The Pennsylvania State University

The Graduate School

Department of Electrical Engineering

CLUSTERING AND CLASSIFICATION OF IMAGES BASED ON SALIENT DATA POINTS

A Thesis in

Electrical Engineering

by

Nachiket Kare

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 2016

The thesis of Nachiket Kare was reviewed and approved* by the following:

David J Miller
Professor of Electrical Engineering
Thesis Advisor

Vishal Monga
Associate Professor of Electrical Engineering

Kultegin Aydin
Professor of Electrical Engineering
Hear of Electrical Engineering Department

*Signatures are on file in the Graduate School

ABSTRACT

In this thesis work, we consider the problem of object clustering and classification from a set of images. The problem being considered here is supervised since we know the class labels of the images. Saliency detection is used as a tool to narrow down the object location in the image. In this thesis work we study a variant of the popular K-means clustering algorithm. Clustering is applied separately for each class, with the bag of salient point feature vectors representing an image assigned to one of a set of cluster "families" where each class is represented by multiple cluster families. A cluster family can be thought of as a sub-class or a class within a class. A supervised variant of this scheme is also proposed whose parameters can be initialized by the (unsupervised) clustering approach. In this case, the "clustering" solution is learned based on a gradient descent procedure, with respect to a "soft" training set measure. We aim to experiment with the number of clusters families used to represent each class and the number of clusters present in each cluster family.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Firstly, I would like to thank my thesis adviser, Dr. David Miller. Dr. Miller has been very supportive throughout this work, he has guided me throughout my thesis work and has answered all my questions without any hesitation. His expertise in Machine Learning and Clustering algorithms is amazing and I’m thankful to him that he gave me a chance to work under him. I would also like to thank Dr. Vishal Monga for being on my committee. Dr. Monga has always encouraged me to try new ideas throughout the time I was enrolled on his course. I would also like to thank Harshil Shah, Chetan Chimate and Abhishek Pathak for helping me with my research. Finally I would like to thank my parents for supporting me throughout my Master’s degree. This would not have been possible without their support and motivation throughout my studies.

Chapter 1

Introduction

We as human beings are quick to identify a “visually noticeable” object from its surroundings. All such “visually noticeable” objects can be labelled as salient objects. Identification of such salient objects is often attributed to changes in color or gradient or both at the object boundary. Identifying salient objects in an image using computer vision and machine learning algorithm has been a profound challenge. There has been extensive research in this direction [2, 3, 4, 5, 6]. All such methods measure the visual importance of the pixel and generate a saliency map image, which can then be used to classify the salient object.

In this thesis work we consider the clustering and classification of images, with each image represented by its salient data points. The salient data points are basically feature vectors extracted at the most salient points from the saliency map image. Saliency detection is used as an aid to help locate the position of the object of interest in the image. The objects of interest considered here are very distinct compared to each other and are noticeably different from their surroundings. We propose a variant of the K-means algorithm for clustering bags of salient points (each representing an image). We also propose a probabilistic classification model that is essentially a supervised classifier variant of our clustering algorithm. We use an iterative approach for minimizing a “soft” estimate of the training set classification rate based on the gradient descent approach.

Our Contribution

In this thesis work, we present an algorithm for clustering bags of salient data points extracted from images. This approach is supervised since we know the object labels for the images in the training set. Although saliency detection has been used for object detection and classification in the past [2, 3, 4, 5], our method differs from other approaches based on the following two points: 1) a clustering algorithm which is a variant of the popular k-means algorithm and 2) a probabilistic classification model proposed to represent the bags of salient points.

In the clustering algorithm proposed in this thesis work, we assume that each class is represented by a group of cluster families. Each cluster family consists of several clusters. Each image is assigned to a distinct cluster family. All the salient points belonging to that image are constrained to be assigned to the clusters within the same cluster family. Assigning images to cluster families helps us group similar images. Assigning salient points in that image to clusters within the cluster family enables us to collect similar features. This results in a two level clustering that is not only are images clustered and assigned to cluster families, but points within the images are assigned to clusters within the cluster families. This is similar to the bag of features approach [16] but with clustering as the prime objective. We propose a unique cost/evaluation function for this clustering method to work and converge to a correct output.

Our second contribution is to represent the points belonging to an image by a probabilistic variant of the cluster families obtained from the above algorithm. We model each class in this way. A soft classification error function is introduced as our training objective function which is minimized via a gradient descent algorithm. The model so obtained is then used for object classification.

Overall Outline

This thesis work is arranged as follows. In this next chapter we present a summary of the literature reviewed. We explain and elaborate on the definition of saliency and the methods used to compute saliency from an image. In chapter 3, we present out algorithm for clustering and classification of images represented by salient data points. In the same chapter some feature extraction methods are discussed. Chapter 4 lists all the results and relevant inferences. We conclude in chapter 5 by stating the future scope for this research work.

Chapter 2

Literature Review and Experimental Comparisons

In this chapter we present a brief overview of all the methods that we reviewed. A brief explanation of saliency detection is also provided here. A few questions like what saliency means and how to compute saliency for a given image are answered in this section. We reviewed and implemented four different saliency score computation method [2, 3, 4, 5]. The output for these methods were compared and the best method was chosen.

Previous Work

Saliency detection is a method used to highlight/identify the most salient object from the image. A salient object can be anything that is salient (strikingly different) from its background. There has been a lot of research along these lines in the past. All the papers propose a methods to distinguish the object of interest from its surrounding. We can see an example of saliency detection in fig. 1 and fig. 2 below. As can be seen from fig. 2, we get a clear idea about the shape and location of the object of interest in fig. 1.

The saliency detection methods that have been proposed so far, can be classified into three groups: 1) which concentrate on heuristic saliency features [2, 3, 4], 2) which concentrate on discriminative saliency features [1, 12] and 3) other methods which cannot be included in these two groups [13]. Pixel/patch based contrast methods [3, 8, 9], region-based contrast methods [4] and pseudo background method [10, 11] can be included in the group of methods which rely on heuristic saliency features. Pixel base methods are usually dense methods that is they compute the saliency scores at each and every pixel of the image. Region based methods on the other hand are sparse methods, they only consider a few regions from the image to compute saliency score. The pseudo background methods are effective if the salient objects in the dataset don’t touch the image boundaries. The discriminative saliency features include methods like [1], which give more importance to modelling the salient data with a mathematical model.

