Hacettepe University Department of Computer Engineering

Name & Surname	: Baturalp Furkan KARAMUS
Identity Number	: 21527137
Course	: BBM418 - Computer Vision Laboratory
Assignment	: Assignment1
Advisors	: Assoc. Prof. Dr. Nazlı İKİZLER CİNBİŞ, TA. Özge YALÇINKAYA
Due	: 13.04.19
E-mail	: b21527137@cs.hacettepe.edu.tr

1 Introduction

1.1 Problem

In this assignment, we will analyze 4 different feature detection based algorithm depends their accuracy on detect features.These algorithms are Gabor Filter Bank,Average SIFT,Bag Of Visual Words and Bag Of Visual Words with spatial tiling. The problem is analysing this algorithm and compare which one is the best solution for us.

This report contains all the steps involved in the process, as well as how the problems encountered in these stages are optimized and relevant.

2 First Part

In this section we use two different algorithm for extract features from image. One of them is Gabor Filter Bank and the other one is SIFT descriptor. Both of them have different attiributes pros and cons. Now we will discuss this algorithms and analyse test results.

2.1 K-NN Classification

In this assignment we use K-NN classification for gettin class based accuracy after every algorithm.K-NN classification based on nearest neighbours. According to nearest neighbours it counts as a prediction about this image. If prediction match actual class of image it is a succesful match.

While finding nearest neighbour this algorithm use euclidean distance but at the end of this report we will discuss about drawbacks on euclidean distance is the best or any other algorithm exist better than include euclidean distance.

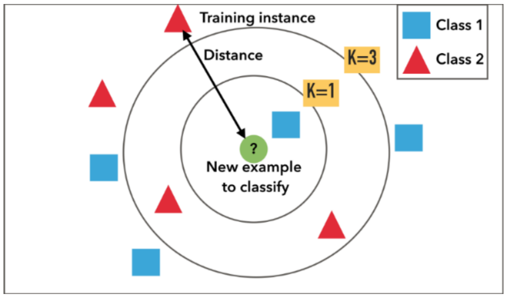

Figure 1: Basically k-nn algorithm.

2.2 Gabor Filter Bank

Gabor filter-based texture feature extraction derives a Gabor filter bank consisting of filters with different frequencies and orientations to perform a multi-channel feature representation of a texture pattern, which is in line with the multi-channel filtering mechanism of the human visual system in perceiving texture information.

Gabor filters are orientation-sensitive filters, used for texture analysis.The typically travel in packs, one for each direction. A gabor filter set with a given direction gives a strong response for locations of the target images that have structures in this given direction. For instance, if your target image is made of a periodic grating in a diagonal direction, a gabor filter set will give you a strong response only if its direction matches the one of the grating.

A Gabor filter is a linear filter formed by multiplying a Gaussian function with a harmonic function.

$$g(x, y; \lambda, \theta, \phi, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right)\cos\left(2\pi\frac{x'}{\lambda} + \phi\right) \tag{1}$$

As we see there are many different parameter takes this equation. In this parameter theta and sigma parameter are important for creating filter bank.

X' and Y' values

$$x' = x\cos\theta + y\sin\theta, y' = -x\sin\theta + y\cos\theta. \tag{2}$$

As a result of the product, x and y coordinates are rotated clockwise from the x and y coordinates x 'and y' coordinates are obtained.

Theta(θ)

This angle determines the direction of the edges to be detected. When the angle is 0, the edges that are perpendicular to the x axis are detected. When the angle is 90, the edges that are perpendicular to the y axis are detected. As the angle is increased, the edges with increasing angles are detected according to the rotation direction of the rotation matrix.

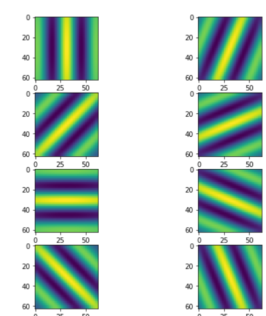

Figure 2: Some of gabor filters that i used.(Real images include just black white.)

Software Usage

I used 40 different gabor filters for this part of experiment.And then take average value for every different filter response. Add this values to an array. And that way i can extract features from both training and query images. After that i append image code to this feature arrays last element. Then put them to k-NN classification for getting class based accuracy.

Experiment Results

In this part firstly i choose 3 images from 3 different class in query images. And 5 images with the most similarity to these images were analyzed.


