Generating Representative Sets and Summaries for Large Collection of Images Using Image Cropping Techniques and Result Comparison

Abdullah Al-Mamun
Department of Computer Science
University of Trento
Trento, Italy

Dhaval Gandhi
Department of Computer Science
University of Trento
Trento, Italy

Sheak Rashed Haider Noori
Department of Compute Science & Engineering
Daffodil International University
Dhaka, Bangladesh

***Abstract*—The collection of photos hosted on photo archives and social networking sites has been increasing exponentially. It is really hard to get the summary of a large image set without browsing through the entire collection. In this paper two different techniques of image cropping (random windows technique and sequential windows technique) have been proposed to generate effective representative sets. A ranking mechanism has been also proposed for finding the best representative set.**

Keywords—summarization; representative set; image collection; diversity; coverage

I. Introduction

Due to technological growth in electronic gadgets and digital media such as mobiles, tablets, digital cameras, memory cards and many more, there is a big increase in large image collection in our hard drives and in the web storage. The main purpose of capturing photos is to keep and refresh memories about our life events. The new coming trend is to share photos with family and friends using social networking web-sites. Nevertheless, the growth of images raises challenges such as the difficulty in browsing large image set while avoiding large number of duplicates and similar images.

The goal of this paper is to find the diverse representative set and summary of the collection of images. To deal with this problem, firstly, focus has been put on a cropped image that is called window. This means the concentration is placed on some portion instead of whole image to generate the representative set. This approach is initiated from the idea of n-gram model. An n-gram is a subsequence of n items from a given sequence. It is a model [1] based on text and is widely used in statistical natural language processing. The items can be phonemes, syllables, letters, words or base pairs according to the application. The idea is to find out whether this model can be applied to process images and what will be the outcomes. Therefore, two algorithms are proposed for windows cropping, namely random and sequential. That is, the cropping points are generated randomly and sequentially respectively.

On these cropped images image recognition is performed. Because image features have many properties that make them suitable for matching and differing images of an object or scene.

Some of the features are invariant to image scaling and rotation [2]. Scale-invariant feature transform (or SIFT) algorithm is used for finding and computing descriptors of each images. SIFT features are extracted from a set of candidate images and stored in a database. By applying K-means clustering algorithm those descriptors for image features go into clusters respectively. After that the centroid image of all clusters are fetched which means that the centroid images are the representative images of each respective cluster.

As it is intended to have highly representative images, namely it is needed a small set of images that are highly dissimilar; a ranking mechanism is developed to select more representative images from the given image set created above. First, image matching is performed by individual comparison of each features based on Euclidean distance of their feature vectors. Then the ratio of number of matching points to summation number of the detected key points between images is computed. A higher ratio indicates a larger possibility of similarity of two images. At last the ratio is sorted by ascending order, so that we are free by top *n* images from the sorted array. The overall process is depicted in figure 1. The evaluation of the performances of these two methods is performed by subjective evaluation.

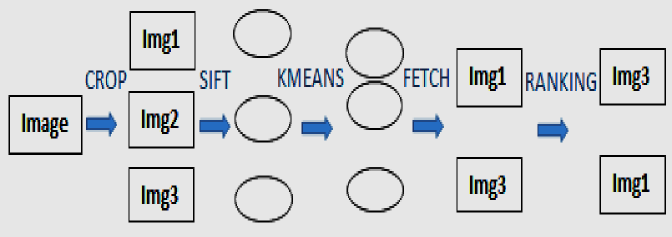

Fig. 1. proposed process of image processing

Some of the important related works are reported in Section 2. The problem statement more specifically is described in Section 3. In Section 4, a brief introduction of the followed approaches, used to generate representative sets is discussed. Section 5 gives the detailed windows cropping mechanism. Section 6 provides the clustering details and in Section 7 the ranking mechanism is provided. The evaluation of the experiment is presented in Section 8.

II. Related Work

The problem for generating representative set and summarization has been done by many researchers. Some of the excellent and most related work are reported here. Scene Summarization for Online Image Collections by Simon [5], examines the distribution of images in the collection to select a set of canonical views to form the scene summary, using clustering techniques on visual features using SIFT. The author summarized all images based on likelihood, coverage and orthogonality. The proposed approach followed in this paper is similar to the clustering technique and selection categories. But, the key difference between [5] and the proposed approach is generating windows before clustering phase. In [6], the author summarizes images based on spatial patterns in photo sets, as well as textual-topical patterns and user (photographer) identity cues. The key difference between [6] and the proposed approach is that the geo-referenced images are not used in the experiments of the later case. In this paper strong focus is placed on low level features of the images. In [7], the author generates diverse and representative image search results for landmarks based on context- and content-based tools. To do that the author used location and other metadata as well as tags associated with images, and the images' visual features. The work [7] is somewhat different with Simon because the author starts from the tags that represent landmarks while the author used also SIFT for the visual features comparison. The difference with the approach proposed in this paper is that metadata associated with images as well as image tags are not used. The only concentration is put on visual features of images.

