geometric cues, which are the focus of this thesis. It also ties together Bag-of-Words and image matching, and discusses *discriminativeness vs geometric invariance* in that light.

Chapter 4 examines Dense SIFT in depth and posits *relative orientation* of keypoints as a contributing factor to the potency of Dense SIFT in a Bag-of-Words framework. This newfound knowledge is then used to devise in-plane rotation invariant descriptors with boosted discriminativeness in Chapter 5.

Chapter 6 explores alternative forms of spatial relationships that can be incorporated in much the same way as the context developed in the preceding chapter.

Chapter 7 ports *orientation* context onto Convolutional Networks. It provides a brief overview of Convolutional Neural Networks, followed by the description of a new convolutional layer that is aware of local in-plane rotations, resulting in compaction of the number of learnable parameters. Then we propose a variant of the layer that makes special allowance for *relative orientation* information, to seek representations that are inherently rotation invariant. This alleviates the need for in-plane rotation data augmentation while requiring fewer distinct kernels to be learned.

Chapter 2

Background and Related Work

As mentioned earlier, the requirements placed upon image descriptors are conditional on the application at hand. Before we get started with a discussion of image descriptors for image classification, it would serve us well to briefly examine the Machine Learning aspects of classification as well as the typical image classification pipeline.

In This Chapter

- Binary and Multi-class Classification (Section 2.1)
- Walk-through of the Image Classification Pipeline (Section 2.2)
- Discussion of Image Features and Feature Hierarchy (Section 2.2.1)
- Brief Look at Different Feature Encoding Schemes (Section 2.2.2)

2.1 Machine Learning Perspective on Image Classification

Machine Learning entails solving a task by leveraging available labeled data (supervised) or unlabeled data (unsupervised) to deduce the parameters of a model such that some objective function representative of the performance on the task is maximized (or minimized). For the purposes of classification or regression, most Machine Learning methods operate on features of fixed length, for that allows one to reason in a $d-$dimensional vector space. Once features are represented as points in this $d-$dimensional vector space, one has all the tools of Linear Algebra at one's disposal to formulate and attack the learning problem.

The problem of Supervised Binary Classification amounts to finding a boundary in the vector space that best separates the data points belonging to the two classes, with the hope that the data sample-set this boundary is found for is a good representative of the underlying feature-class distribution. *Support Vector Machines* are an example of this. One need also ensure that the learned boundary is unaffected by the peculiarities of the sampling and the noise contained in the data sample, i. e., it should *generalize* well on out-of-training data. This boundary may not necessarily be a contiguous, smooth boundary. Cases where there are pockets of one class's data points embedded in the other class's data points would require a different, more convoluted way of partitioning the feature space into the two classes. *k-Nearest Neighbours* and *Classification Trees* are an example of this. There is often a limit placed on how complex this boundary can be, which points once again at *generalization.*

Multi-class classification problem can be thought of in two ways. The intuitive way is to construct a partitioning of the feature space such as to identify feature clusters that map to each class. This gets hairy quickly as the number of classes increase, often accompanied by fuzzying boundaries of the feature clusters. The second way is to reduce the problem of Multi-class classification to a set of Binary Classification problems:
OVA: For k classes, one can choose to solve k One-vs-All classification problems, where k boundaries are constructed separating each class from the rest, and the final decision is taken based on confidence scores coming from each of the k classifications. This may suffer from unbalanced class distributions, and would

require special handling. Also, the confidence scores produced by the k classifiers need to be comparable [Tewari and Bartlett, 2007].
OVO: The alternative is to solve $k(k-1)/2$ One-vs-One classification problems, where a boundary separating each pair of classes is learned, and the final classification decision taken based on the votes each class received.

There is a plethora of Supervised Classification techniques in literature, but here we touch upon two that are most common in practice.

2.1.1 Support Vector Machines

Considering a *linear* Binary Classification problem with data points of the two classes well separated, one can find infinitely many hyperplanes that can be used as the boundary. See Figure 2.1.

However, not all hyperplanes would generalize well, and it is in our interest to find one which is the furthest away from the nearest data points of both classes, i. e., has the *maximum margin*. Being the furthest possible distance away from the nearest data points of either class ensures that the boundary has room to forgive transgressions of the unseen data points that would have caused the classification decision to erroneously invert, were the margin very small. In the adjacent figure, we see that hyperplane H_1 is more adept at dealing with deviations of test data points from the training data points than hyperplane H_2.

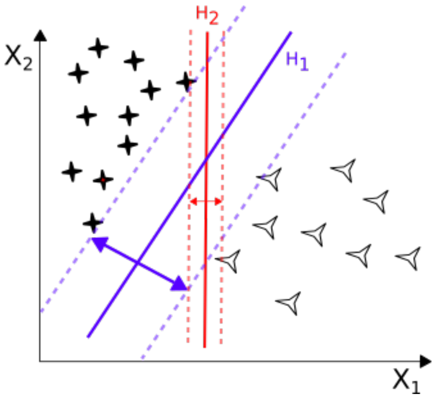


Figure 2.1: Hyperplanes and their margins for linear classification of separable data

Consider feature-label pairs $(\vec{x_1}, y_1)\,..\,(\vec{x_n}, y_n)$, where $\vec{x_i}$ are feature vectors in d−dimensional space and $y_i \in \{1, -1\}$ are the labels for the two classes.

A hyperplane in this space can be expressed as $\vec{w}.\vec{x} - b = 0$, where $\vec{w}$ is the normal vector of the hyperplane and b is the offset of the hyperplane from origin along $\vec{w}$.

Then the margin planes can be expressed with equations $\vec{w}.\vec{x} - b = -1$ and $\vec{w}.\vec{x} - b = 1$, separated by a margin of $\frac{2}{||w||}$. Thus, to maximize the margin, one needs to minimize $||\vec{w}||$.

Additionally, the data points must be constrained to lie outside the margins by imposing:

$$y_i(\vec{w}.\vec{x_i} - b) \geq 1 \qquad \forall 1 \leq i \leq n$$

Allowing for the case where the data in non-separable, the SVM problem can be expressed as below by introducing slack variables ζ_i and the regularization parameter λ:

$$\begin{aligned} &\underset{\vec{w}}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n}\zeta_i + \lambda||\vec{w}||^2 \\ \textit{subject to}: \quad &\zeta_i \;\geq\; 0 \quad \forall i = 1..n \\ &\zeta_i \;\geq\; 1 - y_i(\vec{w}.\vec{x_i} - b)\forall i = 1..n \end{aligned}$$

The above optimization problem when formulated as its dual, allows the use of similarity measures or *Kernels* other than the dot-product. Since we would be dealing with histograms for the most part, we make use of the Histogram Intersection Kernel which is defined as

$$K(H^a, H^b) = \sum_{i=1}^{n} \min(h_i^a, h_i^b)$$

where h_i^x is the count in the i^{th} bin of Histogram H^x.

2.1.2 Decision Trees and Forests

Decision Trees work by casting Classification problems as *recursive partitioning* problems on the feature space, i. e., the boundary separating the classes may be arbitrarily complex and the space is hierarchically partitioned using one or more features at a time. The models are regularized by restricting the granularity of the partitioning.

Typically, each node of the tree partitions the feature space along one dimension, resulting in axis-parallel cuts. Hence, even data that may be separated by a linear combination of multiple variables would necessitate several repeated cuts by the tree, taking up several levels.

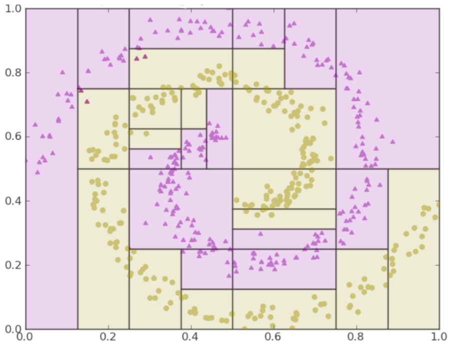


Figure 2.2: Representation of the partitioning of feature space induced by Decision Trees. Source: classes.cs.uchicago.edu

There are further techniques used to improve the generalization, such as Ensemble Learning, which leverages 'wisdom of the crowds' by using a diverse set of hypotheses. One of the ways to encourage diversity is using multiple trees trained on randomly sampled subsets of the training data. This approach is known as *Bagging* and the resulting tree ensembles are known as *Decision Forests*.

Requirements From Features

From the discussion thus far:

1. Features must have the same dimensions
2. Features of different classes must be *discriminative*, i. e., well separable either in feature space or in some readily discernible kernel space
3. Features sparsity is preferable because among other advantages, it allows easier partitioning of data with axis parallel cuts close to the origin. Image retrieval additionally relies on feature sparsity for efficient indexing
4. Feature dimension d should be restricted (Curse of Dimensionality)

This, of course is a gross oversimplification of Machine Learning, but would suffice as background necessary to follow the discussion to come. For a more nuanced handling of the subject matter, refer to one of the standard texts in Machine Learning.

2.2 Image Classification Pipeline

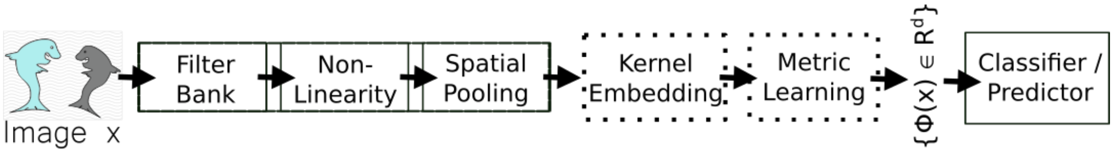


Figure 2.3: Typical Stages of the Image Classification Pipeline

Stages of the image classification pipeline:

- **Filter Bank / Low Level Features**: Features that lie a level above pixel representation. These describe the happenings in regions of small spatial support, and can be alternatively viewed as the responses of a set of filters convolved with the image. These are the building blocks of global image descriptors, and thus many of the requirements for various invariances need to met at this stage.
- **Non-Linearity / Encoding**: Taking the vastly differing number of low level features per image and converting to a fixed length representation in a way that semantically similar content lies closer together in this space than does semantically dissimilar content.

- **Spatial Pooling**: Partitioning the global representation into sub-representations by choosing to account for different sub-regions of the image such as to retain some semblance of spatial structure of the image in the representation.
- **Kernel Embedding***: An optional step which defines an alternative similarity measure in the feature space should you not be satisfied with the inner product. It can be viewed as a mapping to a higher dimensional space such that the classes that were not separable (for the purposes of classification) become separable in this new feature space. We briefly touched upon this in the previous section.
- **Metric Learning***: An optional step, where, together with the classifier the system also learn which dimensions of the feature space are actually important.
- **Classifier**: Learned in a supervised manner, i. e., through example feature-label pairs provided during the training phase.

There are often multiple instantiations of the first three stages, progressively constructing higher level features from lower level features. We'll touch upon the first and the second stage briefly as those would be the focus of development in later chapters. The next section also discusses feature hierarchy and provides the first glimpse of *image context.* We leave the discussion of *Spatial Pooling* to the next chapter, where we present it in light of image context.

2.2.1 Image Features

For our image (global) descriptors to afford various invariances, it is germane to build up these descriptors out of features that embody the same invariances. Under changes of global illumination or slight 3D rotation or translation, simple features such as corners, edges and edge-orientations do not change much. Thus progressive hierarchies of features may be constructed out of the features lower in the hierarchy, and gradually the notion of semantic parts and semantic objects begins to emerge. In going up the hierarchy, geometric relationship between the lower level parts may be preserved or discarded entirely. Depending on the image or task at hand, these features are variously termed as *mid-level* features, and may or may not have a semantic element attached [Bansal, 2015] [Boureau et al., 2010].

The image may be regularly sampled to extract these low level features, an approach that harkens to the *Filter Bank* perspective. Alternatively, one can use *Keypoint* detectors to identify image regions that can repeatably and reliably be localized in the image and anchor the extraction of features to these locations. Additionally, this feature hierarchy is generic enough to lend itself to tasks other than image classification. For instance, features extracted at Keypoints are used as landmarks in *homography* wherein the transformation between the viewpoints of two or more images needs to be estimated.

Hand-crafted descriptors

Haar-like Features: Calculate the difference between average intensities in adjacent rectangular windows with varying spatial arrangements. See Figure 2.4

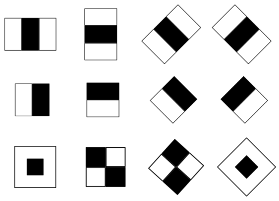


Figure 2.4: A representative set of Haar-like low level features

SIFT: Scale Invariant Feature Transform [Lowe, 2004] comprises of two parts: scale invariant oriented keypoints, i. e., regions of the image that can be readily and repeatably localized despite changes in image scale or illumination, and the actual descriptor which computes a histogram of gradients at the keypoint. The keypoints are identified as the maxima or minima in the *Difference of Gaussians* pyramid [Lindeberg, 2015] and oriented along the dominant local gradient.

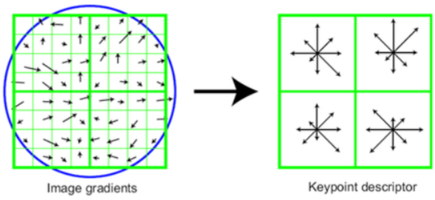


Figure 2.5: Example of SIFT orientation histograms for an 8 × 8 neighbourhood [Lowe, 2004]

The keypoint descriptor utilizes orientation histograms with 8 bins, extracted from 4 × 4 pixel regions. The orientations are weighted by the magnitudes of the gradients. Typically a neighbourhood of size 16 × 16 is chosen, tiled by the 4 × 4 pixel regions, each contributing a size 8 orientation histogram. This results in a 128-dimensional descriptor. The orientations may further be weighted by a Gaussian centered at the keypoint location. Figure 2.5 shows this process for an 8 × 8 neighbourhood.

For scale invariance, the 16 × 16 neighbourhood where the descriptor is computed is re-scaled by the keypoint scale. We'll call this **Keypoint SIFT**. If the orientations that the descriptor bins are computed with respect to the dominant gradient of the keypoint, the resulting descriptor is rotationally invariant and we term it **Relatively Aligned Keypoint SIFT**. If, however, the orientations binned are absolute, we term the resulting descriptor **Upright Keypoint SIFT**.

Dense SIFT: One may choose instead to regularly sample SIFT descriptors from the image rather than at the keypoints. This is scale and local gradient orientation agnostic, and can be seen as another variety of *Upright SIFT*. See Figure 4.1.

HOG: Histogram of Oriented Gradients [Dalal and Triggs, 2005] again leverages orientation information, paired with a discriminatively trained classifier for detecting humans, primarily through silhouette gradients. This shares similarity with Dense SIFT in that it does not use the dominant local gradient to align the neighbourhood, and regularly samples the descriptor from the image.

GIST: Treat the image as a manifold and extract statistics of the orientations as well as spectrograms from different locations in the image [Oliva and Torralba, 2001]

2.2.2 Encoding Image Descriptors

We have already seen some degree of non-linearity and quantization in the hand crafted local descriptors encountered before. Here we discuss some methods that make the goals of encoding more explicit, i. e., sparsity of representation, and fixed dimensional representations. Our objective is to take the varying number of local features from across the image and map the combined set from an image to a d-dimensional vector space.

Sparsity in features is desirable because feature space partitioning methods have an easier time defining axis parallel cuts, and features can be stored more efficiently.

Bag of Visual Words

Akin to words in text corpora, images can be described by means of visual "words" constructed from local image features [Sivic et al., 2005]. In the case of document classification, a histogram with occurrence counts of various words is called a Bag-of-Words. The process starts by mapping the local image features to a codebook or a dictionary of a particular size, such that each feature identifies itself by a representative code word. This is typically achieved through unsupervised clustering in the feature space, with each cluster of similar local features represented by a particular code word. Then the occurrence count of code words of all types is computed per image, and this histogram is called the Bag-of-Visual-Words.

Often, the cluster boundaries are fuzzy and a hard assignment may not be the right choice. The alternative is to assign each feature to multiple clusters with associated confidence weights, and sum up these weight contributions to each code word rather than occurrence counts. This is known as *soft assignment*.

Bag of Words encoding can be applied to any type of local feature, and the encoding results in a descriptor of a fixed length, dependent only on the granularity of the code book. It forgoes all information about the spatial distribution of the local features, which gains it a degree of immunity to various geometric transformations.

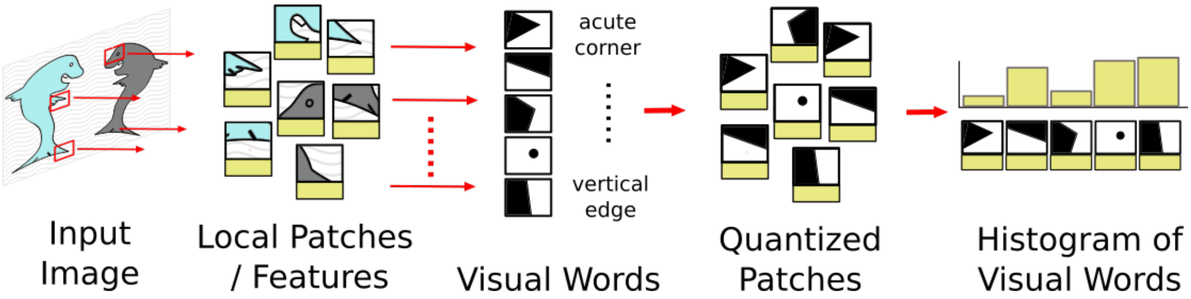


Figure 2.6: Schema for Bag of Visual Words encoding

More Descriptive Encoding Schemes

The Bag-of-Words encoding described above loses all information about the underlying feature space and is only concerned with the partitioning of the feature space. There are schemes like **VLAD** (Vector of Locally Aggregated Descriptors) [Jégou et al., 2010] and **Fisher Vectors** [Verbeek, 2012] which encode first-order and second order statistics of the features in each cluster.

VLAD encoding captures, for each cluster k, the sum $\mathbf{v}_k = \sum_{i=1}^{N} q_{ik}(\mathbf{x}_i - \mu_k)$, where q_{ik} captures the strength of association of feature i to cluster k, $\mathbf{x}_i$s are the features and μ_k is the mean of cluster k.
In a similar fashion, Fisher Vectors capture second-order information.

Brief Word on Deep Vision Pipelines

Deep Vision pipelines like AlexNet [Krizhevsky et al., 2012] and VGG-net [Simonyan and Zisserman, 2014b] substitute hand-engineering of different stages of the pipeline with *end to end* learning of a single multi-layer network that takes the image as input and outputs classification probabilities. The distinct stages mentioned before are implicitly captured by the network with the advantage that they can be jointly optimized for best performance. We refer the reader to Chapter 7 for details.

2.3 Datasets

Fifteen Scene Categories

This dataset [Lazebnik, 2004] contains fifteen natural and man made scenes, ranging from indoor scenes such as 'bedroom', 'store', to man made outdoor scenes such as 'suburbs', 'tall building' to natural outdoor scenes such as 'mountain', 'open country'. The scenes are all shot with the horizon line (regardless of its visibility) almost horizontal in the image, i. e., there isn't a diverse range of in-plane rotations in the dataset. It comprises of ≈ 4400 images in all, with some categories having ≈ 350 images per category and others with ≈ 150.
For our purpose, we do a 90-10 split of the dataset ensuring a similar split for each category, using 90% of the images for training and cross-validation purposes and the remaining 10% for testing.

