Thesis

Performance evaluation of Local Features for Object Discovery

Submitted by

Jatin V. Bhikadiya

Computer Science Department

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Spring 2015

Master's Committee:

Advisor: Bruce A. Draper
Co-Advisor: Ross J. Beveridge

Daniel J. Bates

Abstract

Performance evaluation of Local Features for Object Discovery

Object recognition is one of the most challenging tasks in computer vision. A common approach in recognizing an object begins by detecting local features in image using a feature detector and describing detected features in terms of feature vectors using a feature descriptor. Many local feature detectors and feature descriptors have been proposed in literature. This work evaluates performance of two successful feature detectors and five feature descriptors on three datasets with unique characteristics. Based on the information content in a given dataset we find general trends on the performance of local features. Our findings will guild computer vision practitioners selecting between alternative local feature detector and local feature descriptor to design highly accurate recognition systems.

Acknowledgements

I would like to thank my advisors, Dr. Bruce Draper and Dr. Ross Beveridge, for all the guidance and support they provided over the years. It has been a great privilege to work with and learn from them. I would also like to thank Dr. Dan Bates for his valuable feedback as my external committee member.

I am grateful to all members of the Visionaries group who have contributed towards my learning experiences. I would, in particular, like to mention, Maggie, Hao, Prady and Rahul, who peacefully shared an office with me for a year and have helped me with their research knowledge and skills. I also want to thank Hrushi, Wimmy, Somdada, Nikhil and Jindal for always being responsive to all my questions.

A special thanks to *'amajing friends'*, Anna, Chacha, Phase, Ajay, Damle, Puntu and Bandu for being such good friends all these years. You all are like a second family to me!

My thanks also goes to many others who have helped me along the way: faculty and staff of the CS and ECE departments at Colorado State University.

Last, but definitely not the least, I want to thank my parents and my family for being my source of comfort and strength.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

Object category recognition categorizes an image based on the presence of a particular object or set of objects in that image. The challenges involved in object recognition are to capture the variations in appearance, shape, viewpoint and texture of different objects belonging to the same object category. People can generalize over instances of object classes even if they have never seen the particular instance before. However, it is difficult for a computer vision system to generalize across different viewpoints of a given object. For example, it is not easy for a computer to learn how an airplane would appear from various angles, hence recognizing a new instance of the same airplane from a different viewpoint is difficult. A common approach for object category recognition is to extract important feature point locations in an image and describe them using feature descriptors that helps to classify the category of the image. Different combinations of image feature point detectors, descriptors and classification techniques have been proposed to solve object category recognition problems. Determining which technique to use for a given dataset is an open problem. This thesis evaluates combinations of feature points detectors [41] [27] and feature descriptors [34] [60] [57] [7] [46] on three datasets [43] [14] [11].

1.1. The Goals

Determining which feature detector and feature descriptor will work best depends upon the type of information in the image data. There is a wide spectrum of datasets whose content varies from localized information to globalized information. In order to measure the impact of this factor on the choice of the feature detector and descriptor, we have used three datasets. The datasets used for this thesis are Flowers [43], Caltech-256 [14] and 15

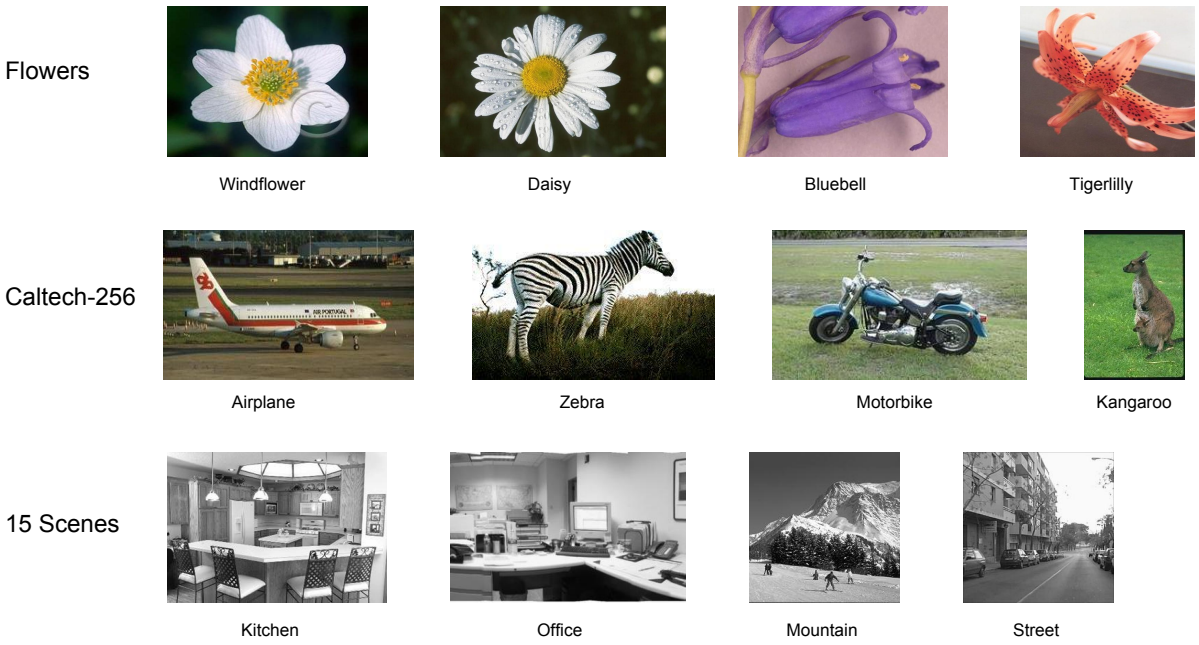


FIGURE 1.1. Image Samples from Three Datasets used in this Thesis. Flowers [43], Caltech-256 [14], 15 Scenes [11].

Scenes [11]. Image samples from these three datasets are shown in Figure 1.1. The first row shows four images from Flowers dataset. The task is not to label images as flower but to sub-categorize them as a specific class of flower e.g. Daisy or Windflower. In the middle row, there are four images from the Caltech-256 object category dataset. Each image has a central object and the image is assigned a label based on that object. The last row illustrates four images from the 15 Scenes dataset. Each image is a collection of a few objects which generalizes the image as a particular scene.

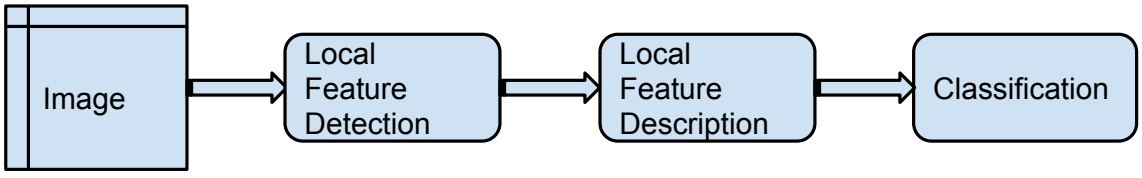


FIGURE 1.2. A High-level Overview of an Object Recognition System

A high level description of an object recognition system is shown in Figure 1.2. First, local feature points are extracted from an image. Local features can be points, edges or blobs. Two commonly used local feature point representations are as follows:

- Interest Points: Interest points are the locations in an image whose local neighbors have high variation in intensity values. The position of interest points can be computed by an interest point detection algorithm [29] [41] [31].
- Grid Points: Grid Points are placed on image at regular or random spacing between them [57] [27]. Unlike interest points, Grid Points don't use image content information to determine location.

Interest points are key locations in an image that may possess invariance to scale, rotation, viewpoint or illumination changes. On the other hand, a regular grid of points offers better coverage over an image with a uniform number of feature points per unit image area. Hessian-Laplace [41] and Grid Points based approaches are selected to evaluate in this thesis. We have selected Hessian-Laplace as a representative of interest point detectors because it outperforms other commonly used interest points like Harris-Laplace, Difference of Gaussian, Salient Regions and Maximally Stable Extremal Regions as per the performance evaluations [39] [3] [55] [58]. Grid Points approach is selected as it is a recent technique that has drawn attention from many researchers [44] [27]. Figure 1.3 shows location of the feature points for both approaches on one image from each dataset mentioned in Figure 1.1.

The difference between both of these feature point selection methods can be noticed in Figure 1.3. Looking at the motorbike and flower images, the Hessian-Laplace detector determines location of feature points on the central object only and there are very few feature points on the background. However, the Grid Points method locates nearly half of the feature

points on background and those points may not be useful to describe the image. For the kitchen scene image, Hessian-Laplace finds most parts of the kitchen but misses cabinet and kitchen roll. However, Grid Points covers all the essential parts to describe the kitchen scene. Hence, the Hessian-Laplace is useful when the background or context is not important to characterize an image and Grid Points can be useful when whole image describes its category.

Another popular approach for image representation is to use global image features instead of extracting local patches. We have also considered a global feature based approach [46] to compare it with the local feature based approach.

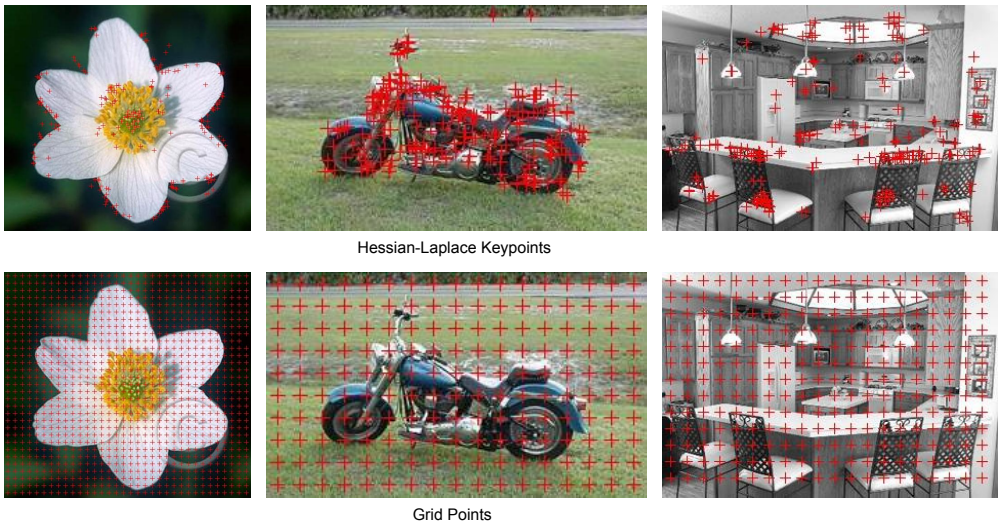
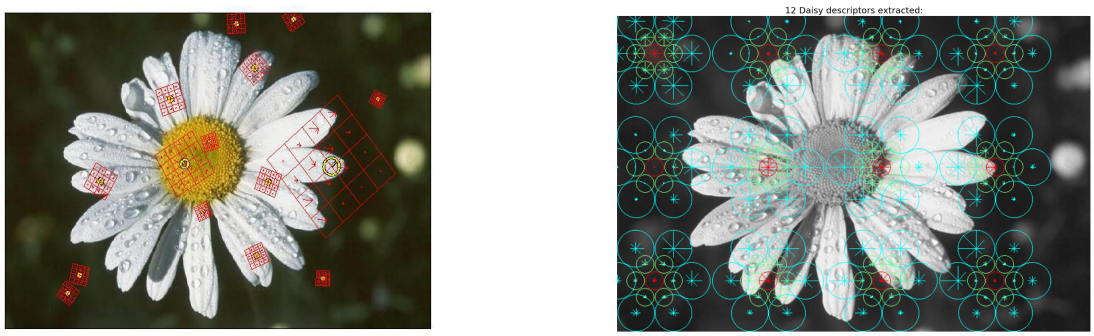


FIGURE 1.3. Visual Comparison of Interest Points on One Image from Each Dataset shown in Figure 1.1.

After extracting feature point locations using Hessian-Laplace or Grid Points over an image, the next task is to represent local image area around those points. Local image regions around feature points can be described by local feature descriptors. Many local feature descriptors have been introduced in the literature [23] [62] [26] [25] [46] [34] in the context of object recognition. Subsequent studies have compared of their performance on different datasets [42] [48][2] [49] [9],

There is little knowledge on how these local feature descriptors perform on the other type of datasets with unique recognition challenges. For this purpose, we have selected a few local feature descriptors which have been shown to perform successfully on a specific type of dataset, and evaluate how well they perform on different type of datasets. This thesis evaluates performance of five local feature descriptors on the selected datasets. These local feature descriptors are as follows:

- **SIFT** - Based on the image gradient orientation histogram in the local area around feature points [34].
- **DAISY** - Similar to the SIFT but extracted over a circular pattern [57].
- **LBP** - A texture descriptor that works by comparing pixel intensity between a central pixel and neighboring pixels [60].
- **HOG** - Counts occurrences of gradient orientation in localized portion and uses overlapping local contrast normalization to improve accuracy [7].
- **GIST** - A global descriptor which captures the "gist" of the image in a low dimensional vector [46].



(A) SIFT Descriptors on Daisy Flower (B) DAISY Descriptor on Daisy Flower

FIGURE 1.4. Visualization of the Feature Descriptors.

Visual representation of SIFT and DAISY descriptors is shown in Figure 1.4. It gives a general idea on how they use a local neighborhood to describe feature points.

1.2. Evaluation Approach

Evaluating the performance of feature points and feature descriptors is a difficult task. The results of classifying images tell us how good a detector and descriptor are describing categories in a dataset. There are two types of classification methods: a) supervised classification and b) unsupervised classification or clustering. Use of a supervised classifier introduces dependency on additional parameters compared to more general unsupervised classification. For supervised classification, the evaluation results will depend on the particular choice of classifier and parameters for that classifier, for example using support vector machine classifier and radial basis function. To avoid this problem the decision is made to use unsupervised clustering method.

Clustering looks at how well images group into categories based upon the choice of image feature detector and feature descriptor. In particular, hierarchical clustering is performed on images. Hierarchical clustering generates many clusters and there is a challenge in deciding on how many clusters to look at. The method taken here examines all clusters and a cluster with the largest concentration of particular category is considered as representative cluster of that category. Average performance of all categories is measured to decide the feature detector and feature descriptor that performs better.

Chapter 2 presents a literature review on existing evaluation approaches for local feature detectors and feature descriptors. Also, it provides a detailed description of selected feature detectors and feature descriptors. Chapter 3 explains our image representation technique and classification technique as well as the evaluation criteria. Implementation details on

evaluation framework are given in Chapter 4. Chapter 4 also presents useful findings of our work in great detail. Finally, Chapter 5 draws conclusion from our results and mentions a few points on future work.

CHAPTER 2

Literature Review

Object classification typically involves three distinct steps, as shown in Figure 2.1. First, a local feature detector selects a set of feature points in an image. The feature detector may use the image information to locate points, or it may use a pre-defined pattern of points [41] [27]. Once the interest points are selected, the next task is to describe the neighborhood around the point using a feature descriptor [34] [60] [57] [7] [46]. Feature descriptors characterize the local visual appearance in terms of a feature vector. In the final step, a classifier assigns a label to the image based on its feature vectors. This can be done using supervised classification or unsupervised clustering.

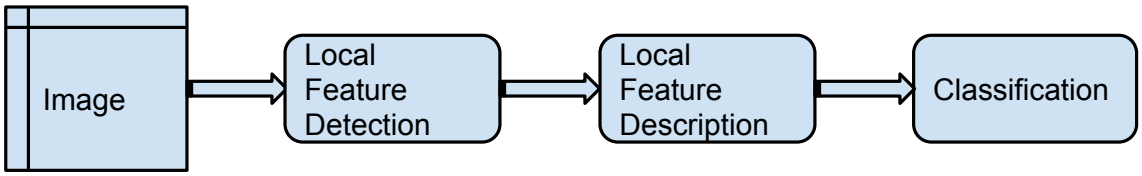


Figure 2.1. A High-level Overview of an Object Classification System

In the past two decades, several feature detectors and feature point descriptors have been proposed [34] [60] [57] [7] [46] [41] [27]. This raises an important question - which feature detector and feature descriptor should we use? As mentioned in Chapter 1, different object class datasets possess different characteristics. For example, the information content in some datasets is local while other datasets contain more global information. The question posed by this thesis is how to pick the best feature detector and descriptor, given a vast dataset of set of object classes.

Previous studies have tried to answer this question by evaluating feature detectors, feature descriptors, or both. For instance, Mikolajczyk et al. [39] compared feature detectors

and descriptors in the framework of an object recognition system. In the following sections, we present the relevant literature of feature detectors and feature descriptors, and also discuss performance evaluation criteria and datasets. Section 2.1 presents feature detectors, performance evaluation measures and previous feature detector evaluations. Section 2.2 does the same for feature descriptors. Section 2.3 explains evaluation frameworks used in the literature and also describes the evaluation framework used in this thesis. Lastly, we present three datasets used in this thesis in Section 2.4.

2.1. Local Feature Detectors

In Chapter 1, the task of detecting feature point locations was defined. Feature points are key locations in an image that should ideally be scale, viewpoint and illumination invariant. Mikolajzyk et al. [58] define six properties of an ideal feature point detector. The most desirable property to evaluate a feature detector is repeatability. It is a measure of the positional stability of a feature point under changes in scale, viewpoint and illumination. Given two images of the same scene from different viewpoints, a good feature detector should find the same feature points in both images. For example, Figure 2.2 shows feature points extracted from two images of the same scene at different scales [58]. The center of the circles are feature point locations and radius of the circle represents the scale used for Gaussian smoothing. Repeatability, the ability to find the same points in both scenes, is one criterion for evaluating feature detectors.

Another approach, and the one taken in this thesis, is to evaluate feature detectors in the context of object recognition. This approach analyzes image clusters instead of individual feature points. Based on the purity of image clusters, they can be compared. Section 3.2.2 presents detailed evaluation criterion for this approach.

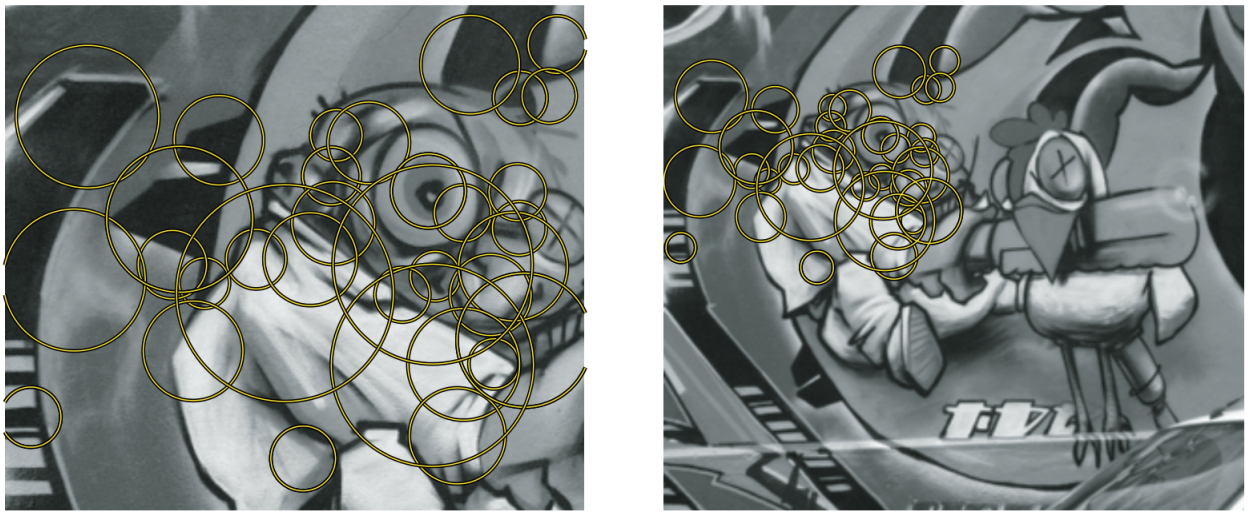


FIGURE 2.2. Hessian-Laplace Detector applied to Images with Change in Scale. This Image is from the Work of Mikolajzyk [58].

