LAYOUT-AWARE MIXTURE MODELS FOR PATCH-BASED IMAGE REPRESENTATION AND ANALYSIS

BY

XINQI CHU

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2014

Urbana, Illinois

Doctoral Committee:

Professor Thomas S. Huang, Chair
Professor Mark Hasegawa-Johnson
Associate Professor Feng Liang
Assistant Professor Paris Smaragdis

ABSTRACT

Image and video representation and modeling is an important topic in computer vision and image processing. An image model provides an abstraction of the large amount of data contained in an image and enables the systematic development of algorithms for accomplishing a particular image-related task, such as detection, recognition and segmentation (analysis) as well as inpainting, summarization and colorization (synthesis). Since an image is usually comprised of millions of pixels, developing models in such a high dimensional space is not always feasible. One of the most popular ways of modeling images is to break them into patches; the reason is that not only is the dimensionality reduced, but it is easier to define similarities between patches as they experience less distortion as compared with defining similarity between images. Patch-based image models are often more flexible in modeling appearances by exploring redundancies in image and videos. By adjusting the patch size, these models trade off the good qualities of each end of the spectrum - the discriminative power of images and the representational power of pixel histograms. When breaking an image into a collection of patches, one must be able to model two kinds of information in order to describe the image completely. On one hand, one must be able to model the patch appearance with some statistical model; on the other hand, there must be some other statistics to describe how the patches are organized together in an image. We call the first kind the "appearance model" and the second the "layout model". In this thesis, we describe the historical progress made in the past decade starting from patch-based appearance models without considering layout information, onto how spatial modeling improves performance and enables applications in analysis tasks such as recognition, detection and segmentation as well as synthesis tasks such as colorization by explaining our works in the past three years. This thesis proposes both a discriminative formulation as well as a generative formulation in describing patch layouts.

The algorithm developed upon the discriminative framework achieves state-of-the-art results in the joint detection and its subcategory recognition problem. Algorithms developed for these models are also discussed in the process with results and examples.

To my parents, for their love and support.

ACKNOWLEDGMENTS

I would like to express my deepest appreciation and sincere gratitude to my supervisors, Professor Thomas S. Huang and Research Scientist Tian-Tsong Ng from A*STAR, for their invaluable advice and great support. These individuals encouraged me to get through the tough time during the research. I not only learned specialized knowledge from them, but also the spirit of exploring scientific problems. This work would not have been possible without their excellent guidance and persistent encouragement. My gratitude also goes to my PhD committee members, Professor Mark Hasegawa-Johnson, Professor Feng Liang and Professor Paris Smaragdis of the University of Illinois at Urbana-Champaign, with whom I took courses during the first few years of my PhD studies, and who have continued to offer guidance in my work.

I also would like to thank my collaborators, Associate Professor Chan Kap Luk, Associate Professor Shuicheng Yan, Dr. Liyuan Li, Yingzhen Yang, Zhaowen Wang, Xiaoyuan Yu and Jianchao Yang, for their intriguing insights and valuable suggestions.

I am grateful to the Department of Electrical and Computer Engineering (ECE), University of Illinois at Urbana-Champaign, for kindly providing excellent courses, resources and other facilities for my research.

Finally, I would like to thank my parents and my friends Nan Shen, Eik Shern Lee, Honghai Yu, Xin Zhang, Jing Shen, Benjamin Chidester, Patrick Johnstone and Chang Qu for their kind support.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Image and video representation and modeling has been an important topic in computer vision and image processing. Image models can help to develop algorithms for specific image-related tasks or several related tasks at hand, for example, object detection and object recognition. These two problems have been studied for many years due to both scientific and industrial interests. The two problems have been studied extensively but separately until recently when the detection and recognition features started to be jointly learned in [1] from a multi-task deep learning perspective [2]. Feature learning strategies such as dictionary learning and sparse coding [3, 4] and especially deep learning [5] have achieved huge success in recent years in the big data era. One may have the impression that as long as the dataset is huge, deep learning by itself can be the strategy that achieves the state-of-the-art performance. In this thesis, we argue that not only is feature learning important for the joint recognition and detection problem, but also the normalization strategies with respect to layouts of the patches in the image. We show that by normalizing across layouts of image patches in a discriminative setting, one can achieve even better results than treating deep learning as a black box. And it also improves various performance benchmarks in a generative setting.

Our layout normalization strategy not only improves performances of detection and recognition results, but also has the potential to provide information such as pose, which is important in industries such as retail analytics as well as inferring human behaviors.

Hence, we motivate our layout normalization strategy from two aspects. Firstly in section 1.1 we shall look at several real industrial examples that motivate the joint detection and recognition framework with our layout normalization strategy, which has the potential to provide additional related information which is important to particular industries. And then we show that the normalization strategy with respect to layouts, complementary to the feature learning framework, can improve the bias-variance trade-off [6] by a significant portion and therefore yields better performance on detection and recognition while the normalization and feature extraction process are only performed once.

1.1 Motivations

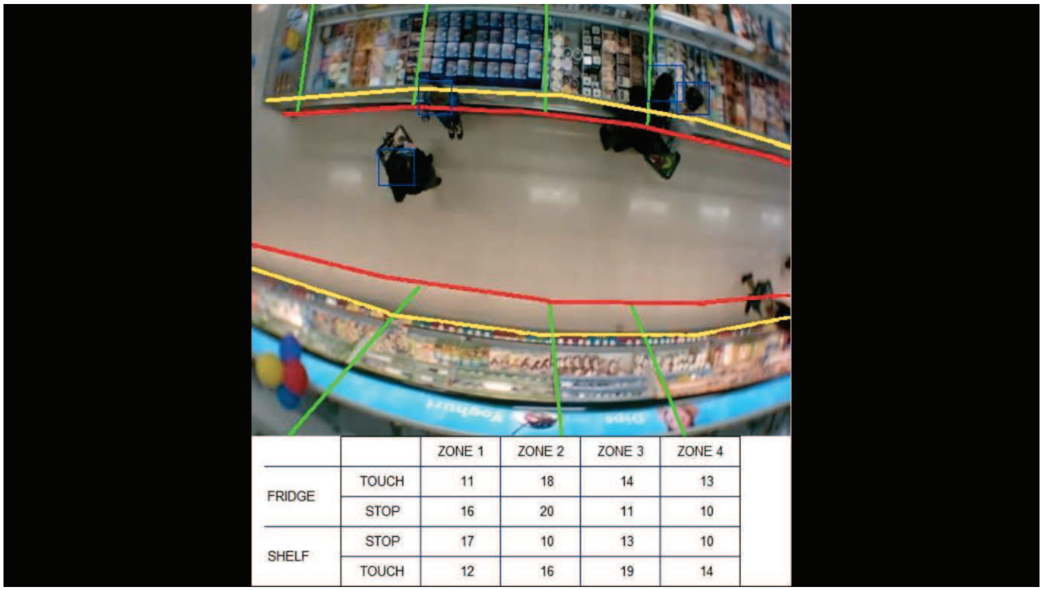
The video analytics industry is burgeoning with applications in retail analytics as shown in Figure 1.1, government and industrial surveillance (as shown in Figure 1.2), traffic analysis and so on. The challenge here is to harness the real big data in video storage to ensure safety as well as providing supporting evidences and insights for business as well as government decisions. In retail analytics, owners hope to analyze shopper behavior in order to enhance sales as shown in Figure 1.1. This type of application requires detecting and tracking people in surveillance cameras from overhead position and also detecting parts such as arms. Furthermore, the orientation of the body is also important to indicate the type of product that the customer may be looking at. The orientation of the body can be derived if we consider and estimate pose while performing detection and recognition.

1.1.1 Subcategory recognition

Apart from retail analytics, government or industrial tender also calls for subcategory recognition in which the differences between the subcategories appear to be more subtle than the intra-class variations.

Here we refer to the requirement specification of a recent tender, the mobile analytics system, which involves multiple mobile cameras. Tracking across camera networks has been a longstanding goal in surveillance; the system in this tender requires the following two items as quoted from the tender:

- Facial recognition. The video analytics system shall perform near-real-time facial recognition on video footages streamed from body-worn cameras and in-vehicle camera systems against a database of at least one thousand (1,000) subjects.



		ZONE 1	ZONE 2	ZONE 3	ZONE 4
FRIDGE	TOUCH	11	18	14	13
	STOP	16	20	11	10
SHELF	STOP	17	10	13	10
	TOUCH	12	16	19	14

Figure 1.1: Object detection and recognition applied to retail analytics where shop owners are provided with information such as the number of people who have stopped by or touched certain products.

- Repeated subject appearance. The video analytics system shall be able to detect and index a subject who is not in the aforementioned database and who makes repeated appearances in the field of view of the camera network.

Both requirements following the same vein as a single category must be detected from the video; after that a subcategory recognition system must be able to distinguish between different sub-classes that share similar appearances for which in later chapters we show that a normalization strategy achieves better performance if we consider the problems together.

1.1.2 Attribute analysis

Another recent tender, “Leasing of a video trawling and analytics system” asks for describing “physical attributes of a person, which include, but not limited to the following: age, gender, headdress; sunglasses; and backpacks, colours, clothe patterns, textures.” Not to our surprise, the state-of-the-art attribute modeling framework [7] uses a pose normalized part-based model on a deep neural network.

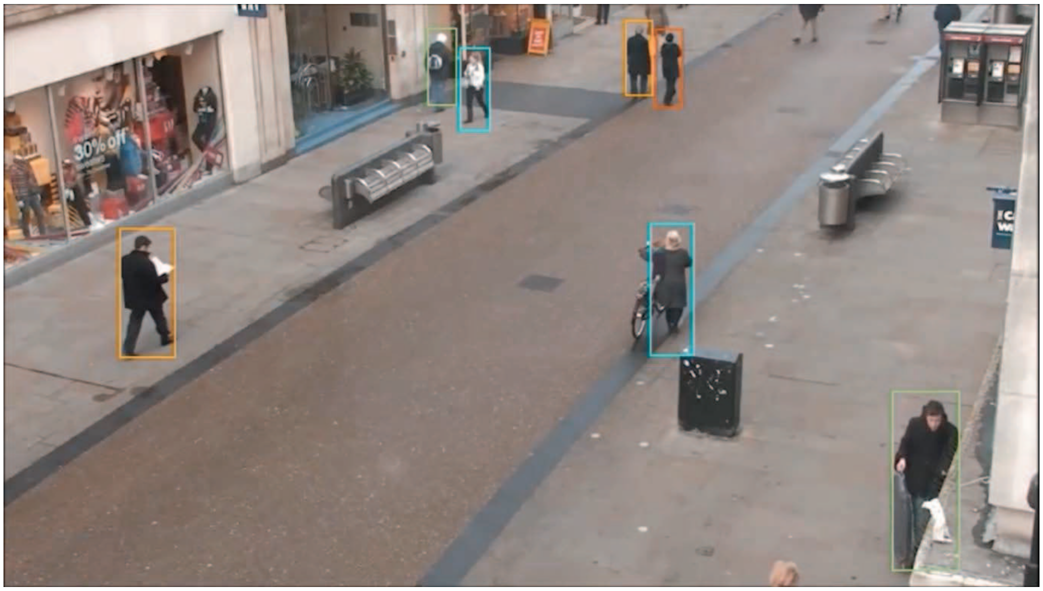


Figure 1.2: Joint detection and person recognition for surveillance application.

1.1.3 Pose and behavior estimation

In the context of pose and behavior estimation, a sample of industry requirement is as follows:

- The system shall detect a subject waving at the system.
- The system shall detect a subject with his head completely occluded from the Surveillance camera(s).
- The system shall detect a subject who is banging his head against the wall or door.
- The system shall detect a subject who is kicking the wall or door.

This system must be able to recognize all the joints of a human body before these items can be realized. These joints are then estimated robustly with the algorithm [8] that is implemented in the Kinect system, not surprisingly, which also uses parts for training which means pose normalization and estimation are performed.

1.1.4 Multiple requirements in one software

It can be readily observed from the previous sections that a full-fledged system that includes detection, recognition and tracking components, which are required for the full solution, and often, these requirements have to be carried out in real time. As feature extraction is the most time consuming part, it is required that the feature extraction and normalization part of the algorithm can only be performed once and for all. In the following section, we shall discuss in depth the reason why layout normalization procedures are complementary to powerful feature learning and as important for achieving the state-of-the-art in the joint detection and its subcategory recognition system.

1.2 Layout Normalizations and Patch-Based Systems

Achieving high detection and recognition accuracy has been at the core of the goals of the computer vision field for many years. High accuracy can be achieved via a good feature representation which can maximize the inter-class distance while keeping the intra-class distance small. The ideal feature representation should allow zero intra-class distance and infinite inter-class distance as stated in the kernel target alignment work [9]. Recent advances on deep learning show that, as we go up the feature hierarchy, images with same labels tend to cluster together whereas images with different labels tend to be away from each other [10]. In the same vein, [11] achieves state-of-the-art performance in face identification by explicitly maximizing and minimizing inter-class distance and intra-class distance respectively.

Another way to maximize inter-class distance and minimize intra-class distance is to break the image into patches, and cluster the patches with similar appearances while keeping dissimilar patches apart from each other. Patch-based image and video representation has been a prevailing way of performing various tasks in the image processing and computer vision community for the past decade. In 1999, Leung and Efros [12] convincingly demonstrated the capability of synthesizing a large class of textures by sampling patches from a reference image. Since then, patch-based image models have shown increasingly more vitality and have found many other successful image processing applications, a few of the prominent ones being image inpainting [13, 14], image denoising [15], image super-resolution [16] and so on. A large amount of work in computer vision representations and applications has come to use patch as the basic operating element. Why is patch-based representation such a good idea? It is because local patches often experience much less distortion than global images and therefore it becomes easier to define the similarity between two local patches. This idea is essentially the key to the success of a class of local keypoint descriptors such as SIFT [17], local binary pattern (LBP) [18] and SURF[19], which are essentially based on image patches.

Patch-based representation is a trade-off between template and histogram representations, and the parameter that defines this trade-off is the patch size. Pixel layout information is inherently related to this trade-off parameterized by patch size. In a template-based method, layout information is completely retained, but its generalization ability suffers and can only represent a limited amount of images. As the patch size gets smaller, similar patches occur with higher and higher frequency, until it reaches the other end of the spectrum, i.e. histogram-based representation of an image with just 256 possibilities of occurrence (assuming 8 bit), though the spatial information is completely lost; however, it can represent a very large number of images and it is invariant to many forms of transformations, e.g. rotation, translation, etc. By choosing the patch size sensibly, we are able to reduce the variations between intra-class distances and increase the inter-class distances.

1.3 Variations, Invariance, Normalization

Feature extraction methods are traditionally designed for capturing high variance components, such as principal component analysis, which provides a projection of each input vector to a low-dimensional coordinate vector. Variations in the directions of the principal components are perfectly captured by the PCA, whereas changes in the directions that are orthogonal to the principle components are lost. The assumption that we use PCA is that directions where there is very little change in the data do not matter and can be considered noise. However, the directions with low-variance need not be uninformative; in fact in practice, these directions can bear very important information in distinguishing between subcategories. For example in face recognition, the detected faces from videos often have mis-alignment, pose variations as well as scaling differences, in this scenario, as most variations in the pixel space will be explained by these causes, so identity bearing information will lie in the low-variance components. While cutting the low-variance directions out it is quite possible that the most important information that can distinguish between different faces is lost. This is the reason that alignment must be performed in face recognition tasks. In the recent deep face work [20], the authors also show that an alignment step is crucial to large scale face verification. This empirically implies that deep learning is also not able to discover distinguishing information while large variations exist. The alignment step in this work [20] is a normalization step that reduces the intra-class distance while increases the inter-class distance.

The notion of invariance involves discounting variations in image detection and recognition tasks. We summarize here two related approaches that can be used to achieve invariance: the informative invariance that we achieve by normalization, i.e. to estimate the dominant orientation of the gradient and extract feature by normalizing the image to the dominant direction, and the uninformative invariance for which we find an invariant operator that maps images subject to the same type of variation to one target vector in the output space (for example, by dividing a color vector by the sum of its components). Uninformative invariance is also achieved by representing the image by its histogram, or by modifying the training data set of a classifier by adding jittered samples.

Uninformative invariance By the definition in [21] and restated in [22], an invariant operator is a non-invertible function on the space of features x which maps all features belonging to an orbit of the group actions into the same point. Its trade-off between discriminative power and invariance is studied in [22], which concluded that, from an information-theoretic point of view, imposing invariance results in reduced (rather than improved) system performance. The example here is a face verification scenario when the training images are not aligned. A descriptor can be designed to be viewpoint invariant, which means the statistical description of the descriptor is simpler. This will result in high bias classifier (not enough parameters to describe appearances), which has higher potential to lose discriminative details. The authors of [22],[23] try to learn an optimal trade-off between discriminative power and invariance of a classifier.

Another common approach is to feed images of different variations into a learning machine, and hope the invariant properties can be learned. This approach is the prevailing one now in image classification [24, 25]. This type of approach may not be as good as its informative counterpart in capturing discriminative variation components. For example, in face verification [20], suppose instead of performing alignment, one can feed all the labeled but unaligned images into the network classifier. If our classification model is too "simple" and has very few parameters, then it may have large bias (but small variance) and the learned feature may lose identity bearing information; if we design the model to be too "complex" with many parameters to describe the large variations, then it may suffer from overfitting (but have smaller bias) if we do not have enough data samples. Another example is that, if we have a large amount of different pose configurations of pedestrians, we need a large amount of training data to be able to learn a dense complex manifold that spans the space of all pose variations. This complex classifier will have more parameters and thus requires more samples to train. These viewpoints are also argued in [2], which states that as generalization is mostly achieved by a form of local interpolation between neighboring training examples that is based on the smoothness assumption in the data manifold, it is insufficient to deal with the curse of dimensionality, because as the number of interacting variation factors such as pose, viewpoints and illumination grow, the complexity of the target function may grow exponentially with the number of relevant interacting factors, which needs exponentially more samples according to the sample complexity bounds. As the number of training samples goes up, the variance of the classifier goes down and subsequently yields lower generalization error.