To test for rotational-invariance as well as the discriminativeness of the learned descriptors, we create a new dataset derived from Fifteen Scene Categories dataset. This dataset starts off with the same 90-10 split. The test set is augmented with rotated copies of the original test set, with rotational angles picked at random from $\{180, 165, 90, 75, 30, -30, -75, -90\}$. Further, rotated copies of 10% images randomly picked from the training set are added to the test set. The final test set is comprised of one part unseen un-rotated images, one part seen but rotated images and one part unseen but rotated images. The training set does not see in-plane rotations. We use *imrotate* in Matlab, with bilinear interpolation and images remaining uncropped. We would refer to this dataset as **Fifteen Scenes Rot** dataset.

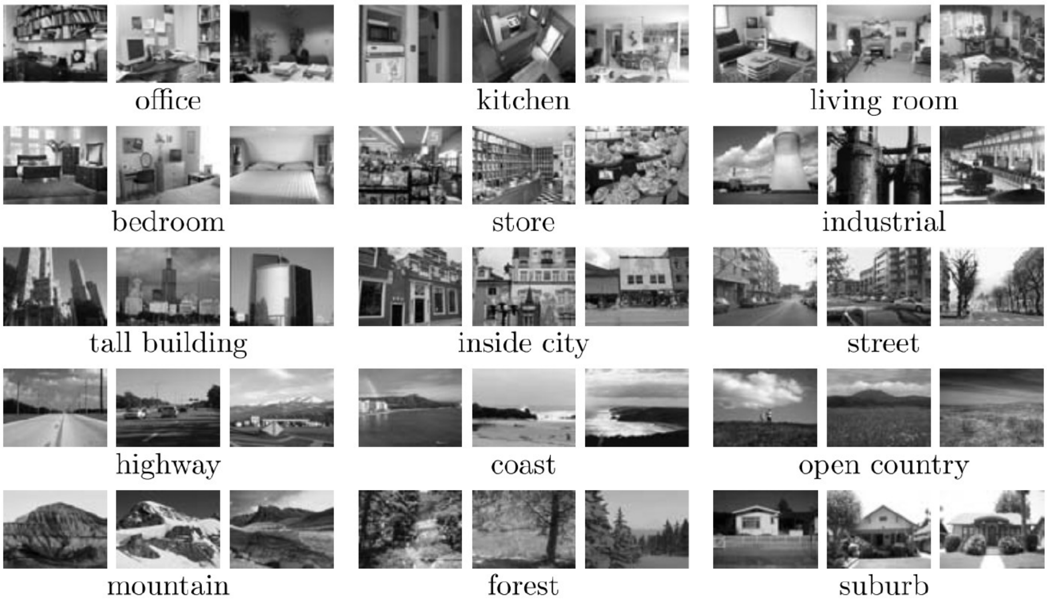
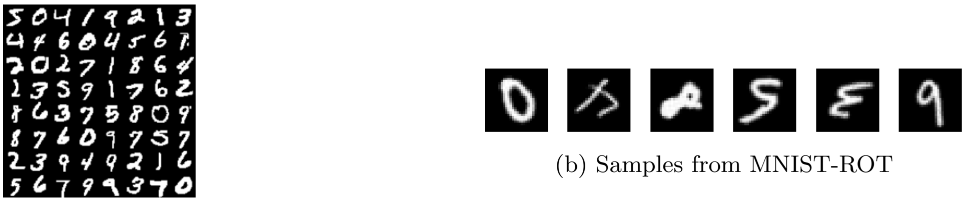


Figure 2.7: Categories represented in Fifteen Scene Categories Dataset [Lazebnik, 2004]

MNIST

The MNIST [LeCun et al., 1998] dataset of handwritten digits comprises of a training set with 60k images and a test set with 10k images. Variants of MNIST exist to capture additional modes of variation such as in-plane rotation, background noise and clutter. MNIST-ROT contains 12k train images and 50k test images, generated from MNIST by rotating randomly picked samples by an angle picked uniformly at random between 0 and 2π.



(a) Samples from MNIST dataset

(b) Samples from MNIST-ROT

Figure 2.8: MNIST Dataset and Variants

Chapter 3

A Detailed Look At Contextual Cues In Image Representation

The dominant theme of this chapter is the capture of additional cues that go beyond the small support regions seen by local features. The explicit aim is making image representations well separable feature space for the purpose of image classification. The capture of these additional cues has a bearing on invariance to geometric transformations as well as the sparsity of the representation, in addition to discriminativeness.

In This Chapter

- Types of contextual cues (Section 3.1)
- Mechanisms of encoding *context* in the typical image classification pipeline (Section 3.2)
- Geometric Context, Rotation Invariance and Feature Hierarchy (Section 3.3)
- Summary of the primary directions of reasoning about contextual cues in this thesis (Section 3.4)

3.1 Broad Categorization of Context

One possible way of thinking about the available cues is at the semantic level:

- Semantic Context: Typically mid-level (parts of objects) or high level (objects) information that assists directly or indirectly with the end goal. This information has an equivalent linguistic label in the human brain, like car, hammer, tail light and so on. For instance, knowledge of the object being manipulated changes the probability priors of actions when it comes to activity recognition.
- Non-semantic Context: Low level information, such as texture, shape of super-pixel, statistics regarding color or location to aid in the end goal directly or end up as blocks that higher level context is constructed out of. For instance, presence of leaf or dirt like textures would help classify an image as outdoor vs indoor. Another example would be statistics of straight lines, which can be used to distinguish indoor/man-made scenes from natural scenes.

An alternative classification is based on whether the cues come from within the image or require an additional modality for capture.

- Extrinsic Context: Information such as GPS coordinates, camera orientation, focal length etc. that are captured by means beyond the image.
- Intrinsic Context: Information such as distribution of local features in the image, feature co-occurrence etc. which goes beyond location agnostic local descriptors and can be extracted from the image.

Extrinsic and Intrinsic context can both be Semantic or Non-semantic in nature.

3.2 Utilizing Contextual Cues

Given the diverse set of cues that may be assimilated into a descriptor, these are some of the ways one may proceed in:

3.2.1 Contextual Cues As Features

If the cues are extracted from spatially restricted regions (local cues), then they may be pooled into global cues much the same way as local descriptors are (See BoW and other encoding methods Section:2.2.2). This global cue/context may then be used as an image descriptor by itself or appended to other global descriptors of the image.
Example: Color histograms or texton histograms extracted from superpixels, that may directly be used as local descriptors and pooled through Bag-of-Visual-Words into a global descriptor. Else, color and texton histograms may directly be constructed for the entire image and appended with other global descriptors of the image [Hoiem et al., 2007]. Explicit feature co-occurrence may also be captured and used as feature directly, a move that is seen as going from Visual Words to Visual Phrases by considering feature pairs and triples [Zhang et al., 2011a] [Zhang et al., 2011b].
Shape Context, where edge elements, rather than keypoints are sampled from the shape and their relative spatial distribution is recorded by means of log-polar histograms centered at each element [Belongie et al., 2002]. This context information is then used to describe the elements, to aid in shape correspondence matching.

Much of these techniques retain geometric invariance either by using the local dominant gradient as the reference direction or by using direction agnostic pooling. See Figure 3.1(b).

3.2.2 Augment Local Features With Cues From The Spatial Neighbourhood

Context extracted from the spatial neighbourhood may simply be appended to the local feature. In BoW or frameworks that require clustering of the underlying feature space, an increase in local feature dimensionality due to appended context cues may be undesirable. An alternative is to use this information as the parameter of a function that transforms the local descriptor. We would see this in action in Section 5.2. Regardless of the mechanism employed, the objective is to make the local features more distinctive, either by introducing additional dimensions of differentiation or by mapping to some other space, using information from the local neighbourhood or the entire image plane.

Example: Using the super-pixel example again, one may decide to keep track of the number of superpixels in contact with each superpixel, or perhaps the ratio of area covered by each super-pixel vs the neighbouring super-pixels. Weak geometric information can be captured by augmenting or substituting local features with feature distribution statistics in the neighbourhood, such as sector wise co-occurrence statistics in the feature neighbourhood [Liu et al., 2012] [Liu et al., 2008]. This differs from Visual Phrases in that the co-occurrence relationship established here is statistical rather than concrete, utilizing histograms for co-occurrence counts or a weighted sum of descriptors in the neighbourhood. Descriptiveness of local features may also be improved by incorporating information about the neighbouring features into the encoding step [Gao et al., 2010]. There have also been attempts at learning the pooling regions in feature neighbourhood [Simonyan et al., 2014],as well as the use of the mean and variance of the spatial occurrence of features as context [Krapac et al., 2011].

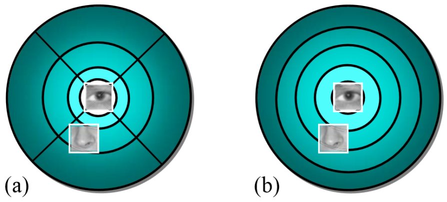
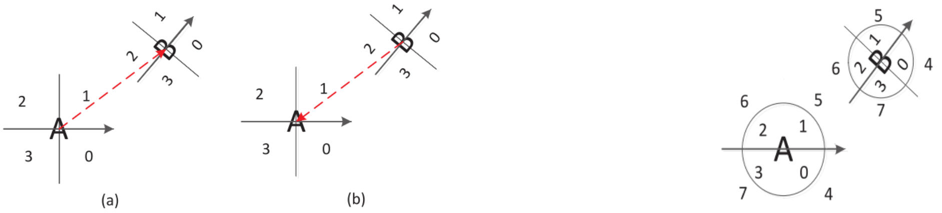


Figure 3.1: Spatial Histograms used to construct 2nd Order Features where for important feature code-word pairs (w_a, w_b), the average spatial distribution of w_b w.r.t w_a is captured, and vice versa [Liu et al., 2008]

This means of capturing statistical information about co-occurring features as well as keypoint centered partitioning of the image plane would be a recurring theme in this thesis.



(a) Feature Pairs

(b) Plane partition

Figure 3.2: Ideas for pairwise feature co-occurrence and the partitioning of image plane as the first step to statistical co-occurrence capture [Liu et al., 2012]

3.2.3 Global Feature Distribution Statistics

Similar to partitioning of local feature neighbourhood to pool contextual information, a partitioning induced on the entire image plane for global feature pooling can also encode information about the spatial distribution of features.
Example: Spatial Pyramid Kernels [Lazebnik, 2004] hierarchically partition the image plane into regular regions and do BoW pooling separately in each region. The BoW histograms from these regions are then appended together, with histograms from higher levels (smaller pooling regions) weighted more than those from lower levels.
PHOG or Pyramidal Histogram of Oriented Gradients is a variant of Spatial Pyramids, with learned relative weighting of descriptors from different levels, and used for shape representation [Bosch et al., 2007]. It uses the shape information as context and disambiguates similar shapes with appearance information.

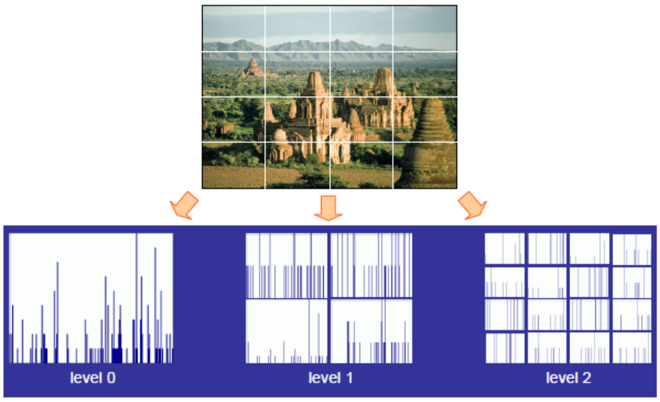


Figure 3.3: Spatial Pyramid Levels [Lazebnik, 2004]

The partitioning induced by Spatial Pyramids and derivatives is image axis oriented and leads to a loss of in-plane rotation invariance as well as a loss of invariance to large translations regardless of the invariant status of the constituent local descriptors.

3.2.4 Voting or Constraints in Post Processing

Instead of employing context to disambiguate features, one can instead use these extra cues to disambiguate, refine or validate the end results of the task. This is achieved by checking for the satisfaction of certain geometric constraints explicitly, or by having feature contexts vote for plausible hypotheses in a Generalized Hough Transform setting.

Example: In image classification and image retrieval, the veracity of putative classifications and matchings can be checked through geometric validation via RANSAC [Philbin et al., 2007] or neighbourhood feature consistency [Sivic and Zisserman, 2003]. For matching tasks, RANSAC may again be employed, or alternatives such as higher order graph matching [Duchenne et al., 2011] that incorporate strong feature co-occurrence beyond feature pairs and triplets.

3.2.5 Part Based Models

These are also based on Generalized Hough Transform but we distinguish these from the previous case because contextual verification here is a part of inference.

Example: Implicit Shape Models [Leibe et al., 2004] have the offset from object center associated with each code book entry for all objects. The detected parts use this associated geometric information to vote for object hypothesis in the image space based. Later work improves upon ISM by utilizing motion information to disambiguate between objects of different classes with similar appearance [Wang et al., 2011]. Constellation models explicitly encode relative location and relative scale information of parts, and both appearance and shape can be jointly learned though Expectation Maximization [Fergus et al., 2003].

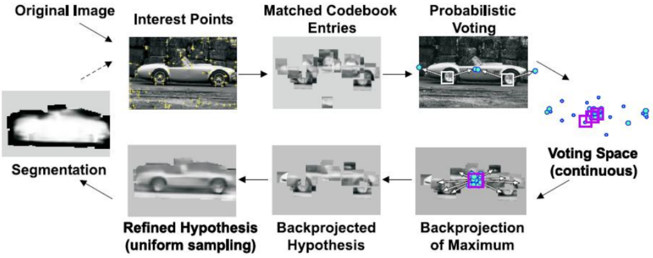


Figure 3.4: Closing the inference loop in Implicit Shape Models [Leibe et al., 2004]

Deformable Part Models [Felzenszwalb et al., 2008] [Felzenszwalb et al., 2010] use HOG pyramids to model parts, with root filters providing detections windows to constrain the search space of parts. Spatial constraints are imposed by jointly optimizing part matching and displacement of the parts from the corresponding roots, i. e., part appearance matching and displacement of parts from their corresponding reference locations in the model are jointly optimized.

3.3 Feature Hierarchy, Rotation Invariance and Geometric Context: A Case for Disctinctiveness Of Visual Words in BoW

From low level features to parts to objects: In section 2.2.1), we discussed the construction of global image representations from low level image descriptors. We also mentioned feature hierarchies where instead of directly constructing global image representations, one constructs progressively complex features out of simpler features such that a notion of (semantic or non-semantic) parts and objects emerges.

Why do we need for a notion of parts to emerge when global representations can just as easily be constructed out of lower level features?
To better understand this, consider the implications on separability of classes in feature space. Low level features are essentially textons or texture descriptors. They can be used to make a distinction between classes with vastly differing texton statistics such as outdoor vs indoor scenes. However with textons one can only encode texture statistics in the image, and perhaps some structure through Spatial Pyramids. However for classes that may be similar in terms of texture statistics, such as an office scene and a living room, one would need more descriptive statistics to be able to make a distinction. Encoding these features in a BoW framework with an intersection kernel can be viewed as comparison of the occurrence frequency of different types of textons.

The notion of parts is naturally of import in image matching and homography estimation applications wherein pairwise correspondences between two images need to be established. Low level descriptors, by virtue of their low descriptive power are ill suited for these applications because they would lead to far too many putative correspondences. Image classification using *parts* as local features, particularly in a BoW framework with similarity defined by histogram intersection, can then be seen as an approximation to image matching [Jegou et al., 2008]. Here intersections per bin can be seen tending to an explicit correspondence relationship as the distinctiveness of the codewords increases, and away from the occurrence frequency interpretation. A natural outcome of increasing discriminativeness, seeing from a matching perspective, would be feature sparsity because we expect different classes or class subsets to have at least a few unique and distinguished parts.

There is a limit to the distinctiveness achievable in going up the feature hierarchy though. As one starts approaching the notion of objects, if one has used strong geometric relationships between constituents, one tends to run into problems with object deformation and out of plane object rotations. If one has not used any geometric context, one risks false matches between objects with similar appearance statistics but different geometric arrangements and suffers a loss in distinctiveness. This sweet spot of how far one can take geometric relationship information is directly result of the inherent structure in the images, and hence is dependent on application.

In the previous section we have seen ways of strongly and weakly capturing geometric context. In the interest of discriminativeness and invariance to local deformations and in-plane rotations, it would be prudent to strongly capture geometric relations between low level descriptors (that are in-plane rotation invariant) to come to the abstraction level of parts, and then construct weak geometric context at the part level to get to the abstraction level of objects, taking care to encode geometric relations in a manner invariant to in-plane rotations.

3.4 Aspects Of Context Explored In This Work

In this thesis, we uncover the global context inherently captured in Dense SIFT. (Chapter 4). Our primary focus would be making local features more descriptive and distinctive by incorporating this contextual information from feature neighbourhoods. Scaling and translation invariance would be achieved by means of multi-scale keypoints, with feature neighbourhoods described as a multiple of $log(keypoint\, scale)$. The secondary objective is rotational invariance, achieved by aligning the local coordinate system along the dominant gradient when extracting context in feature neighbourhoods. (Chapter 5)

Rather than simply append contextual information to keypoint appearance dimensions and run into *Curse of Dimensionality*, we would use a 2D histogram approach that clusters appearance dimensions and context dimensions independently in their corresponding lower dimensional spaces. (Sections 5.1.1, 6.1). We would also look at incorporation of contextual information through modification to the appearance descriptor (Section 5.2).

We would also look at weak feature co-occurrence relationships as context (Chapter 6), and make use of image plane partitioning schemes popular in prior art to add to the discriminativeness of contextual information (Section 5.3).

In Chapter 7 we incorporate relative orientation contextual information into the CNN pipeline.

Context In Deep Neural Networks

We will touch upon prior work on context capture in Deep Learning in Chapter 7 Section 7.1.2.

Chapter 4

Detour: Why does Dense SIFT work so well?

In this chapter we examine the ”secret sauce” behind the efficacy of *Dense SIFT* over *Keypoint SIFT* at image classification and retrieval tasks. Recall from Chapter 2 that Dense SIFT, in its most basic version, proceeds by regularly sampling SIFT descriptors from the image at a particular scale, with a uniform reference direction for the SIFT descriptors (pointing up). Keypoint SIFT, however, seeks to extract SIFT descriptors from *keypoints*, i. e., locations in an image which can be repeatedly and reliably identified and localized in an image. Keypoints often have scale information attached [Lindeberg, 2015] , which is used to define the size of the patch used to construct the SIFT descriptor at that particular location, with the dominant gradient at that patch dictating the reference direction. See Figure 4.1.

