Object Retrieval in Past Video Using Bag-of-Words Model

Quoc-Huu Che
HCMC University of Science
Advanced Program
in Computer Science
Email: cqhuu@apcs.vn

Tu-Khiem Le
HCMC University of Science
Advanced Program
in Computer Science
Email: ltkhiem@apcs.vn

Manh-Tien Nguyen-Hoang
HCMC University of Science
Advanced Program
in Computer Science
Email: nhmtien@apcs.vn

Van-Tu Ninh
HCMC University of Science
Advanced Program
in Computer Science
Email: nvtu@apcs.vn

***Abstract*—With the advanced development of technology, Computer Vision plays an important role in enhancing machines to overcome human limitations. One of the limitations is human's memorization ability, especially memorizing things such as personal items. It is annoying to waste time finding lost things manually by recall or notes. For those reasons, the authors propose a novel method using image annotation and retrieval to find things. Its aim is to assign keywords to images and retrieve information through videos by images. The authors apply Bag-of-Words model which is one of the most famous methods in image annotation and retrieval. Through experiments, the results give that the accuracy of authors' system is 55.43%. This novel method can open new horizon in robotics to help improving human's life.**

I. Introduction

In daily life, finding lost things is not a pleasant experience. A great number of people agree that finding lost objects is a daily life problem [1]. According to Rodney E. Peters, Richard Pak, Gregory D. Abowd, Arthur D. Fisk and Wendy A. Rogers [1], 43% young adults, 38% middle-aged adults and 22% older adults agreed the statement "Losing objects is a recurring problem for me" [1]. In terms of time spent on finding objects, 21% young adults, 45% middle-aged adults, 23% older adults admit that it is a burden of their lives [1]. In the top 10 everyday things we often forget, essentials are highly rated [2]. Essentials are necessary personal items which are daily used such as keys, phone, purse and wallet [2]. 37% of Brits admit that they often spend an average of 15 minutes per day finding these things [3].

Due to those numerical data, we can consider finding lost object as a big problem we need to resolve. This problem can take advantage of advanced algorithms in Computer Vision. Thus, the authors propose a novel method using image annotation and retrieval to find objects with the accuracy of 55.43% and precision of 49%. In general, this method decreases time spent on finding things and increases the accuracy of objects retrieval through images. It finds the object by images extracted from an automatically recorded video of the place we want to find that object.

Implementing Bag-of-Words model [4] consists of 4 main steps: 1) Local Feature Extraction [5], 2) Codebook Generation [6] [7], 3) Vector Quantization, 4) Use different distance metrics to find proper result.
There are extra modifications to enhance the accuracy of algorithm and reduce the computational cost in step 2 and 3. Step 4 might use different distance metrics such as symmetric and asymmetric distance interchangeably to determine which metric gives better result.

In our system, user first records a video of the place he/she wants to find objects. The system gets features of images extracted from the video using VL_SIFT library [8]. Quantization process is executed to create Bag-of-Words histogram. The result histogram is used to find images which have similar appearances of objects through visual instance search [9] based on different distance metrics. The authors' contributions include:

- We propose a method of implementing Bag-of-Words model with detailed numerical parameters to apply to finding object by image retrieval.
- We propose using L2 Norm to normalize the histogram. This proposal contributes to increase the accuracy of result up to 55.43%.

The authors conduct experiments on 2 testing videos of length 12.38 minutes and 2.08 minutes around a house. By querying 245 images of objects, We determine a way to implement Bag-of-Words model to find objects with the accuracy of 55.43% and the precision of 49%.

The authors divide the content of this paper as follows. In section 2, the authors review some background and related work. Section 3 is the proposed method of our research. The experiment results, evaluation and interpretation are found in Section 4. The conclusion of the research and future work are presented in Section 5.

II. Background and Related Work

A. Background

Image annotation is the process of assigning text keywords to the image which represents fully its visual content [10]. The main method of image annotation that the authors use is assigning a histogram to each image. The image retrieval process to find objects is executed and evaluated based on the histogram.

Our proposed method uses the Bag-of-Words model, which is well-known in document classification and object categorization [11]. In Computer Vision, Bag-of-Words model transforms information of an image into a single histogram based on the extracted features [12]. The algorithm uses features of the image to create words. In detail, the words can be generated by clustering the features of all images in the dataset. The histogram of an image represents frequency of the words by counting number of features of the image which is most likely to every single word. By using this model, comparison of two images is equivalent to comparison of two histograms created from the data.

Feature Extraction is an important step for further work. The better features the detector extracts, the more precise result we receive in later steps. In this process, features are extracted and demonstrated by 128-dimensional vector [8]. There are many methods, detectors and filters which can be used to extract features from an image. Details are mentioned in section III.C.