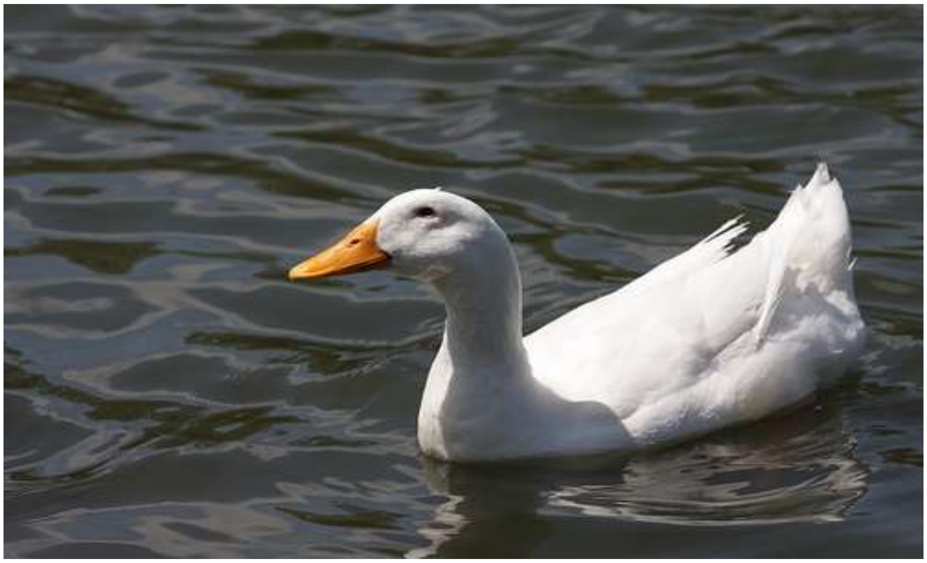

Figure 1: Original image from iCoseg dataset

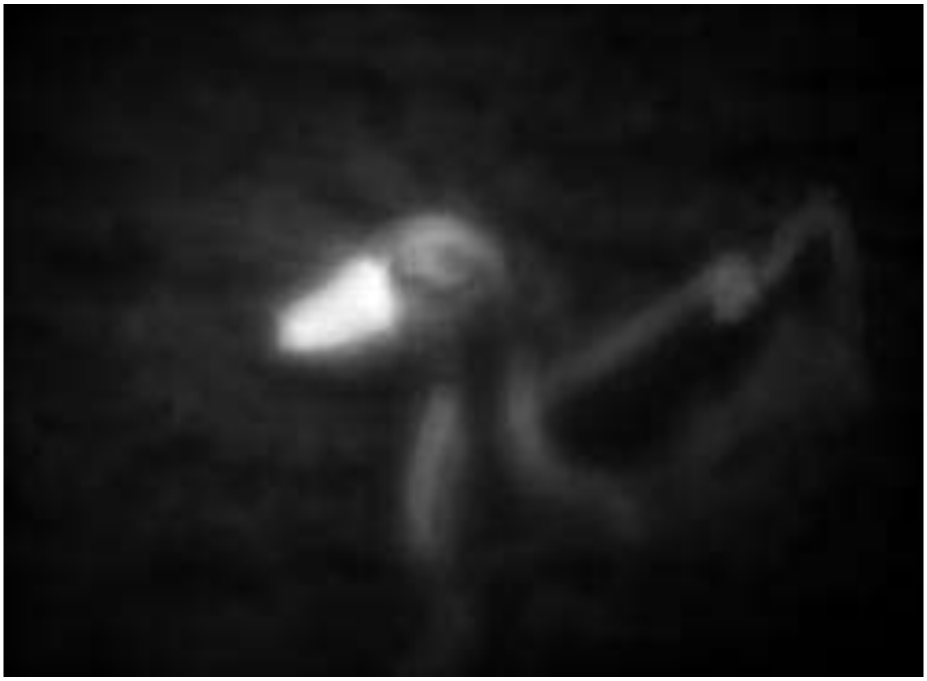

Figure 2: Saliency map image for Figure 1

The most basic assumption used in all these methods is the fact that the object of interest is strikingly different from the surrounding. Most of the methods [1, 2] cannot detect more than one object of interest from the images. These methods will only look for the target object in all the images and will suppress any distractors. However there have been a few methods [5] which consider that both the target object and the distractors share some common visual attributes which can be used effectively to detect the object of interest.

The problem of object detection and discovery has been tackled extensively in the computer vision field. Although the results of most of these approaches are promising they impose certain condition on the captured objects. They either require the object to be perfectly centered or require that the foreground and the background be very different for successful object detection. However these conditions are very hard to be satisfied in practice. For this thesis work, we propose a supervised learning approach instead of the semi-supervised approach proposed in [1]. In the next paragraph we discuss the method proposed in [1] briefly and then we discuss the algorithms reviewed for saliency score computation.

The authors are treating the object detection problem as an unsupervised learning problem in [1]. The input to this algorithm is a set of unlabeled images and the number of classes in the dataset. Their method can be divided into three steps. First they divide the data into positive and negative bags based on the computed saliency score. Second, they collect S most salient windows from each image and derive initial class labels using K-means. They are using the method proposed in [2] for computing the saliency scores. An evaluation score is also computed during this step which measures the probability of the jth point from bag x_i being associated with class k. It is computed using the following expression,

$$q^k(x_{ij}) = [1 + \exp(-\sigma||x_i - c_k||^2)]^{-1}$$

Where c_k represent the K centroids obtained after clustering the S most salient windows.

They then formulate the problem as a weakly supervised multiple class learning with two hidden parameters: H_k containing observed bag-level cluster labels and H_y as unobserved instance level cluster labels. A positive bag is labelled as +1 and a negative bag is labelled by -1. The instance labels are updated using the discriminative EM algorithm. The discriminative EM algorithm differs from the standard EM algorithm and it (the disc EM algorithm) minimizes the following loss function,

$$l(\theta, Y, X) = -\log(\sum_{H_K}\sum_{H_Y} \Pr(Y, H \mid X; \theta))$$

Where $\theta = \{g^1, g^2 \ldots g^K\}$ represents the model parameter and g^k is the appearance model for the k^{th} object class. In the above expression, all the bags are represented by $X = \{x_1, x_2 \ldots x_n\}$ and their labels are represented by $Y = \{y_1, y_2 \ldots y_n\}$ and H represents the pair of hidden variables H_y and H_k.

The probability $\Pr(Y, H \mid X; \theta)$ is defined by the following expression,

$$\Pr(Y, H \mid X; \theta) = \prod_{t=1}^{K}\prod_{i=1}^{n}[\left(q_i^t\right)^{1(t = k_i)}\left(1 - q_i^t\right)^{1(t \neq k)} . s_i]$$

Where, K is the number of classes in the dataset, n is the total number of bags in the dataset (includes both the positive and negative bags) and q_i represents the measure for at least one instance x_{ij} in bag x_i belonging to the t^{th} class and it is defined as follows,

$$q_i^t = 1 - \prod_{j=1}^{m}(1 - p_{ij}^t)$$

And the probability p_{ij}^t is derived from the above computed evaluation score.

$$p_{ij}^k = \Pr\left(k_{ij} = k \,\middle|\, x_{ij}; \theta\right) \propto \prod_{t=1}^{K}\left(q_{ij}^t\right)^{1(t = k_i)}\left(1 - q_{ij}^t\right)^{1(t \neq k)}$$

In the third and final step they use the K learned object models to perform object detection and to assign class labels.