In This Chapter

- Postulate that Dense SIFT descriptors encode more than the local gradients
- Examine the evidence that this extra information contributes to the discriminativeness of the overall descriptor (Section 4.1)
- Argue for explicit capture of this extra information (Section 4.3)

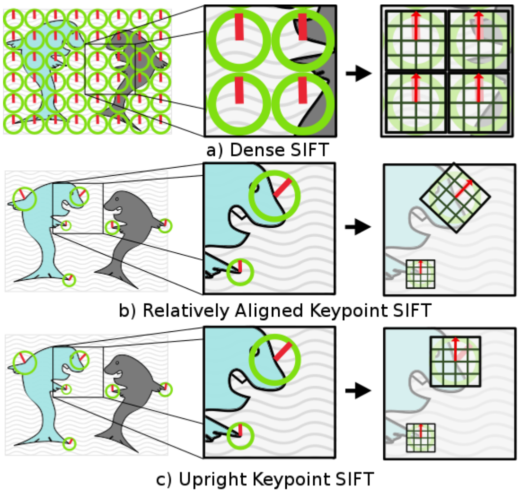


Figure 4.1: Visualization of SIFT patches for Dense SIFT and Keypoint SIFT

Dense SIFT, as its name suggests, manages to sample a much larger fraction of the image with all its local descriptors combined. Keypoint SIFT, on the other hand, covers a far smaller portion of the image. *Do the larger number of samples explain the success of Dense SIFT? Is that all that there is to it? What does this have to do with context?*

In the next section we factor out the differences in sampling between the two approaches to make an apples-to-apples comparison. We'll see that Dense SIFT fares better, in part due the contextual information captured intrinsically as a result of the mechanics the SIFT extraction process.

4.1 Factoring Out the Number of Samples and Scale Information

To keep the comparison fair, consider two variants of Keypoint SIFT: the traditional, dominant gradient oriented Keypoint SIFT (interchangeably called Relatively Aligned Keypoint SIFT), and dominant gradient agnostic Keypoint SIFT with the reference direction pointing along image **y** axis (interchangeably called Upright Keypoint SIFT). They differ only in the orientations of the SIFT patches.

Let us pick one of the spatial bins close to the center of the SIFT patch and observe the orientation histogram coming from this sub-patch. Assume that orientation binning proceeds from 0 tp 2π. Relatively aligned SIFT, on account of being aligned along the dominant gradient at that patch location, would have a predictable pattern to the orientation histogram coming from the chosen spatial bin, with a peak in the first bin, and occasionally in the last bin, and tapering out towards the middle. Upright Keypoint SIFT, however, can have the peak in any bin. Refer to Figure 4.2.

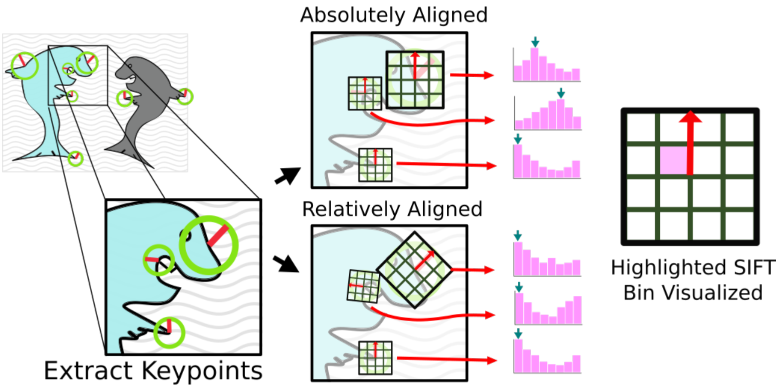


Figure 4.2: Mechanism of capture of relative orientation in Upright SIFT

Ruminating over this for a moment, we see that given the Upright SIFT descriptors from two patches, we can at once point out the absolute orientations of the two and can deduce the relative orientations between the two. *This additional information remains the only contributing factor to the increased classification accuracy of Upright Keypoint SIFT over Relatively Aligned Keypoint SIFT.*

Supporting the above claim, figure 4.3 shows the classification accuracies attained from multi-class (OVO) SVM classification on soft asssignemnt Bag-of-Visual-Word descriptors constructed from the Upright and Relatively Aligned variants of SIFT on the Fifteen Scene Categories dataset. Refer to Section 8.2 for implementation details. The classification accuracies are plotted for SIFT dictionaries of various sizes. Multi-scale keypoints are extracted with Difference-of-Gaussians Laplacian Pyramids, and only keypoints with a scale > 1.0 considered, yielding 250 keypoints per image.

4.2 Quantifying the Effect of Number of Samples

We consider the impact of the number of samples by lowering the scale cut off for keypoints to allow more keypoints to be taken into consideration for both SIFT variants (≈ 400/image, up from ≈ 250 in the previous case). It is obvious from Figure 4.3 that the number of samples too have a role to play in the discriminativeness of the descriptor. Yet there is need to be wary of the increased number of samples, not only for the added cost of processing the descriptors, but also for the potential of these small scale keypoints to really be inconsequential clutter that ends up harming the overall discriminativeness. We would come across this in Chapter 6.

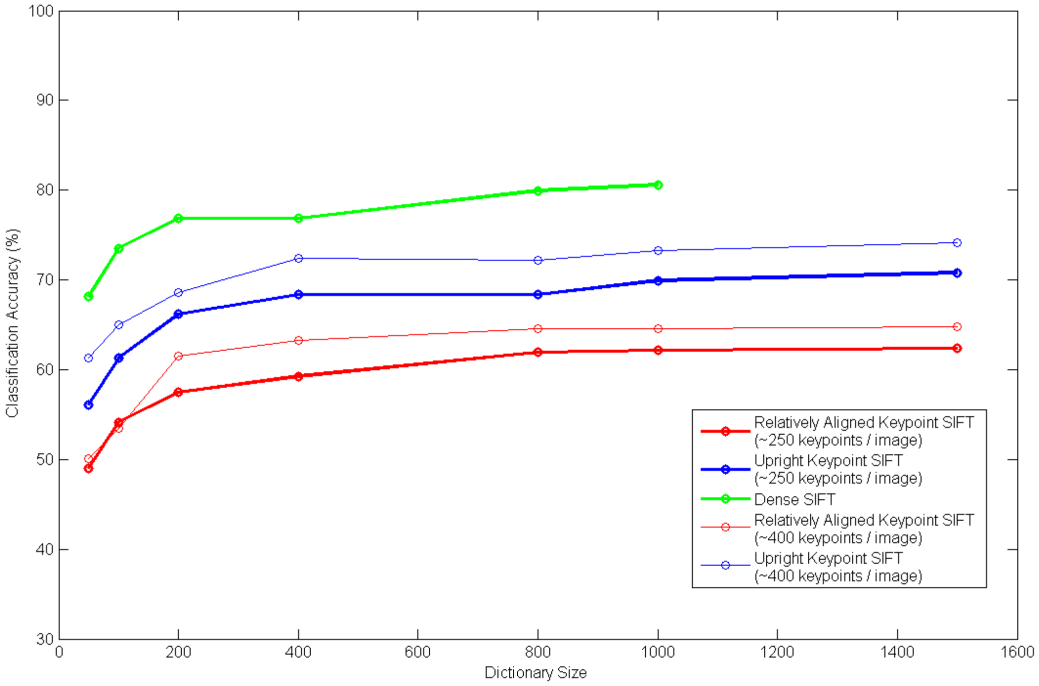


Figure 4.3: Classification accuracy of BoW of various SIFT descriptors on Fifteen Scene Categories dataset

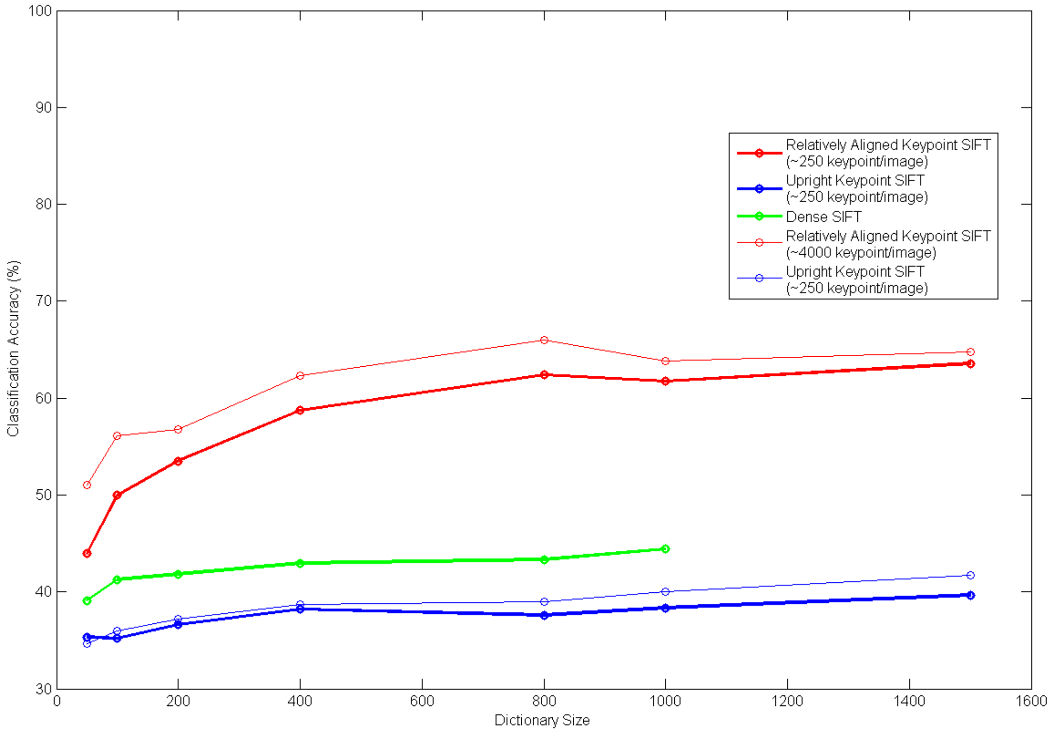


Figure 4.4: Classification accuracy of BoW of various SIFT descriptors on Fifteen Scenes Rot dataset. See 4.3

4.3 Discriminativeness and Geometric Invariance Redux

The compromise between *discriminativeness* and *geometric invariance* in the case of Upright (Dense and Keypoint) SIFT and Relatively Aligned SIFT can be intuited from the description of the approaches earlier in Section 4.1. To make it more concrete, we look at the classification accuracies of these approaches on a variant of the Fifteen Scene Categories dataset. This dataset variant is specifically meant to demonstrate susceptibility to in-plane rotations. Briefly, un-rotated images of the scenes are used for training. At test time, the system sees previously unseen un-rotated images and rotated copies of both seen and unseen images. Refer to 2.3 for in depth discussion of the rationale, and other possible offshoots of the dataset.

Figure 4.4 shows that while Relatively Aligned Keypoint SIFT maintains its classification accuracy, it is clear that Upright SIFT is in-adept at handling in-plane rotations, trading it for increased discriminativeness in the absence of rotations.
Is this trade off unchanging and fundamental? Must we always pay the price of one for the other?

From Figure 4.2 we can see that the distinctiveness of the local SIFT descriptors across the image in case of Upright SIFT would make the clustering easier and less error prone. On the other hand, the similarity in the profiles of Relatively Aligned SIFT descriptors makes it harder to reliably set up distinct clusters. This already hints that the next logical step should be to explicitly capture Relative Orientation context in a way that makes the local descriptors more discriminative, yielding possibly more discriminative global descriptors post Bag-of-Words. We follow up on this in Chapter 5.

Chapter 5

Intentionally Capturing Relative Orientation Context

Having uncovered the serendipitous contribution of **Relative Orientation** information to the discriminativeness of Dense SIFT descriptors in Chapter 4, and noting the absence of in-plane rotation invariance in *Upright SIFT* (Dense SIFT and Upright Keypoint SIFT), a natural follow up question is: *Can such Relative Orientation be captured explicitly, to boost the discriminativeness of the descriptors while preserving the in-plane rotation invariance associated with Relatively Aligned Keypoint SIFT?*.

In This Chapter

- Use of Keypoint Relative Orientation histograms constructed in the neighbourhoods of all keypoints as context (Section 5.1)
- Orthogonal clustering of appearance (SIFT) and context to yield 2D Histograms (Section 5.1.1)
- Incorporation of Relative Orientation back directly into SIFT without requiring orthogonal clustering (Section 5.2)
- Considering context from keypoint neighbourhoods on a directional basis, with an accompanying feature selection method to reduce the impact of clutter (Section 5.3)

5.1 Explicitly Encoding Relative Orientation Context: First Steps

Pondering over the Relative Orientation context captured implicitly in Upright SIFT, one notices that each local descriptor incorporates the local gradients pooled in the SIFT patch and additional contextual information pooled at the image level. A naive attempt to build on this intuition to explicitly capture relative orientation context is shown in Figure 5.1. For each keypoint, a certain circular neighbourhood is defined, centered at the keypoint, with radius proportional to the *log* scale of the keypoint. A normalized histogram of size 8 bins the relative orientations of keypoints occurring in the neighbourhood with respect to the central keypoint.

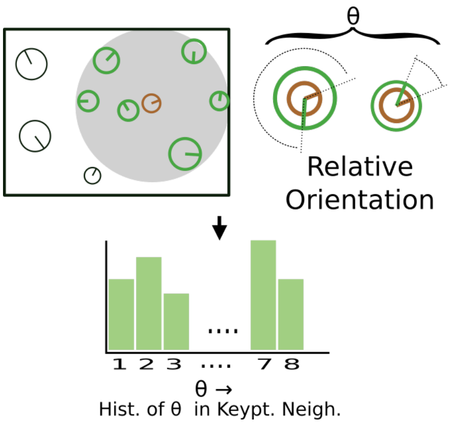
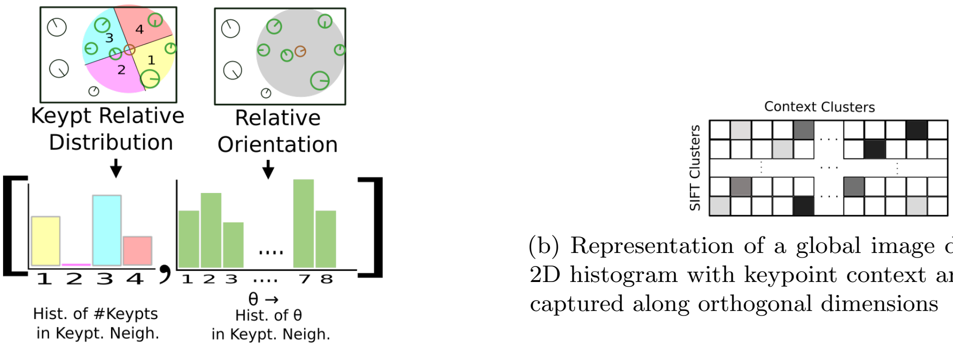


Figure 5.1: Relative Orientation Histogram captured in keypoint neighbourhood

Additional statistics from the neighbourhood may also be captured and appended to the *Relative Orientation* histogram. Figure 5.2a shows a precursor to directional neighbourhood pooling capturing keypoint density in each sector of each keypoint neighbourhood.


1
2
3
4
Keypt Relative Distribution
Relative Orientation
1 2 3 4
1 2 3 .... 7 8
θ →
Hist. of #Keypts in Keypt. Neigh.
Hist. of θ in Keypt. Neigh.
Context Clusters
SIFT Clusters

(a) Additional statistics captured in keypoint neighbourhood

(b) Representation of a global image descriptor as a 2D histogram with keypoint context and appearance captured along orthogonal dimensions

Figure 5.2: Representation of the context information that is clustered orthogonal to the central keypoint's appearance information (SIFT Clusters), resulting in the 2D histogram shown in (b)

5.1.1 Orthogonal Clustering of Appearance and Context

One could take the context histograms described in the section before and append them to the 128 dimensional SIFT descriptor and cluster these 140 (128 + 8 + 4) dimensional descriptors with k-means to obtain a (soft assignment) Bag-of-Words representation. In practice, we found that the clustering was unsatisfactory, with the 128 appearance dimensions dominating the result regardless of the relative weighting of the appearance and context dimensions. We tried 1:1, 1:10, 1:30 relative weight ratios for appearance and context in the construction of the local descriptor and none yielded satisfactory clustering.

To remedy this, we choose to cluster keypoint appearance (SIFT) and Relative Orientation context independently, and create a soft assignment 2-Dimensional histogram with appearance bins and context bins along orthogonal directions. See Figure 5.2b. This approach, in general, alleviates the adverse effects of curse of dimensionality by clustering appearance and context in their respective lower dimensions. It can be viewed as a hierarchical clustering that takes away the onus of feature distinctiveness from fine clustering in appearance space, to somewhat coarser clusters in appearance space further refined by clustering along context within appearance clusters, with the additional advantage of reduced running time.

Table 5.1 shows classification accuracy trends for Orthogonal capture of Relative Orientation for various Context Dictionary sizes on both datasets. The impact of Context Dictionary size appears to be minimal for both versions of the dataset. However, the impact of the number of keypoints is a bit more interesting. Increasing the number of keypoints sees a consistent but unsubstantial improvement for the first dataset. This contrasts with the marked increase in classification accuracy seen in Relatively Aligned and Upright SIFT on increasing the number of keypoints. On the dataset to test in-plane rotational invariance, *the trend reverses* and an increase in the number of keypoints sees a performance hit.

	Fifteen Scene Categories						Fifteen Scenes Rot					
	$\approx 250\ keypts/img$			$\approx 400\ keypts/img$			$\approx 250\ keypts/img$			$\approx 400\ keypts/img$		
SIFT	Context Dictionary			Context Dictionary			Context Dictionary			Context Dictionary		
Dictionary	10	20	40	10	20	40	10	20	40	10	20	40
50	59.7	60.8	64.4	63.9	65.3	66.8	55.0	60.3	63.3	57.0	60.5	63.3
100	62.8	63.9	65.0	66.8	68.2	65.5	62.0	65.7	66.9	60.1	62.0	64.1
200	65.3	65.0	66.6	64.8	66.8	66.6	63.7	66.9	69.2	64.9	64.9	65.6
400	64.8	67.9	66.8	67.3	68.2	67.0	65.6	69.8	70.4	65.1	65.9	66.1
800	66.4	67.0	67.0	67.7	68.4	67.9	67.7	70.4	70.9	66.9	66.6	66.0
1000	66.1	67.0	66.8	67.0	68.8	68.4	67.3	70.2	70.6	66.5	66.6	66.3

Table 5.1: Classification Accuracy(%) for Orthogonally Clustered Relative Orientation Context on **Fifteen Scene Categories Dataset** and **Fifteen Scenes Rot Dataset**, With Different SIFT Dictionary Sizes, Different Context Dictionary Sizes and Different Number of Keypoints Per Image.
Neighbourhood Size = $80 \times log(scale)$
(Note that Context Dictionary Size is different from the number of bins in the Relative Orientation histogram)

Figure 5.3 (green) shows that Orthogonal encoding of Relative Orientation context with a *Context Dictionary size of 20* performs better that Relatively Aligned SIFT, but is still below Upright Keypoint SIFT on the Fifteen Scene Categories Dataset.