In [8], the problem is of selecting iconic images to summarize general visual categories. The author defined conic images as high-quality representatives of a large group of images consistent both in appearance and semantics. The approach to find such groups was to perform joint clustering in the space of global image descriptors and latent topic vectors of tags associated with the images. The author has also used a ranking mechanism. The key difference is that in [8], it has assumed that there is one iconic view of the scene rather than a diverse set of representative views as are shown in this work.

In the absence of location metadata, temporal metadata was also considered in the past for the purpose of photo collection summarization. In [11], Graham et al. describe an algorithm to heuristically select representative photos for a given time period in a personal collection, utilizing patterns in human photo-taking habits. Additional time-based work aims to detect events in personal collections [10], which could be the basis for collection summarization. However, again, all these projects considered single-photographer collections only. Several projects [9, 10] use geographic data to organize photo collections in novel ways, for example, by detecting significant events and locations in a photo collection. Such structures could indeed be the basis for collection summarization. However, geographic data of the images used in the experiment presented in this paper are not available. In medical science the image processing is also being conducted to quantify how body cells move, divide and die over time. Traxtile [12] has been developed with Python program as a tool for directly measuring rates of cell division and death, by observing events under various conditions using time-lapse imaging, where the cell of interest is centered in an image tile that is cropped and zoomed from the original, and corresponding tiles from preceding and following frames are displayed in a montage.

In general the work presented in this paper is different in terms of windows generation technique in initial stage and ranking mechanism.

III. Problem Statement

It is initiated by defining some terminologies. Throughout the report, the term photo is used interchangeably with image, all of which refers to an ordinary 2D image. Collection is defined as a set of photos and windows as cropped images. The representative set is loosely defined as a subset that captures representativeness, relevance and breadth in the original collection.

Two different notations for the random windows and sequential windows are used. For the random window and for the sequential window, IN-C and SN-C are used respectively, where *N* is number of windows of each image and C is coverage need to be covered.

For example, with number of random or sequence windows that are 3 and with coverage 85%, it is represented as I3-85 if random images and S3-85 for sequential images respectively. A summary is a set of photos ordered by applying ranking mechanism and selecting any arbitrary number of images from the given representative set.

Given a set of photos P, the goal is to compute a representative set RS ⊆ P and then summary S ⊆ RS such that S represents highly diverse representative images of the photo set P.

IV. Approach for Generating Representative Set and Summary

In this section the selection criteria for the representative set and overview of the proposed solution are introduced.

A. Selection Criteria

As there is no accurate formal model which constitutes a "good" representative set and summary of a collection of images, some simple heuristics are followed that try to model and capture human attention. These heuristics are as follows:

- Images are taken at a location that provides views of some important objects or landmarks.
- Image is more relevant and should be included in representative set, if it matches with many other images of the collection. The representative set and summary should contain highly distinct or diverse images.

B. Overview of Proposed Solution

The proposed solution has three different phases. In the first phase "Pre-cluster" there are three different techniques, followed by second phase "Clustering" phase which generates the representative set and the last phase is "Ranking mechanism" through which the summary is generated. The overview can be seen in the figure 2.