The authors tested the results to this algorithm on SIVAL, CMU-Cornel iCoseg and a 3D object category dataset. For the feature vector they are extracting color moments, edge histograms and GIST features from the images. For each image, the positive bags contain 70 most salient windows and the negative bags contain 40 least salient windows. This method is very efficient compared to other existing methods for salient object detection, but there are a few limitations: inability to detect more than one object in the image and the output of the classification is influenced by the initialization of clusters in the EM algorithm. Although in their algorithm they are using derived labels, the method still remains unsupervised, since we don’t have an idea of the class labels.

We tried four saliency detection algorithm [2, 3, 4, 5] to compare their results. All the codes/algorithms were implemented on a laptop, with core i3 processor and 4GB of RAM. MATLAB was used to implement and test the results. We tried implementing [2] since it was used in [1] to compute the saliency score. The method discussed in [2] defines an image window’s saliency as the cost of composing the window using the remaining part of the image. They believe that the more a segment is split by the window boundary, the less salient the window becomes. Given an image window, the composition problem is defined as finding optimal parts of the same area as that window. They are using a segmentation based representation since it is compact and informative. The image is divided into segments using [17]. For two segments p and q in the segmented image, they are computing two distances: 1) the appearance distance represented by d_a, which is the intersection distance between their LAB color histograms. The intersection distance is defined as,

$$d(H1, H2) = \sum_{I} \min(H1(I), H2(I))$$

2) The spatial distance represented by d_s, which is the Hausdorff distance normalized to be between [0,1] by the longer image dimensions. Their composition cost is defined as,

$$c(p,q) = [1 - d_s(p,q)] \cdot d_a(p,q) + d_s(p,q) \cdot d_a^{max}$$

Where d_a^{max} is the largest appearance distance in the image

The image is divided into windows. For each window they find the segments inside the window and the segments outside the window. Then for each segment inside the window, they find their active area i.e. the area of the segment inside the window. Now for each segment inside the window they add the minimum cost of composing that segment based on the above formula. They repeat the procedure for all the windows in the image. This computed cost for each window is the saliency score for that window.

The partial code that we implemented had a time complexity of O(N^4) (N is the number of segments in the segmented image), which was very high compared to other saliency detection algorithm that were implemented. Due to the large time complexity we chose to not consider the results for the method in [2].

The authors of [3] believe that the image information can be decomposed into two parts: information associate with the novelty part and the redundant information that should be suppressed by coding systems. In image statistics these redundant information corresponds to statistical invariant properties of our environment. The method proposed in this paper is based on the 1/f law. The law states that on a log-log scale, the amplitude spectrum of natural images averaged over orientation, lies approximately on a straight line. The information that jumps out of this straight line curve, is considered to be salient. These results also hold true for a log scale representation. Given an image, the Log spectrum is computed from the down-sampled image with its vertical dimension equal to 64px. Then they compute the average spectrum which is obtained by convolving the log spectrum with a uniform weighted mask. The residual spectrum is obtained by subtracting the averaged spectrum from the log spectrum which is plotted and analyzed to get the salient object from the image.

This method is very easy to implement and was very fast. It provides accurate results for images where the object of interest is noticeably different from its surrounding. But this method doesn’t provide the correct output for images where the object of interest blends well with the background.

The results for [3] can be found in fig. 3. The image on the left is the original image, the image in the middle is the saliency map image and the image on right is the output of the algorithm.

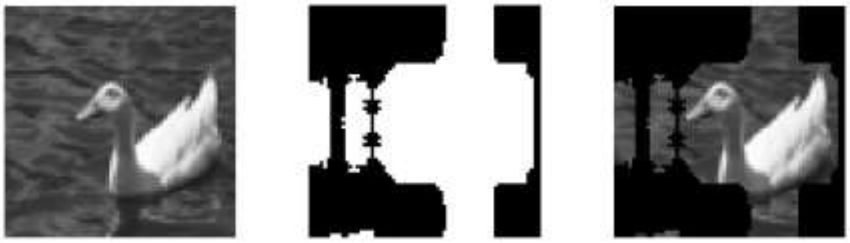

Figure 3: Results for paper [3]

Although this method is very fast, the output image (the image on the far right) in fig. 3 does contain some part of the background. This method may not work if the object of interest is very similar to its background. Also they don’t account for scale variations in their algorithm.

The method proposed in [4], is a regional contrast based saliency score computation algorithm, which simultaneously evaluates global contrast differences and spatial coherence. The method used here is a histogram based contrast method to define saliency values for image pixels using color statistics of the input image. The saliency value of a pixel is defined as the sum of the Euclidean distances between that pixel’s LAB value and all other unique pixel values in the LAB color space. The saliency score is computed by using the following equation:

$$S(I_k) = \sum_{\forall I_i \in I} D(I_k, I_i)$$

They are using a histogram based approach to speed up the computation under the assumption that images seldom contain all possible RGB values. Each channel is quantized to contain only 12 different and distinct values. We tried implementing their method, but were not as promising as compared to other methods reviewed. The time complexity of this method was O(N) where N is the number of unique pixel values in the image. This method was observed to be very fast as compared to [2]. We felt that some valuable information may be lost in the process of quantizing each channel to contain only 12 values.