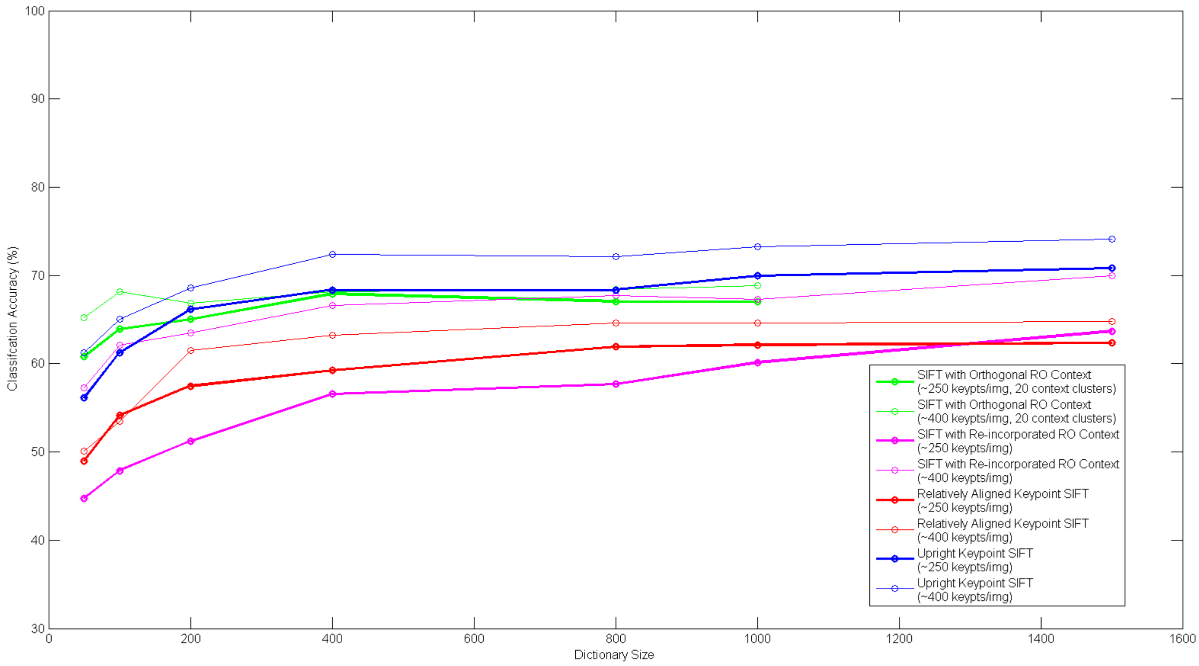


Figure 5.3: Classification Accuracy for SIFT BoW with Relative Orientation Context for the Fifteen Scene Categories Dataset. Context is extracted from neighbourhoods of size $80 \times log(scale)$

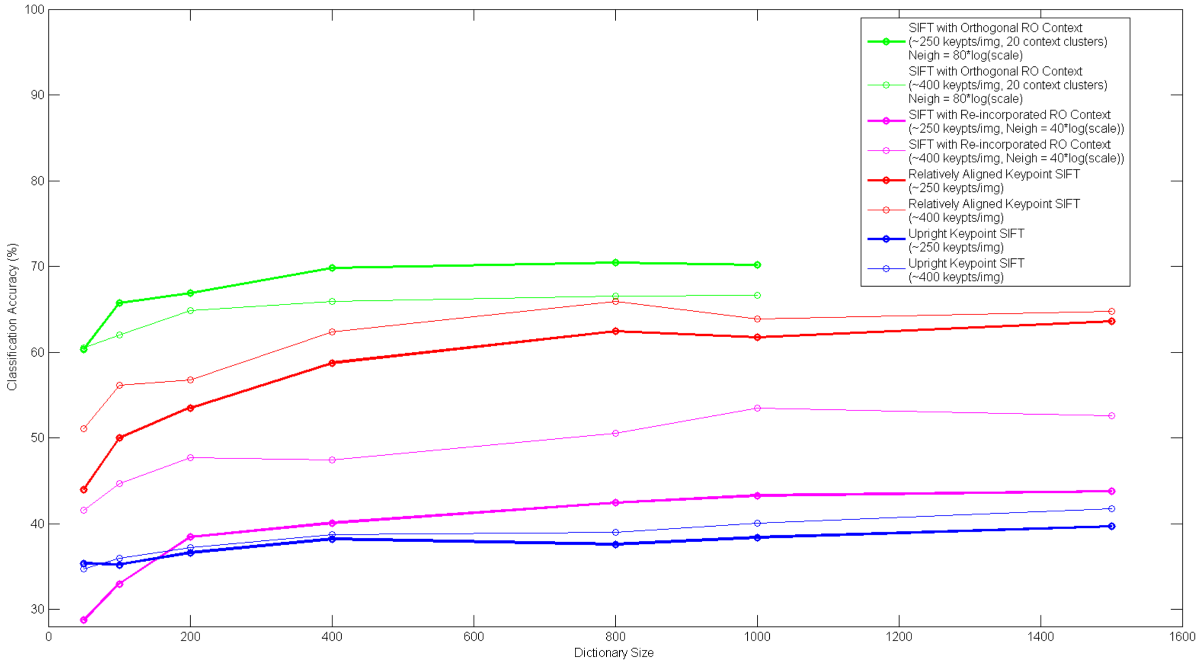


Figure 5.4: Classification Accuracy for SIFT BoW with Relative Orientation Context for Fifteen Scenes Rot Dataset. Note the difference is keypoint neighbourhood sizes between Orthogonally clustered Relative Orientation and Re-incorporated Relative Orientation.

Figure 5.4 (green) shows that on the dataset designed to test rotational invariance, the Orthogonal capture of Relative Orientation with a *Context Dictionary size of 20* performs significantly better than Relatively Aligned Keypoint SIFT.

5.2 Re-incorporating Relative Orientation Information into SIFT

Having seen the promise of explicitly capturing relative orientation information in the section above, and the fact that Upright SIFT implicitly captured similar information, it is natural to question whether clustering context in an orthogonal direction is absolutely necessary. Given that our appearance Bag-of-Words size gets multiplied by the dictionary size of context clustering, it may prove prudent to try and re-incorporate this information into SIFT itself.

As mentioned previously, simply appending the context histogram to the SIFT descriptor of the central keypoint proved ineffective. We propose an alternate scheme that uses the Relative Orientation Context histogram to compute an average relative orientation of keypoints per keypoint neighbourhood and use it to modify the orientation of the central keypoint. See Figure 5.5 for a visual explanation.

There are two possible ways of implementing this. The first is to simply average the relative orientations of the neighbouring keypoints (and not the bin representatives). Since angles are a circular quantity, the right approach would be to use complex numbers to compute the mean. See equation below.

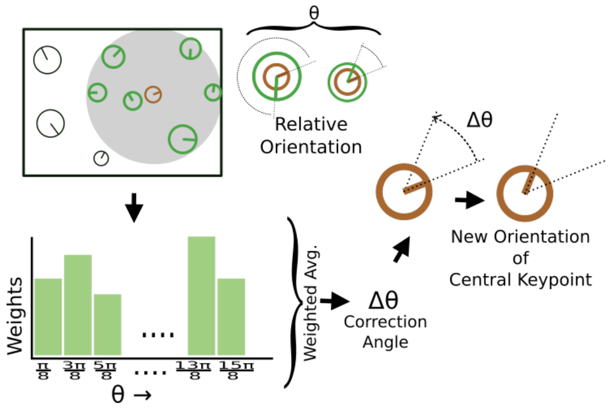


Figure 5.5: Scheme for re-incorporation of Relative Orientation into SIFT

$$\bar{\alpha} = atan2(\sum_{j=1}^{n} sin\alpha_j, \sum_{j=1}^{n} cos\alpha_j) \tag{5.1}$$

Naive averaging of relative orientations can't be expected to be rotationally invariant because the relative orientation angle values are conditional on the absolute angles, while with circular quantities handled properly it would be rotationally invariant. In practice however, we find that the version with proper handling of circular quantities fares worse than even Relatively Aligned SIFT, prompting us to pick the former despite its sensitivity to in-plane rotations.

Tables 5.2 and 5.3 make concrete why despite the resilience of the version with correctly handled circular quantities to in-plane rotation, there is no incentive to pick it over vanilla Relatively Aligned Keypoint SIFT. We also observe that for both versions, increasing the number of keypoints has a substantial impact on performance. The trends with respect to the size of the neighbourhood however are harder to pin down, but are mostly insignificant.

We see in Figure 5.3 (magenta) that re-incorporation of Relative Orientation context with fewer (≈ 250) keypoints performs about the same as Relatively Aligned Keypoint SIFT. Upon increasing the number of keypoints (≈ 400), a large increase in classification accuracy is seen, matching the performance of Orthogonal capture of Relative Orientation. It performs better than Relatively Aligned Keypoint SIFT, and uses far fewer dimensions than Orthogonally clustered Relative Orientation.

In Figure 5.4 (magenta) however, we see that this approach lags behind Relatively Aligned Keypoint SIFT substantially for the rotated version of Fifteen Scene Categories dataset for reasons discussed earlier. The trend seen earlier with regards to the number of keypoints holds here.

	Fifteen Scene Categories				Fifteen Scenes Rot			
	≈ 250 $keypts/img$		≈ 400 $keypts/img$		≈ 250 $keypts/img$		≈ 400 $keypts/img$	
SIFT	Neighbourhood Size $x \times log(scale)$				Neighbourhood Size $x \times log(scale)$			
Dictionary	x =40	x =80	x =40	x =80	x =40	x =80	x =40	x =80
50	35.7	34.7	52.1	47.4	41.4	37.9	45.8	43.3
100	42.4	38.8	53.5	54.8	48.8	45.0	51.2	49.3
200	42.4	43.0	56.6	57.5	49.0	48.6	54.0	52.1
400	49.8	49.2	59.0	59.7	52.1	53.0	56.7	54.0
600	51.3	48.5	62.1	60.4	57.0	51.0	56.9	55.6
800	53.2	51.1	62.1	60.4	57.7	50.3	58.0	57.3
1000	52.6	52.0	63.9	60.4	56.6	56.1	58.6	56.9
1500	55.6	53.1	62.8	60.4	56.3	54.3	60.6	57.5

Table 5.2: Classification Accuracy(%) for **Angle Range Restricted** Re-incorporated Clustered Relative Orientation Context on **Fifteen Scene Categories Dataset** and **Fifteen Scenes Rot Dataset**, With Different SIFT Dictionary Sizes, Different Neighbourhood Sizes and Different Number of Keypoints Per Image.

	Fifteen Scene Categories				Fifteen Scenes Rot			
	≈ 250 $keypts/img$		≈ 400 $keypts/img$		≈ 250 $keypts/img$		≈ 400 $keypts/img$	
SIFT	Neighbourhood Size $x \times log(scale)$				Neighbourhood Size $x \times log(scale)$			
Dictionary	x =40	x =80	x =40	x =80	x =40	x =80	x =40	x =80
50	37.0	44.8	55.7	57.2	28.7	29.1	41.6	37.9
100	43.7	47.9	59.0	62.1	33.0	32.1	44.7	42.0
200	51.7	51.2	62.4	63.5	38.4	37.0	47.7	42.7
400	58.4	56.6	65.7	66.6	40.0	39.3	47.4	48.2
600	58.6	54.3	63.9	66.6	41.7	41.4	50.2	47.3
800	57.2	57.7	65.9	67.7	42.4	40.4	50.6	48.8
1000	60.6	60.1	64.1	67.3	43.3	41.4	53.5	49.3
1500	63.0	63.7	66.4	69.9	43.7	42.0	52.6	51.0

Table 5.3: Classification Accuracy(%) for Re-incorporated Clustered Relative Orientation Context on **Fifteen Scene Categories Dataset** and **Fifteen Scene Rot Dataset**, With Different SIFT Dictionary Sizes, Different Neighbourhood Sizes and Different Number of Keypoints Per Image.

5.3 Directional Context Capture

Going off the road from thinking about the information that is captured as context, we pause for a moment to consider where the context is captured from. It stands to reason that not the complete circular neighbourhood of a keypoint would be a reliable source of context, and some regions of the neighbourhood may in fact be contributing spurious information that hampers discriminativeness of the descriptor.

To examine the effect of keypoint neighbourhood on descriptor discriminativeness, we propose two schemes to capture context from certain regions of the neighbourhood. We investigate this in the setting of the two Relative Orientation context schema described earlier in the chapter, i. e., Orthogonally Clustered and Re-incorporated context. The first method captures context one quadrant at a time from the neighbourhood, with the quadrant boundaries parallel and perpendicular to the dominant gradient direction at the keypoint. We suspected that the area covered by quadrants might be too small, so we consider a second approach where context is captured in pairs of adjacent quadrants (halves). See Figure 5.6 for details.

The context captured from all four regions is taken into consideration in isolation, and four sets of features are constructed as per the Relative Orientation encoding schema described earlier. This results in 4× the number of features as before, necessitating some means of feature selection to remove some feature+context combinations as per the premise that context from not all regions may be useful.

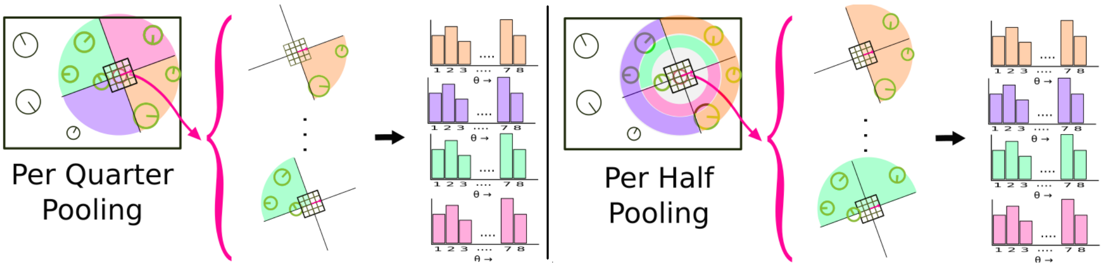


Figure 5.6: Representation of Per-quarter and Per-half neighbourhood definition for extraction of context. Features resulting from each of the above defined neighbourhood segments are clustered and pooled independently and appended together to yield the final representation. Feature selection is then applied to get rid of non-useful features.

5.3.1 Discriminative Feature Selection

Before we look at the results of directional context in detail, we'll discuss and develop feature selection strategies using the setting of *re-incorporated Relative Orientation* context as example. These would be later be applied every where context is captured directionally. With feature selection, the objective is to select a subset of the features such as to improve the generalization by lowering the propensity of the model to overfit. Feature selection strategies abound in literature, ranging from decision tree based [Deng and Runger, 2012] to heuristic based methods [Figueroa, 2015] [Kraskov et al., 2003], with the aim of reducing redundancy or correlation of features.

Our aim diverges a bit from typical feature selection, since we seek to remove spurious features from the collective rather than the redundant ones. One can then either choose to down weigh the effect of these spurious features through a method such as tf-idf weighting [Manning et al., 2008] [Sparck Jones, 1972], or look for discriminative features per classification category and use the ensemble of those. We propose a simple approach to do the latter, inspired by Discriminative Feature Extraction [Doersch et al., 2013].

Let ρ_+ be the feature space density of a class and ρ_- be the feature space density of the complementary classes. Mid-Level Feature Extraction through Discriminative Mode Seeking [Doersch et al., 2013] operates through mode seeking on the density ratio ρ_+/ρ_- rather than on ρ_+, resulting in visual elements that belong in one class and not in the others. We consider an approximation of this by evaluating these ratios on code-word density in clustered feature space.

Let H be the BoW histogram of size D. Let individual bin counts of H be given by h_i, $i \in 1..D$.
Let $\bar{H}^C$ be the mean of BoW histograms of class C. Let $\bar{H}^{\bar{C}}$ be the mean of mean BoW histograms of all classes except C.

$$\bar{H}^{\bar{C}} = mean_{\bar{C}} \left\{ \bar{H}^{\mathbf{c}} : \mathbf{c} \in \bar{C} \right\} \tag{5.2}$$

Then, one possible way of defining a discriminative subset of h_is is,

$$F_1 := \left\{ h_j : j \in 1..D, \exists C : \frac{\bar{h}_j^C}{\bar{h}_j^{\bar{C}}} > 1.0 \right\} \tag{5.3}$$

One can make the selected features be more distinctive by constructing $\hat{H}^{\bar{C}}$ with components as below:

$$\hat{h}_j^{\bar{C}} := \max_{\mathbf{c} \in \bar{C}} \left\{ \bar{h}_j^{\mathbf{c}} \right\}, \, j \in 1..D \tag{5.4}$$

and select features as

$$F_2 := \left\{ h_j : j \in 1..D, \exists C : \frac{\bar{h}_j^C}{\hat{h}_j^{\bar{C}}} > 1.0 \right\} \tag{5.5}$$

In practice we see that F_1 picks up $\approx 68\%$ of the histogram bins, whereas F_2 picks up $\approx 55\%$ of the histogram bins. Table 5.4 compares F_1 and F_2 with the results obtained with tf-idf weighting.

We can see that tf-idf is not adept at weighing down the spurious features and ends up compromising performance. F_1 ends up performing slightly better than the case with all features, so there is some degree of clutter rejection that is occurring. F_2 ends up picking far too few useful features in the quest for distinctive features, and negatively impacts performance.

	≈ 400 *keypts/img*							
	Neigbourhood Size 80 × $log(scale)$							
SIFT	All Feat.		tf-idf		F_1		F_2	
Dictionary	Qua.	Half	Qua.	Half	Qua.	Half	Qua.	Half
200	65.03	66.37	63.0	62.6	66.6	66.8	60.1	62.8
400	68.6	68.37	62.6	61.0	70.6	69.0	64.1	63.9
800	69.71	69.93	60.8	60.1	69.3	71.5	66.1	67.0
1000	69.27	71.49	60.4	61.9	70.2	72.4	67.9	67.0

Table 5.4: Classification Accuracy(%) for directional pooling of Relative Orientation that is **Re-incorporated** into SIFT orientation for **Fifteen Scene Categories Dataset**, with Different SIFT Dictionary Sizes, Neighbourhood Size of 80 × $log(scale)$ with various Feature Selection schemes.

We will use F_1 as our feature selection scheme for all the directional context results that follow.

5.3.2 Directional Context Results

Table 5.5 shows the result of directionally pooled Orthogonally clustered Relative Orientation context on Fifteen Scene Categories dataset. We see that the results are about the same for both the directional context pooling configurations and don't diverge much from the no-directional pooling approach. The performance is better than Relatively Aligned SIFT, but lags behind Upright Keypoint SIFT. Changing the number of keypoints doesn't have an appreciable impact.

	No Directional Pooling				Per Quarter Pooling				Per Half Pooling			
	≈ 250 *keypts*		≈ 400 *keypts*		≈ 250 *keypts*		≈ 400 *keypts*		≈ 250 *keypts*		≈ 400 *keypts*	
SIFT	Contxt Dict.		Contxt Dict.		Contxt Dict.		Contxt Dict.		Contxt Dict.		Contxt Dict.	
Dict.	10	20	10	20	10	20	10	20	10	20	10	20
100	65.3	65.0	64.8	66.8	66.6	65.5	63.3	67.0	66.1	67.5	65.9	67.7
200	64.8	67.9	67.3	68.2	65.9	66.4	65.7	67.3	65.9	67.5	65.3	67.5
400	66.4	67.0	67.7	68.4	68.4	67.7	66.1	66.4	67.0	67.7	68.6	68.2
800	66.1	67.0	67.0	68.8	66.8	66.1	66.8	65.9	66.1	66.8	67.3	68.4

Table 5.5: Classification Accuracy(%) for directional pooling of Relative Orientation Context that is **Orthogonally Clustered** for **Fifteen Scene Categories Dataset**, With Different SIFT Dictionary Sizes, Different Context Dictionary Sizes and Different Number of Keypoints Per Image.
Neighbourhood Size = 80 × $log(scale)$
(Note that the Context Dictionary Size is different from the number of bins in the Relative Orientation histogram)

Looking at the performance for the rotated version of Fifteen Scene Categories dataset in Table 5.6, we notice that directional context pooling does improve performance over the no-directional context pooling case. We also notice that *Per-Half* pooling does marginally better than *Per-Quarter* pooling, and increasing the number of keypoints has a slight negative impact. This slight negative impact may be due to the spurious nature of the smaller scale keypoints admitted into consideration. It nevertheless demonstrates that it is more discriminative than Relative Aligned SIFT without compromising on rotational invariance.