Informative Invariance Invariance can also be achieved by normalization procedures, such as breaking the object into parts as in pose normalization. The set of parts to detect is much more common in appearance: first, they are smaller in size and thus appear similar due to natural invariance to viewpoint deformations and secondly, though the number of classes will increase, by decomposing the object into parts, one will have part-models that are trained by as many samples as there are for the object. In this way, the manifold of appearance models becomes simpler, and at the same time can be trained by the same number of training samples. According to the sample complexity bound, at high probability, the generalization error will decrease exponentially with respect to the number of training samples. Furthermore, by training models with respect to parts, one is able to estimate the pose parameters during the testing time, which are useful and important information that needs to be extracted out from images and videos.

It is important to distinguish between the related but distinct goals of uninformative invariance, which is the learning invariant features rather than to learn to disentangle explanatory factors. One important difference is the preservation of information. Invariant features, by definition, have reduced sensitivity in the direction of invariance with a non-invertible mapping, which may lose discriminative information which is essential to achieve high accuracy, and furthermore, we may not be able to get the information about that variation parameter, for example the pose parameters or the configuration of parts.

Variations are common among different tasks Although detection and subcategory recognition are different tasks, the variations that we face here are exactly the same, such as pose variations, viewpoint, illumination, local translation, scale. Here in this thesis we propose to handle these variations once and for all the tasks at hand, namely pose normalization as well as translation normalization.

Multi-task features In order to save computation further, we propose to perform feature extraction only once throughout the image and use the extracted features for both detection and recognition. Here we use Overfeat [1] in this thesis as it is trained in a multi-task scenario. In multi-task feature learning works [2], different tasks with supervisions can help to disentangle the information better as multi-task learning exploit commonalities between different learning tasks in order to share statistical strength by learning representations that capture underlying factors. This hypothesis seems to be confirmed by a number of empirical results.

1.4 Generative Layout Models

The bag-of-words model [26],[27] balances the advantages of both sides and uses a histogram of patches as features for image classification. The BoF model treats an image as a loose collection of unordered appearance descriptors (e.g., the SIFT descriptor[17]) extracted from local patches, quantizes them into discrete visual words, and then computes a compact histogram representation for semantic image classification. However, the BoF model discards all the global layout information of the local descriptors, which is informative or even crucial for discriminative analysis, and therefore its performance is limited. To be able to describe global patch layout, spatial pyramid matching (SPM) [28] has been employed and become extremely popular. The SPM method partitions the image into increasingly finer spatial sub-regions and computes histograms of local descriptors from each sub-region. The resulting spatial pyramid representation is a computationally efficient extension of the orderless BoF model, and has shown very promising performance in various benchmarks. Since then, modelling spatial information has become a trend in image analysis. [29] improved the SPM method by max pooling over the sparse representations of the local descriptors with respect to a learned over-complete dictionary. Perronnin and Dance [30] proposed a Fisher kernel approach based on the Gaussian mixture model (GMM) to aggregate the set of descriptors by considering their zero-, first- and second-order statistics. By mapping the image into a high-dimensional feature space, the obtained image feature works rather well with linear classifiers. Zhou et al. [31] proposed a similar approach by modelling the spatial information from global and local statistics of a Gaussian mixture model.

In Gaussian mixture model, the parameters of the Gaussian components are not shared; sharing of parameters can allow a better generalization of the Gaussian mixture model. The work epitome was proposed by another senior Nebojsa by allowing the GMM to share their parameters across the individual Gaussian components, which resembles panorama stitching work in the image domain; therefore in this work it is called parameter panoramization. It allows the model to represent larger and more sophisticated patterns with a smaller parameter space with less variance. These models are learned by compiling patches drawn from input images into a condensed image model. It was shown in [32] that the image epitome is an image summary of high "completeness." The epitome idea has also found its use in representing audio information [33] and human activities [34]. In [32], the image frames from a panoramic video are automatically stitched together to form a panorama due to epitome's ability in exploring image similarities. However these panoramic parametric models did not try to model spatial information; therefore, when it comes to classification related tasks, the performance suffers [35]. It also does not preserve the original layout while it generates the summary and many applications beyond that.

1.5 Contributions

This thesis makes four main contributions to the goal of modeling spatial patch layout for the tasks in detection and recognition. Firstly, it proposes a latent variable based layout normalization framework which delivers state-of-the-art results in the problem of joint detection and recognition with a discriminative model. Secondly, for analysis applications such as recognition and detection in a generative setting, it proposes a principled way of modeling spatial layout of patches in the existing panoramic GMM model and provides an algorithm to perform learning and inference; these algorithms significantly improve the performance of the original work. After that, for synthesis applications such as colorization that require spatially smooth results, we propose to consider inter-patch layout during the inference process.

1.6 Thesis Outline

Discriminative layout model The first part of the thesis (Chapters 2-5) proposes a normalization framework which improves the bias-variance tradeoff of classifiers for both detection and its subcategory recognition. The model is trained in a discriminative setting by a latent SVM via iterative refinement process to improve detection precision. The work is evaluated on the bird dataset [36] for the joint detection and recognition task and has achieved state-of-the-art performance.

Generative layout model In Chapters 6-9, we [35] propose to model patch global layout in generative probabilistic model where the parameters specify the spatial distributions of the patches in the panoramic GMM model. Therefore the extended model is able to achieve good performance in various inverse problems such as image completion, misalignment face recognition and object detection.

Inter-patch Layout Chapter 10 discusses the colorization application by considering inter-patch layouts. Patches of similar texture can be from very different parts of the object; in many applications such as image classification and summarization we do not need to distinguish because that information is accurate while essential statistics are extracted to perform the analysis. However, sometimes it is necessary to impose the smoothness constraint to encourage neighboring patches to have similar labels. This constraint, which is implemented by an MRF, is essential to our work in image automatic colorization [37] especially while two regions that are of similar textures are from different parts of an object. The layout-smooth constraint, which makes sure that each region is consistently colorized and results in a more pleasant result visually, is common among graphics applications such as seam carving [38] and content-driven retargeting [39]. We also extended this work to a more general learning problem in the same year [40].

Chapter 11 concludes this thesis.

CHAPTER 2

LATENT PATCH LAYOUT MODEL FOR JOINT DETECTION AND SUBCATEGORY RECOGNITION

Single object-category detection and its corresponding subcategory recognition are two related important research directions that are shown to be the problem abstraction of following three research problems as well as industrial system pipelines:

1. Face detection and verification
2. Person detection and re-identification
3. Fine-grained categorization

These problems have been extensively motivated in Chapter 1 due to their industrial significance.

2.1 Characteristics of the Joint Detection and Subcategory Recognition Problem

To be able to achieve high accuracy in the problem of joint detection and recognition, we found that the problem has several characteristics once the two component problems are put together.

Challenges faced by the two problems are common Both detection and recognition tasks require us to account for variations such as pose, illumination, viewpoint and translation; in this work, these variations are dealt with once and for all for the two connected problems. Namely, we address a latent variable model for translation normalization, part layout model for pose normalization and a mixture model for viewpoint normalization. The three normalization schemes are applied once to the system for both detection and recognition problems.

Importance of detection/localization precision in the pipeline Among the two components in this pipeline, detection and recognition, the latter is dependent on the former. Current detection research [41, 42, 43, 44, 45] evaluates detection algorithms using protocols such as PASCAL VOC 2006, 2007, 2010 [46, 47]. These datasets provide ground-truth bounding boxes for a number of object classes. At test time, the goal is to predict the bounding boxes of all objects for a particular class in an image or report that the object class is not found. The system will output a set of predicted bounding boxes with corresponding scores. A predicted bounding box is considered correct if it overlaps more than 50 percent with a ground-truth bounding box; otherwise, the bounding box is considered a false positive detection.

Under this evaluation criterion, a good detector does not necessarily constitute a good initialization for the later recognition task, because the misalignment will induce large translation variations for the recognizer, especially in the fine-grained case such as the capability to differentiate between pedestrians, faces as well as subcategories of birds, as the appearance variations induced by translation will be larger than the differences between the subcategories. Hence an accurate and precise detection is even more important; otherwise, the error will propagate to the recognition stage which renders poor recognition performance. Therefore in this work we propose a latent model for "good positive mining" in the detection phase to perform concurrently with "hard negative mining" for precise detection of objects.

Task dependent selection of deep features Detection aims to distinguish the foreground objects from the background whereas subcategory recognition is set to differentiate between objects (cropped out by detections) from the same category. First of all, the features that we use must be different for the two different tasks. The feature that is used for detection must be able to distinguish the basic category and the background whereas the feature that is used for recognition must be able to differentiate between the parts of subcategories. Feature learning of the joint detection and recognition problem has recently been considered in the deep learning literature [1], which shows that the different tasks can be learned simultaneously using a single shared network and demonstrate that the same feature can be used in tasks such as detection, recognition and localization and achieve high accuracy on the ImageNet challenge [25]. This is the work that is most similar to this thesis. However this work [1] focuses on the feature learning side of the problem by sharing the bottom feature extraction layers for different tasks which is also envisioned in [2]. This recent work does not consider the problem such as importance of normalization operations that deals with subcategory dissimilarities, nor does it explore single-category similarities through a better Bias-Variance trade-off as discussed in 1. Moreover their evaluation dataset ILSVRC13 is mostly comprised of basic-level categories; it is not designed for evaluating the problem of the joint single category detection and subcategory recognition.

We approach the subcategory detection and recognition problem from a vision system perspective by exploring same-category similarities and propose a new mechanism for the industrial system pipeline.

As feature per say, traditional pipelines such as GRID often use different features for detection (Haar) and recognition (Eigenface); this requires the system to perform feature extraction twice through the data at the detected regions. Nevertheless, the Overfeat work [1] trains a feature extractor through a convolutional network to simultaneously classify, locate and detect objects in images which can boost the classification accuracy and the detection and localization accuracy of all tasks. Also another recent work that performs extensive evaluation for deep features [48] shows that features learned through a deep neural network (only classifiers are different) are almost all good for all tasks including object detection, basic-category recognition as well as subcategory recognition (fine-grained recognition). In light of these evidences, our approach uses pretrained deep network features (Overfeat [1]) which are then trained and selected by SVMs for different tasks, namely detection and recognition. Hence the expensive feature extraction computation is only performed once through the data, while the resulting feature vectors can be used by two tasks at the same time.

2.2 Normalization-based learning for joint detection and recognition

In this thesis, we choose to perform normalization against different variations instead of using an invariant classifier as a black box. Among all the variations that are detrimental to detection and recognition performances,

Table 2.1: Traditional research problems

Research Problem	Evaluation assumptions	References
Detection	50% overlap with groundtruth,	Pascal VOC 2010
Subcategory recognition	bounding box is given	[49, 50]
Classification	predicted labels only	ILSVRC 2013
Localization	object presence given	ILSVRC 2013

pose/articulation, viewpoint and translation are the three that have the most effect on the detection and recognition accuracy. In this work we propose normalizations can be used effectively to account for these different variations and improve performance for the joint detection-recognition pipeline. The normalization schemes we adopt here are pose normalization, viewpoint normalization and translation normalization.

Especially for translation normalization, the concept of "good positive mining" is proposed and operated concurrently with the hard negative mining to accurately position the bounding boxes of the parts in our new joint detection-recognition learning algorithm 1.

Here we discuss the assumptions and evaluation methods of the traditional research problem of detection, subcategory recognition, classification and localization, and compare with the research problem when all of these independent problems are considered jointly.

As in Table 2.1, for a detection to be considered correct, the predicted box must match the groundtruth by at least 50 percent (using the PASCAL criterion of union over intersection), regardless of the subsequent tasks such as subcategory recognition which requires precise alignment from the detection result. However, the current subcategory recognition works[49][50] [51][52][53][54] assumes that an accurate bounding box is given before the subcategory recognition task is performed while such information is provided in the subcategory recognition dataset [36]. As subcategory recognition relies heavily on the precise localization of object parts, there seems to be a gap between the detection-recognition system pipeline. Another difference with traditional object detection is that, although we are able to detect and recognize multiple subcategories, there is only one general category that we propose to detect. This specific problem allows us to explore similar structures that are shared between different subcategories and also avoid building separate detectors for each subcategory which is computationally expensive.

Image classification on the ImageNet [24] database regards the accuracy of predicted image labels as the evaluation criteria irrespective of localization as well as detection. Similarly, the localization task [55] assumes object presence information is given by the image classification/detection task and sometimes assumes that there is one type of class in each image [1, 56], hence the localization operation is often performed after classification or detection. The detection task differs from localization in that there can be any number of objects in each image (including zero)[1].

The normalization framework proposed here considers the joint problem of detection and recognition, in which we account for the imprecise localization problem from traditional detection research with a latent variable model that also performs pose normalization, translation normalization and viewpoint normalization that helps with all tasks such as detection, recognition and localization. These normalizations are only performed once for all the tasks to achieve high accuracy in detection, localization and subcategory recognition.

2.2.1 Pose Normalization

According to research in human kinetics [57], the human body has 244 degrees of freedom. There are around 230 joints in the body, most of which have one degree of freedom (DoF). Some joints have multiple degrees of freedom (DoF) - for example, the hip and the shoulder joints have at least 3 DoF. A bird has many degrees of freedom too, such as the different orientations while the bird is flying. The visual differences between the same type of bird with wings open and closed is arguably much larger than the difference between different types of birds as we can observe in [36]. To achieve pose normalization in detection and recognition, a part-based approach is generally adopted. This normalization framework for joint detection and recognition is illustrated in Figure 2.1.

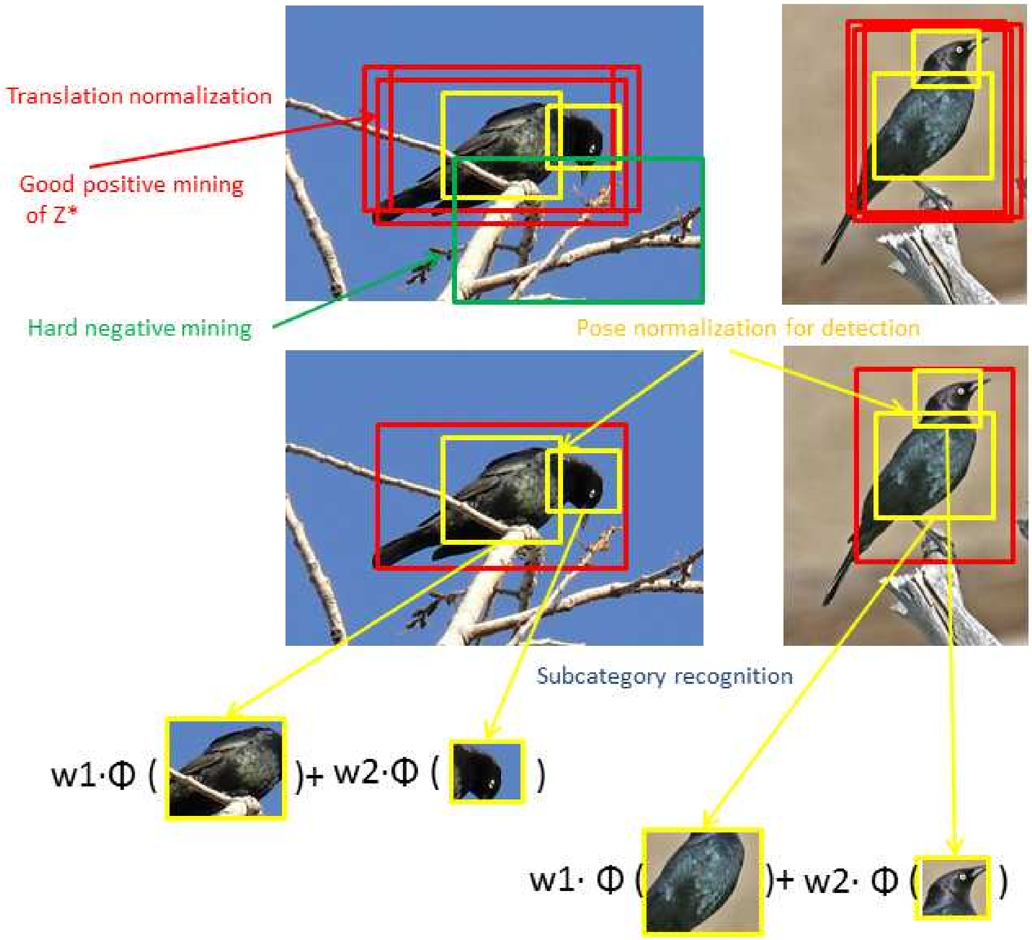

Translation normalization
Good positive mining of Z*
Hard negative mining
Pose normalization for detection
Subcategory recognition
w1·Φ ( )+ w2·Φ ( )
w1· Φ ( )+ w2· Φ ( )

Figure 2.1: Joint detection and subcategory recognition results from four different viewpoints.

Part-based pose normalization for subcategory recognition The premise of fine-grained classification is the assumption that the objects of a super-category share common shape configuration among all subclasses. Research in cognitive psychology [58] has suggested that fine-grained recognition relies on identifying the subtle differences in appearance of specific object parts. Recent works in computer vision have shown the part-based mechanism to be an effective approach for fine-grained recognition [50, 51], Birdlets [52], DPD [53] and POOF [54] in which descriptor for each pair of keypoints is learned for discriminative mid-level features. In [52], the authors proposed a pose-normalized representation for recognition using poselets [44], whereas deformable part models [42] were used in [53] for part localization. In recent work [49] part annotations are transferred from objects with similar global by a nonparametric label transfer technique. Application of deep features also uses a part-based pose normalization for attribute prediction [7].

Part-based pose normalization for detection It has also been shown that part-based pose normalization is an effective approach for object detection, starting with explicit modeling of body parts with pictorial structures [59, 60] and the later poselets [44, 61] and related works. These strongly supervised models often suffer from imprecise human labeling of part bounding boxes. Through implicit modeling of object parts, the deformable parts model (DPM) was proposed [42] and won the PASCAL challenge in 2009. This work models parts with additional learned filters in positions anchored with respect to the whole object bounding box, allowing parts to be displaced from this anchor with learned deformation costs. However, as the optimization is non-convex, the results are not always optimal as mentioned in the strongly supervised DPM [62], which adapted the weakly supervised method [42] for the strongly supervised setting in which part locations are annotated at training time to improve initialization of the optimization.