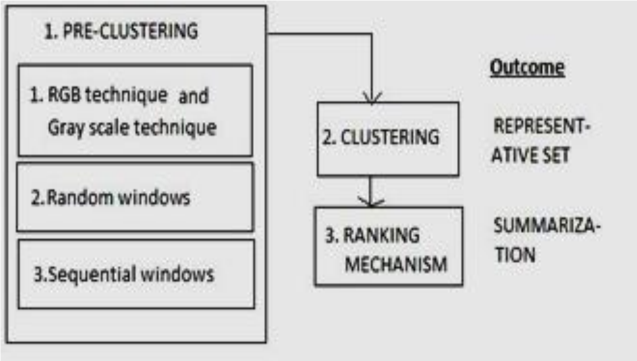

Fig. 2. Overall scenario of proposed solution

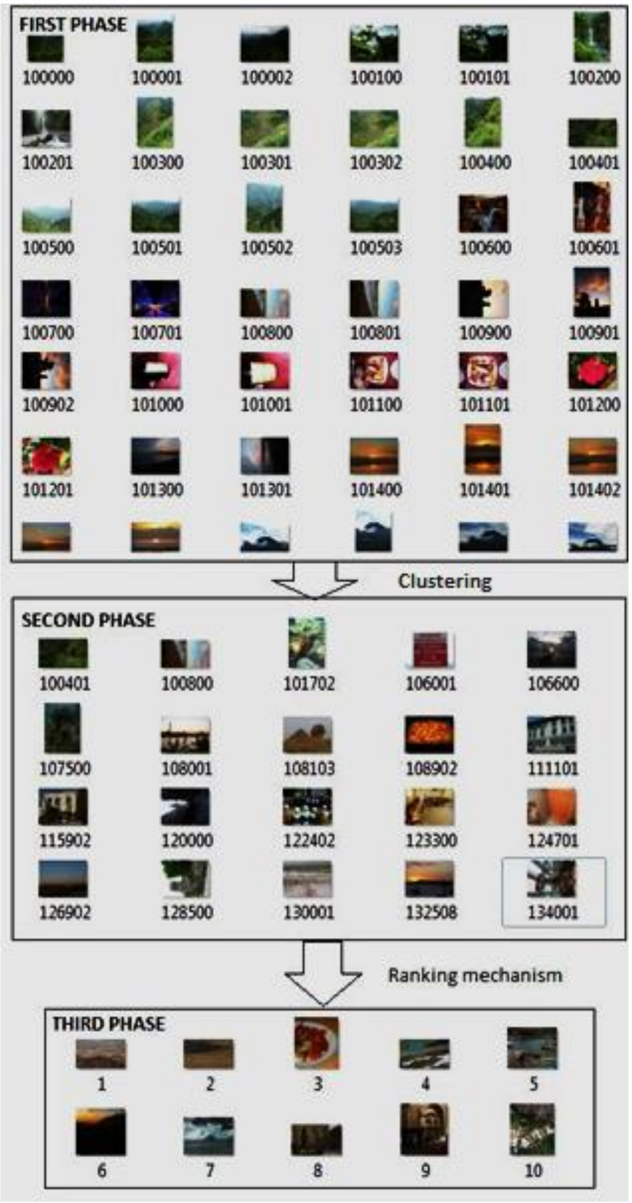

Fig. 3. Overall output scenario of proposed solution

In particular:

- Different pre-clustering techniques namely random and sequence are applied in order to generate windows. The input will be an original large image set.
- Clustering was applied to group appearance-wise similar images. The input will be windows generated from the pre-cluster phase. Then centroids of all clusters are fetched to generate a representative set.
- Ranking mechanism is applied on the representative set and a summary is generated by selecting any arbitrary number of top ranked images. So the input will be the representative set and the output will be a summary.

By applying these three phase, it will get the final outcome “summary”. One can see the graphical overview of the scenario in figure 3 for summary of 10 images.

In the following subsections the algorithm is explained. First, different pre-clustering techniques are described. Then the clustering algorithm for fetching centroid is provided. Finally, how to rank each image of the representative set in order to generate summary is presented.

V. Pre-Cluster

In pre-cluster three different techniques are used which are as following.

A. RGB and Gray scale Techniques

Using these two pre-cluster techniques, images are stored in RGB form and clustering algorithm (k-means) is applied. For gray scale technique, all the images are converted in gray scale and are stored in vectors and again the cluster algorithm is applied in second phase. From the result of clusters it is found that the pictures with same appearance with different pixel values do not go to the same cluster. It means taking a picture of any particular landscape with normal mode and taking the same picture with zoom, would not go in the same clusters although both have same visual features. Because of this big drawback it is not possible to stand on finding a representative set and summarization which holds the selection criteria. Therefore, experiments with this pre-cluster stage are stopped and focus is set only on the next two techniques.

B. Random Windows

As it is mentioned earlier that the idea of cropping images initiated from n-gram model, however random windows are not suitable for the n-gram model. It is initiated with taking n-windows of same size with certain fix coverage of each images from random pixel point of an image. More formally, for each image of the dataset *P* and for the fix coverage *C* (i.e. window should cover some fixed portion of image like 85%, 75% and 66%), generate *N* random windows from (r_x, r_y) pixel point (i.e. 3 windows of 75% coverage of each images) where r_x and r_y is random point from where the image with fix coverage will be cropped.

One can see from the figure 4 that the black dots are windows initial points which are totally random. After having the N windows set of original data set, sift algorithm is applied to generate and store descriptors of each random windows for the next stages. Descriptors are K-by-128 matrix, where each row gives an invariant descriptor for one of the *K* key points. The procedure is mentioned in algorithm 1 and the output can be seen in figure 5 with respect to the original image.

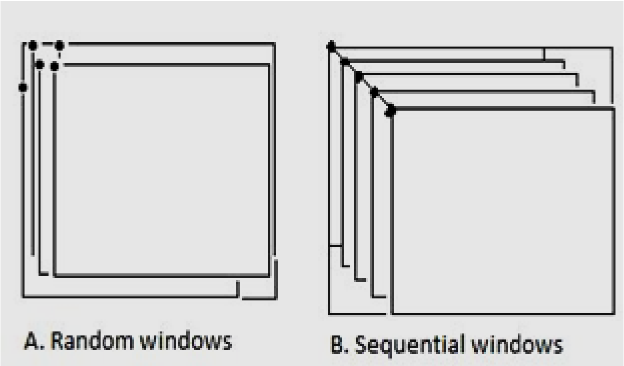

Fig. 4. Random and sequential windows

Algorithm 1: Random windows generation

```
Input: original image set
Output: N random windows of each images with certain coverage C.
1: set the coverage C and the number of windows N to generate random windows
2: for each image img do
get the size of img [rows columns] = size(img) manipulate window width and height
according to coverage C:
        w_width = C * column;
        w_height= C * rows;
3: for each image img, to generate N random windows do
generate random pixel point (r_x,r_y);
crop the image img : cropped_img = imcrop(img, [r_x, r_y, w_width, w_height])
save cropped_img (window) at the output directory
4: end for
5: end for
```

C. Sequential Windows

In this technique, N sequential windows are generated in order to implement n-gram model. So, the coverage C remains same as the previous technique but the window point is now fixed. More formally, for the each image of the dataset P and for the fix coverage C, N sequential windows are generated from (s_x,s_y) pixel point where s_x and s_y is sequential pixel point from where the image with fixed coverage will be cropped. Here sequence of windows are kept diagonally. One can see from the figure 4, windows initial points which are totally diagonal.