B. Related Works

Through time, people realize the drawbacks of Bag-of-Words model and try to improve it. Faster implementation and accepted accuracy of the model are needed to apply in many real problems. Some drawbacks are improved and mentioned below.

In Bag-of-Words model, codebook generating and vector quantization take a great amount of time to execute. Therefore, choosing appropriate method which is fast and accurate is very important. We sometimes even have to make a trade-off. A survey of different methods implemented in process of Bag-of-Words model is proposed to compare and give an overview of Bag-of-Words process [13]. Bag of Words model still have problem when processing with a large collection of images. Indexing quickly and accurately in large data plays an important role in using Bag of Words model. Comparison of different methods to implement Bag of Words model is proposed and experimented by Mohamed Aly [14].

In view of application, Bag-of-Words model is also applied to experiment on scene classification by evaluating the Bag-of-Words representations [15]. They are used to compare to find which category a scene belongs to.

Finding lost objects using Bag-of-Words model is still a new approach in software application. This problem is resolved by using chips and tracking devices. In detail, with the rapid development of technologies, many methods using hardware to track and find objects are highly recommended. One of the methods is using Radio-frequency identification (RFID) to keep track and find objects' position [16] [17]. Another method is using the robot interacting with Web to get background knowledge about objects' environment [18]. However, the drawback of those methods is that we need to use many expensive devices such as electronic chips [16], Ultrasonic position detection system [17], robot connected to the Internet and so on [18].

III. Method

A. Overview

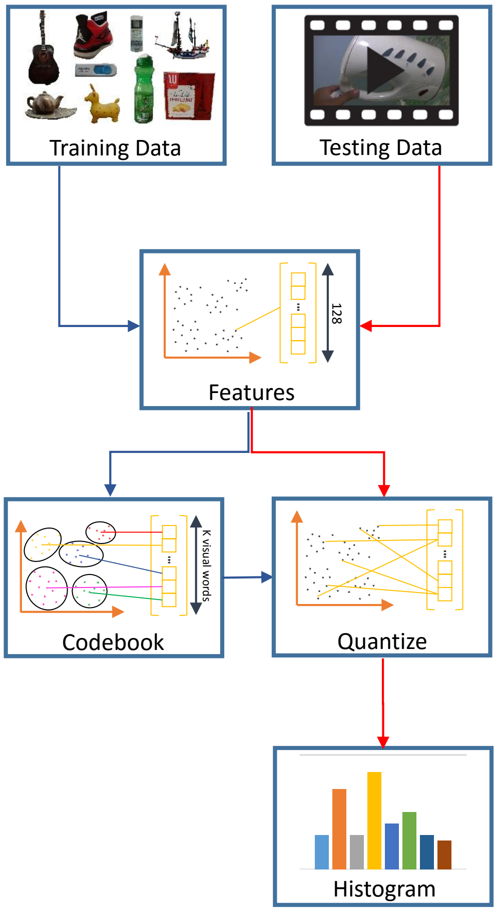

Fig. 1. Overview of our proposed system applying Bag of Words model to find objects

Our goal is to solve the following problem: given an object and a video, find all time intervals in the video which that object appears. The image containing the object may contain many outliers that can reduce accuracy of the result. Thus, getting data may have many restrictions that make it become a big issue in our experiment, more details are given in subsection III.B.

In this part, the authors introduce a method which takes advantage of Bag-of-Words model. There are 4 main processes: data initialization, data analysis, database organization and data relationship model function.

The first process is mainly building up two datasets: training data and testing data. Training data are the input objects and testing data are videos. The second process describes how data is transformed into the model. In the third process, the histogram of each image is created. The fourth process is applying many comparison functions between model vectors of both data in training data and testing data. After the 4 main processes are completed, we receive the result and print the answer. Overview of our proposed system is demonstrated in Figure 1.

B. Data Initialization

1) Training Data: We create training data by capturing images of an object in many different views. In this process, the background of images is not synchronized, which may contain noises. Noises are features which are not relative to the ones in main objects. Therefore, the authors propose changing the backgrounds color of the images to white. The authors highly propose eliminating the background of the images in training data manually by tools. This proposal reduces the noises of the features in the image after extracting and increases the accuracy of calculation in other steps.

2) Testing Data: Testing data is a video. There are many restrictions of collecting data. Any object in both training data and testing data must occupied approximately greater than $\frac{1}{3}$ of the image and must not be blurred. The background behind the object may be smooth. In this process, objects must be recorded at least 2 to 5 seconds and at most 10 to 12 seconds for frame extracting convenience. The authors convert the video to mp4 format (25 frames per second). The authors choose 3 representative frames in each second.