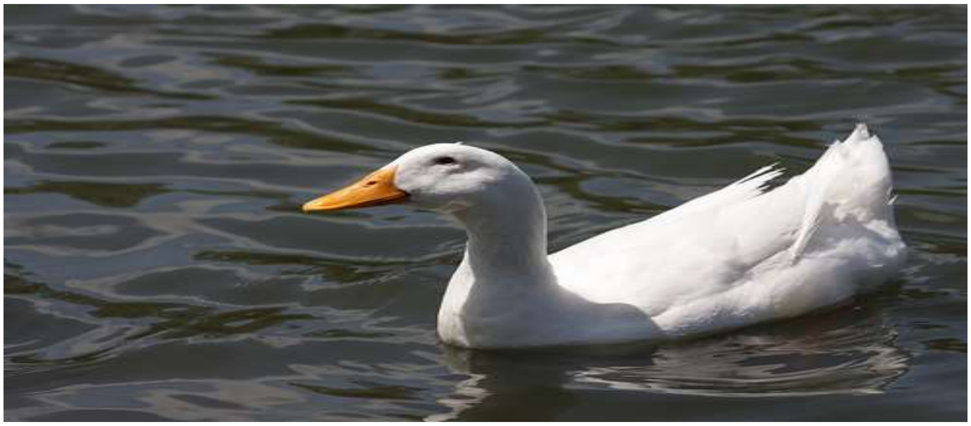

Figure 4: Original Input Image

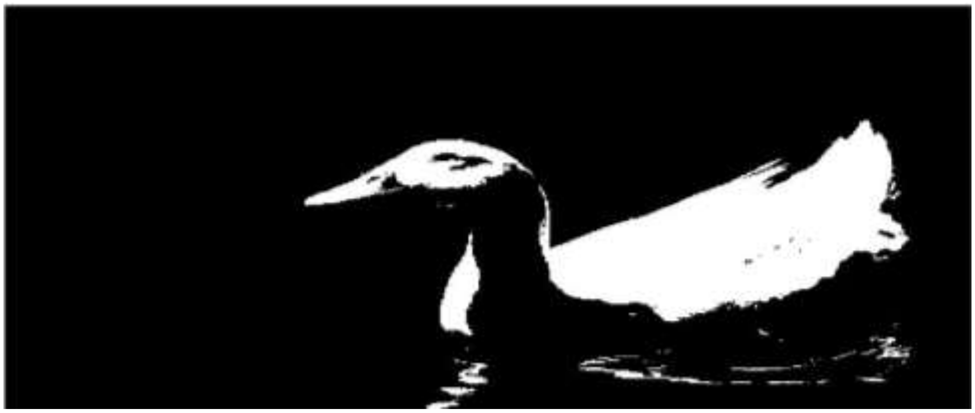

Figure 5: Output Image for fig. 4

Fig. 4 is the input image to the algorithm in paper [4]. Fig. 5 is the output image for fig. 4. From fig 5, we can clearly identify the object of interest in the image, but overall this particular method was very sensitive to the threshold value for converting the saliency map image to black and white.

The authors of [5] believe that not only is the object important in the image, but its background conveys context which might be useful in the identification process. Their method is based on the idea that salient regions are distinctive with respect to both their local and global surroundings. First the image is divided into 7x7 overlapping patches. They then define a dissimilarity matrix which is proportional to the difference in appearance and inversely proportional to the positional distance. They compute two distances: 1) d_{color} which is the Euclidean distance between the vectorized patches centered at pixel i and j and 2) $d_{position}$ which is the Euclidean distance between the position of patches centered at pixels i and j. Using these two computed distances they define a dissimilarity measure defined by,

$$d(p_i, p_j) = \frac{d_{color}(p_i, p_j)}{1 + c \cdot d_{position}(p_i, p_j)}$$

The value for the constant c in the above equation is 3. This value of the parameter c in the above expression is not justified. It appears that the authors have just hand-picked this value by using heuristic methods. No experimental justification is provided for choosing the value of c as 3. Pixels i having high value of the dissimilarity measure for all pixels j, is considered to be salient. We are using the MATLAB code provided by the authors to test this algorithm. Amongst all the methods that were reviewed, this method was found to be the most efficient and reliable.

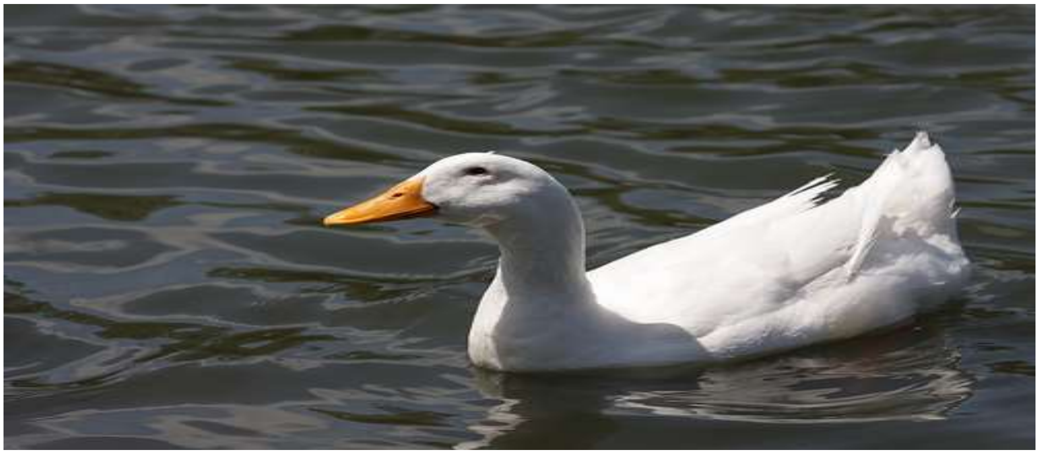

Figure 6: Original Input image

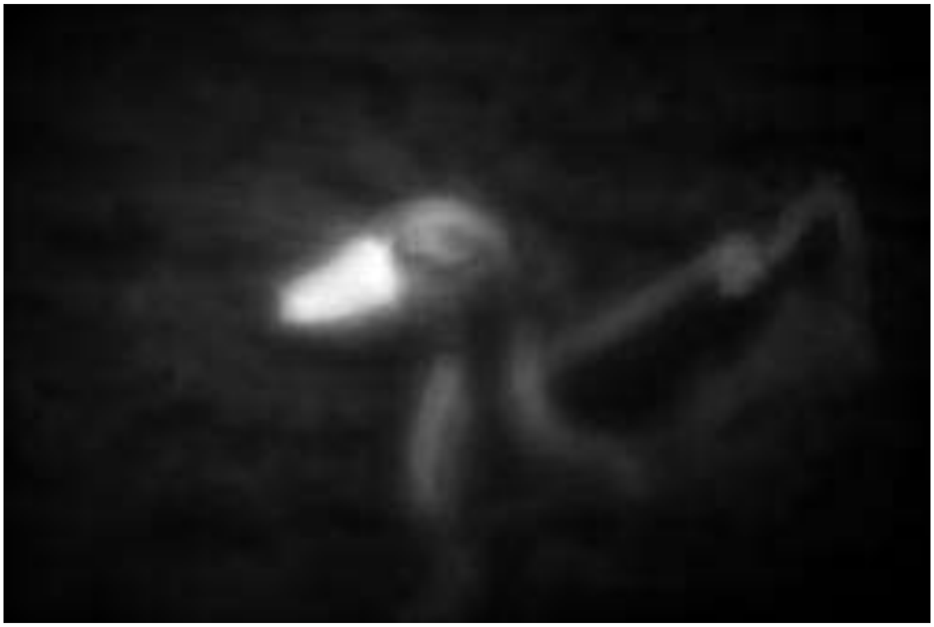

Figure 7: Output image for fig. 6 using method [5]

Fig. 6 is the input image and Fig. 7 is the output image for method in [5]. As we can see from Fig. 7 we can clearly identify the object of interest, in this case the goose from the output image. We are using this method for computing saliency score in our algorithm. The image in Fig. 6 was taken from the CMU Cornell iCoseg dataset.

