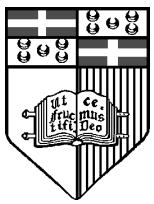
Image Classification using Bag of Visual Words and Novel COSFIRE Descriptors

Matthew Grech

Supervisor: Dr. George Azzopardi



Faculty of ICT

University of Malta

May 2016

Submitted in partial fulfillment of the requirements for the degree of B.Sc. ICT in Artificial Intelligence (Hons.)

UNIVERSITY OF MALTA
FACULTY/INSTITUTE/CENTRE/SCHOOL ICT Faculty

DECLARATION OF AUTHENTICITY FOR UNDERGRADUATE STUDENTS

Student's I.D. /Code 497192 (M)

Student's Name & Surname MATTHEW GRECH

Course B.Sc. in Information Tech (Hons) AI

Title of Long Essay/Dissertation

IMAGE CLASSIFICATION USING BAG OF VISUAL WORDS AND NOVEL COSFIRE DESCRIPTORS

I hereby declare that I am the legitimate author of this Long Essay/Dissertation and that it is my original work.

No portion of this work has been submitted in support of an application for another degree or qualification of this or any other university or institution of higher education.

I hold the University of Malta harmless against any third party claims with regard to copyright violation, breach of confidentiality, defamation and any other third party right infringement.

Signature of Student

30/05/2016
Date

11.06.2015

Abstract

The main task of a keypoint descriptor is to describe an interesting patch(keypoint) in an image. This project proposes a new keypoint descriptor, based on the trainable COSFIRE filters that are used for keypoint detection and pattern recognition, to describe keypoints found in an image. A keypoint is a particular patch within an image that is deemed to be an interesting patch by a keypoint detector. A visual descriptor effectively describes the detected keypoints by being robust to changes in different image conditions while also being distinctive between different keypoints.

We analyse the popular Bag of Visual Words (BOVW) image classification model, by examining each step of this model and choosing the best design configuration, starting from the extraction and description of the image keypoints to the classification of unseen image dataset.

The proposed solution takes into consideration the configuration parameters found in the COSFIRE filters to effectively construct the novel keypoint descriptor. Different COSFIRE descriptor configurations were proposed in this project and their performance was assessed, along with other popular keypoint descriptors, on the popular procedure defined by [1], where different image conditions, such as variation of viewpoint or blur, are taken into account to test the descriptor's effectiveness. The best COSFIRE descriptor was then chosen along with the state-of-the-art SIFT descriptor [2] to evaluate their accuracy rate using the BoVW model.

We evaluated our COSFIRE descriptors along with other popular keypoint descriptors such as SIFT [2] and BRISK [3]. The performance of the COSFIRE-336 descriptor achieved the best performance results amongst the configurations proposed in this project, exceeding the SIFT and the BRISK descriptors' performance in various image conditions. The COSFIRE-336 keypoint descriptor achieved an impressive accuracy rate when evaluated using the BoVW model, achieving a higher accuracy rate than SIFT on an unseen dataset of 15 different categories.

Acknowledgements

I would like to express my deepest appreciation to my supervisor Dr. George Azzopardi for his dedication, constant guidance, motivation and passion throughout this entire project, without which this work would not have been possible. His expertise in computer vision and machine learning combined with his ability of explaining complex concepts in such a way that everyone would be able to understand is what made him a great mentor.

I would also like to thank all the other lecturers of the AI department at University of Malta who introduced me to several other important areas of artificial intelligence, which essentially motivated me for several other projects. Last but not least, I would like to thank my family for their support throughout this entire journey.

Contents

List of Figures

List of Tables

1 Introduction

A brief definition of image classification would be the process of placing two images that contain similar components (both images containing a chair for example) in to the same category. The main task of image classification is to describe the important image features found in the images of the training set as can be seen in Fig.1(a) and associate them with their given label (such as Swan, Butterfly etc.) and then being able to give the appropriate label to the unseen images in the test dataset as seen in Fig.1(b). This pattern recognition approach is also known as supervised learning since the process of image classification consists of building a model for each image category in the training phase, which can then be used for predicting the categories of images in an unseen dataset. There are several applications of image classification and they are quite important in today's world. These include face recognition for security purposes, traffic lights recognition for driver-less cars, scene categorization for robot to act accordingly and even in medical imagery classification to help detect certain diseases, such as leukaemia by screening of blood and bone marrow smears [4].

Image classification, like other computer vision applications, rely heavily on the description of certain salient features in the images provided. Such features can be automatically detected by various keypoint detection techniques, such as the scale spaces generated by Difference-of-Gaussians [5] and Hessian-Laplace [6].

Once the keypoints are extracted from each image, they are described using a keypoint descriptor. A keypoint descriptor is a method that describes a keypoint from the characteristics of its neighbourhood and represent it in a data structure, such as a vector of real numbers. Different keypoint descriptors have different properties. For example, if the keypoint descriptor being used is scale independent and two keypoints are the same but of different scale, the descriptor will generate similar feature vectors. Two descriptors that have gained certain popularity are the SIFT [2] and the SURF [8] descriptors due to their performance and accuracy.

(a) (b)

Figure 1: Sample images from different categories of the training (a) and testing (b) dataset. These categories are extracted from the CALTECH-256 dataset [7] and will be used for the evaluation of the descriptors.

1.1 Motivation

Keypoint description is one of the most fundamental steps in computer vision since it facilitates the representation of a particular image regardless of its scale, transformation or the position (of the object being described). In image classification, making use of an image descriptor which has high performance, accuracy and robustness to noise while being invariant to image conditions such as different lighting, rotation, and different viewpoint angles, has a critical impact on the final result of the image classification's process. This is because it assumes that the descriptor is able to recognize the same interest points under different conditions, but able to be distinctive between different keypoints. If the descriptor fails on this task, then the allocations of interest points of a particular image category might be affected and result in reduction of distinctiveness between image categories.

There are several other applications where keypoint description plays an important role besides image classification. For example object recognition and tracking systems rely heavily on image descriptors to describe what they are seeing. Tracking systems are currently being used mostly in robotics [9] so robots can adapt to the surrounding

environment. Other applications include panorama stitching [10], where a panorama is constructed from a sequence of images, and 3D scene modelling [11].

Our novel visual descriptor is based on the COSFIRE filters and as shown in [12] [13] these proposed filters have been proven to have outstanding results, while being conceptually simple and easy to implement. The COSFIRE filters' precision rates of over 96% in several different fields [12] [13], such as the detection of retinal vascular bifurcations, recognition of handwritten digits and recognition of traffic signs in complex scenes. However these filters output are tuples in which they vary in cardinality and therefore can not be applied to machine learning algorithms due to the inability to having a fixed size output. This is where our proposed COSFIRE descriptor comes in, since it is able to convert the output of the COSFIRE filters into a fixed size vector while still retaining the information of the filters.

1.2 Approach

The approach to this project consists of constructing and evaluating different keypoint descriptors based on the configuration parameters of the COSFIRE filters and compare the results with other popular keypoint descriptors that are currently being used in the computer vision area. A perfect descriptor should be distinctive from one interest point to another but also robust to different changes in viewing conditions and light for example. There are several procedures that one can follow to compare the performance of a selected number of descriptors but the most popular and reliable one is the one proposed by K. Mikolajczyk and C. Schmid [1]. A great observation made in this procedure is that the performance of a wide range of popular visual descriptors resulted to be independent from the keypoint detector. This means that for this project, the choice of the keypoint detector would not affect the performance of the visual descriptors because its performance is independent of the patches provided.

This procedure uses a dataset that consists of five different image conditions which are blur, JPEG compression, light, viewpoint, zoom and rotation, each category containing one original image and 5 other transformed images of the original one. Each image has a homography that transforms to the original image. This is done so that

the interest points detected in the transformed image can be mapped to the original image and then measure how well the descriptors can describe the same patch. The transformation between the images is, however, sufficient enough to introduce noise in the detected interest points. Therefore corresponding matches below a certain threshold are only accepted for further computation. This threshold is also known as the overlap error. The recall and precision are then measured for all the corresponding regions of the transformed image and the original image, generating the performance of the descriptor.

The proposed novel COSFIRE (Combination of Shifted Filter Responses) descriptor is based on the COSFIRE filters which has been found highly effective in various object recognition applications [14] [15] [13]. COSFIRE is a trainable filter which is used for keypoint detection and pattern recognition. The main idea of a COSFIRE filter is to model an arrangement of contour parts that form a shape of interest. The response of this filter in a given point is computed as a function of the shifted responses of simpler filters. Using shifted responses of simpler filters, such as Gabor filters (that is a linear filter which is used for edge/line detection), corresponds to combining their respective supports at different locations to obtain a more sophisticated filter with a bigger support. The specific function that is used to combine filter responses in this filter is the geometric mean. Different COSFIRE filters may be configured to be selective for different number of contour parts [12]. For instance, a COSFIRE filter selective for corners might have lower number of involved contour parts than a COSFIRE filter that is selective for a pedestrian. A descriptor must always generate a vector with the same length irrespective of the complexity of the concerned feature. With this in mind, in this project we propose to convert the data structure that describes a COSFIRE filter into a descriptor that generates fixed length vectors.

1.3 Scope

The scope of this project is to create a COSFIRE descriptor representation for each keypoint detected. The COSFIRE descriptors will then be implemented within the Bag of Visual Words image classification model. The performance of the resultant

COSFIRE descriptor will then be compared with other popular descriptors that are currently being used such as the Scale Invariant Feature Transform (SIFT) [2], and BRISK [3] descriptors.

For this project, the CALTECH-256 data set [7] will be used as a primary source for benchmarking, evaluating the descriptor and analysing the descriptors' accuracy results. This dataset contains over thirty thousand images and 256 object categories, with at least eighty images per category. Given the time constraints, here we use 15 categories, namely: swan, butterfly, spider, toad, baseball-bat, billiards, binoculars, elephant, soccer ball, airplanes, chess-board, llama, car-side, treadmill and bulldozer.

1.4 Aims and Objectives

1.4.1 Aims

- To develop a COSFIRE descriptor and tune it for maximum accuracy in different image conditions and minimize the computational time.
- To compare its accuracy with other popular descriptors that are currently being used in this image classification technique using the same dataset.

1.4.2 Objectives

- To analyse the COSFIRE filter's configuration and output response parameters in depth, examining which variables are the most significant.
- To implement the novel COSFIRE descriptor.
- To evaluate the COSFIRE's descriptor performance under different image conditions such as blur, different viewpoints and lighting, while configuring the COSFIRE filters' configuration parameters and picking different combinations of variables for the novel descriptor. At each evaluation benchmark, the performance will be compared to other popular descriptors.
- To analyze current implementations of the Bag of Visual Words classification

model and choosing the most optimal one for this project and evaluate our novel COSFIRE descriptor.

1.5 Report Layout

The rest of this documentation is organized as follows: Chapter 2 will briefly define concepts that are important for this project, followed by an extensive research of current bag of visual words classification model implementations being used, including keypoint detectors and descriptors. Chapter 3 demonstrates the design of the solution proposed to this project followed by Chapter 4 where the algorithms of the structure of the previous chapter are explained in finer detail. Chapter 5 demonstrates the effectiveness of the COSFIRE descriptors compared to other descriptors by comparing the accuracy rates and evaluating the results. Chapter 6 summarizes this project and its results obtained and define scope for future work that was not covered in this project due to time constraints.

2 Background and Literature Review

2.1 Background

This section consists of brief definitions of important terms that used throughout this project such as keypoints, descriptors and bag of visual words model.

