Automatic Animal Species Identification Based on Camera Trapping Data

by

Baoliang Wang

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science
University of Alberta

Abstract

The classification of animal images based on camera trapping data is an important and challenging task in the domains of computer vision, machine learning and ecological management. This thesis presents an animal species identification system that can automatically identify the species of an animal captured in an image by a camera trap. We use the Fisher Vector coding approach on top of the dense Scale Invariant Feature Transform features and the cell-structured Local Binary Pattern descriptors to generate a fixed length vector representation for each image and then feed this vector representation to the linear Support Vector Machines for learning and classification. Unlike traditional Bag of Visual Words models that only use a generative method or discriminative method, the powerful Fisher Kernel framework combines the advantages of both generative and discriminative approaches to encode image descriptors and then classify images. The key idea is to characterize an image with a gradient vector derived from a generative probability model and to subsequently feed this gradient vector to a discriminative classifier. Instead of only using zero-order image statistics like in conventional approaches, the Fisher Vector coding method retains zero-order, first-order and second-order information and thus allows less image approximation error. Extensive experimental study shows that our method achieves the highest classification accuracy compared to various conventional.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor Professor Osmar R. Zaïane for the support of my master study and research. His guidance and advice helped me all the way of research and writing of this thesis. I could not have pictured a better supervisor for my master study.

In addition, I would like to thank the rest of my thesis committee: Professor Pierre Boulanger and Professor Erin Bayne for their encouragement, insightful comments, and hard questions.

I would also like to thank Dr. Tyler Muhly and Michelle Hiltz for offering me the summer internship opportunity in their research project and leading me to working on my thesis topic.

In addition, I would like to thank my family: my parents and my brother, for supporting me studying abroad and for everything they did for me.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

The following table describes the significance of various abbreviations and acronyms used throughout the thesis.

Abbreviation	Meaning
AITF	Alberta Innovates Technology Futures
ASIS	Animal Species Identification System
BoW	Bag of Words
BoVW	Bag of Visual Words
CBIR	Content-based Image Retrieval
cLBP	cell-structured Local Binary Pattern
EM	Expectation Maximization
GPS	Global Positioning System
LBP	Local Binary Pattern
LLC	Locality-constrained Linear Coding
p.d.f.	probability density function
RFID	Radio Frequency Identification
SIFT	Scale Invariant Feature Transform
SPM	Spatial Pyramid Matching
SVMs	Support Vector Machines
VLAD	Vector of Locally Aggregated Descriptors

Chapter 1

Introduction

1.1 Motivation

The worldwide industrial development and human infrastructure potentially have great impacts on the ecosystem. The increment of land usage for human society (e.g., city expansion) keeps narrowing the territory of the wildlife and isolating some wildlife species from the others. In addition, oil and gas pipelines and human infrastructure, like highway and railway, could cut off terrestrial animals' migration routes, and thus threaten the survival of those animals. The extensive study of the impact of industrial development and human infrastructure on animal population size should be performed before any construction.

Over the years, ecologists have worked to understand population sizes, distribution and movement of wildlife in order to make informed polices regarding preserving biodiversity and allowing the development of the industry and human infrastructure. Wildlife monitoring and surveying thus is a critical need all over the world.

Recently, advancements in computer engineering and multimedia technologies have enabled the production of digital images and the storage of large scale image collections with little cost. This has popularized the usage of imaging devices and increased the size of image collections, including medical imaging, photo archives, and so on. In the realm of wildlife monitoring and surveying, camera trapping has become one of the most cost-effective methods to conduct animal monitoring and surveying [53, 31]. The motion-triggered camera trap usually provides a visual sen-

sor to record the presence and activity of wide array of species and provides information on location- and/or time-specific information on movement and behaviour.

The Alberta Innovates Technology Futures (AITF)[1] has deployed a large number of remote cameras in the natural environments, such as mountain and forest, in the province of Alberta. Those cameras can capture many species, such as *Deer*, *Bear*, *Wolf*, *Rabbit*, and so on. The camera trapping approach generates a huge volume of monitoring data, which poses new challenges and promises to animal monitoring and management research. Currently, the image sequences are analyzed by ecologists, which is extraordinarily tedious and expensive. For example, it usually takes weeks to months for ecologists to identify the animal species from 100,000 camera trapping images. The naïve way to identify the animal species impedes research progress. Therefore, AIFT has a critical demand of designing a tool that can help them automatically identify the animal species from an immense amount of image data in order to save costs.

AITF has collected a database of labeled animal images, where each label identifies the species of an animal captured in an image. For simplicity, we assume each image contains only one animal or multiple animals that come from the same species. This is typically the case in the AITF database. Only a minor portion of images contain multiple animals of the same species. Notice that the same animal may appear in more than one image, which is not a problem because our goal is to identify animal species rather than counting individual animals. In fact, keeping multiple images of the same animal helps us accommodate viewpoint, pose, scale and appearance changes.

The purpose of the automated animal species identification system (ASIS), or the animal image classification system[2], is to separate animal image sequences into different categories. An ideal animal image classification system should perform like humans and be able to filter out irrelevant images into different categories with arbitrary high accuracy and no hesitation. However, sometimes the problem is difficult and ambiguous even for humans. For example, Figure 1.1 shows two infrared

[1] See more about the Alberta Innovates Technology Futures at http://www.albertatechfutures.ca/

[2] We will use the terms animal species identification and animal image classification interchangeably.

Figure 1.1: Two *Deer* images captured by the remote camera. These two *Deer* are surrounded by the box and it is hard to see them.

images, each of which contains a deer but were very difficult to identify. High dynamic backgrounds, illumination changes, viewpoint changes, and partially captured animal's body make the classification problem even more challenging.

Animal species identification is a challenging research task and will be revolutionary to the field of wildlife management if an automated animal image classification tool is successful. Currently, image classification is a critical topic in the domains of computer vision and machine learning, and remains one of the most difficult tasks in the community. A standard approach for modern image classification system is to extract a set of local patch descriptors from images, encode them into a high dimensional feature space, pool them into a global-, image-level signature, and then classify new images. Figure 1.2 shows the pipeline of modern image classification tasks. Other components such as image pre-processing and classification post-processing might be necessary for specific cases to achieve satisfactory performance.

The paradigm discussed above follows the spirit of Bag of Visual Words (BoVW) model, which was originally borrowed from the text retrieval field. In the field of text retrieval, the Bag of Words (BoW) model is used to represent a document by counting the textual keyword occurrences, and then generates a fixed length vector representation. From a large set of documents, the similarity between two docu-

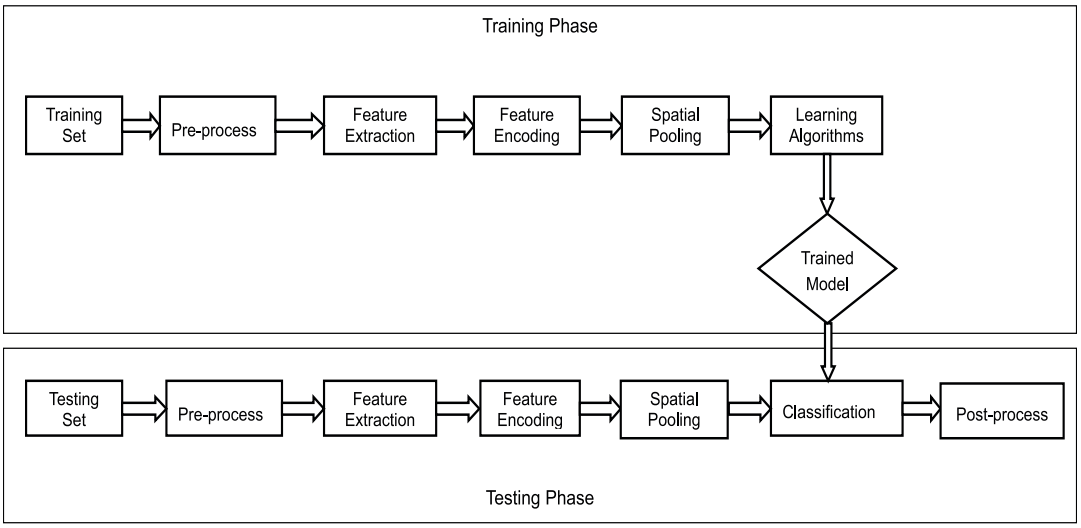


Figure 1.2: The pipeline of general image categorization task.

ments can be easily measured based on the vector representation. This simple idea actually shows the mechanism of common search engine technologies and achieves a great success. Sparked by this success, researchers borrowed the idea and formulated the BoVW model for image classification tasks. For the sake of clarity, we term the representation model as BoVW for image classification while BoW refers to the model used in the text retrieval field.

The BoVW model is the most widely used image approximation model in modern image classification systems. Conventionally, the BoVW model approach uses a clustering algorithm, namely k-means, to create a high dimensional visual codebook based on the extracted local image features. Here a local feature in an image can be considered as a textual word in a document. A codeword in the visual codebook corresponds to the textual keyword. Then, images are quantized into fixed length vectors by counting visual word occurrences based on the codebook. Next, a linear or kernel Support Vector Machines (SVMs) classifier is applied to classify images. In fact, this stream of image classification framework has achieved the best performance on some publicly available datasets [35, 64, 68]. However, there are several shortcomings of the typical BoVW paradigm that weakens the performance of this approach. First, the traditional BoVW model ignores the spatial information in the image because it only counts the visual word frequency. Second, the

conventional pipeline works in a discriminative manner, i.e., it directly estimates a discriminative function for the class label and does not model the statistical distribution of the image data.

The advanced Fisher Vector coding approach [48, 50, 49] is based on the Fisher Kernel framework [26] which combines the advantages of both generative and discriminative approaches. The key idea of the Fisher Kernel is to characterize an image with a gradient vector derived from a generative probability model (e.g., the visual codebook) and to subsequently feed this representation to a discriminative classifier.

1.2 Thesis Statement

This thesis concentrates on building an ASIS that can automatically identify the species of an animal captured in an image by a camera trap. Following the aforementioned image classification pipeline, it is hypothesized that animal species identification can be achieved by representing each image as a point in high dimensional feature space and then classifying an image based on similarity scores with training examples. We aim at addressing the following statements:

1. The dense Scale Invariant Feature Transform (SIFT) features and the cell-structured Local Binary Pattern (cLBP) descriptors could be used for image visual feature coding for the purpose of accurate animal species identification.

2. Using the Fisher Vector coding approach to combine the dense SIFT features and the cLBP descriptors could achieve a higher animal species identification accuracy.

3. The integration of spatial pyramid strategy in the Fisher Vector coding approach would achieve a higher species identification accuracy.

4. The linear SVMs classifier could ensure high computational speed and still have a satisfactory identification performance.

5. It is not necessary to remove the background from an animal image for animal species identification and still have good accuracy.

1.3 Thesis Contribution

Our contribution is fivefold. First, to the best of our knowledge, it is the first work to use the Fisher Vector coding method on top of the dense SIFT and the cLBP descriptors to identify animal species based on camera trapping data. In our work, the proposed method is shown to be effective and efficient to achieve satisfactory performance over the alternatives. Second, the combination of the SIFT and the cLBP is demonstrated to be more effective in describing the animal images for classification purpose. In fact, the SIFT and the cLBP are complementary in describing large patch signature and small cell texture as stated in [62, 66]. Third, we adopt the spatial pyramid strategy to incorporate the rough geometry information in the image, which can improve the classification accuracy. Fourth, applying our approach, we demonstrate the different performances between raw images and images with background manually cropped out. The performance gap on these two image sets clarifies that it is not necessary to remove the background from an animal image for animal species identification and still keep a good accuracy. Lastly, our approach can categorize thousands of images efficiently by applying linear learning and classification algorithms.

1.4 Limitations and Constraints

There are many limitations and constraints to the thesis project. First and foremost, computational cost is vital to the system. Considering that our image dataset usually has a size of tens of gigabytes, it is not possible to load all the data into memory all at once, even for modern computer systems. Even if the image data could fit into memory and accounts for most of the memory capacity, it would be ineffective in terms of computational cost and running time. It is essential to design a system with low memory cost and high running speed.

Other limitations include the way the images are collected and the characteristic of images. The image dataset is collected by fixed camera traps throughout a year in the mountain and forest areas in northeast Alberta. Thus, different backgrounds are present in the image dataset among camera traps. Even for a single fixed cam-

era trap, the background varies largely with seasons. In addition, leaves and twigs are often waving with the wind. Moreover, diurnal images have large illumination changes because of different reflections and cloud movements, and nocturnal images have poor visibility. Last but not least, an animal often occupies a very small part of the image which makes it hard to extract effective features.

Briefly, high dynamic background, different poses, illumination changes and complex non-rigid articulation increase the difficulty to extract features from animal regions and present a challenge to obtain effective and discriminative features. Besides, the huge volume of image data leads to the high demand of computational efficiency.

1.5 Thesis Outline

The rest of the manuscript is organized as follows:

1. Chapter 2 introduces the background knowledge of the thesis and the value of building an automatic ASIS based on camera trapping data. In addition, it presents the previous and related work in the domain of natural image classification and pattern recognition.

2. Chapter 3 describes the dense SIFT feature extraction and the cLBP texture descriptor extraction which are used for describing camera trapping images and discusses the essence of these image features.

3. Chapter 4 largely presents the Fisher Vector coding method on top of the dense SIFT and the cLBP features. Then spatial pyramid strategy is discussed in order to incorporate the rough geometry information. Last, linear and kernel SVMs are briefly described.

4. Chapter 5 discusses the k-means visual codebook learning method, the Vector Quantization coding method [15], the Spatial Pyramid Matching method [35], the Locally-constrained Linear Coding method [61], and the Vector of Locally Aggregated Descriptors [27] for performance comparison purpose.

5. Chapter 6 describes how the camera trapping data is collected and prepared for experiments. In addition, this chapter presents the varying performance with different parameter settings. Detailed analyses about the approach are presented. Furthermore, we briefly illustrate the application we developed for ecologists.

6. Chapter 7 concludes the thesis and highlights some ideas for further work and research.

Chapter 2

Background and Related Work

This chapter introduces the general background knowledge about the subjects of wildlife monitoring, computer vision and machine learning, and then reviews popular image categorization approaches based on the Bag of Visual Words (BoVW) model.

2.1 Background

2.1.1 Wildlife Monitoring

Over the years, the number of wildlife species keeps declining and some animals isolated mostly due to the development of industry and human infrastructure. In order to preserve biodiversity, surveying and monitoring methods that are reliable and efficient for rapid estimation of animal richness and trends are crucial [54].

Counts of dung, nests, trails, calls and direct observation along line transects are widely employed for richness assessment [17, 19, 5, 42, 10, 52]. Figure 2.1 shows an example of line transect sampling. Along the sampling line, there are four animals (four stars with arrows perpendicular to the sampling line) detected. In many cases, surveying signs of those animals has been used due to poor visibility of the actual animals in forests. Whenever animals or signs occur in groups or individually, wildlife professionals can estimate the richness and density by various models. Although the variations of line transect sampling offer efficient and effective ways to monitor wildlife species, they are labour-expensive and do not work well in some situations, for example, on small survey plots, when the wildlife

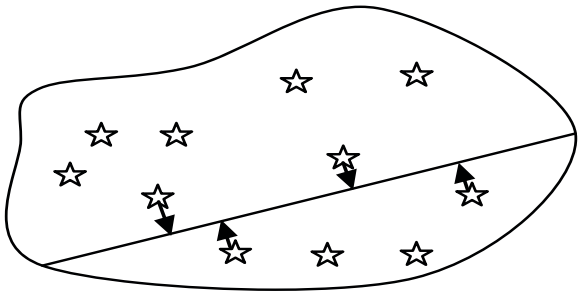


Figure 2.1: Line transect sampling approach with a single, randomly placed line. Here four animals (denoted by stars) are detected and their distances from the transect are recorded.

species have a strongly aggregated distribution or when species that are on the line are not easily detected [12].

Animal tracking has gained popularity with the assistance of global positioning system (GPS) and radio frequency identification (RFID) technologies [1, 33]. The GPS-collars and RFID ear-tags can be used for different purposes: animal movement tracking, territory range measurement, population estimation, etc. This recently introduced method is of great help in disease propagation and survival research since it is very difficult to capture the same animal by traditional ways in mark-release-recapture analysis.