Query Image

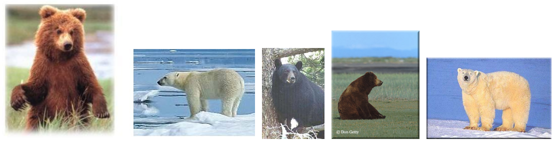

Figure 3: Most Similiar Images(From left to right)


Query Image

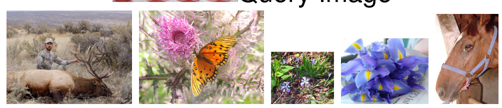

Figure 4: Most Similiar Images(From left to right)


Query Image

Figure 5: Most Similiar Images(From left to right)

The top 5 test results for the query picture at the bottom of the line above are listed at the top. This result taken by gabor filter. By used with 40 different gabor filter. While doing this K-NN classification is used. This test's result nearly %33 accuracy. The reason for the test result to be more accurate when the picture of the selected picture is bear indicates that the responses of the filters are more characteristic in these pictures. Also fire-truck image include that.

Table 1: Gabor Filter Bank Accuracy

Part		
Total	Correct	Accuracy
50	11	%22.0
3	1	%33.33333

For 50 query image feature 1-nn classification gives 11 correct answer.For 3 query image feature 1-nn classification gives 1 correct answer.

Implementation Details

In this part there is some explanation about test steps:

1-Create Gabor Filter Bank:
In this step we create different orientated kernels for each has different responses.
2-Get Mean Value Filtered Images:
In this step we get average value of filtered image for each orientation. After that we get 1x40 vectors. These are our features.This process applied for both query and train images.
3-K-NN classification:
In this step we apply k-nn classification to query and train features. That returns nearest neighbours to amount of k value. According to experiment this k value assigned as 1.
4-Get Class Based Accuracy:
Final step we compare query image and nearest neighbour. If query image overlap to nearest neighbour that is correct result.After the all comparison this steps returns accuracy value.

2.3 SIFT DESCRIPTOR

A SIFT feature is a selected image region (also called keypoint) with an associated descriptor. **Keypoints** are extracted by the SIFT detector and their descriptors are computed by the SIFT descriptor.

A SIFT keypoint is a circular image region with an orientation. It is described by a geometric frame of four parameters: the keypoint center coordinates x and y, its scale (the radius of the region), and its orientation (an angle expressed in radians). The SIFT detector uses as keypoints image structures which resemble “blobs”. By searching for blobs at multiple scales and positions, the SIFT detector is invariant (or, more accurately, covariant) to translation, rotations, and re scaling of the image.

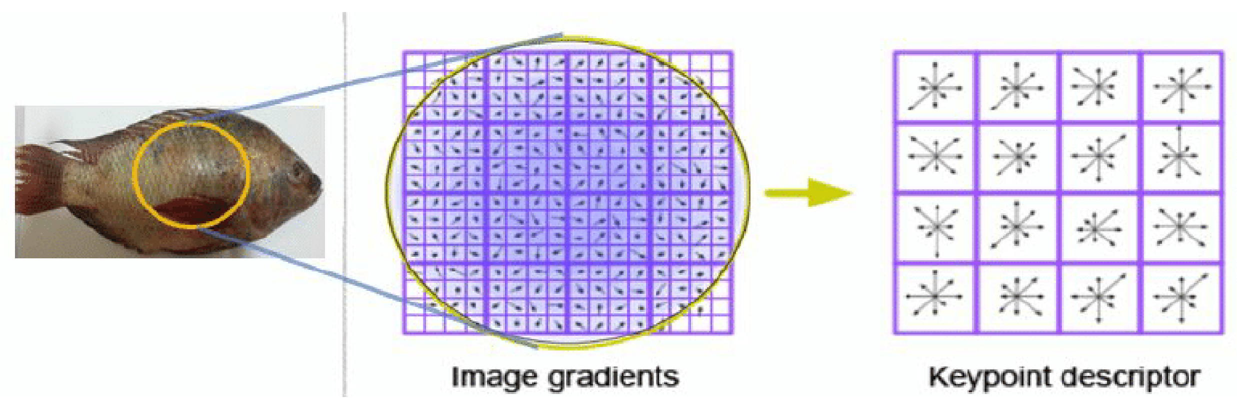

Figure 6: SIFT interest points examples)