2.2.2 Translation normalization

Normalizing translations and performing alignment during training is important to both the task of object detection and its subcategory recognition. The other approach to deal with the issue of translation is to feed in the classifier with a lot of translated examples to make it translation invariant; however, the object appearance manifold coupled with its translated variants can form a space where the manifold is discontinuous so that translated objects are not accounted for in the learned classifier, especially when the classifier is of a large capacity [63]. Hence, translation normalization, a.k.a. precise alignments, are better suited for variations induced by different translations. In the subsequent paragraphs we argue that translation normalization operation is beneficial for both object detection as well as its subcategory recognition.

Translation normalization for detection Human are better at qualitative and conceptual decisions while machines are better at decisions that requires very high precision. Detectors are trained with bounding boxes that are labeled by human subjects. Hence the bounding boxes that are used during the training phase are not precise. This is shown experimentally in Figure 6 of the DPM work [64]. This result shows that, when a part-based model is not used, a latent correction operation of human labeled bounding boxes can improve the detection by 33%. However their work allows the parts to be discovered by the algorithm which loses the semantic meaning and its optimization is sensitive to poor initialization.

Latent good positive and hard negative mining To achieve translation normalization, we use a latent variable approach for which we use the term "good positive mining" in contrast to the "hard negative mining" [65, 42] in the detection literature. To achieve a very low false positive rate, hard negative mining is often used to retrain the detection system iteratively. In this iterative process, an initial model is trained using all positive examples and a randomly selected subset of negative examples, and this initial training set is progressively augmented with false positive examples produced while scanning the images with the model learned so far [65]. While mining hard negatives is important to remove false alarms, we argue that good positive must also be mined to achieve high localization performance. In each iterative process, the initial model is used to generate updated bounding boxes which are in turn to be used as the new set of positive examples for the new detector training. In this thesis, good positive mining and hard negative mining are performed concurrently and iteratively following an E-M style as presented in Algorithm 1.

Translation normalization for recognition Translation normalization literature is most extensive in the area of image alignment and face recognition. As a subcategory recognition problem, face recognition, face alignment is often performed as an intermediate step after face detection. In image alignment, it is often shown that unsupervised alignment performs better than its supervised counterparts [66, 67, 68, 69, 70, 71, 72]. These alignment methods presume that the object class as well as an initial bounding box is given beforehand, and most of these methods are computationally expensive.

Hence we propose that a good positive mining approach should directly solve the alignment problem in the detection phase through good positive mining through a unsupervised latent approach. The recent deep face work [20] also shows that an alignment step is crucial to large scale face verification. It is also supported by empirical evidence that the unsupervised alignments often perform better than supervised ones [73] in subcategory recognition tasks.

Contribution To summarize our contributions, we firstly propose a normalization framework to account for variations such as pose and translation in the industrially motivated joint detection and recognition problem. We formulate and implement a latent variable model which yields a positive-negative retraining algorithm for mining good positives and hard negatives concurrently for translation normalization. In the end we provide evaluation on the Caltech-UCSD Birds-200-2011 [36] and our method is compared favorably with other state-of-the-art methods on the joint detection-recognition problem. The rest of the related chapters are organized as follows. We describe the normalization framework and apply it to a joint detection and recognition problem in Chapter 3 and Chapter 4, Experimental evaluation of the method is presented in Chapter 5.

CHAPTER 3

JOINT DETECTION AND RECOGNITION WITH LATENT PART LAYOUT

The part-based latent layout model is proposed here for both detection and subcategory recognition. Here we roughly follow the notation of [64, 42] and introduce more notations and equations when necessary. Let $P = [P_0, P_1 ... P_n]$ denote the object P_0 and its n parts where $P_i = \{w_i^d, w_i^r, v_i, d_i\}_{i=1}^n$. Here the w_i^d and w_i^r refers to the detection weights and the recognition weights that are learned through different objective functions during the training stage. $[w_0^d, w_1^d, ..., w_n^d]$ denotes the corresponding whole-object SVM detection weight and its part SVM detection weights. v_i is a two-dimensional vector specifying an anchor position for part P_i relative to the root position, and d_i is a four-dimensional vector specifying coefficients of a quadratic function defining a deformation cost for each possible placement of the part relative to the anchor position. Let $Z = [z_0, z_1, ..., z_n]$ denote the a placement of bounding boxes of object P_0 and its n parts, where $z_i = [x_i, y_i]$; here z_i is a two-dimensional vector indicating the top left corner of the object/parts. To simplify the notation we assume that the scale level is fixed in an image pyramid.

CNN features are extracted from placements of parts as well as the whole object to yield a feature representation of $[\phi(z_0), ..., \phi(z_i) ..., \phi(z_n)]$ where z_0 and $z_1, ..., z_n$ are whole-object and part locations and $\phi(z_i)$ is the feature representation of part P_i. In our experiments, we extract deep convolutional features $\phi(z_i)$ from an ImageNet pre-trained CNN Overfeat [1] as it is trained with multi-task applications in mind as discussed in previous sections. Crops of each region are warped to 231 × 231 to suit the Overfeat network input size.

3.1 Inference: Detection and Recognition

In this section, we would assume that the latent part model is learned, i.e. $\{w_i^d, w_i^r\}_{i=1}^n$ have been instantiated during the training stage which is discussed later in Chapter 4.

Detection phase Similar to [42], during the detection phase, the scoring function of a placement Z is given by the sum of scores of each inner product which do not incorporate any knowledge of how objects and their parts are constrained geometrically, plus a score of the placement of each part relative to the root (the regularization term), which is often referred to as the deformation cost that penalizes the cases where the parts are placed far away from its statistics of its original positions during the training time:

$$J(Z) = \sum_{i=0}^{n} w_i^d \cdot \phi(z_i) - \sum_{i=0}^{n} d_i \cdot \phi_d(dx_i, dy_i) + b \tag{3.1}$$

where

$$(dx_i, dy_i) = (x_i, y_i) - (2(x_0, y_0) + v_i) \tag{3.2}$$

gives the displacement of the i-th part relative to its anchor position and

$$\phi_d(dx, dy) = (dx, dy, dx^2, dy^2) \tag{3.3}$$

are deformation costs extended by putting into their respective quadratic terms. To compute the detection probability map, for root locationz_0, we compute the overall score for the best placement of corresponding part locations, which is a set of Z, $Z = [z_0, ..., z_n]$ such that

$$Z_0^* = \arg\max_Z J(Z) \tag{3.4}$$

and the final score for this location z_0 is computed as $J(Z_0^*)$.

We then adopt a sliding window approach to compute $J(Z_0^*)$ for each z_0. A detection probability map can then be generated for the entire image. It is obvious that each location z_0 is coupled with an optimal placement of its parts Z_0^*. The locations Z_0^* with the responses of $R_T^* = \{Z^* | S(Z_0^*) > T\}$ (R to denote only root locations in the entire image, $R = \{z_0\}$) and T the detection threshold, following a non-maximum-suppression (NMS)[74], are considered to be the final predicted bounding boxes. Following the implementation strategy of [1], though, we are implementing this in a sliding window manner, rather than computing the entire convolutional neural network for each window of the input one at a time. Because the convolution operations share computations common to overlapping regions, we just apply each convolution over the entire image, so that the computations common to neighboring windows are only done once.

Recognition phase After the objects and their parts have been detected and localized as m bounding boxes with respect to the root locations, $R^*_T = \{r^*_l\}^m_{l=1}$. We learn a one-versus-all linear SVM using the same deep feature representation as we used during the detection phase $[\phi(z_0), ..., \phi(z_i)..., \phi(z_n)]$. Suppose that the n part-based SVM classifiers for each type of subcategory has been learned to have parameters $W^r = \{w^r_i\}^n_{i=0}$ for all category types $s \in \mathbf{S}$, where $\mathbf{S}$ is the set of all objects and parts labels. Recognition of the object at a detected location r_l can be performed with the equation as follows:

$$C_s(r_l) = \sum_{i=0}^{n} w^r_i \cdot \phi(z^*_i), \forall s \in \{\mathbf{S}\} \tag{3.5}$$

where

$$z^*_0 = r_l \tag{3.6}$$

is the root location that is selected during the detection phase. The subcategory type that has the highest score of $C_s(r_l)$ among all $s \in \{\mathbf{S}\}$ is chosen to be the correct type of subcategory. Note that our approach uses the deep features once which are then selected by SVM for different tasks; therefore during the inference stage the feature extraction part which is the most computationally expensive operation is only performed once throughout the data. Thanks to the deep feature in [1], which is learned in a multi-task setting, enough information is contained to be selected by SVM for different tasks.

CHAPTER 4

JOINT TRAINING FOR DETECTION AND RECOGNITION

As our task is to detect a single object category and be able to distinguish between its subcategories, and all subcategories should share the same types of parts, the part model can be hand-crafted once rather than letting the algorithm discover automatically [42] which is more suitable for a database with a large number of class types. Therefore we model the part configurations explicitly as in the strongly supervised case in [62].

In the training samples, each image contains objects that are annotated with their bounding boxes and the image coordinates of the parts $B = [b_0, ..., b_i, ..., b_n]$. Each annotated object and its parts are also labeled with the category types as well as the part types, $S = [s_0, ..., s_k, ..., s_n]$.

For detector training, we define positive examples to be the ones that overlap with the annotated bounding box B by at least 70% which is also the threshold that is used by the current state-of-the-art [45] in ImageNet object detection challenge [25]. Negative examples are sampled from the images or the rest part of images that do not contain the target object to avoid confusion with positive examples.

We use the classical latent SVM to train the detector. Let $D = (< x_1, y_1 > , ..., < x_N, y_N >)$ be a set of labeled examples, where $y_i \in \{-1, 1\}$ indicating negative and positive labels. Given a part image x that we want to classify, and some learned model parameters w^d, we select a label $y_i \in \{-1, 1\}$ as follows:

$$y = sign(f_{w^d}(x)) \tag{4.1}$$

where $f_{w^d}(x)$ is a scoring function defined as:

$$f_{w^d}(x) = \max_{z \in \mathbf{Z}} w^d \phi(x, z) \tag{4.2}$$

where z is a latent variable chosen among the set $\mathbf{Z}$. For object detection, $\mathbf{Z}$

is a set of bounding boxes locations over the object or one of its parts x, and maximizing over $\mathbf{Z}$ amounts to finding a bounding box containing the object or one of its parts. Once we are given the training data D, we can train the system by optimizing the latent SVM formulation,

$$\min_{w^d} \frac{1}{2}\|w^d\|_2^2 + C\sum_{i=1}^{N} L(y_i, \max_{z\in\mathbf{Z}} w^d\phi(x,z)) \tag{4.3}$$

where L is the hinge loss defined as $L(y,\hat{y}) = \max(0, 1 - y\hat{y})$. It has been noted in literature that this objective function is semi-convex [64, 42] in which if the latent position z is fixed then the scoring function f becomes convex in w^d. If the latent values z for the positive examples are not fixed we can compute a local optimum of Eqn. 4.3 using an EM-like coordinate descent algorithm:

1. Set w^d to be fixed and search over the latent positions over the positive examples, which is essentially the inference step.

$$z^* = \arg\max_{z\in\mathbf{Z}} w^d \cdot \phi(x,z) \tag{4.4}$$

2. Set z^* to be fixed for positive examples and optimize the objective function in Eqn. 4.3.

A coordinate descent algorithm will always improve or maintain the value of the objective function and will converge to a local minimum [75]. However in practice, the search space for the first step is huge, so here we adopt a strategy of iterative refinement within a limited search range. We initialize the latent part positions to be the annotated positions, i.e. $Z^0 = [z_0^0, ..., z_i^0, ..., z_n^0] = [b_0, ..., b_i, ..., b_n] = B$ and define the search range to be within one percent of the size of the image, i.e. 3×5 pixels if the image size is 300×500. This threshold should be changed with respect to the accuracy of human annotation of the dataset.

4.1 Concurrent Good-Positive and Hard-Negative Mining

As discussed in the previous section, for the positive examples we treat both the part locations and the annotated location of the object as latent variables Z with initialization Z^0 provided by the annotations. This initialization is refined in each iteration until the consecutive detection results do not differ more than one pixel in position, either horizontal or vertical. In the same iteration, hard negatives are also discovered during the inference phase of the iteration, i.e. Eqn. 4.4, but this time we search in the range of negative examples. These examples are then cumulatively added to the negative training set for the next round of iteration following the procedure in [42].

4.2 SVM Training as Feature Selection

For training the classifiers for subcategory recognition, we employ a one-versus-all linear SVM using the final locations Z^* refined by the good positive mining procedure elaborated in the previous section. We still use the pre-trained Overfeat [1] deep feature representation extracted from the refined locations Z^* for training these SVMs. Since the part type labels $S = [s_0, ..., s_k, ..., s_n]$ are being shifted together during the refinement phase, these original labels are still used for training the one-versus-all linear SVMs for the subcategory recognition task; we choose to minimize the following objective for each object/part s:

$$\min_{w_s^r} \frac{1}{2}\|w_s^r\|_2^2 + C\sum_{i=1}^{N} L(l_i, w_s^r\phi(x)) \tag{4.5}$$

where

$$l_i = 1 \quad if \quad s = s_k, \quad else \quad l_i = -1.$$

Algorithm 1 Joint training of latent variable detection and recognition

```
Require: Positive and negative examples $D$ sampled from initialized positions $Z^0 = [z_0^0, ..., z_i^0, ..., z_n^0]$ and training labels, $S = [s_0, ..., s_k, ..., s_n]$ of all respective latent objects and their parts.
  Extract deep feature $\phi(x)$ in a sliding window manner as in[1].
  for all $< x_i, y_i > \in D$, samples in the detection training set do
    Initialize $Z_0 \leftarrow Z^0$
    while $\|Z_{t-1} - Z_t\| > 1$ do
      Learn detection weights $W_t^d$ by optimizing Eqn.4.3
      with positive locations $Z_t$ and augmented negative set $E$,
      $\mathbf{Z_{t-1} \leftarrow Z_t}$
      Good positive mining by inference $Z_t$
      with updated weights $W_t^d$ using Eqn. 4.4
      Hard negative mining of set $E'$ with updated weights $W_t^d$
      $\mathbf{E \leftarrow E \cup E'}$
    end while
  end for
  Learn recognition weights $W^r$
  by optimizing Eqn.4.5 with training labels $S$ and final detection locations $Z^*$
  return $W^d, W^r$
```

CHAPTER 5

EVALUATION ON THE JOINT ALGORITHM

For evaluation of our proposed framework and algorithm, the fine-grained Caltech-UCSD birds dataset [36] (CUB200-2011) is used as the benchmark. We chose this dataset for evaluation of our algorithm because it best simulates the single category detection with its subcategory recognition scenario while not assuming that the bounding box is given at the testing time. It contains 11,788 images of 200 bird species. Each image has one bird inside and is annotated with its bounding box and 15 part locations per image; these parts are the beak, back, breast, belly, forehead, crown, left eye, left leg, left wing, right eye, right leg, right wing, tail, nape and throat. Along with the dataset there also comes 322 binary attribute labels from Mechanical Turk workers. The only catch is that our method has the ability for multiple object detection in a single image while the birds dataset does not have multiple subcategory objects in one image. We use the suggested train-test splits in the dataset, which includes around 30 training samples for each species. Two types of parts from the dataset are used for our evaluation, the head and body, following the common protocol of [52] which is also followed by [53]. For current subcategory recognition methods such as POOF [54], DPD [53], DeCaf [76], NP-Transfer [49], Align [73], the bounding boxes are given during the testing time. It is interesting to show that our detection-recognition framework can achieve performance which is on par with the state-of-the-art when the bounding box is unknown at the testing time as in Table 5.1. We can observe in Table 5.1 that our framework can achieve similar performance in a more difficult scenario than other methods. Some examples of our joint detection and recognition are shown in Figure 5.1.

Table 5.1: Comparison of states-of-the-art in subcategory recognition

POOF[54] with bounding box	56.9%
DPD[53] with bounding box	51.0%
NP-Transfer[49] with bounding box	57.8%
DeCaf[76] with bounding box	65.0%
Align[73] with bounding box	62.7%
Ours **without** bounding box	64.8%

Table 5.2: Effects of pose and translation normalizations in subcategory recognition.

	With	Without	Normalization Effect
Translation normalization	64.8%	59.5%	+5.3%
Pose normalization	64.8%	51.7%	+13.1%

5.1 Pose/translation normalization effect

As discussed in Chapter 1, here we show evidence that deep learning, though it has huge capacity and describing power, can still be improved by finding a better bias-variance trade-off; in this work we use the strategy of normalization. In these further experiments we first evaluate the effect of pose normalization. Here we still use the deep feature of Overfeat [1], but only preserve the object z_0 and remove the part-based configurations $[z_1, ...z_n]$, and we observe that the performance drops by 13.1%. This shows that a pose normalization procedure helps to get a better bias-variance trade-off even with the most powerful features and classifiers, which also counteracts the arguments that vision research may not be as helpful as deep learning research.

The second experiment we conduct here is to evaluate the effect of translation normalization. Here, only during the training time, we remove the iterative refinements of the bounding boxes of objects and its parts; i.e. the latent variables Z are instantiated by the annotated bounding boxes B without subsequent “good positive mining” iterative procedure. Note that in our experiment, the learned weights for detection W^d are not used for the later recognition task, but we use the human annotated parts for learning the subcategory recognition weights W^r. A performance drop of 5.3% is observed in this experiment scenario. The effects of pose and translation normalization are summarized in Table 5.2.