The idea is to capture certain coverage without losing any sort of information. In other words for any N windows, we must have all images between (0,0) and (X,Y) where X and Y are height and width of an original image, diagonally.

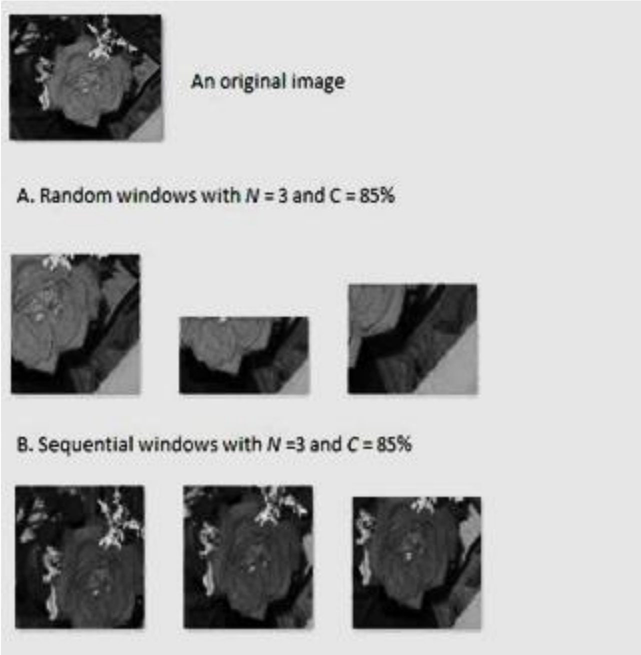
An original image
A. Random windows with N = 3 and C = 85%
B. Sequential windows with N =3 and C = 85%

Fig. 5. Output of random and sequential windows

After having the n sequential windows set of original data set, we apply sift algorithm to generate and store descriptors of each random window for the next stages. The procedure is mentioned in algorithm 2 and the output can be seen in figure 5 with respect to the original image.

Algorithm 2: Sequence windows generation

```
Input: original image set
Output: N sequential windows of each image with certain coverage C.
1: set the coverage C and the number of windows N to generate
2:  for each images img do
3:  get the size of img [rows columns] = size(img)
4:  manipulate window width and height according to
    coverage C:
w_width = C * column;
w_height= C * rows;
5:  generate initial window from the initial pixel
point(s_x0, s_y0) of the image img s_x0=0 and s_y0=0;
    cropped_img=imcrop(img, [r_x, r_y, w_width,
    w_height]);
    save cropped_img (window) in output directory
6:  for each image img, to generate N sequence windows
    do
    ( for i=N;I>1;i--)
      generate sequential pixel point (s_x,s_y) s_x= (rows –
      w_height)/N-1;
      s_y= (columns – w_width)/N-1; crop the image img
      cropped_img=imcrop(img, [s_x, s_y, w_width,
      w_height]
      save cropped_img (window) at the output directory
  7:  end for
  8:  end for
```

VI. Clustering

As it is previously discussed descriptors of each image of the collection are needed in order to cluster. Therefore, first, it is started with the generation of descriptor of each image and then those are saved into a single array in order to cluster. The input will be image collection of the pre-cluster phase. Technically, the focus is set on the highly dissimilar representative images. For that local features of images are used. The local approach represents each image by a set of local featured descriptors computed at some interesting points inside the image [3]. For finding and computing descriptors of each images SIFT algorithm is used.

Now, K-means algorithm is applied on an array of descriptors of images. In statistics and data mining, K-means clustering is a method of cluster analysis which aims to partition n observations into *k* clusters in which each observation belongs to the cluster with the nearest mean [7]. Since the dataset is a large set (minimum 1000 images and maximum 5000 for *N*=5 windows) and the goal is to generate precise, minimal redundant and diverse informative overview of the image collection, it is decided to apply K- means twice. So, first, K-means with the *k* value is applied and a small subset of image set is generated then K-means is applied again in order to get reduced set of small set, which is more diverse .When K- means is applied, cluster results, centroids, sum and distances are achieved.

Now centroid image of all clusters are fetched which are the representative image of each clusters. The concept is, to find for each cluster the least distance image to the centroid. The distances are found from K-means output and after calculating the distances; we sort the nearest image of the cluster centroid for each cluster. That will be the representative set of the image collection.

After having 1st k-means subset, again K-means is applied second time on representative set which becomes precise and small representative set of the large image data set. So, from this phase we generate representative set and it is also useful for the next phase namely Ranking Mechanism. One can see the procedure for generating representative set in algorithm 3.

