RMIT at TRECVid 2012: Instance Search

Amir H. Rouhi
RMIT/CSIT
Melbourne/Australia
amir.rouhi@rmit.edu.au

James A. Thom
RMIT/CSIT
Melbourne/Australia
james.thom@rmit.edu.au

Abstract

The paper introduces the procedure used by RMIT for Instance Search (INS) task 2012.

1-The submitted runs used Open-SURF Interest-Points but the runs were incomplete. The submitted runs were:

- RMIT-SURF-1: Based on the maximum score of the matching frames with any instances (filtered on top three matching frames).
- RMIT-SURF-2: Based on the average score of the top three matching frames with any instances.
- RMIT-SURF-3: Based on the average score of the top three matching frames with an instance (contained redundant answers).

We subsequently generate another complete run based on RMIT-SURF-1 over the whole dataset.

2-All runs performed poorly however RMIT-SURF-1 was slightly better at precision at n shots (for $n \leq 30$)

3-It is important to consider how raw scores from Interest-Point of different instances are combined.

4-Selecting maximum score of the top three matches between frames and instances seems to be more effective than averaging the top three scores.

1. Introduction

Instance Search (INS) is a pilot task introduced in TRECVid 2010 and continued in the following years. The task is designed based on some given visual examples. The visual examples are in three major categories: Person, Location and Objects. Each category contains several instance images. Each instance or visual example is represented by a pair of images:

1- The image of the instance of entity (Figure 1-a).
2- The mask image (Figure 1-b).

As the INS is still a pilot task in TRECVid, only detecting instances among short video clips is requested. There is no need to determine exactly in which frames or which time period of the dataset videos, the instance is detected. Meanwhile the entities are introduced by limited number of instance images, 5 or so, with their appropriate masks. The INS 2012 test data including 21 groups of entities as the visual query images and 4958 short video clips as the referenced dataset.

INS task has some specifications that make it different from other image/video searching tasks such as similarity detection or copy finding. As an instance is a portion of the whole image, in most of the cases, using global descriptors would not be a suitable choice in this task. Our literature study over the past participants shows that their approaches are mostly based on local feature detectors and descriptors such as SIFT (Scale Invariance Feature Transform) and SURF [3] [4] [5] [6] [7]. In this experiment we employ SURF interest points of the query images and dataset frames. We employed the built in library matching function of SURF as the base of the similarity measurement. The matching function itself is based on the count of the matching Interest-Points of the query image and dataset video frame. The SURF detector used in our experiment is Open-SURF 2008 [2].

2. Approach Overview

For explaining the details of the current experiment it would be better to categorize the process in the following three sections.

2.1. Query image pre-processing

In INS 2012 the instances were categorized in 21 groups and each group contains about 5 or so images and their related masks. The masks were designed in a way that by overlapping with the instance image, only the instance of the entity will be highlighted and all the other image information will be neglected. The important point is that the instance is not necessarily in a rectangular shape. In our experiment we noticed that there is difference on the SURF results when we consider only the instance extracted by the mask with the instance in a limited rectangle form. When we select the instance pixels only (Figure 1-c), the number of Interest-Points detected by SURF is less than the number of features when we select the small rectangular image that contains the instance and its peripheral area pixels (Figure 1-d). The reason is that some interest points will be detected when the pixels gradient varies in between the instance pixels and its surrounding area pixels. With respect to this fact, we reduce the size of the instance image in a way that contains the instance and a minimum space of its

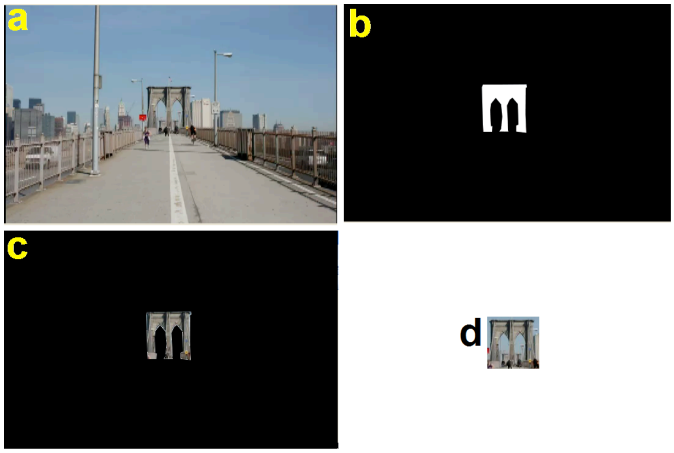

Figure 1: Extracting instance before applying SURF.
a-Main instance image. b-Instance image mask. c-Overlapping mask and instance image. d-Small size of the instance image containing the instance and its surrounding area pixels.