Schmid et al. [52] compared local feature detectors [5] [16] [20] [19] [12] in the context of image matching. For performance evaluation, they measured repeatability under changes in scale, rotation and illumination. They concluded that among the detectors available at the time, the Harris detector yielded the best results. It should be noted that their analysis was based on only two scenes.

Mikolajczyk et al. [41] proposed a novel feature detector and compared it with existing detectors [30] [32] based on repeatabilty. They found their proposed detector provided excellent matching results. Subsequently, Mikolajczyk et al. [39] evaluated local feature detectors [22] [36] [33] [41] [40] in the context of object recognition as described above. They analyzed the Harris-Laplace, Difference-of-Gaussian, Hessian-Laplace, Salient Regions and Maximally Scale Extremal Regions(MSER) detector, and reported high performance for Hessian-Laplace keypoints [41] combined with SIFT descriptors [42].

Subsequent studies by Bauml et al. [2], Mikolajczyk et al. [58], Bhatia et al. [3], Lankinen et al. [24] and Stark et al. [55] confirmed that the Hessian based detector outperforms other detectors on various recognition problems. However, their results were limited to specific

recognition frameworks and datasets available at the time. Hence, we have selected the Hessian-Laplace keypoint detector as one option within our evaluation framework.

Chapter 1 described two methods for finding feature point locations. One is to use image information to decide the location of feature points and the other is to lay feature points over the image in a pre-defined pattern. Many researchers have used dense points on a regular grid [27] [57] [44] [11] with promising results. Nowak et al. [44] showed that densely sampled features outperform interest point detectors as a part of a Bag of Features [6] strategy. The winners of recent PASCAL challenges [10] have also used grid points, reinforcing their suitability for object recognition. Based on these references we focus on two methods for finding feature points in images - The Hessian-Laplace detector and Grid Points.

2.1.1. Hessian-Laplace Keypoint Detector [3]. The Hessian-Laplace detector locates interest points on images and defines scales for those points. Hessian matrices are used to find the locations of points, and a Laplacian function computes the scales for those points. A Hessian matrix is composed of the second order partial derivatives of the image I, which can be expressed as:

$$H(X;\sigma_D) = \begin{bmatrix} I_{xx}(X;\sigma_D) & I_{xy}(X;\sigma_D) \\ I_{yx}(X;\sigma_D) & I_{yy}(X;\sigma_D) \end{bmatrix}$$

where I_{xx}, I_{yy} and I_{xy} are second order derivatives of image I computed at point X using Gaussian kernels with standard deviation σ_D.

To find the location of points, a scale-space representation of the image is built by convolving it with Gaussians of increasing standard deviations. The scale of an image within the image pyramid is defined by the standard deviation of the Gaussian used to generate it.

Once we have a scale-space image representation of the image, feature points are extracted at each scale as follows:

(1) Calculate the determinant of the Hessian matrix at each pixel.

(2) Compare the determinant value at a pixel with determinant values at its adjacent pixels in a 3×3 neighborhood.

(3) If the value of the determinant at the current pixel is greater than value of the determinant at its 8 neighboring pixels, and above a given threshold, then there is a feature point associated with the given pixel location. The threshold is used to eliminate points with weak maxima.

The locations of interest points are defined at different levels of the scale-space representation. Because of Gaussian smoothing, the location of an interest point varies according to scale. To find the characteristic scales at which the interest points convey the most information, Mikolajczyk et al. [38] concluded that the Laplacian is optimal for detecting characteristic scales. As we need to compare responses of Laplacian function at different scales, a scale normalized Laplacian function is used which can be expressed as

$$Laplacian(X; \sigma_D) = {\sigma_D}^2 |I_{xx}(X; \sigma_D) + I_{yy}(X; \sigma_D)|$$

where I_{xx} and I_{yy} are second order derivatives of image I computed at point X using Gaussian kernels with standard deviation σ_D.

To select the characteristic scale for feature points detected in a scale-space image, the Laplacian function is calculated over all scales. The scale at which this Laplacian function attains a local maxima is assigned as the characteristic scale. In the case of more than one local maxima, the point is assigned multiple scales.

2.1.2. Grid Points [59] [11]. Vogel et al. [59] used random 10×10 pixel patches to represent scene content resulting in a high classification rate. Following Vogel, Fei-Fei et al. [11] implemented an evenly sampled grid of interest points. They sampled a grid of patches, spaced at 10×10 pixels in an image. The patches were randomly sampled in sizes between 10 to 30 pixels. Subsequently, Lazebnik et al.[27] used 16×16 pixel patches computed over a grid with spacing of 8 pixels to extract SIFT descriptors for each patch. They combined Grid Points with a Spatial Pyramid Matching approach which used spatial information of the image features in the recognition task. Lazebnik et al. partitioned images into smaller and smaller sub-regions. For each sub-region, interest points were sampled at uniform grid points and SIFT feature vectors were computed and histogrammed. The final image representation was the concatenation of these histograms.

In this thesis, we evaluate feature points sampled at the center of 16×16 non-overlapping image patches. Feature descriptors are then extracted at the center of these patches and occurrence histogram of the feature descriptors are computed using Bag of Features approach. The Bag of Feature approach is discussed in Section 3.1.

2.1.3. Hessian-Laplace vs Grid Points. In this section, advantages and disadvantages of Hessian-Laplace and Grid Points over each other are discussed. The Hessian-Laplace detector focuses on regions that can be localized easily and contain high information about an image. Also, it yields a high repeatability rate. On the downside, as it uses image information to find location of the feature points, the number of feature points extracted using Hessian-Laplace detector varies a lot. It can go upto few thousands for a very high contrast image, which makes it difficult to select important feature points. Sometimes, for

low contrast images, it may fail to produce even a single feature point which results in no useful image information.

On the other side, Grid Points yield very low repeatability compared to Hessian-Laplace. Higher repeatability can be achieved by making sampling density extremely high. However, in this case the number of features will grow to be unacceptably large. As a trade-off, overlapped patches of a predefined size can be used which may result in somewhat higher repeatability. On the plus side, dense sampling of points on a regular grid results in good coverage of the entire image and produces a constant number of features per image area. An image or image parts with low and high contrast will contribute equally in this case. This is very useful for scene interpretation tasks where the entire image describes the class of that image.

2.2. Local Feature Descriptors

Once a set of feature points are extracted from an image, the next task is to encode the local area around those feature points as a feature vector. Local feature descriptors generate feature vectors based on the image patch around the feature points. Most feature descriptors strive to be invariant to translation, rotation and scale. In chapter 1, we presented five feature descriptors: SIFT [33], LBP [60], HOG [7], DAISY [56] and GIST [46]. In this section, previous comparisons among feature descriptors are discussed.

It is important to know how to evaluate feature descriptor performance in various recognition frameworks. A popular approach to evaluate descriptor is to use precision-recall criterion. In this approach, first the images in a given dataset are represented in terms of feature vectors. Once we have the feature vectors, each image is assigned a label using a supervised classifier or an unsupervised clustering technique. Based on these assigned labels

and true labels of images, precision is defined as the fraction of retrieved images that are relevant, while recall means the fraction of relevant images that are retrieved. For example if we are trying to label dog images, precision tells us how many images are dog out of the images classified as dog, while recall tells us how many dogs are actually found out of the total number of dogs. The same evaluation terminology can be applied to different recognition frameworks such as Scene Classification, Person Re-identification, Human Detection and Face Recognition. Another criterion to evaluate feature descriptors will be computational efficiency. Many real-time applications require local descriptors that are fast to compute and result in a small length vector which make other tasks, such as classification, faster.

Mikolajczyk et al. [42] evaluated various local feature descriptors for the purpose of image matching using the above mentioned precision-recall criteria. They demonstrated that SIFT based descriptors are superior to other descriptors available at that time, in terms of invariance to rotation, scale and affine transformations. Bauml et al. [2] selected prominent local feature descriptors and evaluated them on the task of person re-indentification. They have shown that SIFT based feature descriptors outperform SURF and Shape Context descriptors. Pinto et al. [48] compared local image descriptors for invariant object recognition tasks. However, they used synthetically generated image data to perform the experiments which can't be generalized to natural datasets. Gil et al. [13] evaluated state-of-the-art descriptors for the problem of visual Simultaneous Localization and Mapping (SLAM). However, their evaluation was focused on matching context and measuring how well similar landmarks in different images of same category are grouped for the different descriptors. They have shown that an extension of SIFT, Gradient Location and Orientation Histogram (GLOH), obtained

the best results. Based on these various performance evaluations, our first choice of descriptor for the evaluation is the SIFT descriptor so that it can be compared with newly available descriptors.

Out of the many other available local feature descriptors, we wanted to test the better performing and most commonly used descriptors. Ren et al. [49] demonstrated that Histogram of Orientation Gradients (HOG), originally introduced for Human Detection [7] [66], outperforms SIFT for the task of object recognition. A dense grid-based version of SIFT, known as DAISY [56], is more robust to geometric and photometric transformations as compared to SIFT [57]. Subsequently, DAISY has been evaluated for object recognition by Zhu et al [65] and has been shown to perform better and faster than SIFT descriptor.

A popular texture based descriptor, Local Binary Pattern (LBP), has been shown to perform excellently for face detection [1] [15]. Cevikalp et al. [4], Heikkila et al. [18] and Satpathy et al. [51] evaluated Local Binary Patterns and its variants for object category recognition and showed that Local Binary Patterns can be useful for invariant object category recognition. Apart from the above mentioned local feature descriptors, a global image representation, known as GIST [46] has been shown to be successful in retrieving relevant images in a large scene dataset [9].

Taking these previous studies into account, we have decided to add Local Binary Patterns (LBP), Histogram of Orientation Gradients (HOG), DAISY and GIST descriptors to our evaluation list. In the following, the descriptors used in our work are discussed briefly.

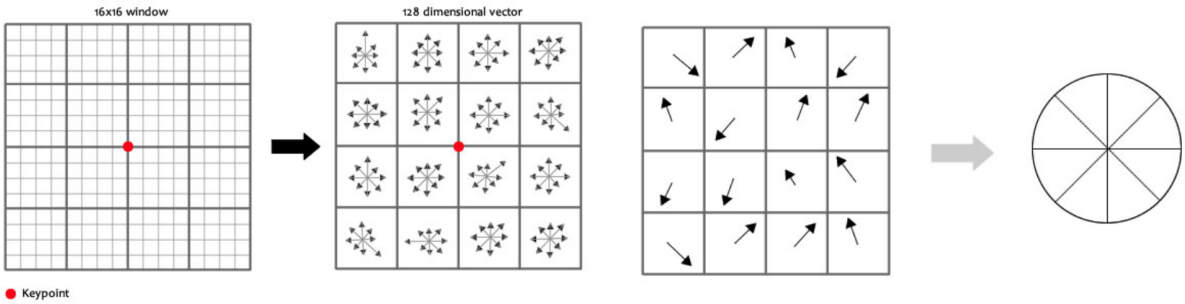
2.2.1. Scale Invariant Feature Transform [34]. The Scale Invariant Feature Transform (SIFT) descriptor is useful to make interest point description partially invariant to illumination and viewpoint changes [33]. Construction of the SIFT descriptors is as follows:

(1) For each interest point, consider the 16×16 pixel region centered at interest point and compute the gradient orientation at each pixel in that region.

(2) Divide this 16×16 region into 16 sub-regions such that each sub-region is of size 4×4.

(3) For each sub-region, construct an 8-bin histogram of gradient orientations. Gradient orientation at a pixel varies between 0 to 360. So gradient orientation between 0 to 44 gets added to the first bin, between 45 to 89 gets added to the second bin and so on. Also, the amount added to bins is weighted by a Gaussian function.

(4) Finally, the 8-bin histograms of all 16 sub-regions are concatenated to construct the SIFT descriptor of length 128. The illustration of SIFT descriptor construction is shown in Figure 2.3.



(A) Gradient Orientation (B) 8-bin Histogram Generation

Figure 2.3. An Illustration of SIFT Descriptor Construction. This Figure is from the Work of Lowe et al. [33]

Following this description construction, two more post-processing steps are performed to make it less sensitive to illumination changes. First, for the linear illumination invariance or uniform contrast, the 128 length vector is normalized to unit length. Second, to reduce the effect of large gradient magnitudes resulting from non-linear sources like camera saturation, any descriptor elements higher than 0.2 are cut off to 0.2 and descriptor is re-normalized to unit length.

2.2.2. Local Binary Patterns [60]. Texture in an image has two important aspects, a pattern and its strength or contrast. For any descriptor these two properties are an interesting pair to get rotation and illumination invariance. Rotation affects spatial pattern of the texture but not contrast while illumination affects contrast but not rotation. Based in these assumptions Ojala et al. [45] introduced basic Local Binary Patterns(LBP) descriptor which is capable of separating the texture's pattern from contrast information.

The Local Binary Patterns operator is usually applied to the gray scale image and works on a 3×3 pixel block. An illustration of the LBP extraction is shown in Figure 2.4. It compares the gray-scale value of the center pixel with gray-scale values of its neighboring pixels to generate a bit code. If the neighbor pixel has a larger value than the center pixel then it generates 1. It generates 0 if the center pixel has larger value than the neighbor pixel. For a 3×3 block, the 8 neighbors of the center can be represented with the 8-bit integer value which is assigned to the center pixel. This 8 bit label is assigned to all the pixels in an image and these values are histogrammed to 256 bins to represent the image texture.

Increasing the size of the neighborhood increases the length of feature vector exponentially as 2^P, where P is the number of neighbors. For example, if the block size is 5×5, length of the feature vector will be $2^{24} = 16777216$. It is computationally inefficient to classify the

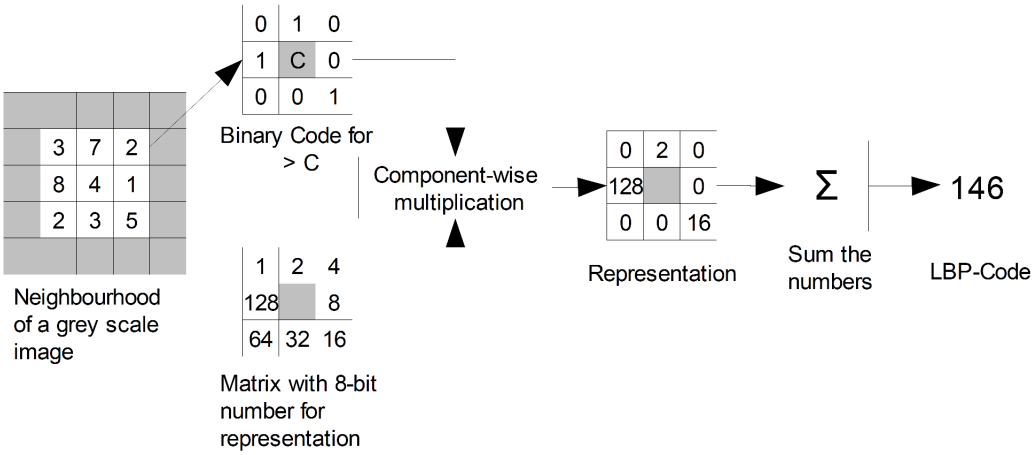


FIGURE 2.4. Illustaration of Local Binary Pattern Construction. This Figure is from the Work of Lindahl et al. [28].

images with such long feature vectors. To reduce the length of the feature vector, a uniform binary pattern is used. The uniform binary patterns also exploits the fact that there are at most two transitions from one to zero and zero to one in the majority of the LBP codes [45].

In a Uniform LBP the number of transitions from 0 to 1 and 1 to 0 are counted from the bit code generated with the basic LBP. Based on the number of transitions, a new label is assigned to the center pixel. If the numbers of transitions in the bit pattern are less than 2, the pattern is known as uniform. For example, the patterns 11111111 (0 transitions) and 00110000 (2 transition) are uniform patterns whereas the patterns 01101111 (3 transitions) and 01010100 (6 transition) are non uniform patterns. In uniform LBP there is a separate label for each uniform pattern and all the non-uniform patterns are assigned a single label. Thus, there are a total of 58 uniform patterns for the 8 bit LBP as shown in Appendix A. Considering one label for all the non-uniform patterns, a 59 dimensional histogram is extracted for an image as a feature vector.

There are two main reasons for omitting non-uniform patterns. Ojala et al. [45] showed that, in their experiments with texture images, uniform patterns account for nearly 90% of

all patterns when using 8-bit LBP. Another reason to consider uniform patterns that they are relatively more stable. Considering only uniform patterns reduces the number of labels and makes it more reliable for better classification.

2.2.3. Histogram of Orientation Gradients [7]. Histogram of Orientation Gradients(HOG) is similar to the gradient based descriptor SIFT where descriptor extraction starts by calculating gradient orientation at each pixel. The original HOG was developed for human detection and used a patch size of 64×128. However we have used a patch size of 16×16 to extract descriptors at the local feature points. The patch is then divided into a grid of cells and blocks for the normalization purpose. We have selected cell size as 4×4 and one cell per block, resulting in a total of 16 blocks. For each cell, the gradient orientations are histogrammed into 8 bins in the range of 0 to 180 degrees. The magnitude of the gradient vector determines the contributions added to the histogram. The next step is block normalization which divides histogram for each cell by magnitude of the vector. Lastly, block normalized histograms are concatenated to produce a 128 element feature descriptor.

2.2.4. DAISY [56]. The DAISY descriptor is a histogram of gradient orientations extracted on densely sampled feature points in an image. DAISY descriptor is similar to SIFT descriptor but with two major differences:

- DAISY uses circular neighborhood instead of rectangle neighborhood used in SIFT descriptor.
- SIFT descriptor is a histogram of gradient orientations weighted by the Gaussians while DAISY uses convolution of gradient in a specific direction with several Gaussian filters.

To compute DAISY descriptors, a certain number of orientation maps G_o, one for each quantized direction o, are first computed. Convolved orientation maps are obtained by convolving each orientation map with Gaussian kernel of different σ. Computation time of DAISY is reduced by obtaining large Gaussian kernel from several small consecutive kernels.

Next, the neighborhood around each pixel is divided into circles on a series of rings centered at a given pixel. At each circle, a vector is made by gathering the values of all the convolved orientation maps with corresponding Gaussian smoothing. The final DAISY descriptor is extracted by concatenating all the vectors from circles, after they are normalized to unit form.

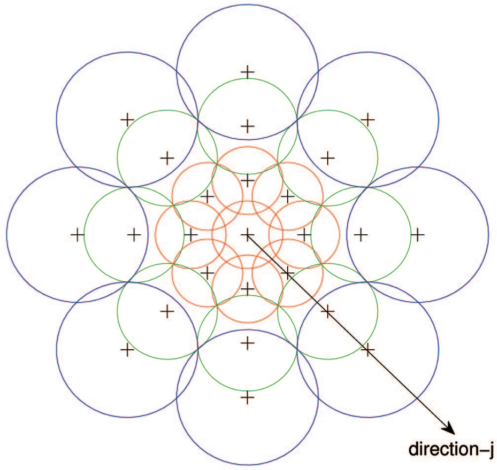


FIGURE 2.5. Illustration of DAISY Construction. This Figure is from the Work of Tola et al. [57].