Algorithm 3: Clustering and generating representative set

```
Input: resultant image set of pre-cluster
Output: the representative set
1:  for each images img do
      get descriptor or key points by calling sift function
      [image, descriptor] =sift (img)
      save each image descriptor in an array
      descriptor_images[img] = descriptor
2: end for
3: set number of clusters k and apply k-means on the
     descriptor_images array
   [Id, C, D] = kmeans(descriptor_images, k)
          where Id is image identification number, C is the
          assigned cluster number and the D is distance from
          the assigned cluster and other clusters as well.
4: find centroid image of each clusters:
      for each images i and j of cluster C
      if distance_image_i < distance_image_j
              //store the least distance image
              centroid = image_i
      end if
      save centroid at the output directory of the
      representative set
5: end for
```

VII. Ranking Mechanism

Given the output from previous clustering step, it is desired to eliminate possible similar images in order to get more representative ones out of these inputs which compose the summarization. One other motivation behind this mechanism is to generate best set of images of the representative set and it helps the human based evaluation. In other words it would be hard to evaluate all images by participant. Instead of the full set we select most desirable set of images for example 10 images out of 20 images in order for the evaluation. Technically It goes like, on top of the function *Match.m* [2] some changes are made to fulfill the needs of the experiments of this paper and the number of matching points are then calculated. Besides, the number of key points of each image are achieved through *Sift.m*[2]. Then the ratio of number of matching points to summation number of detected key points among images is computed. A higher ratio indicates a larger possibility of similarity. Function *sum ()* is chosen here because the image sets used in this paper have the following features:

- They are supposed to be very dissimilar, which means the number of matching points, namely the numerator could be very small
- The number of key point's detected can varies a lot. for example, one is 100, the other one is 3000.

Based on these, the values are more like 10/(100+3000), (or 10/max(100,3000) if *max()* is used), where function *sum()* can represent the dissimilarity well. However, the function *max()* can be chosen since it also can do a good approximation. By sorting the ratios of all images, one is free to choose arbitrary number of images and these images are highly representative. The Ranking mechanism is presented in algorithm 4.

Algorithm 4: Ranking mechanism

```
Input: representative set
Output: ranking of each images of representative set
1:  for each image i do
      for each image j from i+1 to n do
       // function Keypoint return the number of detected
       keypoints of an image using SIFT
       //function Matchingpoints return the number of
       matched points between to images calculate
       similarity ratio r[i j]=
   Matchingpoints(i,j)/Sum(Keypoint(i)+Keypoint(j));
2:    end for
3:   end for
4: for i from 0 to n do
```

```
    for j from i+1 to n
do
        compute the maximum similarity value mr[i]=Max[i j];
    end for
end for
sort mr[i]; // in ascending order.
select arbitrary number of images from array s which will be the summary S;
```

VIII. Evaluation

A. Initial Experimental Setup

First the data set is defined. The data set is of 1000 holiday images (ordinary 2D high quality images) taken from INRIA holidays data-set[1]. The dataset only contains images without any tags. The dataset includes a very large variety of scene types (natural, man-made, water and fire effects, etc) and images are in high resolution. Using windows cropping techniques we increase image sets from 1000 to *N**1000 images. For example, if 3 windows are found per image then there will be 3000 images for the overall experiments. For the evaluation, 6 data set have been selected on the basis of overall time, coverage and number of windows and also for the overall comparison only the sift output is used. The values for coverage *C*, number of windows *N* for the experiments are set. The idea is to set coverage higher in order to lose less information. It is considered to have only *N*=3 and 5 windows and *C* =66%, 75% and 85%. Two result image sets (one for random and the other for sequential windows) are selected of each coverage *C* for the evaluation while taking care of different *N* values.

For the clustering phase, the input image set or windows set is obtained from the first phase. After many initial experiments k=90 is set for the first clustering. As it is mentioned earlier that k-means is applied two times in order to get reduced set and highly dissimilar images, after having 90 clusters again k-means is applied with the value of k=20. The idea is to keep same k, so that windows of the original image mostly go to the same cluster because of high coverage *C*. The outcome of this phase is the representative set which have 20 images of the original data set.

For the ranking mechanism, the input of 20 images are taken from the second clustering phase and then ranking mechanism is applied on it and it gives rank of all images. Now, for taking into consideration of human based evaluation, 3 different sets of images top 10, 15 and 20 images are selected and these becomes summarization of the original big set. By doing this, it is intended to see whether changing the number of images in the summary set does give different results.

A questionnaire [figure 6] is devised for human evaluation of the experiment outcomes from different methods and is spread to 24 interviewees. It is intended to see the rating of the users or human evaluator for each sets. It is also intended to see, changing the number of images inside the summary affects the result. For the experiment, 10, 15 and 20 images for the summary are chosen for the each set. So, human evaluator has to check total 18 sets (6 generated set * 3 different summary of each image set). Before starting questionnaire, the original image collection is shown and the research perspective of this paper is explained to the interviewees.

As 3 sets of each image set are obtained which contains 10, 15 and 20 images respectively, the fixed pattern for each set is followed which can be seen in figure 6.

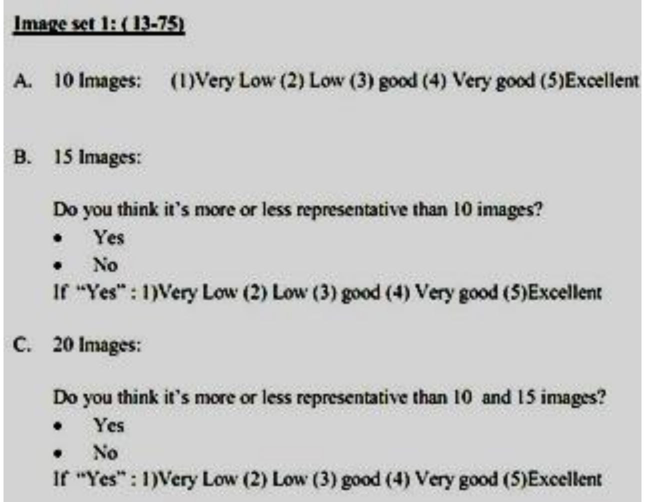

Image set 1: (I3-75)

A. 10 Images: (1)Very Low (2) Low (3) good (4) Very good (5)Excellent

B. 15 Images:

Do you think it's more or less representative than 10 images?

