PERFORMANCE ANALYSIS OF IMAGE CLASSIFICATION ALGORITHMS

Surbhi Saxena[1], Dr.R.S.Ochawar[2]
[1]Department of Electronics Engineering, RCOEM, Nagpur, Maharashtra, India
[2]Associate Professor, Department of Electronics Engineering, RCOEM, Nagpur, Maharashtra, India

ABSTRACT

Image classification plays an integral role in computer vision. Given an image a computer must be able to classify what the image represents. While this is a fairly simple task for humans, it is not an easy task for computers. Computers must go through a series of steps in order to classify a single image. In this paper, we used a saliency based segmentation process to compare two different classification methods. Both K-Nearest-Neighbor (KNN) and Support-Vector-Machine (SVM) classification are designed using VHDL code. We were able to observe that the SVM classifier outdoes the KNN classifier. For future work, we hope to use more categories for the objects and to use more sophisticated classifiers.

Keywords - *KNN (k nearest neighbor), SVM (support vector machine), Classification, Saliency, HMMD.*

I. INTRODUCTION

The classifier plays an important role in image processing. The human ability to analyze and classify objects and scenes rapidly and accurately is something that everybody finds highly useful in everyday life [1] [2]. Classification techniques are used in a variety of fields including marketing and business intelligence, biotechnology, multimedia and security [3]. Classification process has an input dataset called the training set which consists of example records with a number of attributes. The objective of a classification process is to use this training dataset to build a model which can then be used to assign unclassified records into one of the defined classes. A typical classification method consists of four steps as shown in fig.1. First step is segmentation process for segmentation we use saliency map based segmentation; second step is feature extraction, third is histogram generation and fourth is classification.

In this paper we are comparing two different classification methods using saliency map segmentation process and architecture for classifiers on modelsim. In section 2 and 3 we will discuss the segmentation process and feature extraction. In section 4 we will explain the two different classification methods using modelsim. We have used KNN and SVM.

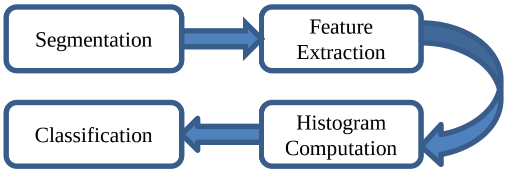

Figure 1 A conceptual illustration of the process of image classification.

II. IMAGE SEGMENTATION (SALIENCY MAP BASED SEGMENTATION)

Segmentation is the process that splits an image into several meaningful regions. Salient region detection in image is very useful for image processing [4, 5]. In this paper an improved approach to detect salient region is presented. Here we present details of our approach for saliency determination and its use in segmenting whole objects. An overview of process is presented in fig.2.

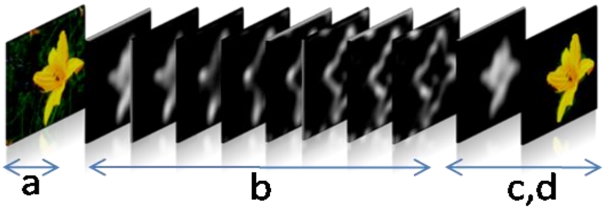

Figure 2 Overview of the process finding salient regions. (a)Input image. (b)Filter input image to get Saliency maps. (c)Final saliency map and the segmented image. (d)The output image containing the salient object that is made of only those segments that have an average saliency value greater than the mean.

Saliency map based image is obtained using following steps.

1. Initially, images in the database are taken as input image.
2. Calculate entropy of the input image.
3. Apply Gaussian filter on the images 8 times and get saliency maps.

4. Calculate mean of the input image.
5. Compare saliency maps with mean. If saliency map is larger than mean, then put foreground pixel as 1 otherwise it becomes 0.
6. Final saliency mapped image we get.
7. Overlap this saliency mapped image with input image.
8. The output image is nothing but a segmented image which containing the salient object.

III. IMAGE FEATURE EXTRACTION

This phase of work aims to extract features from the images. Here energy and visual features of colour and edges are extracted and stored in the feature database. For extracting the above features, HMMD colour space and edge histogram descriptor are used respectively.

The energy feature vectors include information about the high frequency bands. The high frequencies largely occur for a shorter period than that of low frequencies [6].

HMMD is a colour space descriptor and it specifies the colour points. It also allows colour space preference. The description utilizes the selected colour space and it gives scalar quantization for image processing. The colour quantization descriptor divides the colour space into bins [7] [8]. Four components of HMMD colour space are hue, min, max, and diff. Other than these, fifth component of HMMD is sum.

The relationship between RGB and HMMD is given by,

a) Max = max(R, G, B)
b) Min = min(R, G, B)
c) Diff = Max-Min
d) Sum = (Max + Min) /2

A descriptor for image matching used is edge histogram descriptor. It represents the local edge distribution in an image. There exist four types of edges described by edge histogram descriptor. They are

a) Vertical edge
b) Horizontal edge
c) Diagonal edge
d) Anti-diagonal edge

Algorithm:

The following steps provide the detailed working of the proposed system.