2.1.1 Keypoint detection and description

As described briefly in the problem definition, keypoints are salient image patches that contain rich local information. The task of a keypoint detector is to identify the image patches that, according to the detector, are important (for example edges and corners). If the image detector is scale invariant, interest points that are equivalent but of different scale, will be registered as the same keypoint. This results in features being highly distinctive [2]. Three types of keypoint detection methods are based on Difference-of-Gaussians [5], Hessian-Laplace [6] and MSER [16]. Once keypoints are

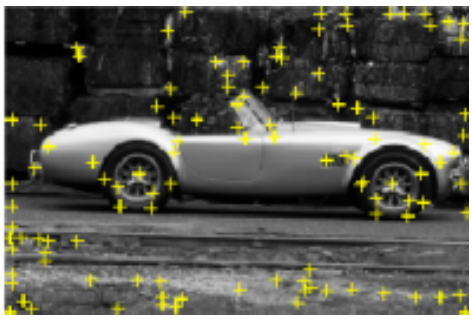


Figure 2: Detected keypoints of a test image [17].

detected, as seen in Fig. 2, the next step is to describe them in a rich representation format by extracting specific characteristics from the keypoints and their neighbourhoods. It is important to note that keypoint descriptors should be scale independent, robust against image transformations and independent of keypoint position. The most commonly used keypoint descriptors are the SIFT [2] and the SURF [8] descriptors.

2.1.2 Bag of Visual Words

In the bag of visual words image classification model, the feature vectors generated by the keypoint descriptor are grouped into a set of given number of clusters using a vector quantization algorithm (such as K-Means [18]). This process forms a codebook as seen in Fig. 3, which represents the visual features extracted from the training set. The next step consists of representing each image into a histogram of codewords, by first applying the keypoint detector and descriptor to every training image, and then matching every keypoint with those in the codebook. The result being a histogram where the bins correspond to the quantized keypoints in the codebook, also known as codewords, and the count of every bin corresponds to the number of times the corresponding codeword matches a keypoint in the given image. In this way, an image can be represented by a histogram of codewords. The histograms of the training images can then be used to learn a classification model.

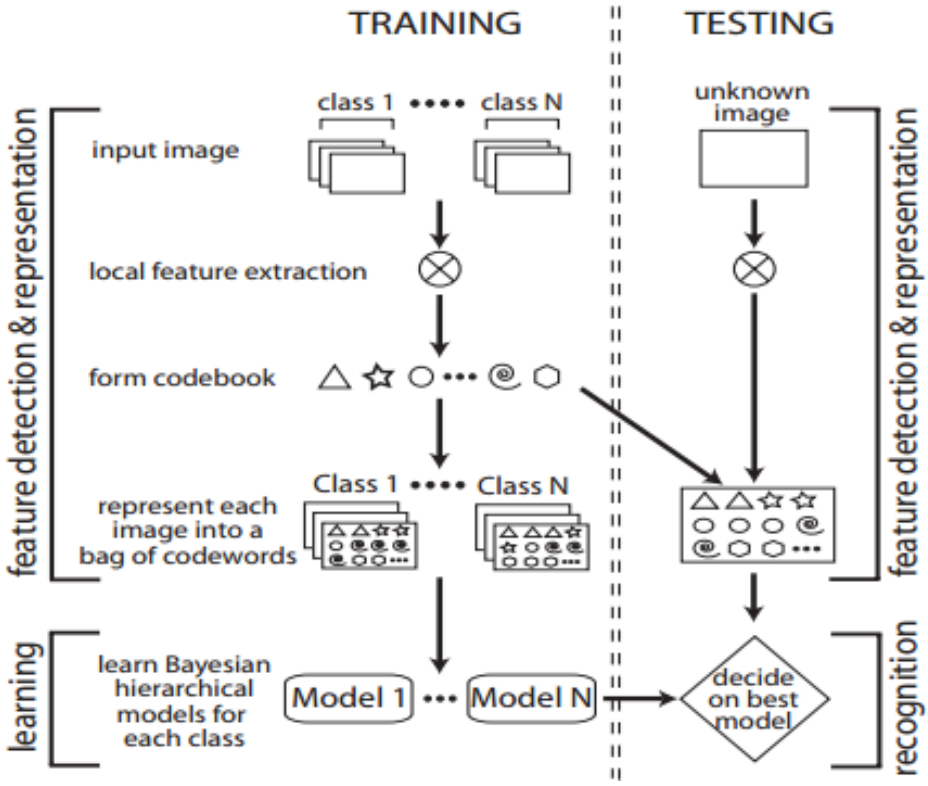


Figure 3: Schematic overview of the Bag of Visual Words algorithm. Taken from [19].

2.2 Literature Review

There are several approaches to image classification [20] [21] which are popular in computer vision. These are generally categorized on their primary common approach which include the basis of characteristic used, the basis of training samples used (which is divided into supervised and unsupervised classification), the basis of pixel information used and on the basis of spatial information. There are several classifiers on the basis of spatial information and these include contextual classifier and textual classifier.

The bag of visual words approach [19] evolved from the textual classification area, and due to its simplicity, which is one of its main advantage, became really popular due to requiring minimal computational power and achieving high classification accuracy rates. Several adaptations of this classification model were implemented in different projects with different choices of visual descriptors and classification models, all of which presented excellent evaluation results.

Figure 4: Variation of the threshold to detect MSER keypoints.

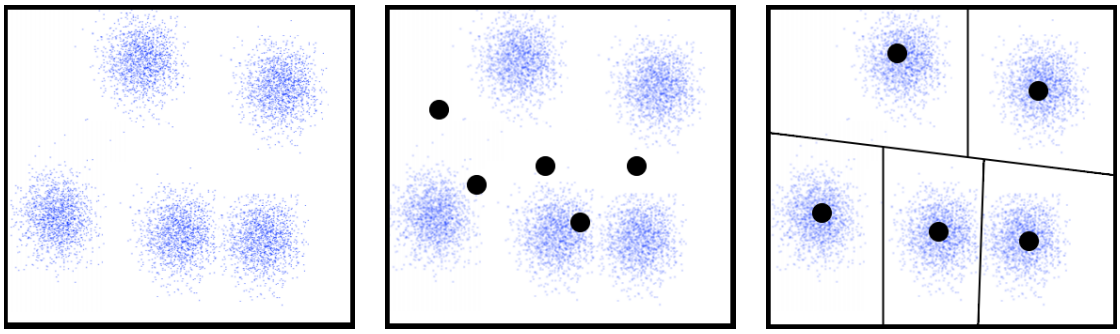
The first step of this classification model consists of detecting and extracting features from the training and testing dataset. This is done using a keypoint detector. There are several popular and effective keypoint detectors. These include the Harris-detector, Fast-Hessian detector, Hessian-Laplace detector and the MSER detector. In [8], the Fast-Hessian detector was compared to the difference of Gaussians (DoG) detector [2], Harris-Laplace detector [22] and Hessian-Laplace detector [22]. In recent comparisons, the MSER detector scored excellent results, achieving high repeatability rates on images with different transformations and even outperforming popular detectors in certain transformations [23].

MSER's concept of detecting keypoint regions is practically rather simplistic. Its first step is to detect extremal regions, which are all pixels in an image which are above and below a certain pre-defined threshold. The detector then chooses maximally stable parts from those detected image patches. This is done by varying the threshold as seen in Fig. 4 and keeping those regions which have a minimum change and therefore remaining stable during this process.

The keypoint detection phase is followed by the keypoint description phase, where each keypoint region detected is described accordingly using a vector of a certain fixed length (length of the vector varies from descriptor to descriptor). There are several popular visual descriptors which has scored high accuracy rates due to their distinctiveness and robustness features. These include the SURF descriptor [8], SIFT descriptor [2], GLOH descriptor and the BRISK descriptor [3]. The BRISK descriptor consists of a sampling pattern that is composed of circles concentric with the respective keypoint. Each sampling point is smoothed using Gaussian smoothing and depending if

its distance, between the points of the respective circle, is below or above a pre-defined threshold, is categorized into short-distance or long-distance set. The long-distance set is used to determine the orientation of the keypoint and the short-distance set is used to compare intensities that essentially build the BRISK descriptor. The SIFT descriptor is a 3D histogram of gradient locations and orientations where the contribution to the location and orientation bins is weighted by the gradient magnitude that yields a 128 dimensional vector [2]. It uses difference-of-Gaussians to find keypoints that are invariant to scale and orientation, followed by dividing each keypoint region into 4x4 grid cells and computes the orientations of the grid to generate a histogram for each cell. The SURF descriptor [8] is based on the same concepts of SIFT but using different algorithms to achieve its goal. SURF descriptor calculates the Laplacian of Gaussian by convolution with box filters. This leads to a huge performance advantage since convolution with box filters can be calculated with Summed Area Tables (algorithm that returns the sum of values in a particular grid) and this can also be done in parallel.

After the keypoint detection stage is completed, the next step is to generate the codebook so that each image in the training and testing dataset can be represented using codewords from the visual dictionary. Since the keypoints do not contain any labels, an unsupervised method is required to compute the vocabulary. Clustering algorithms are used specifically for this purpose, where vectors are clustered into K clusters defined by the user, using a pre-defined distance metric. Hierarchical clustering is one clustering algorithm which constructs nested partitions to form a tree of clusters using either bottom-up or top-down strategy. However this clustering method has a noticeable weakness which makes it less practical. Its distance measurements are sensitive to noise and outliers. A popular clustering algorithm that is widely used within the Bag of Visual Words model is the K-Means clustering algorithm, due to its simplicity and robustness to noise. After plotting all the data points in N dimensional space, where N is the vector size, it randomly intializes K cluster centers as seen in Fig. 5(b) (for clarity purposes we set N=2 in Fig. 5) and assigns each vector to each cluster. The algorithm then calculate the means of each centroid and update their positions.



(a) (b) (c)

Figure 5: This figure displays the 2-D data points in 2 dimensional space. The black dots displayed in (b) are the initial centroids locations, chosen randomly and (c) illustrates when the K-Means algorithm converges.

These steps are repeated until the cluster centers no longer change (converges).

Several implementations of the Bag of Visual Words model took different approaches to detect and describe interest regions [24] [25] [26] [27]. Many of the proposed descriptors are distribution-based [28], that is, they use histogram to represent different characteristics. Most popular descriptors that are commonly used in the bag of words paradigm are the GLOH [1], LBP [28] , SIFT [2] and SURF [8]. The GLOH descriptor [1] is similar to SIFT. This descriptor replaces the Cartesian location grid used by the SIFT with a log-polar one, and applies PCA to reduce the size of the descriptor [1]. The local binary pattern (LBP) [28] is another descriptor which is a powerful illumination invariant texture primitive. The histogram of the binary patterns computed over a region is used for texture description. The LBP operator describes each pixel by the relative grey levels of its neighbouring pixels. If the grey level of the neighbouring pixel is higher or equal, the value is set to one, else to zero [28]. In [8] a scale and rotation invariant interest point detector and descriptor SURF (Speeded up Robust Features) is introduced. It approximates and even outperforms in certain cases other image descriptors with respect to repeatability, distinctiveness and robustness, and it is computed and compared much faster as well. The detector is based on the Hessian matrix [29] [6] but uses a very basic approximation, named Fast-Hessian detector. This was achieved by relying on integral images for image convolutions. It builds on the strengths of the leading existing detectors and descriptors

and by simplifying these methods to the essential. Since only 64 dimensions are used, the time for feature computation and matching is significantly reduced and the robustness is also increased. SURF includes a new indexing step, based on the sign of the Laplacian [8], in which increases the matching speed and the robustness of the descriptor.