- Yes
- No

If "Yes" : 1)Very Low (2) Low (3) good (4) Very good (5)Excellent

C. 20 Images:

Do you think it's more or less representative than 10 and 15 images?

- Yes
- No

If "Yes" : 1)Very Low (2) Low (3) good (4) Very good (5)Excellent

Fig. 6. Questionnaire pattern for each result sets

To understand deeply, an example is explained here. Suppose one of the representative set called "I3-75" is under evaluation. As it is mentioned earlier that 20 images are considered in a representative set, first interviewees are requested to rate 10 top ranked images generated from ranking mechanism and then 15 top ranked images are shown and participants are asked whether they think that these 15 images are more or less representative than 10 images. If they say "yes", then they are requested for the rating again otherwise directly whole 20 images summary is shown and the same question is asked in perspective of 10 and 15 images summary. Again if they say "yes", then again they are asked to rate for the 20 images summary. The pattern of the questionnaire will be same for rest of the five result sets.

B. Initial Time Based Evaluation

From figure-7 it can be observed, there is nominal difference in the time taken by windows generating algorithm (blue bar) for *N*=3 and *N*=5. The time taken by clustering algorithm is higher for both *N*=5, *C*=85% random and sequence windows. The reason of this is to fetch sift descriptors of all the windows and then process for the clustering. While for the set *N*=3, *C*=66% sequence took very less time. There is no much difference in time taken by the ranking mechanism for all sets. The set *N*=5, *C*=75% sequence windows took long time for the ranking mechanism in compare to others. Maximum total time is consumed by *N*=5, *C*=85% random windows set. The time taken by clustering and fetching centroids increases with the increase of number of windows *N* and the coverage *C*.

[1] INRIA project web-link: http://lear.inrialpes.fr/~jegou/data.php.

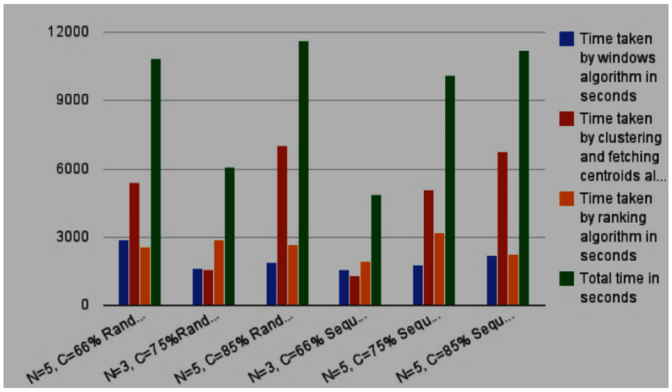

Fig. 7. Time consumed by different cropped window

C. Human Based Evaluation

In this section the human based evaluation results are presented with the statistical evaluation. It is started with analyzing the participants rating and then the number of images inside the summary. First, all different summaries individually and then the common observation are evaluated. Once again, it can be recalled that six different summaries with different number of images inside are evaluated, where, three random window results namely I5-66, I3-75 and I5-85 as well as three sequential window results namely S3-66, S5-75 and S5-85 are used.

Figure 8 shows the participants rating summary of 10 images. One can observe that the worst result with the highest votes is I5-66. Similarly for the bad, medium and good results with the highest votes are I5-85, I3-75 and S3-66 respectively. The most excellent result is S5-85.

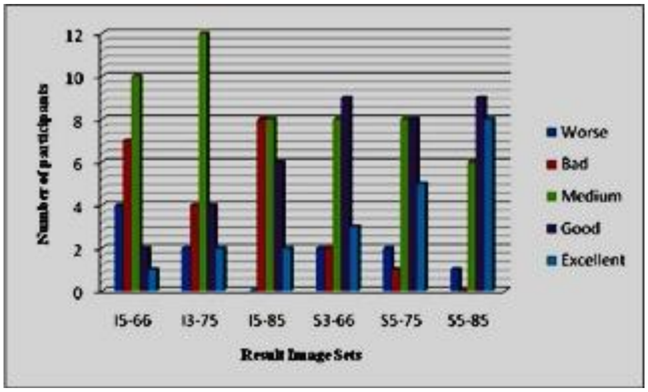

Fig. 8. Participant ratings for summary of 10-images

Figure 9 shows the participants rating summary of 15 images. One can observe that the worst result with the highest votes is I5-66. Similarly for the bad, medium and good results with the highest votes are I5-66, I3-75 and S5-75 respectively. The most excellent result is again S5-85 for the summary of 15 images.