Chapter 3

Methodology

In this chapter we present our algorithm for clustering and classification. We propose a variant of the K-means clustering algorithm in this work. The main idea of this algorithm is to represent each class by a group of cluster families. A cluster family can be thought of as a “sub-class” to represent bags of feature vectors. We provide a pseudo code for the algorithm. But before presenting the algorithm, we first introduce the dataset used to test the efficiency of the algorithm.

Dataset

We use the CMU Cornell iCoseg dataset in this thesis work. The algorithm presented in [1] uses the same dataset. This is a very standard dataset used in object detection and object classification problems. This dataset was chosen since it has a clear categorization of different objects which is really helpful when dealing with object detection problems. We believe that this is a good way to establish comparison between our work and the method presented in [1]. The result for the algorithm can be found in chapter 4.

Algorithm

We propose our clustering and classification method in this section. We extract a 520 dimensional feature vector from each salient data point in all the images from the dataset. The algorithm for feature extraction is presented next. The algorithm provided in [5] is used for saliency score computation. From all the reviewed method for saliency score computation, this method was found to be better. The MATLAB code for this algorithm was obtained from the author’s website.

The feature extraction algorithm is as follows:

1. Divide the image into 7x7 non-overlapping windows.
2. For each window, compute the saliency score
3. Store the image number, window location, saliency score, color moments and GIST features for the window.
4. Repeat the above steps for all windows in the image and all images in the dataset

A 520 dimensional feature vector is extracted from each window. For each image, the feature vectors are sorted in descending order of their saliency scores. Only the top 100 feature vectors are stored and all the remaining feature vectors are discarded.

Supervised Clustering algorithm

Since the clustering algorithm is supervised we know the class labels for the images in the dataset. Each class in the dataset is represented by a group of cluster families. A cluster family can be thought of as a “sub-class” to represent bags of feature vectors. Each image in a class is assigned to a cluster family from the group of cluster families representing that particular class. The dataset is divided into training and test set. From the training set a certain number of images are randomly selected for initializing the cluster families. The following iterative approach is used to initialize and update the cluster centers for these cluster families. This algorithm is an extension of the standard K-means clustering algorithm.

Before executing the algorithm, the images are grouped based on their class labels. For each class the images are then divided into training and test set. The following steps repeated for each class in the dataset.

The supervised clustering algorithm is as follows:

1. Initialization Step
    a. Select M images from the training set for initializing the cluster families
    b. Using K-means, define ‘n’ centroids for each of the M cluster families
2. Assignment Step
    a. For each image in the training set, find the sum of minimum Euclidean distance from each of the M cluster families
    b. Find the minimum distance and assign the image to that cluster family and its(the image’s) feature vectors to clusters within this cluster family with the minimum distance
    c. Repeat the steps for all the images in the training set
3. Update centroids Step
    a. For each ‘n’ clusters within the M cluster families, update the centroids by first aggregating the values of the feature vectors assigned to that cluster and then by dividing the aggregate by the number of points assigned to that cluster

The above mentioned steps are repeated for all the classes in the dataset. We then classify the points in the test set using the following algorithm

Classification algorithm based on the cluster families obtained from the above algorithm:

1. For each image in the test set, find the sum of minimum Euclidean distance from all the cluster families
2. Assign the image to the cluster family with the minimum value

The results to the above algorithm can be found in chapter 4.

The clustering method discussed and presented above, is very similar to the standard K-means clustering approach. The following equation lists the distortion function for the method discussed. This distortion cost is computed for each class in the dataset.

$$D = \sum_{i=1}^{\#\,images} \sum_{j=1}^{\#\,families} z_{ij} \cdot \sum_{k=1}^{\#\,clusters} v_{jk} \cdot \sum_{l=1}^{\#\,points} ||x_{(i,l)} - y_{jk}||^2$$

The variables v_{ij} and v_{jk} in the above expression are indicator variables, and can be defined as follows,

$z_{ij} = 1$ $\quad \sum_l v_{jk|l} \cdot \left|x_{(i,l)} - y_{jk}\right|^2 < \sum_l v_{jk'|l} \cdot \left|x_{(i,l)} - y_{jk'}\right|^2$

$z_{ij} = 0$ $\quad$ otherwise

$$\left.\begin{array}{ll} v_{jk|l} = 1 & \left|x_{(i,l)} - y_{jk}\right|^2 < \left|x_{(i,l)} - y_{jk'}\right|^2 \quad \forall k \\ v_{jk|l} = 0 & \text{otherwise} \end{array}\right\} \begin{array}{l} \text{l = 1…#points,} \\ \text{j = 1…#families} \end{array}$$

Next we present a graph to suggest experimentally that the above distortion is strictly decreasing with each iteration of the clustering algorithm. Since, we have 6 classes in our dataset, we present the following 6 graphs for the distortion cost vs the number of iteration of the clustering algorithm

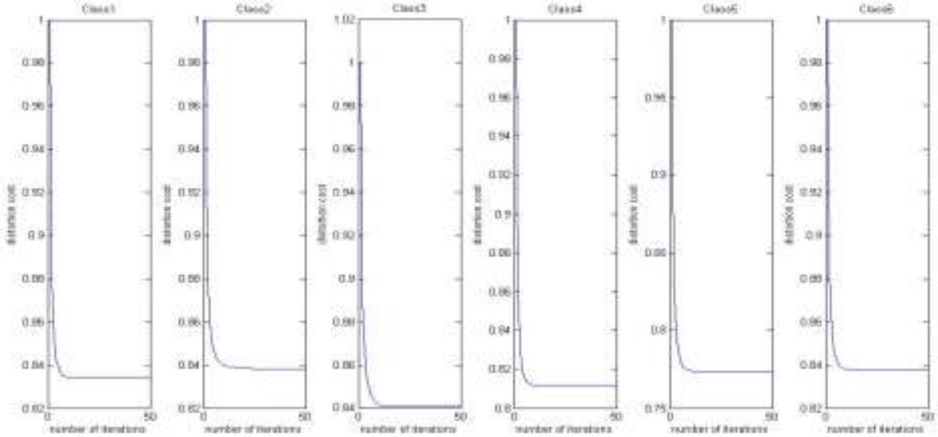

Figure 8: Distortion cost vs number of iterations

As can be seen from the above graph the value of the distortion cost decreases as the value for number of iteration increases for each class. Also if we observe the graphs closely, after a certain number of iterations, all the 6 graphs stay constant, which implies that out algorithm converges.