Most commonly used classifiers for the bag of visual words model as can be seen in several implementations [30] [27] [31] include Support Vector Machine [32], Nearest-neighbour Classification [33] and Artificial Neural Network (ANN) [34]. Artificial Neural Network (ANN) [34] is a classification method where it is an interconnected group of nodes that are of similar character to the vast network of neurons in a brain. An ANN consists of a sequence of layers, and in turn each layer consists of a set of neurons. All neurons of every layer are linked by weighted connections to all neurones on the preceding and succeeding layers. It uses a non-parametric approach and the performance and accuracy depends upon the network structure and the number of inputs. Several advantages of ANN include being a non-parametric classifier, being universal functional approximator with arbitrary accuracy and consisting of data driven self-adaptive technique, where it also handles efficiently noisy inputs. However, ANN has several disadvantages as well. These include being semantically poor (which means that although certain tasks can be achieved, it is not that easy to explain how the result of that task was achieved) and training being time consuming and difficulty on choosing the right type of network architecture. Several deep learning approaches evolved from Artificial Neural Networks such as BoWDNN-R and BoWDNN-S [35]. In deep learning approaches each layer extracts features from the output of the previous layer, and a feature is learnt hierarchically from pixels all the way to the classifier.

Support Vector Machine classifier [32] builds a hyper plane or set of hyper planes in a high dimensional space that is used for classification. A satisfying separation is achieved by the hyper plane that has the largest distance to the nearest training data point of any class. SVM uses non-parametric with binary classifier approach and can handle more input data very efficiently. Several advantages of SVM are that it gains flexibility in the choice of the form of the threshold while containing a non-linear

transformation and providing a good generalization capability. It is also simple to manage decision rule complexity and error frequency. Disadvantages of SVM includes the result transparency is low, the training being time consuming, the structure of algorithm being difficult to understand and the determination of optimal parameters is not easy when there is non-linearly separable training data.

Nearest-neighbour Classification [33] is another classification method that is based on K-Nearest Neighbours algorithm and is a non-parametric method used for classification. Given a set of n points, K-nearest neighbour lets you find the k closest points to a query point. Advantages of nearest-neighbour classification include having the cost of the learning process being zero and complex concepts can be learned by local approximation using simple procedures. Disadvantages of nearest-neighbour classification include the computationally being expensive to find the K nearest neighbours when the dataset is very large and performance being dependant on the number of dimensions that it has and therefore it is not a scalable approach.

3 Specification and Design

The main idea of how a COSFIRE filter is constructed and operates is fairly simple. The first thing that is required is an input image that would be used as the prototype pattern. The user selects the number of circles as seen in Fig. 6, which indicates the points of interests of the selected prototype. The points of interests are detected by symmetric Gabor filters, which are filters used for line detection. The responses of the Gabor filters are then blurred and shifted so that the filter allows a certain tolerance threshold to the orientation and position of the involved contour parts.

A COSFIRE filter takes input from the responses of orientation-selective filters, such as Gabor filters that are characterized by the the scale λ and the orientation θ. A Gabor filter is a bandpass filter which is used to detect edges/lines from images [36]. As explained in [12], the COSFIRE filter thresholds the input responses of the Gabor filters by a fraction t_1 of the maximum value. The thresholded responses of a Gabor filter are denoted by $|g_{\lambda,\theta}(x,y)|_{t_1}$. The data structure of a COSFIRE filter is a set of

4-tuples:

$$S_f = \{(\lambda_i, \theta_i, \rho_i, \phi_i) | i = 1...n_f\}$$

The subscript f stands for the local prototype pattern of interest. Every tuple (λ, θ, ρ, ϕ) represents the properties of a particular contour part. The parameter λ refers to the scaling factor, θ indicates the orientation while ρ and ϕ represent the polar coordinates of its location with respect to the center of the filter. The cardinality of the set S_f depends on the complexity of the given prototype pattern. For simple patterns, a COSFIRE filter results in a set of few tuples, but for a more complicated shape, it may result in high cardinality. This creates an obstacle for the construction of the novel COSFIRE descriptor since each keypoint generates different amount of tuples and therefore each keypoint would have different descriptor size. It is compulsory that every descriptor result in vectors of the same size in order to be able to compare them.

To tackle this problem, the main idea to solve this obstacle is to construct a histogram for each of the four parameters that describe the involved contour parts in a COSFIRE filter. Therefore with this setup, each description vector will be of equal size and can therefore be used in classification models such as BoVW.

A descriptor should be distinctive from one feature to another but also robust to changes in different images such as different viewing conditions, blurring, different lightening, zooming and different image compressions. To analyse the performance of

Figure 6: The local pattern pre-defined by the user to be used as a COSFIRE filter.

(a) (b)

Figure 7: Example of images from the viewpoint (Graffiti) category. (a) shows the original image while (b) illustrates the viewpoint transformation image of (a).

the COSFIRE descriptors under different configurations, the procedure as defined in [1] was used for this project. This methodology makes use of a dataset of categories that consists of images that are under different conditions such as having the same image from different angle. The first image of each category is the original one as illustrated in Fig. 7(a) while each other image is under a specific condition as seen in Fig. 7(b). Each transformed image has a transformation defined by a homography such that the interest points can be mapped to the first image. The dataset [1] consists of eight categories. In viewpoint changes category, the camera's view is changed by 20 to 60 degrees. In the blur and zoom category, the images are blurred and zoomed incrementally while the light category consists of changing the camera's position to adjust for different light and the JPEG category is produced using XV software (a program used to display and modify images) with the image quality varying from 40% to 2%.

The first step of the procedure to evaluate the visual descriptors consists of detecting keypoints in each image of the dataset, so that computation on the detected interest points can be followed. Several popular keypoint detectors were considered but Maximally Stable Extremal Regions (MSER) detector was used for this project. MSER is based on the idea of choosing regions which stay nearly the same through a different range of threshold values. These are also known as extremal regions. A threshold is then found for when an extremal region is maximally stable, that is the

local minimum of the relative growth of its square. MSER detector was chosen due to its superior properties over other detectors, which are:

- Invariance: MSER is invariant to affine transformations.
- Stability: Features are only selected if they show the same support over a range of different thresholds.
- Light and viewpoint change: MSER possesses the highest repeatability score for these types of conditions when compared to other detectors.

As already mentioned briefly in Section 1.2, the transformed image introduces a certain amount of noise in the detected keypoints, therefore the first step is to acquire the corresponding regions. This is done by measuring the overlap error between the detected. The overlap error is defined as the ratio of the intersection and union of the keypoint regions by:

$$\epsilon_S = 1 - (A \cap H^T B H)/(A \cup H^T B H)$$

where A is a keypoint from the original image, B is a keypoint from the transformed image, and H represents the homography of the transformed image. The procedure defined in [1] assumes that a keypoint region corresponds to the original image if the overlap error is less than 0.5 ($\epsilon_S < 0.5$). After retrieving the corressponding regions between the two images, the total correct matches out of the correspondences are then calculated so that the performance of the descriptor can be evaluated. The repeatability score is the total amount of overlap between the keypoints of two images, and the MSER with respect to the detected regions and the MSER detector achieved a high repeatability score in several categories [23]. Comparison and evaluation of different keypoint detectors is beyond the scope of this project since our focus is on the description of the keypoint.

Two keypoint regions detected by the MSER keypoint detector are a correct match if the distance between their respective descriptors does not exceed a defined threshold t. This threshold is then varied so we can plot different values of recall versus 1-

precision, where we obtain a curve of the performance of the respective descriptor. The higher the curve, the better the performance of the descriptor. The recall is the number of correctly matched keypoint regions divided by the corresponding keypoint regions between the two images. This is defined by:

$$Recall = \frac{\#Correct\ matches}{\#Correspondences}$$

The number of false matches in relation with the total number of matches is represented by 1-precision. With this result, we can deduce the number of correct matches by $\#correspondences.recall$ and the number of false matches by $\#correspondences.recall.(1 - precision)/precision$.

$$1 - precision = \frac{\#\ false\ matches}{\#correct\ matches + \#false\ maches}$$

The performance of a descriptor is then visualized with recall versus 1-precision graph where the higher the recall, the more robust the descriptor is to changes. Different COSFIRE configurations will be evaluated using this procedure and the most optimal configuration will be selected for the novel COSFIRE descriptor.

This COSFIRE descriptor will then be evaluated along with other popular visual descriptors on the CALTECH-256 dataset using the Bag of Visual Words (BOVW) image classification model. The structure of this classification model is described in Section 2.1.2. Since as it was concluded in [1] that keypoint detectors do not affect the performance of the descriptors, for this image classification model, we decided to use the SIFT detector as proposed by Lowe [2]. The keypoints are detected from a difference-of-gaussians scale space. A Gaussian scale space consists of a collection of images that has been smoothed as follows:

$$I_\sigma = g_\sigma * I, \sigma \geq 0$$

Scales are then sampled at logarithmic steps for each octave, where o_{min} is the first octave index. Difference-of-gaussians is obtained by the difference of two successive

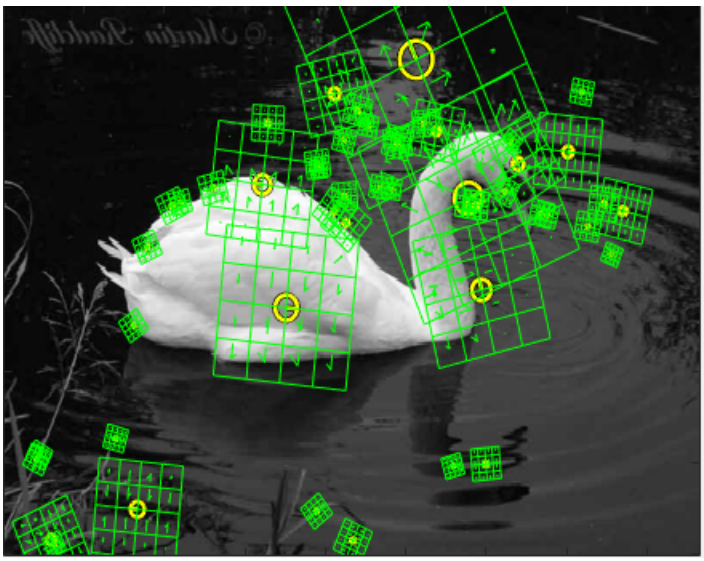


Figure 8: Features detected by SIFT's detector [2]. Yellow circle indicates the position of the detect keypoints. It's size indicates the scale at which the keypoint was detected and the line connecting the center to the circumference indicates the dominant orientation of the keypoint. The green boxes illustrates the histogram of gradients per each bin, which generates the SIFT descriptor.

scales of the Gaussian scale space:

$$DoG_{\sigma(o,s)} = I_{\sigma(0,s+1)} - I_{\sigma(o,s)}$$

where s is the number of scales per octave, σ is the scale of the image and o is the number of octaves. The images are then searched through scale and space for local extrema, where keypoint frames are constructed based on local extrema. However, a considerable amount of keypoints are noise, such as low contrast responses and responses close to edges, and should be filtered so they won't be a conflict for the performance of the descriptors. Local extrema of the difference-of-gaussians scale spaces which are negligible are often noise and therefore removed if its value is lower than the peak threshold t_p. The study in [37] explains in detail how peaks which are too flat do not yield reliable features and are thresholded using edge threshold t_e.