2.2.5. GIST [46]. The GIST feature is a global descriptor which characterizes important statistics about a scene [46]. The idea behind GIST features is to capture a set of perceptual dimensions like naturalness, openness, roughness, expansion and ruggedness, that represent the dominant spatial structure of a scene. The GIST feature is computed by convolving an oriented filter with the given image at several scales and orientations. This will measure

the high-frequency and low-frequency repetitive gradient directions of an image. The scores for the filter convolutions at each scale and orientation are used to calculate the final GIST descriptor. In this work, GIST descriptors are calculated for gray scale image using a filters at 8 orientations and 4 scales. This way the final descriptor of length 960 will be extracted for an entire image.

2.3. Evaluation Approach

Performance evaluation of local feature detectors and descriptors is a challenging task. Mikolaczyk et al. [40] evaluated feature detectors by matching a reference image with the deformed image. Various deformations such as change in scale, blur and lighting condition were present in the deformed images. The detectors were evaluated by their repeatability ratios and total number of correspondences for the different viewpoints and deformations of an image. Performance evaluation was done by comparing how well the detector can cope with deformations. Similar approach was followed by Schmid et al. [52] for image matching, Bauml et al. [2] for person re-identification problem and Bhatia [3] for feature matching. For local feature descriptors, a similar evaluation approach was proposed by Mikolaczyk et al. [42], where feature descriptors were matched on the object location. Later, Zhang et al. [64] proposed comparison criteria for various detectors and descriptors using a mid-level image representation method known as Bag of Features. Instead of evaluating detectors and descriptors individually, combinations of the detector-descriptor were compared. They showed that to achieve the best possible performance, it is necessary to use detectors and descriptors in combination with a classifier.

We followed the above mentioned approach and compared the combinations of detector-descriptor using the Bag of Features image representation. Feature detector-descriptor can

be evaluated by comparing the classification results. In this thesis, we have used unsupervised hierarchical clustering to classify or group the images into the categories. This approach is chosen because of the simplicity and less number of input parameter compared to a supervised classifier. Section 3.2 discusses specific details on unsupervised clustering technique and evaluation criterion.

2.4. Dataset Selection

In this work we are evaluating two local feature detection approaches, Hessian-Laplace and Grid Points. Furthermore, we are also evaluating five feature descriptors including four local feature descriptors, SIFT, LBP, HOG and DAISY, and one global feature descriptor, GIST. However, unlike previous comparison studies discussed in the above sections, our goal is to eliminate dataset bias from our results. To elaborate, we will be using three datasets which vary in terms of the information contained in their image categories. These datasets are selected based on their inter-class and intra-class variability. From previous works, the most popular dataset for the performance evaluation on object category recognition is Caltech-101 [55] [65] [39] [24]. However, in this dataset objects are of similar size and orientation and lack in rich background. This means that, it has a very low intra-class variability as shown in the Figure 2.6. Because of these limitations of Caltech-101, we have selected Caltech-256 [14], which is a very diverse dataset with 256 natural object categories containing high intra-class variability and rich backgrounds.

To select other datasets, there is a wide range of available datasets from those with very little inter-class variability to those having high amount of inter-class and intra-class variability. We have selected Flowers dataset [43], with a very fine distinction between the

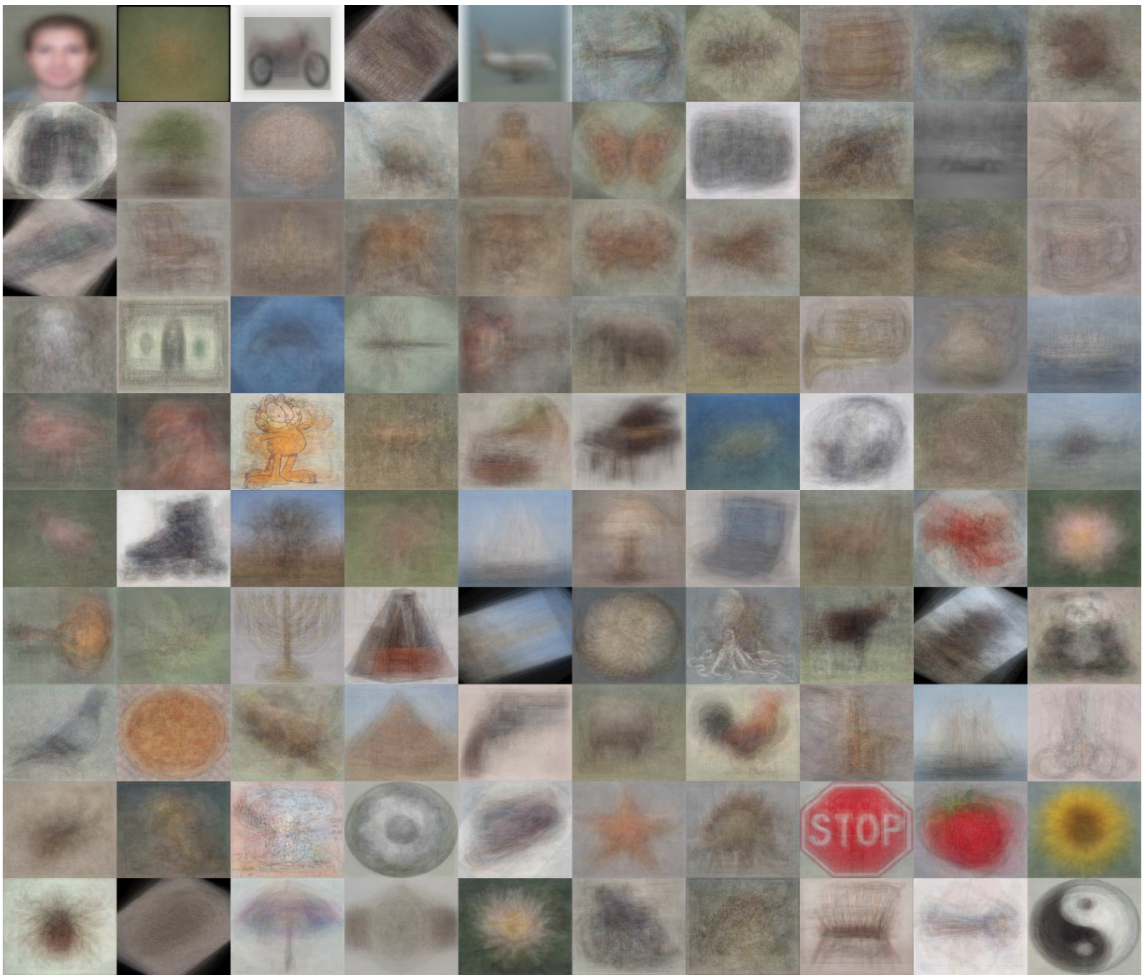


FIGURE 2.6. The Caltech-101 average image. This image is from the work by Zhang et al. [63]

categories, and 15 Scenes datasets [11], where each category is generalized by the set of objects. A brief overview of the datasets used in our work is given in the Section 3.3.

This literature survey took us through various performance evaluation techniques used to compare detectors and descriptors. It also provided us with a stable direction to pursue our aim of updating the literature in object recognition with precise and reliable results for independent and combined performances of detectors and descriptors. Our approach differs from these previous works because we attempt to establish our results on more challenging datasets which have diverse information content.

CHAPTER 3

Methodology

The previous literature reviewed on the feature detectors and descriptors along with ways to evaluate their performance. This chapter presents the methodology chosen here to perform a series of experiments and evaluate the performance of feature detectors and descriptors. As discussed in Section 1.1, object recognition is a three step process: a) Localize points on images using a feature detector, b) Describe localized points using a feature descriptor and c) Label the images using a classifier. To combine the first two steps, the Bag of Features [6] approach is used that converts an image into a single feature vector. Once all images in a dataset are represented in terms of feature vectors, unsupervised clustering is used to assign labels to them. The performance of these feature detectors and descriptors is measured and compared using a statistical measure, the F measure, is used. Section 3.1 describes Bag of Features approach for image representation. Section 3.2 talks about clustering and evaluation. Lastly, detailed information on datasets is presented in Section 3.3.

3.1. Image Representation

This section explains how an image is represented in terms of a vector which contains meaningful information about that image. This representation helps to compare an image with other images and measuring similarity. Bag of Features [6] technique is selected for this representation. There are two main reason for selecting this technique: a) The Bag of Features technique is very popular now-a-days owing to its good performance and simplicity. b) It is an orderless collection of local features extracted using a feature detector and described using a feature descriptor. These two steps of extracting and describing features are

independent of each other which makes it easier to compare local feature detectors and local feature descriptors. Also, as this approach discards spatial information, it is theoretically easy to understand and efficient to compute.

The Bag of Features approach is analogous the to Bag of Words [17] approach used in textual information retrieval. In Bag of Words, each document is represented as a normalized histogram of word counts. As a first step, a dictionary is created using set of different words obtained by merging all text documents of a collection. For each document, the frequency of word occurrence in the document is calculated for all words in the dictionary. So each document is represented as a sparse vector in which each element is a term(word) from the dictionary and the value of that element is the frequency of occurrence of that term in the document. This histogram is then divided by the total number of dictionary words in the document. The Bag of Words approach is order-less because ordering of words in the document has been lost. Similar to this, Bag of Features is used to represent images in terms of vectors. In the following subsections, two essential steps, dictionary creation and histogram representation, for the Bag of Features approach are explained.

3.1.1. Bag of Features Vocabulary Construction. The Bag of Features representation, similar to Bag of Words in textual information retrieval, can be used for the task of object recognition. In this technique, documents are replaced by images and words are replaced by local feature vectors. A dictionary, also known as a visual vocabulary, can be constructed using these local feature vectors . As shown in Figure 3.1, local features are localized in images using a feature point detector. These localized feature points are then described using a feature descriptor. The next step is to create a visual vocabulary using these feature descriptors.

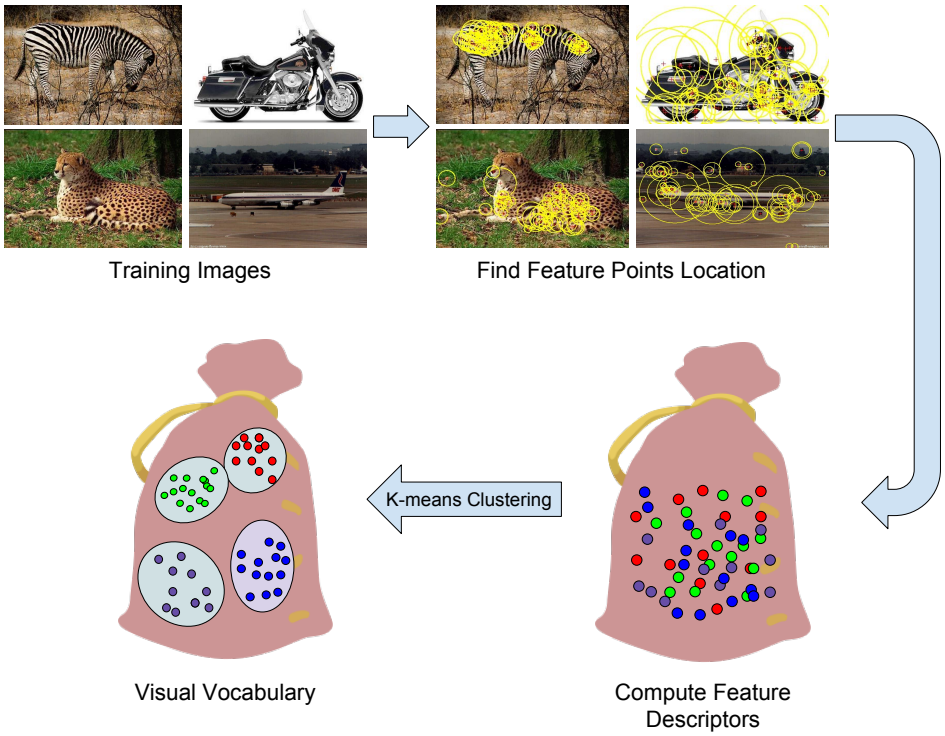


FIGURE 3.1. Bag of Features Vocabulary Construction

In the Bag of Words approach, only one word is used as a representative for a set of similar words while creating dictionary. For example, words like "messages", "message" and "messaging" will be treated as a same word. Likewise, similar feature descriptors should be grouped such that each group represents a local area of an object. To group similar feature descriptors K-means[35] algorithm is used. K-means generates clusters of similar feature descriptors, and average of all feature descriptors in a cluster can be used as a representative of feature descriptors in that cluster. K-means algorithm works as follows:

Inputs to K-means are a set of descriptors and the value of k, where k is the number of clusters to be generated. K-means clustering generate centers of k clusters as output. These cluster centers are the words of the visual vocabulary. This visual vocabulary will be used in the next step to extract Bag of Features histogram for any query image.

Algorithm 1 K-means for Bag of Features

1: Extract the feature descriptors of all images and be the set of data points $X = \{x_1, x_2, ..., x_n\}$
2: Randomly select a set of k cluster centers $C = \{c_1, c_2...c_k\}$
3: **for all** Data point **do**
4: Calculate the Euclidean distance between data point and cluster centers.
5: Data point is assigned to the cluster with minimum Euclidean distance.
6: **end for**
7: Recalculate new cluster centers using :

$$c_i = \frac{1}{m} \sum_{j=1}^{m} x_j$$

where m represents number of data points in cluster c_i.
8: Repeat steps 3 to 6 until no data point is reassigned to different cluster.

3.1.2. Bag of Features Histogram Representation. Once a visual vocabulary is constructed using a set of images from all categories in a dataset, any image of those categories can be converted into a k-length histogram vector using the visual vocabulary. K is the number of clusters used as a input parameter to K-means algorithm. Figure 3.2 shows steps to represent an image in terms of a histogram vector. Feature points are located in a given image and feature descriptors are computed at those points. The feature detector and feature descriptor should be the same as the ones used while creating the visual vocabulary. Next, each feature descriptor is assigned to a nearest cluster center in the visual vocabulary using a nearest neighbor algorithm. A K-length histogram of those feature descriptors' count is then extracted to represent that image. As number of feature descriptors varies per image, the final histogram is normalized by dividing it with the number of feature descriptors. Once images are represented in terms of Bag of Features histogram, they can be assigned a label using a classifier. The classification approach used is discussed in Section 3.2.

3.1.3. Parameters for Bag of Features. One of the limitations of the K-means algorithm is that it does not determine the value of k. We ran a few experiments to determine

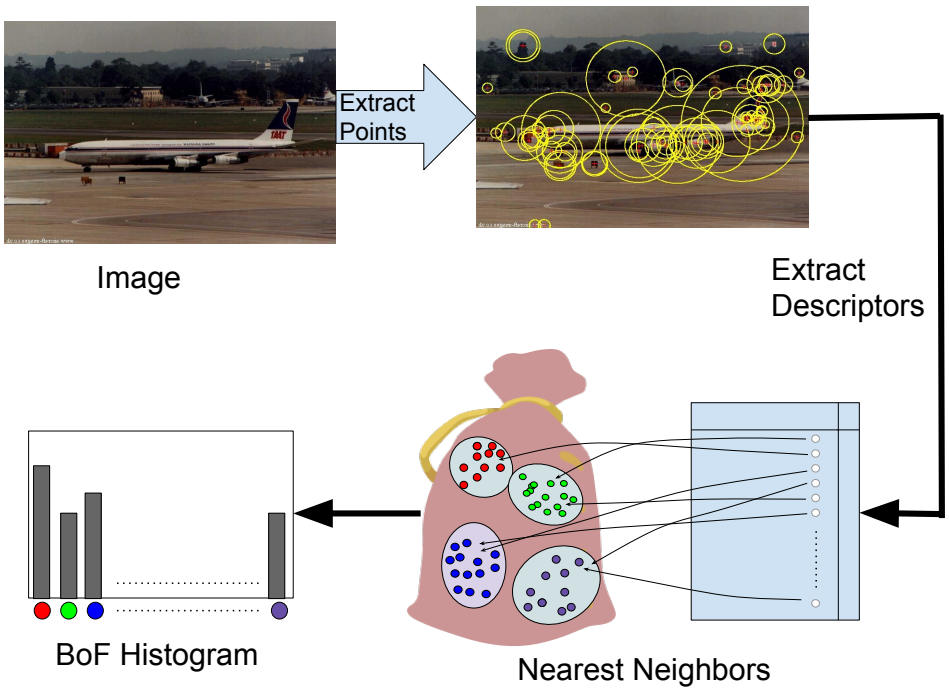


FIGURE 3.2. Bag of Features Vector Construction

the value of k for a given number of feature descriptors. Also, the value of k need not be perfectly optimized as the goal of this thesis is to compare performance on categorization, rather than comparing Bag of Features technique with other image representation techniques.

Some other parameters to consider while creating the visual vocabulary are the number of images per class and number of descriptors per image. Only a few images of each category should be sufficient to characterize that particular category for the Bag of Features approach. For example, if the categories are car and zebra, a few images of both classes will be enough to generate clusters of local parts such as wheels, headlights, zebra legs and nose. A feature detector may detect large number of feature points. However, it is computationally inefficient to use all feature descriptors extracted at the feature points for k-means clustering. For these reasons, we have selected a limited number of images per class and feature points per image.

Specific implementation details about the number of images, number of features per image and value of k are given in Section 4.1.

Note: For global descriptor GIST, image representation is done by directly extracting the GIST feature descriptor on images. One important property of the GIST descriptors is to use spatial information for image representation. In contrast, the Bag of Features approach discards spatial information.

3.2. Unsupervised Clustering

The previous section discussed on how to represent an image in terms of Bag of Features histogram. Using this representation, each image can be assigned a label using a supervised classifier, or images can be grouped into clusters of images using unsupervised clustering. We are following the second approach of unsupervised clustering because of its simplicity, speed and fewer number of input parameters.

Clustering is a task of grouping data points, images in this thesis, in such a way that images in the same group are more similar to each other than to those in other groups. These groups of images are called clusters. There are two families of clustering algorithms, Flat Clustering and Hierarchical Clustering, depending on how they work. Flat Clustering partitions given data such that all groups are independent of each other. The K-means algorithm, discussed in 3.1.1, is an example of Flat Clustering. Hierarchical Clustering partitions data into a hierarchy of clusters that can be visualized using a structure known as dendrogram. Agglomerative Clustering and Divisive Clustering are two types of Hierarchical Clustering. Agglomerative is bottom-up clustering which starts with each image as its own cluster. At each iteration, it merges two most similar clusters until all the images are merged into one cluster. On the other hand, Divisive Clustering is a top-down approach and starts

with all images in one cluster. At each iteration it splits the cluster into two or more clusters until all the images are singleton clusters. A dendogram generated using Agglomerative Hierarchical clustering is shown in Figure 3.3.