Discriminative Classifier

The supervised clustering method presented above only weekly uses the class labels to partition the training data. The results for the above algorithm can be further improved if the class labels are used to learn a discriminative classifier. So, above clustering algorithm can be used for initializing a discriminative classifier. We define the data for each class using Gaussian distribution. Then using a gradient descent algorithm, the values for the parameters of the Gaussian model are updated till the value of the error function is less than a threshold.

We define the error function as,

$$E = \sum_{i=1}^{T}\sum_{j=1}^{K} \delta(class(i) \neq j) * \ P_{j/i}$$

Where, T = number of training Image in the dataset

K = number of classes in the dataset

class() = class of image ‘i’

The probability used in the above error function is defined as,

$$P_{j/i} = \sum_{l=1}^{M} P_{lj/i}$$

Where M = the number of families in each class

The quantity $P_{lj/i}$ is the joint probability of belonging to class j and cluster family l for the ith image. The joint probability is defined as,

$$P_{lj/i} = \frac{\prod_{m=1}^{N}\sum_{r=1}^{n} f(x_{(i,m)}, \mu_{(j,l,r)}, {\sigma_{(j,l,r)}}^2)}{\sum_j \sum_l \prod_{m=1}^{N}\sum_{r=1}^{n} f(x_{(i,m)}, \mu_{(j,l,r)}, {\sigma_{(j,l,r)}}^2)}$$

In the above expression the function f is a Gaussian distribution with mean μ and variance σ. It is defined as,

$$f(x, \mu, \sigma) = \frac{1}{\sqrt{2 * \pi * \sigma^2}} * e^{-{(x - \mu)^2}/{(2 * \sigma^2)}}$$

After computing all the probability values defined by the above equations, we then update the values of mean and variance using the gradient descent algorithm,

$$\mu_r(j, l)_{new} = \mu_r(j, l)_{old} - \gamma * \frac{\partial}{\partial \mu_{(j,l,r)}} E$$

$$\sigma_r(j, l)_{new} = \sigma_r(j, l)_{old} - \gamma * \frac{\partial}{\partial \sigma_{(j,l,r)}} E$$

We repeat the above steps till the value of the error function doesn’t fall below a set threshold. The value of the constant γ is chosen so that the algorithm strictly descends in E.

The derivative of the error function E in the above expression can be computed as follows,

For the derivative with respect to μ, the joint probability function is a function of the mean. Hence we consider the derivative of the joint probability function $P_{lj/i}$ with respect to μ. We can also write the joint probability function as,

$$P_{lj/i} = \frac{e^{\sum_{m=1}^{N} \log(\sum_{r\prime=1}^{n} f(x_m^{(i)}, \mu_{r'l'}^{(j)}, \sigma^2))}}{\sum_{j\prime} \sum_{l\prime} e^{\sum_{m=1}^{N} \log(\sum_{r\prime=1}^{n} f(x_m^{(i)}, \mu_{r'l'}^{(j)}, \sigma^2))}}$$

For simplification of the derivative, we use the following notation,

$$Q(m, i) = (\sum_{r'=1}^{n} f\left(x_m^{(i)}, \mu_{r'l'}^{(j)}, \sigma^2\right)$$

$$N = e^{\sum_{m=1}^{N} \log(\sum_{r\prime=1}^{n} f(x_m^{(i)}, \mu_{r'l'}^{(j)}, \sigma^2))}$$

And,

$$Z(j',l') = \ e^{\sum_{m=1}^{N} \log(\sum_{r\prime=1}^{n} f(x_m^{(i)},\mu_{r'l'}^{(j)},\sigma^2))}$$

$$Z = \sum_{j\prime}\sum_{l\prime} e^{\sum_{m=1}^{N} \log(\sum_{r\prime=1}^{n} f(x_m^{(i)},\mu_{r'l'}^{(j)},\sigma^2))}$$

So we can write the derivative as,

$$\frac{\partial P_{lj/i}}{\partial \mu_{\tilde{r}\tilde{l},\tilde{k}}^{\tilde{j}}} = \frac{\frac{\partial N}{\partial \mu_{\tilde{r}\tilde{l},\tilde{k}}^{\tilde{j}}} \cdot Z - \frac{\partial Z}{\partial \mu_{\tilde{r}\tilde{l},\tilde{k}}^{\tilde{j}}} \cdot N}{Z^2}$$

$$\frac{\partial N}{\partial \mu_{\tilde{r}\tilde{l},\tilde{k}}^{\tilde{j}}} = \begin{cases} N \cdot \sum_{m=1}^{N} \frac{\partial Q(m,i)}{\partial \mu_{\tilde{r}\tilde{l},\tilde{k}}^{\tilde{j}}} \cdot \frac{1}{Q(m,i)} & (\tilde{l},\tilde{j}) = (l,j) \\ 0 & (\tilde{l},\tilde{j}) \neq (l,j) \end{cases}$$

Where,

$$\frac{\partial Q(m,i)}{\partial \mu_{\tilde{r}\tilde{l},\tilde{k}}^{\tilde{j}}} = \left(\prod_{k \neq \tilde{k}} f\left(x_{m,k}^{(i)}, \mu_{\tilde{r}l,k}^{(j)}, \sigma^2\right)\right) \cdot \left(\frac{-2 \cdot \left(x_{m,\tilde{k}}^{(i)} - \mu_{\tilde{r},l,\tilde{k}}^{(j)}\right)}{2\sigma^2}\right) \cdot f(x_{m,\tilde{k}}^{(i)}, \mu_{\tilde{r},l,\tilde{k}}^{(j)})$$

And,

$$\frac{\partial Z}{\partial \mu_{\tilde{r}\tilde{l},\tilde{k}}^{\tilde{j}}} = Z(\tilde{j},\tilde{l}) \cdot \sum_{m=1}^{N} \frac{\partial \tilde{Q}(m,i)}{\partial \mu_{\tilde{r}\tilde{l},\tilde{k}}^{\tilde{j}}} \cdot \frac{1}{\tilde{Q}(m,i)}$$

Where,

$$\tilde{Q}(m,i) = \sum_{r'=1}^{n} f(x_m^{(i)}, \mu_{r',\tilde{l}}^{\tilde{j}}, \sigma^2)$$

$$\frac{\partial \tilde{Q}(m,i)}{\partial \mu_{\tilde{r}\tilde{l},\tilde{k}}^{\tilde{j}}} = \left(\prod_{k \neq \tilde{k}} f\left(x_{m,k}^{(i)}, \mu_{\tilde{r}l,k}^{(\tilde{j})}, \sigma^2\right)\right) \cdot \left(\frac{-2 \cdot \left(x_{m,\tilde{k}}^{(i)} - \mu_{\tilde{r},l,\tilde{k}}^{(\tilde{j})}\right)}{2\sigma^2}\right) \cdot f(x_{m,\tilde{k}}^{(i)}, \mu_{\tilde{r},l,\tilde{k}}^{(\tilde{j})})$$

Using the above derivative terms, we formulate the gradient rule and thus update the values of the Gaussian model.