After the keypoints have been filtered, the COSFIRE and SIFT descriptors describe these keypoints. The COSFIRE descriptor creates a COSFIRE filter for each keypoint location, taking into consideration the scale which represents the size of the bounding

box around the keypoint and the dominant orientation as can be seen in Fig. 8. The resulting vectors of each descriptor are then concatenated and clustered using K-Means to form a visual vocabulary. The main idea of K-Means is to define K number of centers which later on will define the clusters. The initialization of the centers is crucial since intializing them at different places (randomly) will cause a different result. The work in [38] proposed an algorithm that initializes the centroids as far as possible from each other and has since seen as the currently best solution to initialize the centroids. The default Lloyd's algorithm [39] is used for this K-Means, which consists of attributing the closest cluster to each keypoint vector and set the position of each cluster to the mean of all the keypoint vectors belonging to that particular cluster until convergence.

After K-means forms a visual vocabulary of K number of *words*, which represents our keypoint vocabulary as depicted in Fig. 3, the next step is to represent our image using a fixed-size vector. When a particular image returns $x_1, x_2, .., x_N$ descriptions, their distances from each visual word of the vocabulary are measured, using Euclidean distance, and the closest visual word is assigned to each description. This is also known as hard quantization. A histogram is then constructed having the visual vocabulary as bins, and adding value to the appropriate bin for each visual word that is found closest to any of the keypoint descriptors in the given image.

SVM is commonly used in BoVW models [27] [31] [26] due to its superiority in performance over K Nearest Neighbour(KNN) classification. Therefore for this project the SVM classifier was used to classify the histograms using linear kernel. Since SVM works better with normalized histograms, the histograms are normalized beforehand to unit length.

A single histogram, generated from the vocabulary, for an image, results in information loss of where particular keypoints are potentially being situated through other images of the same category. In order to compensate to some extent for this limitation, this project considers several spatial tiling configurations, where the image is split into 2x2 and 3x3 spatial tiles, generating a histogram for each tile as seen if Fig. 9 and concatenating them afterwards.

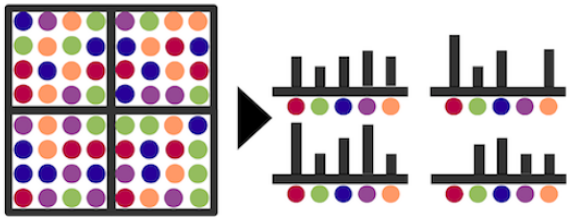


Figure 9: An example of 2x2 spatial tiling of an image. The circles illustrates the codewords that are formed through the K-Means clustering. Instead of assigning a histogram (having codewords as bins) per image, here we assign a histogram per spatial tile and then concatinating the histograms together.

4 Methodology

The COSFIRE filter's configuration of each keypoint detected within an image returns a set of 4-tuples (λ, θ, ρ and ϕ) in which this set varies in size according to how much information each keypoint contains. Due to different COSFIRE filters having different cardinality, we propose to construct a histogram for each variable to generate a fixed size COSFIRE descriptor. The first histogram consists of the λ values determined by the COSFIRE's filter pre-defined λ parameter with 5 bins, which are $2\sqrt{2}$, 4, $4\sqrt{2}$, 8, $8\sqrt{2}$. This is followed by the histogram of the orientations θ, having the pre-defined θ values with 16 bins (the parameter θ has possible values of 0, 0.4, 0.8, 1.2, 1.6, 2, 2.4, 2.7, 3.1, 3.5, 3.9, 4.3, 4.7, 5.1, 5.5, 5.9). A polar grid is then created to quantify the responses by their location (ρ_i,ϕ_i), using the maximum ρ value as the maximum boundary of the circle as seen in Fig. 10. The polar grid is divided into a number of pre-defined sectors and circles, producing several areas depending on the number of circles and sectors used. If for example the values of the sectors and circles for the COSFIRE descriptor are 4 and 1 respectively, it generates 4 polar grid areas as seen in Fig. 10.

Each area will be assigned with area ID and a check is then made that loops through all the areas, and if (ρ_i,ϕ_i) fits the criteria (located within the sector boundary and circle boundary), it will be assigned the respective area ID. Each tuple is then plotted on the polar grid according to the ρ and ϕ value, each value pair describing its location accordingly. A histogram is then generated having all the areas within the

polar grid as histogram bins and the values of how many tuples fall under each area respectively. All three histograms are separately normalized to l_2 in order to stabilize the values since the number of features detected for each keypoint are different for each feature. These histograms are then concatenated to form one distinct histogram for each keypoint, which would be the COSFIRE description for the respective keypoint patch. Initially the novel descriptor is constructed using $hist_\lambda + hist_\theta + hist_{seg}$, but different combinations of the parameters will be evaluated in the evaluation section (such as $hist_\theta + hist_{seg}$), resulting in the COSFIRE descriptor varying in size. The number of sectors and circles will also be tuned to achieve the maximum performance.

If we use 5 distinct λ values, 12 distinct θ values and 4 polar grid areas we end up with a descriptor of size 21 (5 + 12 + 4). The length of the descriptor can be controlled by changing the possible parameter values of λ, θ, and the number of areas in the polar grid. Different configurations of these parameters will be analyzed and re-designed to improve the performance of the COSFIRE descriptor in the evaluation chapter. These include increasing or decreasing the number of segments in the polar grid, trying different combinations of variables in conjuction with each other. We will then compare the effectiveness of this novel descriptor with the state of the art SIFT descriptor to analyse how our descriptor is performing.

Several COSFIRE descriptor's configurations were implemented and evaluated throughout the procedure proposed by [1] to be evaluated for their performance. The first steps of this procedure consisted of acquiring the MSER features' information

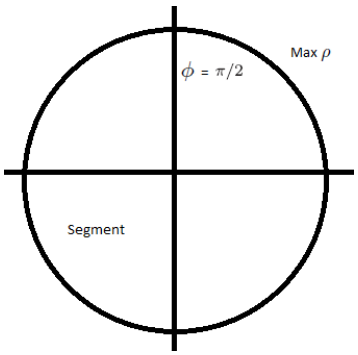
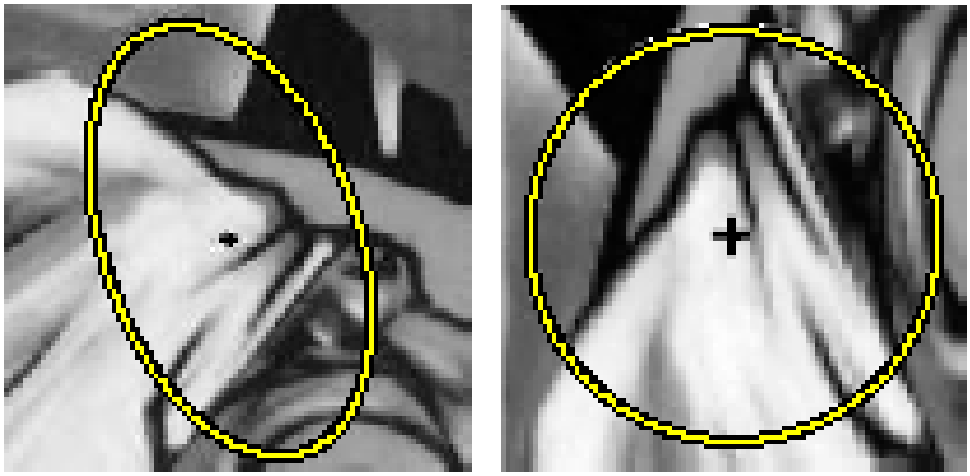


Figure 10: Polar grid consisting of 4 sectors and 1 circle, resulting in 4 segments. Each keypoint's location (ρ_i,ϕ_i) is plotted on this grid to aggregate the keypoints according to their locations.



(a) (b)

Figure 11: These figures show the detected feature (a) as detected by the MSER detector and the normalized region of that feature, in which it become scale and rotation invariant.

of each image. This information consisted of the Cartesian location and the ellipse parameter's values (since the features are detected in the form of an ellipse). Since different keypoint regions having different scale, this creates an obstacle in measuring the performance of the descriptor purely for photometric transformations. This procedure tackles this problem by, using affine covariant construction, maps all the ellipses of the keypoint regions to a circular patch of constant radius. This process makes the keypoint patch scale and affine invariant but not invariant to photometric transformations, therefore exactly what we need as can be seen in Fig. 11.

Although the keypoint patches are now invariant to geometric transformations and scale, they are still not invariant to rotation, and this invariance is also required so that the performance of descriptors are purely based on photometric transformations. This is achieved by normalizing the image patch with the direction of the dominant gradient. The COSFIRE descriptors along with the SIFT descriptor then describe the image patch and store their descriptions to the respective feature. The descriptors performance is then measured by varying the distance threshold value t between the descriptors as described in Section 3.

The COSFIRE desriptor with the highest performance is then chosen as the novel COSFIRE descriptor. This descriptor along with the SIFT descriptor are then im-

plemented in the BoVW image classification model to analyse their performance at generating and assigning codewords that are eventually used to define the images of the dataset selected.

The categories from the Caltech dataset [7] where chosen carefully such that they contain at least 108 images each category to expand the training dataset as much as possible to provide a more concise accuracy result. Each category is then divided in two parts, the training and the testing dataset, where the ratio would be 70 images for training and 38 images for testing. This results in accuracy rates being low in variance due to 70 images for training, therefore more precise. Before classifying images, most implementations resize images to be of a certain height so the features from one image to another will be roughly the same scale. This project followed IMAGENET [40] height's resize value and resized the images to have a maximum side length of 300 pixels.

The feature detector [2] that is used in this classification model, is mainly controlled by 3 parameters [41] [37] which where tuned as illustrated in Fig. 12, and these are the Peak Thresh, Edge Thresh and First Octave Index:

- By default, the detector starts each image's scale space at full resolution. However if the First Octave Index is set to -1, the scale space starts at a higher resolution and hence very small features are extracted which was shown to be useful, therefore it was set to -1 for this project.

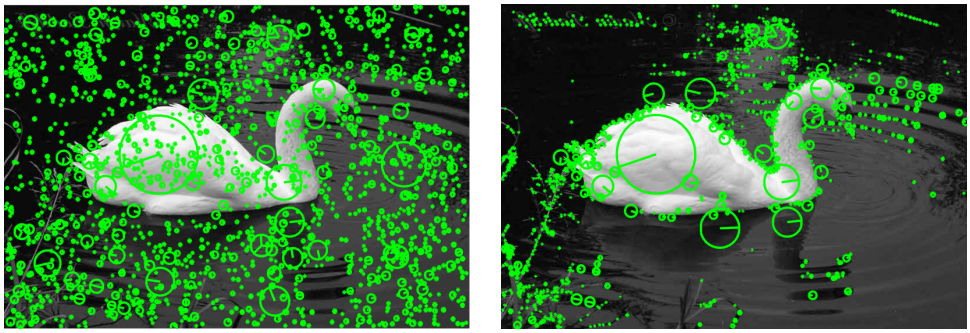
- The edge threshold removes peaks of the DoG scale space where the curvatures are negligible. This is done since if such peaks are not removed, they would yield badly localized frames). When this parameter was being evaluated, the value 60 fitted most to this dataset.

- The peak threshold removes the peaks of the Difference of Gaussian(DoG) scale space that have little value. Peak threshold of value 5 was the most suitable for this dataset.