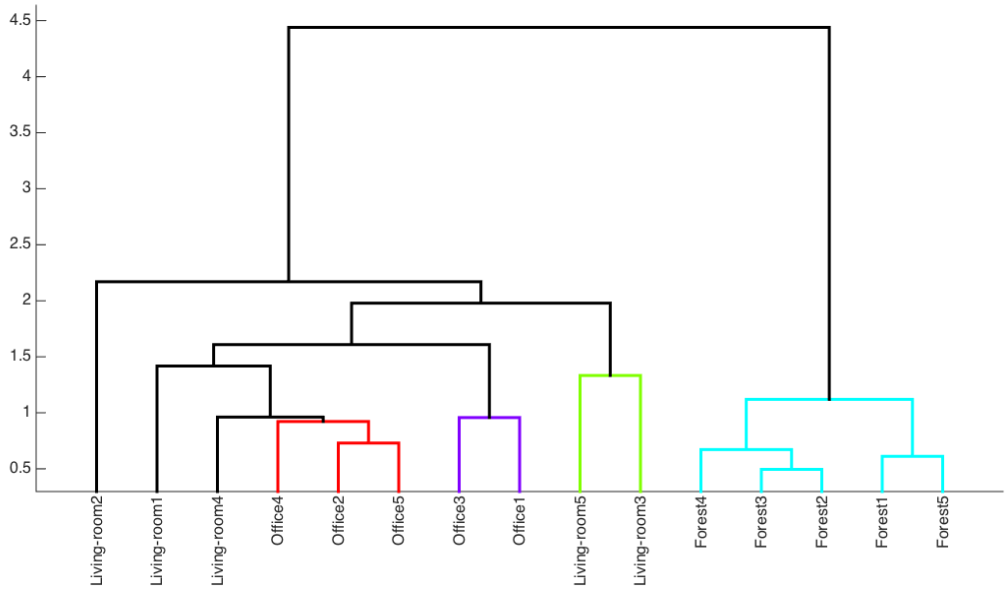


FIGURE 3.3. Example of Dendrogram.

Various partitions can be achieved using Hierarchical Clustering. Because of this advantage of Hierarchical clustering over flat clustering, where only one partition of data is possible, we have selected Hierarchical Clustering for classification. Now the question arises as to which Hierarchical Clustering should be used, Agglomerative or Divisive. Divisive Hierarchical Clustering starts with all images in a single cluster and splits the images into two or more clusters at each iteration. To split a cluster, the global distribution of the data should be known. For example, if the task is to segment an image by clustering pixel values, we can find good splits by constructing histogram of RGB values. However, for classification it is a bit difficult to find this kind of global distribution of data and decide the split. So to make our approach simpler, we choose Agglomerative Hierarchical Clustering.

3.2.1. Agglomerative Hierarchical Clustering. Agglomerative Hierarchical Clustering is a bottom-up approach where each image is in its own cluster at the beginning, and iteratively all images are merged into a single cluster. This approach generates a hierarchy of clusters known as dendrogram. The basic algorithm for Agglomerative Hierarchical Clustering is as follows: Step 2 in the above algorithm is to find inter-similarity matrix

Algorithm 2 Agglomerative Hierarchical Clustering

1: Assign each image to a cluster of its own
2: Compute the inter-similarity matrix between all cluster
3: Merge the most similar pair of clusters, say i and j, to form a new cluster k
4: Update similarity matrix by removing entries for i and j and adding a entry for k
5: Go to step (3) until all of the clusters are merged into one cluster

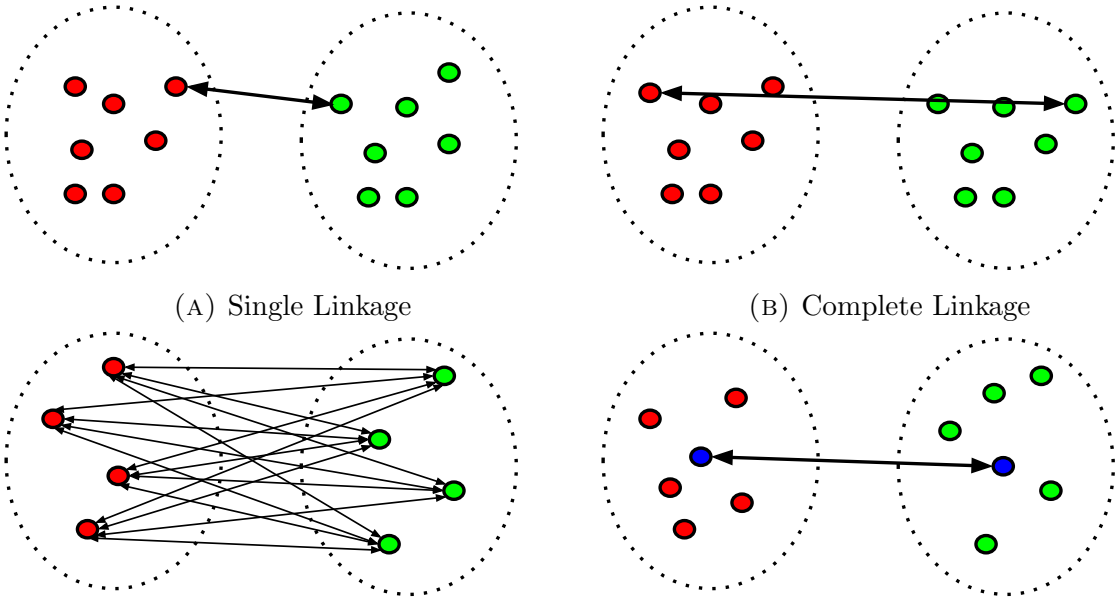
between clusters. There are two cases while finding similarity between two clusters: a) Both the clusters are singleton clusters having only one image in them. Similarity between two singleton clusters is the similarity between images in those clusters. b) If even one of them is not a singleton cluster, linkage criteria between them decides similarity between them. In following sections, five different linkage criteria are defined and differences between four of them are visually shown in Figure 3.4.

3.2.1.1. *Single Linkage [53].* In the single linkage method, distance between two clusters is the minimum distance between any image in the first cluster and any image in the second image. It is defined as:

$$d_{AB} = \min_{a \in A, b \in B} d(a, b)$$

where a and b belong to cluster A and B respectively.

3.2.1.2. *Complete Linkage [8].* This method is contrary to the single linkage criterion since it uses the pair of images which are least similar or at maximum distance to each other



(A) Single Linkage (B) Complete Linkage

(C) Average Linkage (D) Centroid Linkage

FIGURE 3.4. Linkage critria to merge two clusters in Agglomerative Hierarchical Clustering

in two clusters. It can be expressed as

$$d_{AB} = \max_{a \in A, b \in B} d(a, b)$$

where a and b belong to cluster A and B respectively.

3.2.1.3. *Average Linkage [54].* The average linkage method calculates distance between all pairs in two clusters, and averages these clusters. The equation to find the distance between two cluster is as:

$$d_{AB} = \frac{1}{m * n} \sum_{i=1}^{m} \sum_{j=1}^{n} d(a_i, b_j)$$

where a and b belong to cluster A and B respectively, m is the size of cluster A and n is the size of cluster B.

3.2.1.4. *Centroid Linkage [21].* Centroid linkage defines distance between two clusters as the distance between centroids of those clusters.

$$d_{AB} = d(\overline{a}, \overline{b})$$

where $\overline{a}$ and $\overline{b}$ are centroids of the A and B respectively.

3.2.1.5. *Ward's Method [61].* Ward's method does not define the distance between two clusters. However, it says that the distance between them is proportional to the increase in their variance when we merge those two clusters. If clusters A and B are merged into cluster C, then the increase in their variance while will be defined as :

$$\Delta\sigma = \sum_{i=1}^{m+n} \|c_i - \overline{c}\|^2 - \sum_{j=1}^{m} \|a_i - \overline{a}\|^2 - \sum_{k=1}^{n} \|b_k - \overline{b}\|^2$$

where m is the size of cluster A, n is the size of cluster B, $\overline{a}$ is the mean of cluster A, $\overline{b}$ is the mean of cluster B and $\overline{c}$ is the mean of merged cluster C. $\Delta\sigma$ is known as merging cost, and two clusters with minimum merging cost are merged at each iteration.

3.2.2. Interpretation and Evaluation of Dendrograms. The result of Agglomerative Hierarchical Clustering generates dendgrograms which represents a hierarchy of clusters. Figure 3.5 shows a dendrogram generated using Agglomerative Hierarchical Clustering on 15 images of three categories from the 15 Scenes dataset.

As shown in Figure 3.5, image sample labels are plotted against their distance at which they are merged into clusters. For this example, we use Ward's method for linkage between clusters and euclidean distance for similarity between image samples. At the first iteration, leaf nodes with images Forest2 and Forest3 are merged into one cluster. At second iteration,

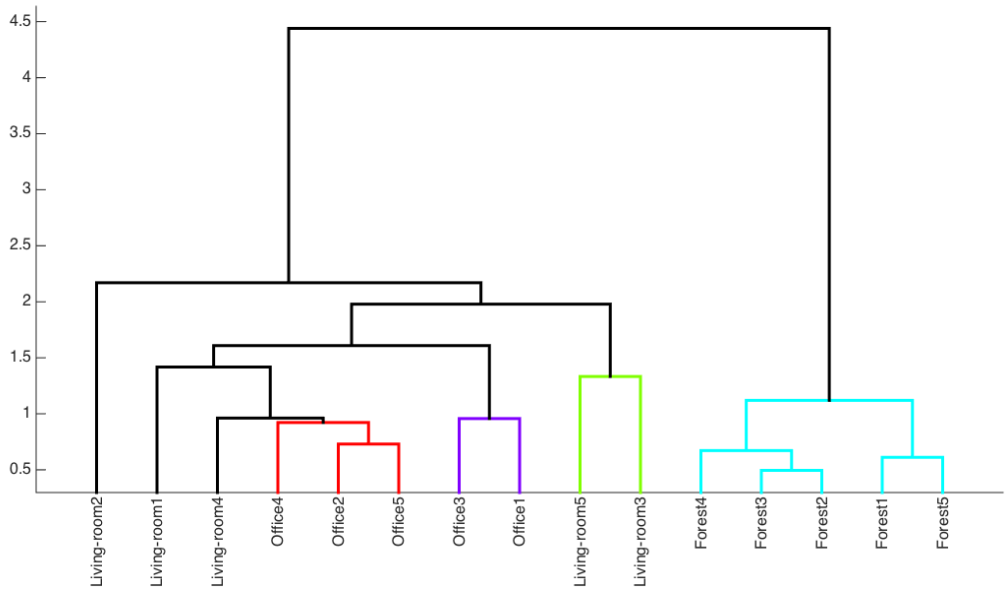


FIGURE 3.5. Dendrogram Generated using Agglomerative Hierarchical Clustering with Matlab [37]. Each Data Point is a GIST Descriptor of Images shown in Appendix B. Images are from Three Scene Categories in the 15-Scenes Dataset [11].

cluster which contains Forest4 is merged with cluster containing images Forest2 and Forest3. Looking at the left side of the dendgorgram, Office1 and Office3 are merging into a cluster, while Office2, Office4 and Office5 are merging into one cluster. However, the cluster containing Office2, Office4 and Office5 merges with Livingroom1 and Livingroom4 successively. This is because both the categories, Office and Living-room, are indoor scenes and can be easily confused with each other. The images used to generate this dendrogram are shown in Appendix B. If categories are totally separable then the dedrogram will look different.

Figure 3.6, shows dendrogram generated for Forest, Living-Room and MITCoast categories. It can be observed that three pure clusters are formed first and merged into one at root level.

In Figure 3.5 and Figure 3.6, it should be noticed that hierarchical clustering produces large number of clusters. Not all of these clusters contain useful groups and specific clusters

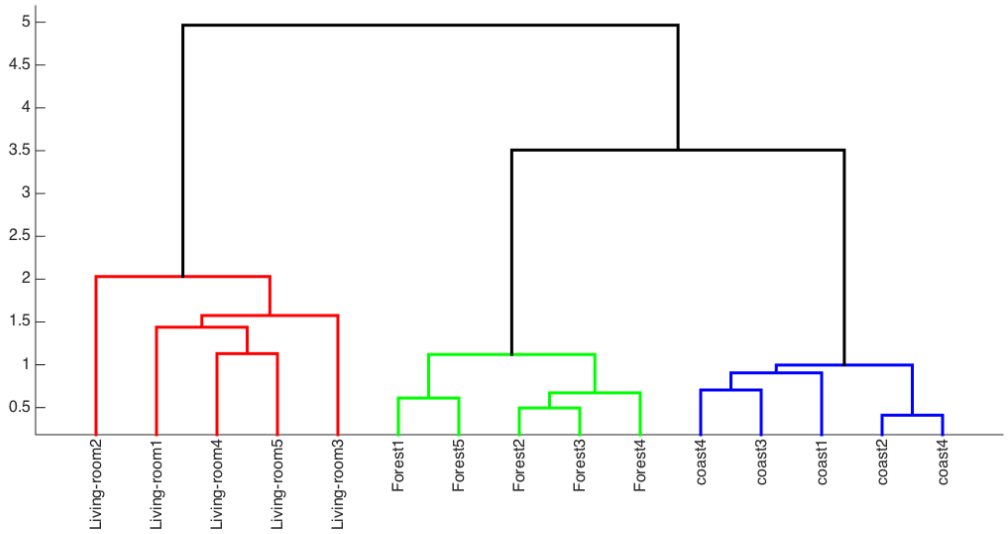


FIGURE 3.6. Dendrogram Generated for Categories Forest, Living-Room and MITCoast [11].

should be selected from the dendrogram. Common method to select clusters is to cut the dendrogram at a constant height. For example, cutting dendrogram shown in Figure 3.6 at height 2.5 will result in three clusters. These three clusters are pure and each of them contain 5 images from one class. However, for the dendrogram in Figure 3.5, such a cut will not produce pure clusters. A question arises on how many clusters are really important and how to select those clusters. In this thesis, we are going to select only one cluster for each category. So total number of selected clusters will be equal to the number of categories in a given dataset.

For each category, all the clusters are ranked by asking how well each cluster performs at discovering that category. The best cluster among all the clusters will be selected as a representative of that category. To select the dominant cluster, we are using F-measure which is a harmonic mean of precision and recall. Before going for the formal definitions of these terms, let's assume the given task is to discover the category dog in Caltech-256 [14]which has 100 images of dogs in it. So the goal is to find a single cluster, that has the maximum

numbers of dog images and the minimum numbers of non-dog images. Consider a cluster in dendrogram with total 50 images out of which 35 are dog images and 15 are non-dog images. Now, precision is the number of correct results divided by the number of all returned results. Recall is number of correct results divided by total number of labeled images of that category. Thus for the above mentioned cluster and category dog, precision will be $35/(35+15) = 0.7$ and recall will be $35/100 = 0.35$. F measure, the harmonic mean of precision and recall, will be $(2*0.7*0.5)/(0.7+0.35) = 0.47$. For the dog category, the F measure for all the clusters will be calculated, and the clusters with highest F measure will be assigned to it. This cluster is not removed from the dendrogram as it may be assigned to a different category or it can be the child of the clusters which is assigned to a different category.

The criteria to select a cluster from dendrogram for any category can be formally defined as follows:

Notations:

- H is a cluster tree or a dendrogram
- L_j is the object category j and its set of images
- C_i is the cluster contained by H and its set of images
- P is Precision, R is Recall and F is F measure

Given a cluster C_i and a category L_j, the precision and recall can be written as :

(1) $$P(C_i, L_j) = \frac{|C_i \cap L_j|}{C_i}$$

(2) $$R(C_i, L_j) = \frac{|C_i \cap L_j|}{L_j}$$

Using Equation 1 and 2, F measure for the cluster C_i and the category L_j, can be written as :

$$F(C_i, L_j) = \frac{2 * P(C_i, L_j) * R(C_i, L_j)}{P(C_i, L_j) + R(C_i, L_j)} \tag{3}$$

For the category L_j, a cluster with the highest F measure is obtained according to equation:

$$F(L_j) = \max_{C_i \in H} F(C_i, L_j) \tag{4}$$

The F measure value ranges from 0 to 1. The higher the F measure, better is the clustering. It indicates that the feature detector and feature descriptor used to represent images are doing a better job of describing images. This approach of selecting one cluster for each category using F measure addresses the following concerns:

- It ignores clusters with either higher precision or higher recall as compared to the cluster that gets selected. F measure tries to maximize precision and recall instead of selecting tiny pure clusters at the bottom level or large clusters at the top level of the tree.
- Even if there is more than one cluster which represent a category with a similar F measure, only one will get selected and others will be rejected. However, if these clusters fall under a same parent, this approach will select the parent cluster for that category.

Based on the F measure of the cluster selected for each category, our experiments can be evaluated by averaging F measures over all the categories. Section 4.1 discusses more about experimental evaluations.

3.3. Datasets

Every available dataset is limited to specific visual appearance. For example, Caltech-101 is an object class dataset in which each image contains only single object. To include more variety, we have selected three well known datasets, Caltech-256 [14], 15-Scenes [11] and Flowers [43], for experimental purpose. Each of these datasets is chosen because of its unique characteristics. Summary of these datasets is as follows:

3.3.1. Caltech-256. Caltech-256 is an object dataset with 257 classes, 256 object classes and a clutter class, totaling 30607 images. There are minimum 80 images per class. Each of these classes has one central object in it which describes that particular class. It is a highly complex dataset because of very high inter-class and intra-class similarity. Taxonomy and average images of all classes are shown in Appendix C.

3.3.2. 15-Scenes. 15-Scenes [11] is one of the most diverse scenes dataset available in literature. It contains 10 outdoor classes and 5 indoor classes. The number of images in each class varies in between 200 to 400 and average image size is 300×250 pixels. Example images from all 15 categories are shown in Appendix C.2.

3.3.3. Flowers. Flowers [43] datasets has 17 classes of different species of flowers. Each flower subcategory has 80 images, totaling 1380 images. As all the classes are species of flowers, they have very low inter-class similarity. Each categoriy has very high intra-class similarity due to large illumination, viewpoint and scale variations in images. Appendix C.3 shows example images from this dataset.

CHAPTER 4

Results

A series of experiments are performed to achieve our goal of comparing the performance of feature detectors and feature descriptors on three datasets. This chapter presents implementation details on experiments followed by a few useful findings. Implementation details are discussed in Section 4.1. Section 4.2 compares feature detectors' performance and deduces the type of information content supported by each detector. Pair wise performance of detectors and descriptors is analyzed in section 4.3. Next, category level performance is discusses in Section 4.4. Lastly, Section 4.5 shows how the GIST descriptor design is highly influenced by the 15-Scene dataset.

4.1. Implementation Details

In this thesis, two local feature detectors and five feature descriptors are evaluated on three datasets. Out of these five feature descriptors, four are local descriptors and one is a global descriptor. There will be 8 combinations of 2 local feature detectors $\times$ 4 local feature descriptors + 1 global feature descriptor to evaluate on each dataset. Each of these 9 combinations is run 12 times on each dataset by randomly selecting training and testing sets.

To represent images using local feature detectors and local feature descriptors, we use the Bag of Feature approach. For 8 of the above mentioned 9 experiments that include local features, we need to create a visual vocabulary as a first step towards the Bag of Features. The following choices are made while creating a visual vocabulary:

- Number of categories: All the 15 and 17 categories are used for 15-Scenes and Flowers datasets respectively. For Caltech-256, 20 categories are selected randomly for each run.
- Number of images per category: 50 images for Caltech-256, 60 images for Flowers and 65 images for 15-Scenes per category are selected randomly as training set. The remaining images from each category are added to the testing set.
- Number of feature points: We compare two methods for locating feature points in an image:

  A) Hessian-Laplace detector may detect hundreds or thousands of feature points in an image, depending on the image content. We limit the Hessian-Laplace operator to 250 feature points per image in these experiments. If fewer feature points are extracted, we use all of them.