Table 5.7 shows the result of directionally pooled Relative Orientation context that is re-incorporated into SIFT, again evaluated on Fifteen Scene Categories dataset. We see substantial gains in classification accuracy brought along by directionally pooled context, in stark contrast to the case without directional pooling for fewer keypoints ($\approx 250/img$). There is some increase for the case with more keypoints as well. We also see that *Per-Half* pooling of context does substantially better than *Per-Quarter* pooling, with a larger neighbourhood size proving beneficial too. Re-incorporated Relative Orientation context is able to outperform Orthogonally clustered Relative Orientation.

	No Directional Pooling				Per Quarter Pooling				Per Half Pooling			
	≈ 250 *keypts*		≈ 400 *keypts*		≈ 250 *keypts*		≈ 400 *keypts*		≈ 250 *keypts*		≈ 400 *keypts*	
SIFT	Contxt Dict.		Contxt Dict.		Contxt Dict.		Contxt Dict.		Contxt Dict.		Contxt Dict.	
Dict.	10	20	10	20	10	20	10	20	10	20	10	20
100	62.0	65.7	60.1	62.0	63.4	65.6	62.0	65.3	62.0	64.7	60.4	64.0
200	63.7	66.9	64.9	64.9	64.2	66.5	64.5	66.5	63.7	64.8	64.6	66.9
400	65.6	69.8	65.1	65.9	66.3	67.6	66.6	67.4	66.0	67.6	66.3	68.3
800	67.7	70.4	66.9	66.6	69.5	69.8	68.3	68.6	69.6	71.2	67.1	68.6

Table 5.6: Classification Accuracy(%) for directional pooling of Relative Orientation Context that is **Orthogonally Clustered** for **Fifteen Scenes Rot Dataset**, With Different SIFT Dictionary Sizes, Different Context Dictionary Sizes and Different Number of Keypoints Per Image.
Neighbourhood Size = $80 \times log(scale)$
(Note that the Context Dictionary Size is different from the number of bins in the Relative Orientation histogram)

	No Directional Pooling				Per Quarter Pooling				Per Half Pooling			
	≈ 250 *keypts*		≈ 400 *keypts*		≈ 250 *keypts*		≈ 400 *keypts*		≈ 250 *keypts*		≈ 400 *keypts*	
SIFT	Neigh. Size $\mathbf{x} \times log(scale)$				Neigh. Size $\mathbf{x} \times log(scale)$				Neigh. Size $\mathbf{x} \times log(scale)$			
Dict.	x =40	x =80	x =40	x =80	x =40	x =80	x =40	x =80	x =40	x =80	x =40	x =80
200	51.7	51.2	62.4	63.5	65.0	65.9	65.9	66.6	65.9	66.1	66.8	66.8
400	58.4	56.6	65.7	66.6	68.8	67.7	68.4	70.6	70.2	69.5	68.6	69.0
800	57.2	57.7	65.9	67.7	69.7	67.9	67.3	69.3	68.4	70.8	68.8	71.5
1000	60.6	60.1	64.1	67.3	65.3	69.9	69.5	70.2	69.3	72.4	69.9	72.4

Table 5.7: Classification Accuracy(%) for directional pooling of Relative Orientation that is **Re-incorporated** into SIFT orientation for **Fifteen Scene Categories Dataset**, With Different SIFT Dictionary Sizes, Different Neighbourhood Sizes and Different Number of Keypoints Per Image.

On the rotated version of the Fifteen Scenes dataset, the story is a bit different though. Table 5.8 shows that although directional pooling of context is indeed significantly beneficial, *Per-Quarter* pooling with more keypoints ≈ 400 outperforms *Per-Half* pooling. Also, smaller neighbourhood sizes fare better here. Though the lack of rotational invariance makes it lose out to Orthogonally clustered Relative Orientation by a fair margin.

	No Directional Pooling				Per Quarter Pooling				Per Half Pooling			
	≈ 250 *keypts*		≈ 400 *keypts*		≈ 250 *keypts*		≈ 400 *keypts*		≈ 250 *keypts*		≈ 400 *keypts*	
SIFT	Neigh. Size $x \times log(scale)$				Neigh. Size $x \times log(scale)$				Neigh. Size $x \times log(scale)$			
Dict.	x =40	x =80	x =40	x =80	x =40	x =80	x =40	x =80	x =40	x =80	x =40	x =80
200	38.4	37.0	47.7	42.7	44.0	42.7	53.3	48.3	39.7	40.7	50.2	47.9
400	40.0	39.3	47.4	48.2	48.3	45.7	56.4	52.0	44.0	41.7	53.7	51.3
800	42.4	40.4	50.6	48.8	53.0	47.1	58.0	56.3	47.3	46.0	57.0	52.0
1000	43.3	41.4	53.5	49.3	53.3	51.5	59.0	56.3	49.1	45.0	56.2	53.7

Table 5.8: Classification Accuracy(%) for directional pooling of Relative Orientation that is **Re-incorporated** into SIFT orientation for **Fifteen Scenes Rot Dataset**, With Different SIFT Dictionary Sizes, Different Neighbourhood Sizes and Different Number of Keypoints Per Image.

In light of the effort expended and the increase in the number of feature dimensions, the marginal gains seen with Orthogonally clustered Relative Orientation context don't seem exceedingly promising. Even after feature selection we are dealing with 3× the number of features than in the case of no directional pooling on a method that already has a significantly high number of dimensions (#keypoint clusters × #context clusters). With Re-incorporated Relative Orientation context however, the price is well worth paying given the substantial gain in classification accuracy. Even though the number of features are ≈ 3× #keypoint clusters after feature selection, the number is still well below the number of dimensions of even the non-directional Orthogonally clustered variant, and yields a higher classification accuracy.

Chapter 6

Going Beyond Relative Orientation Context

Chapter 5 posits capture of explicit Relative Orientation context and its incorporation into the SIFT descriptor itself. In doing that it opens up the possibility of incorporation of other types of context extracted from the keypoint neighbourhood. As discussed earlier, the objective of context captured from the neighbourhood of keypoints is to reliably make the keypoint features more descriptive, and in doing that make the pooled global representation more discriminative. Also, continuing from the discussion in Section 4.3, we need to ensure that the context captured is in-plane rotation invariant, an objective that at times takes the back seat if additional discriminativeness is attained at its cost.

In This Chapter

- Propose use of coarse Keypoint SIFT Histograms in keypoint neighbourhoods as context (Section 6.1)
- Examine this new context with and without Relative Orientation information re-incorporated into keypoint orientation (See 5.2) (Section 6.1.1)
- Joint clustering of Relative Orientation context and coarse Keypoint SIFT Histogram context (Section 6.2)
- Context and appearance capture without context clustering (Section 6.2.1)
- Use of Keypoint SIFT Histograms in keypoint neighbourhoods as features (Section 6.3)
- Context extraction from keypoint neighbourhoods on a directional basis (Section 6.4)

6.1 Coarse Keypoint SIFT in Keypoint Neighbourhood as Context

Prior art has seen approaches that encode co-occurring feature pairs and triples as features [Zhang et al., 2011a] [Xiaomeng Wu, 2014]. There is also work on weak geometric context encoding that statistically captures feature co-occurrence relationships [Wu, 2015] [Jegou et al., 2008]. In a similar vein, we propose to capture weak feature co-occurrence relationships through SIFT words that occur in a certain sized neighbourhood of each keypoint. We extract Relatively Aligned Keypoint SIFT descriptors from the image and cluster into a coarse dictionary (smaller size) and a more refined (larger size) dictionary. We then construct a histogram of the *coarse code words* of the SIFT descriptors occurring in each keypoint neighbourhood. These coarse histograms become our context. We follow it up by constructing 2D histograms as in Section 5.1.1, with refined SIFT words for the central keypoint in each neighbourhood forming one dimension, and *clusters* of the coarse neighbourhood histograms forming the other dimension. k-medoids [Park and Jun, 2009] is used for clustering the coarse SIFT histograms. See Figure 6.1 for a visual representation.

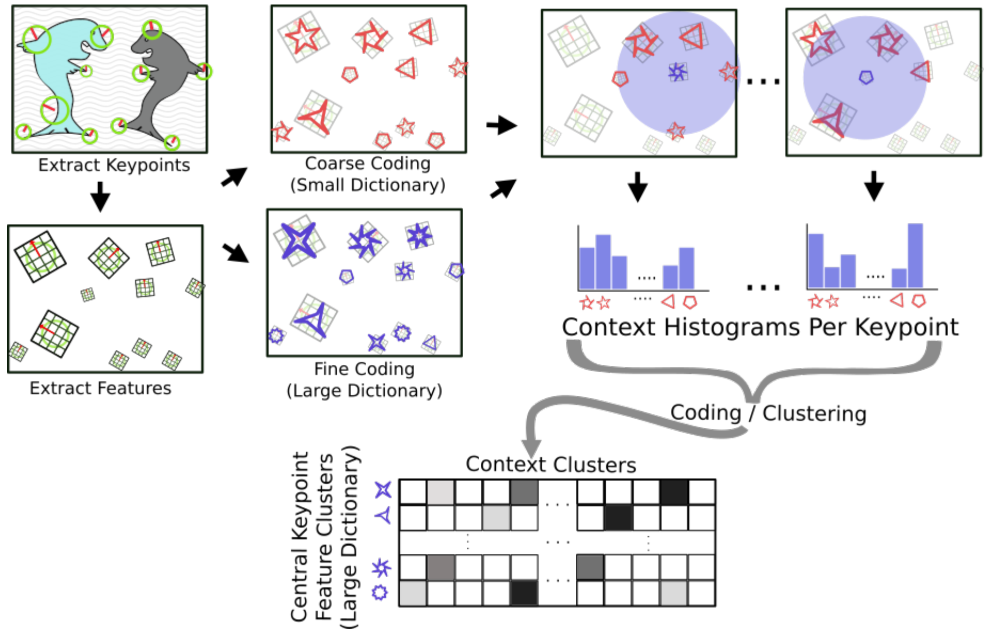


Figure 6.1: Construction of Coarse Keypoint SIFT Context

6.1.1 Coarse Keypoint SIFT and Re-incorporated Relative Orientation as Context

Of course, the premise of this chapter was incorporation of other cues *in addition* to Relative Orientation context, and that still holds. One need only use SIFT descriptors with Relative Orientation re-incorporated, and proceed as described before. Table 6.1 shows that incorporating Coarse SIFT context sees performance at par with Relatively Aligned SIFT. However, upon including (re-incorporated) Relative Orientation information, we see that the conflux of these contexts performs somewhat better than when (re-incorporated) Relative Orientation is the only context used. Neighbourhood size has a negligible impact.

	Re-incorp. RO Context		Coarse Keypoint SIFT Context (Coarse Dict. = 40) w/o re-incorp. RO context				with re-incorp. RO context			
			Neigh. Size $x \times log(scale)$				Neigh. Size $x \times log(scale)$			
			x=40		x=80		x=40		x=80	
SIFT	Neigh. Size		Context Dict		Context Dict		Context Dict		Context Dict	
Dict.	40	80	10	20	10	20	10	20	10	20
50	55.7	57.2	54.8	55.7	53.5	57.9	57.7	57.5	55.7	58.6
100	59.0	62.1	57.9	57.5	57.9	61.0	60.6	61.9	61.7	61.9
200	62.4	63.5	59.2	58.8	59.7	60.8	62.1	63.3	63.3	65.3
400	65.7	66.6	61.9	63.0	62.4	63.7	65.3	65.0	65.7	66.1
800	65.9	67.7	63.9	63.0	62.4	63.0	64.1	64.8	67.3	67.5
1000	64.1	67.3	63.5	63.3	64.1	64.6	66.8	64.6	70.2	68.4

Table 6.1: Comparison of Classification Accuracy(%) of Coarse Keypoint SIFT with and with-out (re-incorporated) Relative Orientation Context on **Fifteen Scene Categories Dataset**. $\approx 400 keypoints/image$
(Note the distinction between Context Dictionary Size and Coarse Dictionary Size)

Table 6.2 shows that the number of keypoints play a significant role in classification accuracy and there is a significant improvement when using more keypoints (≈ 400). Coarse Dictionary Size has only a marginal impact, where increasing it initially sees some gains, but beyond that it is a matter of diminishing returns.

	Coarse Dict. = 40				Coarse Dict. = 20			Coarse Dict. = 400		
	$\approx 250\, keypts/img$		$\approx 400\, keypts/img$		$\approx 400\, keypts/img$			$\approx 400\, keypts/img$		
SIFT	Context Dict.		Context Dict.		Context Dict.			Context Dict.		
Dict.	10	20	10	20	10	20	40	20	80	200
200	50.8	50.3	63.3	65.3	63.9	63.9	63.7	65.7	65.7	65.7
400	53.5	52.3	65.7	66.1	64.1	65.5	66.1	63.9	65.9	65.7
800	55.9	54.8	67.3	67.5	68.4	67.5	65.9	67.3	66.6	66.8
1000	58.8	56.6	70.2	68.4	66.1	66.8	65.3	66.4	66.8	65.3

Table 6.2: Comparison of Classification Accuracy(%) of Coarse Keypoint SIFT with re-incorporated Relative Orientation Context for various Coarse Dictionary sizes and various Context Dictionary Sizes on **Fifteen Scene Categories Dataset**. It also examines the effect of the number of keypoints.
Neighbourhood Size = $80 \times log(scale)$
(Note the distinction between Context Dictionary Size and Coarse Dictionary Size)

6.2 Combining Coarse Keypoint SIFT & Relative Orientation Histograms in Keypoint Neighbourhood

In the previous approach we saw that while Coarse Keypoint Histograms from feature neighbourhoods don't fare too well as context in isolation, their confluence with Relative Orientation context does proffer some promise, though at the cost of *in-plane rotation invariance* due to the method used for including Relative Orientation.

We can try using Orthogonally captured Relative Orientation context in an attempt to further increase discriminativeness without losing out on in-plane rotation invariance. We construct context histograms per keypoint by appending the Coarse Keypoint SIFT histogram in keypoint neighbourhood to the Relative Orientation histogram constructed in the keypoint neighbourhood. This combined histogram is then clustered and used as the second dimension of the 2D histogram, with central keypoint SIFT clusters (unaffected by Relative Orientation context) forming the first dimension of the 2D histogram.

Another reason to investigate the Orthogonal capture modality of Relative Orientation is the fact that its performance is largely unaffected by the number of keypoints, and we may leverage the need for fewer keypoints to offset the cost of computation of this additional context. (See Table 5.1)

Table 6.3 paints a slightly dismal picture where the joint clustering of the two contexts leads to a performance drop when compared to Relative Orientation used as the sole source of context. Tables 6.3 and 6.4 evidence that increasing the number of keypoints improves performance, while *Coarse Dictionary* size has no significant role to play. An increase in *Context Dictionary* size seems to hamper discriminativeness.

	Ortho. Relative Orient. Context (Hist Size = 8)				Coarse Keypoint SIFT + Relative Orient. Hist. Context (Hist Size = 20 + 8)							
	$\approx 250\, kypts/img$		$\approx 400\, kypts/img$		$\approx 250\, keypts/img$				$\approx 400\, keypts/img$			
SIFT	Context Dict.		Context Dict.		Context Dict.				Context Dict.			
Dict.	10	40	10	40	40	80	100	200	40	80	100	200
100	62.8	65.0	66.8	65.5	60.4	61.9	62.4	61.0	63.9	63.9	63.3	64.1
200	65.3	66.6	64.8	66.6	62.1	62.1	61.0	61.0	64.4	62.8	63.5	64.1
400	64.8	66.8	67.3	67.0	62.4	62.6	63.0	61.5	64.4	65.0	64.4	64.1
800	66.4	67.0	67.7	67.9	62.4	61.0	59.7	58.1	66.1	66.4	65.3	64.1

Table 6.3: Classification Accuracy(%) of Coarse Keypoint SIFT with appended Relative Orientation Histogram as Context for various Context Dictionary Sizes and number of keypoints on **Fifteen Scene Categories Dataset**.
Neighbourhood Size = $80 \times log(scale)$
(Note the distinction between Context Dictionary Size and Coarse Dictionary Size)

	Coarse Keypoint SIFT + Rel. Orient. Hist. Context (Hist Size = **20** + 8)						Coarse Keypoint SIFT + Rel. Orient. Hist. Context (Hist Size = **40** + 8)					
	$\approx 250\, keypts/img$			$\approx 400\, keypts/img$			$\approx 250\, keypts/img$			$\approx 400\, keypts/img$		
SIFT	Context Dict.			Context Dict.			Context Dict.			Context Dict.		
Dict.	40	100	200	40	100	200	40	100	200	40	100	200
100	60.4	62.4	61.0	63.9	63.3	64.1	62.1	60.4	60.6	63.0	62.1	64.6
200	62.1	61.0	61.0	64.4	63.5	64.1	62.8	61.0	59.5	63.3	62.4	63.5
400	62.4	63.0	61.5	64.4	64.4	64.1	61.9	61.5	60.1	65.0	64.4	62.8
800	62.4	59.7	58.1	66.1	65.3	64.1	63.3	60.8	61.0	65.5	63.9	64.4

Table 6.4: Classification Accuracy(%) of Coarse Keypoint SIFT with appended Relative Orientation Histogram as Context for various Context Dictionary Sizes and number of keypoints on **Fifteen Scene Categories Dataset**. Neighbourhood Size = $80 \times log(scale)$ (Note the distinction between Context Dictionary and Coarse Dictionary)

6.2.1 Coarse Keypoint SIFT in Keypoint Neigbourhood Without Clustering

We suspected that the act of further clustering Coarse SIFT Histograms may have an adverse role to play, so we try out an approach wherein for each image, Coarse SIFT Histograms per central Keypoint's code word are averaged together. This builds upon the hierarchical clustering interpretation of 2D appearance-context histograms. Instead of using context clusters to disambiguate within appearance clusters, we use the mean of context features falling within the appearance cluster. This produces a descriptor of size: $Fine\, Dictionary\, Size \times Coarse\, Dictionary\, Size$.
This approach too fails to yield promising results, as seen in Table 6.5. The classification accuracy with more keypoints (≈ 400 keypoints/image) matches that for Relatively Aligned SIFT, but falls behind when using fewer keypoints.

	Un-clustered Coarse SIFT Histogram Context			
	$\approx 250\, keypts/img$		$\approx 400\, keypts/img$	
SIFT	Coarse Dict.		Coarse Dict.	
Dict.	20	50	20	50
100	50.8	49.9	59.0	63.7
200	55.9	52.8	59.7	62.4
400	58.1	56.1	61.2	61.7
800	61.0	59.2	62.1	62.1
1000	59.9	57.9	64.8	64.8

Table 6.5: Classification Accuracy(%) of Neighbourhood Coarse Keypoint SIFT BoW appended to Relatively Aligned SIFT BoW, used as global descriptor on **Fifteen Scene Categories Dataset**.
Neighbourhood Size = $80 \times log(scale)$ (Note the distinction between Context Dictionary and Coarse Dictionary)

6.3 Coarse Keypoint SIFT Histogram in Keypoint Neighbourhood as Feature

While the information captured in Neighbourhood Coarse Keypoint SIFT histograms did not fare well when used as context, we investigate if it just might encode enough co-occurrence information to be used directly as the keypoint feature, without needing a second dimension to disambiguate through the central keypoint's SIFT code word. As seen in Table 6.6, it disappointingly performs worse than even Relatively Aligned Keypoint SIFT.