The idea of partitioning a set amount of descriptors into a number of known regions is a process to build the codebook, a collection of codewords. The initial number of

codewords for the vocabulary was set to 1000 since it produces the peak accuracy in various implementations and only increase slightly when it reaches 5000 [26]. Although K-means has its advantages, it also has its downsides. One of the most noticeable downsides is that its sensitive to the initialization of centroids. The work in [38] proposed an algorithm that initializes the centroids as far as possible from each other, which tackles the problem of sensitivity of initialization of centroids.

The image is then divided into NxN spatial tiling, where a histogram, the size of the vocabulary, is then created for each spatial tile, containing the sum of the codewords found in the respective tile. The histograms of all the spatial tiles are then concatenated together to form one histogram to represent the image. This histogram is then normalized to unit length. SVM is then used to classify the images, by representing the images with the concatenated histograms of codewords in N dimensional space, using linear kernel with default C value of 1. The C parameters controls of how much misclassification is allowed in the SVM classification. A small value of SVM yields a large margin hyperplane which is able to differentiate the categories more clearly. This classifier was implemented using the LIBSVM library [42].



(a) (b)

Figure 12: This figure illustrates features extracted from a swan image with default SIFT detector values and the optimized values. (a) shows the image with the default SIFT detector's parameters and (b) with the tuned parameters.

5 Evaluation

For each keypoint detected within an image, a COSFIRE filter is constructed to describe that keypoint patch. The idea is that this novel descriptor is able to describe the image patch in a distinctive manner while being robust to noise. Therefore the COSFIRE filter had to be configured to cater for such task. These included minimal configurations optimizations such as adjusting the lambda (using 5 values was found to be optimal for the COSFIRE filter as illustrated in Section 4) and orientation (using 16 values for each COSFIRE filter as mentioned in Section 4) pre-defined values to be used as a standard for the COSFIRE filter.

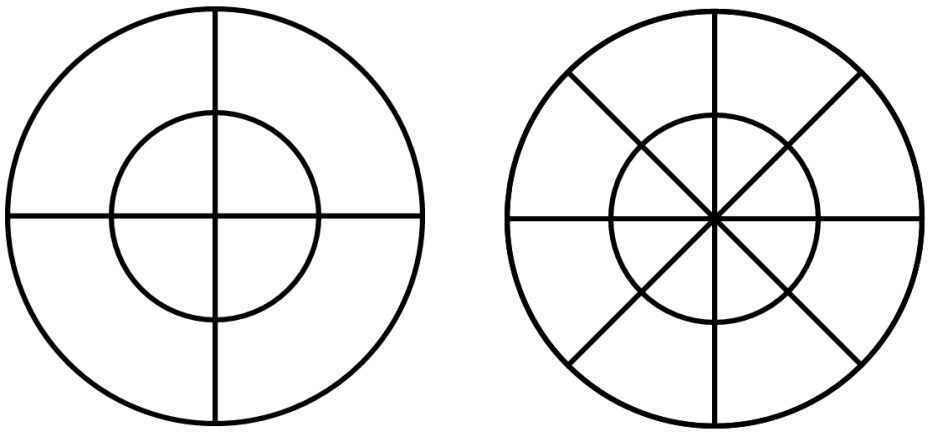
This section is mainly divided into three parts. The first part covers several different COSFIRE descriptors that were configured and evaluated based from the set of 4-tuples of a COSFIRE filter. These configurations are explained in detail of how they are constructed, including the final descriptor size, which will be the main identification of the respective COSFIRE descriptor. The second part will consist of evaluating these descriptor configurations on the procedure proposed by [1], which includes different image conditions and each descriptor will be tested for it. The third part of this section will then cover the evaluation of the bag of visual words classification model.

The COSFIRE filter has mainly 4 output parameters for each tuple for each COSFIRE filter as described in the design section which are $(\lambda_i, \theta_i, \rho_i, \phi_i)$ where λ refers to the scaling factor the interest point was found in, θ being its orientation, and (ρ, ϕ) being its respective location. The first COSFIRE descriptor consists of constructing a histogram for λ, θ and for the location values found in the polar grid using 2 circles and 4 sectors as seen in Fig. 13(a) as $hist_\lambda + hist_\theta + hist_{polargrid(4s2c)}$, where the histograms sizes consists of $5 + 16 + 8$ values which results in the COSFIRE descriptor to be of size 29 (The size of the descriptor will be its identification for this evaluation section). Due to the polar grid having large areas, the second configuration of the descriptor consisted of increasing the number of sectors to 8 as seen in Fig. 13(b), thus having $5 + 16 + 16$ which results in a descriptor size of 37.

Although the polar grid was a good optimization to cluster tuples according to

locations, these two configurations lacked the quantization of λ and θ values over a keypoint region, since producing only one histogram of λ and θ for the whole patch is not very specific and leads to a decrease in the distinctiveness of the descriptor. To tackle this lack of specificity, the next proposed descriptor takes into consideration the location of the λ and θ values found across the keypoint region. Currently the number of ρ values has been configured to 5. The next descriptor was constructed by generating a histogram of the λ values and a histogram of the θ values for the tuples found on each ρ circle. This causes the descriptor to increase drastically in size since we are producing a λ histogram (size:5) and θ histogram(size:16) for each circle, which adds up to $30 + 96 + 16 = 142$ descriptor size.

This can be seen as the first attempt at specifying the locations of where certain scales and orientations which are found within a keypoint region. The next step of this configuration phase consisted of determining which parameter is playing the most important part in the COSFIRE descriptor and if some parameter combinations perform better when dropping other parameters from the histogram equation. Therefore the next descriptor was constructed from orientations of each ρ circle and the histogram produced from the polar grid while ignoring the λ values, resulting in $96 + 16 = 112$ descriptor size. The λ histogram was ignored because it only consists of very few



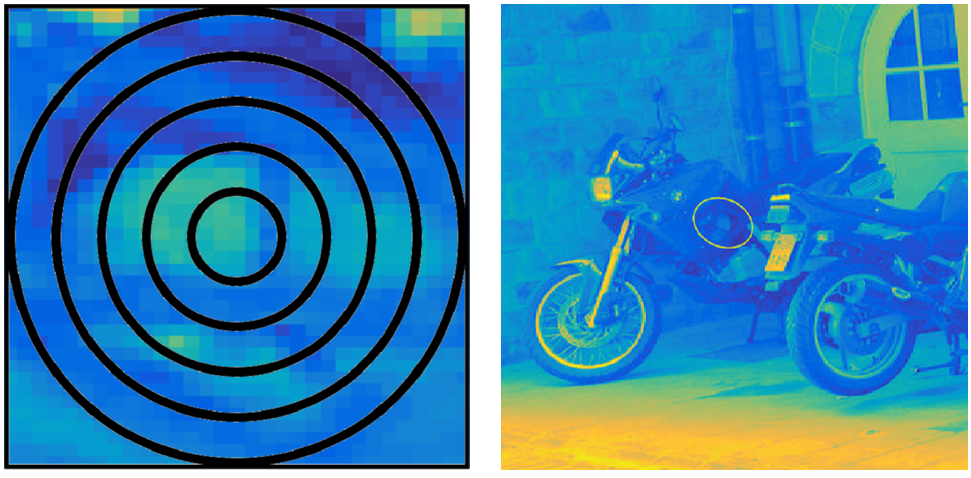
(a) (b)

Figure 13: Different configurations of the polar grid. (a) illustrates the polar grid for 4 sectors 2 circles while (b) displays polar grid for 8 sectors 2 circles.

values and might be interfering with the distinctiveness of the descriptor. This was followed by the descriptor constructed solely of orientations, to analyse how important the orientations are in the descriptor (descriptor size: 96).

The descriptor with $hist_{\lambda} + hist_{\theta} + hist_{polargrid(8s2c)}$ with histograms of λ and θ for each ρ value was proven to outperform the other descriptors in different image conditions, therefore the next phase was to optimize any configuration parameters. Different configurations were evaluated for the ρ parameter values, such a predefined vector of values and a constructed vector of values in steps of 2, but the one that outperformed the others was by dividing the maximum length of the local COSFIRE filter into N circles (let's name the result STEP), and acquiring rho values from 0 to the maximum ρ value in STEP distance. The STEP parameter that essentially defines the ρ values, which are the circles where the bank of Gabor filters responses are detected, was set to an initial value of 5. Other different values were evaluated such as 10 and 15 for this STEP parameter. This increased the size of the descriptor even more since histograms are being generated for each circle. When the STEP parameter is set to 10, the COSFIRE descriptor increased to a size of 247 and when it is set to 15 the size increases to 352.

Although the latter descriptor achieved the highest performance, it is worth noting



(a) (b)

Figure 14: The keypoint region detected in image (b) is divided by the pre-defined STEP variable (5 in this case) as illustrated in (a), producing 5 ρ values, each value being a circle.

Table 1: Summary of the proposed COSFIRE descriptors with different configurations

Descriptor size	**Description**
29	$hist_\lambda + hist_\theta + hist_{polargrid(4s2c)}$.
37	$hist_\lambda + hist_\theta + hist_{polargrid(8s2c)}$.
142	$hist_\lambda + hist_\theta$ for each ρ value (STEP = 5) + $hist_{polargrid(8s2c)}$.
96	$hist_\theta$ for each ρ value (in steps of 5).
112	$hist_\theta$ for each ρ value (in steps of 5) + $hist_{polargrid(8s2c)}$.
247	$hist_\lambda + hist_\theta$ for each ρ value (STEP = 10) + $hist_{polargrid(8s2c)}$.
352	$hist_\lambda + hist_\theta$ for each ρ value (STEP = 15) + $hist_{polargrid(8s2c)}$.
336	$hist_\lambda + hist_\theta$ for each area in the polar grid (STEP = 15 for ρ)

the high dimensionality problem, which causes the curse of dimensionality. Several problems might arise when trying to classify images with high dimensional descriptor, such as *Hughes phenomenon*, which states that as dimensionality increases, the predictive ability decreases. At this phase there is two known issues to the COSFIRE descriptor, the first issue being the curse of dimensionality as mentioned above. The second issue is the fact that although we were taking histograms of λ and θ per each circle, we still don't know if certain λ and θ values are found at the top or at the bottom of each circle for example, and that is a huge obstacle to the descriptor at its goal of being distinctive.

Our next configured descriptor consisted of merging the λ and θ histograms with the polar grid's areas location of the tuples detected in a keypoint region. In other words we create a λ and θ histogram for each area generated by the polar grid, and ignoring the polar grid histogram since we are already applying the location to the other histograms. This COSFIRE descriptor has the size of 336, which is still high dimensionality, but each dimension provides much more information about the location of the tuples within a COSFIRE filter. This creates a huge advantage for us since the dimensionality is reduced and the fact that each dimension contains more useful information makes it optimal for classifiers to classify the training dataset.

Table 1 displays a summary of the proposed COSFIRE descriptors that will be evaluated under different image conditions using the method proposed by [1], as explained in Section 3 and 4.