ENERGY:

1. Initially, segmented image is taken as input image for feature extraction and rescaled to 256 x 256 pixels.
2. Convert RGB colour space of image into grey and calculate energy of that image.
3. The image is subjected to first level lifting wavelet decomposition resulting in difference and average sub bands.
4. Calculate energy of difference sub band.
5. Apply second level lifting wavelet decomposition on difference image of first level.
6. Calculate energy of second level decomposed image energy of second level decomposed difference image.

COLOUR:

1. First take segmented image as input image.
2. Calculate HMMD of input image.
3. Hue is calculated by below formula

First check the maximum and minimum value, if maximum and minimum having the same value then hue is given by

Hue=0

Again check the value of Max if it is equal to the value of red, and also green is greater and equal to the blue then hue is given by

Hue=60*(g-b)/max(R,G,B)-min(R,G,B)

Check the value of Max if it is equal to the value of red, and also green is lesser to the blue then hue is given by

Hue=360+60*(g-b)/max(R,G,B)-min(R,G,B)

Check if the Max value is equal to the green so hue is calculated by

Hue=60*(2+(b-r))/max(R,G,B)-min(R,G,B)

For the other cases hue is calculated by

Hue=60*(4+(r-g))/max(R,G,B)-min(R,G,B)

4. Generate indexed image.
5. Histogram are calculated.

EDGE HISTOGRAM:

1. For an image, image intensity is determined and edge detection mask is initialized.
2. By performing filtering using masks, edge images are created.
3. Histogram are computed.

IV. CLASSIFICATION METHODS

A. K-Nearest-Neighbor Classification:

K-nearest neighbor algorithm [9] [10] is a method for classifying objects based on closest training examples in the feature space. K-nearest neighbor algorithm is among the uncomplicated of all machine learning algorithms. Preparation process for this algorithm consists of only collecting feature vectors and class of the training images. In the classification process, the untagged query image feature is simply appointed to the label of its k nearest neighbors. Typically, the object is classified based on the labels of its k nearest neighbors by majority. If k=1, the object is simply classified as the class of the object nearest to it. When there are only two classes, k must be an odd integer. However, there can still be tether when k is an odd integer when performing multiclass classification. After we decimalize each image to a vector of fixed-length with real numbers, we used the most common distance function for KNN which is city block distance. Fig.3 shows the process diagram for KNN classifier.

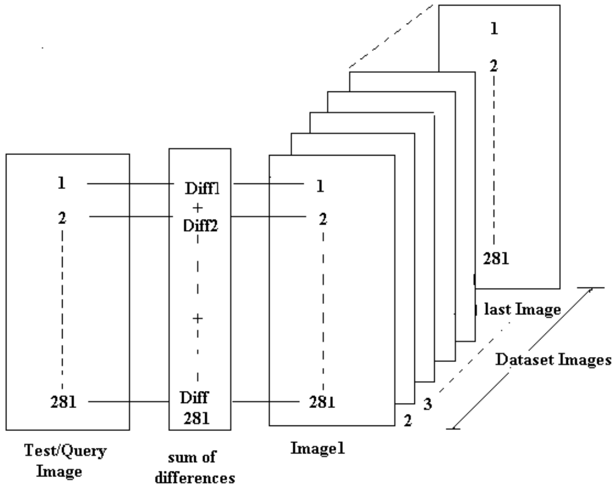

Figure 3 Process diagram for KNN classifier.

The following steps provide the detail of the proposed VHDL code for KNN classifier.

1. Initially store features of image database and query image into memory.
2. Calculate difference between features of first database image and query image.
3. Add all the differences of features calculated in step 2.
4. Repeat step 2 and step 3 for all the images in dataset.
5. Compare sum of differences and find the minimum value of sum.
6. Define class of the minimum sum of differences as the query image class.

B. *Support Vector Machine Classification*

SVM classification [11] [12] uses different planes in space to divide data points using planes. An SVM model is an allocation of the features as points in space, mapped so that the features of the separate categories or classes are divided by a dividing plane that maximizes the margin between different classes. This is due to the fact if the separating plane has the largest distance to the nearest training data points of any class, it lowers the colligation error of the overall classifier. The test features or query features are then mapped into that same space and ventured belong to a category based on which side of the gap they fall on. Since MATLAB's SVM classifier does not support multiclass classification and only supports binary classification, we design multiclass SVM classifier by VHDL code. Fig.4 shows the process diagram for SVM classifier.