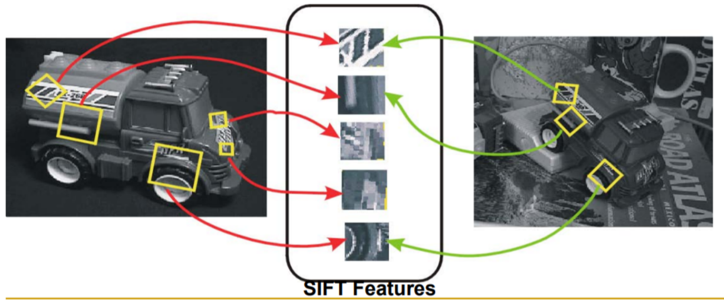

Figure 7: SIFT Key Points

SIFT isn't just scale invariant. You can change the following, and still get good results:

- Scale
- Rotation
- Illumination
- Viewpoint

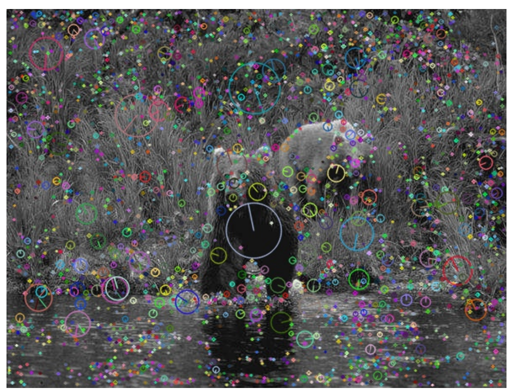

Figure 8: SIFT Key Points

Average SIFT:

In this experiment we use average SIFT algorithm. Average SIFT created by taking mean values of SIFT descriptors and put them all together for applying k-NN classification. In this way we could detect which query image has nearest distance to appropriate train image.

SOFTWARE USAGE:

For doing comparison between images firstly i split class based names from images. Then put them to the last element of each descriptor. Then both train and query image i put all descriptors to an array. Firstly 3 image taken for classify with k-NN. After that all query images are classified with this algorithm.

EXPERIMENT RESULTS:

In this part 3 image chosen and these are chosen images and their 5 best responses.


Query Image

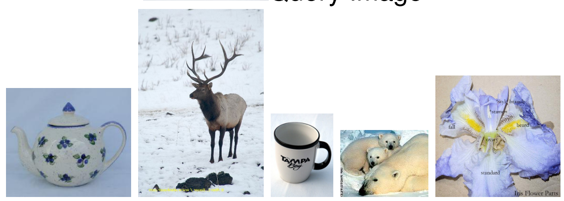

Figure 9: Most Similiar Images(From left to right)


Query Image

Figure 10: Most Similiar Images(From left to right)


Query Image

Figure 11: Most Similiar Images(From left to right)

The top 5 test results for the query picture at the bottom of the line above are listed at the top. This result taken by Average SIFT features. While doing this K-NN classification is used. This test's result nearly %33 accuracy.

The reason for the test result to be more accurate when the picture of the selected image is fire truck.These image indicates much more current colour than others. Because of that image gradients are different from other images. So these are matches easily.

Table 2: AVERAGE SIFT Accuracy

Part		
Total	Correct	Accuracy
50	11	%22.0
3	1	%33.33333

The results of this test were found to be the same as the results obtained in the gabor filter test. In this case, both algorithms were found to be successful in different images. It was observed that the Gabor filter gave better response for bear images and the SIFT algorithm have better response for fire-truck images were better.

Implementation Details:

In this part there is some explanation about test steps:

- Collect all train images.
- Apply all to SIFT and store descriptors in list.
- Take the mean value of each descriptor and store them.
- Apply this steps also for query images.
- Apply k-nn classification for class based accuracy
- Take nearest neighbours and do predictions.
- Return results.

In this part we take average values of descriptors because we basically think that similar image gives closest response for average returns.

3 Second Part

Bag of visual words (BoW) is a strategy that draws inspiration from the text retrieval community and enables ecient indexing for local image features. In document classi

cation, a bag of words is a sparse vector of occurrence counts of words; that is, a sparse histogram over the vocabulary. In computer vision, a bag of visual words is a sparse vector of occurrence counts of a vocabulary of local image features. BoW descriptor extraction consists of 3 main parts. As the illustration at Figure 1 shows, you have to i) extract SIFT features, ii) create a code book, iii) quantize each feature and calculate a histogram from it, with or without a spatial tiling.

3.1 K-Means Clustering

K-Means is one of the most popular "clustering" algorithms. K-means stores k centroids that it uses to define clusters. A point is considered to be in a particular cluster if it is closer to that cluster's centroid than any other centroid.