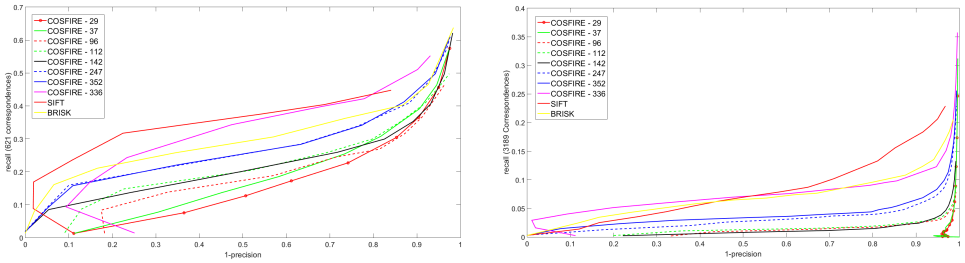
The dataset used by [1] consists of 8 categories and are divided into blur textural

Figure 15: A sample of images from the dataset used [1] for different image conditions evaluation.

scenes, blur structural scenes, JPEG compression on a structured scene, light on a structured scene, viewpoint change in textural scene, viewpoint change in structural scene, zoom and rotation on textural scene and zoom and rotation on structural scene. A structure scene is an image which contains objects while a texture scene contains mostly texture repetition. In order to determine the performance of a descriptor the authors in [1] proposed to obtain the number of correct matches for when a descriptor obtains 400 nearest neighbour matches. This is achieved by setting the visual descriptor to 400 nearest neighbour matches using the corresponding regions of the images defined by the overlap criterion as defined in Section 3.

The performance of the descriptors was first measured on the blur categories. These contains structured images, where for each image blur was introduced, incrementing it for each sequential image, by changing the focus of the camera used. The results are displayed in Fig. 16(b). It was noticed that as the blur increases, the edges in the images start to disappear, which has an impact on the performance of the COSFIRE descriptors and even on the SIFT descriptor as seen on Fig. 16(a).

COSFIRE-336 clearly outperformed the other configured COSFIRE descriptors and matches SIFT's performance as the precision increases, while also surpassed



(a) (b)

Figure 16: Blur categories: Recall vs 1-Precision plots for the COSFIRE descriptors and SIFT descriptor on structured images (a) and textured images(b).

BRISK's performance both in textural and structural categories. COSFIRE-336's performance was consistent in both structural and textural images. It is worth nothing that COSFIRE-112 performed better than COSFIRE-142, which can be interpreted that λ values are not the most important parameter in the COSFIRE descriptor when evaluating blurred images. COSFIRE-352 and COSFIRE-247 achieved the same performance in structural images but COSFIRE-352 performed better in textural scenes which is reasonable due to the fine details found in textural images. Table 2 displays the correct matches achieved by the descriptors for the 400 nearest neighbour matches of each descriptor.

The descriptors were then evaluated on images with different viewpoints,where

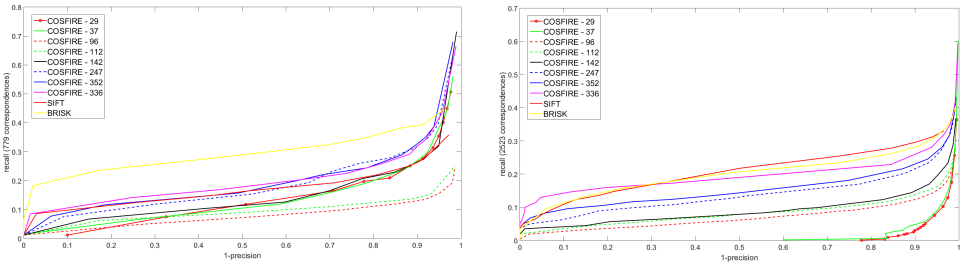
Table 2: Matching table for blur structural category with a fixed number of 400 matches.

Descriptor	**Recall**	**1-P**	**C-M**
COSFIRE-29:	0.36	0.72	98
COSFIRE-37:	0.37	0.72	101
COSFIRE-96:	0.45	0.66	122
COSFIRE-112:	0.49	0.63	132
COSFIRE-142:	0.47	0.64	128
COSFIRE-247:	0.55	0.58	148
COSFIRE-352:	0.55	0.58	148
COSFIRE-336:	0.64	0.51	**174**
SIFT:	0.67	0.49	**181**
BRISK:	0.65	0.51	**175**

Table 3: Matching table for blur textural category with a fixed number of 400 matches.

Descriptor	**Recall**	**1-P**	**C-M**
COSFIRE-29:	0.01	0.94	23
COSFIRE-37:	0.01	0.94	22
COSFIRE-96:	0.04	0.84	63
COSFIRE-112:	0.04	0.82	71
COSFIRE-142:	0.04	0.83	67
COSFIRE-247:	0.08	0.68	127
COSFIRE-352:	0.09	0.64	143
COSFIRE-336:	0.16	0.38	**249**
SIFT:	0.17	0.32	**271**
BRISK:	0.14	0.45	**221**

Descriptor = Descriptor's identity. 1-P = 1-Precision. C-M = Correct Matches obtained for a fixed number of 400 nearest neighbour matching.



(a) (b)

Figure 17: Viewpoint categories: Recall vs 1-Precision plots for the COSFIRE descriptors and SIFT descriptor of images from different viewpoint angles on structured images (a) and textured images(b).

the camera varies from a front-parallel view varying up to 60 degrees. This category was pretty interesting for the COSFIRE descriptors. COSFIRE-352 and COSFIRE-247 matches SIFT performance in the structural images, with COSFIRE-352 scoring higher than COSFIRE-336 in some cases as can be seen in Fig. 17. However the BRISK descriptor outperformed all other descriptors in the structural images. It is important to note the inconsistancy of the COSFIRE-96 and COSFIRE-112. The lack of the λ values really affected both descriptors when dealing with structural images from different viewpoints. In the textural images, the COSFIRE-336 maintained his consistent performance, being close to the SIFT state-of-art descriptor.

Table 4: Matching table for the viewpoint structural category with a fixed number of 400 matches.

Descriptor	**Recall**	**1-P**	**C-M**
COSFIRE-29:	0.24	0.81	75
COSFIRE-37:	0.23	0.82	73
COSFIRE-96:	0.20	0.84	62
COSFIRE-112:	0.21	0.84	65
COSFIRE-142:	0.28	0.78	87
COSFIRE-247:	0.33	0.74	104
COSFIRE-352:	0.36	0.71	**114**
COSFIRE-336:	0.34	0.73	**106**
SIFT:	0.35	0.73	**109**
BRISK:	0.65	0.49	**203**

Table 5: Matching table for viewpoint textural category with a fixed number of 400 matches.

Descriptor	**Recall**	**1-P**	**C-M**
CCOSFIRE-29:	0.05	0.83	69
COSFIRE-37:	0.06	0.79	86
COSFIRE-96:	0.14	0.52	194
COSFIRE-112:	0.15	0.47	210
COSFIRE-142:	0.16	0.45	222
COSFIRE-247:	0.21	0.26	296
COSFIRE-352:	0.23	0.21	316
COSFIRE-336:	0.27	0.05	**381**
SIFT:	0.26	0.10	**361**
BRISK:	0.25	0.12	353

Descriptor = Descriptor's identity. 1-P = 1-Precision. C-M = Correct Matches obtained for a fixed number of 400 nearest neighbour matching.

The performance of the configured COSFIRE descriptors on the JPEG compression images was not too good as can be seen in Fig. 18 when compared with SIFT. Although COSFIRE-336 kept close to SIFT's performance, SIFT performed slightly better than the COSFIRE descriptors in this category. COSFIRE-142 started with a stable performance but COSFIRE-96 and COSFIRE-112 cought up with the performance as the precision increased. The performance of COSFIRE-352 and COSFIRE-247 was more or less the same, which makes COSFIRE-247 more efficient, being 105 dimensions smaller. The BRISK descriptor achieved the best score in this category.

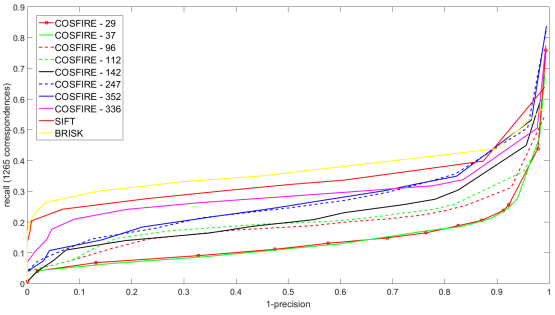


Figure 18: Recall vs 1-Precision for JPEG compression category in a textural scene.

Table 6: Matching table for JPEG category with a fixed number of 400 matches.

Descriptor	**Recall**	**1-P**	**C-M**
COSFIRE-29:	0.35	0.46	215
COSFIRE-37:	0.36	0.46	218
COSFIRE-96:	0.37	0.44	226
COSFIRE-112:	0.38	0.42	233
COSFIRE-142:	0.40	0.39	246
COSFIRE-247:	0.47	0.28	288
COSFIRE-352:	0.48	0.28	289
COSFIRE-336:	0.51	0.22	**311**
SIFT:	0.55	0.17	**333**
BRISK:	0.63	0.05	**380**

In the light category, COSFIRE-352's performance finally shined over COSFIRE-247, having enough correct matches to nearly match with COSFIRE-336 as seen in Table 7. This category continues to show the inconsistency of the COSFIRE-96 and COSFIRE-112 descriptors, where as can be seen in Fig. 19, their performance is way below the performance of COSFIRE-142, and even below COSFIRE-29 and COSFIRE-37 in certain threshold values. Although COSFIRE 142's performance is below COSFIRE-336 and SIFT, it is still important to note that it maintains its consistancy, which can be concluded that λ values are important for the configuration

process of the COSFIRE descriptor. COSFIRE-247 and COSFIRE-352 descriptors performed better than BRISK in this category.

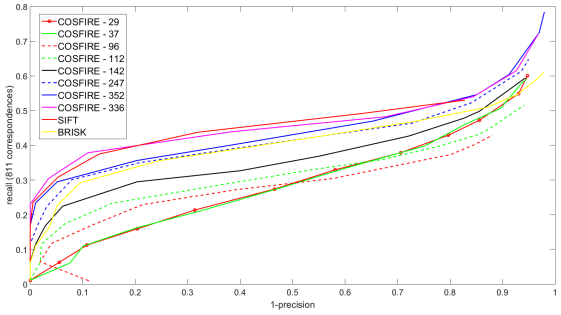
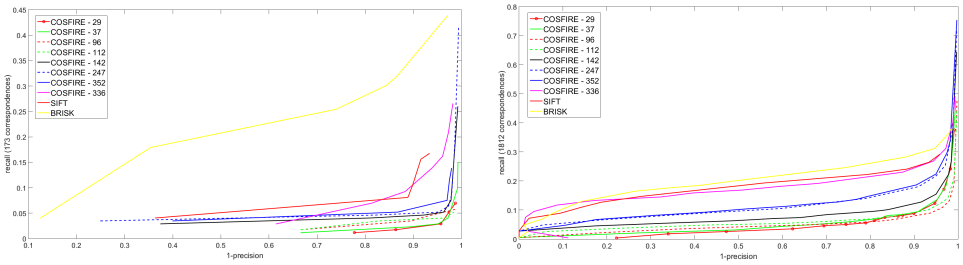


Figure 19: Recall vs 1-Precision for light category in a textural scene.

Table 7: Matching table for Light category with a fixed number of 400 matches.