We present the results of the supervised clustering algorithm in the next section. Although we were unable to implement the discriminative classifier, we plan to do this in a future work.

Chapter 4

Experimental Results

In this chapter we present the results for the above mentioned algorithms. There are two hyper parameters: the number of cluster families and the number of clusters within the cluster families. But before providing the results, we first go over the features extracted from the images.

Features extracted from Image

In this work we extracted two features from the image: Color moments and GIST features. In [1] and [2] the authors are using three features: color histograms, edge histograms and GIST features. But we decided to ignore edge histograms because in [1] and [2] 64 of the edge histograms features were extracted but their values were very sparse since a window of the image may not contain all the 64 edge orientations. Next we go over the two features very briefly and then discuss the results for our algorithm.

Color Moments:

Color moments are a measure that can be used to differentiate images based on their features of color. Once calculated these moments provide a measurement for color similarity between images. These values are then compared to values stored in a database for tasks like image retrieval. The basis of color moments lies in the assumption that the distribution of color in an image can be interpreted as a probabilistic distribution. Since probability distribution can be characterized by a number of unique moments. From all the color moments we are using moment 1(mean) and moment 2(variance).

GIST features:

GIST stands for Gradient-domain Image STitching. The idea of GIST features is to represent the image with a low dimensional representation of the scene, which doesn't require any form of segmentation [14]. GIST features are mostly used in image search and object detection. Given any input image, the GIST features are computed by using the following 3 steps. 1) Convolve the image with 32 Gabor filters at 4 scales and 8 orientations producing 32 feature maps of the same size of the input image. 2) Divide each feature map image into 16 regions (by a 4x4 grid), and then average the feature values within each region. 3) Concatenate the 16 averaged values for the 32 feature maps, resulting in a 512 dimensional gist descriptor. In short GIST summarizes the gradient information in the scene for different parts in the image, which provides a rough description of the scene. GIST features were first introduced in [15].

Results and observations

The results of the clustering algorithm are presented first. There are two variable parameters here: the number of cluster families used to represent each class and the number of clusters within each cluster family.

From the iCoseg dataset, we select images from the following 6 categories: Ferrari, Goose, Panda, Helicopter, Planes and Gymnastics. The iCoseg dataset contains images for 38 different categories. But images in most of the categories are similar to each other, hence we narrowed the dataset to the above mentioned 6 categories.

For each class, 70% of the images are selected in the training set and the rest are put in the testing set. These images are selected randomly. From all the images in the training set, M images are randomly selected for initialization of the cluster families. We vary the value of number of cluster families from 1 to 8 and the values of number of clusters from 10 to 50 in steps of 10. When the value of number of cluster families is 1, our algorithm reduces to the standard K-means clustering algorithm.

#clusters →	10		20		30		40		50	
#families ↓	Mean	Var	Mean	Var	Mean	Var	Mean	Var	Mean	Var
1	60.24	2.27	60.82	3.28	62.71	2.83	61.09	3.98	60.00	2.60
2	66.24	6.4	65.41	6.95	69.06	5.35	68.12	6.43	71.53	3.83
3	71.76	7.21	70.94	4.51	72.35	6.47	71.29	3.77	75.76	3.89
4	72.94	6.35	75.88	3.20	74.00	6.83	75.29	4.40	75.41	5.22
5	75.18	5.27	73.88	6.17	77.65	3.09	75.65	5.05	76.82	3.04
6	73.88	3.36	76.24	4.11	78.71	2.74	76.82	5.05	76.59	5.96
7	76.71	4.99	80.82	3.51	80.00	4.00	80.35	3.04	75.53	7.37
8	75.41	4.02	79.88	3.66	77.65	3.14	80.00	4.99	79.41	5.80

Table 1: Mean and variance for classification rate (training data)

#clusters →	10		20		30		40		50	
#families ↓	Mean	Var	Mean	Var	Mean	Var	Mean	Var	Mean	Var
1	56.87	7.91	55.00	5.74	56.56	5.79	60.63	5.74	57.81	4.94
2	64.06	8.49	63.75	4.22	65.63	7.93	66.56	4.90	69.38	8.31
3	67.81	5.52	71.88	6.42	67.81	7.80	71.88	7.51	68.44	8.26
4	67.19	6.29	70.94	7.66	69.69	8.84	72.19	7.28	67.81	8.60
5	65.94	4.02	72.19	7.28	74.38	7.34	68.75	6.42	66.88	8.74
6	68.44	4.02	72.81	8.84	70.31	7.55	69.38	4.37	70.25	6.78
7	67.19	6.63	72.19	7.28	71.25	4.61	69.69	9.09	65.63	5.51
8	73.13	3.67	70.00	7.10	70.00	9.22	75.00	7.37	66.56	3.31

Table 2: Mean and variance for classification rate (test set)

We run the algorithm 10 times for each pair of values (number of families and number of clusters). After recording the results for classification rate for each run, we compute the mean and variance which can be observed in the above table. The maximum value of classification rate for the training set was: 88.24% and for the test set it was: 90.63%.

Next we plot the graph for the training and testing classification rates. Figure 8 below is the graph of training set classification rates vs the number of cluster families. We start with the value of number of clusters set to 10, which is represented by the graph on far left. The subsequent graphs are for values of number of clusters from 20 to 50 in steps of 10.

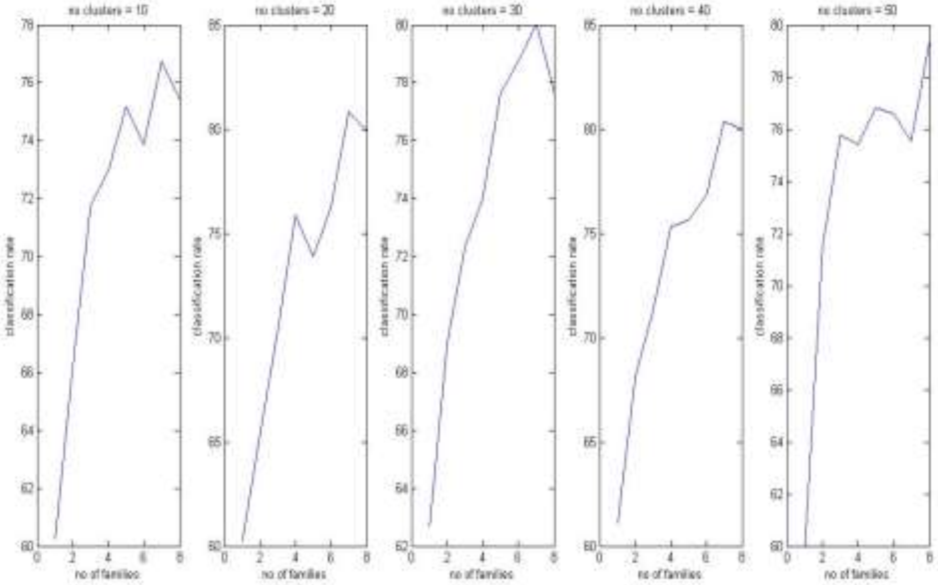

Figure 9: Training Set classification rates

As can be seen from figure 8, the values increases rapidly to a global maxima and the decreases. The maximum value of classification rate is observed when the number of cluster family has a value of 6.