In recent years, a new surveying and monitoring technique using remote photographic devices has become more and more popular. The method is efficient and cost-effective for inventories and in some cases estimation of population sizes, and has tremendous advantage over traditional methods when surveying rare and cryptical animals. Camera trapping has been widely used in population studies of tigers [30], bears [41], birds' nests [3, 4], and fishers [29]. Camera traps offer an important non-invasive approach for studying activity patterns and estimating richness of animals throughout space and time. Roberts confirmed that camera trapping is an efficient, rigorous and cost-effective method for long-term monitoring programs [53].

Given the benefits of the camera trapping approach, the Alberta Innovates Technology Futures (AITF) has deployed a significant number of cameras in the province of Alberta (see Figure 2.2) in order to estimate populations of different mammal

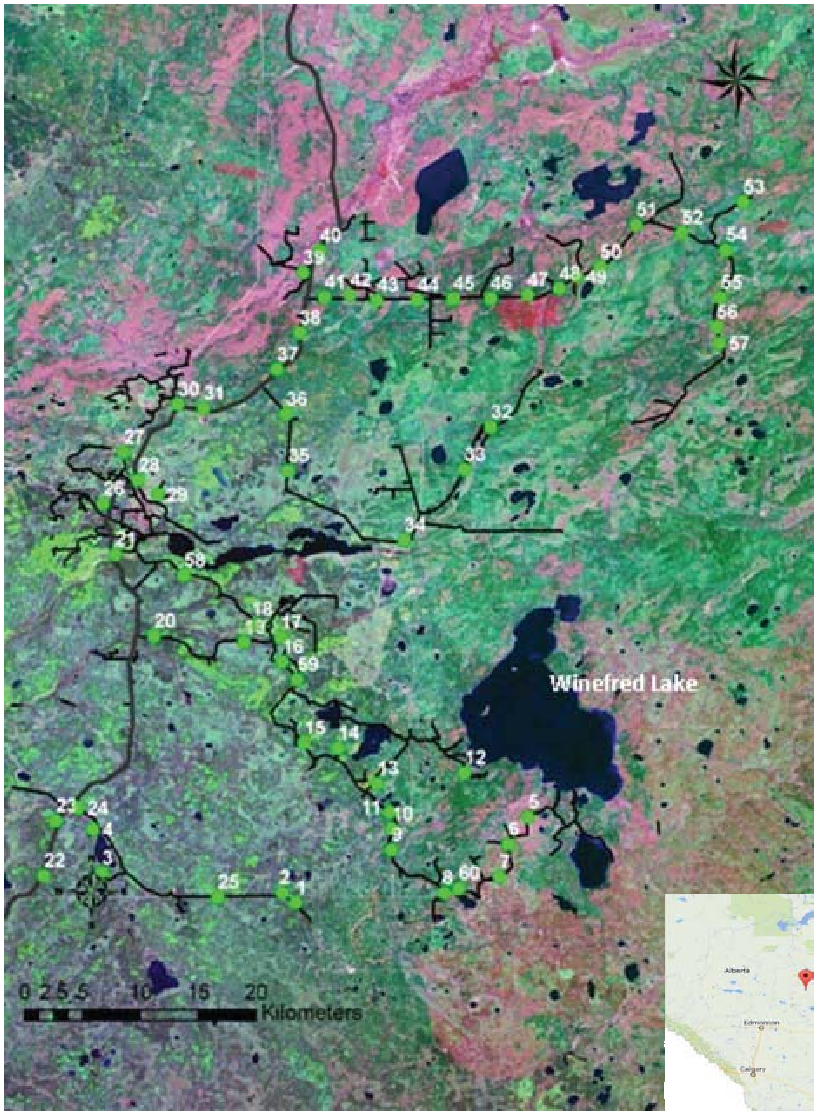


Figure 2.2: An example shows the deployment of 60 infra-red remote cameras (Reconyx PC900 Hyperfire) in Alberta's northeast boreal forest and west of Winefred Lake. The picture is provided by AITF. The lower right corner shows the location of Winefred Lake indicated by the red icon in the province of Alberta.

species. The remote camera is motion-triggered, i.e., it takes a picture once the animal movement is detected. Therefore, the camera trap keeps recording the presence of animals day and night. The scientists analyze the image data to make informed decisions on wildlife management and preserving biodiversity.

2.1.2 Computer Vision

Computer vision is a scientific field that includes approaches for acquiring, processing, analyzing, and understanding images through computers or machines. The primary task of computer vision is to obtain 2D or 3D information by processing the acquired images or videos so that computers can "see" and understand the world as humans do, even take a particular action following their understanding of the world. The ultimate goal of computer vision research is to make computers have the ability to understand the world through visual observation. Before the achievement of this long-term goal, researchers must concentrate on building components of a vision system that can complete some automatic tasks with the help of its visual sensitivity and machine learning algorithms. 3D reconstruction [63] and recognition [23], object categorization [35, 64, 61, 48], tracking [7], etc., are some popular examples of computer vision tasks.

Modern computer vision research originated in the early 1960s, and the earliest visual applications were pattern recognition systems for character recognition in office automation tasks. Among all the visual tasks we desire a computer to perform, analyzing a scene and recognizing all of the constituent objects remains the most challenging [58]. The visual recognition problem is very difficult because the real world around us consists of a mixture of objects, which all occlude one another, appear in different poses, and have intra-class variability due to variations in shape, appearance, and complex articulation. Over the years, significant research efforts have focused on the visual recognition problem and notable progress has been shown.

The visual recognition problem can be discussed along two axes: object detection and object categorization. Object detection is known as a fundamental element in computer vision research for further understanding of image content, recognition

of the containing objects, etc. The problem of object detection can be generally classified into two groups: moving object detection and static object detection[1]. The former, which is widely involved in video surveillance, automatic driving, assistance of the disabled and so on, is usually done by background subtraction techniques. The latter, which refers to detecting instances of objects for a given class and is widely applied to Content-based Image Retrieval (CBIR), is performed by sliding windows based methods, segmentation based methods, etc. Within the object categorization problem, there are two main classes of applications: object instance recognition and object class classification. The former refers to the problem of re-recognizing a known object (e.g., Tom's mug or Henry's mug), which can be viewed from novel viewpoints, partially occluded, or against a highly cluttered background. The most well-known application of object instance recognition is location or landmark matching. The latter, which is also known as generic object categorization, deals with the recognition of categories of objects such as *Car*, *Cat*, *Airplane*, etc., and are widely applied to problems such as human action recognition, scene classification, and object categorization. Our case is considered to fall into the area of generic object categorization.

2.1.3 Machine Learning

Machine learning is the field of building systems that can analyze and learn from data. Such knowledge can be further applied to problems of classification, regression, clustering and density estimation. For example, a machine learning system that is trained on email messages can apply the learned knowledge to determine whether an incoming email is spam or not.

The machine learning algorithms are generally classified as either *supervised* or *unsupervised* learning methods [8]. Supervised learning methods use a set of labeled training examples, each with a feature vector and a class label which is predefined while unsupervised learning methods induce knowledge from data without any corresponding target values. Cases such as the character recognition example

[1]The static object detection means detecting an object from one single image, and thus the object is considered to be static.

in which the goal is to assign each input vector to one of a finite number of discrete categories are known as classification problems. In contrast, the problems whose output is made of one or more continuous values are called regression. Both classification and regression use label information from data and thus are considered as supervised learning problems. The goal of unsupervised learning problems might be either to discover groups of similar samples within the data, which is called clustering, or to determine the estimation of data distribution within the input space, which is known as density estimation.

Modern computer vision tasks such as tracking and visual recognition largely reply on the employment of machine learning algorithms to achieve satisfactory performance. Generative models like Bayesian approaches and discriminative models like SVMs methods are widely employed in visual recognition tasks [37, 56, 35, 64, 61, 68].

2.2 Related Work

The needs of storing large amounts of information and finding useful information from such collections are satisfied with the advent of modern computers. The current text retrieval systems generally have some standard steps [2]. First and foremost, the documents are parsed into a set of words, which are represented by their canonical form (e.g., speak, speaks, speaking, and spoke are represented by their canonical form speak). A unique identifier is then assigned to each of the words, and each document is represented by a vector with components given by the frequency of occurrence of the words the document contains. Therefore, a text is found by computing its vector of word frequencies and returning the documents with the closest vectors. The Bag of Words (BoW) model, which works in the above manner, is one of the most popular and successful approaches among all of models built for text retrieval.

The above idea achieves a great success in the information retrieval domain and is borrowed to solve visual object categorization problems. Computer vision researchers started to draw an analogy between image categorization and text re-

trieval in the early 2000s. In this manner, a collection of images (corresponding to the corpus) is parsed into a set of visual words (corresponding to textual words) based on image regions or patches. Then each image (corresponding to the document) is represented by the extracted local descriptors which all have their own identifiers. Once we encode each image, i.e., each image now is represented by a vector with elements indicating its constituent visual words, learning and classifying algorithms can be applied to do visual categorization. The representation model described above is named Bag of Visual Words (BoVW) model in the computer vision field.

Sivic and Zisserman [56] and Csurka et al. [15] first introduced the BoVW model for visual object categorization. In their work, images were scanned for salient regions and a high-dimensional descriptor is computed for each region. These descriptors are then quantized or clustered into a codebook of visual words, and each salient region is mapped to the visual word closest to it under this clustering. An image is then represented as a bag of visual words, and these are entered into an index for later querying and retrieval. Since its success by mimicking simple text-retrieval systems using the analogy of "visual words", a significant number of approaches based on the BoVW model have been developed.

Li and Perona [37] proposed to adapt the Latent Dirichlet Allocation [9] model to learn and recognize natural scene categories. Images of scenes are represented by a collection of local patches, or codewords obtained by unsupervised learning algorithms. Different from previous approaches where codewords were learned from hand annotations which were tedious and expensive, their approach learns the codewords distribution without supervision. The goal of learning is to find a model that best represents the distribution of these visual words in each category of scenes. In the recognition step, they first identify all the visual words in a given image, and then find the category model that best fits the distribution of the visual words of the particular image.

Grauman and Darrell [22] employed a variant of BoVW based approach to visual object recognition. In their work, each image is represented by a set of unordered local features, and all sets are embedded into a space where they are clus-

tered according to their feature correspondences. The authors first extract Scale Invariant Feature Transform (SIFT) features from the image set and then measure the affinity between any two images using a pyramid-based metric. Then images are clustered based on this affinity by a spectral clustering method which could determine classes from the image set. The potential outliers are removed to further refine each cluster.

Another variant of the BoVW based approach was proposed by Nistér and Stewénius [45], where the authors presented a recognition scheme that scaled efficiently to a large number of objects. The provided live demonstration proves the efficiency and quality of their system which could recognize CD-covers using a databased of 40,000 images. The proposed scheme constructs on the popular techniques of indexing descriptors which are hierarchically quantized in a vocabulary tree and is robust to background clutter and occlusion. The most significant property is that the quantization is defined by the tree and thus leads to efficient and effective image retrieval performance.

The various approaches discussed above are all built upon the idea that each image can be represented by a set of orderless visual words. The inherent property of the BoVW model from text retrieval is that the model does not consider the order of words. When it comes to the visual object categorization problem, the BoVW model completely loses spatial information between salient image regions from which the descriptors are extracted. Considering the fact that all parts of the salient regions are placed in a specific manner, significant research efforts are concentrated on incorporating spatial information instead of using orderless descriptors.

Lazebnik et al. [35] introduced another BoVW model approach that was inspired by [21]. The authors described a holistic approach for recognizing scene categories based on approximating global geometric correspondence. The proposed scheme partitions the image into increasingly fine sub-regions and compute histograms of local features found inside each sub-region. The applied spatial pyramid extends the orderless BoVW image representation and incorporates spatial relationships. By using global cues as indirect evidence about the presence of an object, the approach consistently achieves an improvement over an orderless image repre-

sentation.

Yang et al. [64] extended the nonlinear Support Vector Machines (SVMs) which employs a spatial pyramid matching kernel strategy [35]. The authors argued that the nonlinear SVMs have a complexity $\mathcal{O}(n^2 \sim n^3)$ in training and $\mathcal{O}(n)$ in testing, where n is the size of training set, implying that it is nontrivial to scale up the algorithms to deal with very large datasets. The extended scheme generalizes the vector quantization to sparse coding followed by spatial max pooling and employs a linear spatial pyramid matching kernel based on encoded SIFT descriptors. The new approach scales linearly in training and maintains a constant cost in testing. Moreover, the presented linear spatial pyramid matching based on sparse coding of SIFT descriptors always outperforms the nonlinear kernels.

Traditional spatial pyramid matching is further extended to employ novel descriptors mapping algorithms. Wang et al. [61] replaced the vector quantization coding [35] with Locality-constrained Linear Coding (LLC) which was argued to be simple but effective. LLC utilizes the locality constraints to map each descriptor into its local-coordinate system, and then the mapped code are integrated by max pooling to generate the final representation. Zhou et al. [68] proposed to use the nonlinear supervector coding in place of the vector quantization coding. Both approaches employ a linear SVMs classifier to learn and classify image data.

The BoVW model based approaches discussed above have been dominant in image classification systems. Note that all of those approaches but [37, 68] work in a discriminative manner, which means they do not model the statistical distribution of image patches. An alternative was proposed by Perronnin and Dance [48] which replaces the vector quantization with the Fisher Vector coding using the Fisher Kernel framework [26]. Fisher Kernel is a powerful framework that combines the advantages of both generative and discriminative approaches. The fundamental idea is to represent a signal with a gradient vector derived from a generative probability model and to subsequently forward this representation to a discriminative classifier. In their paper, the authors applied a Gaussian Mixture Model to approximate the distribution of low-level features and thus generate the visual codebook. The Fisher Vector coding utilizes the zero-order, first-order and second-order image statistics

and thus form a dense image approximation. In contrast, the BoVW model based approaches only use the zero-order image statistics and form a sparse global representation of an image. By utilizing more information from images, the Fisher Vector coding approach achieves better performance over the BoVW based approaches.

Perronnin et al. [50] further improved the Fisher Vector coding method [48] by introducing the l_2 normalization which cancels the image-specific information and power normalization which reduces the effect of the sparser Fisher Vector. Perronnin et al. [49] compressed the high dimensional dense Fisher Vector approximation for large scale image classification problems. The Fisher Kernel framework has been applied to face verification [55], person re-identification [40], and video retrieval [44].

Jégou et al. [27] proposed the Vector of Locally Aggregated Descriptors (VLAD) to represent images for classification and searching tasks. In their paper, image regions are extracted using an affine invariant detector and then described using the SIFT descriptor. Each descriptor is then assigned to the closest cluster of a codebook of size *K*. The vector differences between descriptors and cluster centers are accumulated and normalized and then concatenated into a single vector representation. This way of building VLAD captures the distribution of local descriptors and thus is similar to Fisher Vectors [48].

Most of camera trapping based studies use unique coat patterns (e.g., spots or stripes) to identify individual animals of selected species. Bolger et al. [11] used computer software to identify individual animals based on coat patterns for the analysis of mark recapture technique. They matched individual animals by using SIFT keypoints. The work most similar to ours is [66] which used the improved sparse coding spatial pyramid matching on top of the encoded dense SIFT and the cell-structured Local Binary Pattern (cLBP) descriptors. However, they started with images that were manually cropped out of the background, which actually turns the animal species identification task into other problems like individual animal identification.

In the literature of visual object classification, one of the most popular and successful schemes is the pipeline of *low-level feature extraction - local descriptors*

encoding - spatial pooling - classification. [35, 64, 68, 61, 48, 50, 49] are all the notable examples that apply spatial pooling on the top of the encoded local image descriptors, plus a nonlinear SVMs classifier using histogram intersection or Chi-square kernels or a linear SVMs classifier. Following this pipeline, we first propose to use the Fisher Vector patch coding method on top of the dense SIFT and the cLBP descriptors and a linear SVMs classifier to identify animal species based on camera trapping data.

Chapter 3

Feature Extraction

Various image descriptors can be applied to localize the object of interest, match a known object, and recognize an unknown object. Among all of the existing descriptors, such as the Harris Corner feature [25], shape descriptor [6], affine-invariant detector and descriptor [43] and so forth, the Scale Invariant Feature Transform (SIFT) [39] descriptor is empirically demonstrated to be the most effective and appropriate for modern image classification tasks [35, 64, 61, 68, 48]. In our work, animal images are described by the SIFT descriptors. In addition, sparked by [62] and suggested by [66], the cell-structured Local Binary Pattern (cLBP) [46] descriptor is combined with the SIFT feature to generate much more robust image description, which is shown in the experimental results.