K-Means finds the best centroids by alternating between (1) assigning data points to clusters based on the current centroids (2) chosing centroids (points which are the center of a cluster) based on the current assignment of data points to clusters.

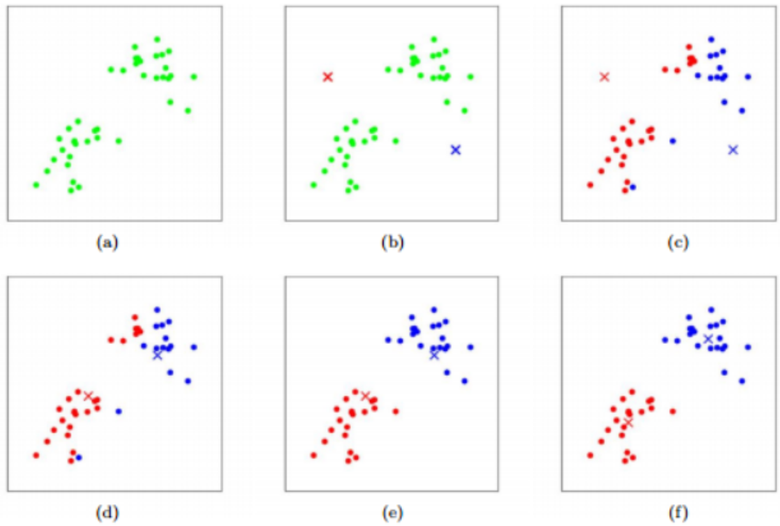

Figure 12: K-Means Clustering

3.2 Bag Of Visual Words

Bag of Visual Words is an extention to the NLP algorithm Bag of Words used for image classification. Other than CNN, it is quite widely used. I sure want to tell that BOVW is one of the finest things I've encountered in my vision explorations until now.

So what's the difference between Object Detection and Object Recognition .. !! Well, recognition simply involves stating whether an image contains a specific object or no. whereas detection also demands the position of the object inside the image. So say, there is an input image containing a cup, saucer, bottle, etc. The task is to be able to recognize which of the objects are contained in the image.

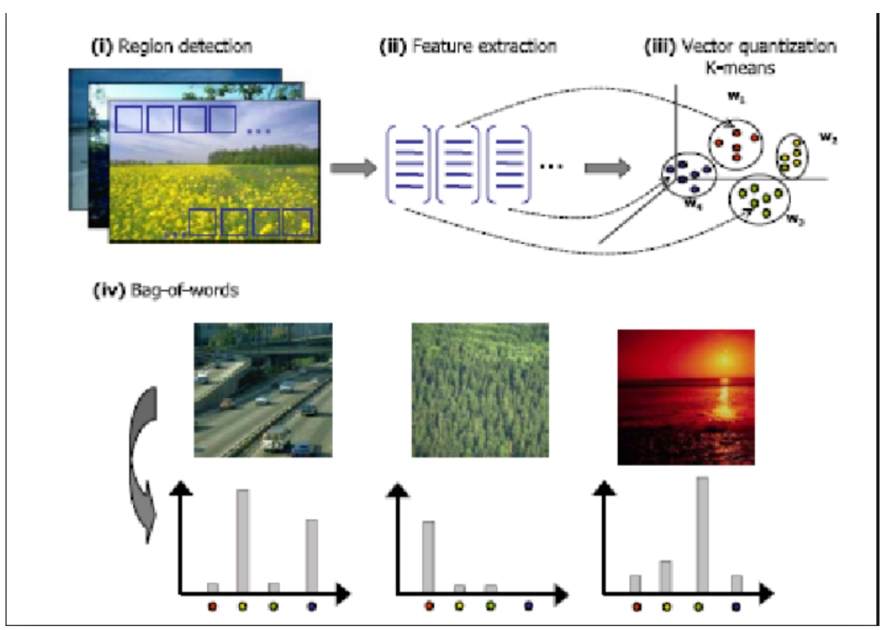

Figure 13: Creation Visual Words

SOFTWARE USAGE:

In this chapter we achieve that create a visual dictionary. To do this i used SIFT features which created before. All train images SIFT features are collected together. After that code book created.
When the code book creation k-means clustering algorithm used. K-Means algorithm applied all train images for extract cluster centroids. After that creating histogram for comparison. Then i do that for both training and query images. Finally these result are being classified with k-NN classification. Then calculate accuracy for correct results. When take the accuracy try these steps with different k value for k-means.