Figure 9 below is the plot of classification rate vs the number of cluster family for the test set. This graph doesn't follow any patter for the changing values of number of clusters. The maximum value for the classification rate is obtained for the value of number of families set to 6, similar to the training set.

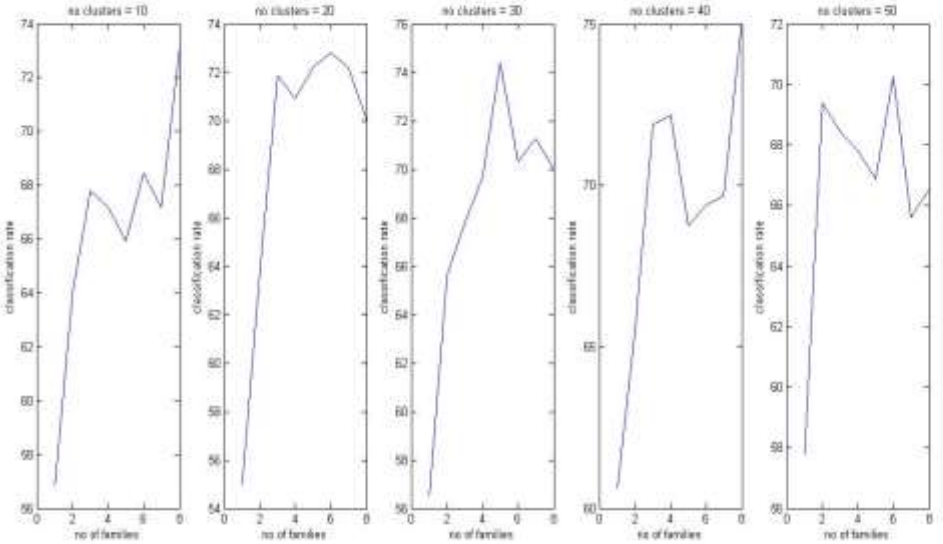

Figure 10: Test set classification rate

We can clearly say that we get the best results when the value of number of cluster families is set to 6 for both the training and the test set.

The results obtained for the clustering algorithm are not very ideal. In ideal case scenario we expect the training set classification rate to be 100% for all the combination of number of cluster families and number of clusters within each cluster family. For the test set classification, ideally the graph should first increase with the number of cluster families and then attain some constant values. We expect both the graphs to be monotonically increasing. We observe here that the classification results are highly influenced by the initialization step in the algorithm.

Chapter 5

Conclusion

As can be seen from the results obtained, the method proposed has promise. There are a few drawbacks for the method proposed in this research work: the method will not work for more than one object of interest in the image. Also this particular method will not work if the object is occluded or partially hidden. This might be due to the use of saliency detection for object detection. Most of the saliency detection method are not able to detect more than one object in the image.

References

[1] JUN-YAN ZHU, JIAJUN WU, YAN XU, ERIC CHANG and ZHUOWEN TU, “Unsupervised Object Class discovery via Saliency-guided multiple class learning”, IEEE transaction on pattern analysis and Machine learning, 2015

[2] J. FENG, Y. WEI, L. TAO, C. ZHANG and J. SUN, “Salient Object detection by Composition”, in Proc. Int. Conf. Computer Vis., 2011

[3] XIAODI HOU and LIQING ZHANG, “Saliency detection: A spectral residual approach”, IEEE Conference on Computer Vision and Pattern recognition, 2007

[4] MING-MING CHENG, NILOY J. MITRA, XIAOLEI HUANG, PHILIP H.S.TORR, and SHI-MIN HU, “Global contrast based salient region detection”, IEEE transactions on pattern analysis and machine learning, 2015

[5] STAS GOFERMAN, LIHI ZELNIK-MANOR and AYELLET TAL, “Context-aware saliency detection”, IEEE transactions on Pattern Analysis and Machine Intelligence, 2012

[6] ALI BORJI, MING-MING CHENG, HUAIZU JIANG AND JIA LI, “Salient Object detection: A survey”, arXiv preprint arXiv:1411.5878, 2014

[7] JINGDONG WONG, “Salient Object Detection”, Microsoft Research

[8] MA and ZHANG, “Contrast based image attention analysis by using fuzzy growing”, ACMMM, 2003

[9] ANCHANTA ET AL., “Frequency-Tuned Salient Region Detection”, IEEE Conference on Computer Vision and Pattern Recognition, 2009

[10] WEI ET AL., “Geodesic Saliency Using Background Priors”, ECCV, 2012

[11] YANG ET AL., “Saliency Detection via Graph-Based Manifold Ranking”, IEEE Conference on Computer Vision and Pattern Recognition, 2013

[12] TIE LIU, JIAN SUN, NAN-NING ZHENG, XIAOOU TANG and HEUNG-YEUNG SHUM, “Learning to detect a salient object”, International Conference of Pattern recognition, 2008

[13] JING and DAVIS, “Submodular Salient object detection”, IEEE Conference on Computer Vision and Pattern Recognition, 2013

[14] MITTHIJS DOUZE, HERVE JEGOU, HARSIMRAT SANDHAWALIA, LAURENT AMSALEG and CORDELIA SCHMID, “Evaluation of GIST descriptors for web-scale image search”

[15] AUDE OLIVA and ANTONIO TORRALBA, “Modelling the shape of the scene: A Holistic representation of the spatial envelope”, International Journal of Computer Vision, 2001

[16] BAIYING LEI, TIANFU WANG, SIPING CHEN, DONG NI and HAIJUN LEI, “Object Recognition based on adaptive bag of features and discriminative learning”, IEEE International Conference on Image processing. 2013

[17] PEDRO F. FELZENSZWALB and DANIEL P. HUTTENLOCHER, “Efficient graph based image segmentation”, International Journal of Computer Vision. 2004