3.1 Scale Invariant Feature Transform

The SIFT [39] is an algorithm to detect and describe image local features and has been widely used for object matching [32], motion tracking [67], and image classification [35, 64, 61, 68, 48] tasks. There is a large amount of choices when extracting image features, especially the choice might vary with different vision tasks. A great number of empirical studies show that the SIFT descriptor generally works well when scale, rotation, translation, or illumination change happens. As stated in [39], the SIFT feature allows for an object to be recognized in multiple images taken from different 3D viewpoints within the background. In particular, since the SIFT feature is highly distinctive, a single SIFT descriptor can contribute to correct

matching with high probability. Based on theoretical analysis and experimental study [39, 35, 64, 61, 68, 48], the SIFT descriptor is recognized as the best choice for object recognition and scene classification. We argue that the SIFT descriptor would be the best choice for the animal species identification task.

The SIFT method takes an image as input and transform it into a large collection of local descriptor vectors. As mentioned above, each of these feature vectors is invariant to scaling, rotation, translation and partially invariant to illumination change and cluttered background. To obtain these local feature vectors, the SIFT method performs in a manner of four cascade filtering approaches.

Scale-Space Extrema Detection

The first step attempts to search over all scales and locations to identify potential interest points that are identifiable from multiple viewpoints and scales. The property of scale-invariant is achieved by first transforming the original image into the scale space using a Gaussian kernel. For the consideration of computation efficiency, the Difference of Gaussian (DoG) function is applied to locate those interest points.

Keypoint Localization

The potential interest points retrieved in the first stage are sensitive to noise and a significant portion of them lie along edges. Thereby, the second step attempts to eliminate points that have low contrast or poorly localized along the edge. A detailed Laplacian model is fit to determine the location and scale at each candidate location to eliminate these false points. Then real keypoints are retained by measuring their stability.

Orientation Assignment

To achieve the rotation-invariant property, the third stage attempts to assign one or more orientations to each keypoint location according to local image gradient directions. To improve the robustness, the directions above 80% of the principal direction are kept as the final orientations. All the future computation is operated on the image which has been transformed relative to the scaling, location and assigned

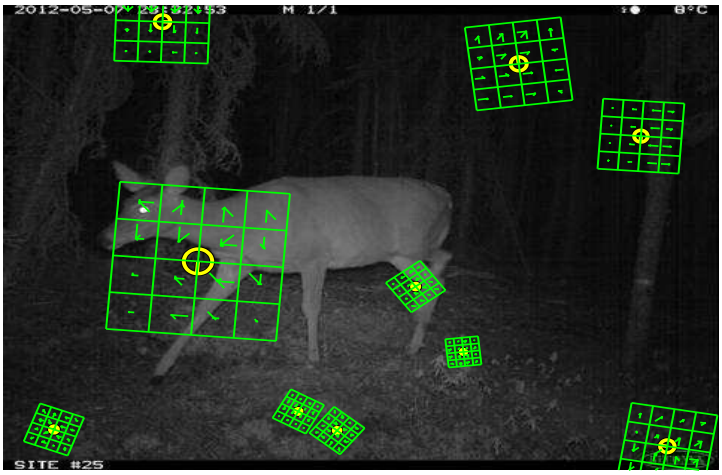


Figure 3.1: An example shows ten detected and localized keypoints (denoted by yellow circles) and ten descriptors (denoted by green grids with local gradients at the selected scale inside each grid) around each keypoint.

orientations for each feature vector. Therefore, the SIFT feature is guaranteed to be invariant to these transformations.

Keypoint Descriptor

The last stage generates the keypoint descriptors by measuring local image gradients at the selected scale in the region around each keypoint. Keypoint descriptors generated in this manner allow for significant degree of local shape distortion and illumination changes. Typically, keypoint descriptors are represented by a set of 16 histograms, aligned in a 4×4 grid, each with 8 orientation bins. Therefore, the resulting feature vector contains 128 entries.

Based on the four steps above, Figure 3.1 shows ten detected and localized keypoints and ten descriptors with local gradients at the selected scale inside each green grid.

The approach transforms the image data into scale-invariant coordinates relative to local descriptors. The resulting feature vectors are widely applied to image matching and visual tasks based on image matching. Generally, there are two types of SIFT features applied to visual classification tasks. The sparse SIFT descriptors,

generated by first locating interest points and then extracting features around these interest points, are empirically shown more suitable for object matching tasks since the features are highly distinctive and thus allow a single feature to be correctly matched with high probability against a large amount of features. In contrast, the dense SIFT descriptors, which are extracted from every overlapped image patch in the image, are proven more appropriate for image categorization tasks. Therefore, the dense SIFT descriptors are employed in our work. Figure 3.2 shows how to extract the dense SIFT descriptors on a *Wolf* image. We extract the SIFT features from each 16 $\times$ 16 grid with the spacing size of 6 pixels. All the descriptors are then concatenated together to approximate the image statistics.

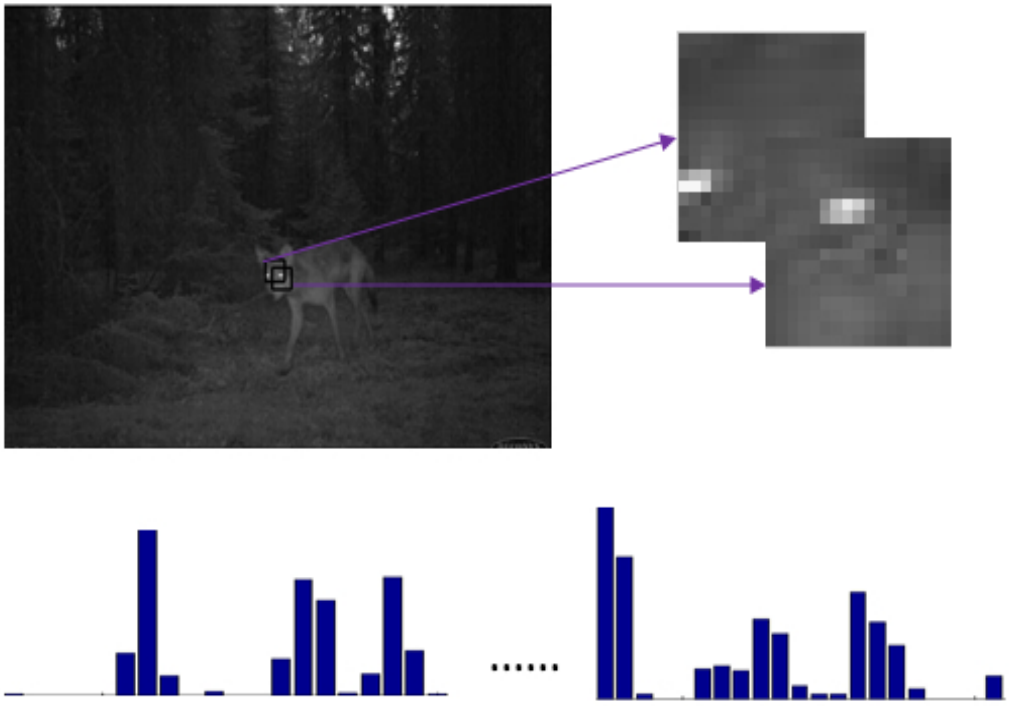


Figure 3.2: The procedure of extracting the dense SIFT features. The image is first divided into 16 $\times$ 16 grids with the spacing size of 6 pixels. The first row shows two overlapped patches at the part of *Wolf* head. The second row is the SIFT descriptor of the upper-left patch with 128 dimensions, and each bar represents the local gradients.

3.2 Cell-structured Local Binary Pattern

The Local Binary Pattern (LBP) [46] is a local texture descriptor that retrieves the appearance of an image in a small neighbourhood around a pixel. It is based on the assumption that texture has two locally complementary aspects, a pattern and its strength. The LBP descriptor is a vector of integer labels, and each of these integer labels represents the pixel in the neighbourhood. Each integer label is either 0 or 1, depending on whether the intensity of the corresponding neighbouring pixel is smaller than the intensity of the central pixel or not. The binary vectors are usually quantized and pooled in local histograms and thus used for texture classification.

Among many variants of LBP descriptors since it was first introduced by [46], the uniform LBP pattern [47] is mostly used because it is tolerant to image rotation which is the desired property for many tasks. The uniform $\mathrm{LBP}^{u}_{p,r}$ descriptors are achieved by circularly sampling around the center pixel, where p refers to the number of neighbouring pixels involved, r represents the sampling radius, and u measures the uniformity of a particular pattern. Figure 3.3 shows three examples of circular LBP patterns. The uniformity measurement is the number of bitwise transitions from 1 to 0 or vice versa when the descriptor is sampled from a circular neighbourhood. If u is set to 2, then a local binary pattern is considered uniform if its uniformity measurement is at most 2. For example, for operator $\mathrm{LBP}^{2}_{8,1}$, the patterns 00000000 (0 transitions), 00011100 (2 transitions), and 11110011 (2 transitions) are uniform while the pattern 11110100 (3 transitions), and 10101010 (7 transitions) are not.

By mapping the uniform LBP descriptors, there is an individual integer label for each uniform pattern and a single integer label is assigned to all non-uniform patterns. In this way, the number of total different labels for mapping LBP patterns around p neighbouring pixels is $p*(p\text{-}1) + 3$. For example, for operator $\mathrm{LBP}^{2}_{8,1}$, the resulting mapping generates 59 output labels for neighbourhood of 8 sampling points.

Further, the cell-structured LBP based on uniformity measurement in [62] was proven to be more effective than the original LBP [46] and the uniform LBP [47] to

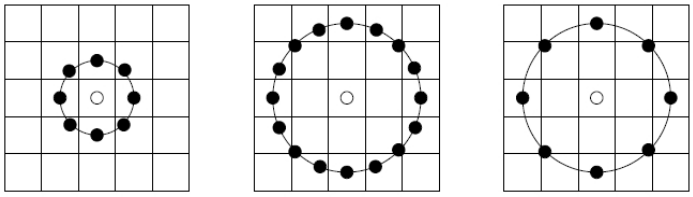


Figure 3.3: The circular LBP examples: $\text{LBP}_{8,1}$, $\text{LBP}_{16,2}$, and $\text{LBP}_{8,2}$. The pixel values are bilinearly interpolated whenever the sampling point is not in the center of a pixel.

describe the local texture of an image. In [62], the input image is first divided into non-overlapping cells with the size 16×16. The uniform LBP patterns extracted from these cells are then concatenated into a cLBP vector, whereas the dense SIFT is extracted from 16×16 overlapping patches with sampling grid size of 6 pixels. However, in our work, we extract the cLBP features from overlapped patches, following the same strategy as extracting the dense SIFT features, since we found that the cLBP descriptors extracted from overlapped patches lead to better performance than those from non-overlapping patches. By convention, we still call this feature as cLBP. Figure 3.4 shows the procedure of extracting the cLBP descriptor from an image and one uniform LBP pattern from the bear ear patch.

We will examine the performance of the SIFT descriptors and the cLBP descriptors on our animal image dataset separately. Also, the performance of the combination of the SIFT features and the cLBP descriptors is examined.

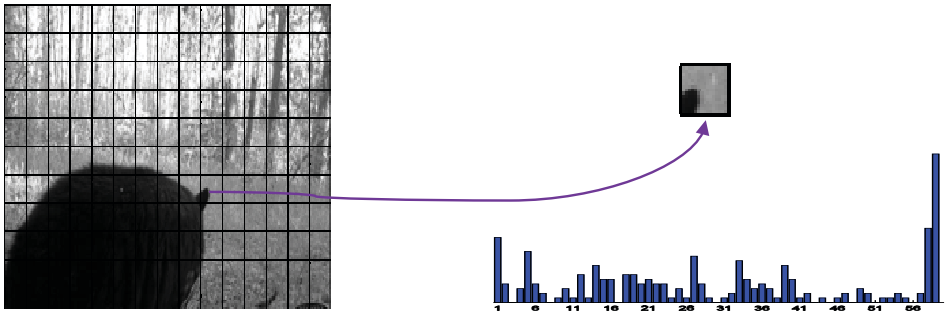


Figure 3.4: In the procedure of extracting the cLBP descriptors, the image is first divided into 16×16 patches from which the uniform LBP descriptors are extracted using $\mathrm{LBP}^2_{8,1}$ and then concatenated into a cLBP vector. The bar table at the lower-right position shows the uniform LBP feature for the *Bear* ear patch shown at the upper right position. The value under the bar varies from 1 to 59 from left to right, and each bar represents a particular pattern.

Chapter 4

Fisher Vector Coding Approach

The pipeline of *low-level feature extraction - local descriptors coding - spatial pooling - classification* has been dominant in image classification methods since the early 2000s. The image classification approaches [35, 64, 62, 68] based on this pipeline have achieved state-of-the-art performance on both small-scale and large-scale image datasets [36, 18, 16]. Therefore, the method we use in this chapter follows this pipeline.

We extract the dense Scale Invariant Feature Transform (SIFT) features and the cell-structured Local Binary Pattern (cLBP) descriptors at the first stage. Next, the visual codebook is constructed using a generative model - Gaussian Mixture Model on top of the dense SIFT features and the cLBP descriptors, respectively. Then, all image descriptors are encoded into a fixed length vector representation using the Fisher Vector coding method [48, 50, 49]. The spatial pyramid strategy [35] is used to incorporate rough geometry information among image regions and the encoded features are pooled over the neighbour regions to obtain the representation of the image. Last, a classifier is built on top of the image vector representations. This chapter describes the construction of the visual codebook using a Gaussian Mixture Model, the Fisher Vector coding method, a spatial pyramid strategy, an average pooling, and the classification method.

4.1 Gaussian Mixture Model Clustering

A Gaussian Mixture Model $p(X \mid \lambda)$ is a parametric probability density function on $\mathcal{R}^D$ represented as a weighted sum of *K* Gaussian component densities:

$$p(X \mid \lambda) = \sum_{k=1}^{K} p(X \mid \mu_k, \Sigma_k)\omega_k, \tag{4.1}$$

where *X* is a *D*-dimensional continuous-valued data (e.g., feature vector), ω_k is the prior probability value or the mixture weight and subjected to $\sum_{k=1}^{K} \omega_k = 1$, and $p(X \mid \mu_k, \Sigma_k)$ is the Gaussian component density. Each component density is a *D*-variate Gaussian function of the following form:

$$p(X \mid \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{D/2} \mid \Sigma_k \mid^{1/2}} e^{-\frac{1}{2}(X-\mu_k)'\Sigma_k^{-1}(X-\mu_k)} \tag{4.2}$$

where the means vector $\mu_k \in \mathcal{R}^D$ and the positive definite covariance matrices $\Sigma_k \in \mathcal{R}^{D\times D}$ of each Gaussian component.

The complete Gaussian Mixture Model is then parameterized by the mixture weights, the means vector and the covariance matrices from all component densities. The parameters are represented by the notation $\lambda = \{\omega_1, \mu_1, \Sigma_1, \omega_2, \mu_2, \Sigma_2, \cdots, \omega_K, \mu_K, \Sigma_K\}$. Note that here the covariance matrices are assumed to be diagonal for computational efficiency and thus the Gaussian Mixture Model is fully specified by $(2 \times D + 1)K$ scalar parameters [48]. All the parameters of Gaussian Mixture Model are learned by the Expectation Maximization (EM) algorithm from the training set of descriptors $x_1, x_2, \cdots, x_N$. Then the Gaussian Mixture Model characterizes the soft data-to-cluster assignments in the following form after all parameters are learned:

$$q_{ki} = \frac{p(X \mid \mu_k, \Sigma_k)\omega_k}{\sum_{j=1}^{K} p(X \mid \mu_j, \Sigma_j)\omega_j}, \quad k = 1, 2, \cdots, K \tag{4.3}$$

In this way, the visual codebook is considered as the generative model - Gaussian Mixture Model. That is, the visual codebook is specified by mixture weights, the means vector, and the covariance matrices. After the visual codebook is constructed from the training set of descriptors, we can encode the image descriptors by these parameters.