EXPERIMENT RESULTS:

Figure 14: Most Similiar Images(From left to right)

Figure 15: Most Similiar Images(From left to right)

Figure 16: Most Similiar Images(From left to right)

Result are close to Average SIFT results. I think that these two methods both use SIFT features because of that this result must be expected.But this test result has %66 accuracy.For 3 query image two of them correct.

Reason is when we using histogram most similar words in our dictionary attached to mostly similar image. Each word look for their own type on the images. And this is the most succesful way in my whole experiment.

K	Total	Correct	Accuracy
10	50	12	%24.00
20	50	12	%24.00
30	50	14	%28.00
200	50	12	%24.00
500	50	15	%30.00

These are the different test results for different K values.According to this result i can say that when the k value is bigger it is as better situation. But in my test this not showing clearly.

Implementation Details:

In this part there is some explanation about test steps:

- Get SIFT features for train image.
- Put descriptors into a stack for k-means process.
- Do k-means for different k values on trains.
- Append image code all descriptors
- Take SIFT features again and build histogram with both train and query images.
- Then with these histograms find closest neighbour with k-nn algorithm and calculate accuracy.

4 Functions:

4.1 K-NN Classification

getAccuracy(testSet, predictions): This function takes all predictions sort from process and returns accuracy for overall.

getResponse(neighbors): Takes neighbours and according to closest return result prediction.

getClassBasedAccuracy(featureTrain,featureQuery): Take train and query features and return accuracy:

euclideanDistance(instance1, instance2, length): Take two vectors and return euclidean distance.

getNeighbors(trainingSet, testInstance, k) Take training and query and return nearest neighbours is amount of k value.

4.2 Image Reading

getFolders(string): Take directory and return folders

getQuery(string) Take directory and return query images.

4.3 Gabor

normalizeVector(list): Normalize given vector for better result.

buildFilters(): Create gabor kernels for different orientation.

process(img, filters): Apply gabor filter for given image.

applyFilterAll(list,imageNames): Apply gabor filter all trains.

applyFilterQuery(list): Apply gabor filter all queries.

4.4 SIFT

applySIFTtrain(list): Apply SIFT process all trains and return features.

getAllDescriptors(list): Return SIFT descriptors for given list.

queryDescriptors(list): Return SIFT descriptors for query images.

stackDescriptors(desList): Stack the given descriptor list. This function's purpose is get ready descriptors for k-means clustering built-in function.

buildHistogram(descriptorList, clusterAlg):

For other process built-in functions used.

5 Conclusion:

5.1 Euclidian Distance:

Euclidean distance represents the distance between two points. Hence it represents physical distance between two points. Eg. While performing K-means clustering, the metric used is Euclidean distance, since points may be in opposite directions but they may fall into the same cluster, if the distance of both points from the centroid is the same.

$$\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{3}$$

Manhattan Distance: The Manhattan distance function computes the distance that would be traveled to get from one data point to the other if a grid-like path is followed. The Manhattan distance between two items is the sum of the differences of their corresponding components.

$$\left(\sum_{i=1}^{n}|x_i - y_i|^p\right)^{1/p} \tag{4}$$

Minkowski Distance: The Minkowski distance is a generalized form of the Euclidean distance (if p=2) and the Manhattan distance (if p=1).

$$\left(\sum_{i=1}^{n}|x_i - y_i|^p\right)^{1/p} \tag{5}$$

Road distance and travel time measurements are the most accurate estimates, but cannot be directly implemented in spatial analytical modeling. Euclidean distance tends to underestimate road distance and travel time; Manhattan distance tends to overestimate both. The optimized Minkowski distance partially overcomes their shortcomings; it provides a single model of travel over the network. The method is flexible, suitable for analytical modeling, and more accurate than the traditional metrics; its use ultimately increases the reliability of spatial analytical models.

5.2 General Comment:

After all these tests and expirements i can say that each one of them have pros and cons. A global Gabor texture feature is associated with a well-defined region. The local SIFT descriptors can represent more general regions. While both are applied here to classifying entire images, they could enable different techniques when applied to classifying regions within larger images. We are exploring the application of SIFT descriptors to labelling arbitrary shaped regions in remote sensed imagery. As can be seen, Gabor features have better determined the bear images, and the SIFT features have achieved this success in fire-truck images.

Bag Of Visual Words is more complex method. It include SIFT descriptors because of that result sometimes similar to SIFT results. When the k value get bigger accuracy is growing. But the runtime of program increase too much.