Despite the disappointing performance, Neighbourhood Coarse Keypoint (global) Bag-of-Words may still encode information complementary to that encoded in Relatively Aligned Keypoint SIFT Bag-of-Words. We append the two together and present the results in Table 6.7. With this additional information, *it manages to outperform Relatively Aligned SIFT by a significant margin.* The performance is in the ball-park of Relative Orientation Context (Table 5.3)

	Neigh. Coarse Keypt. SIFT Bag of Words			Neigh. Coarse Keypt. SIFT Bag of Words		
	≈ 250 *keypts/img*			≈ 400 *keypts/img*		
Feat.	Coarse Dict.			Coarse Dict.		
Dict.	200	400	800	200	400	800
50	28.1	28.3	25.2	47.0	39.0	34.1
100	33.0	30.3	25.8	46.5	44.1	41.4
200	37.0	32.7	31.6	52.6	52.1	46.5
400	38.8	33.4	34.1	55.2	55.0	54.3
800	37.9	37.6	37.9	57.0	57.2	55.5

Table 6.6: Classification Accuracy(%) of Coarse Keypoint SIFT from keypoint Neighbourhoods directly used as feature that is clustered and encoded as BoW on **Fifteen Scene Categories Dataset**.
Neighbourhood Size = $80 \times log(scale)$ (Note the distinction between Feature Dictionary and Coarse Dictionary)

Feat. Dict.	Rel. Aligned Keypt SIFT Bag of Words	$\big[$Rel. Algn. SIFT BoW, Neigh. Coarse SIFT BoW$\big]$
200	61.5	65.3
400	63.3	66.8
800	64.6	70.4
1000	64.6	68.8
1500	64.8	71.5

Table 6.7: Classification Accuracy(%) of Neighbourhood Coarse Keypoint SIFT BoW appended to Relatively Aligned SIFT BoW, used as global descriptor on **Fifteen Scene Categories Dataset**.
Neighbourhood Size = $80 \times log(scale)$, ≈ 400 keypoints/image, Coarse Dictionary Size = 400
(Rel. Aligned SIFT Dictionary Size is the same as the Neighbourhood Coarse SIFT BoW Feature Dictionary Size)

6.4 Directional Capture of Coarse Keypoint SIFT Context

Continuing with the context-clutter rejection approach posited in Section 5.3, we explore if directional capture of Coarse Keypoint SIFT histograms in keypoint neighbourhoods would improve classification accuracy. We do this non-exhaustively for the sake of brevity, picking Orthogonal clustering of Neighbourhood Coarse Context Histograms *appended* with Relative Orientation Histograms as the reference setting.

Table 6.8 shows that directional pooling of context does indeed improve the classification accuracy, but the performance matches that of *directionally pooled* Orthogonally clustered Relative Orientation. The additional contextual information coming via the Coarse SIFT histograms does not seem to be contributing to the discriminativeness in a positive way.

	Directional Ortho. Clustered Rel. Orient.				Coarse Keypt SIFT + Rel. Orient. Hist (Hist Size = **20** + 8)		Directional Coarse Keypt SIFT + Rel. Orient. Hist. (Hist Size = **20** + 8)			
	Per-Quar.		Per-Half				Per-Quar.		Per-Half	
SIFT	Context Dict.		Context Dict.		Context Dict.		Context Dict.		Context Dict.	
Dict.	10	20	10	20	40	100	40	100	40	100
100	63.3	67.0	65.9	67.7	63.9	63.3	66.6	65.9	66.6	66.1
200	65.7	67.3	65.3	67.5	64.4	63.5	65.7	66.8	66.8	67.0
400	66.1	66.4	68.6	68.2	64.4	64.4	67.3	67.3	67.9	67.0
800	66.8	65.9	67.3	68.4	66.1	65.3	67.5	65.9	68.6	68.4

Table 6.8: Classification Accuracy(%) of *Directionally Pooled* Coarse Keypoint SIFT with appended Relative Orientation Histogram as Context for various Context Dictionary Sizes on **Fifteen Scene Categories Dataset**. Neighbourhood Size = $80 \times log(scale)$, ≈ 400 keypoints/image (Note the distinction between Context Dictionary and Coarse Dictionary)

Chapter 7

Orientation Context in Convolutional Neural Networks

Note: *We were unable to debug the implementation and complete the testing of the methods proposed in this chapter in time for the submission. The methods developed and the motivation are described in detail, with the implementation and analysis of the results relegated to a future publication.*

In This Chapter

- Introduction to Deep Learning, with focus on Convolutional Networks (Section 7.1.1)
- Overview of transformation invariance and context capture in Deep Networks (Section 7.1.2)
- Overview of prior attempts at improving neural net architectures through extant computer vision techniques (Section 7.1.3)
- Propose two variants of local rotation aware convlolution layer, for learnable parameter reduction (Section 7.2)
- Propose incorporation of *relative orientation* context in Convolutional Networks for in-plane rotation invariance(Section 7.3)

7.1 Deep Learning Background and Motivation

7.1.1 (Short) Introduction To Deep Learning

Neural Networks are comprised of processing units called *perceptrons*, taking a set of scalar inputs $\mathbf{x}_k$ with weights $\mathbf{w}_k$ associated with each perceptron k. Each perceptron has an associated threshold or *bias* η_k, and a scalar output y_k that is activated if the inner product $\mathbf{w}_k.\mathbf{x}_k$ exceeds the threshold. This can be modeled with a sigmoid as $y_k = sigmoid(\mathbf{w}_k.\mathbf{x}_k - \eta_k) + 1$.

These perceptrons or *neurons* can be put together as a Directed Acyclic Graph in a layered fashion, with neurons in a particular layer being fed in from the previous layer, and its output feeding into the next layer in the hierarchy. These multilayer structures can act as universal approximators with even one hidden layer [Castro et al., 2000] [Hornik et al., 1989]. However, network depth is the key to practical applications of neural networks because with each layer that is removed from the network, the number of perceptrons required to approximate the target function increases exponentially [Hastad, 1986] [Meunier et al., 2010].

Supervised learning of these neural networks entails learning the weights $\mathbf{w}_k$ and threshold η_k $\forall k$ *units*, provided input-output data pairs. Gradient descent is employed for learning, with *backpropagation* of the error term making each perceptron aware of its optimization objective.

Convolutional Neural Networks (CNN) introduce a semblance of regular spatial arrangement to the perceptrons within each layer. Each layer has perceptrons arranged in multiple regular-grid slices of the same size. Each perceptron in layer i is fed in by outputs of all slices of the previous layer $i - 1$ lying with a certain local spatial extent. This local spatial extent is defined as the (spatial) kernel size of the perceptron. All perceptrons within the same slice share the same weights, and hence the output of all units of a single slice can be viewed as akin to a 3D convolution operation followed by a non-linearity. The spatial extent of the convolution operator is made explicit as the *kernel size*. The third dimension of the kernel is implicit and equals the number of slices of the previous layer.

CNNs stack multiple such convolution layers, with the spatial extent reduced between subsequent convolution layers through downsampling or *pooling* in the spatial dimension. For the purposes of classification, CNNs employ *fully-connected* layers as the end stages of the network. Inputs to perceptrons in these fully connected layers get all the outputs of the previous layer. This can alternatively be viewed as a convolution layer with the spatial extent of the convolution operator matching the spatial extent of the previous layer.

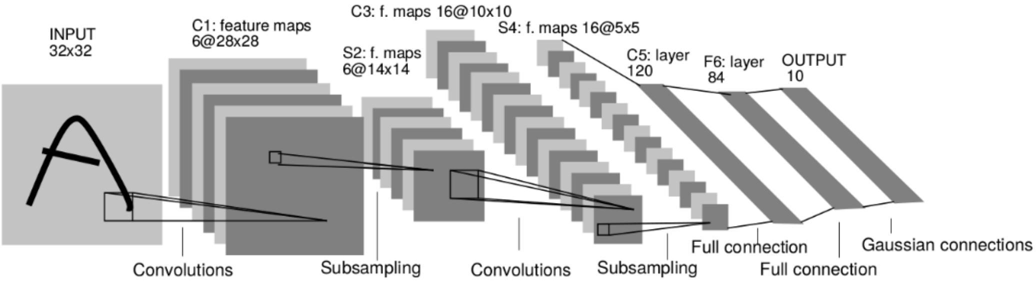


Figure 7.1: Architecture of LeNet, representative of Convolutional Neural Networks [LeCun et al., 1998]

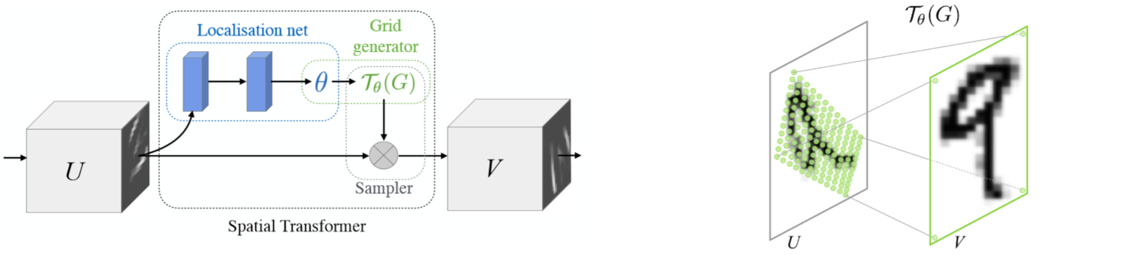
The facility of learning *end-to-end*, i. e., from image directly to classifier scores is one of the key advantages enjoyed by deep architectures, with the downside that the vast number of learnable parameters necessitate a large corpus of labeled examples. The sheer complexity of the underlying model also requires regularization, with methods such as L_2 regularization of weights, or a technique analogous to *bagging*, known as *dropout* [Srivastava et al., 2014] commonly employed.

7.1.2 Invariance and Context In Deep Networks

Deep Networks have certain transformation invariances built into their architecture. Convolutional Networks, by virtue of having the convolutional weights tied for the entire spatial extent of the input, are invariant to translations. Further, pooling layers, which strive to down-sample feature maps by picking the average or the max value in local neighbourhoods, afford an additional degree of invariance to local deformations, and to some extent, rotations.

The highly complex nature of the underlying model also means that various transformation invariances can be learned directly from data if sufficient examples are provided. Data augmentation [Krizhevsky et al., 2012] tries to brute force invariance by applying various transformations to the training data. One downside to this is increased training time. Another issue is the possible divergence of representations in all layers for inputs differing only by a single transformation, i. e., there is no single point of representation of the corresponding transformation. The consequence of this is duplication of kernels, differing only in the transformation group. There have been attempts in the graphics community to have single point representations of various transformations to allow the use of those nodes as knobs to manipulate the transformation in the rendering [Kulkarni et al., 2015]. We would touch upon this in Section 7.2 where we propose compaction of kernel representations under local rotations.

Tiled Convolutional Neural Networks [Ngiam et al., 2010] move away from imposed translation invariance arising out of tied weights and towards learning invariance from data. They propose a tiled pattern of tied convolution weights, with units k steps from each other sharing weights. This approach sees an improvement in classification performance, and learns rotational and scale invariant representations.



(a) Stages of the Spatial Transformer Layer

(b) Parameterized sampling of the image

Figure 7.2: Spatial Transformer Networks [Jaderberg et al., 2015]

Spatial Transformer Networks [Jaderberg et al., 2015] generalize attentional mechanisms for images by proposing a learnable Spatial Transformer module that undoes the effect of in-plane rotation and scaling on the image before passing it on to a standard network for feature extraction and inference. It uses a localization sub-network that parameterizes subsequent sampling of the image. See Figure 7.2b.

Epitomic Convolutions [Papandreou et al., 2015] model deformations through max pooling over subsets of the convolutional kernel. These subsets of the kernel, related to each other through translation and cropping of the kernel, are termed epitomes. See Figure 7.3.

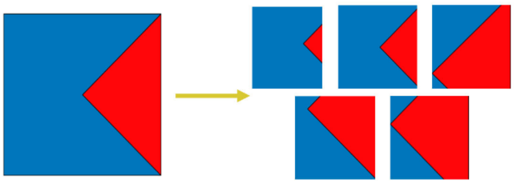


Figure 7.3: Pictorial representation of epitomic convolutional kernels [Papandreou et al., 2015]

Invariance to local transformations can also be modeled by linear combinations of transformed representations [Sohn and Lee, 2012]. This is an idea that we would revisit in Section 7.2.2.

Deep Symmetry Networks (Symnets) [Gens, 2014] generalize the definition of feature maps to symmetry groups apart from translational symmetry group. The primary focus is affine groups, which include translation, rotation, shear and scaling, and is approximated using kernel functions. Discriminative training allows learning of task appropriate invariance, i. e., the concepts of 6 and 9 are not confused.

The structure of the convolution kernel imposes local geometric constraints, and convolutions in subsequent layers can be seen as going up the feature hierarchy, with complex local descriptors being constructed out of simpler local descriptors through imposition of geometric constraints. Pooling, however, strives to weaken the geometric constraints imposed by the subsequent convolution layer by allowing some degree of flexibility in feature co-location. Thus one can observe that the region of the image acted upon by neurons, or the *receptive field* increases in size as one goes higher in the feature hierarchy owing to convolutions and pooling, and the features in the later layers can be interpreted as weak co-occurrence relationships between parts, which in-turn are constructed through stronger co-occurrence between low level features.

Semantic context information has been exploited as a means to constrain and regularize the feature space of the network by learning context prediction as an auxiliary task in parallel with a weak classifier, and using the combined outputs for the final prediction [Kekeç et al., 2014].

The fusion of information from complementary tasks is an idea that has seen wide adoption. One can view one task as the primary objective and the other as providing the context. Two stream networks such as this have found use in action recognition in videos [Simonyan and Zisserman, 2014a] where one network keeps track of spatial information and the other is used to provide temporal context.

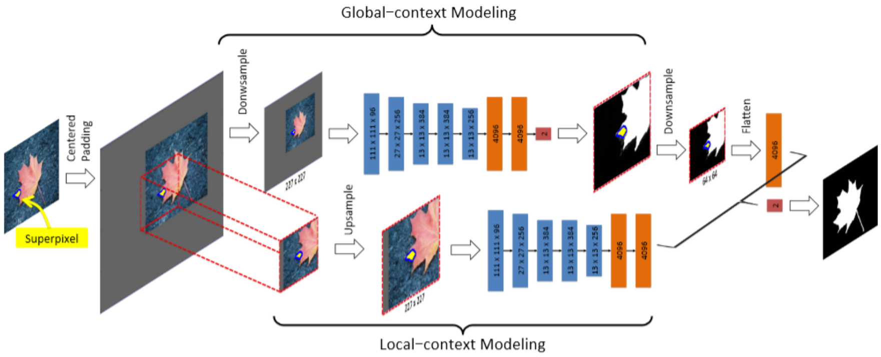


Figure 7.4: Two stream schema used to capture local and global context [Zhao et al., 2015]

Two stream schema also find use in saliency detection where spatially global and local context is captured by the two arms [Zhao et al., 2015].

An alternate mechanism of exploiting auxiliary or surrogate tasks is to pre-train the network on the auxiliary task and fine tune the network for the primary task. This may be viewed purely from the lens of learning as the leveraging of data labeled for a different task, or from the point of view of context as leveraging additional information to shape underlying feature space [Razavian et al., 2014].

Other applications choose to explicitly model geometric and non-geometric contextual relationships for object detection and pose estimation, considering background vs. object relationship and *object vs. object* relationships [Vu et al., 2015].

7.1.3 Modifying Deep Architectures Through Traditional Computer Vision Methods

Here we list a few examples of extant Computer Vision techniques being employed in Deep Learning or equivalences being drawn between traditional pipelines and Deep Learning:

Spatial Pyramid Pooling Networks (SPP-Net) [He et al., 2014] utilize a Spatial Pyramid Pooling layer after the last convolutional layer to create fixed length representations for images regardless of the image size. In doing away with cropping or warping to fit the input image to the expected input size, they report improvements regardless of the network arrangement and task. See Figure 7.5.

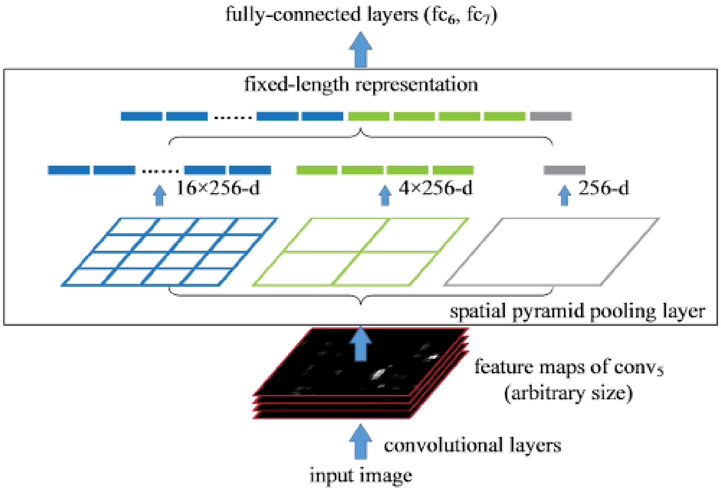

fully-connected layers (fc_6, fc_7)
fixed-length representation
16×256-d
4×256-d
256-d
spatial pyramid pooling layer
feature maps of $conv_5$
(arbitrary size)
convolutional layers
input image

Figure 7.5: Spatial Pooling Layer after a Convolutional Layer with 256 channels [He et al., 2014]

Deep Fisher Networks [Simonyan et al., 2013] explore deep stacking of hand crafted features and representations and its effect on performance. They propose Fisher Layers that operate on densely sampled and de-correlated features coming from the previous layer, compute semi local Fisher Vector encodings, pool features in a 2×2 neighbourhood by stacking them together, and output the L_2 normalized and PCA down projected features to the next layer. See Figure 7.6.

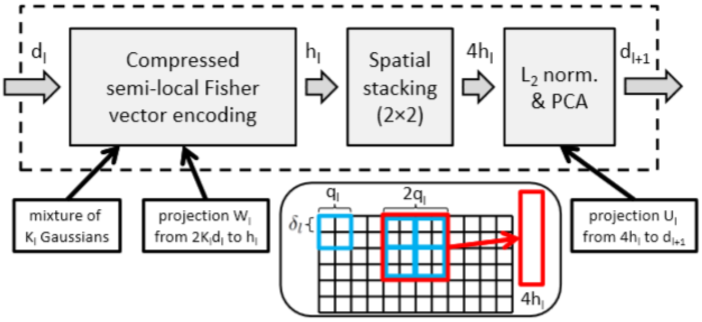

d_l
Compressed semi-local Fisher vector encoding
h_l
Spatial stacking (2×2)
$4h_l$
L_2 norm. & PCA
d_{l+1}
mixture of K_l Gaussians
projection W_l from $2K_l d_l$ to h_l
q_l
$2q_l$
δ_l
$4h_l$
projection U_l from $4h_l$ to d_{l+1}

Figure 7.6: Sub-stages of a Fisher Layer [Simonyan et al., 2013]

In a manner similar to Histogram of Oriented Gradient Pyramids, convolution layers have seen modifications where convolutions are performed in scale space with the same kernel, followed by max pooling of the result over scales to enable scale invariance without increasing the number of parameters [Kanazawa et al., 2014].