4.2 Fisher Kernel Basics and Fisher Vector

Unlike traditional approaches based on the Bag of Visual Words (BoVW) model that do not model the distribution of image patches and only work in a discriminative manner, Fisher Vector utilizes the powerful Fisher Kernel framework which has advantages of both generative and discriminative approaches. Instead of using *k*-means clustering algorithm to generate the visual codebook [35, 64, 61], the Fisher Kernel framework uses a Gaussian Mixture Model to approximate the underlying distribution of low-level image descriptors and thus generate the visual codebook. Then the image can be characterized by a gradient vector obtained from the generative probability model, i.e., the visual codebook. The image approximation learned from the Fisher Kernel framework retains the zero-order, first-order and second-order image statistics, i.e., word frequency, means vector and standard deviation matrices. More details about Fisher Kernel and Fisher Vector are presented in the following paragraphs.

Let $X = \{x_i, i = 1, 2, \cdots, N\}$ represents an example of *N* observations, and let $p(X \mid \lambda)$ denotes a probability density function ($p.d.f.$) whose parameters are indicated by λ, where $\lambda = \{\omega_1, \mu_1, \Sigma_1, \omega_2, \mu_2, \Sigma_2, \cdots, \omega_K, \mu_K, \Sigma_K\}$ represents the vector of parameters of $p(X \mid \lambda)$. The *p.d.f.* models the generative process of elements in *X*.

Now the score function is defined as the gradient of the log-likelihood of the image data on the learned generative model:

$$G_\lambda^X = \nabla_\lambda \log p(X \mid \lambda) \tag{4.4}$$

The gradient Equation 4.4 characterizes the contribution of each single parameter to the generative procedure. More clearly, it describes how the parameters of the generative model $p(X \mid \lambda)$ should be adjusted to more appropriately fit the data *X*. Given the fact that $G_\lambda^X \in \mathcal{R}^K$, we know the dimensionality of G_λ^X only varies with the number of parameters *K* in λ and not with the example size *N*. Therefore, it transforms a sample *X* with a variable length into a fixed length vector whose size is only determined by the number of parameters *K* in the model.

The Fisher Information Matrix is suggested to normalize the inner product term

of two gradient vectors [26]. The Fisher Information Matrix is formulated as follows:

$$F_\lambda = E_X[\nabla_\lambda log\ p(X \mid \lambda)\ \ \nabla_\lambda \log p(X \mid \lambda)'] \tag{4.5}$$

or $F_\lambda = E_X[G_\lambda^X G_\lambda^{X'}]$, where $F_\lambda \in \mathcal{R}^{K \times K}$.

Then the similarity between two samples *X* and *Y* can be measured using Fisher Kernel as follows [26]:

$$K_{FK}(X, Y) = G_\lambda^{X'} F_\lambda^{-1} G_\lambda^Y \tag{4.6}$$

Because both of F_λ and its inverse are positive semi-definite, F_λ^{-1} can be decomposed as $L_\lambda^{'} L_\lambda = F_\lambda^{-\frac{1}{2}} F_\lambda^{-\frac{1}{2}}$, the Equation (4.6) can be reformulated as a dot-product:

$$K_{FK} = \mathcal{G}_\lambda^{X'}\ \mathcal{G}_\lambda^Y \tag{4.7}$$

where

$$\mathcal{G}_\lambda^X = L_\lambda\ G_\lambda^X = L_\lambda \nabla_\lambda \log p(X \mid \lambda) \tag{4.8}$$

Thus the normalized version of gradient vector approximation of *X* is given by Equation (4.8). More clearly, Equation (4.8) is defined as the Fisher Vector of an image *X*. Obviously, the dimensionality of the normalized $\mathcal{G}_\lambda^X$ is thus the same as that of gradient vector G_λ^X. Therefore, a non-linear kernel machine using kernel function K_{FK} is equivalent to a linear kernel machine using $\mathcal{G}_\lambda^X$ as feature vector. Thus, a linear classifier can be applied following the new gradient vector representation $\mathcal{G}_\lambda^X$.

Based on the assumption that the low-level image descriptors in *X* are independent, Equation 4.8 can be reformulated as follows:

$$\mathcal{G}_\lambda^X = \sum_{i=1}^{N} L_\lambda \nabla_\lambda \log p(x_i \mid \lambda) \tag{4.9}$$

Now the Fisher Vector of an image can be considered as the sum of the normalized gradient statistics $L_\lambda \nabla_\lambda \log p(x_i \mid \lambda)$ calculated for each low-level image descriptor. The following transformation

$$x_i \rightarrow \phi_{FK}(x_i) = L_\lambda \nabla_\lambda \log p(x_i \mid \lambda) \tag{4.10}$$

can be thought as embedding the local descriptors into a higher dimensional space which is more beneficial to linear classifiers.

Recall that a *K*-component Gaussian Mixture Model is denoted by $\lambda = \{\omega_1, \mu_1, \Sigma_1, \omega_2, \mu_2, \Sigma_2, \cdots, \omega_K, \mu_K, \Sigma_K\}$, where ω_k is the mixture weight, μ_k is the means vector and Σ_k is the covariance matrix of *k*th Gaussian component. Then:

$$p(x \mid \lambda) = \sum_{k=1}^{K} \omega_k p_k(x \mid \lambda) \tag{4.11}$$

where $p_k(x \mid \lambda) = \frac{1}{(2\pi)^{D/2}|\Sigma_k|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)'\Sigma_k^{-1}(x-\mu_k)}$ and the requirements are $\omega_k \geq 0, \ \forall k$ and $\sum_{k=1}^{K} \omega_k = 1$. The Gaussian Mixture Model parameters are estimated on a large training set of local image descriptors using EM algorithm to optimize a maximum likelihood criterion.

Let q_{ki} be the soft assignment of a single descriptor x_i to the *k*th Gaussian component:

$$q_{ki} = \frac{\omega_k p_k(x_i \mid \lambda)}{\sum_{j=1}^{K} \omega_j p_j(x_i \mid \lambda)} \tag{4.12}$$

and given the F_λ is diagonal, the gradients w.r.t. weights, mean vector and covariance matrices can be defined as follows:

$$\mathcal{G}_{\omega,k}^{X} = \frac{1}{N\sqrt{\omega_k}} \sum_{i=1}^{N} (q_{ki} - \omega_k) \tag{4.13}$$

$$\mathcal{G}_{\mu,k}^{X} = \frac{1}{N\sqrt{\omega_k}} \sum_{i=1}^{N} q_{ki} \left(\frac{x_i - \mu_k}{\sigma_k}\right) \tag{4.14}$$

$$\mathcal{G}_{\sigma,k}^{X} = \frac{1}{N\sqrt{2\omega_k}} \sum_{i=1}^{N} q_{ki} \left\{\frac{(x_i - \mu_k)^2}{\sigma_k^2} - 1\right\} \tag{4.15}$$

Note that the above equations are further normalized by the number sample size *N* to eliminate the dependency on the sample size. Then, the final gradient vector or Fisher Vector is the concatenation of $\mathcal{G}_{\omega,k}^{X}, \mathcal{G}_{\mu,k}^{X}$ and $\mathcal{G}_{\sigma,k}^{X}$ vectors for $k = 1, 2, \cdots, K$, which are further normalized by the sampling size *N*, and the Fisher Vector is therefore $(2 \times D + 1)K$ dimensional.

The description above is the procedure to compute the Fisher Vector representation of an image from a large set of training descriptors and is the main strategy used in [48] for object and scene classification. Perronnin et al. [50] proposed to

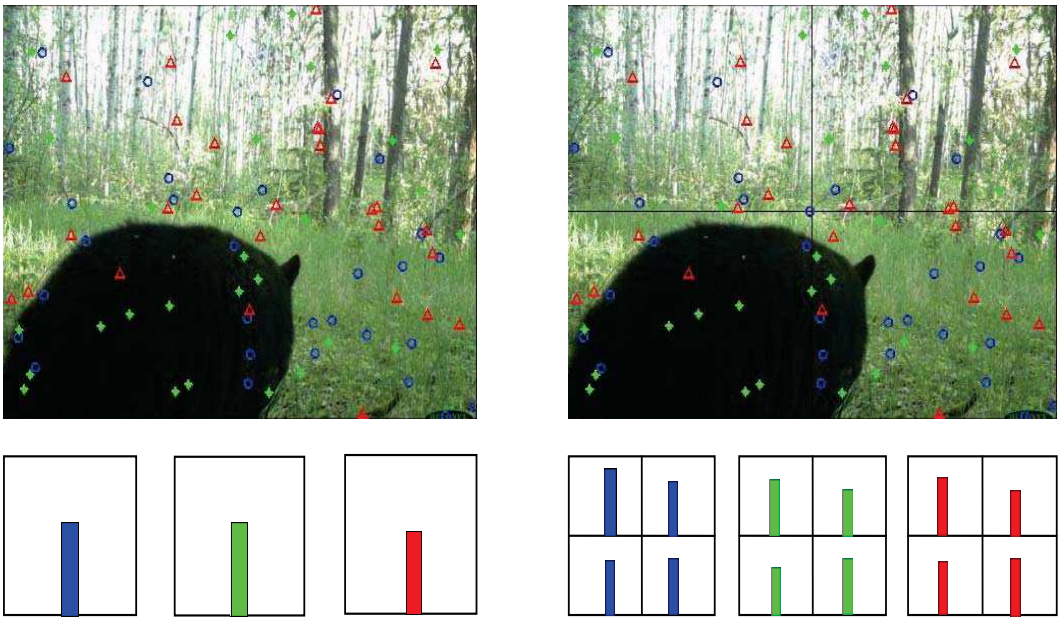


Figure 4.1: *Bear* example of building a two-level spatial pyramid. In the figure, the image has three descriptor types, denoted by blue circles, green stars, and red triangles, respectively. At the top, we subdivide the image into two different levels of resolution. Then, for each resolution level and each channel, we count the features that fall in each spatial bin. Finally, the feature of the *Bear* image is formed by the concatenation of features pooled over different resolution levels.

use two following normalization steps l_2 and power normalization, which have been shown necessary to achieve competitive performance. The introduction of l_2 normalization aims at alleviating the image-specific information as much as possible, which is completed by the following operation $\frac{K(X,Y)}{\sqrt{K(X,Y)K(X,Y)}}$. The power normalization attempts to avoid the fact that the Fisher Vector becomes sparser when the number of Gaussian components increases. It was found in [49] that the distribution of features are more peaky around zero as the number of Gaussian components increases. This is achieved by applying the operation $f(z) = sign(z) \mid z \mid^{\alpha}$ to each dimension of the Fisher Vector, where $0 \leq \alpha \leq 1$ is the normalization parameter. Usually, α is set to 0.5.

4.3 Spatial Pyramid Strategy

Image representation for modern image classification tasks is based on the assumption that the high-level semantic description of the image can be somehow inferred

from low-level image features. The BoVW model works exactly in this way. However, the BoVW model does not consider the spatial information among image regions and thus the structure knowledge of the image is discarded. In order to overcome the underlying weakness, Lazebnik et al. [35] introduced the spatial pyramid strategy to incorporate the rough geometry information of the image. The spatial pyramid strategy partitions the input image into increasing fine sub-regions and then calculates histograms of local features retrieved from each sub-region by pooling image descriptors over neighbour sub-regions. Therefore, the incorporation of rough geometry information extends the orderless representation of the BoVW model. The spatial pyramid strategy is proven to be effective for scene classification [35] and generic object categorization [64, 61].

Figure 4.1 shows an example of how to build a two-level spatial pyramid. We first subdivide the original image into 2×2 sub-regions and then compute a feature histogram from each sub-region. The final image approximation is represented by the concatenation of the pooled features over the sub-regions using the spatial pooling method described in the next section.

4.4 Average Pooling

The technique of pooling the encoded features over neighbour regions has been considered as a key step to continuously improve the understanding of the underlying image content and thus the performance of image classification. Spatial pooling aggregates the variable number of encoded features over the neighbour regions into a fixed length single vector. The global image approximation is then represented by the concatenation of those fixed length vectors over multiple overlapping regions in the image.

Average pooling is one of the most popular spatial pooling methods integrated in the modern image classification framework. Average pooling calculates the average response of each component. Given the encoded feature vector $\mathbf{U}$ where each column corresponds to the responses of all the local descriptors to one specific

codeword in the visual codebook **B**, average pooling is defined as follows:

$$h_j = \frac{1}{N}\sum_{i=1}^{N} u_{ij}, j = 1, 2, \ldots, K. \tag{4.16}$$

where N denotes the number of local descriptors in the region, and h_j is the average response of all local descriptors to a specific codeword.

As a summary, the Fisher Vector computation is shown in Algorithm 1. In the algorithm, spatial pyramid strategy is used.

4.5 Support Vector Machines

Support vector machines, originally developed by Vladimir Vapnik in 1963 and used again extensively in the middle of 1990s, are a set of supervised learning algorithms which are used for data analysis based on statistical learning theory [59]. Unlike traditional approaches (e.g. Neural Networks), which minimize the empirical error on training data, Support Vector Machines (SVMs) are designed to minimize the upper bound of generalization error via maximizing the margin between the separating hyperplane and the data. Under the principle of structural risk minimization, SVMs can generalize well even in high dimensional space while having a small number of training samples. They have been recognized to be superior to traditional empirical risk minimization principle enjoyed by most of artificial neural networks, and have shown good performance in an extensive range of applications, such as text classification [28], speech recognition [20], face recognition [24] and image categorization [35, 64, 61, 48, 49].

The essential fact of SVMs is to find a hyperplane, which can separate the negative and the positive samples with the largest margin achieved by minimizing the Vapnik-Chervonenkis dimension of SVMs. In a binary classification problem, given a set of training examples $\{(x_i, y_i), i = 1, 2, \cdots, N\}$, where *N* is the number of data points and each example has *D* inputs ($x_i \in R^D$), and a class label with one of two values ($y_i \in$ {-1, +1}). All hyperplanes in R^D are parameterized by a vector (**w**) and a constant (*b*), as in the equation

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \tag{4.17}$$

Algorithm 1 Compute Fisher Vector from Image Descriptors Using Spatial Pyramid Strategy

Input:

- Low-level image descriptors (e.g., SIFT and/or cLBP) $X = \{x_i \in \mathcal{R}^D, i = 1, 2, \cdots, N\}$
- Spatial pyramid level L

Output:

- Normalized FV representation $\mathcal{G}_\lambda^X \in \mathcal{R}^{(2D+1)K}$

For each image region $\{R_i, i = 1, \cdots, \sum_{l=0}^{L} 2^{Dl}\}$

- Training set statistics computation
  - For $k = 1, 2, \cdots, K$ initialize the Gaussian Mixture Model parameters
    - $\omega_k \leftarrow 0, \quad \mu_k, \leftarrow mean(X), \quad \Sigma_k \leftarrow var(X)$
  - For $i = 1, 2, \cdots, N$
    - Compute q_{ki} using Equation 4.3
    - For $k = 1, 2, \cdots, K$
      - $\omega_k \leftarrow \frac{1}{N} \sum_{i=1}^{N} q_{ki}$
      - $\mu_k \leftarrow \frac{\sum_{i=1}^{N} q_{ki} x_i}{\sum_{i=1}^{N} q_{ki}}$
      - $\Sigma_k \leftarrow \frac{\sum_{i=1}^{N} q_{ki}(x_i - \mu_k)(x_i - \mu_k)'}{\sum_{i=1}^{N} q_{ki}}$
- Compute the Fisher Vector signature for each image
  - For $k = 1, 2, \cdots, K$:
    - $\mathcal{G}_{\omega,k}^X = \frac{1}{N\sqrt{\omega_k}} \sum_{i=1}^{N} (q_{ki} - \omega_k)$
    - $\mathcal{G}_{\mu,k}^X = \frac{1}{N\sqrt{\omega_k}} \sum_{i=1}^{N} q_{ki} (\frac{x_i - \mu_k}{\sigma_k})$
    - $\mathcal{G}_{\sigma,k}^X = \frac{1}{N\sqrt{2\omega_k}} \sum_{i=1}^{N} q_{ki} \{\frac{(x_i - \mu_k)^2}{\sigma_k^2} - 1\}$
  - Concatenate all FV components into one vector
    $\mathcal{G}_\lambda^X = (\mathcal{G}_{\omega,k}^X, \mathcal{G}_{\mu,k}^X, \mathcal{G}_{\sigma,k}^X, k = 1, 2, \cdots, K)$
- Perform normalization
  - For $i = 1, 2, \cdots, (2 \times D + 1)K$ perform power normalization
    - $[\mathcal{G}_\lambda^X]_i \leftarrow sign([\mathcal{G}_\lambda^X]_i) \mid [\mathcal{G}_\lambda^X]_i \mid^\alpha$
  - Perform l_2-normalization
    $\mathcal{G}_\lambda^X = \frac{\mathcal{G}_\lambda^X}{\sqrt{\mathcal{G}_\lambda^{X\prime} \mathcal{G}_\lambda^X}}$

Pool Fisher Vector over neighbour regions at each resolution level using Equation 4.16 and concatenate all pooled Fisher Vector components into one vector representation

where **w** is the vector orthogonal to the hyperplane.