  B) For Grid Points, we place feature points at the center of 16×16 non overlapping image patches. So number of feature points varies according to image size. However, we resize the image to keep number of feature points around 250.
- Size of the visual vocabulary: For each experiment, there are 1000 images and approximately 250 feature descriptors per image. So there are roughly 250,000 descriptor to cluster. We group them into 4000 clusters.

After obtaining a visual vocabulary, the images in the testing set are converted into 4000 dimensional feature vectors. This feature vector represents an image as a order-less histogram of local image parts. For GIST descriptors, each image in the testing set is directly converted into a 960 dimensional global feature vector (The Bag of Feature step isn't required for GIST).

These feature vectors are grouped using Agglomerative Hierarchical Clustering producing a dendrogram. The most representative cluster for each category is identified from the dendrogram using the F measure. Finally, the F score of the dendrogram is the overall sum of weighted F measure of all the categories.

$$F_H = \sum_{j=1}^{n} \frac{L_j}{N} F(L_j) \tag{5}$$

where n is number of categories in dataset, N is total number of images in training set and L_j is the number of images in category j.

4.2. Feature Detectors' Performance Comparison

The performance of the Hessian-Laplace detector is compared to Grid Points using the F measure for the best selected clusters. For each dataset, 12 experiments are performed and the F measure weighted across object categories for each experiment is calculated using Equation 5. The overall performance is then calculated by averaging the weighted F measure over all 12 experiments. Figure 4.1 compares performance between Hessian-Laplace and Grid Points when used with the four local feature descriptors. This plot doesn't contain results for the GIST descriptor, as GIST is a global descriptor that does not need to localize feature points. The Y axis on the bar plot is the delta (difference) value between the average F measure for Grid Points and the average F measure of Hessian Points. The X axis is the label combination of one of the datasets and one of the local descriptors. From the bar plot in Figure 4.1, one can conclude that:

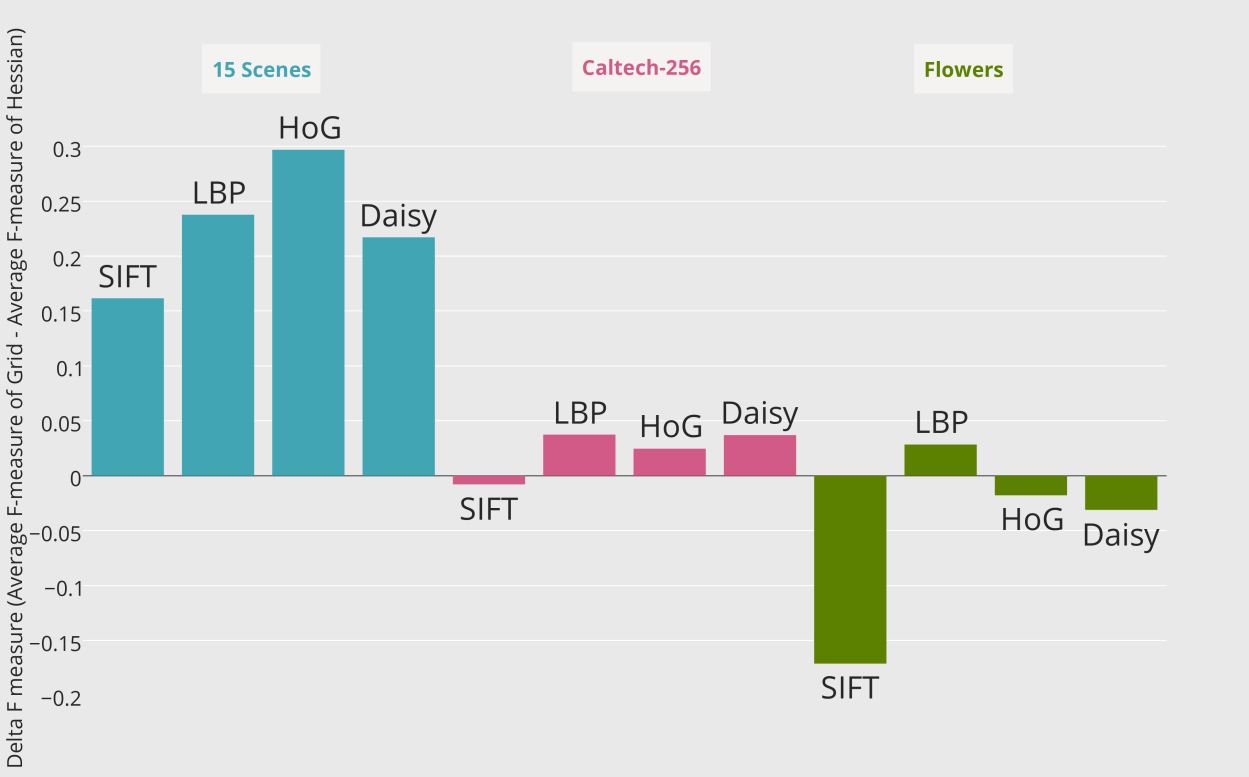


FIGURE 4.1. Delta F measure - Difference between Hessian-Laplace and Grid Points F measure.

(1) Grid Points outperform the Hessian-Laplace detector on the 15 Scenes dataset. This behavior is not dependent on the choice of the feature descriptor. However, the change in F measure varies between 0.15 to 0.3 for different local feature descriptors.

(2) The Hessian-Laplace detector outperforms Grid Points on the Flowers dataset by a large difference in F measure when used in conjunction with SIFT descriptor. It performed poorly on the 15 Scenes dataset.

(3) The other bars on the plot, which have an absolute difference in F measure less than 0.05, should be considered insignificant.

It appears that Grid Points on 15 Scenes performs well because they cover the whole image area. The images in 15 Scene dataset, as shown in Appendix C.2, have more global content. In other words, the images are defined by more than one objects or object parts.

For example, images in the kitchen scene are defined by objects such as microwaves, stoves, tables, cabinets and refrigerators. On the contrary, the images in the Caltech-256 and Flowers datasets are more localized, with single dominant object that much of the image area. Because of this, Grid Points failed to perform well on these datasets.

Hessian-Laplace performs poorly on the 15-Scenes dataset. It selects feature points in image where there is a high variation in intensity values. Thus it may select few points and miss some other important image locations. Incidentally, on the flowers dataset Hessian-Laplace's performs better as it successfully finds feature points on important flower parts such as petal, sepal and pistil.

4.3. Feature Detector-Descriptor Performance

In this section we will discuss about how the different detector-descriptor combinations perform on the Caltech-256, 15 Scene and Flowers. The feature detector-descriptor pair performance is shown in Figure 4.2. On the bar plot, the X axis is the labels for the experiments and the Y axis is value of weighted average F measure for all the experiments.

The first 8 bars on bar plot shows results for the 15-Scene dataset. It can be observed that Grid Points with the HOG outperforms other detector-descriptor combinations on the 15-Scenes dataset. Grid Points combined with LBP is second highest performing combination. The HOG descriptor is less sensitive to the illumination change and 15-Scenes dataset categories have very high variation in terms of illumination. Also, the outdoor categories in 15-Scenes dataset contain texture information that is supported by the LBP descriptor. On the contrast, the Hessian-Laplace performs worst with all the feature descriptors. This is because it fails to find useful feature point location on 15-Scene images.

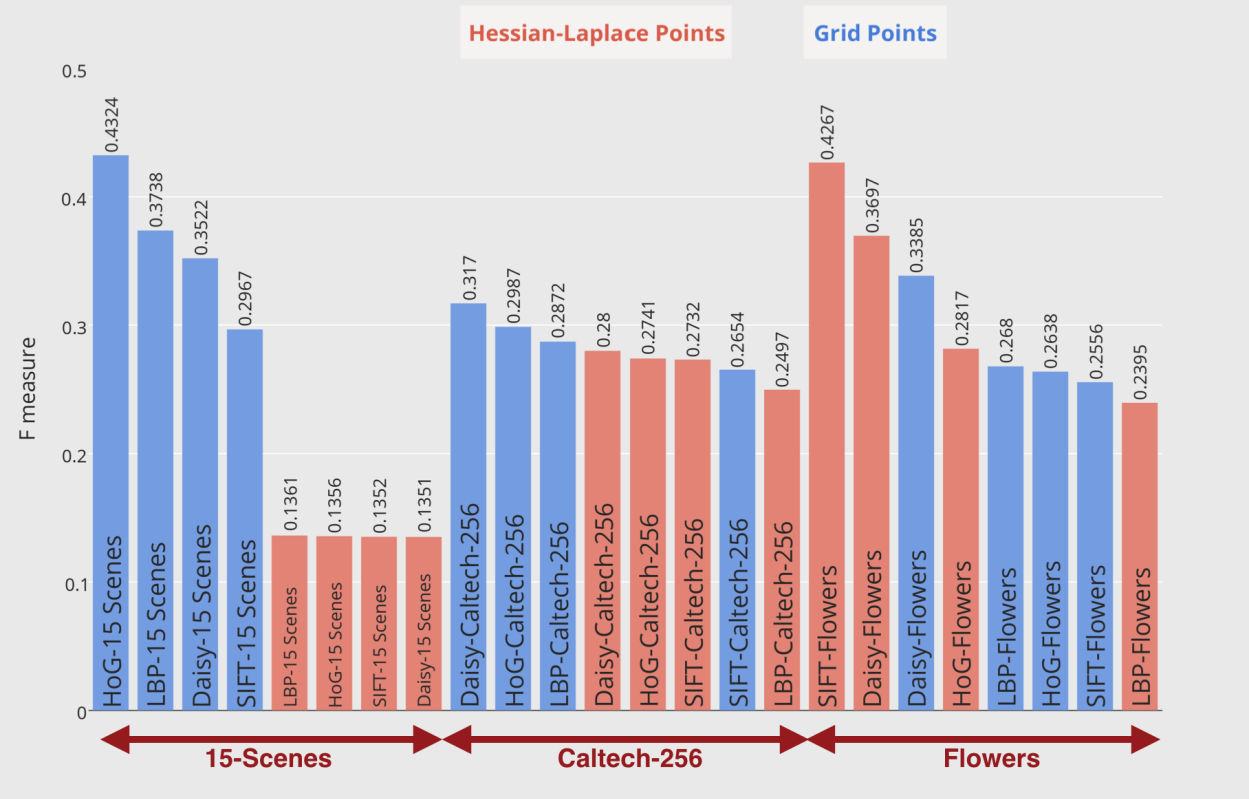


FIGURE 4.2. Feature Detector-Descriptor Performance

For the Caltech-256 dataset, there is no feature detector-descriptor combination that outperforms other combinations by a significant amount of the F measure. Caltech-256 has very high inter-class and intra-class variability. It contains the categories supported by individual combination but overall there is no better performing detector-descriptor combination. The category level results for the Caltech-256 is discussed in the the Section 4.4.

From the last 8 bars, it appears that the Hessian-Laplace with SIFT descriptor outperforms other combinations on Flowers dataset. Hessian-Laplace with DAISY descriptor is second best performing combination on the Flowers dataset. Hessian-Laplace finds feature points location on image where there is a very high intensity variation in local area. The implementation of the Hessian-Laplace keypoint detector is very similar to the SIFT keypoint detection technique. The SIFT descriptors computed over SIFT keypoints yields

high recognition rate for the object categories. [33]. We also see the similar performance with DAISY descriptor as DAISY is similar to SIFT descriptors except that it is faster to compute DAISY.

4.4. Category level Localization information finding

To evaluate feature detectors at category level and see what kind of information content particular detector is supporting, we have defined a *localization score* for each category. We displayed the Hessian-Laplace keypoint locations on each image in Caltech-256 and Flower dataset. Looking at particular images, we observed that not all the Hessian-Laplace keypoints are on the object of interest (object which defines the label) in an image. There are images in which many keypoints are on the background. This happens because of multiple reasons such as low image contrast, very rich background or the object of interest is out of focus. If the Hessian-Laplace finds majority number of feature points on object of interest in image, we call that image a localized image. And if most of the feature points are not on the object of interest in an image then we call that image a non-localized image. Some examples of non-localized images from Caltech-256 and Flowers dataset are shown in Figure 4.3.

We hand labeled all the images in the Caltech-256 and Flowers dataset as either localized image or non-localized image. Using those labels, for each category in both the datasets, *localization score* is calculated as:

$$localization\ score = \frac{No.\ of\ localized\ images\ in\ the\ category}{Total\ number\ of\ images\ in\ the\ category} \tag{6}$$

We hypothesize that if all the images in the category are localized such that most of the Hessian-Laplace points are on the object of interest, Hessian-Laplace detector should perform

FIGURE 4.3. Non-Localized Image Samples from the Caltech-256 and Flowers Datsets.

better compared to Grid Points. If more than half of the images are not localized then Hessian-Laplace will perform similar or worse that Grid Points. So higher the localization score better the performance of the Hessian-Laplace compared to Grid Points.

4.4.1. Localization Results on the Flowers Dataset. To analyze our hypothesis, we calculated the F measure ratio for Hessian-Laplace to Grid Points for each category as per following:

$$F\ measure\ Ratio = \frac{F(HL)}{F(HL) + F(GP)} \tag{7}$$

where F(HL) is F measure for Hessian-Laplace and F(GP) is F measure for Grid Points.

Scatter plot in Figure 4.4 shows the relation between the *localization score* and the F measure ratio for SIFT descriptors on individual categories from Flowers dataset. It can be noticed that higher the *localization score* better is the F measure ratio. We calculate the correlation coefficient and draw the regression line between data points on scatter plot. Both, correlation coefficient and regression line strongly support our hypothesis. Similar plots for other local descriptors are shown in Appendix D.2.

4.4.2. Localization Results on the Caltech-256 Dataset. For each category in Caltech-256 dataset, we calculated localization score using Equation 6. We also observed in Section 4.2 that neither Hessian-Laplace nor Grid Points outperforms on Caltech-256 dataset. However, if we analyze their performance at category level on Caltech-256, we find few categories on which each of the detector performs better compared to the other one. Those categories also support our hypothesis on relation between F measure ratio and *localization score*.

Table 4.1 shows sample categories on which Hessian-Laplace outperforms Grid Points and vice versa. Values in the 2^{nd} to 5^{th} column are difference in F measure between Hessian-Laplace and Grid Points for the particular category. It is shown that higher the *localization*

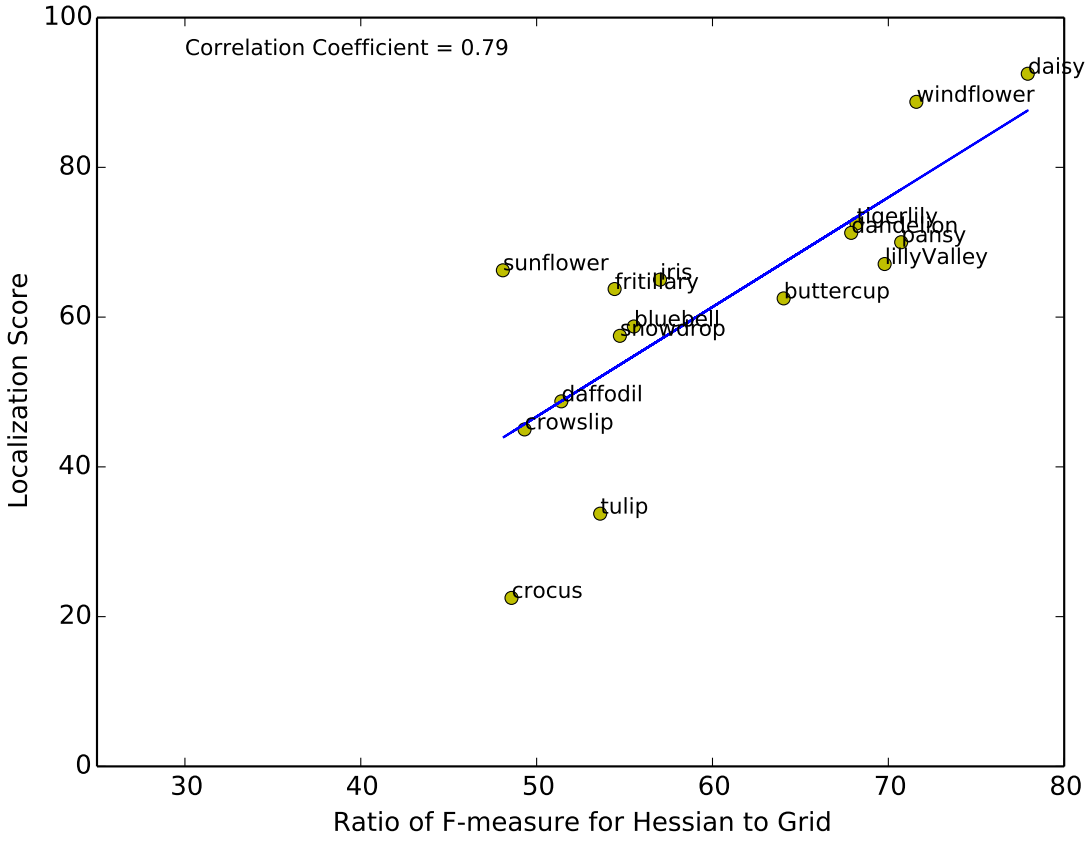


FIGURE 4.4. Localization Score vs F measure ratio between Hessian and Grid for SIFT Descriptors on Flowers Dataset

score better the performance of Hessian-Laplace compared to Grid Points. If the *localization score* is lower, Grid Points performs better than Hessian-Laplace indicated by negative numbers in row 5 to 10. More such categories are shown in Appendix D.3

4.5. PERFORMANCE OF GIST DESCRIPTOR

To summarize the performance of GIST descriptor on three datasets, Caltech-256, 15 Scenes and Flowers, a scatter plot is generated and shown in Figure 4.5. In this scatter plot, the Y axis is the value of the F measure averaged over 12 experiments and the X axis is the dataset labels. For each dataset in the scatter plot, there are five descriptor labels that show

		F(HL) - F(GP)			
	Localization Score	SIFT	Daisy	HoG	LBP
bonsai-101	82.79	0.13	0.04	0.13	0.01
zebra	76.04	0.42	0.07	0.22	0.20
mountain-bike	84.15	0.26	0.12	0.05	-0.03
ice-cream-cone	60.23	0.06	0.04	0.06	0.06
car-side-101	7.76	-0.18	-0.18	-0.26	-0.28
elephant-101	44.27	-0.11	-0.04	-0.17	-0.10
goldfish	27.96	-0.02	-0.09	-0.07	-0.10
iris	34.26	-0.14	-0.14	-0.29	-0.17
comet	27.27	-0.24	-0.42	-0.34	-0.20
kayak	31.07	-0.08	-0.17	-0.27	-0.09

TABLE 4.1. Category Level Localization Results on Caltech-256. F(HP) is F measure for Hessian-Laplace and F(GP) is F measure for Grid Points

their performance (F measure values). In the case of local descriptors, the F measure for the highest performing detector, Hessian-Laplace or Grid Points, is shown on plot. For all the experiments we have used the experimental setup described in Section 4.1. The values of the F measure in the scatter plot are shown in Table D.3. It can be noticed from the scatter plot that the GIST descriptor outperforms other local descriptors on the 15-Scene dataset. However, the GIST descriptor performs worst on the Caltech-256 and Flowers datasets. One should take a moment to think about this behavior of the GIST descriptor. The GIST descriptors was introduced to characterize important statistics about a scene and was shown to perform best on 15-Scene datasets [46]. If spatial location of the objects and background for an object category remains similar across images, GIST does a good job of describing it. On the other hand, local feature descriptors perform better on categories where image information varies a lot with changes in viewpoint, scale and background content. This trend can be clearly noticed on the performance of local descriptors for Caltech-256 and