C. Data analysis

There are two important steps in this section:

1) Features Extraction: There are many methods to extract features of an image such as Speed Up Robust Features (SURF) [19], Haar wavelets, color histogram, Local Binary Patterns (LBP), Histogram of Oriented Gradients (HOG) [20], Hessian Affine Detector [21] and Difference of Gaussian (DoG) [8]. The authors choose Hessian Affine Detector to extract SIFT features from all images in training data and testing data through VL_SIFT library [8].

Figure 2 illustrates the result of features extraction using the Hessian Affine Detector. Each green circle represents for a feature. The feature can also be formed into a 128-dimensional vector for further computing.

2) Codebook Generation: Due to the noises of the background, the extracted features of query data (taken from the train data) cannot be directly compared with the ones of testing data to obtain the result. To solve this problem, the authors convert the features to visual words. In this process, only the features in the training data are clustered into k visual words (k-clusters). The features which are totally different from each other are in different cluster. We use k-means algorithm to execute this process. The biggest problem is that the amount of time spent on clustering a large number of features is huge. To reduce the execution time, we apply approximate k-means algorithm (AKM) proposed by James Philbin [6]. The approach of the proposed algorithm uses data structure kd-tree to reduce the complexity of the algorithm from $o(nk)$ to $o(n\log(k))$.

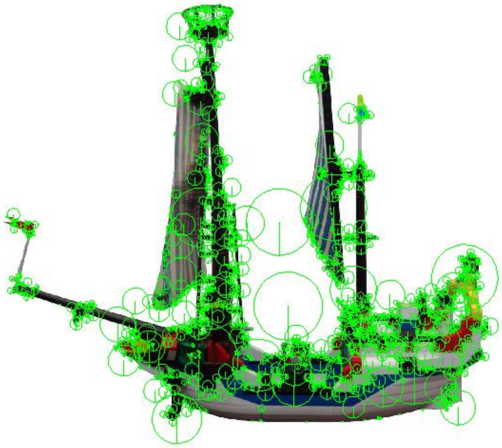

Fig. 2. An image's features extracted by SIFT

D. Database Organization

For every image in both data, the extracted features in the image can be organized as a vector. The idea is that those features can be considered as words. Feature and word are both represented as 128-dimensional vector. A feature is considered belonging to a visual word if their vectors have minimum Euclidean metric:

$$d_{i,j} = \sqrt{\sum_{k=1}^{128} (x_{i_k} - x_{j_k})^2}$$

$$f(A \in D) = \{i \mid (i \in K) \cap \min(\forall j \in K : d_{j,A}) = d_{i,A}\}$$

By that way, we can get information about the number of features in a visual word. Each image is represented as a histogram of k visual words. This process is also called quantization. The complexity of quantization step is $\mathcal{O}(nk)$, which n is the number of descriptors in an image and k is the number of clusters. Hence, the authors propose to use Fast Approximate Nearest Neighbors to improve the performance and execution time [22].

1) TF-IDF: TF-IDF is an abbreviation of Term Frequency - Inverse Document Frequency. TF-IDF weight is widely used in information retrieval and text mining. This weight is used to evaluate how important a word is in a document. TF-IDF can eliminate stop-word in a collection of text successfully. This leads us to an observation: The higher frequency a visual word has, the less important it is. Thus, a visual word with lower frequency can be more distinctive. Hence, the authors apply TF-IDF to give scores to the histogram as follows:

- TF: Term Frequency

$$tf(t,d) = \frac{f(t,d)}{\max\{f(w,d) : w \in d\}}$$

- $f(t,d)$ - frequency of visual word t in histogram d.
- $\max\{f(w,d) : w \in d\}$ - maximum frequency of an arbitrary visual word in histogram.

- IDF: Inverse Document Frequency

$$idf(t,D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

- $|D|$: the total number of histograms in the dataset
- $|\{d \in D : t \in d\} + 1|$: the number of histograms where word t appears.

- TF-IDF Weight:

$$tfidf(t,d,D) = tf(t,d) \times idf(t,D)$$

2) Normalize Vector: To increase the accuracy of our algorithm for images with different scale, the authors normalize the histograms. The new frequency of the histogram is calculated by one of the two formulas as follows:

- L1 Norm: $||x||_1 = \sum\limits_{i=1}^{n} |x_i|$
- L2 Norm: $||x||_2 = \left(\sum\limits_{i=1}^{n} |x_i|^2\right)^{\frac{1}{2}}$

E. Data relationship model function

When we compare two histograms of bag-of-words representation to find the result, it is important to choose suitable distance metric to give robust result. Normally, people often use L2 distance (Euclidean metric) to compare two histograms. But in some cases, L1 distance may give better result.