The classification problem then becomes to find the hyperplane (**w**, b) such that $w \cdot \mathbf{x}_i + b \leq$ -1 for all negative examples and $w \cdot \mathbf{x}_i + b \geq$ +1 for all positive examples. To find the separation hyperplane which has the largest margin, we need solve the following optimization problem:

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{n}\xi_i \tag{4.18}$$

subject to

$$y_i(\mathbf{w}^T \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, i = 1, 2, \ldots, n \tag{4.19}$$

$$\xi_i \geq 0, i = 1, 2, \ldots, n$$

where C is a penalty parameter chosen by the user that controls the trade-off between the margin and the misclassification errors. The larger the C, the higher the penalty to misclassification errors. ξ_i is the non-negative slack variable which measures the misclassification errors. If the data is linearly separable, we can reduce ξ_i to be 0. Otherwise, a non-zero value is assigned to ξ_i. SVMs then give the *generalized separating hyperplane* by minimizing Equation 4.18 under the constraint 4.19.

The optimal hypothesis is then given by Equation 4.20

$$f(\mathbf{x}) = sign(\mathbf{w}^T \cdot \mathbf{x} + b) \tag{4.20}$$

which can be evolved to Equation 4.21 when the data is not linearly separable and kernel tricks are necessary,

$$f(\mathbf{x}) = \sum_{i=1}^{n} y_i\alpha_i K(\mathbf{x}_i, \mathbf{x}) + b \tag{4.21}$$

where $K(\mathbf{x}_i, \mathbf{x}) = (\phi(\mathbf{x}_i), \phi(\mathbf{x}))$ is called kernel function. Notice that the hypothesis may vary on the basis of the linear or kernel trick. In case that the input data to the kernel SVMs lies in a very high dimensional space, the kernel SVMs trick usually scale quadratically or cubically which leads to very expensive computational cost for both training and testing stages.

In this chapter, we present the Gaussian Mixture Model clustering, the Fisher Vector coding approach based on the Fisher Kernel framework, the spatial pyramid strategy, the average pooling, and SVMs. We use Algorithm 2 to summarize how we combine these to build our Animal Species Identification System (ASIS).

Algorithm 2 Build the Animal Species Identification System (ASIS)

Input:

- Training image dataset
- Testing image dataset

Output:

- The trained model
- The species identification result

• Preprocessing

Resize each image down to 180×240 and map each image from colour space to gray scale space.

• Feature Extraction

Extract the dense SIFT features and the cLBP descriptors.

• Fisher Vector Coding

Encode an image using the Fisher Vector coding approach based on Algorithm 1.

• Classify images

Learn a discriminative model and classify new images using a linear SVMs classifier.

Chapter 5

Other Local Descriptor Coding Methods

In light of the modern image classification pipeline, we also examine other patch based descriptor coding approaches, including (1) the Vector Quantization Coding which is used in both of the Bag of Visual Words (BoVW) [15] and the Spatial Pyramid Matching (SPM) [35], (2) the Locality-constrained Linear Coding (LLC) [61], and (3) the Vector of Locally Aggregated Descriptors (VLAD) [27] whose performances are compared to the Fisher Vector representation.

In this chapter, we use the same strategy to extract the dense Scale Invariant Feature Transform (SIFT) features and the cell-structured Local Binary Pattern (cLBP) descriptors as we did for the Fisher Vector coding approach. All approaches but BoVW take the spatial pyramid strategy to incorporate rough geometry information and then use either max pooling (LLC [61]) or averaging pooling (SPM [35] and VLAD [27]) method to aggregate local features over neighbour regions. The feature vector is then forwarded to the Support Vector Machines (SVMs) classifier. Note that for all of these methods the visual codebook is learned using k-means clustering. Therefore, in this chapter, we only present the k-means clustering method, different patch based coding methods, and the max pooling technique.

5.1 *K*-means Clustering

K-means clustering is the most widely used algorithm to generate a visual codebook in image classification system [35, 64, 61]. Given a set of features $\mathbf{X}$ =

$\{x_1, x_2, \cdots, x_N\} \in \mathcal{R}^{D \times N}$ of *N* training features in *D* dimensional space, *k*-means is designed to find *K* vectors $\{\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_K\} \in \mathcal{R}^D$ and a data-to-cluster assignments $\{q_1, q_2, \cdots, q_N\} \in \{1, 2, \cdots, K\}$ such that the cumulative approximation error $\sum_{i=1}^{N} \|\mathbf{x}_i - q_i\|$ is minimized. The standard Lloyd's algorithm [38] that alternates between optimizing the cluster centers ($\mathbf{b}_k = avg\{\mathbf{x}_i : q_i = k\}$) and the data-to-center assignments ($q_i = \arg\min_k \|\mathbf{x}_i - \mathbf{b}_k\|^2$) is applied to construct the visual codebook. In this manner, the visual codebook **B** is constructed.

5.2 Local Descriptor Coding Methods

Let **X** be a set of *D*-dimensional local image descriptors extracted from an image, i.e. **X** = $\{x_1, x_2, \cdots, x_N\} \in \mathcal{R}^{D \times N}$. Given a dictionary with *K* elements, **B** = $\{\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_K\} \in \mathcal{R}^{D \times K}$, different coding frameworks transform each local image descriptor into a *K*-dimensional code to generate the final image representation.

5.2.1 Vector Quantization Coding

The conventional vector quantization coding technique, also known as One-of-N coding method, encodes a local image descriptor into a binary representation, where 1 indicates the corresponding visual word for the local descriptor. It is easy to understand that this representation has a large amount of 0's and only a single 1 and thus it is considered to be very sparse.

The vector quantization method aims to solve the following constrained least square fitting problem:

$$\begin{aligned} &\arg\min_{C} \sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{B}\mathbf{c}_i\|^2 \\ &s.t. \ \ \|\mathbf{c}_i\|_{\ell_0} = 1, \|\mathbf{c}_i\|_{\ell_1} = 1, \mathbf{c}_i \succeq 0, \end{aligned} \tag{5.1}$$

where **C** = $[\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_N]$ is the set of codes for **X**. The cardinality constraint $\|\mathbf{c}_i\|_{\ell_0} = 1$ means that there is only one non-zero element in each code $\mathbf{c}_i$, corresponding to the quantization id of $\mathbf{x}_i$. The non-negative, ℓ_1 constraint $\|\mathbf{c}_i\|_{\ell_1} = 1, \mathbf{c}_i \geq 0$ means that the coding weight for x is 1. In practice, the single non-zero element is found by searching the nearest neighbour.

The vector quantization coding method is used in BoVW [15] and SPM [35]. The main difference between BoVW and SPM is that BoVW uses the orderless codewords while SPM incorporates the rough geometry information. The SPM algorithm partitions the input image into increasingly finer spatial rub-regions, and computes histograms of local features for each sub-region. Typically, in their setting, $2l \times 2l$ sub-regions, $l = 0, 1, 2$ are used. An example of two-level pyramid building is shown in Figure 4.1. By placing a sequence of increasingly finer grids in the feature space, pyramid matching then takes a weighted sum of the number of matches that occur at each level of the pyramid resolution. Two feature vectors are considered to match if they are in the same grid cell. Matches retrieved at coarser resolutions are weighed less than matches retrieved at finer resolutions. The matches are given by the histogram intersection kernel.

5.2.2 Locality-constrained Linear Coding

As suggested by [65] that locality is more fundamental than sparsity since locality will contribute to sparsity while not necessary vice verse, Wang et al. [61] proposed an encoding scheme that incorporated locality constraint instead of the sparsity constraint in [64] in the following form:

$$\begin{aligned} &\arg\min_{\mathbf{C}} \sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{B}\mathbf{c}_i\|^2 + \lambda \|\mathbf{d}_i \odot \mathbf{c}_i\|^2 \\ &s.t.\ \mathbf{1}^{\mathbf{T}}\mathbf{c}_i = 1, \forall i \end{aligned} \tag{5.2}$$

where $\odot$ means the element-wise multiplication, and $\mathbf{d}_i \in \mathcal{R}^K$ is the locality adapter giving different freedom for each basis vector proportional to its similarity to the input descriptor $\mathbf{x}_i$. In particular,

$$\mathbf{d}_i = exp(\frac{dist(\mathbf{x}_i, \mathbf{B})}{\sigma}) \tag{5.3}$$

where $dist(\mathbf{x}_i, \mathbf{B}) = [dist(\mathbf{x}_1, \mathbf{b}_1), \cdots, dist(\mathbf{x}_i, \mathbf{b}_K)]'$, and $dist(\mathbf{x}_i, \mathbf{b}_j)$ is the Euclidean distance between $\mathbf{x}_i$ and $\mathbf{b}_j$. σ adjusts the speed of weight decay for the locality adapter. Usually, the distance $\mathbf{d}_i$ is further normalized to $(0, 1]$ by subtracting the $max(dist(\mathbf{x}_i, \mathbf{B}))$ from $dist(\mathbf{x}_i, \mathbf{B})$.

5.2.3 Vector of Locally Aggregated Descriptors

Jégou et al. [27] proposed a vector representation of an image that aggregates descriptors according to a locality criterion in feature space, which is termed as VLAD. Given the codebook **B** learned by *k*-means clustering algorithm, each image descriptor x_i is then associated to its nearest visual codeword $\mathbf{b}_k = NN(x_i)$. The key idea of VLAD descriptor is to accumulate the vector differences $x_i - \mathbf{b}_k$ of the descriptor x_i assigned to each $\mathbf{b}_k$. In this way, VLAD defines the distribution of the descriptors w.r.t. the clustering center.

The dimension of VLAD representation is *KD*, where *K* is the number of codewords and *D* is the feature dimension. The aggregated feature vector of an image is represented by $v_{k,i} = \sum_{x\ such\ that\ NN(x)=\mathbf{b}_k} x_i - \mathbf{b}_{k,i}$, where x_i and $\mathbf{b}_{k,i}$ represent the ith image descriptor and its corresponding visual codeword $\mathbf{b}_k$. Further, the VLAD vector is l_2 normalized.

5.3 Max Pooling

Max pooling, one of the most widely used and successful spatial pooling methods, calculates the maximum of each component. Max pooling becomes more and more popular because of its great performance when applied with linear classifiers [64, 61]. Given the encoded feature vector **U** where each column corresponds to the responses of all the local descriptors to one specific codeword in the visual codebook **B**, max pooling is formulated as follows:

$$h_j = max\{\mid u_{1j} \mid, \mid u_{2j} \mid, \cdots, \mid u_{Nj} \mid\}, j = 1, 2, \cdots, K. \tag{5.4}$$

where N denotes the number of local descriptors in the region, and h_j is the maximum response of all local descriptors to a specific codeword. In this chapter, max pooling method is used in LLC [61]. Note that average pooling is used for the Fisher Vector coding approach.

In summary, this chapter discusses another visual codebook construction method by *k*-means clustering, which is the most widely used approach in modern image classification tasks. Then, we present several patch based image descriptors encoding methods, such as the Vector Quantization coding, LLC, and VLAD. Another

spatial pooling method - max pooling is also presented. Recall that all methods but BoVW take the spatial pyramid strategy to incorporate the rough geometry information. Moreover, LLC takes the max pooling method to aggregate the encoded features while SPM and VLAD use the average pooling technique. The methods presented in this chapter serve as the comparison group.

Chapter 6

Experiments

Based on the methods and strategies discussed above, we present an extensive experimental study on our in-house animal image data. We first discuss how the image dataset was collected and prepared for our experiments. In the second part, we show our experimental results and experiment analysis.

6.1 Experimental Setup

6.1.1 Image Data Collection

In the middle of the year 2011, the Alberta Innovates Technology Futures (AITF) deployed infrared remote camera devices (Reconyx PC900 Hyperfire) at more than 60 sites (see Figure 2.2) in Alberta's northeast boreal forest, north and west of Winefred Lake, and started to monitor animals' movement and behaviour around the remote devices. Figure 6.1 shows an example of the remote camera. The project aims to collect image data about *Deer* and then estimating the population and density of *Deer* in the area.

The infrared remote camera is motion sensitive and is triggered once the motion is detected in front of it. The camera records the presences of animals during day time and night time. Our dataset was collected from November 2011 to November 2012. Images were captured at a low and irregular frequency, and have a high resolution of 1536×2048. We find there are only a few image sequences having more than 10 continuous images. Figure 6.2 shows four image sequences containing *Bear*, *Deer*, *Wolf*, and *Lynx*. Since the camera traps require regular battery replace-

Figure 6.1: An example shows a remote camera (Reconyx PC900 Hyperfire) is setup with a tree.

ment and transfer of the stored images, humans appear in images and account for an non-negligible portion of the image dataset. Thus humans are considered as a species to be identified.

Table 6.1 shows the statistical information about our in-house dataset. As shown in the first column, we have 19 species including *Human*. The second column shows the total number of images we have for each species. Some species like *Deer* and *Rabbit* have several times more images than other species like *Lynx*, meaning that the dataset is imbalanced.

6.1.2 Image Data Preparation

As shown in Table 6.1, in our in-house dataset, we have 19 species, including *Human* passing in front of cameras and *Unknown Species* identified by ecologists. Note that there are cases where the animal species are wrongly identified by ecologist, as shown in Figure 6.3. Therefore, we first remove such wrong cases by inspecting the database. The number of frames for each species are highly imbalanced and some of them only have a very small number of images. Thus we remove

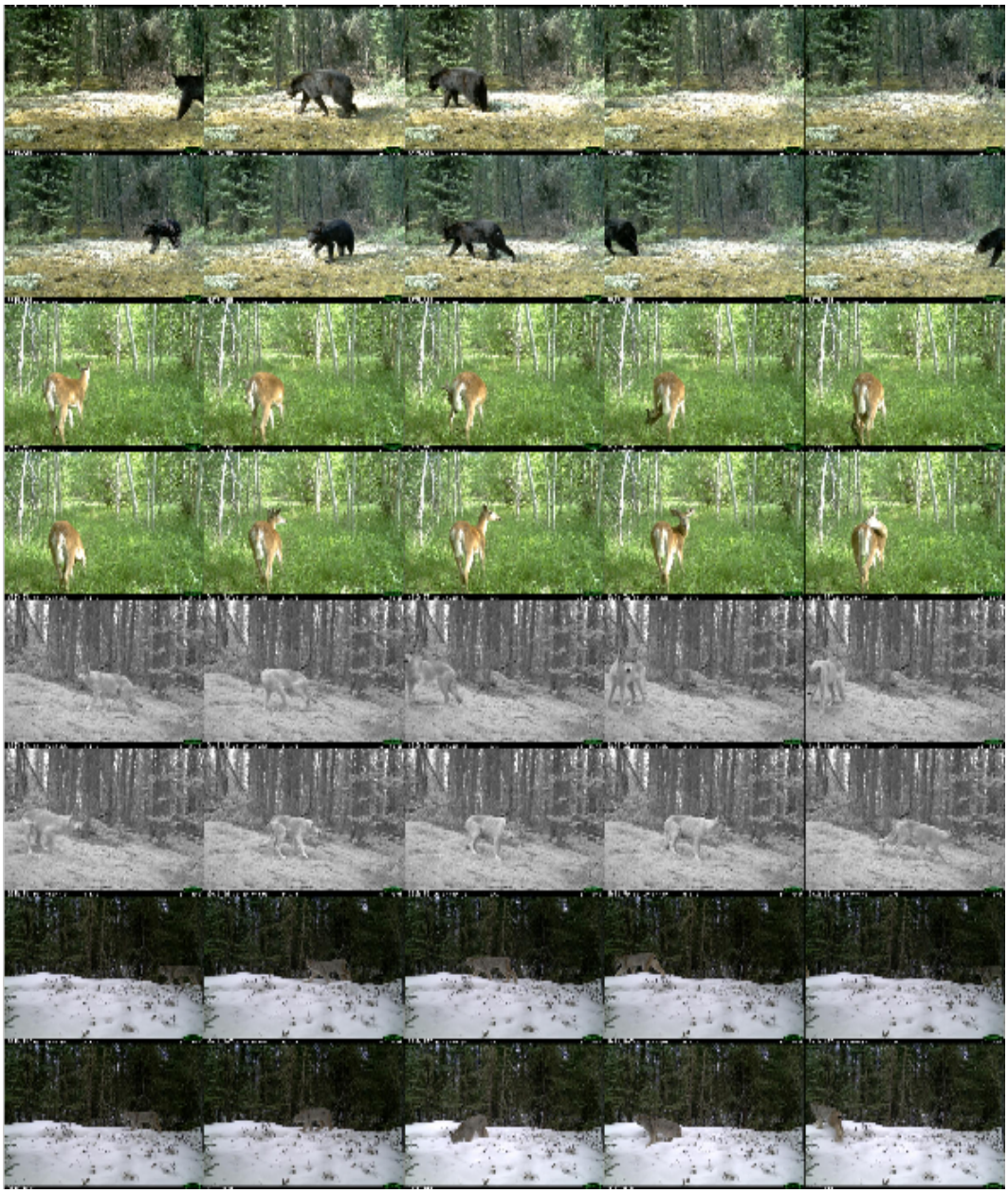


Figure 6.2: Some examples of image sequences. The first two rows is *Bear* sequence, the second two rows is *Deer* sequence, the third two rows is *Wolf* sequence, and the last two rows is *Lynx* sequence.