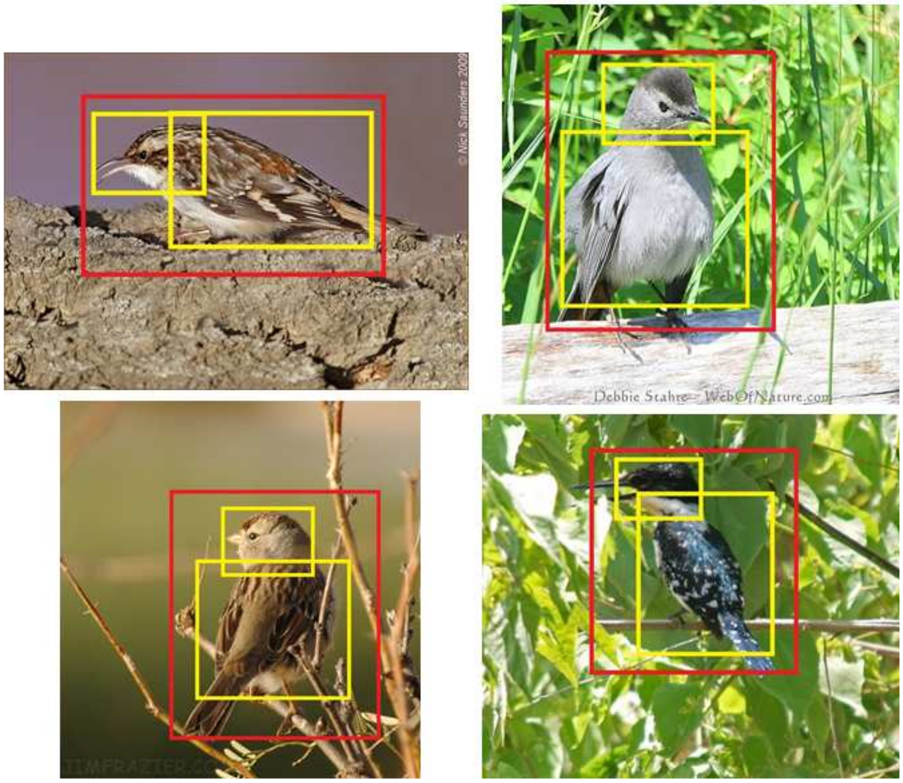


Figure 5.1: Joint detection and subcategory recognition results from four different viewpoints.

5.2 Conclusion and Future Work

In this chapter, we propose a normalization framework to account for variations in the joint detection and recognition problem and devise a unified normalization-based algorithm for joint learning of detection and recognition tasks. This chapter also proposes and implements a latent variable model which yields a positive-negative retraining algorithm for mining good positives and hard negatives for translation normalization and therefore yields higher accuracy than counterpart algorithms that do not perform the iterative fine-tuning process. In future works, we can formulate the problem into regression so as to utilize the circulant structure to incorporate the retraining stage into one-round of training. The deep feature that is learned by Overfeat [1] is for detection and recognition of general object classes, which is not designed specifically for single category detection and its subcategory recognition problem. In future work it is possible to devise a new network to train features for joint detection and subcategory recognition. Other important quantities that can be estimated from this framework such as pose and attributes can be determined simultaneously.

CHAPTER 6

LAYOUT-AWARE EPITOME

Recently, *epitome* has been successfully applied in computer vision as a patch-based generative model of image(s) or video [77, 78]. As a maximum likelihood representation for image data, it can be considered as a tradeoff representation in-between template and histogram. The balance between visual resemblance and generalization of image and video can be adjusted by the sizes of epitome and patch. It has attracted more and more attention in computer vision due to its impressive abilities in many vision tasks.

The "epitomes" were first introduced as simple appearance and shape models in [78]. These models are learned by compiling patches drawn from input images into a condensed image model. It was shown in [32] that the image epitome is an image summary of high "completeness." The epitome idea has also found its use in representing audio information [33] and human activities [34]. Jigsaw proposed in [79] took the epitome beyond square patches and modeled local spatial coherence. The epitome model was also extended to location recognition [80], where it uses each of the entire input image as a patch in which the mappings are fixed during learning and inference. The image frames from a panoramic video are automatically stitched together to form a panorama due to the epitome's ability in exploring image similarities [32]. Most recently, epitome priors were investigated for image parsing in which non-overlapping patches are associated with labels of object classes [81].

Under the generative model framework, the learned epitome is a condensation of image patches, which are however not able to regenerate a meaningful image without guidance by an input image to give a meaningful spatial layout. The input image serves as a location map during the learning and inference process. Since the expected mapping posteriors are only estimated from patch-similarity measurements in inference, it will often cause ambiguities in reconstruction and recognition during the inference process due to the lack of spatial constraints. For example, epitome was used to recover the

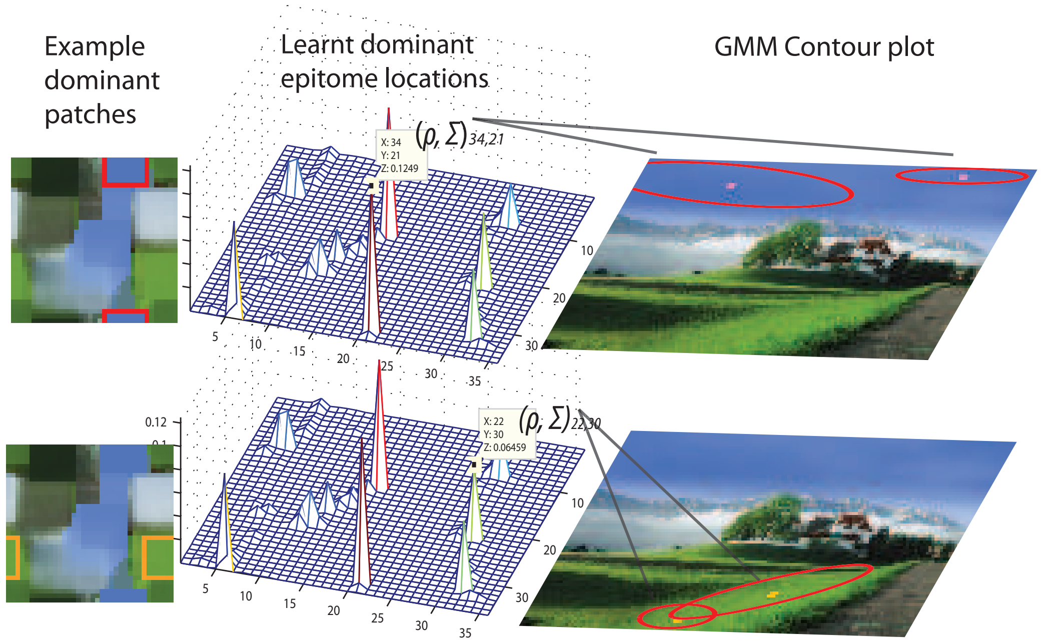


Figure 6.1: A 36 × 36 spatialized epitome (in the first column) is learned from the image in the third column. The distribution in the middle column shows the positions of the significant patches. Note that most locations are of zero value due to regularization. The leftmost image in each row highlights a significant patch in the spatialized epitome. Its associated Gaussian mixture which represents the spatial arrangement of the significant patch in the input image is shown as ellipse contours in the third column.

occluded part of the object in a video by replacing the occlusion with the patches learned from the nearby images without occlusions. However, the conventional epitome model can only assign a patch in the model to a patch in the image according to the patch-wise similarity of intensity. When the occluded area contains patches that are of different appearance from nearby patches in the image, the model would generally fail to assign the correct patch to replace the occlusion. Therefore, the epitome might not be applicable for recognition/detection tasks because of this ambiguity caused by the lack of information about where the patches come from and how similar-patches are distributed on the input images. In [82], a few pairs of long-range patches are randomly selected for each patch for spatial constraints in image reconstruction. Such pairs represent a few specific spatial correlations. They cannot model the general spatial distributions of similar patches, and, in worse cases, may capture false correlation between two long-range patches, e.g. the foreground patch with background patch. As for rebuilding from compressed image, Wang et al. [83] proposed to record the fixed mapping to copy the patches from the epitome to the image locations. The flexibility and optimality of image summarization and inference by generative models are lost in such a hard-coding approach.

Motivated by the aforementioned observations, we propose a new graphical model of epitome to integrate information about the appearance summary and spatial arrangement of patches in the image(s). A set of Gaussian mixtures is introduced into the original graphical model of epitome to relate the appearance and shape with their spatial arrangements on the input images; see Figure 6.1 for illustration. In this way, the model is self-contained with appearance, shape, as well as patch spatial distribution in input images. So by sampling the learned model itself, the spatialized epitome is capable of synthesizing the scenes and objects it "saw" during training (see Section 9.1). With spatial constraints included in the epitome model, the misalignment problem with various variations can be solved automatically because the proposed model allows the patches to organize adaptively during inference. To evaluate on a few tough vision tasks, we investigate by applying the proposed spatialized epitome for misaligned face recognition and cross-pose face recognition, which means to recognize people with poses unseen in the training set. The main contributions of this thesis can be summarized as follows:

1. A new graphical model of epitome which combines the information about patch appearance and its associated spatial distributions.

2. An EM procedure to learn the optimized appearance summary and cluster the spatial distributions of image patches.

3. A likelihood probability by image inference from the spatialized epitome.

4. Investigation on applying the spatialized epitome for a few tough vision tasks including colorization.

Later chapters regarding this work are structured as follows: In Chapter 7, we present the spatialized epitome model and the derivation of the learning procedure. We derive the inference process in Chapter 8. Experiments, including the comparisons with the original epitome, on face recognition with misalignments, cross-pose face recognition, and car detection with and without occlusions, are presented in Chapter 9. The application of epitome for colorization is presented in Chapter 10.

CHAPTER 7

LEARNING A SPATIALIZED EPITOME

An image does not merely consist of patches, and it also depends on how the patches are spatially arranged. In existing epitome [78, 82], for each patch $\mathbf{Z}_k$, the likelihood probability was calculated by an intensity similarity. Therefore, the process of inference and reconstruction on an input image is purely guided by intensity-similarity measure with respect to the training images regardless of how patches are arranged in the training or probe image. We show the problem of this under-constrained process in Chapter 8.

Here we present a generative model combining both patch appearances and arrangements in an image or a collection of images. Suppose P patches are sampled from M images and denote each patch as $\mathbf{Z}_k$. The corresponding mapping random variable is denoted as $\mathcal{T}_k$, which is hidden and unknown. The patch is sampled from the position $\mathbf{y}_k$ in the original image, so $\mathbf{y}_k$ is observed. For each patch in the epitome, we use Gaussian mixture models (GMM) to model the image locations from which the patches are originated. If the size of the epitome is a, then we have $a \times R$ such GMMs. C_k is an R-dimensional binary random variable in which a particular element C_{kr} is equal to 1 and all other elements are equal to 0 when the component r is active. For each observed location $\mathbf{y}_k$, there is a corresponding latent variable C_k. We now define the generative process:

1. Choose a position in the epitome, $\mathcal{T}_k \sim Cat(\boldsymbol{\pi})$.
2. For each of the chosen position $\mathcal{T}_k$,
   (a) Choose a patch $\mathbf{Z}_k$ from $p(\mathbf{Z}_k|\mathcal{T}_k, \mathbf{e})$.
   (b) Choose a component C_k from the GMMs for the given location $\mathcal{T}_k$: $C_k \sim p(C_k|\mathcal{T}_k)$.
   (c) Choose a coordinate $\mathbf{y}_k$ from the component C_k for patch $\mathbf{Z}_k$: $\mathbf{y}_k \sim p(\mathbf{y}_k|\mathcal{T}_k, C_k)$.

This process is illustrated in Figure 7.1. The generation of each patch (intensity) is formulated as:

$$P(\mathbf{Z}_k|\mathcal{T}_k,\mathbf{e}) = \prod_{i\in S_k} \mathcal{N}(z_{i,k};\mu_{\mathcal{T}_k(i)},\phi_{\mathcal{T}_k(i)}), \tag{7.1}$$

where S_k is the set of the coordinates of all pixels in the patch $\mathbf{Z}_k$. The generation of the coordinate of each patch is formulated as:

$$P(\mathbf{y}_k|\mathcal{T}_k, C_{kr}=1) = \mathcal{N}(\mathbf{y}_k;\boldsymbol{\rho}^r_{\mathcal{T}_k=e},\boldsymbol{\Sigma}^r_{\mathcal{T}_k=e}), \tag{7.2}$$

where e represents the location in the epitome that the patch maps to, and the superscript r indicates the rth component of the GMM. Write it in a compact distribution form:

$$p(\mathbf{y}_k|\mathcal{T}_k, C_k) = \prod_{r=1}^{R} \mathcal{N}(\mathbf{y}_k;\boldsymbol{\rho}^r_{\mathcal{T}_k=e},\boldsymbol{\Sigma}^r_{\mathcal{T}_k=e})^{C_{kr}}. \tag{7.3}$$

Given the mapping $\mathcal{T}_k$ of the patch $\mathbf{Z}_k$, there are several Gaussian components in the location $\mathcal{T}_k = e$ to choose from, where e denotes a particular location in the epitome. The probability distribution of choosing each Gaussian component given the location e is

$$p(C_k|\mathcal{T}_k) = \prod_{r=1}^{R} \tilde{\pi}_{\mathcal{T}_k=e,r}^{C_{\mathcal{T}_k=e,r}}. \tag{7.4}$$

Since $p(C_k,\mathcal{T}_k) = p(C_k|\mathcal{T}_k)p(\mathcal{T}_k)$ and the prior on both parameters shall be learned, we use the joint distribution of C_k and $\mathcal{T}_k$ to perform parameter estimation on the mixing coefficients.

7.1 Learning procedure for spatialized epitome

For the P patches generated independently, we have the joint distribution:

$$p(\{\mathbf{Z}_k,\mathcal{T}_k,C_k,\mathbf{y}_k\}_{k=1}^{P},\mathbf{e},\boldsymbol{\pi}) = \\ p(\mathbf{e},\boldsymbol{\pi})\prod_{k=1}^{P} p(\mathbf{Z}_k|\mathcal{T}_k,\mathbf{e})p(\mathbf{y}_k|\mathcal{T}_k,C_k)p(C_k,\mathcal{T}_k), \tag{7.5}$$

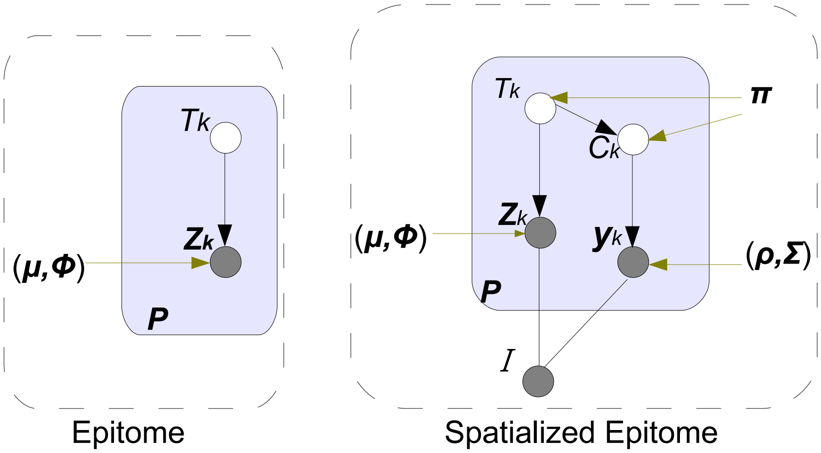


Figure 7.1: The graphical model representations of the epitome and the spatialized epitome. The boxes are "plates" representing replicates.

where $\boldsymbol{\pi}$ are the parameters of the mixing proportions on $\mathcal{T}_k$ and C_k. Since we cannot observe C_k and $\mathcal{T}_k$, we sum over all possible values that they might be taking, and

$$\begin{aligned}\log P(\{\mathbf{Z}_k, \mathbf{y}_k\}_{k=1}^P) &= \log \sum_{\{C_k, \mathcal{T}_k\}} \int_{\mathbf{e}, \boldsymbol{\pi}} p(\{\mathbf{Z}_k, \mathcal{T}_k, C_k, \mathbf{y}_k\}_{k=1}^P, \mathbf{e}, \boldsymbol{\pi}) d(\mathbf{e}, \boldsymbol{\pi}) \\ &= \log \sum_{\{C_k, \mathcal{T}_k\}} \prod_{k=1}^{P} p(\mathbf{Z}_k | \mathcal{T}_k, \mathbf{e}) p(\mathbf{y}_k | \mathcal{T}_k, C_k) p(C_k, \mathcal{T}_k). \quad (7.6)\end{aligned}$$

Now we first assume that the prior on the parameters are flat. We use variational approximation to put the log inside the $\sum$ for tractable optimization, the auxiliary distribution $q(\{\mathcal{T}_k, C_k\}_{k=1}^P)$ is put into the likelihood of data, and then we use the Jensen's inequality [84]:

$$\begin{aligned}\log P(\{\mathbf{Z}_k, \mathbf{y}_k\}_{k=1}^P) &= \log \sum_{\{C_k, \mathcal{T}_k\}} \frac{q(\{\mathcal{T}_k, C_k\}_{k=1}^P) p(\{\mathbf{Z}_k, \mathcal{T}_k, C_k, \mathbf{y}_k\}_{k=1}^P)}{q(\{\mathcal{T}_k, C_k\}_{k=1}^P)} \\ \geq \sum_{\{C_k, \mathcal{T}_k\}} q(\{\mathcal{T}_k, C_k\}_{k=1}^P) \log \frac{p(\{\mathbf{Z}_k, \mathcal{T}_k, C_k, \mathbf{y}_k\}_{k=1}^P)}{q(\{\mathcal{T}_k, C_k\}_{k=1}^P)} \\ = \sum_{\{C_k, \mathcal{T}_k\}} q(\{\mathcal{T}_k, C_k\}_{k=1}^P) \log p(\{\mathbf{Z}_k, \mathcal{T}_k, C_k, \mathbf{y}_k\}_{k=1}^P) \\ - \sum_{\{C_k, \mathcal{T}_k\}} q(\{\mathcal{T}_k, C_k\}_{k=1}^P) \log q(\{\mathcal{T}_k, C_k\}_{k=1}^P) = B. \quad (7.7)\end{aligned}$$

Since $q(\{\mathcal{T}_k, C_k\}_{k=1}^{P}) = \prod_{k=1}^{P} q(\mathcal{T}_k, C_k)$ due to the independence assumption by variational mean field theory [84], we have

$$\begin{aligned}
&\log P(\{\mathbf{Z}_k, \mathbf{y}_k\}_{k=1}^{P}) \geq B = \\
&\qquad \sum_{\{C_k,\mathcal{T}_k\}} \prod_{k=1}^{P} q(\mathcal{T}_k, C_k) \log \prod_{k=1}^{P} p(\mathbf{Z}_k|\mathcal{T}_k, \mathbf{e}) p(\mathbf{y}_k|\mathcal{T}_k, C_k) p(C_k, \mathcal{T}_k) \\
&\qquad\qquad - \sum_{\{C_k,\mathcal{T}_k\}} q(\{\mathcal{T}_k, C_k\}_{k=1}^{P}) \log q(\{\mathcal{T}_k, C_k\}_{k=1}^{P}) \\
&= \sum_{k=1}^{P} \sum_{C_k,\mathcal{T}_k} q(\mathcal{T}_k, C_k)[\log p(\mathcal{T}_k, C_k) + \\
&\qquad\qquad \log p(\mathbf{y}_k|\mathcal{T}_k, C_k) + \log p(\mathbf{Z}_k|\mathcal{T}_k, \hat{\mathbf{e}})] - E. \quad (7.8)
\end{aligned}$$

When $q(\mathcal{T}_k, C_k) = p(\mathcal{T}_k, C_k|\mathbf{Z}_k, \mathbf{y}_k, \hat{\mathbf{e}})$, the lower bound is tight and the entropy $E = 0$, which can be proved by substituting the posterior into the bound. Note that here we can update $p(C_k, \mathcal{T}_k)$, $p(\mathbf{y}_k|\mathcal{T}_k, C_k)$ and $p(\mathbf{Z}_k|\mathcal{T}_k, \hat{\mathbf{e}})$ independently. By iteratively optimizing the bound B, we can derive an EM procedure to learn the spatialized epitome.