1) Definition:

- Metrics: A distance dist: $D \times D \rightarrow \mathbb{R}^+$ is a bivariate operator $(a \in D, b \in D)$ that maps to $\mathbb{R}^+ = [0, \infty)$. It is a metric if
    1) $d(a,b) \geq 0$ with equality if and only if $a = b$
    2) $d(a,b) = d(b,a)$
    3) $d(a,b) \leq d(a,c) + d(c,b) \quad (c \in D)$

  A distance that satisfies 1), 2) but not 3) is called quasimetric. In our problem, we consider the symmetric attribute significantly because of the asymmetric attribute.

2) Data Relationship Formulas:

- Symmetric Distance: There are a lot of symmetric distance formulas. The most well known of this distance metrics is Euclidean distance. Euclidean distance is the representative metric of symmetric distance used in our experiment. With n-dimensional vector $\boldsymbol{a}$ and $\boldsymbol{b}$, Euclidean distance of $\boldsymbol{a}$ and $\boldsymbol{b}$:

$$d_2(\boldsymbol{a}, \boldsymbol{b}) = ||\boldsymbol{a} - \boldsymbol{b}||_2$$

- Asymmetric Distance: Object in the query image can appear in database images, not in reverse order. Some objects in database image may not appear in any query image. It is why asymmetric distance is more suitable than symmetric distance in our problem. It is also the essential idea in asymmetric distance.

  Assume that we have a k-dimensional histogram vector of a query image and the i-th database image: $\boldsymbol{q}$ and $\boldsymbol{r_i}$. The asymmetric dissimilarity of the two images is computed by the following formula:

$$d_p(\boldsymbol{q}, \boldsymbol{r_i}, w) = w\,||\boldsymbol{q} - \min(\boldsymbol{q}, \boldsymbol{r_i})||_p + ||\boldsymbol{r_i} - \min(\boldsymbol{q}, \boldsymbol{r_i})||_p$$

  In this formula, $\min(\boldsymbol{q}, \boldsymbol{r_i})$ is the maximum number of matching pairs of each component j $(j = 1...k)$ of $\boldsymbol{q}$ and $\boldsymbol{r_i}$, w is the weight of asymmetry in the problem. In our problem, we use distance with parameter $p = 2$. Hence, the formula of asymmetric dissimilarity becomes:

$$d_2(\boldsymbol{q}, \boldsymbol{r_i}, w) = w\,||\boldsymbol{q} - \min(\boldsymbol{q}, \boldsymbol{r_i})||_2 + ||\boldsymbol{r_i} - \min(\boldsymbol{q}, \boldsymbol{r_i})||_2$$

$$= w\left(||\boldsymbol{q}||_2^2 - 2\boldsymbol{q}\min(\boldsymbol{q}, \boldsymbol{r_i}) + ||\min(\boldsymbol{q}, \boldsymbol{r_i})||_2^2\right)^{\frac{1}{2}}$$

$$+ \left(||\boldsymbol{r_i}||_2^2 - 2\boldsymbol{r_i}\min(\boldsymbol{q}, \boldsymbol{r_i}) + ||\min(\boldsymbol{q}, \boldsymbol{r_i})||_2^2\right)^{\frac{1}{2}}$$

  We can fix the value of w in the formula [23]. It can also be created by setting a parameter α. The formula to compute w in this situation as follows [23]:

$$w = \alpha . \frac{\sum\limits_{i=1}^{n} \sqrt{||\boldsymbol{r_i}||_2^2 - 2\boldsymbol{r_i}\min(\boldsymbol{q}, \boldsymbol{r_i}) + ||\min(\boldsymbol{q}, \boldsymbol{r_i})||_2^2}}{\sum\limits_{i=1}^{n} \sqrt{||\boldsymbol{q}||_2^2 - 2\boldsymbol{q}\min(\boldsymbol{r_i}, \boldsymbol{q}) + ||\min(\boldsymbol{r_i}, \boldsymbol{q})||_2^2}}$$

Overview of how data is created for comparing to output result is shown in figure 3.

IV. Experiment and Result

A. Experiment

The authors conduct experiments to determine the accuracy of Bag of Words model applied to find objects in frames extracted from a video. We test the method in 5 different aspects of implementing Bag of Words model which includes:

1) The detector used in feature extraction steps.
2) The number of clusters affecting the accuracy of Bag of Words application in finding objects.
3) Term Frequency - Inverse Document Frequency (TF-IDF)
4) Data normalization of L1 Norm and L2 Norm
5) Distance metrics of Symmetric and Asymmetric

Due to 5 different aspects of Bag of Words model, there are 5 experiments that we need to conduct. Details of the 5 experiments are described below:

1) **Experiment on the quality of detectors used in features extraction**.