Table 6.1: The number of frames of each species. The first column shows the animal species, the second column shows the total frames of each species in the database, and the last column shows the total frames of each species used in our experiments.

Species	Number of Frames	Number of Remained Frames
Black Bear	421	382
Coyote	723	675
Human	1,325	1,076
Lynx	752	663
Rabbit	3,017	2,718
Deer	7,729	6,791
Wolf	475	344
Fisher	91	0
Moose	87	0
Caribou	57	0
Red Fox	52	0
Red Squirrel	43	0
Unknown Species	42	0
Grouse	31	0
Cougar	6	0
Wolverine	4	0
Marten	4	0
Grizzly Bear	3	0
Common Raven	2	0
Total	14,864	12,649

the species who have less than 100 frames. In addition, for a significant number of images, animals often show at the border of images and occupy a very small portion of the images. In this case, the animals are very difficult to recognize even by humans. Therefore, we remove such images for the assessment of our method. Figure 6.4 shows examples of the removed samples. As shown in the figure, it is very hard, even impossible, for humans to identify the species without seeing other informative images in this specific sequence. In this way, we have a total of 12,649 frames remaining with 7 species: *Bear, Coyote, Human, Lynx, Rabbit, Deer* and *Wolf*, as shown in Table 6.1 Column 3.

In addition, as a preprocessing step, we first resize the image into the resolution of 180×240 using bicubic interpolation which largely reduces the total size of the dataset. Then all colour images are mapped to gray-scale images by discard-

Figure 6.3: The wrong identification cases by ecologists. The textual description $XX \rightarrow YY$ above each image means species XX is wrongly identified as species YY.

ing colour information since we found that the species identification performances were almost the same by our Animal Species Identification System (ASIS) with or without colour information. This is all we did in the preprocessing stage and we term this dataset as raw images. On the other hand, we prepare another image dataset with backgrounds manually removed. We delimit a bounding box around the animal region and then save the image region inside the bounding box. Figure 6.5 shows an example of how to prepare a clean animal image. In this way, the collected dataset is termed as clean images. The animal body occupies at least half of the cleaned image and then further resized to 180×240 if the size is larger than 180×240. We test the performance of our method on these two datasets and verify how crucial it is to identify animal species on clean images.

6.1.3 Implementation Details

Parameters Setting

In our experiments, typically, we extract the dense Scale Invariant Feature Transform (SIFT) features and the cell-structured Local Binary Pattern (cLBP) descriptors from 16×16 image patches with the sampling grid space of 6 pixels. As discussed in Section 3.1, the SIFT feature extracted from each patch is a 128 dimensional vector. In our experiments, each 128 dimensional feature vector is further projected to 80 dimensional feature space by applying Principle Component Analysis (PCA). For cLBP extraction, we use $\text{LBP}^2_{8,1}$ to circularly extract uniform

Local Binary Pattern (LBP) from each patch. Finally, we concatenate each single vector from each image patch to form a global image vector. Further, we use the late fusion strategy [57] to combine the dense SIFT feature and the cLBP descriptors as the empirical study proves late fusion tends to give better performance in image classification tasks.

As extensive experimental study shows, the classification accuracy varies with the size of visual codebook. Therefore, we run experiments with different codebook sizes for both Gaussian Mixture Model clustering and *k*-means clustering. Typically, in many studies, the *k*-means codebook size *K* is set to 1024, and Gaussian Mixture Model codebook size *K* is set to 64 or 256. Besides, we use the means estimated by *k*-means algorithm to initialize the means vector for Gaussian Mixture Model, and thus the variance based on the learned means to initialize the covariance matrices. We randomly select 250 images from each species to constitute our training set and randomly sample 100 image features as in [35, 64, 61] from each image to create the visual codebook. VLfeat library [60] is used in our experiments.

Multi-class Linear Support Vector Machines

For computational efficiency, we use a linear Support Vector Machines (SVMs) classifier introduced in [64]. The linear SVMs classifier uses one-against-all strategy to train *S* binary linear SVMs, each solving the following unconstrained convex optimization problem

$$\min_{\mathbf{w}_c}\{J(\mathbf{w}_c) = \|\mathbf{w}_c\|^2 + C\sum_{i=1}^{N} l(\mathbf{w}_c; y_i^c, \mathbf{x}_i)\} \tag{6.1}$$

where S is the number of categories, $y_i^c = 1$ if $y_i = c$, otherwise $y_i^c = -1$, and $l(\mathbf{w}_c; y_i^c, \mathbf{x}_i)$ is the loss function. In this manner, the training cost scales linearly with the number of training samples, while the testing cost is constant.

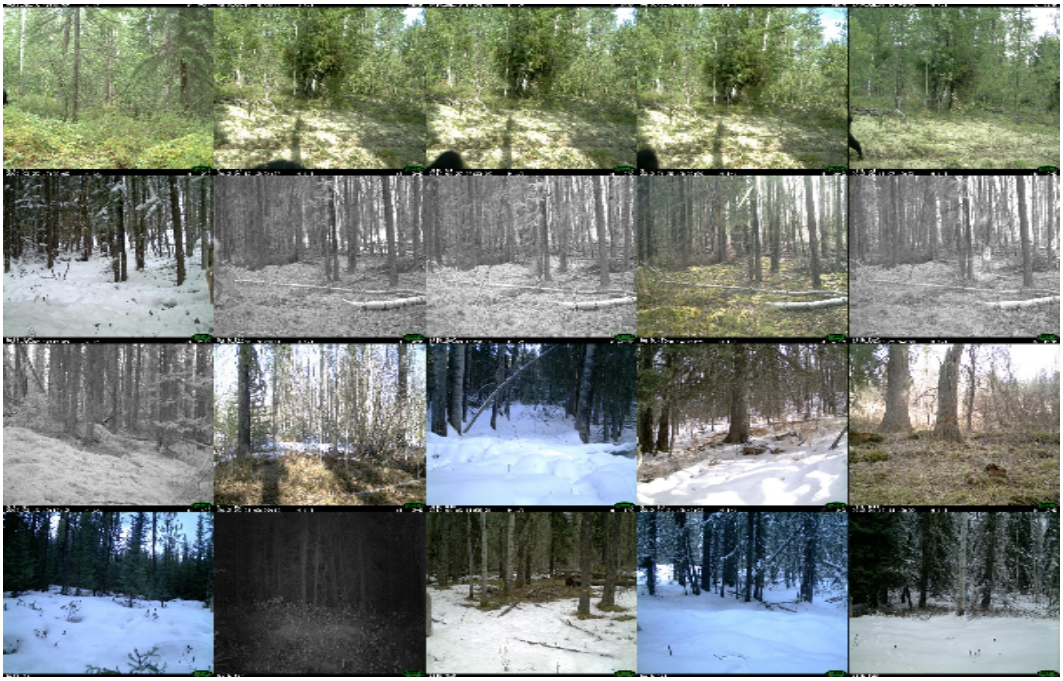


Figure 6.4: Some discarded examples. From top to bottom: *Bear*, *Deer*,*Wolf*, and *Lynx*. These animal samples occupy a very small part of the entire image, and thus are very hard to recognize, even for humans, so we discard these samples.

Figure 6.5: An example shows how to prepare a clean image. From left to right: raw image, build a bounding box, clean image.

Table 6.2: The confusion matrix of species identification based on ASIS using the SIFT features on raw image dataset (accuracy = 86.64% ± 0.13).

		Prediction						
		Bear	Coyote	Human	Lynx	Rabbit	Deer	Wolf
Ground Truth	Bear	93.33%	0.30%	0.3%	1.36%	0.00%	3.94%	0.76%
	Coyote	0.38%	76.47%	0.71%	7.01%	6.64%	3.91%	4.89%
	Human	0.51%	0.58%	92.81%	2.23%	0.17%	2.91%	0.80%
	Lynx	2.57%	12.11%	1.74%	71.82%	5.33%	3.05%	3.39%
	Rabbit	0.08%	1.66%	0.74%	3.34%	92.37%	0.91%	0.90%
	Deer	2.63%	2.90%	1.13%	2.87%	0.83%	85.19%	4.45%
	Wolf	2.13%	6.38%	0.43%	1.49%	1.91%	3.62%	84.04%

6.2 Experimental Result and Analysis

6.2.1 Dataset 1

We conduct our first set of experiments on raw animal images as prepared in Section 6.1.2. In this section, we collect experimental results based on our Animal Species Identification System (ASIS), the original Bag of Visual Words method (BoVW) [15], the Spatial Pyramid Matching approach (SPM) [35], the Locality-constrained Linear Coding approach (LLC) [61], and Vector of Locally Aggregated Descriptors (VLAD) [27]. For each method, we repeat our experiments ten times and thus give the mean classification accuracy and standard deviation for comparison. Especially, we present the confusion matrix of animal species identification based on ASIS with 256 Gaussian components.

The confusion matrix of species identification based on ASIS using the dense SIFT features is shown in Table 6.2. We achieve a classification accuracy of 86.64% ± 0.13 based on 10 runs. As shown in the table, the identification accuracy for *Human* is 92.81%. This is probably because humans in images always show in upright posture and wear colorful clothes. Also, the identification accuracy for *Bear* is higher than 93% which can be attributed to the black appearance of *Bear*. The classification accuracies for *Coyote*, *Lynx*, and *Wolf* are much lower because these three species look quite similar.

Table 6.3 presents the confusion matrix of species identification based on ASIS using the cLBP descriptors. The overall classification accuracy is 76.67% ± 0.24

Table 6.3: The confusion matrix of species identification based on ASIS using the cLBP features on raw image dataset (accuracy = 76.67% $\pm$ 0.24).

		Prediction						
		Bear	Coyote	Human	Lynx	Rabbit	Deer	Wolf
Ground Truth	Bear	86.82%	1.36%	1.21%	1.36%	0.61%	6.52%	2.12%
	Coyote	3.39%	67.29%	2.45%	4.66%	14.16%	3.67%	4.38%
	Human	4.41%	0.53%	88.40%	1.26%	0.00%	4.79%	0.61%
	Lynx	6.00%	9.10%	4.46%	60.15%	13.22%	4.75%	2.32%
	Rabbit	0.75%	2.67%	2.07%	2.76%	89.95%	1.19%	0.62%
	Deer	8.73%	3.12%	5.21%	3.28%	3.82%	71.57%	4.27%
	Wolf	2.77%	6.81%	1.91%	0.21%	3.83%	3.83%	80.64%

Table 6.4: The confusion matrix of species identification based on ASIS using the SIFT and the cLBP features on raw image dataset (accuracy = 86.75% $\pm$ 0.12).

		Prediction						
		Bear	Coyote	Human	Lynx	Rabbit	Deer	Wolf
Ground Truth	Bear	94.70%	0.00%	0.30%	0.61%	0.00%	3.64%	0.76%
	Coyote	0.42%	75.76%	0.89%	6.26%	7.91%	3.91%	4.85%
	Human	0.65%	0.39%	92.86%	2.23%	0.05%	3.20%	0.63%
	Lynx	2.86%	11.43%	1.65%	70.41%	6.63%	3.44%	3.58%
	Rabbit	0.11%	1.56%	1.01%	3.06%	92.79%	0.64%	0.83%
	Deer	2.94%	2.63%	1.12%	2.71%	1.03%	85.31%	4.27%
	Wolf	2.13%	6.17%	0.00%	1.06%	1.70%	4.04%	84.89%

which is much lower than that using the dense SIFT features. As a consistent observation, the classification for each species drops compared to the same approach using the dense SIFT features. This observation clarifies the fact that the dense SIFT features has much more discriminative power in describing camera trapping images. Besides, the cLBP descriptor seems to be more appropriate to describe *Rabbit* images.

Table 6.4 shows the confusion matrix of species identification based on ASIS using the fusion of the dense SIFT features and the cLBP descriptors with the overall accuracy of 86.75% $\pm$ 0.12. In our experiments, we adopt late fusion strategy and we empirically find 0.7 and 0.3 weights for the SIFT and the cLBP output the best result. As we can see, the overall accuracy is boosted. Specially, the classification accuracies for all species but *Coyote* and *Lynx* are boosted albeit slightly. This implies that the combination of the dense SIFT and the cLBP is not a good descriptor for *Coyote* and *Lynx* but it is a good strategy for identifying other species.

The classification performances based on all approaches are shown in Table 6.5. We examine the different performances with different codebook sizes varying from 64 to 2048. However, we did not examine the performance of ASIS and VLAD with codebook size larger than 256 since these two methods capture the distribution of low-level image descriptors and thus retain much more information than other approaches. The resulting feature vector lies in a significantly high dimensional feature space which leads to very high memory usage when the codebook size grows higher than 256. However, as we can see, even with the codebook size of 64, ASIS achieves the best performance among all methods. In addition, our method consistently achieves the highest classification accuracies under different settings. For approaches of BoVW, SPM, and LLC which only consider the codeword frequency or the linear combination of nearest words in the local coordinate system, their performances are inferior to those of ASIS and VLAD. The table demonstrates the effectiveness of ASIS based on the fusion of the dense SIFT features and the cLBP descriptors.

Figure 6.6 shows the top ten hits for each species. The decimal number above each image indicates the probability of mapping the image to the specified species.

These probabilities, also known as confidence scores, are learned according to the method discussed in [51]. The confidence score indicates how confident the system is about a specific prediction. The high confidence score means the system is highly confident about this identification and there is no need to manually double-check it while the low confidence score suggests that the identification result probably cannot be trusted and needs manual double-checking. Figure 6.7 shows some wrong identification cases with low confidence scores. Based on our experiments, we found that the confidence scores of wrong identification cases are usually lower than 0.6 while confidence scores of correct identification cases are almost always higher than 0.75. Those confidence scores are valuable to ecologists for further analysis.

6.2.2 Dataset 2

We conduct our second set of experiments on clean images. Figure 6.8 shows a few samples of clean images. Still, the clean image has dynamic background, illumination changes, and viewpoint changes. However, in this case, we can extract more effective image features from clean images since we have larger opportunity in extracting features from animal region when training the visual codebook and has less negative interference introduced by cluttered background.

In this set of experiments, we compare the performances between the raw image dataset and the clean image dataset based on ASIS. We also vary the codebook size from 64 to 256. From Table 6.6, we see that identification accuracies on clean image dataset consistently higher than those on raw image dataset for the dense SIFT feature and the fusion of the dense SIFT and the clBP features. However, the accuracy decreases when using the cLBP descriptors only.