surrounding areas (Figure 1-d). This process is shown in the Figure 1. We applied this process for all the instances of the query images. As the total number of instances in the 21 groups with 5 or so images in each group is not a huge number, this process was performed manually.

2.2. Video Frame Selection and Feature Extraction

The first phase is frame selection. We simply used FFMPEG software to extract the frames from the dataset video clips in 1 second intervals. The size of the extracted frames remained intact. The format of the extracted frames was in BMP format. In the feature extracting phase we focused on the local features. We select the image Interest-Points produced by Speeded Up Robust Features (SURF). The reason for selecting SURF is mostly based on SURF efficiency due to low dimensionality of its descriptor vector. This specification makes the SURF feature size smaller than the Scale Invariant Feature Transform (SIFT) [1]. Moreover, using hessian matrices in the matching phase, results in the speeded up matching of SURF [1]. The SURF library used in this experiment is Open-SURF [2]. For using Open-SURF we need to set 5 parameters extra to the input image as the main parameter. These 5 parameters affect the final Interest-Points values. The first parameter is a boolean variable (upright) which determines the rotation invariance of the SURF features. In our experiment we set this parameter to False which means SURF is running in rotation invariant mode. The other four parameters and their values in the current experiments are:

- Number of Octaves: 4
- Number of Intervals: 4
- Initial Sampling Steps: 2
- Blob Response Threshold: 0.0001

The reason for selecting the above parameters is mostly based on visual results of some entity instances against the source instance image after some morphological transforms like rotation and stretching. Blob response threshold is very important in the number of detected Interest-Points. Figure 2 depict the Interest-Points in form of lines between instance image on the right and the main image on the left for three different instance rotations. Increasing the value of the Blob Response Threshold, will result in reducing the number of the Interest-Points (lines in Figure 2) which is not desirable. On the other hand, reducing the value of this parameter will result in error in Interest-Points.

In our submitted runs we did not succeed in performing all of the searching process. But in a later run, we completed the instance search and evaluated the results using trec_eval_video software.

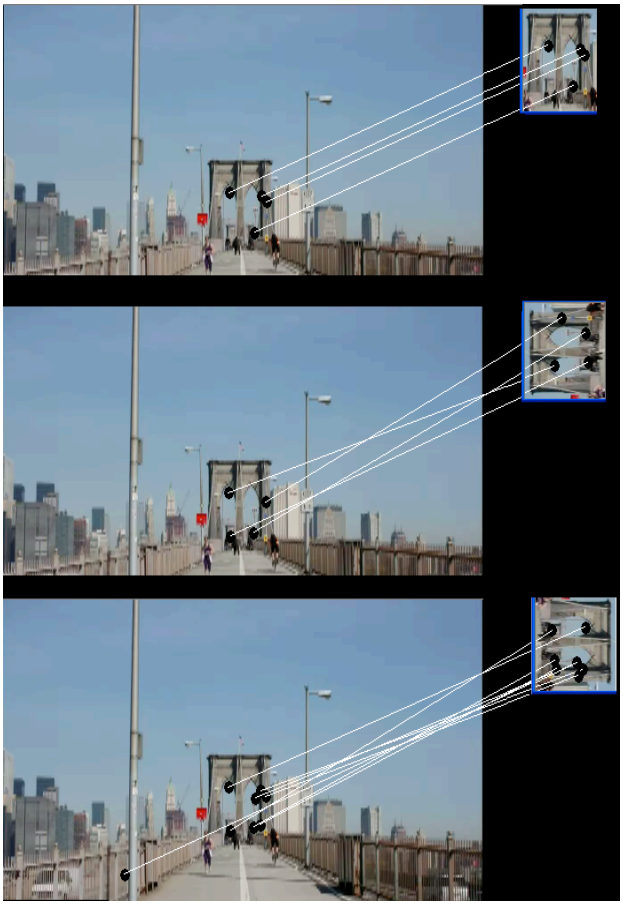

Figure 2: Matching SURF Interest-Points of a sample instance on the right and the main frame image on the left. The matching process is depicted in three different rotations of the sample instance.