The E-Step: By setting the auxiliary distribution to be the posterior of hidden variables, there is

$$\begin{aligned}
&q(\mathcal{T}_k, C_k) = p(\mathcal{T}_k, C_k|\mathbf{Z}_k, \mathbf{y}_k, \hat{\mathbf{e}}) = \frac{p(\mathbf{Z}_k, \mathcal{T}_k, C_k, \mathbf{y}_k, \hat{\mathbf{e}})}{p(\mathbf{Z}_k, \mathbf{y}_k, \hat{\mathbf{e}})} \\
&= \frac{p(\mathbf{Z}_k|\mathcal{T}_k, \hat{\mathbf{e}}) p(\mathbf{y}_k|\mathcal{T}_k, C_k) p(C_k, \mathcal{T}_k)}{p(\mathbf{Z}_k, \mathbf{y}_k, \hat{\mathbf{e}})} \\
&\sim p(\mathbf{Z}_k|\mathcal{T}_k, \hat{\mathbf{e}}) p(\mathbf{y}_k|\mathcal{T}_k, C_k) p(C_k, \mathcal{T}_k) \\
&\qquad = \prod_{i \in S_k} \mathcal{N}(z_{i,k}; \mu_{\mathcal{T}_k(i)}, \phi_{\mathcal{T}_k(i)}) \prod_{r=1}^{R} \mathcal{N}(\mathbf{y}_k; \rho^r_{\mathcal{T}_k=e}, \mathbf{\Sigma}^r_{\mathcal{T}_k=e})^{C_{kr}} p(C_k, \mathcal{T}_k). \quad (7.9)
\end{aligned}$$

The M-Step: Note the equal sign indicates that the bound is tight at this moment; the bound B can be separated into three parts: $B = B_1 + B_2 + B_3$, where B_1 is related to the epitome appearance, B_2 is related to spatial distributions, and B_3 is related to mixing weights. Hence, we can derive the update rules for the three sets of parameters separately.

a) Updating the appearance

Only the term B_1 in B relates to the epitome appearance $\hat{\mathbf{e}}$. Let us denote the estimated distribution $q(\mathcal{T}_k, C_k)$ as q_k for simplicity. B_1 can be expressed as

$$\begin{aligned}B_1 = \sum_{k=1}^{P} \sum_{C_k, \mathcal{T}_k(i)=j} q_k \log p(\mathbf{Z}_k | \mathcal{T}_k, \hat{\mathbf{e}}) = \\ = \sum_{k=1}^{P} \sum_{C_k, \mathcal{T}_k(i)=j} \sum_{i \in S_k} q_k \left[-\frac{1}{2} \log 2\pi\phi_j - \frac{(z_{i,k} - \mu_j)^2}{2\phi_j} \right]. \quad (7.10)\end{aligned}$$

Finding the solution for $\partial B_1 / \partial \hat{\mathbf{e}} = 0$ is equivalent to finding the solutions for $\frac{\partial B_1}{\partial \mu_j} = 0$ and $\frac{\partial B_1}{\partial \phi_j} = 0$, respectively. Hence, the updating rule for μ_j can be obtained as:

$$\mu_j = \frac{\sum_{k=1}^{P} \sum_{C_k, \mathcal{T}_k(i)=j} \sum_{i \in S_k} q(\mathcal{T}_k, C_k) z_{i,k}}{\sum_{k=1}^{P} \sum_{C_k, \mathcal{T}_k(i)=j} \sum_{i \in S_k} q(\mathcal{T}_k, C_k)}, \quad (7.11)$$

and the corresponding updating rule for ϕ_j is:

$$\phi_j = \frac{\sum_{k=1}^{P} \sum_{C_k, \mathcal{T}_k(i)=j} \sum_{i \in S_k} q(\mathcal{T}_k, C_k)(z_{i,k} - \mu_j)^2}{\sum_{k=1}^{P} \sum_{C_k, \mathcal{T}_k(i)=j} \sum_{i \in S_k} q(\mathcal{T}_k, C_k)}. \quad (7.12)$$

This is similar to the original epitome updating rules.

b) Update GMM Means and Covariances

From Eq. (7.8), the bound for the GMM term is simplified as:

$$\begin{aligned}B_2 = \sum_{k=1}^{P} \sum_{C_k, \mathcal{T}_k} q(\mathcal{T}_k, C_k) \log p(\mathbf{y}_k | \mathcal{T}_k, C_k) = \\ = \sum_{k=1}^{P} \sum_{C_k, \mathcal{T}_k} q(\mathcal{T}_k, C_k) \sum_{r=1}^{R} C_{kr} \log \mathcal{N}(\mathbf{y}_k; \boldsymbol{\rho}^r_{\mathcal{T}_k=e}, \boldsymbol{\Sigma}^r_{\mathcal{T}_k=e}). \quad (7.13)\end{aligned}$$

Set the derivative w.r.t. $\boldsymbol{\rho}^r_{\mathcal{T}_k=e}$ to be 0, i.e. $\frac{\partial B_2}{\partial \boldsymbol{\rho}^r_e} = 0$, then there is

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\rho}^r_e} \sum_{k=1}^{P} \sum_{C_k, \mathcal{T}_k} q(\mathcal{T}_k, C_k) \sum_{r=1}^{R} C_{kr} \log \mathcal{N}(\mathbf{y}_k; \boldsymbol{\rho}^r_{\mathcal{T}_k=e}, \boldsymbol{\Sigma}^r_{\mathcal{T}_k=e}) \\ = \frac{\partial}{\partial \boldsymbol{\rho}^r_e} \sum_{k=1}^{P} \sum_{C_k, \mathcal{T}_k} q(\mathcal{T}_k, C_k) C_{kr} \log \mathcal{N}(\mathbf{y}_k; \boldsymbol{\rho}^r_{\mathcal{T}_k=e}, \boldsymbol{\Sigma}^r_{\mathcal{T}_k=e}) \\ = \sum_{k=1}^{P} \sum_{C_k, \mathcal{T}_k} q(\mathcal{T}_k, C_k) C_{kr} (\mathbf{y}_k - \boldsymbol{\rho}^r_e)^T (\boldsymbol{\Sigma}^r_e)^{-1} = 0. \quad (7.14)\end{aligned}$$

From the equation 7.14, we can obtain the updating rule for $\boldsymbol{\rho}_e^r$ as:

$$(\boldsymbol{\rho}_e^r)^T = \frac{\sum_{k=1}^P \sum_{C_k, \mathcal{T}_k = e} q(\mathcal{T}_k, C_k) C_{kr} \mathbf{y}_k^T}{\sum_{k=1}^P \sum_{C_k, \mathcal{T}_k = e} q(\mathcal{T}_k, C_k) C_{kr}}. \tag{7.15}$$

Applying the same deduction for the GMM mean, we take derivative w.r.t $(\boldsymbol{\Sigma}_e^r)^{-1}$ and set it to be 0:

$$\begin{aligned}
&\frac{\partial}{\partial (\boldsymbol{\Sigma}_e^r)^{-1}} \sum_{k=1}^P \sum_{C_k, \mathcal{T}_k} q(\mathcal{T}_k, C_k) C_{kr} \log \mathcal{N}(\mathbf{y}_k; \boldsymbol{\rho}_{\mathcal{T}_k = e}^r, \boldsymbol{\Sigma}_{\mathcal{T}_k = e}^r) \\
&\qquad = \frac{\partial}{\partial (\boldsymbol{\Sigma}_e^r)^{-1}} \sum_{k=1}^P \sum_{C_k, \mathcal{T}_k} q(\mathcal{T}_k, C_k) C_{kr} [-\log 2\pi - \frac{1}{2} \log |\boldsymbol{\Sigma}_e^r| - \\
&\qquad\qquad \frac{1}{2} (\mathbf{y}_k - \boldsymbol{\rho}_e^r)^T (\boldsymbol{\Sigma}_e^r)^{-1} (\mathbf{y}_k - \boldsymbol{\rho}_e^r)] \\
&\quad = \sum_{k=1}^P \sum_{C_k, \mathcal{T}_k} q(\mathcal{T}_k, C_k) C_{kr} [+\frac{1}{2} \boldsymbol{\Sigma}_e^r - \frac{1}{2} (\mathbf{y}_k - \boldsymbol{\rho}_e^r)^T (\mathbf{y}_k - \boldsymbol{\rho}_e^r)] = 0.
\end{aligned} \tag{7.16}$$

Therefore we obtain the updating rule for $\boldsymbol{\Sigma}_e^r$ as

$$\boldsymbol{\Sigma}_e^r = \frac{\sum_{k=1}^P \sum_{C_k, \mathcal{T}_k = e} q(\mathcal{T}_k, C_k) C_{kr} (\mathbf{y}_k - \rho_e^r)(\mathbf{y}_k - \boldsymbol{\rho}_e^r)^T}{\sum_{k=1}^P \sum_{C_k, \mathcal{T}_k = e} q(\mathcal{T}_k, C_k) C_{kr}}. \tag{7.17}$$

c) Update mixing coefficients

From Eq. (7.8), the term related to mixing coefficients can be expressed:

$$B_3 = \sum_{k=1}^P \sum_{C_k, \mathcal{T}_k} q(\mathcal{T}_k, C_k) \log p(\mathcal{T}_k, C_k). \tag{7.18}$$

Denoting $p(\mathcal{T}_k = e, C_k = r) = \pi_{er}$, we can maximize the bound B_3 subject to $\sum_{e,r} p(\mathcal{T}_k = e, C_k = r) = 1$ as:

$$\begin{aligned}
&\frac{\partial}{\partial \pi_{er}} (B_3 + \lambda (\sum_{e,r} \pi_{er} - 1)) \\
&= \frac{\partial}{\partial \pi_{er}} \sum_{k=1}^P \sum_{C_k = r, \mathcal{T}_k = e} q(\mathcal{T}_k, C_k) \log p(\mathcal{T}_k = e, C_k = r) + \lambda \\
&= \sum_{k=1}^P q(\mathcal{T}_k = e, C_k = r) \frac{1}{\pi_{er}} + \lambda = 0.
\end{aligned} \tag{7.19}$$

Table 7.1: The number of parameters for spatialized epitome model.

Epitome($\hat{\mathbf{e}}$)	Gaussians($\boldsymbol{\rho}, \boldsymbol{\Sigma}$)	Mixing Coefficients ($\boldsymbol{\pi}$)
$N \times N \times 2$	$N \times N \times 2$	$N \times N \times R$

Then, we can obtain $\lambda = -P$ and the updating rule of the mixing coefficient as

$$\pi_{er} = \frac{\sum_{k=1}^{P} q(\mathcal{T}_k = e, C_k = r)}{P}. \tag{7.20}$$

7.2 Bayesian Regularization and Priors

Suppose we have R Gaussian components at one epitome location e. The number of parameters for our epitome with a size of $N \times N$ is $N^2 \times (R+4)$. The details are listed in Table 7.1. Since we have a finite training set and a relatively large set of parameters, in order to avoid overfitting, on each location in the epitome we put a Dirichlet-Normal-Wishart prior on the three sets of parameters $\{\boldsymbol{\rho}_e^r, \boldsymbol{\Sigma}_e^r\}_{r=1}^R$ and $\boldsymbol{\pi}_e$, i.e.

$$\begin{aligned} p(\{\boldsymbol{\rho}_e^r, \boldsymbol{\Sigma}_e^r\}_{r=1}^R, \boldsymbol{\pi}_e) = b(\boldsymbol{\gamma}_e) \prod_{r=1}^{R} (\pi_e^r)^{\gamma_e^r - 1} \\ \prod_{r=1}^{R} \mathcal{N}\left(\boldsymbol{\rho}_e^r | \boldsymbol{\nu}_e^r, \frac{\boldsymbol{\Sigma}_e^r}{\eta_e^r}\right) Wi((\boldsymbol{\Sigma}_e^r)^{-1} | \boldsymbol{\beta}_e^r, \tau_e^r), \end{aligned} \tag{7.21}$$

where $b(\boldsymbol{\gamma}_e)$ is the normalizing factor of the Dirichlet distribution and $Wi(.|)$ denotes a Wishart distribution. By determining appropriate values for the hyper-parameters $\{\gamma_e^r, \boldsymbol{\nu}_e^r, \boldsymbol{\Sigma}_e^r, \eta_e^r, \boldsymbol{\beta}_e^r, \tau_e^r\}$ we state our beliefs about the data generation process in terms of a prior distribution. The use of such prior is justified in [85]. By incorporating the prior, the updating rules are derived to be:

$$(\boldsymbol{\rho}_e^r)^T = \frac{\sum_{k=1}^{P} \sum_{C_k, \mathcal{T}_k = e} q(\mathcal{T}_k, C_k) C_{kr} \mathbf{y}_k^T + \eta_e^r \boldsymbol{\nu}_e^r}{\sum_{k=1}^{P} \sum_{C_k, \mathcal{T}_k = e} q(\mathcal{T}_k, C_k) C_{kr} + \eta_e^r}, \tag{7.22}$$

$$\boldsymbol{\Sigma}_e^r = \frac{\sum_{k=1}^{P}\sum_{C_k,\mathcal{T}_k=e} q(\mathcal{T}_k, C_k)C_{kr}(\mathbf{y}_k - \rho_e^r)(\mathbf{y}_k - \rho_e^r)^T}{\sum_{k=1}^{P}\sum_{C_k,\mathcal{T}_k=e} q(\mathcal{T}_k, C_k)C_{kr} + 2\tau_e^r - 2} + \frac{\eta_e^r(\boldsymbol{\mu}_e^r - \boldsymbol{\nu}_e^r)(\boldsymbol{\mu}_e^r - \boldsymbol{\nu}_e^r)^T + 2\boldsymbol{\beta}_e^r}{\sum_{k=1}^{P}\sum_{C_k,\mathcal{T}_k=e} q(\mathcal{T}_k, C_k)C_{kr} + 2\tau_e^r - 2}, \quad (7.23)$$

$$\pi_{er} = \frac{\sum_{k=1}^{P} q(\mathcal{T}_k = e, C_k = r) + \gamma_e^r - 1}{P + \sum_{r=1}^{R}\gamma_e^r - R}. \quad (7.24)$$

The prior penalizes singularities in the log-likelihood function in the case when an epitome patch has only one corresponding patch in the image(s). We also encode our prior belief that the covariance matrices of GMMs are diagonal with diagonal values to be the width of the training image. We adjust the strength of the prior by modifying γ, β and τ which are functions of the equivalent sample size in Bayesian terms. A sparsity inducing prior (Dirichlet) with $\alpha = 0.05$ is used so that most of the mixing coefficients tend to zero and the corresponding Gaussian components will not contribute in modeling the distributions, as shown in Figure 6.1.

CHAPTER 8

INFERENCE BASED ON SPATIALIZED EPITOME

8.1 Inference

We denote the set of learned parameters $\{\hat{\boldsymbol{\rho}}, \hat{\boldsymbol{\Sigma}}, \hat{\mathbf{e}}, \hat{\boldsymbol{\pi}}\}$ of training set $\mathcal{D}$ as $\hat{\boldsymbol{\Theta}}$. Given the data of a training set $\mathcal{D}$, the probability of seeing a given probe image can be directly calculated as:

$$\begin{aligned}
&\log P(I|\mathcal{D}) \simeq \log P(I|\hat{\boldsymbol{\Theta}}) = \log P(I|\boldsymbol{\rho}, \boldsymbol{\Sigma}, \hat{\mathbf{e}}, \boldsymbol{\pi}) \\
&= \log P(\{\mathbf{Z}_k, \mathbf{y}_k\}_{k=1}^{P}|\boldsymbol{\rho}, \boldsymbol{\Sigma}, \hat{\mathbf{e}}, \boldsymbol{\pi}) \\
&= \log \prod_{k=1}^{P} P(\mathbf{Z}_k, \mathbf{y}_k|\boldsymbol{\rho}, \boldsymbol{\Sigma}, \hat{\mathbf{e}}, \boldsymbol{\pi}) \\
&= \sum_{k=1}^{P} \log \sum_{C_k, \mathcal{T}_k} P(\mathbf{Z}_k, \mathbf{y}_k, C_k, \mathcal{T}_k|\boldsymbol{\rho}, \boldsymbol{\Sigma}, \hat{\mathbf{e}}, \boldsymbol{\pi}) \\
&= \sum_{k=1}^{P} \log \sum_{C_k, \mathcal{T}_k} p(\mathbf{Z}_k|\mathcal{T}_k, \mathbf{e}) p(\mathbf{y}_k|\mathcal{T}_k, C_k) P(C_k, \mathcal{T}_k) \\
&= \sum_{k=1}^{P} \log \sum_{C_k, \mathcal{T}_k} \prod_{i \in S_k} \mathcal{N}(z_{i,k}; \mu_{\mathcal{T}_k(i)}, \phi_{\mathcal{T}_k(i)}) \\
&\qquad\qquad \prod_{r=1}^{R} \mathcal{N}(\mathbf{y}_k; \boldsymbol{\rho}^r_{\mathcal{T}_k=e}, \boldsymbol{\Sigma}^r_{\mathcal{T}_k=e})^{C_{kr}} P(\mathcal{T}_k, C_k).
\end{aligned} \tag{8.1}$$

This inference formulation is similar to the way of evaluating the probability value of seeing a new data under a learned GMM. The first step of this derivation follows [86]. The third step uses the assumption that all the patches are independently sampled. The calculated probability value in Eq. 8.1 indicates how likely the probe image is generated by the learned model, and can be directly used for image recognition and object detection purposes.