6.2.3 Impact of Spatial Pyramid Level

We further examine the impact of the number of spatial pyramid levels. Specially, we run our experiments on raw image dataset using the Fisher Vector coding approach with the codebook size of 64. In the Figure 6.9, we report the mean classification accuracy under three spatial pyramid level settings. From the figure, we

Table 6.5: The animal species identification performance based on different approaches on raw image dataset

feature types		codebook size	BoVW	SPM	LLC	VLAD	ASIS
SIFT	cLBP						
✓	×	64	0.5458 ± 0.0203	0.6432 ± 0.0131	0.7004 ± 0.0090	0.8384 ± 0.0023	**0.8545 ± 0.0041**
✓	×	128	0.6083 ± 0.0173	0.7160 ± 0.0087	0.7316 ± 0.0069	0.8456 ± 0.0017	**0.8612 ± 0.0014**
✓	×	256	0.6657 ± 0.0199	0.7516 ± 0.0116	0.7640 ± 0.0031	0.8489 ± 0.0049	**0.8664 ± 0.0013**
✓	×	512	0.7180 ± 0.0094	0.7860 ± 0.0039	0.7782 ± 0.0047	-	-
✓	×	1024	0.7629 ± 0.0027	0.8084 ± 0.0040	0.7987 ± 0.0069	-	-
✓	×	2048	0.7974 ± 0.0058	0.8247 ± 0.0072	0.8089 ± 0.0028	-	-
×	✓	64	0.4165 ± 0.0086	0.5178 ± 0.0100	0.5472 ± 0.0110	0.7045 ± 0.0058	**0.7501 ± 0.0042**
×	✓	128	0.4504 ± 0.0060	0.5422 ± 0.0085	0.5923 ± 0.0040	0.7124 ± 0.0104	**0.7639 ± 0.0057**
×	✓	256	0.4754 ± 0.0017	0.5663 ± 0.0078	0.6436 ± 0.0073	0.7331 ± 0.0059	**0.7667 ± 0.0024**
×	✓	512	0.5118 ± 0.0078	0.6001 ± 0.0061	0.6740 ± 0.0085	-	-
×	✓	1024	0.4912 ± 0.0105	0.6393 ± 0.0057	0.6709 ± 0.0131	-	-
×	✓	2048	0.5447 ± 0.0050	0.6702 ± 0.0076	0.7000 ± 0.0034	-	-
✓	✓	64	0.5606 ± 0.0195	0.6689 ± 0.0084	0.6988 ± 0.0095	0.8406 ± 0.0037	**0.8558 ± 0.0022**
✓	✓	128	0.6323 ± 0.0120	0.7342 ± 0.0077	0.7414 ± 0.0015	0.8431 ± 0.0017	**0.8638 ± 0.0012**
✓	✓	256	0.6888 ± 0.0188	0.7682 ± 0.0078	0.7784 ± 0.0008	0.8501 ± 0.0049	**0.8675 ± 0.0012**
✓	✓	512	0.7300 ± 0.0056	0.7975 ± 0.0032	0.7987 ± 0.0043	-	-
✓	✓	1024	0.7245 ± 0.0108	0.8142 ± 0.0048	0.8022 ± 0.0035	-	-
✓	✓	2048	0.7737 ± 0.0039	0.8306 ± 0.0065	0.8174 ± 0.0049	-	-

Note: '-' means no experiment for the specific setting since vectors lie in a significantly high-dimensional feature space.

Figure 6.6: Top ten hits for each species. From top row to the bottom row: *Bear*, *Coyote*, *Human*, *Lynx*, *Rabbit*, *Deer*, and *Wolf*.

Figure 6.7: Some wrong identification cases. The label above each image "Species1 → Species2" means Species1 is wrongly identified as Species2 by our system. The decimal value in the parentheses is the confidence score for this prediction.

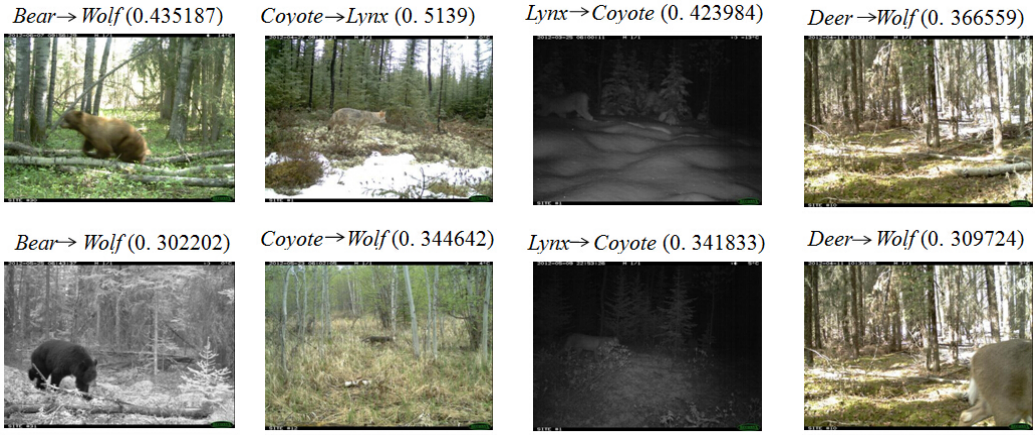


Figure 6.8: Animal images with background manually removed. Each cleaned image has an animal occupying more than half of the entire image.

Table 6.6: Classification performance comparison between raw images and clean images based on ASIS.

features		codebook size	Raw Image Dataset	Clean Image Dataset
SIFT	cLBP			
✓	×	64	0.8545 ± 0.0041	0.9000 ± 0.0016
✓	×	128	0.8612 ± 0.0014	0.9038 ± 0.0019
✓	×	256	0.8664 ± 0.0013	0.9068 ± 0.0031
×	✓	64	0.7501 ± 0.0042	0.7373 ± 0.0024
×	✓	128	0.7639 ± 0.0057	0.7377 ± 0.0052
×	✓	256	0.7667 ± 0.0024	0.7199 ± 0.0042
✓	✓	64	0.8558 ± 0.0022	0.9062 ± 0.0012
✓	✓	128	0.8638 ± 0.0012	0.9084 ± 0.0012
✓	✓	256	0.8675 ± 0.0012	0.9100 ± 0.0027

Figure 6.9: Impact of spatial pyramid level. 1-level means image regions include only the entire image; 2-level means image regions include four additional subregions; and 3-level means image regions include additional 16 subregions at each corresponding resolution level.

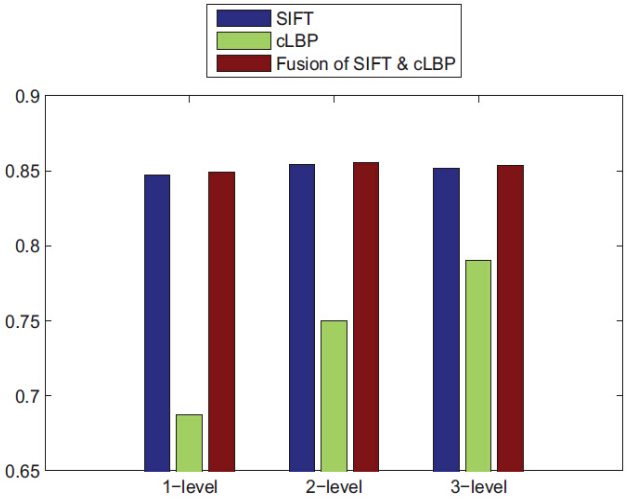


Table 6.7: Impact of the cLBP feature extraction strategy. The species identification accuracies are based on ASIS on raw image dataset.

codebook size	non-overlapped image patch	overlapped image patch
64	0.7030	0.7501
128	0.7250	0.7639
256	0.7363	0.7667

observe that increasing the spatial pyramid level from 1 to 2 results in the higher classification accuracy while further increasing the spatial pyramid level decreases the classification accuracy. The 2-level spatial pyramid is the best choice.

6.2.4 Impact of overlapped cLBP and non-overlapped cLBP

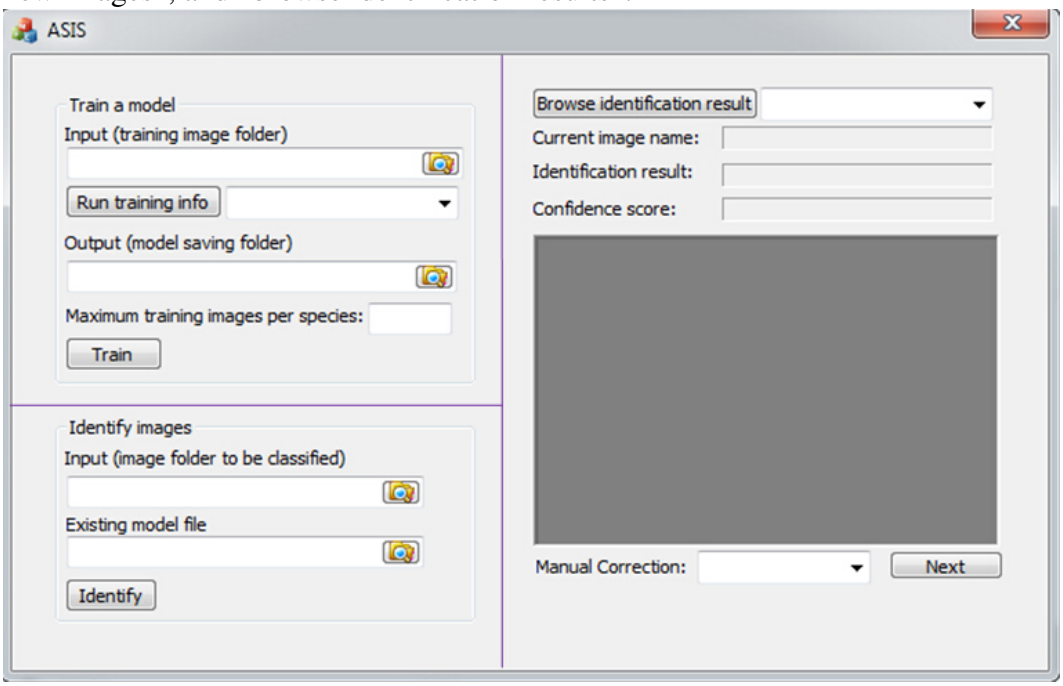
As stated in [62], the authors extract the cLBP features from non-overlapped image patches. However, in our experiments, we follow the same strategy as we did for the dense SIFT feature extraction to extract the cLBP features, i.e., from overlapped image patches. As shown in Table 6.7, the species identification accuracies based on the cLBP features extracted from overlapped image patches are consistently higher than those based on the cLBP features extracted from non-overlapped image patches.

6.2.5 ASIS User Interface Illustration

We developed a user interface using C++ based on OpenCV and LIBSVM [13]. This application is called Animal Species Identification System (ASIS). Note that ASIS runs on the raw image dataset and does not support clean images cut-out since the process of manually cropping the animals is very time-consuming and is not practical for the animal species identification problem. In fact, the process of manually animal image cropping turns the animal species identification task into other problems like individual animal identification.

Figure 6.10 shows the user interface of ASIS. ASIS allows a user to train a new model when it is necessary, classify new images, and browse the classification results. Specially, a user can check whether a specific identification with a low confidence score is correct or not. If it is a wrong identification, manual correction

Figure 6.10: The user interface of ASIS based on Microsoft Foundation Class Library. Mainly, the user interface has three sections: "training a model", "identify new images", and "browse identification results".



is allowed.

A user can select a training image database to train a new model and select a directory to save the trained model. It is very important and useful to offer the option to train a new model, especially when images containing new species arrive. A reasonable assumption is that a user would like to know the basic statistical information of the training dataset after he/she selects the training dataset. It can be achieved by clicking "Run training info" button and the statistical information is shown in the neighbouring drop-down list. Furthermore, a user would like to decide how many images from each species should be used to train the new model based on the statistical information of the training dataset. Clicking "Train" button allows ASIS to start training a new model. An example of the training process is shown in Figure 6.11.

Once we have trained a model based on the training image dataset, we can start identifying new images by selecting the directory of new images and the directory of the trained model. Clicking the "Identify" button starts the identification process.

Figure 6.11: The process of training a new model based on the selected training image dataset. The left image shows the statistical information of the training dataset in the drop-down list.

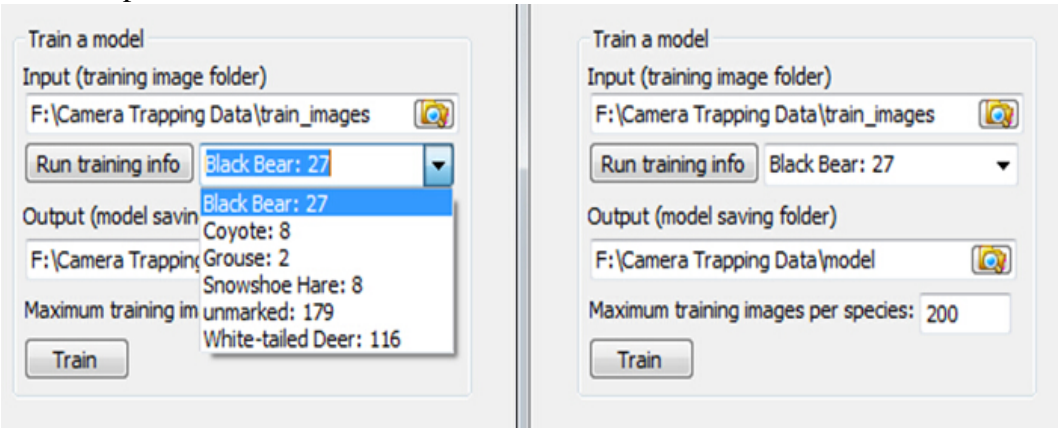
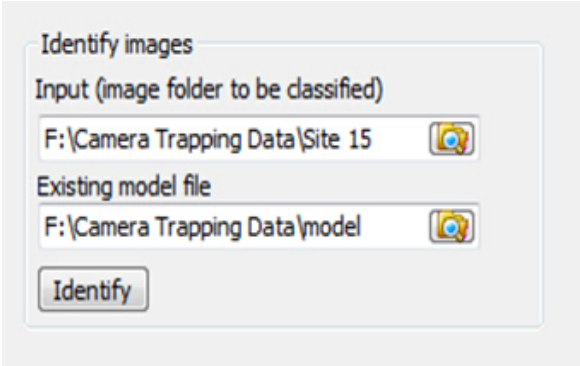


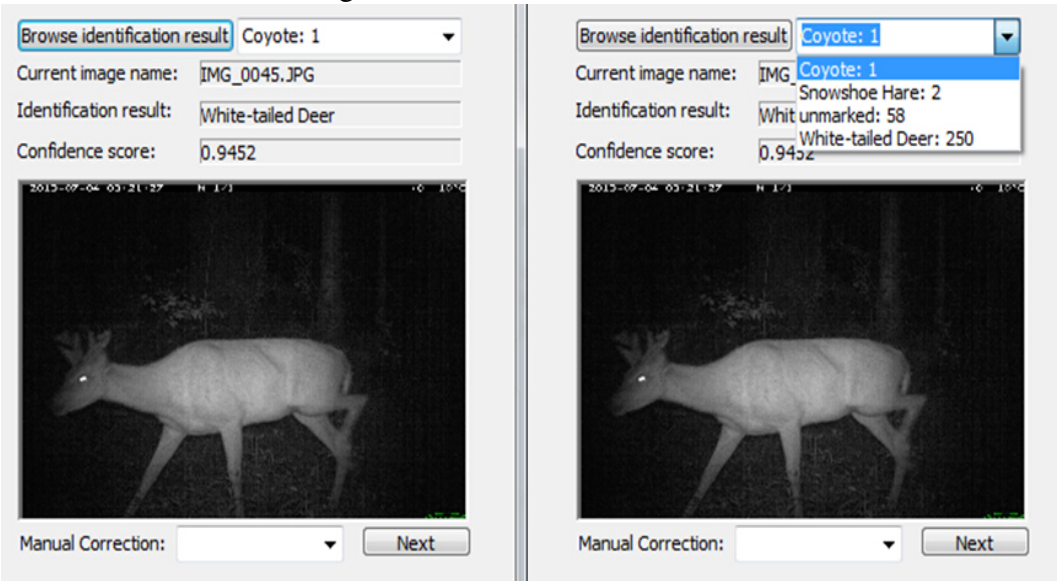
Figure 6.12 shows the process of identifying new images based on the selected model.

Figure 6.12: The process of identifying the selected image dataset based on the selected model.