Figure 10 shows the participants rating summary of 20 images. One can observe that the worst result with the highest votes is again I5-66 which similar to 10 and 15 images results. Similarly for the bad, I3-75 and S5-75 have the highest and same votes. The medium results with the highest and same votes are I5-66, I3-75 and S3-66. The good and excellent with the highest votes is S5-85.

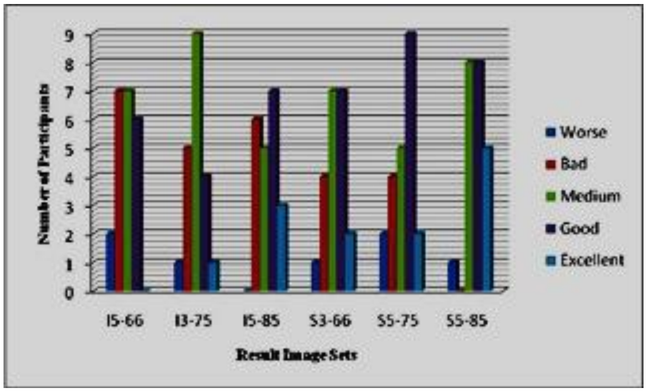

Fig. 9. Participant ratings for summary of 15-images

General observation about the summary of 10, 15 and 20 images are as follows:

- Random window result sets give negative rating with the high portion of votes as worse, bad and medium while sequential window result sets give positive rating with the high portion of the votes in medium, good and excellent.
- If random windows and sequential windows are focused differently, one can observe a common thing. When the coverage increases, the result sets with higher coverage secures good and positive ratings.

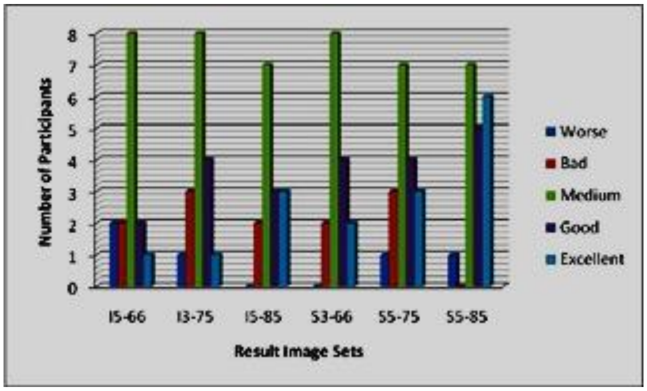

Fig. 10. Participant ratings for summary of 20-images

To understand these two general observations, scores for each summary is calculated. So, the calculation is based on the evaluators rating. For the worse, bad, medium, good and excellent rating, an integer number 1,2,3,4 and 5 is assigned respectively. Now, a formula is devised for calculating score of each result sets. The formula is defined as follows:

Total score = [((NE_worse *1) + (NE_bad *2)+ (NE_medium *3)+ (NE_good*4)+ (NE_excellent*5)) * 100] / Total NE

Here, NE: Number of Evaluators who voted.

Total score for each result set is a calculation for number of evaluators votes for each category multiply the assigned integer number and again multiply 100 and the value is divided by total number of evaluators voted. The reason of multiplying with 100 and division by total number of evaluators voted for the particular set is to achieve unique scoring pattern. As users are asked to look summaries with 15 or 20, they might be satisfied with summaries 10 or 15 images. So, the total number of evaluators votes of summaries 15 and 20 images could be less than the summary of 10 images.

Figure 11 shows the score of each result sets with different number of image. The points made in general observation can be easily observed with this graph. The total scores for each results set with different number of images (i.e. 10, 15 and 20) are more or less similar.

Another evaluation is based on the number of images inside the summary. Here, it is intended to know the evaluator's perception to see the summary with different number of images. As per all the completed experiments for each image set, initially, it is started with 10 images, then 15 images and finally, 20 images of the same representative set are shown.

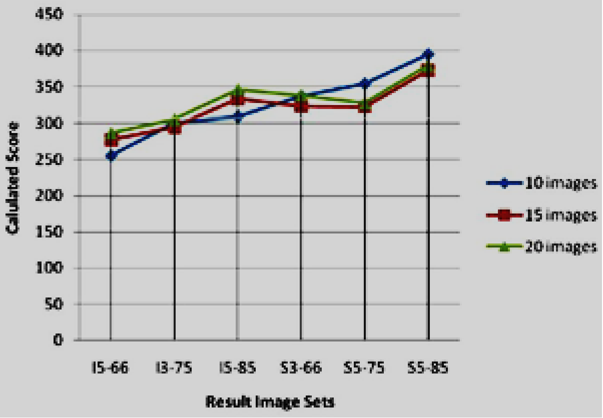

Fig. 11. Overall evaluation

The evaluation states that, after seeing summaries of 10 images, 89% of the evaluators were interested to rate the summary of 15 images. Figure 12 shows that the interest ratio of evaluators for the 15 images set. That means they would like to change the rating for the new 15 images summary of the same representative set. It can be either positive or negative. The result shows four results sets secure higher or equal total score than the 10 images, while two sets namely S5-75 and S5-85 give less score compared to 10 images.

Same for the next set of 20 images, there were 69% evaluator who would like to rate 20 images summaries. Figure 13 shows that the interest ratio of evaluators for the 20 images set. Evaluation shows that the summary with 20 images secures almost same result as 15 images. So, there is no much difference in the total score between the summary of 15 images and 20 images.