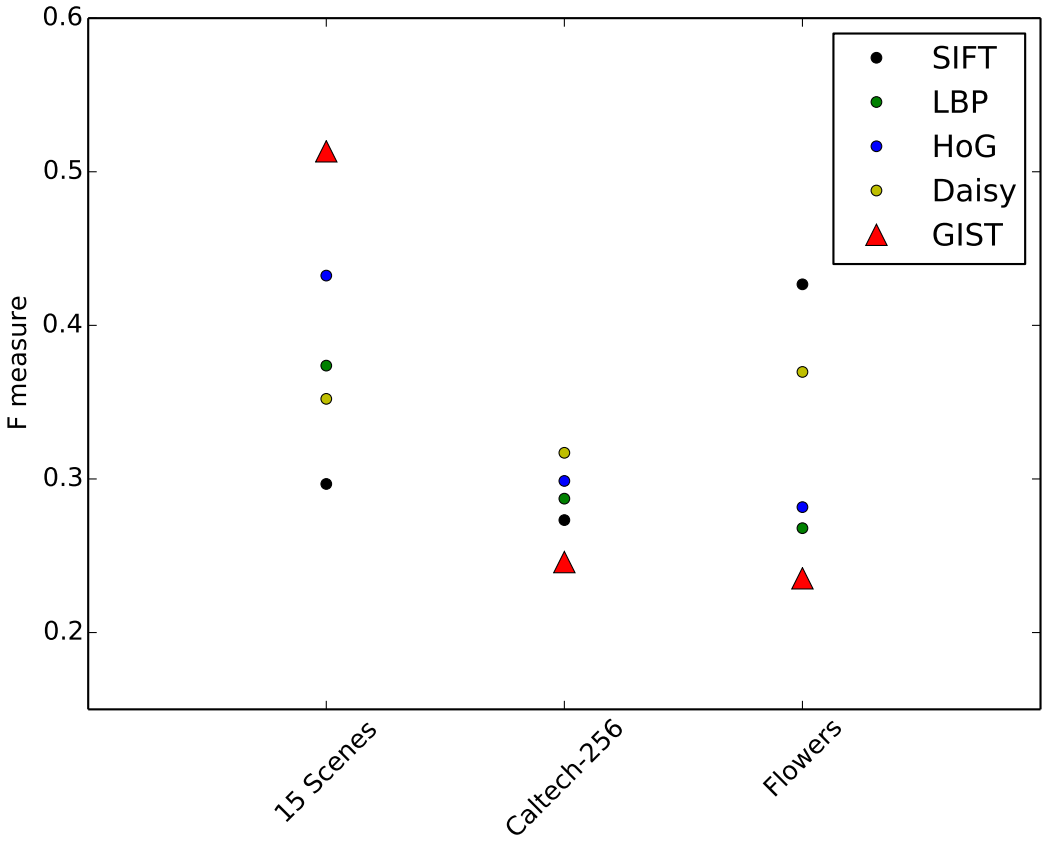


FIGURE 4.5. Datsets influenced perfromance of GIST

Flowers dataset. However, there are few categories in Caltech-256 dataset where the GIST has shown it's dominance because of the scene like spatial structure of those categories. These categories are highlighted in the table of category level performance for Caltech-256 in Appendix D.1.

CHAPTER 5

Conclusion

This thesis compared the performance of two local feature detectors, four feature descriptors and a global feature descriptor on the task of object recognition using the Bag of Features technique. Experiments were conducted on three challenging datasets chosen to evaluate their suitability on different object recognition problems. The contribution of this thesis is that we provide the guidance to future object recognition developers as to which feature detector and descriptor they should use, if they know property of the images in the dataset.

In order to select the feature detector, one should look at the content of the images and see, if they contain more of local or global information. If an image information content is more localized, the quality of the image should be analyzed such as the presence of background with sharp edges, low image contrast or if the object of interest is out of focus. As per our experimental results, we concluded that: a) Grid Points should be selected if the categories are more globalized or defined by a set of objects. Grid Point are also successful when the categories are localized but images in them have focused background content b) The Hessian-Laplace should be selected if the categories are more localized and object of interest in them is highly focused with majority of the sharp edges on it.

We have shown evidences for above two claims at the level of datasets and at the level of category. As shown in Figure 4.1, Grid Points supports the 15 Scenes dataset which contains global categories and the Hessian-Laplace best supports Flowers dataset which is highly localized with little background information. As Caltech-256 is a diverse dataset, it contains both localized and globalized categories, the evidence for detectors' support to

individual categories is shown in Table 4.1. As shown in Figure 4.4, category level analysis on Flowers revealed that more the localized the categories are, the better the performance of the Hessian-Laplace compared to Grid Points.

For the feature descriptors, we observed that HOG and LBP descriptors support natural scene like categories when used with Grid Points. If the categories in an application require high insensitivity to illumination and have more texture information, HOG or LBP can be selected. On the other hand, SIFT and DAISY support localized object categories when used with Hessian-Laplace detector. If the categories have high scale and rotation variation, SIFT or DAISY can be useful. Figure 4.2 shows the influence of local feature descriptors to the content of the categories. The performance of the global descriptor GIST is optimized for distributed scenes in a specific spatial layout. This is not surprising, since it was introduced for the 15-Scenes dataset.

To conclude this thesis, there is no single detector and descriptor that performs best on all different recognition challenges. If one has the knowledge of the information content in the categories, our results will help him/her select the feature detector and feature descriptor to achieve higher classification rate. The results of this work will be useful for the computer vision practitioner to design recognition system with higher accuracy.

As the future work, we plan to analyze feature descriptors performance at the category level for all three datasets. In specific we will look for the image information content supported by each descriptor. Also, we will divide 15 Scenes dataset into two main categories as indoor scenes and outdoor scenes. We hope that this type of categorization will help identify feature detectors and descriptors influenced towards those categories.

Bibliography

[1] T. Ahonen, A Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(12):2037–2041, Dec 2006. ISSN 0162-8828. doi: 10.1109/TPAMI.2006.244.

[2] M. Bauml and R. Stiefelhagen. Evaluation of local features for person re-identification in image sequences. In *Advanced Video and Signal-Based Surveillance (AVSS), 2011 8th IEEE International Conference on*, pages 291–296, Aug 2011. doi: 10.1109/AVSS.2011.6027339.

[3] A Bhatia, R. Laganiere, and G. Roth. Performance evaluation of scale-interpolated hessian-laplace and haar descriptors for feature matching. In *Image Analysis and Processing, 2007. ICIAP 2007. 14th International Conference on*, pages 61–66, Sept 2007. doi: 10.1109/ICIAP.2007.4362758.

[4] H. Cevikalp, Z. Kurt, and AO. Onarcan. Return of the king: The fourier transform based descriptor for visual object classification. In *Signal Processing and Communications Applications Conference (SIU), 2013 21st*, pages 1–4, April 2013. doi: 10.1109/SIU.2013.6531160.

[5] J.C. Cottier. Extraction et appariements robustes des points dinte ret de deux images non e talonne es. Technical Report MSU-CSE-00-2, LIFIA IMAG INRIA Rhone-Alpes, 1994.

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

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893 vol. 1, June 2005. doi: 10.1109/CVPR.2005.177.

[8] D. Defays. An efficient algorithm for a complete link method. *Comput. J.*, 20(4):364–366, 1977. URL `http://dblp.uni-trier.de/db/journals/cj/cj20.html#Defays77`.

[9] Matthijs Douze, Hervé Jégou, Harsimrat Sandhawalia, Laurent Amsaleg, and Cordelia Schmid. Evaluation of gist descriptors for web-scale image search. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, CIVR '09, pages 19:1–19:8, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-480-5. doi: 10.1145/1646396.1646421. URL `http://doi.acm.org/10.1145/1646396.1646421`.

[10] Mark Everingham, S.M.Ali Eslami, Luc Van Gool, ChristopherK.I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, pages 1–39, 2014. ISSN 0920-5691. doi: 10.1007/s11263-014-0733-5. URL `http://dx.doi.org/10.1007/s11263-014-0733-5`.

[11] L. Fei-Fei and P. 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 vol. 2, June 2005. doi: 10.1109/CVPR.2005.16.

[12] Wolfgang Förstner. A framework for low level feature extraction. In *Proceedings of the Third European Conference on Computer Vision (Vol. II)*, ECCV '94, pages 383–394, Secaucus, NJ, USA, 1994. Springer-Verlag New York, Inc. ISBN 0-387-57957-5. URL `http://dl.acm.org/citation.cfm?id=200241.200283`.

[13] Arturo Gil, OscarMartinez Mozos, Monica Ballesta, and Oscar Reinoso. A comparative evaluation of interest point detectors and local descriptors for visual slam. *Machine Vision and Applications*, 21(6):905–920, 2010. ISSN 0932-8092. doi: 10.1007/s00138-009-0195-x. URL http://dx.doi.org/10.1007/s00138-009-0195-x.

[14] G. Griffin, A. Holub, and P. Perona. The caltech-256. Technical report, California Institute of Technology, 2007. URL http://citeseerx.ist.psu.edu/showciting;jsessionid=13B3DB196FF5048696E9AB45F4F02806?cid=3896599.

[15] A Hadid. The local binary pattern approach and its applications to face analysis. In *Image Processing Theory, Tools and Applications, 2008. IPTA 2008. First Workshops on*, pages 1–9, Nov 2008. doi: 10.1109/IPTA.2008.4743795.

[16] Chris Harris and Mike Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988.

[17] Zellig Harris. Distributional structure. *Word*, 10(23):146–162, 1954.

[18] Marko Heikkilä, Matti Pietikäinen, and Cordelia Schmid. Description of interest regions with local binary patterns. *Pattern Recogn.*, 42(3):425–436, March 2009. ISSN 0031-3203. doi: 10.1016/j.patcog.2008.08.014. URL http://dx.doi.org/10.1016/j.patcog.2008.08.014.

[19] Friedrich Heitger, Lukas Rosenthaler, Rdiger Von Der Heydt, Esther Peterhans, and Olaf Kbler. Simulation of neural contour mechanisms: from simple to end-stopped cells. *Vision Research*, 32(5):963 – 981, 1992. ISSN 0042-6989. doi: http://dx.doi.org/10.1016/0042-6989(92)90039-L. URL http://www.sciencedirect.com/science/article/pii/004269899290039L.

[20] R. Horaud and F. Veillon. Finding geometric and relational structures in an image. In *Proceedings of the First European Conference on Computer Vision*, ECCV 90, pages 374–384, New York, NY, USA, 1990. Springer-Verlag New York, Inc. ISBN 0-387-52522-X. URL http://dl.acm.org/citation.cfm?id=89081.89153.

[21] Anil K. Jain and Richard C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988. ISBN 0-13-022278-X.

[22] Timor Kadir and Michael Brady. Saliency, scale and image description. *Int. J. Comput. Vision*, 45(2):83–105, November 2001. ISSN 0920-5691. doi: 10.1023/A:1012460413855. URL http://dx.doi.org/10.1023/A:1012460413855.

[23] Alexander Klser, Marcin Marszaek, and Cordelia Schmid. A spatio-temporal descriptor based on 3d-gradients. In *In BMVC08*, 2008.

[24] J. Lankinen, V. Kangas, and J.-K. Kamarainen. A comparison of local feature detectors and descriptors for visual object categorization by intra-class repeatability and matching. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 780–783, Nov 2012.

[25] I Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008. doi: 10.1109/CVPR.2008.4587756.

[26] Ivan Laptev and Tony Lindeberg. Local descriptors for spatio-temporal recognition. In *In First International Workshop on Spatial Coherence for Visual Motion Analysis*, 2004.

[27] S. Lazebnik, C. Schmid, and J. 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, 2006. doi: 10.1109/CVPR.2006.68.

[28] Tobias Lindahl. Study of local binary patterns, 2007.

[29] Tony Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116, 1998. ISSN 0920-5691. doi: 10.1023/A:1008045108935. URL `http://dx.doi.org/10.1023/A%3A1008045108935`.

[30] Tony Lindeberg and Jonas Grding. Shape-adapted smoothing in estimation of 3-d shape cues from affine distortions of local 2-d brightness structure, 2001.

[31] David G. Lowe. Object recognition from local scale-invariant features, 1999.

[32] David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV '99, pages 1150–, Washington, DC, USA, 1999. IEEE Computer Society. ISBN 0-7695-0164-8. URL `http://dl.acm.org/citation.cfm?id=850924.851523`.

[33] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004. ISSN 0920-5691. doi: 10.1023/B:VISI.0000029664.99615.94. URL `http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94`.

[34] D.G. Lowe. Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1150–1157 vol.2, 1999. doi: 10.1109/ICCV.1999.790410.

[35] J. MacQueen. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability - Vol. 1.*

[36] J Matas, O Chum, M Urban, and T Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761 – 767, 2004. ISSN 0262-8856. doi: http://dx.doi.org/10.1016/j.imavis.2004.02.006. URL `http://www.sciencedirect.com/science/article/pii/S0262885604000435`. British Machine Vision Computing 2002.

[37] MATLAB. *version 8.4.0 (R2014b)*. The MathWorks Inc., Natick, Massachusetts, 2014.

[38] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 525–531 vol.1, 2001. doi: 10.1109/ICCV.2001.937561.

[39] K. Mikolajczyk, B. Leibe, and B. Schiele. Local features for object class recognition. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1792–1799 Vol. 2, Oct 2005. doi: 10.1109/ICCV.2005.146.

[40] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *Int. J. Comput. Vision*, 65(1-2):43–72, November 2005. ISSN 0920-5691. doi: 10.1007/s11263-005-3848-x. URL `http://dx.doi.org/10.1007/s11263-005-3848-x`.

[41] Krystian Mikolajczyk and Cordelia Schmid. Scale and affine invariant interest point detectors. *IJCV*, pages 63–86, 2004.

[42] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors, 2005.

[43] M-E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1447–1454, 2006.

[44] Eric Nowak, Frdric Jurie, and Bill Triggs. Sampling strategies for bag-of-features image classification. In *In Proc. ECCV*, pages 490–503. Springer, 2006.

[45] Timo Ojala, Matti Pietikinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1): 51 – 59, 1996. ISSN 0031-3203. doi: http://dx.doi.org/10.1016/0031-3203(95)00067-4. URL `http://www.sciencedirect.com/science/article/pii/0031320395000674`.

[46] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. J. Comput. Vision*, 42(3):145–175, May 2001. ISSN 0920-5691. doi: 10.1023/A:1011139631724. URL `http://dx.doi.org/10.1023/A:1011139631724`.

[47] Matti Pietikinen, Abdenour Hadid, Guoying Zhao, and Timo Ahonen. Local binary patterns for still images. In *Computer Vision Using Local Binary Patterns*, volume 40 of *Computational Imaging and Vision*, pages 13–47. Springer London, 2011. ISBN 978-0-85729-747-1. doi: 10.1007/978-0-85729-748-8_2. URL `http://dx.doi.org/10.1007/978-0-85729-748-8_2`.

[48] N. Pinto, Y. Barhomi, D.D. Cox, and J.J. DiCarlo. Comparing state-of-the-art visual features on invariant object recognition tasks. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pages 463–470, Jan 2011. doi: 10.1109/WACV.2011.5711540.

[49] Yuan Ren, Hui Wei, and Huilan Luo. Object recognition using words model of optimal size in histograms of oriented gradients. In *Intelligent Information Technology Application, 2009. IITA 2009. Third International Symposium on*, volume 2, pages 412–416, Nov 2009. doi: 10.1109/IITA.2009.189.

[50] Marco San Biagio, Samuele Martelli, Marco Crocco, Marco Cristani, and Vittorio Murino. Encoding classes of unaligned objects using structural similarity cross-covariance tensors. In Jos Ruiz-Shulcloper and Gabriella Sanniti di Baja, editors, *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, volume 8258 of *Lecture Notes in Computer Science*, pages 133–140. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-41821-1. doi: 10.1007/978-3-642-41822-8_17. URL `http://dx.doi.org/10.1007/978-3-642-41822-8_17`.

[51] A Satpathy, Xudong Jiang, and How-Lung Eng. Lbp-based edge-texture features for object recognition. *Image Processing, IEEE Transactions on*, 23(5):1953–1964, May 2014. ISSN 1057-7149. doi: 10.1109/TIP.2014.2310123.

[52] Cordelia Schmid, Roger Mohr, and Christian Bauckhage. Evaluation of interest point detectors. *Int. J. Comput. Vision*, 37(2):151–172, June 2000. ISSN 0920-5691. doi: 10.1023/A:1008199403446. URL `http://dx.doi.org/10.1023/A:1008199403446`.

[53] R. Sibson. SLINK: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34, 1973.

[54] R. R. Sokal and C. D. Michener. A statistical method for evaluating systematic relationships. *University of Kansas Scientific Bulletin*, 28:1409–1438, 1958.

[55] M. Stark and B. Schiele. How good are local features for classes of geometric objects. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Oct 2007. doi: 10.1109/ICCV.2007.4408878.

[56] E. Tola, V.Lepetit, and P. Fua. A Fast Local Descriptor for Dense Matching. In *Proceedings of Computer Vision and Pattern Recognition*, Alaska, USA, 2008.

[57] E. Tola, V. Lepetit, and P. Fua. DAISY: An Efficient Dense Descriptor Applied to Wide Baseline Stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32 (5):815–830, May 2010.

[58] Tinne Tuytelaars and Krystian Mikolajczyk. Local invariant feature detectors: A survey. *Found. Trends. Comput. Graph. Vis.*, 3(3):177–280, July 2008. ISSN 1572-2740. doi: 10.1561/0600000017. URL `http://dx.doi.org/10.1561/0600000017`.

[59] Julia Vogel and Bernt Schiele. A semantic typicality measure for natural scene categorization. In *Pattern Recognition Symposium, DAGM*, 2004.

[60] Li Wang and Dong-Chen He. Texture classification using texture spectrum. *Pattern Recogn.*, 23(8):905–910, August 1990. ISSN 0031-3203. doi: 10.1016/0031-3203(90)90135-8. URL `http://dx.doi.org/10.1016/0031-3203(90)90135-8`.

[61] Joe H. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963. doi: 10.1080/01621459.1963.10500845. URL `http://www.tandfonline.com/doi/abs/10.1080/01621459.1963.10500845`.

[62] Geert Willems, Tinne Tuytelaars, and Luc Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *Proceedings of the 10th European Conference on Computer Vision: Part II*, ECCV '08, pages 650–663, Berlin, Heidelberg, 2008.

Springer-Verlag. ISBN 978-3-540-88685-3. doi: 10.1007/978-3-540-88688-4_48. URL http://dx.doi.org/10.1007/978-3-540-88688-4_48.

[63] Jianguo Zhang. *Dataset Issues in Object Recognition*, pages 29–48. Springer, 2006.

[64] Jianguo Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06. Conference on*, pages 13–13, June 2006. doi: 10.1109/CVPRW.2006.121.

[65] Chao Zhu, Charles-Edmond Bichot, and Liming Chen. Visual object recognition using daisy descriptor. In *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, pages 1–6, July 2011. doi: 10.1109/ICME.2011.6011957.

[66] Qiang Zhu, M.-C. Yeh, Kwang-Ting Cheng, and S. Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1491–1498, 2006. doi: 10.1109/CVPR.2006.119.