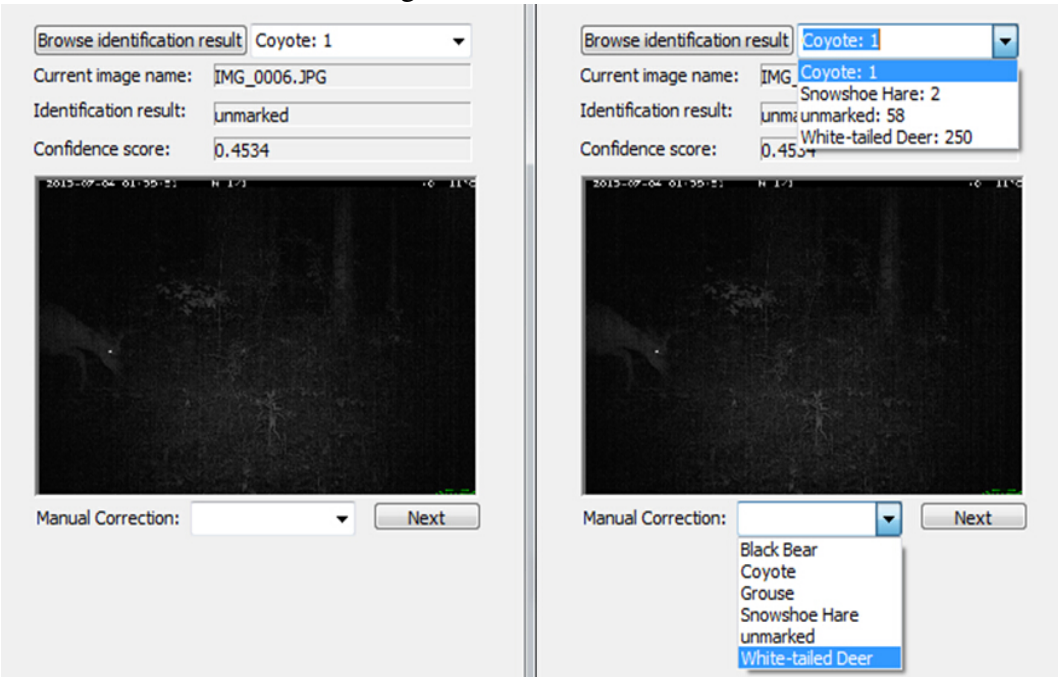
After the process of animal species identification, a user can browse the identification results by clicking the "Browse identification result" button. Figure 6.13 shows an example of correct identification with a high confidence score. In the figure, ASIS shows the statistical information of the identification result of the selected image directory in the drop-down list right to the "Browse identification result" button. In addition, ASIS shows the current image name, the identified result for the current image, and the confidence score for this identification. For a correct identification, a user can click the "next" button to browse the identification result of the next image. Figure 6.14 shows an example of wrong identification with a low

Figure 6.13: An example of the correct identification with a high confidence score of 0.9452. For this case, a user can click the "next" button to browse the identification result for the next image.



confidence score. In the figure, a *White-tailed Deer* is identified as *unmarked* with a low confidence of 0.4534. In this case, a user can correct the identification result by first selecting the correct species from the drop-down list and then clicking the "next" button. The "next" button also brings a user to the identification result of the next image.

Figure 6.14: An example of manually checking the identification result with a low confidence score. ASIS allows a user to manually correct the identification result if it is wrong. It is done by selecting the correct species from the drop-down list left to the "next" button and clicking the "next" button.



Chapter 7

Conclusion and Future Work

7.1 Conclusion

Given a database of animal images with illumination, scale, and viewpoint changes and cluttered background, we propose to use the Fisher Vector coding approach to combine the dense Scale Invariant Feature Transform (SIFT) features and the cell-structured Local Binary Pattern (cLBP) descriptors to automatically identify the animal species in an image captured by a camera trap. The Fisher Vector representation utilizes the powerful Fisher Kernel framework which combines the advantages of both generative and discriminative approaches. In our extensive experiments, we have shown that our approach performs much better than alternative ones in terms of classification accuracy. Besides, since our method works with a linear Support Vector Machines (SVMs) classifier, the computational speed is very fast.

Compared to the Bag of Visual Words (BoVW) model and its variants including BoVW [15], Spatial Pyramid Matching (SPM) [35], and Locality-constrained Linear Coding (LLC) [61], the Fisher Vector coding approach estimates the distribution of the underlying image descriptors and retains much more information of the image and thus results in significantly less approximation error. Taking the zero-order image statistics into account as well as the first-order and the second-order information, the final vector representation is more dense and lies in a high dimensional feature space, which is considered more appropriate to work with a linear SVMs classifier. The benefit of this additional image information is shown in our experiments.

Though the Vector of Locally Aggregated Descriptors (VLAD) [27] approach considers the distribution of the underlying image descriptors, it, on the other hand, uses the non-probabilistic k-means to learn the visual codebook and thus only utilizes the gradients with regard to the means learned by k-means clustering method. From this point, VLAD is viewed as a simplified version of Fisher Vector coding. This explains the improved performance of the Fisher Vectoring coding approach over that of VLAD.

From the experiments, we have demonstrated that the dense SIFT feature is much more powerful than the cLBP descriptor in describing the animal images on two datasets. Moreover, the fusion of these two features can boost somewhat the classification accuracy on both raw image dataset and clean image dataset.

In addition, we show that our method works well even for a small codebook size which thus requires less storage and more efficient memory usage. However, the alternative approaches can only achieve the satisfactory performance with an arbitrary large size of codebook. The small codebook size is a great advantage over alternative methods.

We also show the incorporation of the rough structure information is beneficial for the animal species identification performance. Furthermore, as shown in our experimental results, the identification accuracy on images whose backgrounds are manually removed is improved four percentage point over that of raw images.

Last but not least, we have some suggestions for ecologists about how to set up camera traps in the field and how to obtain better image data for further analysis. First, colour pillars could be put in front of a camera trap at a predefined distance in order to estimate the size of the captured animal based on the distance. Second, in order to obtain more information of the animal, the motion sensor of a camera trap could be replaced with the heat sensor or the depth sensor. This would provide more opportunities for computer vision researchers to manipulate the image data and obtain either the contour or the shape of the animal which can be further utilized for the task of animal species identification. Third, instead of only storing images containing animals when a camera trap is triggered by motion detection, the camera provider could reprogram the camera and store both of the pure background image

and the animal image. This would largely simplify the problem for animal species identification, animal counting, tracking, etc.

7.2 Future Work

There are some directions to go in the future. First, more image features like shape feature should be examined on the image dataset to determine which one has more deterministic power in abstracting the animal image or which of them work better. Second, as we have seen, our classifier indeed works better on images with background removed than on the raw dataset. Therefore, it is worth exploring the background subtraction methods which can extract animal regions for feature extraction. However, based on our preliminary experiments, we found that traditional background modelling methods did not give us satisfactory foregrounds since our images were captured at quite a low and irregular frequency. Third, in our experiments, we randomly sample 100 features from each image to learn a visual codebook. Other feature sampling strategies should be investigated to obtain more meaningful features from the animal region. Fourth, as the research progress in the community reveals, deep learning currently is the most promising direction to solve the unsupervised feature learning and image classification problems [34, 14]. It is a promising direction to apply deep learning methods to our animal species identification task. Fifth, we would like to integrate some biology rules into our system. For example, we could reject some wrong identifications based on the fact that there is no bear in winter. The different colour information of rabbit in winter and in summer could help to build a more accurate representation model. Last, by considering the colour information of an image as the prior knowledge, we could make our routine focus on the non-green image region and thus further reduce the impact of the background.

Bibliography

[1] Samer Alasaad, José E. Granados, Paulino Fandos, Francisco-Javier Cano-Manuel, Ramón C. Soriguer, and Jesús M. Pérez. The use of radio-collars for monitoring wildlife diseases: a case study from Iberian ibex affected by Sarcoptes scabiei in Sierra Nevada, Spain. *Parasites & Vectors*, 6(1):1–5, 2013.

[2] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM Press, New York, 1999.

[3] Jeffrey R. Ball and Erin M. Bayne. Using video monitoring to assess the accuracy of nest fate and nest productivity estimates by field observation. *The Auk*, 129(3):438–448, 2012.

[4] Jeffrey R. Ball, Erin M. Bayne, Craig S. Machtans, Terrell D. Rich, Coro Arizmendi, Dean W. Demarest, and Craig Thompson. Video identification of boreal forest songbird nest predators and discordance with artificial nest studies. In *Tundra to Tropics: Connecting Birds, Habitats and People (Rich, Terrell D. and Arizmendi, Coro and Demarest, Dean W. and Thompson, Craig, Editors). Proceedings of the Fourth International Partners in Flight Conference*, pages 37–44, 2009.

[5] Richard F. W. Barnes. How reliable are dung counts for estimating elephant numbers? *African Journal of Ecology*, 39(1):1–9, 2001.

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

[7] Neil Birkbeck and Martin Jagersand. Visual tracking using active appearance models. In *2013 International Conference on Computer and Robot Vision*, pages 2–9. IEEE Computer Society, 2004.

[8] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[9] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

[10] Yaw Boafo, Mildred Manford, Richard Barnes, Emmanuel Hema, Emmanuel Danquah, Nandjui Awo, and Umaru-Farouk Dubiure. Comparison of two dung count methods for estimating elephant numbers at Kakum Conservation Area in Southern Ghana. *Pachyderm*, pages 34–40, 2009.

[11] Douglas T. Bolger, Thomas A. Morrison, Bennet Vance, Derek Lee, and Hany Farid. A computer-assisted system for photographic mark–recapture analysis. *Methods in Ecology and Evolution*, 3(5):813–822, 2012.

[12] Stephen T. Buckland, David L. Borchers, Alistair I. Johnston, Peter A. Henrys, and Tiago A. Marques. Line transect methods for plant surveys. *Biometrics*, 63(4):989–998, 2007.

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

[14] D. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3642–3649, June 2012.

[15] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, volume 1, pages 1–22, 2004.

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[17] Lee Eberhardt and Robert C. Van Etten. Evaluation of the pellet group count as a deer census method. *The Journal of Wildlife Management*, 20(1):70–74, 1956.

[18] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[19] Peter.J. Fashing and Marina Cords. Diurnal primate densities and biomass in the Kakamega Forest: an evaluation of census methods and a comparision with other forests. *American Journal of Primatology*, 50(2):139–152, 2000.

[20] Aravind Ganapathiraju, Jonathan E. Hamaker, and Joseph Picone. Applications of support vector machines to speech recognition. *Signal Processing, IEEE Transactions on*, 52(8):2348–2355, 2004.

[21] Kristen Grauman and Trevor Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1458–1465. IEEE, 2005.

[22] Kristen Grauman and Trevor Darrell. Unsupervised learning of categories from sets of partially matching image features. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 19–25. IEEE, 2006.

[23] Michael Greenspan and Pierre Boulanger. Efficient and reliable template set matching for 3D object recognition. In *3-D Digital Imaging and Modeling, 1999. Proceedings. Second International Conference on*, pages 230–239, 1999.

[24] Guodong Guo, Stan Z. Li, and Kap Luk Chan. Face recognition by support vector machines. In *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pages 196–201. IEEE, 2000.

[25] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, volume 15, page 50. Manchester, UK, 1988.

[26] Tommi Jaakkola and David Haussler. Exploiting generative models in discriminative classifiers. *Advances in Neural Information Processing Systems*, pages 487–493, 1999.

[27] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3304–3311. IEEE, 2010.

[28] Thorsten Joachims. *Text categorization with support vector machines: Learning with many relevant features*. Springer, 1998.

[29] Mark J. Jordan, Reginald H. Barrett, and Kathryn L. Purcell. Camera trapping estimates of density and survival of fishers Martes pennanti. *Wildlife Biology*, 17(3):266 – 276, 2011.

[30] K. Ullas Karanth and James D. Nichols. Estimation of tiger densities in India using photographic captures and recaptures. *Ecology*, 72(8):2852–2862, 1998.

[31] Roland Kays, Bart Kranstauber, Patrick A. Jansen, Chris Carbone, Marcus J. Rowcliffe, Tony Fountain, and Sameer Tilak. Camera traps as sensor networks for monitoring animal communities. In *LCN*, pages 811–818. IEEE, 2009.

[32] Yan Ke and Rahul Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–506. IEEE, 2004.

[33] So-Hyeon Kim, Do-Hyeun Kim, and Hee-Dong Park. Animal situation tracking service using RFID, GPS, and sensors. In *Computer and Network Technology (ICCNT), 2010 Second International Conference on*, pages 153–156, April 2010.

[34] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural information Processing Systems*, pages 1097–1105, 2012.

[35] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178. IEEE, 2006.

[36] Fei-Fei Li, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106(1):59–70, 2007.

[37] Fei-Fei Li and Pietro Perona. A Bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 524–531. IEEE, 2005.

[38] Stuart Lloyd. Least squares quantization in PCM. *Information Theory, IEEE Transactions on*, 28(2):129–137, 1982.

[39] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

[40] Bingpeng Ma, Yu Su, and Frédéric Jurie. Local descriptors encoded by fisher vectors for person re-identification. In *Computer Vision–ECCV 2012. Workshops and Demonstrations*, pages 413–422. Springer, 2012.

[41] Richard D. Mace, Steven C. Minta, Timothy L. Manley, and Keith E. Aune. Estimating grizzly bear population size using camera sightings. *Wildlife Society Bulletin*, 22:74–83, 1994.

[42] Fernanda F. C. Marques, Stephen T. Buckland, David Goffin, Camilla E. Dixon, David L. Borchers, Brenda A. Mayle, and Andrew J. Peace. Estimating deer abundance from line transect surveys of dung: Sika deer in Southern Scotland. *Journal of Applied Ecology*, 38(2):pp. 349–363, 2001.

[43] Krystian Mikolajczyk and Cordelia Schmid. Comparison of affine-invariant local detectors and descriptors. In *Proc. European Signal Processing Conf*, 2004.

[44] Ionut Mironica, Bogdan Ionescu, Jasper Uijlings, and Nicu Sebe. Fisher kernel based relevance feedback for multimodal video retrieval. In *Proceedings of the 3rd ACM Conference on International Conference on Multimedia Retrieval*, pages 65–72. ACM, 2013.

[45] David Nistér and Henrik Stewénius. Scalable recognition with a vocabulary tree. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2161–2168. IEEE, 2006.

[46] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51–59, 1996.

[47] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, 2002.

[48] Florent Perronnin and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[49] Florent Perronnin, Yan Liu, Jorge Sánchez, and Hervé Poirier. Large-scale image retrieval with compressed fisher vectors. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3384–3391. IEEE, 2010.

[50] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV (4)*, pages 143–156, 2010.

[51] John C Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*. Citeseer, 1999.

[52] Andrew J. Plumptre. Monitoring mammal populations with line transect techniques in African forests. *Journal of Applied Ecology*, 37(2):356–368, 2000.

[53] Nathan J. Roberts. Investigation into survey techniques of large mammals: surveyor competence and camera-trapping vs. transect-sampling. *Bioscience Horizons*, 4(1):40–49, March 2011.

[54] Leandro Silveira, Anah T.A. Jácomo, and José Alexandre F. Diniz-Filho. Camera trap, line transect census and track surveys: a comparative evaluation. *Biological Conservation*, 114(3):351 – 355, 2003.

[55] Karen Simonyan, Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Fisher vector faces in the wild. In *Proc. BMVC*, volume 1, page 7, 2013.

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

[57] Cees GM Snoek, Marcel Worring, and Arnold WM Smeulders. Early versus late fusion in semantic video analysis. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, pages 399–402. ACM, 2005.

[58] Richard Szeliski. *Computer vision: algorithms and applications*. Springer, 2010.

[59] Vladimir Vapnik. *The nature of statistical learning theory*. Springer, 2000.

[60] Andrea Vedaldi and Brian Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008.

[61] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang, and Yihong Gong. Locality-constrained linear coding for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3360–3367. IEEE, 2010.

[62] Xiaoyu Wang, Tony X. Han, and Shuicheng Yan. An HOG-LBP human detector with partial occlusion handling. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 32–39. IEEE, 2009.

[63] Qiong Wu and Pierre Boulanger. Real-time estimation of missing markers for reconstruction of human motion. In *Virtual Reality (SVR), 2011 XIII Symposium on*, pages 161–168. IEEE, 2011.

[64] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1794–1801. IEEE, 2009.

[65] Kai Yu, Tong Zhang, and Yihong Gong. Nonlinear learning using local coordinate coding. In *NIPS*, volume 9, page 1, 2009.

[66] Xiaoyuan Yu, Jiangping Wang, Roland Kays, Patrick A Jansen, Tianjiang Wang, and Thomas Huang. Automated identification of animal species in camera trap images. *EURASIP Journal on Image and Video Processing*, 2013(1):1–10, 2013.

[67] Huiyu Zhou, Yuan Yuan, and Chunmei Shi. Object tracking using sift features and mean shift. *Computer Vision and Image Understanding*, 113(3):345–352, 2009.

[68] Xi Zhou, Kai Yu, Tong Zhang, and Thomas Huang. Image classification using super-vector coding of local image descriptors. In *Computer Vision–ECCV 2010*, pages 141–154. Springer, 2010.