Deformable Parts Models, when unrolled, have been demonstrated to be equivalent to Convolutional Neural Networks [Girshick et al., 2014] and a new architecture called *DeepPyramid DPM* proposed which replaces HOG parts of DPM with learned CNN features. Additional means of fusing CNNs and DPMs have been recently proposed as well [Wan et al., 2015].

7.2 Orientation Aware Convolutions

As mentioned earlier, brute forcing the learning of invariances in CNNs results in widely differing underlying representations across all layers for images differing only by a single transformation. With in-plane rotation data augmentation, this may translate to duplications of convolution kernels within each convolution layer to account for different orientations. The network might also be coaxed to learn rotationally symmetric representations.

We propose two methods here to separate out feature orientation and the feature filter (kernel) so as to require fewer kernels to be learned. A drop in the number of kernels also reduces the implicit third dimension of the feature kernels in the subsequent convolution layer, which further reduces the number of learnable parameters.

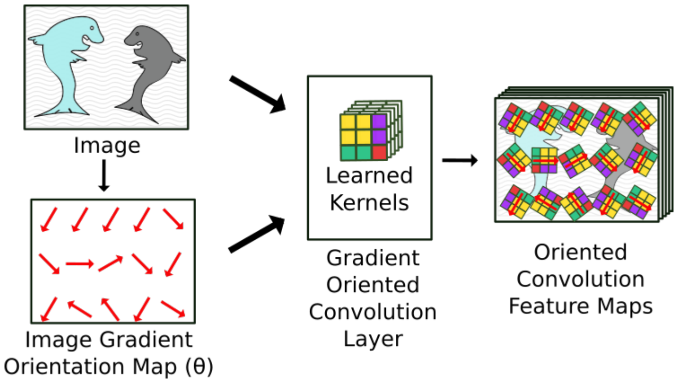


Image
Image Gradient Orientation Map (θ)
Learned Kernels
Gradient Oriented Convolution Layer
Oriented Convolution Feature Maps

Figure 7.7: Schema for Gradient Oriented Convolution Layer

7.2.1 Gradient Oriented Convolution Layer

This convolution layer takes the image gradient orientation map as input in addition to the image. At each convolution location, the kernel is oriented along the gradient direction akin to Keypoint SIFT. See Figure 7.7.
The resulting inner products at each convolution are rotationally invariant but now have lost all orientation information. That information can be re-introduced by concatenating the gradient orientation map with the resulting feature map. Since orientations ($[0, 2\pi]$) are circular quantities, instead of appending the angle orientation map we instead append cosine and sine maps of the orientations. See Figure 7.8.

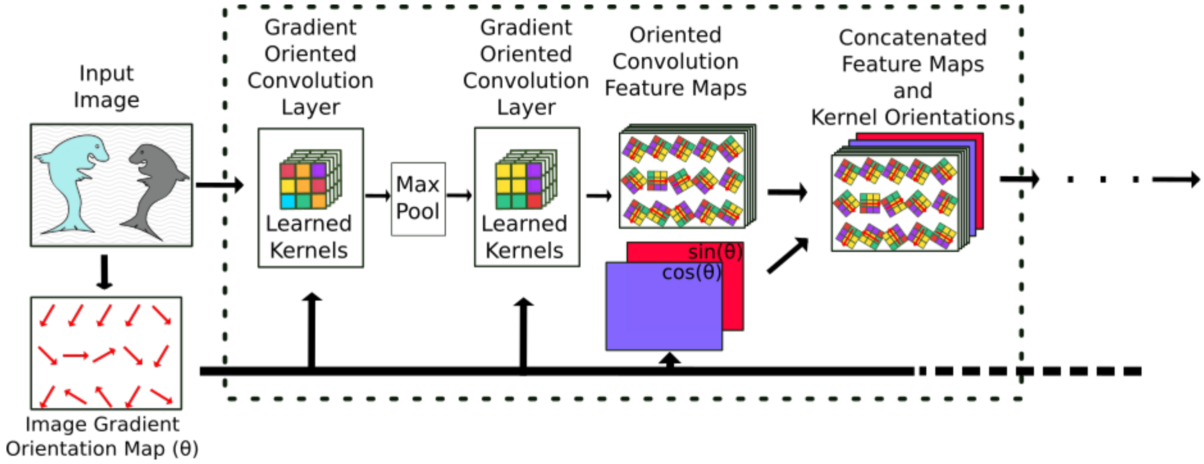

Input Image
Gradient Oriented Convolution Layer
Learned Kernels
Max Pool
Gradient Oriented Convolution Layer
Learned Kernels
Oriented Convolution Feature Maps
Concatenated Feature Maps and Kernel Orientations
sin(θ)
cos(θ)
Image Gradient Orientation Map (θ)

Figure 7.8: Schema for Rotation Group Aware Convolutional Network

Implementing convolutions with arbitrary kernel rotations can get unwieldy, so, as occasionally done in prior work [Sohn and Lee, 2012], we model arbitrary kernel rotations (in-plane) with linear combinations of orthogonally oriented kernels as shown in Figure 7.9. One may alternatively choose to use linear combinations of kernels in 8 orientations.

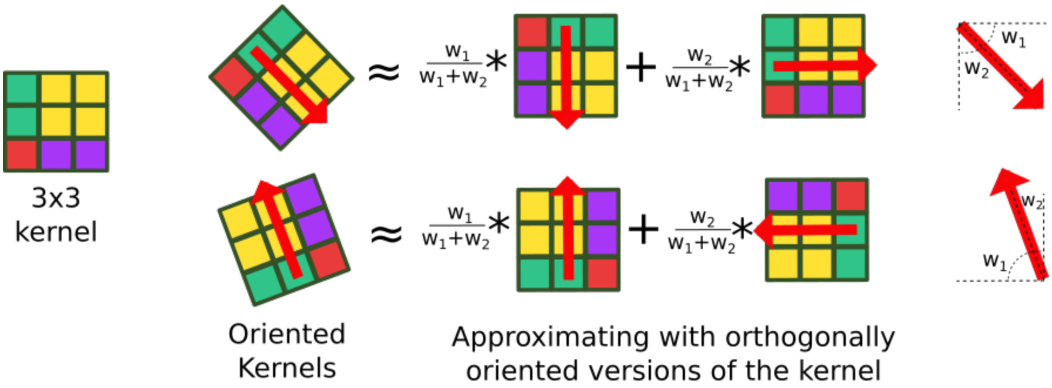


Figure 7.9: Approximating Oriented Convolutions

7.2.2 Kernel Orientation Max Pool Convolution Layer

Alternatively, we can borrow the concept of max pooling over transformations [Papandreou et al., 2015]. At each convolution location (x, y), the convolution result C_{xy} gets the value of the maximum of the convolution results from 4 orthogonally oriented versions of the kernel. Additionally, the selected orientations are put out as an orientation map, either with the chosen kernel indices, as shown in Figure 7.10, or as 2 maps with cosine and sine of the orientations.

As with the method proposed earlier, one may choose to max pool over more finely divided orientations of the convolution kernel.

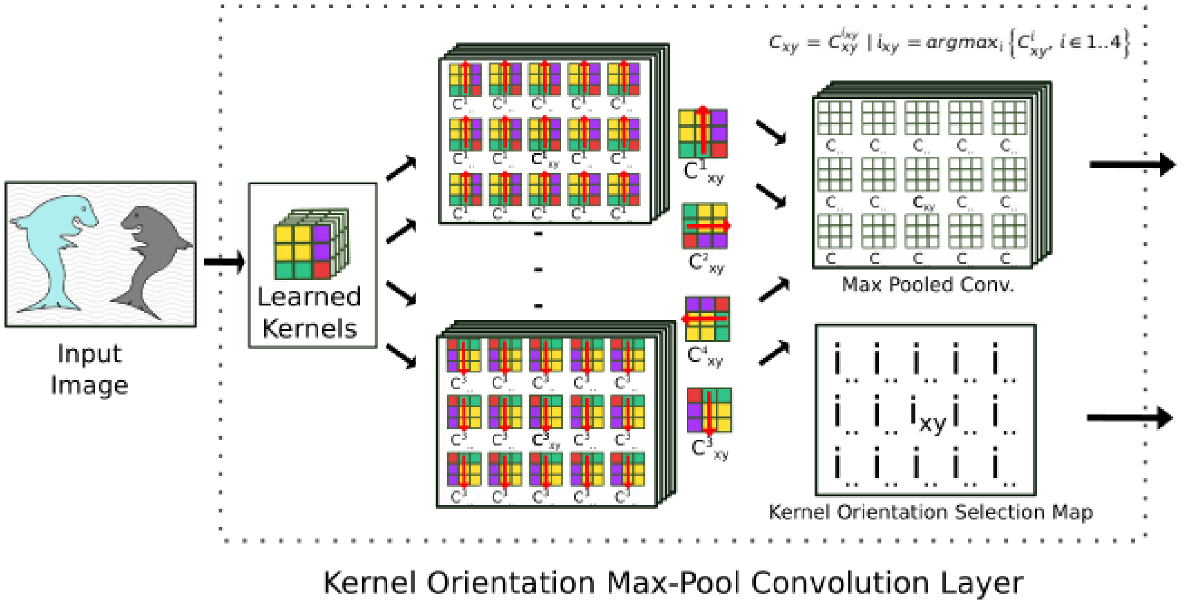

$C_{xy} = C_{xy}^{i_{xy}} \mid i_{xy} = argmax_i \{C^i_{xy}, i \in 1..4\}$
Input Image
Learned Kernels
C^1_{xy}
C^2_{xy}
C^4_{xy}
C^3_{xy}
C_{xy}
Max Pooled Conv.
i_{xy}
Kernel Orientation Selection Map
Kernel Orientation Max-Pool Convolution Layer

Figure 7.10: Kernel Orientation Max Pool Convolution Layer

The orientation selection map produced may be concatenated with the feature map and fed into a subsequent layer, or orientation selection maps may be collated, resized and processed in a parallel arm of the network and merged in at the fully connected layers.

The advantage of decoupling feature representation and feature orientations is that with in-plane rotation data augmentation, the learning of rotation invariance can be concentrated in a small part of the network, without muddling the features and requiring fewer kernels to be learned. See Figure 7.11.

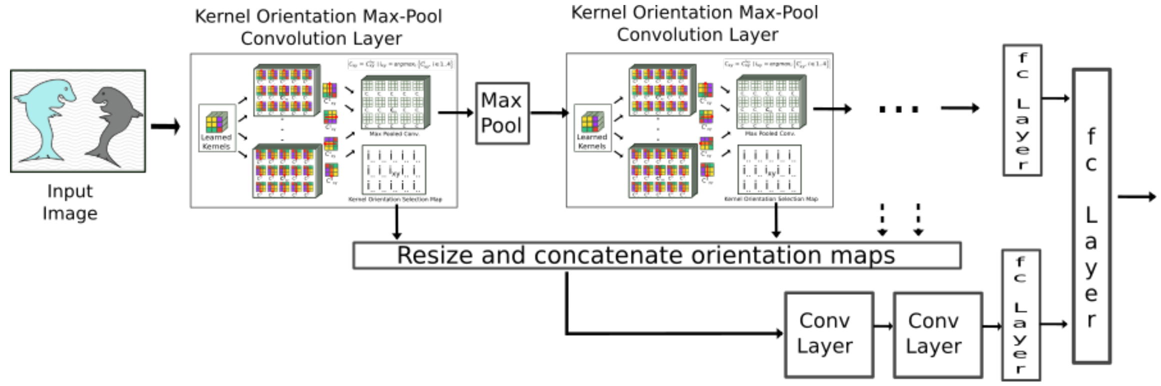

Kernel Orientation Max-Pool Convolution Layer
Kernel Orientation Max-Pool Convolution Layer
Input Image
Max Pool
. . .
Resize and concatenate orientation maps
Conv Layer
Conv Layer
fc Layer
fc Layer
fc Layer

Figure 7.11: De-coupling feature representations and rotation invariance via a two stream processing scheme

7.3 Incorporating Relative Orientation In CNNs

The schema for Rotation Group aware convolutions seen in Figure 7.8 can be adopted to use relative orientation histograms in feature neighbourhoods instead of absolute gradient orientations. The orientation cosine and sine maps used earlier can be substituted with 8 feature maps, each corresponding to one bin of the Relative Orientation histogram constructed in the receptive field of the convolution layer at each spatial location. See Figure 7.12.

This move can be seen as the equivalent of moving from Upright SIFT to Relatively Aligned SIFT with Relative Orientation information, leading to representations that retain their expressiveness while being inherently immune to in plane rotations to a large extent.

The network would still require some data augmentation, but won't need to see rotated copies of all training images because it no longer needs to re-learn feature representations in different orientation.

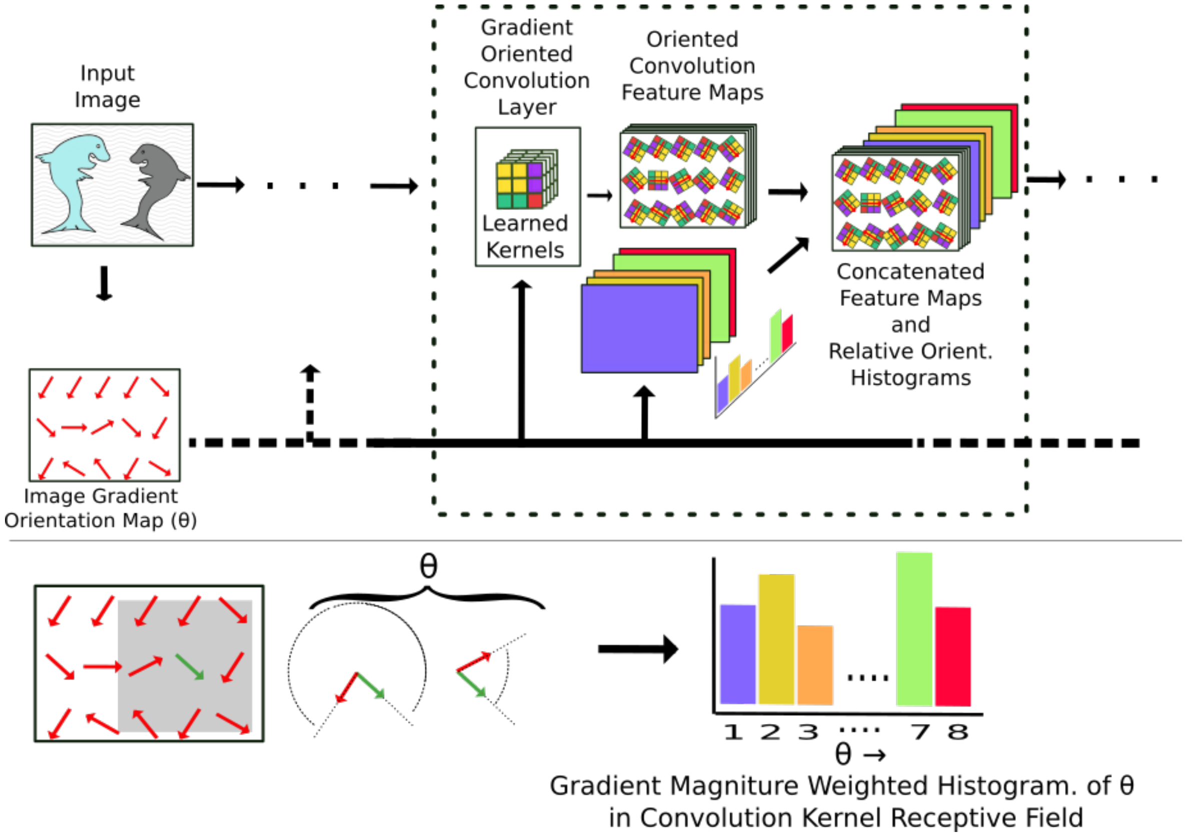


Figure 7.12: Schema for incorporation of Relative Orientation context in Convolution Layer

Note that the computation of relative orientation histograms here would need to take into account the gradient magnitudes because there are no longer the dominant gradients in feature patches.

Chapter 8

Conclusion and Implementation Notes

8.1 Summary

In this thesis, we elaborated on the various ways contextual information can be captured (Sec 3.2), and the association between feature distinctiveness, geometric context, higher level features and the discriminativeness of the global image representation (Sec 3.3).

This was followed by an examination of implicit *relative orientation* contextual information captured by Dense SIFT (Chapter 4). We then proposed to explicitly capture this relative orientation context to increase discriminativeness of image representations without compromising on in-plane rotation invariance and scale invariance (Chapter 5). Towards this we propose orthogonal clustering of appearance and context, in place of joint clustering, such that each may be clustered in their respective lower dimensional spaces without treading into the *Curse of Dimensionality* territory, at least for the clustering. We also proposed a method of re-incorporating relative orientation information back into the appearance dimension (Sec 5.2) so that the orthogonal clustering direction becomes available for incorporation of other contextual information. Although this approach led to more discriminativeness, it came at the cost of in-plane rotation invariance.

The general approach of differentiating within appearance clusters using context, such that may alleviate the effect of fuzziness in appearance space by using larger clusters (smaller dictionaries), harkens to VLAD [Jégou et al., 2010] pooling and Fisher Vectors [Sánchez et al., 2013]. We explore several flavours of this approach when capturing weak co-occurrence relationships (Chapter 6).

We also proposed Directional Pooling of contextual information from feature neighbourhoods in an attempt to reduce the effect of clutter on context (Section 5.3). We also discuss a simple feature selection strategy to pick distinctive features (Section 5.3.1).

We take the lessons learned about absolute and relative orientations of features over to the domain of Convolutional Neural Networks (Chapter 7). We propose two schemes for reducing the number of parameters and distinct kernels required for learning rotational invariance from data by decoupling feature descriptor and orientation information (Section 7.2). We then take the developed framework and substitute absolute orientation maps with relative orientation information, making the architecture inherently rotation invariant. One of the key ideas developed here is that constraining a sub-part of the network to be responsible for the learning of the invariance might fare better than mangling feature representation and invariance learning.

8.2 Implementation Notes

We make use of MATLAB as the prototyping environment, supported by a Parallel Computing Cluster. We use VLFEAT [Vedaldi and Fulkerson, 2008] toolbox for keypoint extraction and SIFT computation. LIBSVM [Chang and Lin, 2011] is used for multi-class (OVO) SVMs, with grid search for the slack hyper-parameter C. For Convolutional Networks, we make use of Caffe [Jia et al., 2014] with the networks trained on NVIDIA K40 GPUs.

Code and other documentation may be accessed at `https://github.com/mehtadushy/SpatialContext`.

8.3 Future Work

The primary focus would be testing the methods proposed in Chapter 7 on various datasets to gauge the efficacy of the proposed approaches.

The main take home message of this thesis is that lessons learned in the past decades of Computer Vision research are directly applicable to Deep Vision pipelines, and it might be worth the effort exploring and porting other techniques to Deep Learning pipeline.

Additionally, we would like to explore the effect of constraining the learning of various invariances to sub-parts of the Deep Learning pipeline for transformations other than in-plane rotations. We would also like to be able to formally argue how such an approach can be seen as a form of regularization on the network.