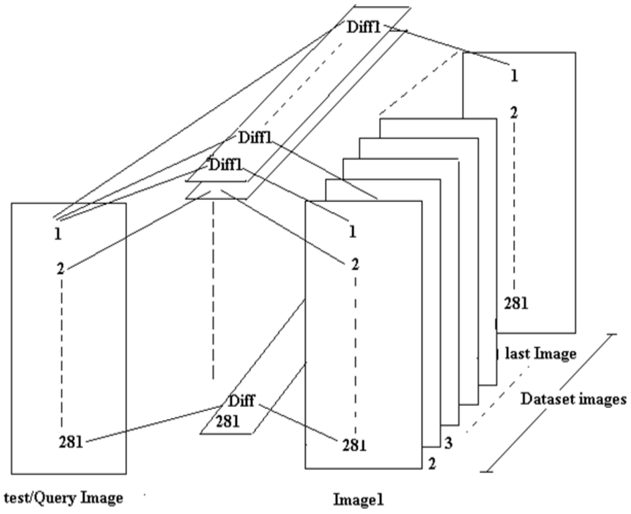

Figure 4 Process diagram for SVM classifier

The following steps provide the detail of the proposed VHDL code for SVM classifier.

1. Initially store features of image database and query image into memory.
2. Calculate difference between first feature of all database images and first feature of query image.
3. Find the minimum difference of features calculated in step 2.
4. Increase support value for the database image which is having minimum difference value.
5. Repeat step 2, 3 and 4 for all the features.
6. Compare the value of support for all database images and find the maximum value.
7. Define class of the image which is having maximum support value as the query image class.

V. RESELT

We performed classification with a caltech101 and sun dataset for 4 classes, with approximately 400 training images and approximately 100 testing images. Each investigation had different images in training and testing compared to the other trail so as to prevent the overlapping of testing and training images in each demonstration. In order to compare the performance levels of KNN classification and SVM classification, we have implemented the framework for the segmentation process as discussed in section 2 and perform classification process using MODELSIM.

Figure 5 Sample test images of the 4 classes. Starting from upper right and in clockwise order: bike, buildings, cares, trees

Table 1. The average performance of KNN classifier

	Car	Bike	Tree	Building
Car	74%	11%	7%	8%
Bike	14%	68%	8%	10%
Tree	9%	12%	79%	0
Building	7%	0	14%	79%

Table 2. The average performance of SVM classifier

	Car	Bike	Tree	Building
Car	84%	10%	0	6%
Bike	10%	77%	6%	7%
Tree	0	12%	86%	2%
Building	0	0	8%	92%

VI. CONCLUSION

In this work, we implemented two different structures for image classification. With an appropriate learning process, we could perceive that SVM classification outdoes KNN classification. Although the performance of KNN classification was very low compared to SVM. This was due to the fact that the KNN classifier could not discriminate buildings whereas SVM was very operative in classifying buildings. If we take the class of building away from the overall experiment, we can see that SVM outdoes KNN by 8-9%. If we add buildings back into the overall experiment however, we can see that SVM outdoes KNN by 13%.

REFERENCES

[1]. S. Thorpe, D. Fize, and C. Marlot. Speed of processing in the human visual system. Nature, 381, 1996, pp. 520–522.

[2]. A. Treisman and G. Gelade. A feature integration theory of attention. Cognitive Psychology, 12, 1980, pp. 97–136.

[3]. R. Szeliski, Computer Vision: Algorithms and Applications, Springer, 2011.

[4]. Radhakrishna Achanta, Francisco Estrada, Patricia Wils and Sabine Susstrunk, Salient Region Detection and Segmentation, ICVS, 2008.

[5]. Xuelong HU, Huining WU, Yuhui ZHANG, Lei SUN, Flower Image Retrieval Based on Saliency Map, International Symposium on Computer, Consumer and Control, 2014.

[6]. N S T Sai, R C Patil, Image Retrieval using 2D Dual-Tree Discrete Wavelet Transform, International Journal of Computer Applications, 14, February 2011.

[7]. NST Sai, Ravindra patil, New Feature Vector for Image Retrieval: Sum of the Value of Histogram Bins IEEE Conference on Advance in Computing, Control and Telecommunication Technologies, 2009.

[8]. P. A. Hemalatha, Image Retrieval by Content Using Descriptors, International Journal Of Engineering Sciences & Research Technology, October, 2013.

[9]. Bremner D, Demaine E, Erickson J, Iacono J, Langerman S, Morin P, Toussaint G, Output sensitive algorithms for computing nearest neighbor decision boundaries, Discrete and Computational Geometry, 2005, pp. 593–604.

[10]. Cover, T., Hart, Nearest-neighbor pattern classification, Information Theory, IEEE Transactions on, Jan. 1967, pp. 21-27.

[11]. Christoper J.C. Burgers, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998, pp.121–167.

[12]. C. Cortes and V. Vapnik. Support-vector network. Machine Learning, 20, 1995, pp. 273-297.