8.2 Recognition and Detection

Suppose there are N epitomes with parameters $\{\boldsymbol{\Theta}_i\}_{i=1}^N$ learned from N classes of visual objects. Denote the label of the input image to be $\mathcal{C}$ and we assume no prior knowledge on label $\mathcal{C}$, so the recognition is achieved by computing the label posterior $p(\mathcal{C}|I)$ using:

$$p(\mathcal{C}|I) = \frac{p(I|\mathcal{C})p(\mathcal{C})}{p(I)} \sim p(I|\mathcal{C}), \tag{8.2}$$

and select the one with the maximum posterior value:

$$\hat{\mathcal{C}} = \arg\max_i P(I|\mathcal{C} = i) = \arg\max_i P(I|\boldsymbol{\Theta}_i), \tag{8.3}$$

where $P(I|\boldsymbol{\Theta}_i)$ can be calculated from Eq. (8.2) which is in turn calculated by Eq. (8.1).

Detection If we scan the input image with multi-scale windows (W), we can perform object detection. In this way, Eq.(8.2) becomes

$$p(\mathcal{C}|W) = \frac{p(W|\mathcal{C})p(\mathcal{C})}{p(W)} \sim p(W|\mathcal{C}). \tag{8.4}$$

The mean-shift approach can be used to select local maxima to locate the target objects in the image.

8.3 Epitomic Reestimation

Using existing epitome for image reestimation, for each patch $\mathbf{Z}_k$, the inference step evaluates how likely each epitome patch is to generate $\mathbf{Z}_k$. Then the estimation step will replace the initialized values of $\mathbf{Z}_k$ with the average votes from the epitome patches according to $q(\mathcal{T}_k)$. Consequently, the estimated texture will be more consistent with the epitome texture. This is how denoising, video super-resolution and other video repairing applications are achieved. However, the position posterior $q(\mathcal{T}_k)$ is evaluated purely based on the intensity similarity between the epitome patches and the image patches [78, 82]. This may give an incorrect estimation when the occluded part has different appearances from nearby patches.

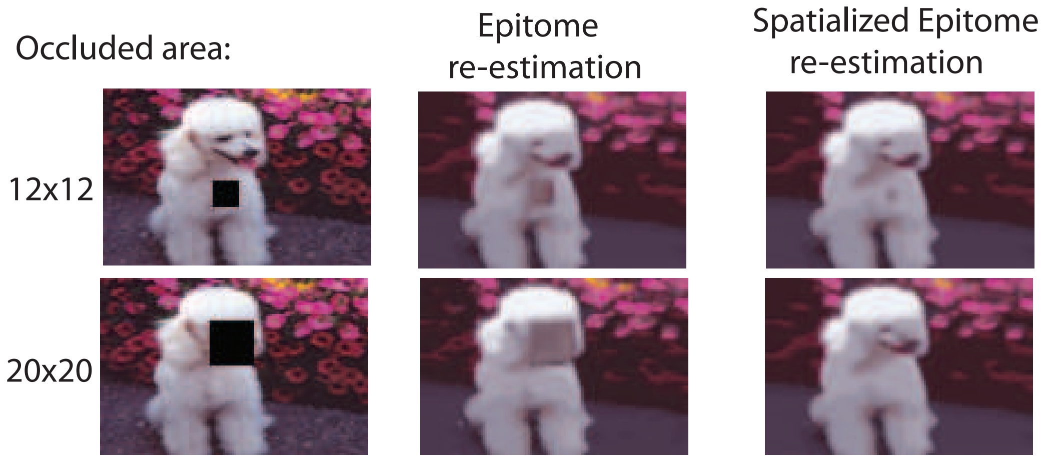


Figure 8.1: The comparison of image reestimation results between epitome and spatialized epitome. Both 40×40 epitomes are learned with patch sizes of 8×8 and 4×4 which is also the patch size used in the reestimation process. During the reestimation process, $40,000$ patches are uniformly sampled from the input image to ensure that all the coordinates are covered for the reestimated image. Since the original epitome just uses a color/intensity similarity to estimate the position posterior, the patches probabilistically chosen from the epitome generate artifacts in the occluded region. In contrast, the spatialized epitome estimates the position posterior based on both intensity similarity and location information; thus, many fewer artifacts are generated due to the spatial constraint. For non-uniform image regions with occlusion, e.g. the second row, spatialized epitome can also restore the occluded region with proper patches.

The reestimation process of spatialized epitome will automatically solve this problem as the position posterior $q(\mathcal{T}_k, C_k)$ takes also the spatial arrangement into account as in Eq. (7.9) in image reestimation. The comparison of existing epitome and spatialized epitome on image reestimation from partially occluded image is given in Figure 8.1.

CHAPTER 9

EXPERIMENTS ON SPATIALIZED EPITOME

In the proposed spatialized epitome, the correlation between the local appearance and spatial arrangement is introduced. This makes it possible to employ epitome for image recognition, object detection, and image reestimation from partial occlusions. To evaluate the performance of the spatialized epitome, several experiments were conducted, including the comparison with existing epitome on face recognition, and applications to several tough vision tasks, e.g., face recognition with misalignments, cross-pose face recognition, occlusion detection, and car detection with a few training samples. The details are described in the following sections. We will provide functional codes such as spatialized epitome learning, inference and synthesis to reproduce the results in this thesis. The codes for the current state-of-the-art results on misalignment face recognition are also provided to facilitate future works.

9.1 Synthesis

Being a self-contained generative model, with both patch intensity and associated spatial distribution, images can be synthesized by ancestral sampling of the proposed model. We show the synthesis results for a scene epitome model (where scene images often consist of large number of redundant patches) as well as for a face epitome model learned from multiple images of the same person in Figure 9.1.

9.2 Generative Face Recognition

In this experiment, we evaluate the effectiveness of our spatialized epitome formulation by face recognition. This generative method does not need to go through any feature extraction or dimensionality reduction step but just

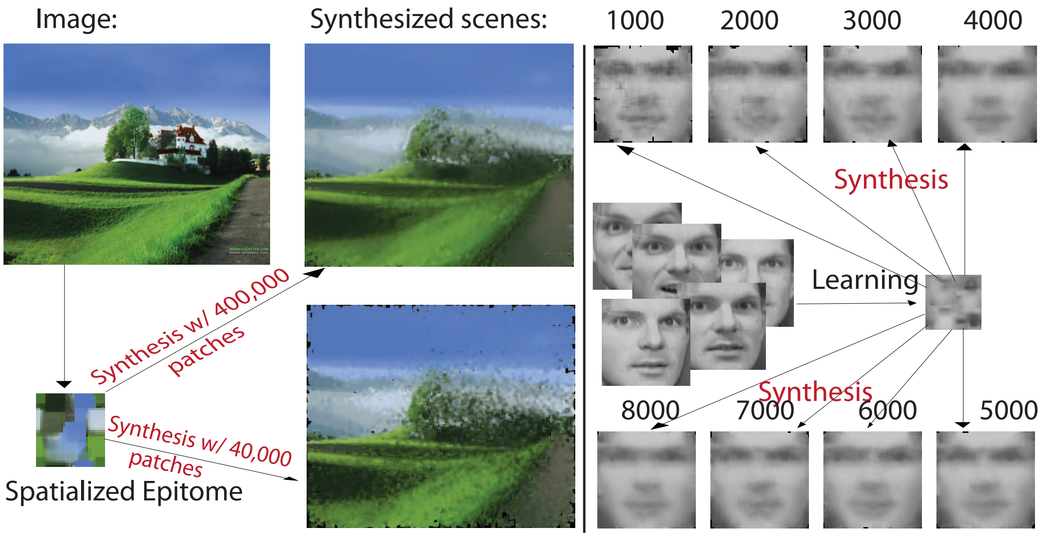


Figure 9.1: The left half of the figure shows the synthesis results for a spatialized epitome learned from a scene image. At the right half of the figure, we show synthesis results for a spatialized epitome model learned from multiple images from the same person.

uses the intensity image as the input and give out the results in probability terms. In order to evaluate the effectiveness of including spatial information, we need to derive a recognition algorithm for the original epitome proposed in [82, 78]. Following the same principle in Chapter 8, the inferred probability of seeing a new image with original epitome is:

$$\begin{aligned}\log P(I|\mathcal{D}) \simeq \log P(I|\hat{\mathbf{e}}) &= \log P(\{\mathbf{Z}_k\}_{k=1}^{P}|\hat{\mathbf{e}})\\ &= \sum_{k=1}^{P} \log \sum_{\mathcal{T}_k} \prod_{i \in S_k} \mathcal{N}(z_{i,k}; \mu_{\mathcal{T}_k(i)}, \phi_{\mathcal{T}_k(i)}) P(\mathcal{T}_k). \quad (9.1)\end{aligned}$$

In this experiment, two benchmark face databases, e.g. ORL and CMU PIE [1] are used. The ORL database contains 400 images of 40 persons, where each image is manually cropped and normalized to the size of 32 × 32 pixels. The CMU PIE (Pose, Illumination, and Expression) database contains more than 40,000 facial images of 68 people. In our experiment, a subset of five near frontal poses (C27, C05, C29, C09, and C07) with illumination indexed as 08 and 11 are used and manually normalized to the size of 32 × 32 pixels. Both original and spatialized epitomes are evaluated with two different patch sizes.

[1] Available at http://www.face-rec.org/databases/.

Table 9.1: Recognition accuracy rates (%) on two face databases.

Database:	ORL		PIE	
Patch Size:	4×4	6×6	4×4	6×6
Epitome	12.0	15.5	8.2	11.2
Spatialized	67.5	88.5	74.1	78.8

We can observe from Table 9.1 that the incorporation of spatial information considerably increases the recognition accuracy. Therefore, the performance of original epitome in later more complex applications are not evaluated.

9.3 Occlusion Detection

For a facial image with occlusions, the occluded parts can be revealed by evaluating the likelihood for one patch or a set of few nearby patches by Eq. (8.1). The set of patch samples with the probabilities lower than a certain threshold are considered to be the patches that are occluded. In this experiment we examine the occlusion detection capability of our spatialized epitome formulation on the CMU PIE and ORL databases. We randomly pick five images of each subject for training; the remaining five images of each person serve as probe images. Then an 18×18 artificial occlusion is generated at a random position in each probe image. Seven images are randomly selected from the probe set and the occlusion detection results are shown in Figure 9.2, where the first row shows the original face images, the second row shows the images with occlusions, the third row shows the detected occlusion regions, and the fourth row shows the reconstructed images by the spatialized epitome.

9.4 Face Recognition with Misalignments

In most of the techniques for face recognition, explicit semantics is assumed for each feature. But for computer vision tasks, e.g., face recognition, the explicit semantics of the features may be degraded by *spatial misalignments.* Face cropping is an inevitable step in an automatic face recognition system,

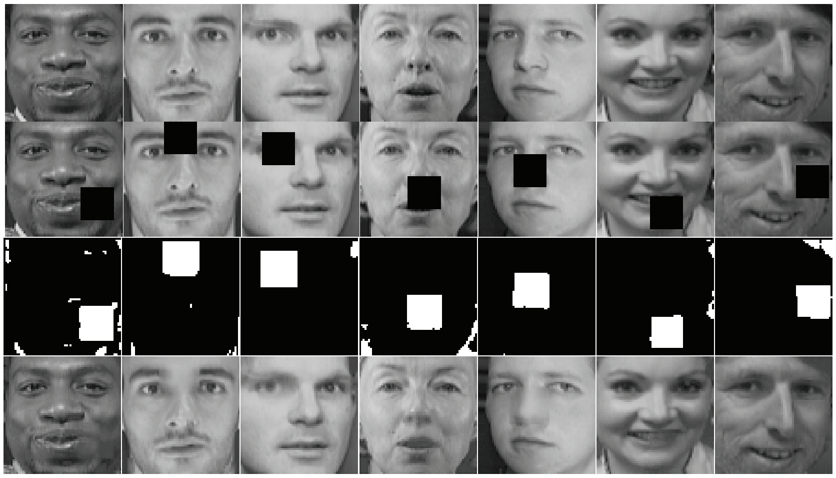


Figure 9.2: Examples of occlusion detection.

and the success of subspace learning for face recognition relies heavily on the performance of the face detection and face alignment processes. Practical systems or even manual face cropping, may bring considerable image misalignments, including translations, scaling and rotation, which consequently change the semantics of two pixels with the same index but in different images [87]. To a certain extent, the spatialized epitome proposed here can naturally adapt to misaligned inputs because: (1) a moderate amount of coordinate shifts caused by the misalignments can also have a high probability value under a Gaussian mixture distribution as long as the "data point" is still in the vicinity; (2) the spatialized epitome is learned from patches of images of different expressions (ORL) or different poses (PIE), so the deformation is learned to account for misalignments on the patch level; and (3) the misalignment effect is reduced from the image level to patch level. We evaluate the performance of our algorithm with respect to each of the misalignment factors, e.g., translation, scaling, and rotation as well as the mixed spatial misalignments to simulate the misalignments brought by the automatic face alignment process. These experiments are also conducted on two benchmark face databases, e.g. ORL and PIE with spatial misalignments for the testing data and no misalignments for the training data. A set of four images from each subject is used for training while the remaining six images of each person are artificially misaligned with a rotation $\alpha \in [-5°, 5°]$, a scaling $s \in [0.95, 1.05]$, a horizontal shift $T_x \in [-1, +1]$, or a vertical shift $T_y \in [-1, +1]$. The value of each misalignment factor is drawn from a uni-

Table 9.2: Recognition accuracy rates (%) on two databases with mixed misalignments. The patch size of 6×6 is used in both learning and recognition.

Database:	ORL			PIE		
Methods	PCA	LDA	Ours	PCA	LDA	Ours
Results	63.2	51.7	88.0	65.9	54.0	67.9

Table 9.3: Cross-pose recognition accuracy rates (%) on PIE database. Each column shows the respective results for each pose. The patch size of 6×6 is used in both learning and recognition.

Methods:	c09	c27	c07	Overall
PCA	34.3	36.1	33.4	34.6
LDA	65.3	66.3	49.1	60.2
Ours	82.4	66.2	72.1	73.6

form distribution. The performance of our algorithm for each misalignment factor is evaluated in Table 9.2 and compared with baseline algorithms such as PCA and LDA (the results come from [87] with four training samples). In the mixed spatial misalignment configuration, the aforementioned effects are added in a random order to the original test image, and the results are shown in Table 9.2.

9.5 Cross-Pose Face Recognition

In the real-world scenario, we may often have to recognize a face with a pose that we have not seen before. We show in this experiment that our spatialized epitome can adapt to unknown pose variations to a certain extent. Here we use a different subset of the PIE database. For each subject in the PIE database, three images with illumination index 8, 11, 21 from each of the two near frontal poses, namely c05 and c29 are chosen as training set. Three images from each of the five different poses (c09, c27, c07, c37, and c11) for each subject are then selected for testing. In both learning and testing, we use patch size of 6×6. Detailed results and comparison with PCA and LDA (with K-nearest neighbor classifier) baselines are listed in Table 9.3.

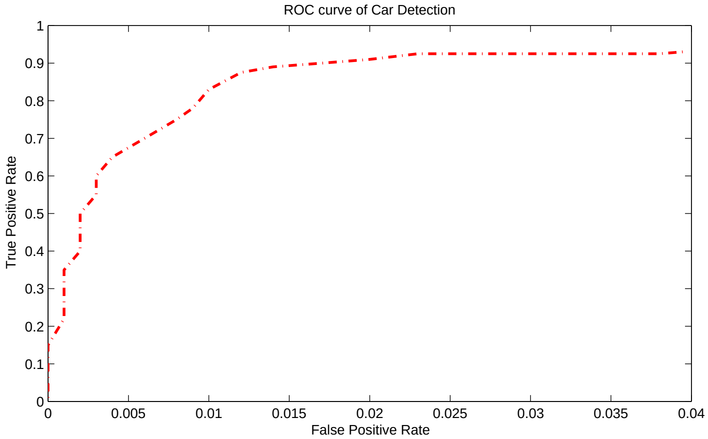


Figure 9.3: The ROC curve of car detection.

9.6 Car Detection

In order to show the detection ability of our spatialized epitome, the UIUC side-view car dataset[2] was used for evaluation. Six representative cars are chosen for learning the car model. During learning, we use gradient images which are extracted from the six Gaussian-smoothed positive training images. We slide the window of size 30×90 over the entire query image and calculate the probability value given by Eq. (8.1). The windows that have probability values above a threshold t are considered to be the locations of the cars. We evaluate performance by comparing the bounding box of detection to the "ground truth" bounding box B_t in manually annotated data. We follow the procedure adopted in the Pascal VOC competition, and compute the area ratio a of $B_p \bigcap B_t$ and $Bp \bigcup Bt$. If $a > 0.5$, then B_p is considered a true positive. By varying the threshold on this confidence, we compute the ROC curve as shown in Figure 9.3. Our method achieves reasonable performance under a less restrictive condition which requires a few training samples and no negative training samples are needed. In this case, conventional supervised learning algorithms are not applicable.

In these experiments, we have shown the strong abilities of spatialized epitome for image representation, pattern recognition, and object detection. Especially, the tests on some tough vision tasks like misaligned and cross-pose face recognition demonstrate the advantages of the spatialized epitome

[2] http://l2r.cs.uiuc.edu/ cogcomp/Data/Car/.

in adapting to variations in real-world conditions.