APPENDIX A

Uniform LBP

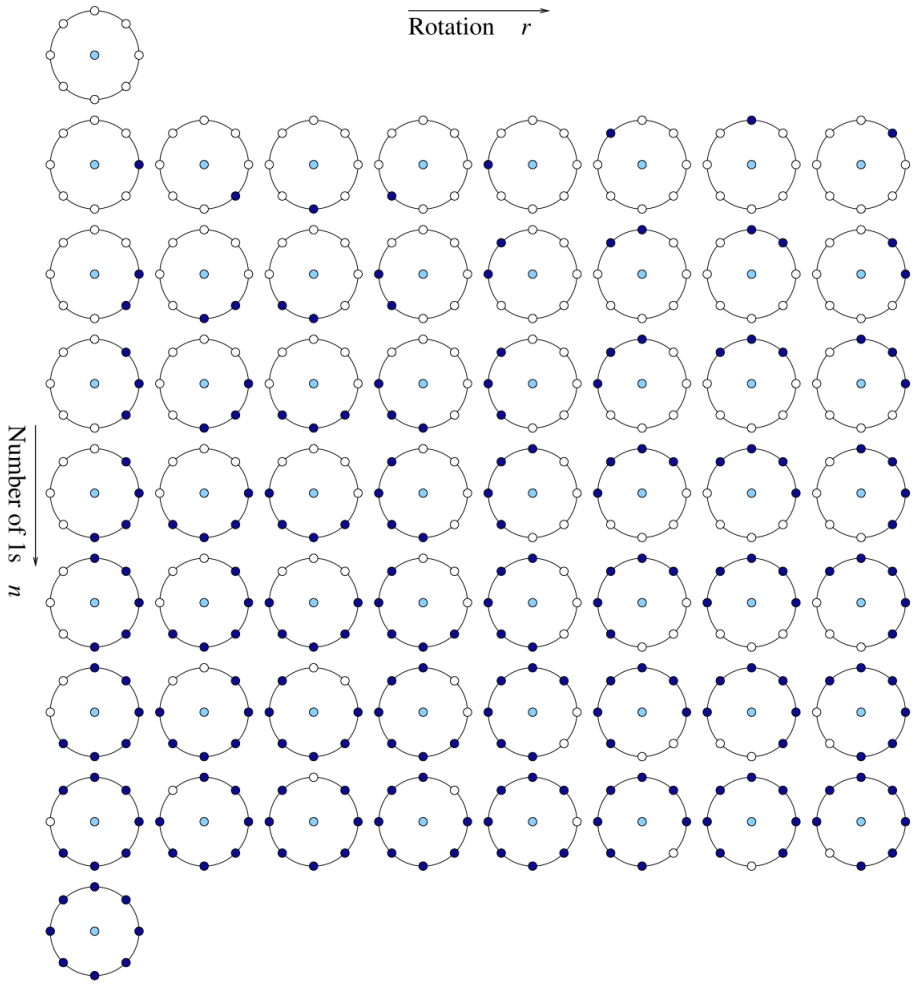


Figure A.1. The 58 different uniform patterns in (8,R) neighborhood. [47].

APPENDIX B

Images used for dendrogram generation

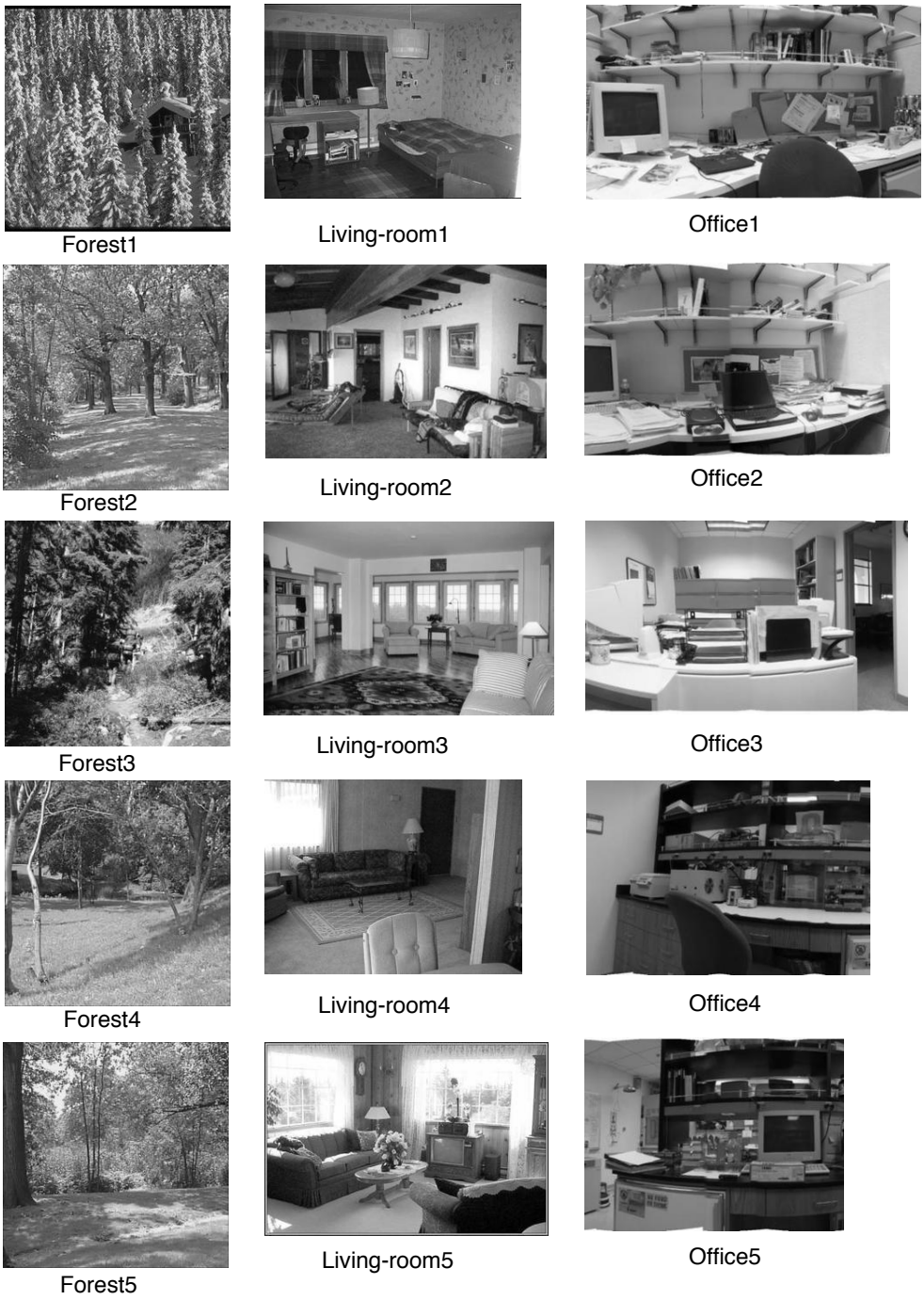


FIGURE B.1. Images used to generate dendrogram shown in Figure 3.3. All the images are taked from 15-Scene dataset [11]

APPENDIX C

More information on Datasets

C.1. Caltech-256

Figure C.1. Collection of average images of all the categories in Caltech-256 [50].

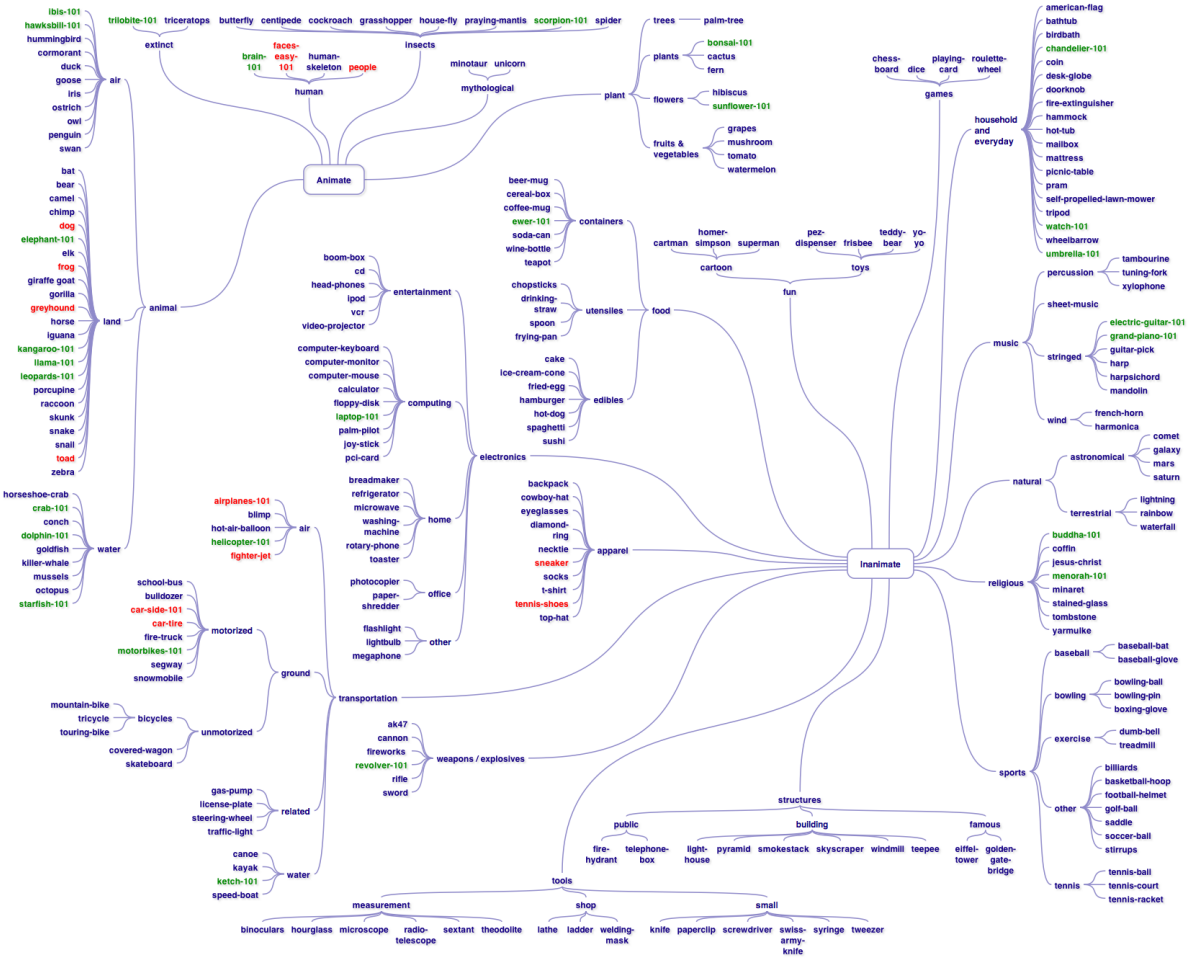


FIGURE C.2. Taxonomy of Caltech-256 classes. Classes in green are taked from Caltech-101. Classes in red are 6 pairs of overlapping categories such as airplane-101 and fighter-jet. [14]

Figure C.3. Example images of 15-Scenes dataset [11]

C.3. Flowers

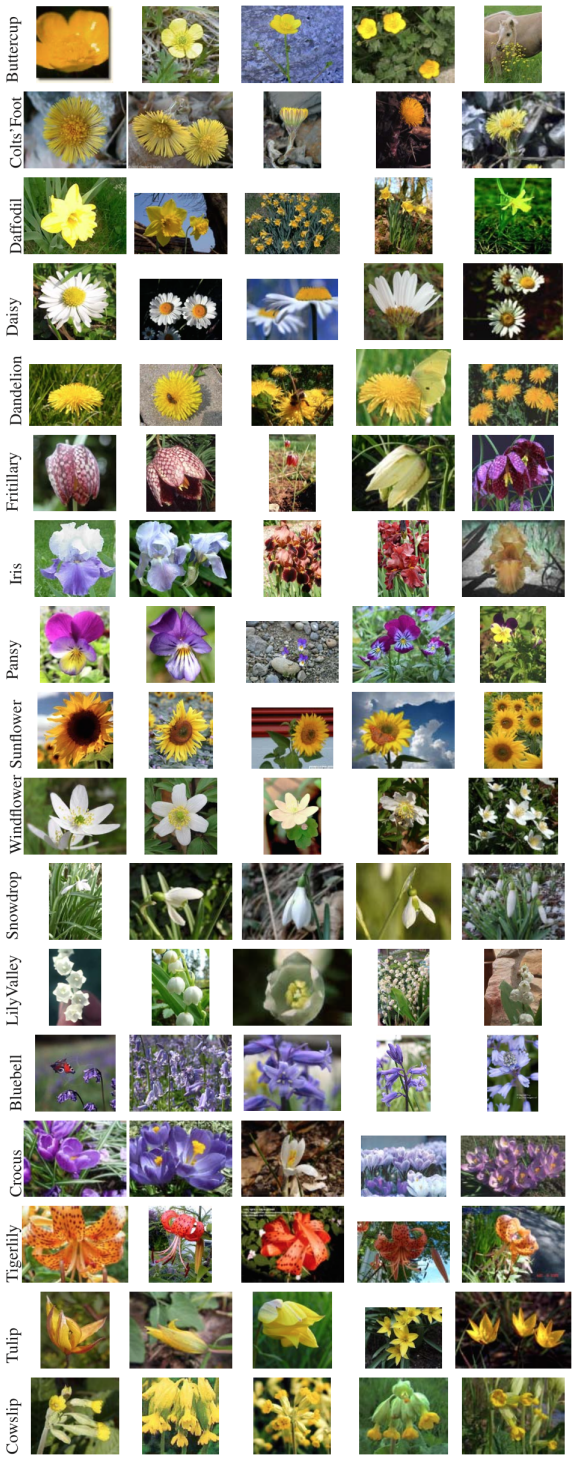


Figure C.4. Example images of Flowers dataset [11]

APPENDIX D

Results Tables and Plots

D.1. Caltech-256 - Category Level F measure for all the categories

Table D.1. Category Level F measure on Caltech-256. GIST outperfoms All other Local Descriptors on 15 highlighted Categories. HL - Hessian Laplace keypoints, GL - Grid Points.

	SIFT		Daisy		HOG		LBP		GIST
	HL	GP	HL	GP	HL	GP	HL	GP	
airplanes-101	0.63	0.58	0.71	0.77	0.74	0.61	0.69	0.62	0.69
backpack	0.34	0.28	0.34	0.40	0.33	0.21	0.25	0.23	0.24
baseball-bat	0.29	0.39	0.25	0.41	0.23	0.44	0.27	0.41	0.32
basketball-hoop	0.25	0.16	0.22	0.12	0.15	0.12	0.14	0.13	0.09
bat	0.15	0.14	0.13	0.19	0.14	0.16	0.15	0.15	0.12
beer-mug	0.14	0.17	0.13	0.26	0.17	0.17	0.13	0.16	0.17
binoculars	0.49	0.41	0.44	0.49	0.37	0.49	0.38	0.44	0.29
birdbath	0.11	0.19	0.14	0.17	0.13	0.17	0.13	0.17	0.11
blimp	**0.09**	**0.09**	**0.10**	**0.12**	**0.10**	**0.11**	**0.10**	**0.18**	**0.19**
bonsai-101	0.27	0.14	0.27	0.24	0.34	0.21	0.26	0.25	0.19
boom-box	0.17	0.24	0.15	0.19	0.26	0.21	0.17	0.16	0.13
bowling-ball	**0.17**	**0.19**	**0.17**	**0.22**	**0.16**	**0.17**	**0.12**	**0.20**	**0.26**
boxing-glove	0.16	0.18	0.12	0.21	0.16	0.19	0.14	0.18	0.15
brain-101	0.24	0.24	0.23	0.36	0.20	0.30	0.21	0.33	0.19
breadmaker	**0.36**	**0.30**	**0.33**	**0.37**	**0.39**	**0.26**	**0.30**	**0.30**	**0.42**
buddha-101	0.11	0.15	0.25	0.17	0.17	0.18	0.12	0.20	0.12
bulldozer	0.16	0.18	0.15	0.19	0.21	0.23	0.19	0.18	0.17
cactus	0.17	0.20	0.17	0.20	0.13	0.26	0.15	0.18	0.18

calculator	**0.19**	**0.22**	**0.12**	**0.18**	**0.26**	**0.20**	**0.12**	**0.22**	**0.27**
camel	0.18	0.17	0.20	0.16	0.27	0.25	0.18	0.16	0.16
cannon	0.20	0.21	0.19	0.21	0.17	0.23	0.16	0.19	0.20
canoe	0.16	0.17	0.14	0.17	0.14	0.19	0.13	0.18	0.15
car-side-101	0.33	0.52	0.55	0.73	0.62	0.88	0.58	0.86	0.60
car-tire	**0.18**	**0.14**	**0.13**	**0.13**	**0.11**	**0.11**	**0.15**	**0.14**	**0.28**
cartman	0.19	0.19	0.24	0.20	0.17	0.16	0.19	0.15	0.14
cereal-box	0.20	0.23	0.21	0.24	0.36	0.25	0.17	0.19	0.15
chandelier-101	0.13	0.17	0.17	0.32	0.16	0.27	0.17	0.22	0.21
chess-board	0.35	0.20	0.29	0.25	0.28	0.27	0.16	0.22	0.15
chimp	0.18	0.22	0.18	0.26	0.17	0.22	0.17	0.20	0.15
cockroach	0.28	0.17	0.27	0.16	0.17	0.15	0.15	0.16	0.20
coffee-mug	0.13	0.16	0.15	0.19	0.13	0.18	0.14	0.13	0.13
comet	0.10	0.34	0.09	0.51	0.10	0.43	0.08	0.28	0.15
computer-keyboard	0.13	0.16	0.24	0.15	0.11	0.20	0.10	0.18	0.20
computer-mouse	0.17	0.18	0.20	0.25	0.24	0.19	0.17	0.19	0.22
cormorant	0.13	0.31	0.18	0.39	0.18	0.41	0.17	0.29	0.18
crab-101	0.16	0.21	0.18	0.16	0.25	0.15	0.20	0.13	0.20
desk-globe	0.27	0.18	0.19	0.32	0.17	0.19	0.14	0.20	0.16
diamond-ring	0.20	0.21	0.24	0.23	0.22	0.23	0.18	0.19	0.16
dice	0.19	0.16	0.18	0.16	0.12	0.23	0.20	0.25	0.08
dog	0.14	0.16	0.15	0.18	0.14	0.18	0.16	0.18	0.11
dolphin-101	0.25	0.15	0.16	0.20	0.16	0.26	0.14	0.20	0.22
doorknob	0.11	0.22	0.13	0.24	0.15	0.22	0.12	0.21	0.13
drinking-straw	0.12	0.16	0.11	0.12	0.13	0.13	0.12	0.11	0.12
dumb-bell	0.14	0.15	0.11	0.13	0.13	0.17	0.15	0.16	0.15
eiffel-tower	0.21	0.18	0.15	0.14	0.12	0.14	0.11	0.16	0.19