Descriptor	**Recall**	**1-P**	**C-M**
COSFIRE-29:	0.59	0.42	230
COSFIRE-37:	0.62	0.40	240
COSFIRE-96:	0.54	0.48	208
COSFIRE-112:	0.56	0.46	218
COSFIRE-142:	0.64	0.38	247
COSFIRE-247:	0.68	0.34	263
COSFIRE-352:	0.70	0.32	272
COSFIRE-336:	0.72	0.30	**279**
SIFT:	0.73	0.29	**282**
BRISK:	0.76	0.27	**294**

The performance of COSFIRE-336 was pretty close to that of SIFT in zoomed and rotated structured images as can be seen in Table 9. In this category, as can been in Fig. 20, it is shown clearly how λ and θ values are important to be grouped by specific areas as how COSFIRE-336 is designed. This can be speculated from the performance of the other COSFIRE's descriptors. It is also worth noting that in this category, in the structural images, COSFIRE-29 and COSFIRE-37 got a performance equal to that of COSFIRE-96 and COSFIRE-112, which again proves that λ values in



(a) (b)

Figure 20: Zoom and rotation categories: Recall vs 1-Precision plots for the COSFIRE descriptors and SIFT descriptor of rotated and zoomed structural images (b) and textural images (a).

Table 8: Matching table for the zoom and rotation textural category with a fixed number of 400 matches.

Descriptor	**Recall**	**1-P**	**C-M**
COSFIRE-29:	0.07	0.97	8
COSFIRE-37:	0.07	0.97	8
COSFIRE-96:	0.07	0.97	8
COSFIRE-112:	0.07	0.97	8
COSFIRE-142:	0.12	0.96	13
COSFIRE-247:	0.13	0.96	14
COSFIRE-352:	0.13	0.96	14
COSFIRE-336:	0.19	0.93	**21**
SIFT:	0.22	0.92	**25**
BRISK:	0.44	0.85	**49**

Table 9: Matching table for zoom and rotation structural category with a fixed number of 400 matches.

Descriptor	**Recall**	**1-P**	**C-M**
COSFIRE-29:	0.11	0.76	96
COSFIRE-37:	0.12	0.72	112
COSFIRE-96:	0.11	0.74	104
COSFIRE-112:	0.13	0.70	121
COSFIRE-142:	0.16	0.64	145
COSFIRE-247:	0.23	0.49	204
COSFIRE-352:	0.23	0.47	211
COSFIRE-336:	0.30	0.32	**273**
SIFT:	0.32	0.27	**291**
BRISK:	0.35	0.21	**317**

Descriptor = Descriptor's identity. 1-P = 1-Precision. C-M = Correct Matches obtained for a fixed number of 400 nearest neighbour matching.

a keypoint region should always be considered.

The average false positive rate, false negative rate, true positive and the f-measure of all the images per category of the 400 nearest neighbours matches was then calculated so that the descriptors' performance at each condition transformation is taken into account. Due to COSFIRE-29 and COSFIRE-37 low performance, they were eliminated from the following table. The table's values represent how a descriptor's performance change, over different image conditions changes. From Table 10, it became clear that COSFIRE-336 is the most optimal COSFIRE descriptor. However it can be observed that COSFIRE-96 and COSFIRE-112 outperform COSFIRE-142 in certain categories, but their performance lacks consistency.

Table 10: Matching table for the average score for each category with a fixed number of 400 matches.

Category	Descriptor	Recall	1-Precision	FN	F-Measure
	COSFIRE-96:	0.32	0.79	0.68	0.25
	COSFIRE-112:	0.36	0.77	0.64	0.28
	COSFIRE-142:	0.30	0.81	0.70	0.23
Blur - Structural	COSFIRE-247:	0.35	0.77	0.65	0.28
	COSFIRE-352:	0.36	0.77	0.64	0.28

	COSFIRE-336:	0.48	0.69	0.52	**0.37**
	SIFT:	0.55	0.65	0.45	**0.42**
	BRISK:	0.51	0.67	0.49	**0.39**
Blur - Textural	COSFIRE-96:	0.02	0.93	0.98	0.03
	COSFIRE-112:	0.02	0.92	0.98	0.04
	COSFIRE-142:	0.02	0.92	0.98	0.04
	COSFIRE-247:	0.04	0.85	0.96	0.07
	COSFIRE-352:	0.05	0.84	0.95	0.07
	COSFIRE-336:	0.09	0.69	0.91	**0.14**
	SIFT:	0.13	0.59	0.87	**0.20**
	BRISK:	0.08	0.73	0.92	0.12
JPEG - Structural	COSFIRE-96:	0.23	0.68	0.77	0.27
	COSFIRE-112:	0.25	0.65	0.75	0.29
	COSFIRE-142:	0.27	0.64	0.73	0.31
	COSFIRE-247:	0.32	0.56	0.68	0.37
	COSFIRE-352:	0.33	0.55	0.67	0.38
	COSFIRE-336:	0.38	0.49	0.62	**0.44**
	SIFT:	0.43	0.43	0.57	**0.48**
	BRISK:	0.47	0.37	0.53	**0.54**
Light - Structural	COSFIRE-96:	0.55	0.55	0.45	0.49
	COSFIRE-112:	0.58	0.53	0.42	0.52
	COSFIRE-142:	0.63	0.49	0.37	0.56
	COSFIRE-247:	0.68	0.45	0.32	0.61
	COSFIRE-352:	0.71	0.42	0.29	0.64
	COSFIRE-336:	0.72	0.41	0.28	**0.65**
	SIFT:	0.73	0.41	0.27	**0.65**
	BRISK:	0.69	0.43	0.31	0.62
Viewpoint - Structural	COSFIRE-96:	0.14	0.91	0.86	0.11
	COSFIRE-112:	0.15	0.90	0.85	0.12
	COSFIRE-142:	0.20	0.87	0.80	0.15

	COSFIRE-247:	0.25	0.84	0.75	0.19
	COSFIRE-352:	0.27	0.83	0.73	**0.21**
	COSFIRE-336:	0.27	0.83	0.73	**0.20**
	SIFT:	0.26	0.84	0.74	**0.20**
	BRISK:	0.50	0.68	0.50	**0.38**
Viewpoint - Textural	COSFIRE-96:	0.07	0.79	0.93	0.10
	COSFIRE-112:	0.08	0.76	0.92	0.12
	COSFIRE-142:	0.08	0.75	0.92	0.13
	COSFIRE-247:	0.12	0.64	0.88	0.18
	COSFIRE-352:	0.14	0.60	0.86	0.20
	COSFIRE-336:	0.20	0.45	0.80	**0.29**
	SIFT:	0.21	0.44	0.79	**0.30**
	BRISK:	0.20	0.46	0.80	**0.29**
Zoom and rotate - Textural	COSFIRE-96:	0.06	0.99	0.94	0.02
	COSFIRE-112:	0.07	0.99	0.93	0.02
	COSFIRE-142:	0.06	0.99	0.94	0.02
	COSFIRE-247:	0.13	0.98	0.87	0.04
	COSFIRE-352:	0.12	0.98	0.88	0.04
	COSFIRE-336:	0.16	0.97	0.84	**0.05**
	SIFT:	0.18	0.97	0.82	**0.05**
	BRISK:	0.33	0.94	0.67	**0.10**
Zoom and rotate - Structural	COSFIRE-96:	0.07	0.89	0.93	0.08
	COSFIRE-112:	0.09	0.87	0.91	0.10
	COSFIRE-142:	0.10	0.85	0.90	0.11
	COSFIRE-247:	0.15	0.79	0.85	0.16
	COSFIRE-352:	0.15	0.78	0.85	0.16
	COSFIRE-336:	0.22	0.71	0.78	**0.23**
	SIFT:	0.24	0.69	0.76	**0.25**
	BRISK:	0.24	0.66	0.76	**0.26**

From Table 10 it became clear that COSFIRE-336 outperformed the other COSFIRE descriptors. This is reasonable since for each area in the polar grid, each λ and θ values are being considered, which makes the descriptor very specific, while in the other descriptors such as COSFIRE-142, you would not know if certain λ or θ values are found in the top of the circle or at the bottom for example.

5.1 Image Classification using Bag of Words

Our next step consists of using this COSFIRE descriptor and evaluated it within the Bag of Visual Words(BoVW) image classification model. As expained in Section 3, the SIFT detector was used to acquire the keypoints of the images. For the BoVW classification model, the CALTECH-256 dataset was used to extract images from a maximum of 15 categories which are: swan, butterfly, spider, toad, baseball-bat , billiards , binoculars, elephant, soccer-ball, airplanes, chess-board, llama, car-side, treadmill and bulldozer. Each of this categories were divided into a ratio of 70 images for the training dataset and 38 images for the testing dataset. This ratio was decided so that the codewords of the vocabulary will be more well-formed due to having more training data, and therefore result in a more stable solution.

There are different types of evaluation methods that can give a clear indication of a descriptors performance. These are called cross validation methods and basically measures the accuracy your visual descriptor (for example) will achieve on unseen dataset. For this project's evaluation, the K-Fold Cross Validation was used due to its algorithm being the most related to what we require. Basically the K-Fold Cross Validation splits the training dataset into K Folds (divides the training dataset into smaller equal subsets) and run the classification for K number of times, each time picking a different subset. The cross validation accuracy is then calculated by taking the average of all accuracies.

As we explained in Section 4, for this implementation we decided to choose a codeword vocabulary of size 1000. This means that after all the keypoints are gathered from the training dataset, they will be clustered into 1000 codewords. The training and testing images will then be assigned these codewords so they can be represented

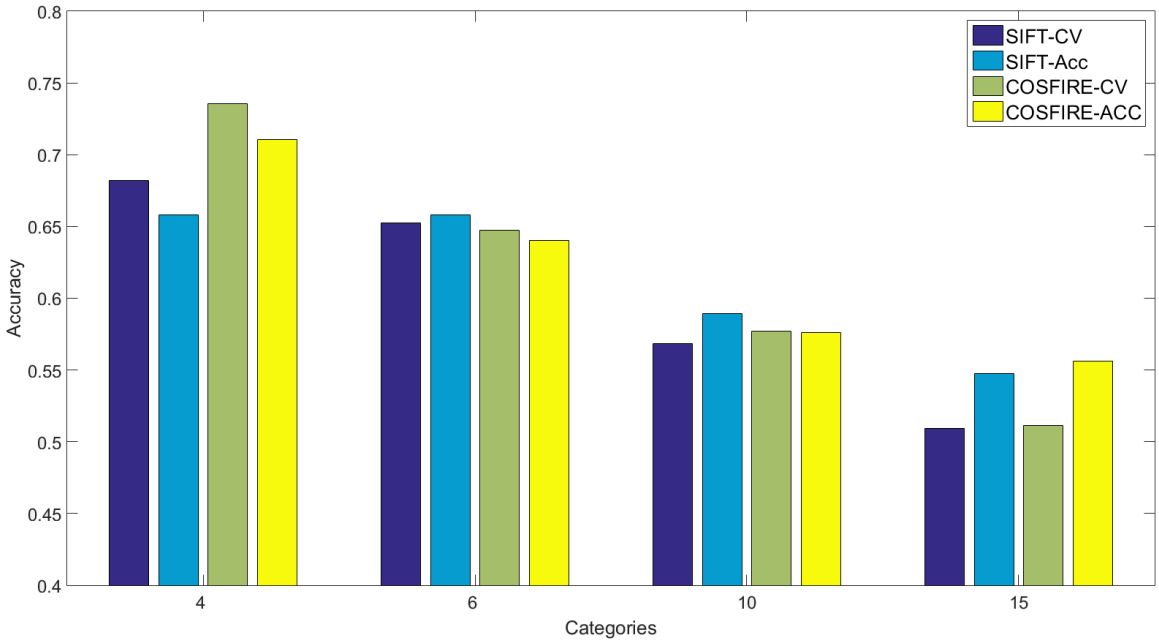


Figure 21: Bag of Visual Words Evaluation: Cross Validation accuracy and accuracy on the test dataset, using categories of the CALTECH-256 dataset, and performing different evaluation benchmarks of 4, 6, 10 and 15 categories.

with the vocabulary we just created. Due to images containing different amount of keypoints, the histogram generated from the assignment of codewords must be normalized to unit length, since if not, will cause problems in the classification (distances difference between histograms will be governed by who has the most features).