2.3. Instance Feature extraction and Feature Matching

For feature extraction of the instance query images, we employed the Open-SURF with the same values of

the parameters used in the dataset video frames. The only difference is that this phase was performed only in real-time mode. In the matching phase we used the standard built-in matching function provided by the Open-SURF library. This function has an option for generating visual matching output. Figure 2 demonstrates the visual results of the matching function of Open-SURF. The number of matching points is depicted in the form of lines between the main image and the instance image.

3. Results

In the submitted runs we could not complete the experiments. Table 1 shows the percentage of completed INS search for each entity group in the submitted runs. The overall average of all completed search is 37.18%.

The results of the uncompleted experiment were submitted with three different sets of results. In the three sets we select the top three maximum matching numbers of the instance query image and each video clip in the dataset. The difference between the first and the second run is in the approach for calculating the score among those three top results for each video clip. The third run was exactly same as the second run but included the redundant video clips in the result set.

3.1. First Run

The first run was submitted as RMIT-SURF-1. In the first run we remove the detected clips if the variance of the top three matching results were higher than a threshold value. Blob detection is very error prone if the size of blobs is very tiny. But on the other hand, tiny values for blob threshold offer more accurate and number of Interest-Points. The error emerged in some cases and we noticed that the difference between the first and the last of the top three results is big in those cases. For resolving this problem we should define a dynamic threshold values for such cases. This value is calculated based on the Formula 1. We apply the formula on the top three result of the each instance search result for each video clip.

$$\text{MaxScore} - \text{MinScore} \leq (\text{MaxScore} / 2) \qquad (1)$$

If the condition of Formula 1 is true then the detected video clip is accepted results for the instance query image otherwise the clip is removed from the result set. The score of the video clip would be the max score among top three in this case. As each entity has more than one instance, we finally sort the scores of selected video clips of all instances in the entity and select top n items.

3.2. Second Run

The second run was submitted as RMIT-SURF-2. In the second run we did not apply the Formula 1 on the top three results of video clips. The final score of each video clip is calculated based on the average of the top three results. We finally sort the scores of video clips for all the instances in the entity and select the top n items.

3.3. Third Run

The third run was submitted as RMIT-SURF-3. The method used for arranging the result set of data was exactly same as the second run. The only difference was in having redundant video clips id in the final result. As there exists five or so number of instances in each entity, so it is natural for the algorithm that retrieve same videos for instances in the entity set. In the first two runs we had selected the max score among the redundant video clips. In the third run we did not. But after submission we were informed that redundant video clip id's are not allowed and the TRECVid staff, removed the redundant values in a pre-processing phase which performed by themselves. They told us they just retained the records with higher scores.

The summary of the results of these three runs can be seen in the Table 2, Table 3 and Table 4.

3.4. Complete Run

As the submitted three runs were not completed over the entire data set, we subsequently completed the experiments for this run. We evaluated the results of this run ourselves by trec_eval_video software and reference qrel file of INS task 2012. The approach used in the complete run is similar to the method in the first run. Table 5 shows the results of this run. The result does not show significant improvement in the effectiveness of the results in spite of running over all the data set. Even it shows worse results in some cases which can be interpreted as high false positive rate. Finally we can conclude that using Open-SURF with the parameter set we selected for this experiment, would not be an efficient and effective tool. Some morphological preprocessing phase over instance image and frame images is needed. Segmentation can be suggested as a good pre processing phase that can improve the effectiveness of the SURF result as well as the efficiency due to reducing the dimensionality of colour variety. This hypothesis needs to be tested.