electric-guitar-101	**0.15**	**0.15**	**0.12**	**0.16**	**0.14**	**0.17**	**0.15**	**0.14**	**0.17**
elephant-101	0.13	0.24	0.13	0.18	0.12	0.28	0.15	0.25	0.15
elk	0.16	0.18	0.16	0.25	0.18	0.25	0.19	0.17	0.11
ewer-101	0.19	0.12	0.26	0.15	0.26	0.14	0.13	0.18	0.16
faces-easy-101	0.68	0.66	0.85	0.73	0.64	0.76	0.47	0.69	0.49
fern	0.27	0.35	0.25	0.33	0.22	0.24	0.28	0.30	0.33
fire-extinguisher	0.20	0.16	0.20	0.25	0.23	0.15	0.17	0.18	0.13
fire-hydrant	0.12	0.15	0.12	0.15	0.12	0.17	0.11	0.16	0.13
fireworks	0.24	0.18	0.29	0.41	0.24	0.15	0.19	0.23	0.15
flashlight	0.20	0.22	0.19	0.32	0.20	0.37	0.20	0.38	0.31
floppy-disk	0.22	0.18	0.20	0.15	0.17	0.16	0.15	0.13	0.11
football-helmet	0.29	0.17	0.15	0.21	0.20	0.14	0.14	0.14	0.16
frying-pan	0.25	0.22	0.20	0.24	0.20	0.26	0.17	0.26	0.24
galaxy	0.49	0.17	0.40	0.20	0.37	0.23	0.24	0.26	0.17
giraffe	0.16	0.16	0.13	0.17	0.19	0.16	0.15	0.18	0.16
goat	0.17	0.17	0.14	0.16	0.17	0.22	0.17	0.18	0.12
golden-gate-bridge	0.20	0.25	0.29	0.26	0.24	0.21	0.21	0.16	0.17
goldfish	0.14	0.16	0.14	0.23	0.13	0.20	0.11	0.21	0.11
golf-ball	0.29	0.12	0.11	0.15	0.18	0.12	0.14	0.13	0.10
goose	0.15	0.21	0.12	0.23	0.14	0.23	0.16	0.24	0.15
gorilla	0.36	0.40	0.34	0.36	0.36	0.48	0.38	0.47	0.21
grapes	0.28	0.39	0.26	0.58	0.27	0.51	0.28	0.42	0.20
grasshopper	0.21	0.22	0.23	0.20	0.21	0.33	0.27	0.32	0.18
guitar-pick	0.31	0.14	0.28	0.45	0.28	0.22	0.14	0.19	0.18
harmonica	0.16	0.17	0.11	0.24	0.14	0.16	0.11	0.16	0.11
harp	0.12	0.15	0.13	0.31	0.29	0.42	0.20	0.42	0.14
helicopter-101	0.15	0.13	0.35	0.24	0.26	0.18	0.16	0.20	0.15

hibiscus	0.42	0.22	0.42	0.27	0.28	0.39	0.25	0.30	0.21
horse	0.28	0.39	0.27	0.37	0.27	0.32	0.28	0.32	0.28
horseshoe-crab	0.15	0.17	0.16	0.19	0.18	0.23	0.15	0.21	0.14
hot-air-balloon	0.19	0.13	0.13	0.16	0.13	0.18	0.13	0.15	0.14
hot-dog	0.11	0.13	0.20	0.15	0.12	0.17	0.12	0.15	0.15
hot-tub	0.14	0.36	0.14	0.26	0.12	0.30	0.15	0.27	0.30
hourglass	0.11	0.13	0.11	0.16	0.15	0.17	0.13	0.15	0.15
house-fly	0.45	0.15	0.22	0.11	0.10	0.12	0.12	0.11	0.12
human-skeleton	0.19	0.12	0.42	0.14	0.16	0.12	0.15	0.15	0.11
ice-cream-cone	0.18	0.11	0.17	0.13	0.17	0.11	0.18	0.12	0.11
iguana	0.21	0.20	0.21	0.24	0.22	0.24	0.20	0.19	0.19
iris	0.10	0.24	0.11	0.25	0.13	0.42	0.12	0.30	0.18
joy-stick	0.15	0.18	0.16	0.16	0.17	0.17	0.15	0.18	0.15
kangaroo-101	0.24	0.20	0.19	0.20	0.15	0.23	0.15	0.28	0.15
kayak	0.12	0.21	0.11	0.28	0.10	0.37	0.11	0.20	0.17
ketch-101	0.29	0.22	0.46	0.24	0.30	0.27	0.24	0.23	0.25
killer-whale	0.18	0.27	0.20	0.35	0.17	0.31	0.11	0.21	0.20
knife	0.15	0.17	0.15	0.17	0.14	0.18	0.17	0.18	0.16
ladder	0.24	0.28	0.23	0.27	0.28	0.30	0.23	0.26	0.27
laptop-101	0.18	0.17	0.19	0.23	0.36	0.24	0.23	0.17	0.17
lathe	0.16	0.26	0.14	0.31	0.19	0.34	0.15	0.30	0.19
leopards-101	0.51	0.54	0.64	0.99	0.60	0.84	0.71	0.51	0.86
license-plate	0.42	0.14	0.26	0.25	0.30	0.25	0.16	0.42	0.21
light-house	**0.26**	**0.25**	**0.26**	**0.25**	**0.33**	**0.27**	**0.29**	**0.27**	**0.33**
lightbulb	0.14	0.20	0.17	0.13	0.21	0.23	0.20	0.17	0.14
llama-101	0.14	0.22	0.16	0.22	0.14	0.20	0.12	0.22	0.14
mailbox	0.14	0.14	0.14	0.13	0.14	0.15	0.13	0.14	0.09

mars	0.13	0.26	0.12	0.58	0.12	0.33	0.11	0.34	0.42
megaphone	0.14	0.22	0.22	0.13	0.15	0.13	0.16	0.15	0.15
menorah-101	0.18	0.14	0.29	0.26	0.24	0.13	0.18	0.14	0.14
microwave	0.22	0.32	0.18	0.21	0.43	0.54	0.19	0.44	0.12
minaret	**0.15**	**0.22**	**0.16**	**0.27**	**0.18**	**0.31**	**0.16**	**0.30**	**0.31**
motorbikes-101	0.55	0.70	0.55	0.80	0.55	0.72	0.55	0.72	0.61
mountain-bike	0.42	0.16	0.34	0.22	0.20	0.15	0.16	0.19	0.31
mushroom	0.39	0.41	0.31	0.38	0.37	0.40	0.35	0.38	0.25
mussels	0.12	0.24	0.12	0.26	0.12	0.27	0.12	0.28	0.12
necktie	0.17	0.19	0.17	0.23	0.14	0.17	0.18	0.17	0.19
owl	0.13	0.18	0.10	0.23	0.10	0.20	0.12	0.21	0.15
palm-pilot	**0.16**	**0.17**	**0.11**	**0.15**	**0.16**	**0.15**	**0.12**	**0.13**	**0.18**
palm-tree	0.22	0.22	0.21	0.18	0.18	0.19	0.19	0.27	0.20
paper-shredder	**0.11**	**0.25**	**0.09**	**0.34**	**0.09**	**0.29**	**0.13**	**0.30**	**0.38**
paperclip	0.15	0.22	0.22	0.12	0.16	0.12	0.16	0.13	0.14
penguin	0.18	0.22	0.23	0.23	0.21	0.22	0.25	0.25	0.21
pez-dispenser	0.14	0.13	0.11	0.18	0.12	0.15	0.10	0.15	0.12
picnic-table	0.15	0.15	0.14	0.18	0.13	0.28	0.12	0.20	0.15
porcupine	0.24	0.21	0.27	0.37	0.15	0.28	0.18	0.21	0.16
pram	0.19	0.16	0.26	0.16	0.17	0.15	0.14	0.13	0.13
praying-mantis	0.17	0.14	0.18	0.15	0.13	0.16	0.13	0.17	0.12
pyramid	0.15	0.14	0.14	0.20	0.14	0.17	0.12	0.15	0.12
raccoon	0.24	0.34	0.21	0.23	0.28	0.36	0.25	0.26	0.21
radio-telescope	0.13	0.14	0.13	0.23	0.11	0.23	0.09	0.23	0.16
rainbow	0.13	0.34	0.11	0.35	0.11	0.30	0.12	0.35	0.29
refrigerator	0.10	0.15	0.11	0.25	0.22	0.20	0.15	0.20	0.16
rifle	**0.15**	**0.15**	**0.13**	**0.13**	**0.11**	**0.14**	**0.15**	**0.15**	**0.16**

saddle	0.24	0.19	0.23	0.19	0.20	0.23	0.22	0.23	0.13
school-bus	0.32	0.22	0.29	0.24	0.27	0.47	0.20	0.27	0.14
scorpion-101	0.22	0.15	0.17	0.17	0.19	0.18	0.13	0.15	0.14
skateboard	0.16	0.12	0.11	0.13	0.15	0.13	0.13	0.14	0.15
smokestack	0.23	0.16	0.15	0.16	0.10	0.19	0.10	0.21	0.21
snake	0.22	0.27	0.19	0.21	0.21	0.34	0.24	0.28	0.20
soccer-ball	**0.19**	**0.24**	**0.19**	**0.34**	**0.20**	**0.22**	**0.19**	**0.21**	**0.35**
socks	0.18	0.17	0.17	0.15	0.14	0.14	0.13	0.15	0.13
soda-can	0.14	0.13	0.12	0.14	0.11	0.15	0.13	0.14	0.12
spaghetti	0.08	0.14	0.07	0.23	0.08	0.16	0.08	0.20	0.15
speed-boat	0.14	0.35	0.27	0.41	0.18	0.47	0.13	0.38	0.15
spoon	0.19	0.16	0.19	0.21	0.16	0.25	0.17	0.26	0.11
starfish-101	0.17	0.16	0.14	0.19	0.23	0.19	0.20	0.19	0.16
steering-wheel	0.30	0.23	0.19	0.26	0.15	0.17	0.21	0.12	0.15
sunflower-101	**0.28**	**0.14**	**0.24**	**0.37**	**0.20**	**0.17**	**0.15**	**0.22**	**0.29**
superman	0.12	0.14	0.18	0.14	0.15	0.19	0.09	0.13	0.13
swan	0.19	0.21	0.18	0.29	0.19	0.25	0.16	0.20	0.18
swiss-army-knife	0.19	0.23	0.17	0.25	0.18	0.24	0.20	0.24	0.23
sword	0.12	0.25	0.14	0.29	0.12	0.35	0.16	0.35	0.13
syringe	0.14	0.21	0.13	0.15	0.13	0.19	0.11	0.19	0.16
t-shirt	0.39	0.36	0.36	0.39	0.41	0.36	0.34	0.34	0.34
teapot	0.15	0.20	0.16	0.18	0.18	0.22	0.14	0.21	0.18
telephone-box	0.16	0.23	0.20	0.20	0.33	0.40	0.23	0.32	0.14
tennis-ball	0.24	0.15	0.24	0.22	0.13	0.20	0.15	0.19	0.22
tennis-court	0.19	0.22	0.16	0.33	0.15	0.27	0.14	0.21	0.24
tennis-racket	0.26	0.12	0.21	0.13	0.15	0.11	0.12	0.12	0.16
theodolite	0.25	0.19	0.23	0.21	0.17	0.14	0.19	0.19	0.20

toad	0.25	0.26	0.28	0.25	0.32	0.36	0.26	0.40	0.20
tomato	0.24	0.32	0.25	0.23	0.12	0.22	0.19	0.42	0.18
tombstone	0.18	0.20	0.15	0.21	0.16	0.23	0.17	0.24	0.14
touring-bike	0.40	0.19	0.27	0.33	0.28	0.11	0.16	0.20	0.23
treadmill	0.25	0.29	0.30	0.30	0.26	0.28	0.36	0.31	0.25
triceratops	0.25	0.17	0.19	0.19	0.20	0.16	0.24	0.18	0.15
tricycle	0.19	0.19	0.18	0.16	0.18	0.16	0.15	0.10	0.14
trilobite-101	0.20	0.31	0.32	0.49	0.30	0.70	0.26	0.59	0.19
tripod	0.24	0.26	0.20	0.25	0.14	0.19	0.18	0.19	0.12
tuning-fork	0.22	0.13	0.19	0.15	0.18	0.12	0.15	0.18	0.16
umbrella-101	0.17	0.17	0.20	0.19	0.14	0.18	0.15	0.20	0.17
vcr	0.18	0.17	0.19	0.26	0.20	0.19	0.17	0.18	0.23
video-projector	0.19	0.18	0.32	0.17	0.27	0.16	0.22	0.21	0.19
washing-machine	0.18	0.14	0.13	0.20	0.15	0.22	0.13	0.19	0.15
watch-101	0.24	0.24	0.22	0.28	0.28	0.23	0.29	0.27	0.20
waterfall	0.29	0.34	0.39	0.32	0.25	0.32	0.27	0.33	0.14
watermelon	0.15	0.16	0.13	0.19	0.16	0.16	0.16	0.19	0.13
wheelbarrow	0.13	0.14	0.17	0.16	0.13	0.17	0.13	0.14	0.13
windmill	**0.11**	**0.14**	**0.12**	**0.15**	**0.13**	**0.15**	**0.12**	**0.14**	**0.18**
wine-bottle	0.20	0.16	0.18	0.16	0.30	0.16	0.21	0.16	0.14
xylophone	0.10	0.16	0.13	0.18	0.11	0.18	0.12	0.18	0.12
yarmulke	0.33	0.20	0.34	0.32	0.19	0.22	0.20	0.25	0.29
yo-yo	**0.11**	**0.12**	**0.10**	**0.16**	**0.11**	**0.13**	**0.08**	**0.14**	**0.16**
zebra	0.61	0.19	0.56	0.49	0.47	0.25	0.46	0.26	0.16

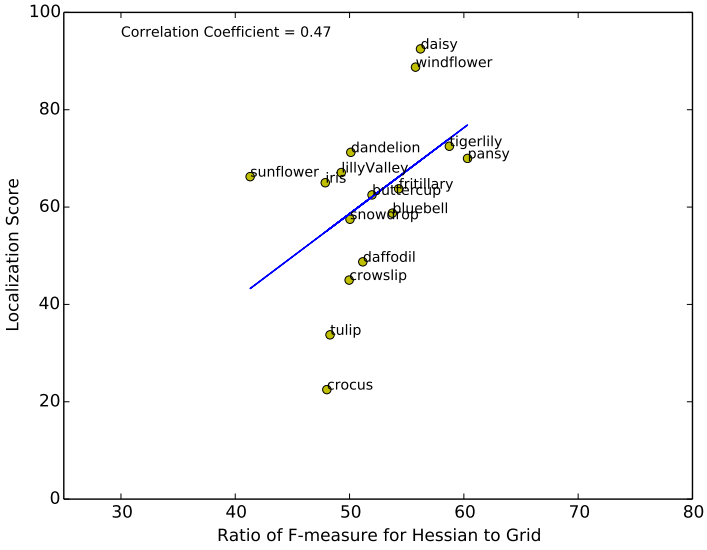


Figure D.1. Localization Score vs F measure Ratio between Hessian and Grid for HOG Descriptors on Flowers Dataset

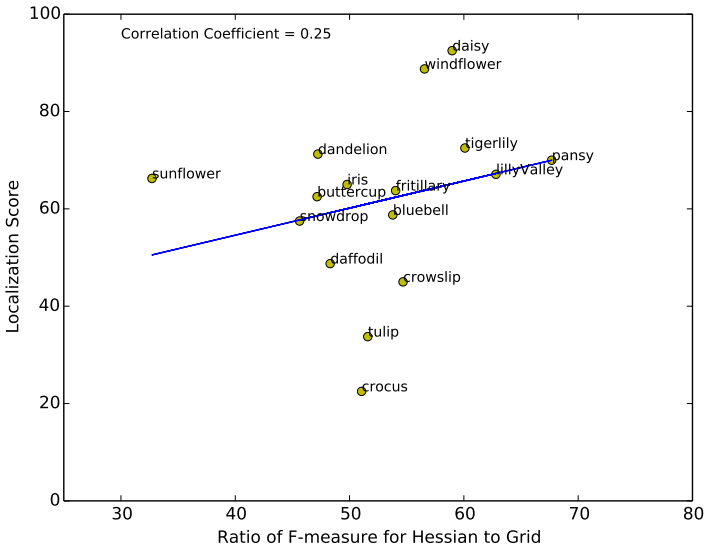


Figure D.2. Localization score vs F measure ratio between Hessian and Grid for DAISY Descriptors on Flowers Dataset

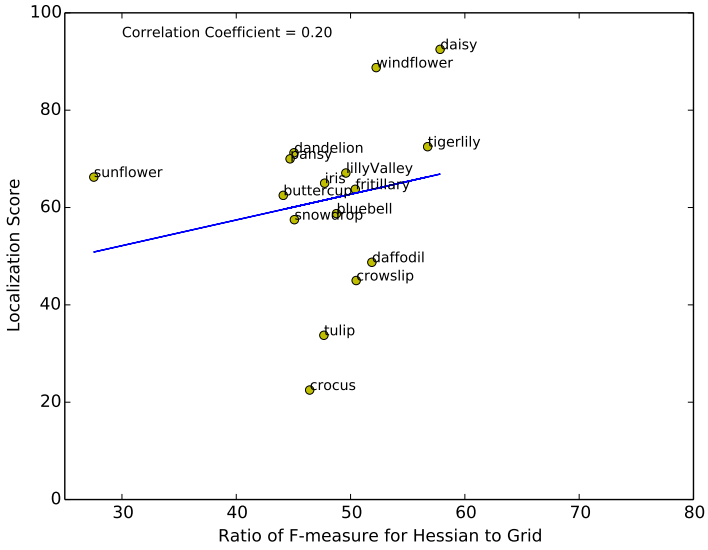


FIGURE D.3. Localization score vs F measure ratio between Hessian and Grid for LBP Descriptors on Flowers Dataset

Table D.2. Category Level Localization Results on Caltech-256. F(HP) is F measure for Hessian-Laplace and F(GP) is F measure for Grid Points

		F(HL) - F(GP)			
	Localization Score	SIFT	Daisy	HoG	LBP
bonsai-101	82.79	0.13	0.04	0.13	0.01
theodolite	78.57	0.06	0.02	0.03	0.01
pram	64.77	0.03	0.10	0.02	0.02
human-skeleton	72.62	0.07	0.28	0.04	0.00
mountain-bike	84.15	0.26	0.12	0.05	-0.03
ice-cream-cone	60.23	0.06	0.04	0.06	0.06
zebra	76.04	0.42	0.07	0.22	0.20
cockroach	68.55	0.11	0.10	0.02	-0.00
menorah-101	95.51	0.04	0.03	0.11	0.05
tennis-racket	74.07	0.14	0.08	0.04	-0.00
ketch-101	82.88	0.07	0.22	0.03	0.01
basketball-hoop	58.89	0.10	0.09	0.03	0.00
car-side-101	7.76	-0.18	-0.18	-0.26	-0.28
elephant-101	44.27	-0.11	-0.04	-0.17	-0.10
goldfish	27.96	-0.02	-0.09	-0.07	-0.10
iris	34.26	-0.14	-0.14	-0.29	-0.17
comet	27.27	-0.24	-0.42	-0.34	-0.20
kayak	31.07	-0.08	-0.17	-0.27	-0.09
horse	33.70	-0.11	-0.10	-0.05	-0.04
picnic-table	38.46	-0.00	-0.05	-0.15	-0.08
gorilla	31.13	-0.04	-0.02	-0.12	-0.08

TABLE D.3. Average F-measure Scatter Plot that shows Dataset influenced performance of GIST

	SIFT	LBP	HoG	Daisy	GIST
15 Scenes	0.30	0.37	0.43	0.35	0.51
Caltech-256	0.27	0.29	0.30	0.32	0.25
Flowers	0.43	0.27	0.28	0.37	0.24