The COSFIRE-336 descriptor achieved 73.5714% 10-Fold cross validation accuracy on the training dataset and 71.0526% accuracy on the testing dataset when 4 categories were used, surpassing the accuracies of the SIFT descriptor. As the categories incremented accordingly the COSFIRE-336 descriptor still managed to keep it's high accuracy rate even at when 15 categories was used.

Our next step was to check if the accuracy increases if we apply spatial tiling to the images. We evaluated 2x2 spatial tiling, which results into 4x1000=4000 histogram size for each image. The accuracy rates dropped slightly for both descriptors as can be seen in Fig. 22, however SIFT descriptor maintained his accuracy rate better than COSFIRE-336. Due to higher spatial tiling resulted in decrease in performance for both descriptors, we base our descriptors' performance when no spatial tiling was

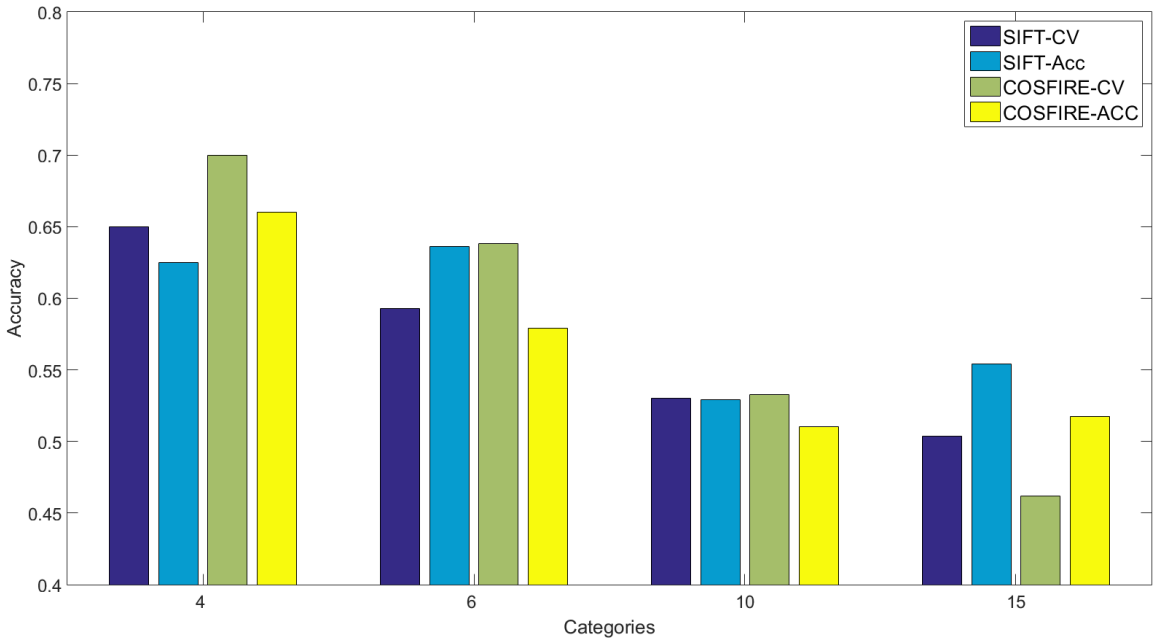


Figure 22: Bag of Visual Words Evaluation: Cross Validation accuracy and accuracy on the test dataset, using categories of the CALTECH-256 dataset, and performing different evaluation benchmarks of 4, 6, 10 and 15 categories using 2x2 spatial tiling.

computed.

6 Conclusions and future work

In this project we presented different COSFIRE descriptors configured solely from the output parameters of the COSFIRE filters. In order to compare their performances, we extracted keypoint regions from images under different conditions and analysed their performance behaviour. In most cases COSFIRE-336 proved to be the most stable, reaching SIFT performance in some cases. There were also unstable descriptors such as COSFIRE-96 and COSFIRE-112 who lacked performance consistency throughout all the categories. However the COSFIRE-96 and COSFIRE-112 reached an important implication for the COSFIRE descriptor, which is that the λ histogram is an obstacle in certain image conditions such as blur, where as can be seen in Fig. 16, their performance outperform the COSFIRE-142 descriptor. The COSFIRE-336 was then chosen for the BOVW classification model and achieved a satisfying accuracy rate compared to that of SIFT, by scoring higher cross-validation accuracy and accuracy

on the test dataset than the state-of-the-art SIFT descriptor as illustrated in Fig. 21.

It is important to note that the comparison between the configured descriptors is not heavily intensive due to lack of datasets with different image conditions. However it was still enough to distinguish if a certain descriptor is weak in textural or structural images/regions in a specific image condition such as blur. It is also worth noting that due to time constraints, further analyses and optimizations for the chosen COSFIRE descriptor were not possible and remain for future work. These include running the experiments on the whole 256 categories of the CALTECH-256 categories and analyse the descriptors' behaviour as the categories are increased drastically. Another important future work would be analysing if certain areas in a keypoint region are more important than the others, for example focusing on some specific areas in the polar grid we designed.

References

[1] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 10, pp. 1615–1630, 2005.

[2] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[3] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 2548–2555, IEEE, 2011.

[4] M. M. Amin, S. Kermani, A. Talebi, and M. G. Oghli, "Recognition of acute lymphoblastic leukemia cells in microscopic images using k-means clustering and support vector machine classifier," *Journal of medical signals and sensors*, vol. 5, no. 1, p. 49, 2015.

[5] T. Lindeberg, "Scale-space theory: A basic tool for analyzing structures at different scales," *Journal of applied statistics*, vol. 21, no. 1-2, pp. 225–270, 1994.

[6] K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 1, pp. 525–531, IEEE, 2001.

[7] P. P. Griffin G., Holub AD., "The caltech 256," 2007.

[8] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer vision–ECCV 2006*, pp. 404–417, Springer, 2006.

[9] S.-W. Ha and Y.-H. Moon, "Multiple object tracking using sift features and location matching," *International Journal of Smart Home*, vol. 5, no. 4, pp. 17–26, 2011.

[10] L. Juan and O. Gwun, "Surf applied in panorama image stitching," in *Image Processing Theory Tools and Applications (IPTA), 2010 2nd International Conference on*, pp. 495–499, IEEE, 2010.

[11] G. T. Flitton, T. P. Breckon, and N. M. Bouallagu, "Object recognition using 3d sift in complex ct volumes.," in *BMVC*, pp. 1–12, 2010.

[12] G. Azzopardi and N. Petkov, "Trainable cosfire filters for keypoint detection and pattern recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 2, pp. 490–503, 2013.

[13] G. Azzopardi and N. Petkov, "Cosfire: A trainable features approach to pattern recognition,"

[14] G. Azzopardi and N. Petkov, "Automatic detection of vascular bifurcations in segmented retinal images using trainable cosfire filters," *Pattern Recognition Letters*, vol. 34, no. 8, pp. 922–933, 2013.

[15] G. Azzopardi, N. Strisciuglio, M. Vento, and N. Petkov, "Trainable cosfire filters for vessel delineation with application to retinal images," *Medical image analysis*, vol. 19, no. 1, pp. 46–57, 2015.

[16] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and vision computing*, vol. 22, no. 10, pp. 761–767, 2004.

[17] "Region of interest(roi)." `http://www.robots.ox.ac.uk/~az/icvss08_az_spatial.pdf`. Accessed: 2015-10-05.

[18] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Applied statistics*, pp. 100–108, 1979.

[19] 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*, vol. 2, pp. 524–531, IEEE, 2005.

[20] P. Kamavisdar, S. Saluja, and S. Agrawal, "A survey on image classification approaches and techniques," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, no. 1, pp. 1005–1009, 2013.

[21] D. Lu and Q. Weng, "A survey of image classification methods and techniques for improving classification performance," *International journal of Remote sensing*, vol. 28, no. 5, pp. 823–870, 2007.

[22] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *International journal of computer vision*, vol. 60, no. 1, pp. 63–86, 2004.

[23] P. Martins, C. Gatta, and P. Carvalho, "Feature-driven maximally stable extremal regions.," in *VISAPP (1)*, pp. 490–497, 2012.

[24] T. Deselaers, L. Pimenidis, and H. Ney, "Bag-of-visual-words models for adult image classification and filtering," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pp. 1–4, IEEE, 2008.

[25] R. Shekhar and C. Jawahar, "Word image retrieval using bag of visual words," in *Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on*, pp. 297–301, IEEE, 2012.

[26] J. Yang, Y.-G. Jiang, A. G. Hauptmann, and C.-W. Ngo, "Evaluating bag-of-visual-words representations in scene classification," in *Proceedings of the international workshop on Workshop on multimedia information retrieval*, pp. 197–206, ACM, 2007.

[27] S. Banerji, A. Sinha, and C. Liu, "A new bag of words lbp (bowl) descriptor for scene image classification," in *Computer Analysis of Images and Patterns*, pp. 490–497, Springer, 2013.

[28] M. Heikkilä, M. Pietikäinen, and C. Schmid, "Description of interest regions with local binary patterns," *Pattern recognition*, vol. 42, no. 3, pp. 425–436, 2009.

[29] T. Lindeberg, "Feature detection with automatic scale selection," *International journal of computer vision*, vol. 30, no. 2, pp. 79–116, 1998.

[30] K. Sujatha, G. Karthiga, and B. Vinod, "Evaluation of bag of visual words for category level object recognition,"

[31] 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*, vol. 2, pp. 2169–2178, IEEE, 2006.

[32] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144–152, ACM, 1992.

[33] M.-L. Zhang and Z.-H. Zhou, "A k-nearest neighbor based algorithm for multi-label classification," in *Granular Computing, 2005 IEEE International Conference on*, vol. 2, pp. 718–721, IEEE, 2005.

[34] S.-C. Wang, "Artificial neural network," in *Interdisciplinary Computing in Java Programming*, pp. 81–100, Springer, 2003.

[35] Y. Bai, W. Yu, T. Xiao, C. Xu, K. Yang, W.-Y. Ma, and T. Zhao, "Bag-of-words based deep neural network for image retrieval," in *Proceedings of the ACM International Conference on Multimedia*, pp. 229–232, ACM, 2014.

[36] T. Aach, A. Kaup, and R. Mester, "On texture analysis: Local energy transforms versus quadrature filters," *Signal Processing*, vol. 45, no. 2, pp. 173–181, 1995.

[37] "Sift detector." `http://www.vlfeat.org/api/sift.html`, 2007. Accessed: 2016-03-30.

[38] C. Zhang and S. Xia, "K-means clustering algorithm with improved initial center," in *Knowledge Discovery and Data Mining, 2009. WKDD 2009. Second International Workshop on*, pp. 790–792, IEEE, 2009.

[39] S. P. Lloyd, "Least squares quantization in pcm," *Information Theory, IEEE Transactions on*, vol. 28, no. 2, pp. 129–137, 1982.

[40] "Imagenet." http://image-net.org/download-features, 2014. Accessed: 2016-03-30.

[41] "Sift detector and descriptor." http://www.vlfeat.org/overview/sift.html, 2007. Accessed: 2016-03-30.

[42] "Libsvm library." https://www.csie.ntu.edu.tw/~cjlin/libsvm/, 2005. Accessed: 2015-10-04.