CHAPTER 10

COLORIZATION BY EPITOME

Colorization adds color to grayscale images by assigning color values to images which only contain a grayscale channel. It not only increases the visual appeal, but also enhances the information conveyed by scientific images. For example, the grayscale images acquired by scanning electron microscopy (SEM) can be made more illustrative by adding different colors to different parts of the images. However, the manual colorization is tedious and time consuming, so it is not suitable for batch process. To overcome this problem, we propose an automatic colorization method by epitome. We train the epitome from one manually colorized nano mushroom-like image, and use that epitome to automatically colorize the other nano mushroom-like image, which eliminates the need for human labor and makes the batch colorization process possible.

Based on the source of the color information used to colorize the grayscale images, existing colorization techniques fall into two main categories: user scribble based methods and color transfer methods. The user scribble based method in [88] asked users to draw color scribbles in the grayscale image, and the algorithm propagated the user-provided color to the whole image requiring that similar neighboring pixels should receive similar color. Later, L. Qing et al. [89] proposed a method which required less human intervention. The user scribbles were employed for texture segmentation and user-provided color was propagated within each segment. Using a similar color image as a reference, the color transfer methods such as [90] performed colorization by transferring the color from the reference image to the grayscale image, either automatically or with user intervention. However, the pixel-level matching based on luminance value and neighborhood statistics adopted by [90] suffered from spatial inconsistency and the user-provided swatches were required to guide the matching process in many cases. Using [91] improved the spatial consistency by an image space voting scheme. Their method first transferred color to a few pixels in the target image with high confidence, then applied the method in [88] to colorize the whole image, treating the colorized pixels in the first step as the scribbles. However, their method required a robust segmentation of the reference image, which was difficult in many cases without user intervention.

Similar to [90], our automatic colorization method transfers the color information from the reference image to the target grayscale image. Since most existing colorization methods need user interactions for color selection or segmentation, a robust and automatic colorization algorithm is preferable. In order to approach this problem, it is worthwhile to exploit the biological characteristics of human visual system. The average human retina contains many more rods than cones [92] (92 million rods versus 4.6 million cones). Rods are more sensitive to cones but they are not sensitive to color, so that most of visually significant variation arises only from luminance differences. This fact suggests that we do not need to search the whole reference image for the color patches to colorize the target image; instead we can reduce the search space for color patches, or equivalently find an effective color summary of the reference image, to improve the efficiency and alleviate color assignment ambiguity. In [90], such a color summary is a set of source color pixels randomly sampled, which is, however, subject to noise in the raw pixels.

In order to find an effective and compact summary of the color information in the reference image, we adopt the condensed image appearance and shape representation, i.e. epitome [93]. Epitome consolidates self-similar patches in the spatial domain, and the size of the epitome is much smaller than that of the image it models. By virtue of the generative graphical model, epitome can be interpreted as a tradeoff between template and histogram for image representation and it has been applied to many computer vision tasks such as object detection, location recognition, and synthesis [94, 35]. Epitome summarizes a large number of raw patches in the reference image by only representing the most constitutive elements. In our epitomic colorization scheme, the color patches used to colorize the target grayscale image are retrieved from the epitome trained with the reference image, rather than from the raw image patches. Epitome proves to be an effective summary of the color information in the reference image, which produces more satisfactory colorization results than [90] in the experiments.

10.1 Description of Automatic Colorization by Epitome

Given a reference color image cI and the target grayscale image gI, we aim to automatically colorize gI with the color information from cI. We achieve this goal by first training an epitome e from the reference image, then performing inference in e so as to transfer the color information of the color patches of $\hat{\mathbf{e}}$ to the corresponding grayscale patches of gI. Note that the grayscale channel of gI is retained as the luminance channel after the color transfer process. We will illustrate the training and inference process in detail in the following subsections.

10.2 Training the Epitome

Epitome is a latent representation of an image, which comprises hidden variables and parameters required to generate the image patches according to the epitome graphical model. Epitome summarizes a large set of raw image patches into a condensed representation of a size much smaller than the original image, and it approaches this goal in a manner similar to Gaussian mixture model with overlapping means and variances.

The epitome e of an image I of size $M \times N$ is a condensed representation of size $M_e \times N_e$ where $M_e < M$ and $N_e < N$. The epitome contains two parameters: $\mathbf{e} = (\boldsymbol{\mu}, \boldsymbol{\phi})$. $\boldsymbol{\mu}$ and $\boldsymbol{\phi}$ represent the Gaussian mean and variance, respectively, and both are of size $M_e \times N_e$. Suppose Q patches are sampled from the reference image, i.e. $\{\mathbf{Z}_k\}_{k=1}^{Q}$, and each patch $\mathbf{Z}_k$ contains pixels with image coordinates $\mathbf{S}_k$. Similar to [93], the patches are square and we use fixed patch size throughout this chapter. These patches are densely sampled and they can be overlapping with each other to cover the entire image. We associate each patch $\mathbf{Z}_k$ with a hidden mapping $\mathcal{T}_k$ which maps the image coordinates $\mathbf{S}_k$ to the epitome coordinates, and all the Q patches are generated independently from the epitome parameters and the corresponding hidden mappings as follows:

$$p(\mathbf{Z}_k|\mathcal{T}_k, \mathbf{e}) = \prod_{i \in \mathbf{S}_k} \mathcal{N}(z_{i,k}; \boldsymbol{\mu}_{\mathcal{T}_k(i)}, \boldsymbol{\phi}_{\mathcal{T}_k(i)}), k = 1..Q \tag{10.1}$$

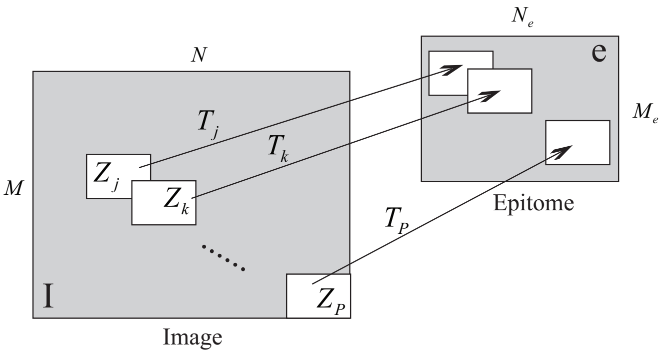


Figure 10.1: The mapping $\mathcal{T}_k$ maps the image patch $\mathbf{Z}_k$ to its corresponding epitome patch with the same size, and $\mathbf{Z}_k$ can be mapped to any possible epitome patch according to $\mathcal{T}_k$.

and

$$\prod_{k=1}^{Q} p(\{\mathbf{Z}_k\}_{k=1}^{Q} | \{\mathcal{T}_k\}_{k=1}^{Q}, \mathbf{e}) = \prod_{k=1}^{Q} p(\mathbf{Z}_k | \mathcal{T}_k, \mathbf{e}). \tag{10.2}$$

where $z_{i,k}$ is the pixel with image coordinates i from the k-th patch. Since $z_{i,k}$ is independent of the patch number k, we simply denote it as z_i in the following text. $\mathcal{N}(\cdot; \mu, \phi)$ represents a Gaussian distribution with mean $\hat{\mu}$ and variance $\hat{\phi}$

$$\mathcal{N}(\cdot; \hat{\mu}, \hat{\phi}) = \frac{1}{\sqrt{2\pi\hat{\phi}}} \exp^{-\frac{(\cdot - \hat{\mu})^2}{2\hat{\phi}}}.$$

Based on Eq. (10.1), the hidden mapping $\mathcal{T}_k$ can be interpreted as a hidden variable that indicates the location of the epitome patch from which the observed image patch $\mathbf{Z}_k$ is generated, and it behaves similar to the hidden variable in the traditional Gaussian mixture models that specifies the Gaussian component from which a specific data point is generated. Also, $\mathcal{T}_k$ maps the image patch to its corresponding epitome patch, and the number of possible mappings that each $\mathcal{T}_k$ can take, denoted as L, is determined by all the discrete locations in the epitome ($L = M_e \times N_e$ in our setting). Figure 10.1 illustrates the role that the hidden mapping variables play in the generative model, and Figure 10.2 shows the epitome graphical model, which again demonstrates its similarity to Gaussian mixture models. $\boldsymbol{\pi} \overset{\Delta}{=} \{\pi_l\}_{l=1}^{L}$ indicates the prior distribution of the hidden mapping. Suppose $\mathcal{T}_{k,l}$ is the l-th mapping that $\mathcal{T}_k$ can take, then

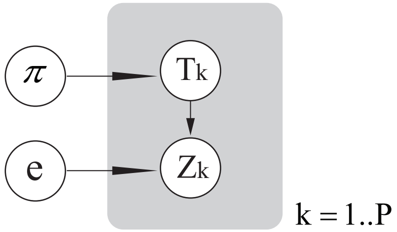


Figure 10.2: The epitome graphical model.

$$p(\mathcal{T}_k) = \prod_{l=1}^{L} {\pi_l}^{\delta\left(\mathcal{T}_k = \mathcal{T}_{k,l}\right)},$$

which holds for any $k \in \{1..Q\}$. δ is an indicator function and δ equals to 1 when its argument is true, and 0 otherwise.

Our goal is to find the epitome $\hat{\mathbf{e}}$ that maximizes the log likelihood function:

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}} \log p\left(\{\mathbf{Z}_k\}_{k=1}^{Q} | \mathbf{e}\right). \tag{10.3}$$

Given the epitome $\mathbf{e}$, the likelihood function for the complete data, i.e. the image patches $\{\mathbf{Z}_k\}_{k=1}^{Q}$ and the hidden mappings $\{\mathbf{Z}_k\}_{k=1}^{Q}$, is derived in the following according to the epitome graphical model:

$$\begin{aligned} p(\{\mathbf{Z}_k, \mathcal{T}_k\}_{k=1}^{Q} | \mathbf{e}, \boldsymbol{\pi}) &= \prod_{k=1}^{Q} p(\mathbf{Z}_k, \mathcal{T}_k | \mathbf{e}, \boldsymbol{\pi}) \\ &= \prod_{k=1}^{Q} p(\mathcal{T}_k) p(\mathbf{Z}_k | \mathcal{T}_k, \mathbf{e}) \\ &= \prod_{k=1}^{Q} \prod_{l=1}^{L} \left[\pi_l \prod_{j \in \mathbf{S}_k} \mathcal{N}(z_j; \boldsymbol{\mu}_{\mathcal{T}_{k,l}(j)}, \boldsymbol{\phi}_{\mathcal{T}_{k,l}(j)}) \right]^{\delta\left(\mathcal{T}_k = \mathcal{T}_{k,l}\right)} \end{aligned} \tag{10.4}$$

We use the expectation-maximization algorithm [95] to maximize the likelihood function Eq. (10.3) and learn the epitome $\hat{\mathbf{e}}$, following the procedure introduced in [96].

The E-step: The posterior distribution of the hidden variables, i.e. the hidden mapping is

$$
\begin{aligned}
q(\mathcal{T}_k) &\overset{\Delta}{=} p(\mathcal{T}_k|\mathbf{Z}_k, \mathbf{e}, \boldsymbol{\pi}) \\
&= \frac{p(\mathbf{Z}_k|\mathcal{T}_k, \mathbf{e})p(\mathcal{T}_k)}{\sum_{\mathcal{T}_k} p(\mathbf{Z}_k|\mathcal{T}_k, \mathbf{e})p(\mathcal{T}_k)} \\
&= \frac{\prod_{l=1}^{L} \left[\pi_l \prod_{j\in\mathbf{S}_k} \mathcal{N}(z_j; \boldsymbol{\mu}_{\mathcal{T}_{k,l}(j)}, \boldsymbol{\phi}_{\mathcal{T}_{k,l}(j)})\right]^{\delta\left(\mathcal{T}_k=\mathcal{T}_{k,l}\right)}}{\sum_{\mathcal{T}_k} \prod_{l=1}^{L} \left[\pi_l \prod_{j\in\mathbf{S}_k} \mathcal{N}(z_j; \boldsymbol{\mu}_{\mathcal{T}_{k,l}(j)}, \boldsymbol{\phi}_{\mathcal{T}_{k,l}(j)})\right]^{\delta\left(\mathcal{T}_k=\mathcal{T}_{k,l}\right)}}.
\end{aligned}
\tag{10.5}
$$

We observe that $q(\mathcal{T}_k)$ corresponds to the responsibility in Gaussian mixture models.

The M-step: We obtain the expectation of the log-likelihood function for the complete data with respect to the posterior distribution of the hidden mapping from the E-step as follows:

$$
\begin{aligned}
&E\left[\log p\left(\{\mathbf{Z}_k, \mathcal{T}_k\}_{k=1}^{Q} |\mathbf{e}, \boldsymbol{\pi}\right)\right] \\
&= \sum_{k=1}^{Q}\sum_{l=1}^{L} q(\mathcal{T}_k = \mathcal{T}_{k,l}) \cdot [\log \pi_l + \log p\left(\mathbf{Z}_k|\mathcal{T}_k = \mathcal{T}_{k,l}, \mathbf{e}\right)].
\end{aligned}
\tag{10.6}
$$

Maximizing Eq.(10.6) with respect to $(\mathbf{e}, \boldsymbol{\pi})$, we get the following update of the parameters of the epitome and $\boldsymbol{\pi}$:

$$
\boldsymbol{\mu}_j = \frac{\sum\limits_{k=1}^{Q} \sum_{i\in\mathbf{S}_k} \sum_{\mathcal{T}_k} \delta(\mathcal{T}_k(i) = j) q(\mathcal{T}_k) z_i}{\sum\limits_{k=1}^{Q} \sum_{i\in\mathbf{S}_k} \sum_{\mathcal{T}_k} \delta(\mathcal{T}_k(i) = j) q(\mathcal{T}_k)}
\tag{10.7}
$$

$$
\boldsymbol{\phi}_j = \frac{\sum\limits_{k=1}^{Q} \sum_{i\in\mathbf{S}_k} \sum_{\mathcal{T}_k} \delta(\mathcal{T}_k(i) = j) q(\mathcal{T}_k) (z_i - \boldsymbol{\mu}_j)^2}{\sum\limits_{k=1}^{Q} \sum_{i\in\mathbf{S}_k} \sum_{\mathcal{T}_k} \delta(\mathcal{T}_k(i) = j) q(\mathcal{T}_k)}
\tag{10.8}
$$

$$
\pi_l = \frac{\sum\limits_{k=1}^{Q} p\left(\mathcal{T}_k = \mathcal{T}_{k,l}\right)}{Q}, l = 1..L.
\tag{10.9}
$$

The index j indicates the epitome coordinates in Eq. (10.7) and Eq. (10.8).

We alternate between E-step and M-step until convergence or the maximum number of iterations (20 in our experiments) is achieved, and then obtain the resultant epitome $\hat{\mathbf{e}}$ from the reference image cI.

Note that the preceding training process is applicable for a single type of feature of cI. We use two types of feature to train the epitome, i.e. the YIQ channels and the dense sift feature [28]. We convert cI from the RGB color space to the YIQ color space where Y channel represents the luminance and IQ channels represent chrominance information. Moreover, dense sift feature is computed for each sampled patch. A $K \times K$ patch is evenly divided into $R \times R$ grids, and the orientation histogram of the gradients with eight bins is calculate for each grid, which results in an $8R^2$-dimensional dense sift feature vector for each patch. R is typically set to be 3 or 4. We then train the epitome $\mathbf{e} = \left(\mathbf{e}^{YIQ}, \mathbf{e}^{dsift}\right)$ for the YIQ channels and the dense sift feature, and the epitome for YIQ channels ($\mathbf{e}^{YIQ}$) share the same hidden mapping with the epitome for the dense sift feature ($\mathbf{e}^{dsift}$) in the inference process [94]:

$$p(\mathbf{Z}_k|\mathcal{T}_k, \mathbf{e}) = p(\mathbf{Z}_k^{YIQ}|\mathcal{T}_k, \mathbf{e}^{YIQ})^{\lambda} p(\mathbf{Z}_k^{dsift}|\mathcal{T}_k, \mathbf{e}^{dsift})^{1-\lambda}, \tag{10.10}$$

where $\mathbf{Z}_k^{YIQ}$ and $\mathbf{Z}_k^{sift}$ represent the YIQ channel and the dense sift feature of patch $\mathbf{Z}_k$ respectively, $\mathbf{e}^{YIQ}$ and $\mathbf{e}^{dsift}$ represent the epitome trained from the YIQ channels and dense sift feature of cI respectively. $0 \leq \lambda \leq 1$ is a parameter balancing the preference between color and dense sift feature.

10.3 Colorization by Epitome

With the epitome $\hat{\mathbf{e}}$ learned from the reference image, we colorize the target grayscale image gI by inference in the epitome graphical model. Similar to the epitome training process, we densely sample $\hat{Q}$ patches $\{\hat{\mathbf{Z}}_k\}_{k=1}^{\hat{Q}}$ from gI (these patches cover the entire gI). With the hidden mapping associated with patch $\hat{\mathbf{Z}}_k$ denoted as $\hat{\mathcal{T}}_k$, the most probable mapping of the patch $\hat{\mathbf{Z}}_k$, i.e. $\hat{\mathcal{T}}_k^*$, is formulated as follows:

$$\hat{\mathcal{T}}_k^* = \arg\max_{\hat{\mathcal{T}}_k} p\left(\hat{\mathcal{T}}_k|\hat{\mathbf{Z}}_k, \hat{\mathbf{e}}, \boldsymbol{\pi}\right) \tag{10.11}$$

which is essentially the same as the E-step Eq. (10.5). We take the grayscale channel of gI as the luminance channel (Y channel) of itself. Since the color information (IQ channels) is absent in gI, we only use the epitomes corresponding to the Y channel and the dense sift feature to evaluate the right-hand side of Eq. (10.12). The color information is then transferred from the epitome patch, whose location is specified by $\hat{\mathcal{T}}_k^*$, to the grayscale patch $\hat{\mathbf{Z}}_k$. We denote the target image after colorization as gI_c. Since $\{\hat{\mathbf{Z}}_k\}_{k=1}^{\hat{Q}}$ can be overlapping with each other, the final color (the value of IQ channels) of a pixel i in image gI_c is averaged according to:

$$gI_c\left(i\right) = \frac{\sum\limits_{k=1}^{\hat{Q}} \sum\limits_{j \in \hat{S}_k} \delta\left(j = i\right) \hat{\mathbf{e}}_{\hat{\mathcal{T}}_k^*(j)}^{IQ}}{\sum\limits_{k=1}^{\hat{Q}} \sum\limits_{j \in \hat{S}_k} \delta\left(j = i\right)}, \tag{10.12}$$

where $\hat{S}_k$ is the image coordinates of patch $\hat{\mathbf{Z}}_k$, and $\mathbf{e}_{\hat{\mathcal{T}}_k^*(j)}^{IQ}$ represents the value of the IQ channels in the epitome $\mathbf{e}$ at location $\hat{\mathcal{T}}_k^*\left(j\right)$.