   The authors compare the quality of detectors based on the output of the Bag of Words model applied to finding objects. The main goal is to determine the best detector to extract features efficiently.

2) **Experiment on the quality of 2 normalization methods**.

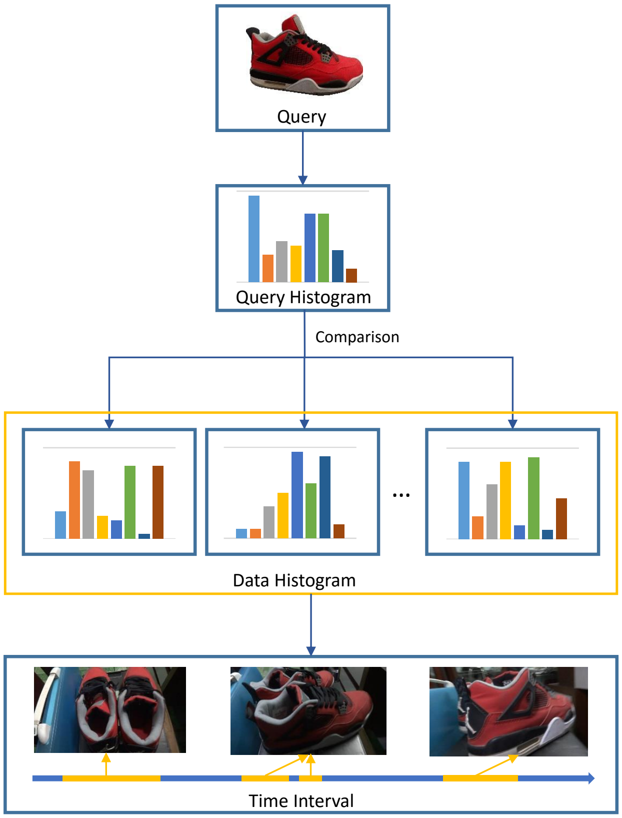

Fig. 3. Overview of how data is created and compared to give the result

The authors choose the best detector in experiment 1 to conduct experiment on the quality of two normalization methods: L1 Norm and L2 Norm. The objective is to determine which normalization approach gives better result in our problem.
L1 Norm and L2 Norm are mentioned in section III.D.2.

3) **Experiment on the effect of TF-IDF**
The authors choose the method which gives best result in experiment 2 to conduct experiment on the effect of Term Frequency - Inverse Document Frequency to the result. The main objective is to determine if we should use TF-IDF or not.
Term Frequency - Inverse Document Frequency is mentioned in section III.D.1.

4) **Experiment on the quality of distance metrics**
Combining results from 3 experiments above, the authors now compare the histogram of the images. The authors compare the effectiveness of two distance metrics: Symmetric distance (Euclidean distance) and asymmetric distance. The main goal is determining which metric is best for representing the relationship of the query and database images to give a good result.

5) **Experiment on the effect of number of clusters to the result**
The accuracy of our system depends much on the right number of visual words (clusters). The authors consider the importance of this experiment to reduce the storage memory and computational cost.

Note: Except for the fifth experiment, the default number of clusters which is used to give the result of our problem when implementing Bag of Words model is 50,000 clusters.

The purpose of this experiment is to point out a potential method to implement Bag-of-Words model in finding objects through past videos problem.
The number of query images is 247. For each query image, the authors choose top 10 output images and compare to the ground truth. In total, the database contains 2470 answer images. In each experiment of IV.B.1, the method is considered to give good result based on the number of true positive images (correct images compared to ground truth).

Bag-of-Words model represents extracted features as a histogram for each image. The result of our proposed method depends much on the Bag-of-Words histogram. Therefore, the feature extraction step has significant meaning. It is the groundwork to create and give comments on the Bag-of-Words histogram. Due to those reasons, images of object which have more local keypoint descriptors and features are more likely to be found. Intuitively, we expect objects which have many different and special texture or great variance of colors with a clear view of image has a high probability to give good result. Our system may be able to point out the appearance of those objects in different intervals of time with high accuracy. In order to verify our prediction, we conduct experiments on 30 different objects, 3 different query images per object on average. In 30 different objects, 40% of objects are considered to have good criteria as we expect by observation. We receive the result in section IV.B.2. The number of output images is chosen based on the threshold. By this way, we can evaluate exactly the accuracy of our proposed method based on appropriate number of output images in a dataset.
Query objects are shown in Figure 4 as a table. In that table, row 1 and 2 contain objects which are considered to have good criteria.