List of Figures

2.1 Hyperplanes and their margins for linear classification of separable data 6
2.2 Representation of the partitioning of feature space induced by Decision Trees. Source: classes.cs.uchicago.edu 7
2.3 Typical Stages of the Image Classification Pipeline 7
2.4 A representative set of Haar-like low level features 8
2.5 Example of SIFT orientation histograms for an 8 × 8 neighbourhood [Lowe, 2004] 9
2.6 Schema for Bag of Visual Words encoding 10
2.7 Categories represented in Fifteen Scene Categories Dataset [Lazebnik, 2004] 11
2.8 MNIST Dataset and Variants 11

3.1 Spatial Histograms used to construct 2nd Order Features where for important feature code-word pairs (w_a, w_b), the average spatial distribution of w_b w.r.t w_a is captured, and vice versa [Liu et al., 2008] 14
3.2 Ideas for pairwise feature co-occurrence and the partitioning of image plane as the first step to statistical co-occurrence capture [Liu et al., 2012] 15
3.3 Spatial Pyramid Levels [Lazebnik, 2004] 15
3.4 Closing the inference loop in Implicit Shape Models [Leibe et al., 2004] 16

4.1 Visualization of SIFT patches for Dense SIFT and Keypoint SIFT 19
4.2 Mechanism of capture of relative orientation in Upright SIFT 20
4.3 Classification accuracy of BoW of various SIFT descriptors on Fifteen Scene Categories dataset 21
4.4 Classification accuracy of BoW of various SIFT descriptors on Fifteen Scenes Rot dataset. See 4.3 21

5.1 Relative Orientation Histogram captured in keypoint neighbourhood 23
5.2 Representation of the context information that is clustered orthogonal to the central keypoint's appearance information (SIFT Clusters), resulting in the 2D histogram shown in (b) 24
5.3 Classification Accuracy for SIFT BoW with Relative Orientation Context for the Fifteen Scene Categories Dataset. Context is extracted from neighbourhoods of size $80 \times log(scale)$ 25
5.4 Classification Accuracy for SIFT BoW with Relative Orientation Context for Fifteen Scenes Rot Dataset. Note the difference is keypoint neighbourhood sizes between Orthogonally clustered Relative Orientation and Re-incorporated Relative Orientation. 25
5.5 Scheme for re-incorporation of Relative Orientation into SIFT 26
5.6 Representation of Per-quarter and Per-half neighbourhood definition for extraction of context. Features resulting from each of the above defined neighbourhood segments are clustered and pooled independently and appended together to yield the final representation. Feature selection is then applied to get rid of non-useful features. 28

6.1 Construction of Coarse Keypoint SIFT Context 32

7.1 Architecture of LeNet, representative of Convolutional Neural Networks [LeCun et al., 1998] 38
7.2 Spatial Transformer Networks [Jaderberg et al., 2015] 39
7.3 Pictorial representation of epitomic convolutional kernels [Papandreou et al., 2015] 39
7.4 Two stream schema used to capture local and global context [Zhao et al., 2015] 40
7.5 Spatial Pooling Layer after a Convolutional Layer with 256 channels [He et al., 2014] . . . 40

7.6 Sub-stages of a Fisher Layer [Simonyan et al., 2013] . . . 41
7.7 Schema for Gradient Oriented Convolution Layer . . . 41
7.8 Schema for Rotation Group Aware Convolutional Network . . . 42
7.9 Approximating Oriented Convolutions . . . 42
7.10 Kernel Orientation Max Pool Convolution Layer . . . 43
7.11 De-coupling feature representations and rotation invariance via a two stream processing scheme . . . 43
7.12 Schema for incorporation of Relative Orientation context in Convolution Layer . . . 44

List of Tables

5.1 Classification Accuracy(%) for Orthogonally Clustered Relative Orientation Context on **Fifteen Scene Categories Dataset** and **Fifteen Scenes Rot Dataset**, With Different SIFT Dictionary Sizes, Different Context Dictionary Sizes and Different Number of Keypoints Per Image. Neighbourhood Size = $80 \times log(scale)$ (Note that Context Dictionary Size is different from the number of bins in the Relative Orientation histogram) 24

5.2 Classification Accuracy(%) for **Angle Range Restricted** Re-incorporated Clustered Relative Orientation Context on **Fifteen Scene Categories Dataset** and **Fifteen Scenes Rot Dataset**, With Different SIFT Dictionary Sizes, Different Neighbourhood Sizes and Different Number of Keypoints Per Image. 27

5.3 Classification Accuracy(%) for Re-incorporated Clustered Relative Orientation Context on **Fifteen Scene Categories Dataset** and **Fifteen Scene Rot Dataset**, With Different SIFT Dictionary Sizes, Different Neighbourhood Sizes and Different Number of Keypoints Per Image. 27

5.4 Classification Accuracy(%) for directional pooling of Relative Orientation that is **Re-incorporated** into SIFT orientation for **Fifteen Scene Categories Dataset**, with Different SIFT Dictionary Sizes, Neighbourhood Size of $80 \times log(scale)$ with various Feature Selection schemes. 29

5.5 Classification Accuracy(%) for directional pooling of Relative Orientation Context that is **Orthogonally Clustered** for **Fifteen Scene Categories Dataset**, With Different SIFT Dictionary Sizes, Different Context Dictionary Sizes and Different Number of Keypoints Per Image. Neighbourhood Size = $80 \times log(scale)$ (Note that the Context Dictionary Size is different from the number of bins in the Relative Orientation histogram) 29

5.6 Classification Accuracy(%) for directional pooling of Relative Orientation Context that is **Orthogonally Clustered** for **Fifteen Scenes Rot Dataset**, With Different SIFT Dictionary Sizes, Different Context Dictionary Sizes and Different Number of Keypoints Per Image. Neighbourhood Size = $80 \times log(scale)$ (Note that the Context Dictionary Size is different from the number of bins in the Relative Orientation histogram) 30

5.7 Classification Accuracy(%) for directional pooling of Relative Orientation that is **Re-incorporated** into SIFT orientation for **Fifteen Scene Categories Dataset**, With Different SIFT Dictionary Sizes, Different Neighbourhood Sizes and Different Number of Keypoints Per Image. 30

5.8 Classification Accuracy(%) for directional pooling of Relative Orientation that is **Re-incorporated** into SIFT orientation for **Fifteen Scenes Rot Dataset**, With Different SIFT Dictionary Sizes, Different Neighbourhood Sizes and Different Number of Keypoints Per Image. 30

6.1 Comparison of Classification Accuracy(%) of Coarse Keypoint SIFT with and with-out (re-incorporated) Relative Orientation Context on **Fifteen Scene Categories Dataset**. $\approx 400 keypoints/image$ (Note the distinction between Context Dictionary Size and Coarse Dictionary Size) 32

6.2 Comparison of Classification Accuracy(%) of Coarse Keypoint SIFT with re-incorporated Relative Orientation Context for various Coarse Dictionary sizes and various Context Dictionary Sizes on **Fifteen Scene Categories Dataset**. It also examines the effect of the number of keypoints. Neighbourhood Size = $80 \times log(scale)$ (Note the distinction between Context Dictionary Size and Coarse Dictionary Size) 33

6.3 Classification Accuracy(%) of Coarse Keypoint SIFT with appended Relative Orientation Histogram as Context for various Context Dictionary Sizes and number of keypoints on **Fifteen Scene Categories Dataset**. Neighbourhood Size = 80 × $log(scale)$ (Note the distinction between Context Dictionary Size and Coarse Dictionary Size) 33

6.4 Classification Accuracy(%) of Coarse Keypoint SIFT with appended Relative Orientation Histogram as Context for various Context Dictionary Sizes and number of keypoints on **Fifteen Scene Categories Dataset**. Neighbourhood Size = 80 × $log(scale)$ (Note the distinction between Context Dictionary and Coarse Dictionary) 34

6.5 Classification Accuracy(%) of Neighbourhood Coarse Keypoint SIFT BoW appended to Relatively Aligned SIFT BoW, used as global descriptor on **Fifteen Scene Categories Dataset**. Neighbourhood Size = 80 × $log(scale)$ (Note the distinction between Context Dictionary and Coarse Dictionary) 34

6.6 Classification Accuracy(%) of Coarse Keypoint SIFT from keypoint Neighbourhoods directly used as feature that is clustered and encoded as BoW on **Fifteen Scene Categories Dataset**. Neighbourhood Size = 80 × $log(scale)$ (Note the distinction between Feature Dictionary and Coarse Dictionary) 35

6.7 Classification Accuracy(%) of Neighbourhood Coarse Keypoint SIFT BoW appended to Relatively Aligned SIFT BoW, used as global descriptor on **Fifteen Scene Categories Dataset**. Neighbourhood Size = 80 × $log(scale)$, ≈ 400 keypoints/image, Coarse Dictionary Size = 400 (Rel. Aligned SIFT Dictionary Size is the same as the Neighbourhood Coarse SIFT BoW Feature Dictionary Size) 35

6.8 Classification Accuracy(%) of *Directionally Pooled* Coarse Keypoint SIFT with appended Relative Orientation Histogram as Context for various Context Dictionary Sizes on **Fifteen Scene Categories Dataset**. Neighbourhood Size = 80×$log(scale)$, ≈ 400 keypoints/image (Note the distinction between Context Dictionary and Coarse Dictionary) 35

Bibliography

[Bansal, 2015] Bansal, A. (2015). Mid-level Elements for Object Detection. page 8.

[Belongie et al., 2002] Belongie, S., Belongie, S., Malik, J., Malik, J., Puzicha, J., and Puzicha, J. (2002). Shape Matching and Object Recognition Using Shape Contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(24):509–522.

[Bosch et al., 2007] Bosch, a., Zisserman, A., and Munoz, X. (2007). Representing shape with a spatial pyramid kernel.

[Boureau et al., 2010] Boureau, Y. L., Bach, F., LeCun, Y., and Ponce, J. (2010). Learning mid-level features for recognition. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pages 2559–2566.

[Castro et al., 2000] Castro, J. L., Mantas, C. J., and Benıtez, J. (2000). Neural networks with a continuous squashing function in the output are universal approximators. *Neural Networks*, 13(6):561–563.

[Chang and Lin, 2011] Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[Chatfield et al., 2014] Chatfield, K., Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Return of the Devil in the Details: Delving Deep into Convolutional Nets. *arXiv Prepr. arXiv ...*, pages 1–11.

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

[Deng and Runger, 2012] Deng, H. and Runger, G. (2012). Feature selection via regularized trees. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pages 1–8. IEEE.

[Doersch et al., 2013] Doersch, C., Gupta, A., and Efros, A. (2013). Mid-level visual element discovery as discriminative mode seeking. *Adv. Neural Inf. ...*, pages 1–9.

[Duchenne et al., 2011] Duchenne, O., Bach, F., Kweon, I.-S., and Ponce, J. (2011). A tensor-based algorithm for high-order graph matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(12):2383–2395.

[Felzenszwalb et al., 2008] Felzenszwalb, P., McAllester, D., and Ramanan, D. (2008). A discriminatively trained, multiscale, deformable part model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE.

[Felzenszwalb et al., 2010] Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2010). Object Detection with Discriminative Trained Part Based Models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1627–1645.

[Fergus et al., 2003] Fergus, R., Perona, P., and Zisserman, A. (2003). Object class recognition by unsupervised scale-invariant learning. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–264. IEEE.

[Figueroa, 2015] Figueroa, A. (2015). Exploring effective features for recognizing the user intent behind web queries. *Computers in Industry*, 68:162–169.

[Gao et al., 2010] Gao, S., Tsang, I. W. H., Chia, L. T., and Zhao, P. (2010). Local features are not lonely - Laplacian sparse coding for image classification. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pages 3555–3561.

[Gens, 2014] Gens, R. (2014). Deep Symmetry Networks. *Nips 2014*, pages 1–9.

[Girshick et al., 2014] Girshick, R., Iandola, F., Darrell, T., and Malik, J. (2014). Deformable Part Models are Convolutional Neural Networks.

[Hastad, 1986] Hastad, J. (1986). Almost optimal lower bounds for small depth circuits. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pages 6–20. ACM.

[He et al., 2014] He, K., Zhang, X., Ren, S., and Sun, J. (2014). Spatial Pyramid Pooling in Deep Convolutional. pages 346–361.

[Hoiem et al., 2007] Hoiem, D., Efros, A. A., and Hebert, M. (2007). Recovering surface layout from an image. *International Journal of Computer Vision*, 75(1):151–172.

[Hornik et al., 1989] Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366.

[Jaderberg et al., 2015] Jaderberg, M., Simonyan, K., Zisserman, A., et al. (2015). Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2008–2016.

[Jegou et al., 2008] Jegou, H., Douze, M., and Schmid, C. (2008). Hamming embedding and weak geometric consistency for large scale image search. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, 5302 LNCS(PART 1):304–317.

[Jégou et al., 2010] Jégou, H., Douze, M., Schmid, C., and Pérez, P. (2010). Aggregating local descriptors into a compact image representation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3304–3311. IEEE.

[Jia et al., 2014] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*.

[Kanazawa et al., 2014] Kanazawa, A., Sharma, A., and Jacobs, D. (2014). Locally scale-invariant convolutional neural networks. *arXiv preprint arXiv:1412.5104*.

[Kekeç et al., 2014] Kekeç, T., Emonet, R., Fromont, E., Trémeau, A., and Wolf, C. (2014). Contextually constrained deep networks for scene labeling. In *BMVC*.

[Krapac et al., 2011] Krapac, J., Verbeek, J., and Jurie, F. (2011). Spatial Fisher Vectors for Image Categorization.

[Kraskov et al., 2003] Kraskov, A., Stögbauer, H., Andrzejak, R. G., and Grassberger, P. (2003). Hierarchical clustering based on mutual information. *arXiv preprint q-bio/0311039*.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

[Kulkarni et al., 2015] Kulkarni, T. D., Whitney, W. F., Kohli, P., and Tenenbaum, J. (2015). Deep convolutional inverse graphics network. In *Advances in Neural Information Processing Systems*, pages 2530–2538.

[Lazebnik, 2004] Lazebnik, S. (2004). Spatial Pyramid Matching. *Work*, 3(9):401–415.

[LeCun et al., 1998] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[Leibe et al., 2004] Leibe, B., Leonardis, A., and Schiele, B. (2004). Combined Object Categorization and Segmentation with an Implicit Shape Model. *ECCV'04 Work. Stat. Learn. Comput. Vis.*, (May):1–16.

[Lindeberg, 2015] Lindeberg, T. (2015). Image matching using generalized scale-space interest points. *Journal of Mathematical Imaging and Vision*, 52(1):3–36.

[Liu et al., 2008] Liu, D., Hua, G., Viola, P., and Chen, T. (2008). Integrated feature selection and higher-order spatial feature extraction for object categorization. *26th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR*.

[Liu et al., 2012] Liu, Z., Li, H., Zhou, W., and Tian, Q. (2012). Embedding spatial context information into inverted filefor large-scale image retrieval. *MM 2012 - Proc. 20th ACM Int. Conf. Multimed.*, pages 199–208.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive Image Features from Scale Invariant Keypoints. 60(2):91–110.

[Manning et al., 2008] Manning, C. D., Raghavan, P., and Schütze, H. (2008). Scoring, term weighting and the vector space model. *Introduction to Information Retrieval*, 100:2–4.

[Meunier et al., 2010] Meunier, D., Lambiotte, R., Fornito, A., Ersche, K. D., and Bullmore, E. T. (2010). Hierarchical modularity in human brain functional networks. *Hierarchy and dynamics in neural networks*, 1:2.

[Ngiam et al., 2010] Ngiam, J., Chen, Z., Chia, D., Koh, P. W., Le, Q. V., and Ng, A. Y. (2010). Tiled convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1279–1287.

[Oliva and Torralba, 2001] Oliva, A. and Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175.

[Papandreou et al., 2015] Papandreou, G., Kokkinos, I., and Savalle, P.-A. (2015). Modeling local and global deformations in deep learning: Epitomic convolution, multiple instance learning, and sliding window detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 390–399.

[Park and Jun, 2009] Park, H.-S. and Jun, C.-H. (2009). A simple and fast algorithm for k-medoids clustering. *Expert Systems with Applications*, 36(2):3336–3341.

[Philbin et al., 2007] Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. (2007). Object retrieval with large vocabularies and fast spatial matching. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE.

[Razavian et al., 2014] Razavian, A. S., Azizpour, H., Sullivan, J., and Carlsson, S. (2014). CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. *2014 IEEE Conf. Comput. Vis. Pattern Recognit. Work.*, pages 512–519.

[Sánchez et al., 2013] Sánchez, J., Perronnin, F., Mensink, T., and Verbeek, J. (2013). Image classification with the fisher vector: Theory and practice. *Int. J. Comput. Vis.*, 105(3):222–245.

[Simonyan et al., 2013] Simonyan, K., Vedaldi, a., and Zisserman, a. (2013). Deep Fisher Networks for Large-Scale Image Classification. *Adv. Neural ...*, (iii):1–9.

[Simonyan et al., 2014] Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Learning local feature descriptors using convex optimisation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(8):1573–1585.

[Simonyan and Zisserman, 2014a] Simonyan, K. and Zisserman, A. (2014a). Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576.

[Simonyan and Zisserman, 2014b] Simonyan, K. and Zisserman, A. (2014b). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[Sivic et al., 2005] Sivic, J., Russell, B. C., Efros, A. A., Zisserman, A., and Freeman, W. T. (2005). Discovering objects and their location in images. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 370–377. IEEE.

[Sivic and Zisserman, 2003] Sivic, J. and Zisserman, A. (2003). Video google: A text retrieval approach to object matching in videos. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1470–1477. IEEE.

[Sohn and Lee, 2012] Sohn, K. and Lee, H. (2012). Learning invariant representations with local transformations. *arXiv preprint arXiv:1206.6418*.

[Sparck Jones, 1972] Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

[Srivastava et al., 2014] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

[Tewari and Bartlett, 2007] Tewari, A. and Bartlett, P. L. (2007). On the consistency of multiclass classification methods. *The Journal of Machine Learning Research*, 8:1007–1025.

[Vedaldi and Fulkerson, 2008] Vedaldi, A. and Fulkerson, B. (2008). VLFeat: An open and portable library of computer vision algorithms.

[Verbeek, 2012] Verbeek, J. (2012). Fisher vector image representation Fisher vector representation.

[Vu et al., 2015] Vu, T.-H., Osokin, A., and Laptev, I. (2015). Context-aware cnns for person head detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2893–2901.

[Wan et al., 2015] Wan, L., Eigen, D., and Fergus, R. (2015). End-to-end integration of a convolution network, deformable parts model and non-maximum suppression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 851–859.

[Wang et al., 2011] Wang, Z., Cui, J., Zha, H., Kegesawa, M., and Ikeuchi, K. (2011). Object Detection by Common Fate Hough Transform. pages 613–617.

[Wu, 2015] Wu, X. (2015). Robust Spatial Matching as Ensemble of Weak Geometric Relations. 1:1–12.

[Xiaomeng Wu, 2014] Xiaomeng Wu, K. K. (2014). Image Retrieval Based on Spatial Context With Relaxed Gabriel Graph Pyramid. pages 6879–6883.

[Zhang et al., 2011a] Zhang, S., Tian, Q., Hua, G., Huang, Q., and Gao, W. (2011a). Generating descriptive visual words and visual phrases for large-scale image applications. *IEEE Trans. Image Process.*, 20(9):2664–2677.

[Zhang et al., 2011b] Zhang, Y., Jia, Z., and Chen, T. (2011b). Image retrieval with geometry-preserving visual phrases. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pages 809–816.

[Zhao et al., 2015] Zhao, R., Ouyang, W., Li, H., and Wang, X. (2015). Saliency detection by multi-context deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1265–1274.