10.4 Experimental Results

We show colorization results in this section. As mentioned in section 10.2, we use square patches of size $K \times K$, and the size of epitome is half of the size of the reference image. We densely sample patches with horizontal and vertical gap of ωK pixels, where ω is a parameter between $[0, 1]$ and it controls the number of sampled patches.

Figure 10.3 shows the result of colorization for the dog image. We convert the original image to grayscale as the target image. The patch size is 12×12 and the parameter λ balancing between the color and the dense sift feature is 0.5. We compare our method to [90] which transfers color from the reference image to the target image by pixel-level matching. The result produced by [90] lacks spatial continuity and we observe small artifacts throughout the whole image. On the contrary, our method renders a colorized image very similar to the ground truth. This example also demonstrates that the learned epitome, which is a summary of a large number of sampled patches, contains sufficient color information for colorization.

Figures 10.4 and 10.5 show the colorization result for the nano mushroom-like images and the cheetah. The patch size is chosen as 12 × 12 and 15 × 15, respectively, and λ is set to be 0.8 for both cases. The method of [90] still generates artifacts around the top and bottom of the mushroom-like structure, while our method produces a much more spatially coherent result. Moreover, we transfer the correct color for the cheetah to the target image, which results in a more natural colorization result than that of [90].

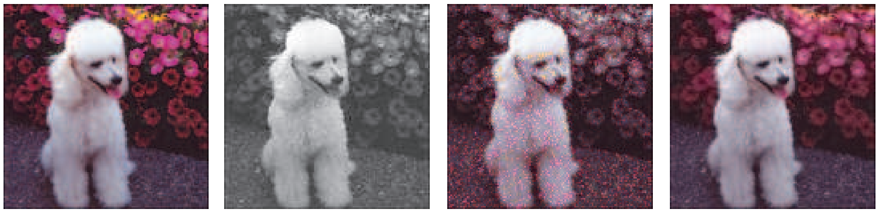


Figure 10.3: The result of colorizing the dog. From left to right: the reference image, the target image (obtained by converting the reference image to the grayscale), the result by [90], and our result.

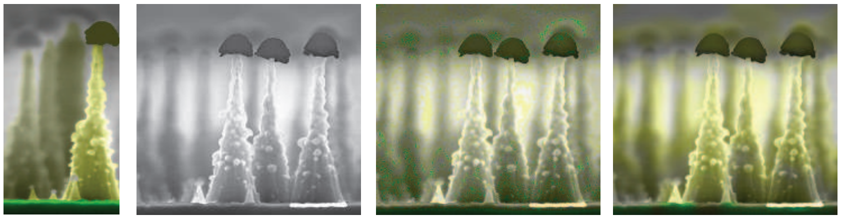


Figure 10.4: The result of colorizing the nano mushroom-like images.

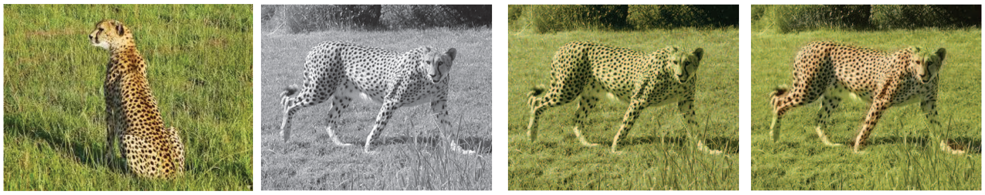


Figure 10.5: The result of colorizing the cheetah.

CHAPTER 11

CONCLUSIONS

In this thesis, we proposed layout-aware models for the problems of object detection and recognition. A discriminative latent layout model is proposed for pose as well as translation normalization. The new model is capable of normalizing with respect to variations for both detection and recognition problems and achieves the state-of-the-art result in the joint detection and its subcategory recognition task. In a generative setting, a new graphical model for epitome, i.e. the spatialized epitome, is proposed to model spatial layout of the patches. The new epitome model integrates both the local appearance and spatial arrangement for image representation. Employing the powerful generative model framework in both learning and inference, the spatialized epitome is flexible for image representation, discriminative for pattern recognition, adaptive to variation, and robust for object detection. Experiments on several tough vision tasks have shown its superiority over the original epitome model in image modeling. In addition, we present an automatic colorization method using epitome in this thesis. While most existing colorization methods require tedious and time-consuming user intervention for scribbles or segmentation, our epitomic colorization method is automatic. Epitomic colorization exploits the color redundancy by summarizing the color information in the reference image into a condensed image shape and appearance representation. Experimental results show the effectiveness of our methods.

REFERENCES

[1] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, “Overfeat: Integrated recognition, localization and detection using convolutional networks,” in *International Conference on Learning Representations (ICLR 2014)*. CBLS, April 2014.

[2] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 8, pp. 1798–1828, Aug 2013.

[3] J. Yang, K. Yu, and T. Huang, “Supervised translation-invariant sparse coding,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[4] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach, “Supervised dictionary learning,” in *Advances in Neural Information Processing Systems 21*, 2009, pp. 1033–1040.

[5] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” in *Science*, 2006.

[6] S. Geman, “Neural networks and the bias/variance dilemma,” in *Neural Computation*, 1992.

[7] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. Bourdev, “Panda: Pose aligned networks for deep attribute modeling,” in *CVPR*, 2014.

[8] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, “Real-time human pose recognition in parts from a single depth image,” in *CVPR*, 2011.

[9] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. Kandola, “On kernel-target alignment,” in *NIPS*, 2001.

[10] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *ICML*, 2014.

[11] Y. Sun, X. Wang, and X. Tang, “Deep learning face representation by joint identification-verification,” in *NIPS*, 2014.

[12] A. A. Efros and T. K. Leung, “Texture synthesis by non-parametric sampling,” in *IEEE International Conference on Computer Vision*, Corfu, Greece, September 1999, pp. 1033–1038.

[13] A. Criminisi, P. Perez, and K. Toyama, “Object removal by exemplar-based inpainting,” in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 2, 2003, pp. II–721–II–728 vol.2.

[14] I. Drori, D. Cohen-Or, and H. Yeshurun, “Fragment-based image completion,” in *ACM SIGGRAPH 2003 Papers*, ser. SIGGRAPH ’03. New York, NY, USA: ACM, 2003. [Online]. Available: http://doi.acm.org/10.1145/1201775.882267 pp. 303–312.

[15] A. Buades, B. Coll, and J. M. Morel, “A non-local algorithm for image denoising,” in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, 2005, pp. 60–65 vol. 2.

[16] J. Yang, J. Wright, T. Huang, and Y. Ma, “Image super-resolution as sparse representation of raw image patches,” in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 2008, pp. 1–8.

[17] D. Lowe, “Distinctive image features from scale-invariant keypoints,” *IJCV*, 2004.

[18] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” 2002, iEEE Transactions on Pattern Analysis and Machine Intelligence 24(7):971 - 987.

[19] H. Bay, T. Tuytelaars, and L. Gool, “Surf: Speeded up robust features,” in *Computer Vision ECCV 2006*, ser. Lecture Notes in Computer Science, A. Leonardis, H. Bischof, and A. Pinz, Eds. Springer Berlin Heidelberg, 2006, vol. 3951, pp. 404–417.

[20] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, “Deepface: Closing the gap to human-level performance in face verification,” in *Conference on Computer Vision and Pattern Recognition (CVPR).*

[21] L. V. Gool, T. Moons, E. Pauwels, and A. Ooserlinck, “Vision and lies approach to invariance,” in *Image and Vision Computing*, 1995.

[22] X. Shi and R. Manduchi, “Invariant operators, small samples, and the bias-variance dilemma,” in *CVPR*, 2004.

[23] M. Varma and D. Ray, “Learning the discriminative power-invariance trade-off,” in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, Oct 2007.

[24] A. Krizhevsky, I. Sutskever, and G. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 25*, 2012.

[25] J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, and L. Fei-Fei., “Imagenet large scale visual recognition competition,” in *ILSVRC*, 2013.

[26] T. Hofmann, “Probabilistic latent semantic analysis,” in *In Proc. of Uncertainty in Artificial Intelligence, UAI99*, 1999, pp. 289–296.

[27] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, “Local features and kernels for classification of texture and object categories: A comprehensive study,” in *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW ’06. Conference on*, 2006, pp. 13–13.

[28] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: S-patial pyramid matching for recognizing natural scene categories,” in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2, 2006, pp. 2169–2178.

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

[30] F. Perronnin and C. Dance, “Fisher kernels on visual vocabularies for image categorization,” in *Computer Vision and Pattern Recognition, 2007. CVPR ’07. IEEE Conference on*, 2007, pp. 1–8.

[31] X. Zhou, N. Cui, Z. Li, F. Liang, and T. Huang, “Hierarchical gaussianization for image classification,” in *Computer Vision, 2009 IEEE 12th International Conference on*, 2009, pp. 1971–1977.

[32] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani, “Summarizing visual data using bidirectional similarity,” in *CVPR*, 2008.

[33] A. Kapoor and S. Basu, “The audio epitome: A new representation for modeling and classifying auditory phenomena,” in *ICASSP*, 2005.

[34] N. Cuntoor and R. Chellappa, “Epitomic representation of human activities,” in *CVPR*, 2007.

[35] X. Chu, S. Yan, L. Li, K. L. Chan, and T. S. Huang, “Spatialized epitome and its applications,” in *CVPR*, 2010, pp. 311–318.

[36] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," Tech. Rep. CNS-TR-2011-001, 2011, California Institute of Technology.

[37] Y. Yang, X. Chu, T. T. Ng, A.-S. Chia, J. Yang, H. Jin, and T. Huang, "Epitomic image colorization," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, May 2014.

[38] S. Avidan and A. Shamir, "Seam carving for content-aware image resizing," in *ACM SIGGRAPH 2007 papers*, ser. SIGGRAPH '07, 2007.

[39] L. Wolf, M. Guttmann, and D. Cohen-Or, "Non-homogeneous content-driven video-retargeting," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, 2007, pp. 1–6.

[40] Y. Yang, X. Chu, F. Liang, and T. S. Huang, "Pairwise exemplar clustering," in *AAAI*, 2012.

[41] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2005.

[42] P. Felzenszwalb, R.B.Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," in *IEEE PAMI*, 2010.

[43] K. T. Murphy, A. Eaton, and W. D. Freeman, "Object detection and localization using local and global features," *Lecture notes in computer science*, 2006.

[44] L. Bourdev and J. Malik, "Poselets: Body part detectors trained using 3d human pose annotations," in *Computer Vision, 2009 IEEE 12th International Conference on*, Sept 2009, pp. 1365–1372.

[45] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[46] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, in *The PASCAL Visual Object Classes Challenge 2007*, 2007.

[47] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, in *The PASCAL Visual Object Classes Challenge 2010*, 2010.

[48] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *CVPR Workshop*, 2014.

[49] C. Gring, E. Rodner, A. Freytag, and J. Denzler, "Nonparametric part transfer for fine-grained recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[50] S. Yang, L. Bo, J. Wang, and L. Shapiro, "Unsupervised template learning for fine-grained object recognition," in *NIPS*, 2012.

[51] N. Zhang, R. Farrell, and T. Darrell, "Pose pooling kernels for subcategory recognition," in *CVPR*, 2012.

[52] R. Farrell, O. Oza, N. Zhang, V. I. Morariu, T. Darrell, and L. S. Davis, "Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance," in *ICCV*, 2011.

[53] N. Zhang, R. Farrell, , and T. Darrell, "Deformable part descriptors for fine-grained recognition and attribute prediction," in *ICCV*, 2013.

[54] T. Berg and P. N. Belhumeur, "Poof: Part-based one-vs-one features for fine-grained categorization, face verification, and attribute estimation," in *CVPR*, 2013.

[55] Q. Dai and D. Hoiem, "Learning to localize detected objects," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, June 2012.

[56] M. B. Blaschko and C. H. Lampert, "Learning to localize objects with structured output regression," in *ECCV*, 2008.

[57] V. M. Zatsiorsky and B. I. Prilutsky, *Biomechanics of Skeletal Muscles.* Human Kinetics, 2012.

[58] E. Rosch, C. B. Mervis, W. D. Gray, D. M. Johnson, and P. Boyes-Braem, "Basic objects in natural categories," in *Cognitive Psychology*, 1976, pp. 382–439.

[59] P. Felzenszwalb and D. Huttenlocher, "Efficient matching of pictorial structure," in *CVPR*, 2000.

[60] M. Fischler and R. Elschlager, "The representation and matching of pictorial structures," in *IEEE Transactions on Computers*, 1973.

[61] L. Bourdev, S. Maji, T. Brox, and J. Malik, "Detecting people using mutually consistent poselet activations," in *European Conference on Computer Vision (ECCV)*, 2010.

[62] H. Azizpour and I. Laptev, "Object detection using strongly-supervised deformable part models," in *ECCV*, 2012.

[63] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, “Intriguing properties of neural networks,” in *ICLR*, 2014.

[64] P. Felzenszwalb, D. McAllester, and D. Ramanan, “A discriminatively trained, multiscale, deformable part model,” in *CVPR*, 2008.

[65] J. F. Henriques, J. Carreira, R. Caseiro, and J. Batista, “Beyond hard negative mining: Efficient detector learning via block-circulant decomposition,” in *ICCV*, 2013.

[66] G. Huang, M. Mattar, H. Lee, and E. Learned-Miller, “Learning to align from scratch,” in *Advances in Neural Information Processing Systems 25*, 2012, pp. 773–781.

[67] S. L. M. Cox, S. Sridharan, and J. Cohn., “Least squares congealing for unsupervised alignment of images,” in *CVPR*, 2008.

[68] M. Cox, S. Sridharan, S. Lucey, and J. Cohn, “Least squares congealing for large numbers of images,” in *ICCV*, 2009.

[69] M. A. Mattar, A. R. Hanson, and E. G. Learned-Miller, “Unsupervised joint alignment and clustering using bayesian nonparametrics,” in *UAI*, 2012.

[70] J. Zhu, L. V. Gool, , and S. C. Hoi, “Unsupervised face alignment by nonrigid mapping,” in *ICCV*, 2009.

[71] G. B. Huang, V. Jain, and E. Learned-Miller, “Unsupervised joint alignment of complex images,” in *ICCV*, 2007.

[72] X. Liu, Y. Tong, and F. W. Wheeler, “Simultaneous alignment and clustering for an image ensemble,” in *ICCV*, 2009.

[73] E. Gavves, B. Fernando, S. Snoek, C., and T. A., Tuytelaars, “Fine-grained categorization by alignments,” in *ICCV*, 2013.

[74] J. Canny, “A computational approach to edge detection,” in *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1986.

[75] J. Bezdek, R. Hathaway, R. Howard, C. Wilson, and M. Windham, “Local convergence analysis of a grouped variable version of coordinate descent,” *Journal of Optimization Theory and Applications*, vol. 54, no. 3, pp. 471–477, 1987.

[76] J. Donahue, Y. Jia, O. Vinyals, J. Homan, N. Zhang, E. Tzeng, and T. Darrell, “Decaf: A deep convolutional activation feature for generic visual recognition,” in *ICML*, 2014.

[77] V. Cheung, B. Frey, and N. Jojic, "Video epitomes," in *CVPR*, 2005.

[78] N. Jojic, B. Frey, and A. Kannan, "Epitomic analysis of appearance and shape," in *ICCV*, 2003.

[79] A. Kannan, C. Rother, and J. Winn, "Clustering appearance and shape by learning jigsaws," in *NIPS 19.* Cambridge, MA: MIT Press., 2006.

[80] K. Ni, A. Kannan, A. Criminisi, and J. Winn, "Epitomic location recognition," in *CVPR*, 2008.

[81] J. Warrell, S. Prince, and A. Moore, "Epitomized priors for multi-labeling problems," in *CVPR*, 2009.

[82] V. Cheung, N. Jojic, and D. Samaras, "Capturing long-range correlations with patch models," in *CVPR*, 2007.

[83] H. Wang, Y. Wexler, E. Ofek, and H. Hoppe, "Factoring repeating content within and among images," in *ACM SIGGRAPH*, 2008.

[84] C. Bishop, *Pattern Recognition and Machine Learning.* Springer, 2006.

[85] D. Ormoneit and V. Tresp, "Averaging, maximum penalized likelihood and Bayesian estimation for improving Gaussian mixture probability density estimates," in *IEEE Trans. on Neuro Networks*, 1998.

[86] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2nd ed. Wiley Interscience, 2001.

[87] H. Wang, S. Yan, T. Huang, J. Liu, and X. Tang, "Misalignment-robust face recognition," in *CVPR*, 2008.

[88] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 689–694, 2004.

[89] Q. Luan, F. Wen, D. Cohen-Or, L. Liang, Y.-Q. Xu, and H.-Y. Shum, "Natural image colorization," in *Rendering Techniques*, 2007, pp. 309–320.

[90] T. Welsh, M. Ashikhmin, and K. Mueller, "Transferring color to greyscale images," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 277–280, 2002.

[91] R. Irony, D. Cohen-Or, and D. Lischinski, "Colorization by example," in *Rendering Techniques*, 2005, pp. 201–210.

[92] C. A. Curcio, K. R. Sloan, R. E. Kalina, and A. E. Hendrickson, "Human photoreceptor topography," *Journal of Comparative Neurology*, vol. 292, pp. 497–523, 1990.

[93] N. Jojic, B. J. Frey, and A. Kannan, "Epitomic analysis of appearance and shape," in *ICCV*, 2003, pp. 34–43.

[94] K. Ni, A. Kannan, A. Criminisi, and J. Winn, "Epitomic location recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 12, pp. 2158 –2167, Dec. 2009.

[95] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society: Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[96] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.