The interest level to see summary of 10 to 15 images and 15 to 20 images shows that the number of images inside the summary affects the result.

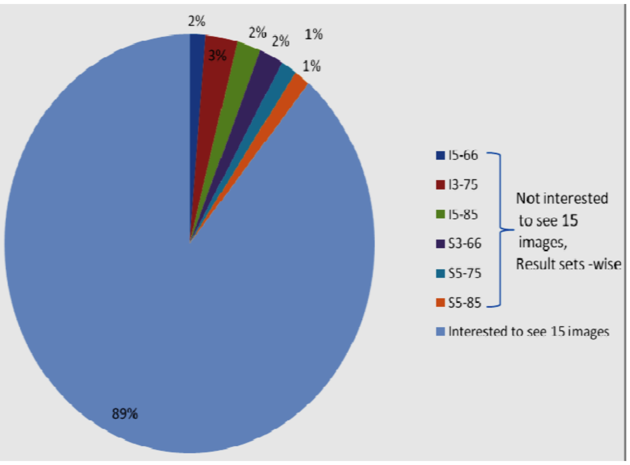

Fig. 12. Distribution of interest of participant to look summary of 15 images

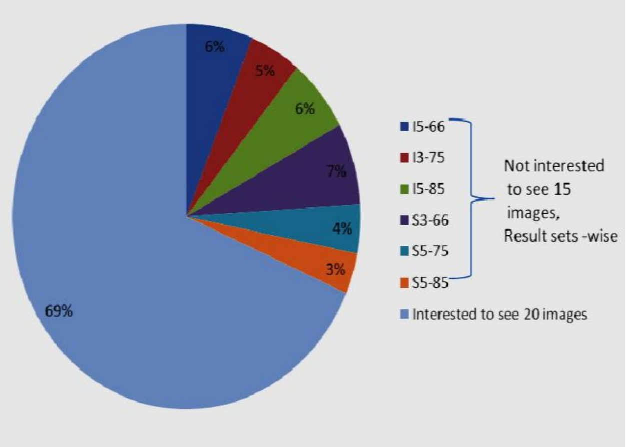

Fig. 13. Distribution of interest of participant to look summary of 20 images

IX. Conclusion and Future Work

The focus of this work is to generate the best representative set and summary of the large dataset by cropping images randomly and sequentially with different coverage. Though, the algorithm takes too much time for the overall computation, a good human based evaluation is found for sequential datasets rather than random datasets. It is observed that the higher coverage gives the best result regardless of sequential or random windows. In addition, it is also found that the number of images inside a summary varies the results.

For the future work, a further analysis on more values of number of windows *N* and coverage *C* can be done. The work can also be extended for finding out the faster algorithm to deal with the larger dataset. It is also planned to see the number of images inside the summary aspect in order to have a more comprehensive conclusion.

References

[1] Cavnar, William B., and John M. Trenkle. "N-gram-based text categorization." Ann Arbor MI 48113.2 (1994): 161-175..

[2] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision,60(2):91-110,2004.

[3] Lowe, David G. (1999). "Object recognition from local scale-invariant features". Proceedings of the International Conference on Computer Vision. 2. pp. 1150–1157.doi:10.1109/ICCV.1999.790410.

[4] Mor Naaman, Yee Jiun Song, Andreas Paepcke, and Hector Garcia-Molina. Automatic organization for digital photographs with geographic coordinates. In Proceedings of the Fourth ACM/IEEE-CS Joint Conference on Digital Libraries, 2004.

[5] I. Simon, N. Snavely, and S. M. Seitz. Scene summarization for online image collections. In ICCV '07: Proceedings of The 11th IEEE international Conference on Computer Vision. IEEE, 2007.

[6] A. Jaffe, M. Naaman, T. Tassa, and M. Davis. Generating summaries for large collections of geo-referenced photographs. Proc. Int. Conf. on World Wide Web, pages 853–854, 2006.

[7] Lyndon S. Kennedy and Mor Naaman. Generating Diverse and Representative Image Search Results for Landmarks. Proceedings of the 17th international conference on World Wide Web, NY,USA, 2008.

[8] T. L. Berg and D. A. Forsyth. Automatic ranking of iconic images. Technical report, U.C. Berkeley, January 2007.

[9] Pigeau and M. Gelgon. Organizing a personal image collection with statistical model-based ICL clustering on spatio-temporal camera phone meta-data. Journal of Visual Communication and Image Representation, 15(3):425–445, September 2004.

[10] Matthew Cooper, Jonathan Foote, Andreas Girgensohn, and Lynn Wilcox. Temporal event clustering for digital photo collections. In Proce of the eleventh ACM international conference on Multimedia, pages 364– 373. ACM Press, 2003.

[11] Adrian Graham, Hector Garcia-Molina, Andreas Paepcke, and Terry Winograd. Time as essence for photo browsing through personal digital libraries. In Proceedings of the Second ACM/IEEE-CS Joint Conference on Digital Libraries, 2002.

[12] Benjamin S. Braun. BioTechniques, Vol 59, No. 2, August 2015, pp. 82-86.