B. Result

1) **Choosing the best method for BoW model**
The authors choose the default method is using Hessian Affine Detector, L2 normalization with TF-IDF, asymmetric distance and codebook with 50,000 clusters .
Experiment on the quality of detectors used in features extraction.
We compare detectors in features extraction step: Hessian Affine, DoG and PySIFT. The result is graphed as column chart in Figure 5. According to the chart Figure

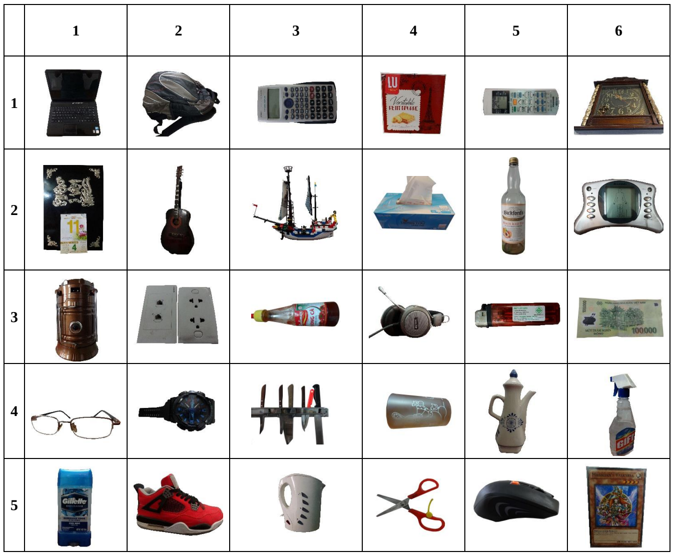

Fig. 4. Query objects

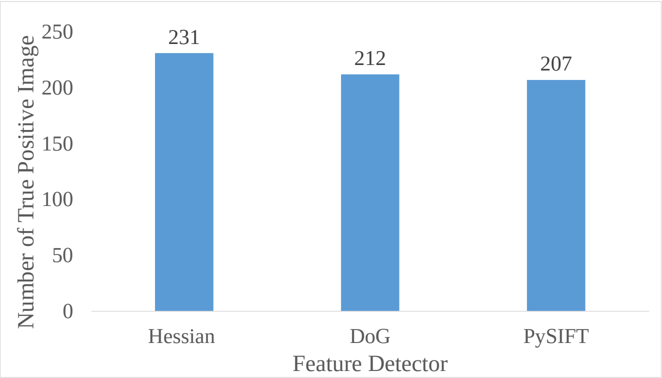

Fig. 5. Experiment on the quality of Hessian Affine detector, DoG detector and PySIFT detector

5, the Hessian Affine detector gives the best results with 231 accepted images after applying Bag of Words model to find objects. Although the resulting difference of 3 detectors is not significant, we still vote for Hessian Affine Detector. Therefore, we still keep the Hessian Affine detector for later experiment.

Experiment on the quality of 2 normalization method.

We compare 2 different normalization methods: L1 Normalization and L2 Normalization. The result is graphed as column chart in Figure 6.

Due to the result shown in Figure 6 with the best output of 231 accepted images, we still keep the L2 Normalization to apply to Bag of Words model in our problem. The result between two normalize method are significantly different.

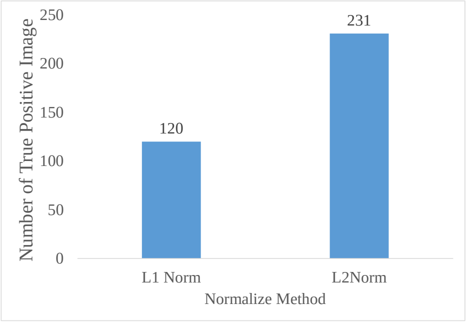

Fig. 6. The number of accepted images between L1 and L2 normalization methods

Experiment on the effect of TF-IDF We compare 2 different methods to check the effect of TF-IDF. The experiment result is demonstrated as column chart in Figure 7.

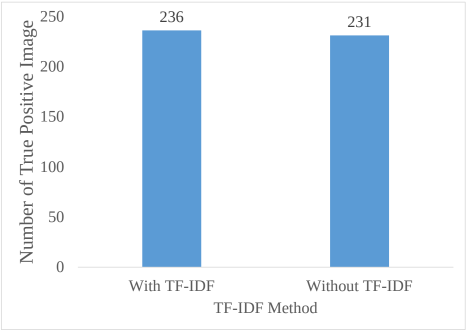

Fig. 7. The effect of TF-IDF to the result

We can make a conclusion based on the result shown in Figure 7 to keep using TF-IDF to rank the output images and reduce unrelated output. This decision plays an important role in weighting the images to receive high-accurate result.