Entity Group	Number of completed search	%Complete
9048	37560	50.11%
9049	28264	37.71%
9050	43144	57.56%
9051	36218	48.32%
9052	43956	58.64%
9053	28001	37.36%
9054	28569	38.11%
9055	23542	31.41%
9056	11874	15.84%
9057	45140	60.22%
9058	3906	5.21%
9059	16419	21.90%
9060	25992	34.68%
9061	42328	56.47%
9062	18963	25.30%
9063	12239	16.33%
9064	43044	57.42%
9065	15759	21.02%
9066	23828	31.79%
9067	35036	46.74%
9068	21511	28.70%

Table 1: percentage of completed INS search in the first experiment. Overall average is 37.18%.

First Run				
Interpolated R/P			Precision at n shots	
Recall	Precision		n	Precision
0.0	0.01999		5	0.0100
0.1	0.0000		10	0.0050
0.2	0.0000		15	0.0033
0.3	0.0000		20	0.0025
0.4	0.0000		30	0.0017
0.5	0.0000		100	0.0020
0.6	0.0000		200	0.0017
0.7	0.0000		500	0.0007
0.8	0.0000		1000	0.0004
0.9	0.0000			
1.0	0.0000			

Table 2: Results of the first run.

Second Run				
Interpolated R/P			Precision at n shots	
Recall	Precision		n	Precision
0.0	0.0120		5	0.0000
0.1	0.0069		10	0.0000
0.2	0.0000		15	0.0000
0.3	0.0000		20	0.0000
0.4	0.0000		30	0.0000
0.5	0.0000		100	0.0029
0.6	0.0000		200	0.0033
0.7	0.0000		500	0.0064
0.8	0.0000		1000	0.0032
0.9	0.0000			
1.0	0.0000			

Table 3: Results of the second run.

Third Run				
Interpolated R/P			Precision at n shots	
Recall	Precision		n	Precision
0.0	0.0042		5	0.0000
0.1	0.0069		10	0.0000
0.2	0.0000		15	0.0000
0.3	0.0000		20	0.0024
0.4	0.0000		30	0.0016
0.5	0.0000		100	0.0014
0.6	0.0000		200	0.0014
0.7	0.0000		500	0.0010
0.8	0.0000		1000	0.0005
0.9	0.0000			
1.0	0.0000			

Table 4: Results of the third run.

Complete Run				
Interpolated R/P			Precision at n shots	
Recall	Precision		n	Precision
0.0	0.0101		5	0.0000
0.1	0.0086		10	0.0000
0.2	0.0065		15	0.0000
0.3	0.0048		20	0.0000
0.4	0.0000		30	0.0000
0.5	0.0000		100	0.0000
0.6	0.0000		200	0.0015
0.7	0.0000		500	0.0037
0.8	0.0000		1000	0.0066
0.9	0.0000			
1.0	0.0000			

Table 5: Results of the complete run.

References

[1] Bay, H. and Ess, A. and Tuytelaars, T. and Van Gool, L. Speeded-Up Robust Features(SURF), *Computer vision and image understanding*, 110(3):346--359, 2008.

[2] Evans, C. Notes on the Open-SURF library. *University of Bristol*, CSTR-09-001, 2009.

[3] Delezoide, B. and Precioso, F. IRIM at TRECVID 2011: Semantic indexing and instance search. *Proceedings of TRECVID 2011.*

[4] Bailer, W. and Sorschag, R. and Lee, F. and Stiegler, H. and Schwendt, G. Joanneum Research and Vienna University of Technology at TRECVID 2011: Semantic Indexing and Instance Search. *Proceedings of TRECVID 2011.*

[5] Bailer, W. and Lee, F. and Stiegler, H. and Sorschag, R. Joanneum Research and Vienna University of Technology at TRECVID 2010. *Proceedings of TRECVID 2010.*

[6] Zhu, C.Z. and Satoh, S. Large Vocabulary Quantization for Instance Search at TRECVID 2011. *Proceedings of TRECVID 2011.*

[7] Shishibori, M. and Ohnishi, M. and Tanioka, Y. and Kita, K. Instance Search and Content-based Copy Detection Experiments for TRECVID 2011. *Proceedings of TRECVID 2011.*