Experiment on the quality of distance metrics

We compare the 2 distance metrics to determine the exact metrics to use in our system. The purpose of this experiment is to check the relationship between query image and result images if it is symmetric or asymmetric. The experiment result is represented as column chart in Figure 8.

According to the information shown in Figure 8, asymmetric distance with high accuracy of accepted images result, is strongly proposed to use.

Experiment on the effect of number of clusters to the result

The purpose of this experiment is to check the effect of number of clusters to the output. The result of this experiment is graphed as column chart in Figure 9.

2) **Experiment on the effect of texture and colors of**

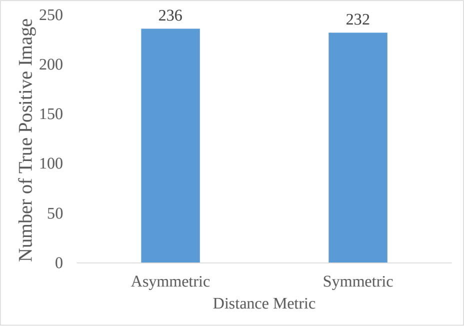

Fig. 8. The number of accepted images between the symmetric distance and asymmetric distance

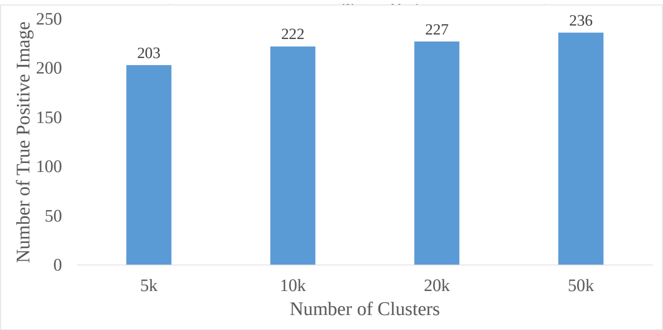

Fig. 9. The effect of number of clusters to the result

objects to the accuracy of finding things by Bag-of-Words model

An object in query data may have different views. For all images of object with the same category, we get the answer of the second when that object appears in the video. The answer is compared with the groundtruth to calculate the accuracy of our system. If any second in the answer agrees with the groundtruth, it is true positive. If any second in the answer does not exist in the groundtruth, it false positive.

The precision and accuracy of our system is computed based on the following formulas:

$$balance\ accuracy = \frac{0.5 * tp}{tp + fn} + \frac{0.5 * tn}{tn + fp}$$

- tp: the number of true positive answers - the number of images detected by the system and they contain the object.
- tn: the number of true negative answers - the number of images which are not detected by the system and they do not contain the object.
- fp: the number of false positive answers - the number of images detected by the system but they do not contain the object.
- fn: the number of false negative answers - the number of images which are not detected by the system but they contain the object.

The result of creating threshold to receive the number of output images instead of 10 images as mentioned in experiment 1. The accuracy of finding objects is graphed in Figure 10. According to the result in Figure 10, objects which are predicted to give good result actually have high accuracy. Other objects give poor result with low accuracy. The accuracy of them change insignificantly. This result supports the hypothesis of the authors that objects with many different and special texture or great variance of colors. The accuracy of finding these objects is 64.23%. In contrast, the accuracy of finding objects with flat texture or invariance of colors is 49.78%. In general, our system's accuracy of finding an object is 55.43%. The authors also propose user to choose the appropriate threshold based on his/her using purpose.

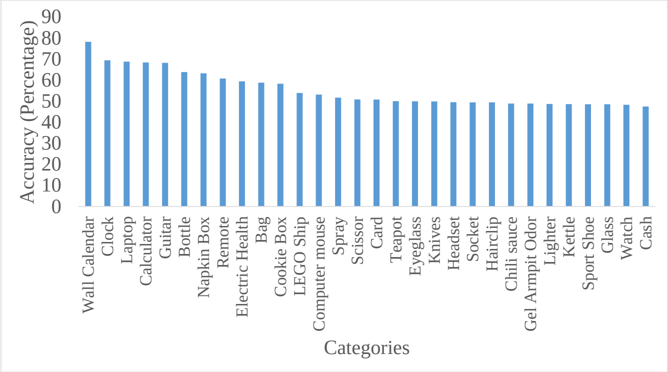

Fig. 10. Balance Accuracy of our system with threshold of 1.9

V. Conclusion

As the result received through experiment, the authors make the following contributions: We propose an optimal method to apply Bags-of-Words model to image retrieval with high accuracy. We propose to use L2 Normalization to increase accuracy of our system up to 55.43%.

We are still studying the threshold to determine at what score, an image can have potential to give good result or bad result. We also try to find out a method to improve our system to find objects with invariance of colors or few texture with the accuracy as high as the ones that which has a lot of special texture or variance of colors.

In the future, the authors expect our system to be developed and applied in many different fields. For example, 3 applications in 3 different fields can be developed based on our system:

1) In film industry, especially in 3-D film production, our system can be developed to find the scenes which contain specific objects or phenomena to add effects to the film automatically.
2) In Smart Home development, our system can be researched to create smart house which can recognize dangerous objects around their houses and give warnings.

3) In Robotics, our system can be developed to attach with a robot to help people finding lost object in dangerous places.

References

[1] Rodney E. Peters, Richard Pak, Gregory D. Abowd, Arthur D. Fisk, and Wendy A. Rogers, "Finding lost objects: Informing the design of ubiquitous computing services for the home," 2004.

[2] Matt Schiesl, "http://listverse.com/2012/10/27/top-10-everyday-things-we-forget/," October 27, 2012 and last access April 25, 2016.

[3] Saleem Ahmad, Muhammad Ziaullah, Leena Rauniyar, Meng Su, and Yue Zhang, "How does matter lost and misplace items issue and its technological solutions in 2015 -a review study," *IOSR Journal of Business and Management (IOSR-JBM)*, vol. 17, pp. 79–84, April 2015.

[4] Josef Sivic and Andrew Zisserman, "Video google: A text retrieval approach to object matching in videos," *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, vol. 2, pp. 1470 – 1477, 13-16 Oct. 2003.

[5] Andr Aichert, "Feature extraction techniques," January 2008.

[6] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman, "Object retrieval with large vocabularies and fast spatial matching," 2007.

[7] Jianxin Wu, Wei-Chian Tan, and James M. Rehg, "Efficient and effective visual codebook generation using additive kernels," *The Journal of Machine Learning Research*, vol. 12, pp. 3097–3118, February 2011.

[8] David G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91 – 110, 2004.

[9] Jingjing Meng, Junsong Yuan, Gang Wang, and Jianbo Xu, "Object instance search in videos," 2013.

[10] Ameesh Makadia, Vladimir Pavlovic, and Sanjiv Kumar, "Baseline for image annotation," *International Journal of Computer Vision*, vol. 90, pp. 88–105, October 2010.

[11] Yin Zhang, Rong Jin, and Zhi-Hua Zhou, "Understanding bag-of-words model: A statistical framework," *International Journal of Machine Learning and Cybernetics*, vol. 1, pp. 43–52, December 2010.

[12] G. Qiu, "Indexing chromatic and achromatic patterns for content-based colour image retrieval," *PATTERN RECOGNITION*, vol. 35, pp. 16751686, August 2002.

[13] Jialu Liu, "Image retrieval based on bag-of-words model," April 2013.

[14] Mohamed Aly, Mario Munich, and Pietro Perona, "Indexing in large scale image collections: Scaling properties, parameter tuning, and benchmark," October 2010.

[15] Jun Yang, Yu-Gang Jiang, Alexander G.Hauptmann, and Chong-Wah Ngo, "Evaluating bag-of-visual-words representations in scene classification," pp. 197–206, 2007.

[16] Pascal Knierim, Jens Nickels, Steffen Musiol, Bastian Knings, Florian Schaub, Bjr n Wiedersheim, and Michael Weber, "Find my stuff: A search engine for everyday objects," 2012.

[17] Toyohisa Nakada, Hideaki Kanai, and Susumu Kunifuji, "A support system for finding lost objects using spotlight," pp. 321–322, 2005.

[18] Mehdi Samadi, Thomas Kollar, and Manuela Veloso, "Using the web to interactively learn to find objects," pp. 2074–2080, 2012.

[19] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, "Surf: Speed up robust features," *Computer Vision and Image Understanding*, vol. 110, pp. 346359, June 2008.

[20] Carl Vondrick, Aditya Khosla, Tomasz Malisiewicz, and Antonio Torralba, "Hoggles: Visualizing object detection features," 2013.

[21] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *International Journal of Computer Vision*, vol. 60, pp. 63–86, October 2004.

[22] Marius Muja and David G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," *International Conference on Computer Vision Theory and Applications*, 2009.

[23] Cai-Zhi Zhu, Herve Jegou, and ShinIchi Satoh, "Query-adaptive asymmetrical dissimilarities for visual object retrieval," *2013 IEEE International Conference on Computer Vision*, pp. 1705 – 1712, 2013.