Persons-In-Places: a Deep Features Based Approach for Searching a Specific Person in a Specific Location

Vinh-Tiep Nguyen, Thanh Duc Ngo, Minh-Triet Tran, Duy-Dinh Le and Duc Anh Duong
University of Information Technology, University of Science
E-mail: {tiepnv, thanhnd}@uit.edu.vn, tmtriet@fit.hcmus.edu.vn, {duyld,ducda}@uit.edu.vn

Keywords: video instance search, deep neural network, location search, person search

Received: March 29, 2017

Video retrieval is a challenging task in computer vision, especially with complex queries. In this paper, we consider a new type of complex query which simultaneously covers person and location information. The aim to search a specific person in a specific location. Bag-Of-Visual-Words (BOW) is widely known as an effective model for presenting rich-textured objects and scenes of places. Meanwhile, deep features are powerful for faces. Based on such state-of-the-art approaches, we introduce a framework to leverage BOW model and deep features for person-place video retrieval. First, we propose to use a linear kernel classifier instead of using L_2 distance to estimate the similarity of faces, given faces are represented by deep features. Second, scene tracking is employed to deal with the cases face of the query person is not detected. Third, we evaluate several strategies for fusing individual person search and location search results. Experiments were conducted on standard benchmark dataset (TRECVID Instance Search 2016) with more than 300 GB in storage and 464 hours in duration.

Povzetek: V prispevku je opisana metoda povpraševanja po osebi in lokaciji iz video vsebin.

1 Introduction

With the rapid growth of video recording devices, many videos from diverse domains such as professional or amateur film making, surveillance and home recording are being created. These vast video collections are being shared on video broadcasting sites (e.g., YouTube). One of the most fundamental needs is to help users find exactly what they are looking for in video databases. To search directly on videos, we consider an approach—visual instance search on video databases. The term *instance search* (INS) is defined formally by TRECVID [13]: finding video segments of certain specific object, place or person, given visual examples from a video collection. There are varieties of query types including rich-textured, fairly-textured or deformable object. These make instance search is a very challenge task since we do not know any prior information about the query.

The objective of this problem is to find the person *and* the location in a large-scale video dataset. This type of query is important since person and location are two most popular query objects. It has many applications in practice such as: surveillance systems, personal video archive management. This query topic is also a very hard topic because there are many variations in size, light condition, view change. Figure 1 gives an example of this type of query. Images in the first row are examples of a pub that a user want to search. These images cover multiple views of a location with many irrelevant or noisy objects such as humans, decorations. These objects may cause low retrieval accuracy due to noisy features. Images in the second row are examples of the person that the user also needs to find if he appears at the pub. Persons are special query objects because they are 3D object with multiple views and deformable with different cloths texture features. All of these make our retrieval task with this compound query to be more challenging.

A very natural approach is to combine the scores of recognizing face and location. There are some challenges in this approach:

- The scores are independent and incomparable. It makes typical fusion techniques such as average fusion inefficient.
- Frames with very clear and recognizable faces often have large proportions in appearance but less information about the context scene. Hence, frames which have higher score in recognizing a face may have lower score or low rank for a location, and vice versa. This gives the low performance when simply combining these scores.
- In a video scene that contains a person and a location, both of them are not always shown perfectly: the person may change their head pose in multiple directions while the location may change points of view by the time. However, query examples do not cover all views of target objects.

Most state-of-the-art object instance retrieval systems are based on bottom-up approach with a very well-known

Figure 1: A query topic includes location examples (first row images) and person examples (second row images) marked by magenta boundaries.

model Bag-of-Visual-Words (BOW) [23] which benefits from powerful local descriptors for matching textures, then checks the geometric consistency to further improve the accuracy. This approach relies on the *key assumption* that two similar objects share significant number of local patches that can be matched against each other.

When searching on rich-textured instances which contain enough discriminative texture patterns (e.g. locations, buildings, book covers, paintings, etc.), there are some *ambiguous patches* that share similar shapes with the query instance but belong to an irrelevant object. However, ratio of these patches is low, thus the similarity scores of images containing correct instance are higher than incorrect ones. Moreover, its extensions e.g. geometric consistency checking [16][30], query expansion [8][7][1] also further significantly improve the performance of the searching system.

When searching on highly flexible appearance object such as human, the performance is still very low due to the limited capacity of representation of the BOW model. For the first video segment that the query person appears, the problem is equivalent to face recognition without using other information such as cloths texture feature. From that segment to the end of a scene, people are likely to be in the same place even his/her face disappears. In this paper, we propose a system which leverages both BOW and Convolutional Neural Network (CNN) based feature for retrieving this new type of query. For location search, we combine BOW based and CNN based features to improve the performance. For person search, we use VGG-face feature for recognizing the first video shot that the target person appears. In stead of using distance metric such L_2, we propose to use a linear kernel method to learn high-level feature encoded by a deep CNN. Finally, in order to boost the recall of the system, we implement scene tracking to keep track shots following the high response ones.

The rest of this paper is organized as follows. Section 2 presents related work. Details of our instance search framework is presented in Section 3. Section 4 presents our experiment results on TRECVID dataset. Finally, Section 5 concludes the paper.

2 Related work

To improve the performance of INS systems, multiple techniques have been proposed, such as rootSIFT feature [1], large vocabulary [16], soft assignment[17]. Among them, spatial verification is one of the most effective approaches, and also serves as the prerequisite step for other advanced techniques such as query expansion. Spatial verification can be classified into two categories: spatial reranking [16] [30] [33] and spatial ranking [10] [5] [21]. These approaches work very well on big and rich-textured object such as location.

To further improve the performance, Wan et al. explore deep learning techniques with application to instance search task[31]. They show that deep learning feature from CNN model pre-trained on large-scale dataset can be used for representing image or object in new instance search task. Moreover, by retraining the deep models on the new domain, the retrieval performance could be boosted significantly. Although the amount of training data is only a few examples per query object, pre-trained network with parameters learned from previous large-scale dataset makes fast convergence on new data domain.

In addition to retrain the CNN network, Babenko et al. also investigate the performance of compressed deep features, where plain PCA or a combination of PCA with discriminative dimensionality reduction result in very short codes with state-of-the-art performance [4]. They explain that passing an image through the network discards much of the information that is irrelevant for classification (and for retrieval). Thus, CNN based neural codes from deeper layers retain less (useless) information than unsupervised aggregation-based representations. Therefore PCA com-

pression works better for neural codes. Beside deep encoding technique, the authors also introduces and evaluates a new simple and compact global image descriptor and investigates the reasons underlying its success [3]. They show that, feature aggregation using sum-pooling technique outperform when using max-pooling on deep features from fully connected layers [18], VLAD[2], democratic aggregation[11] which successfully applied on SIFT feature.

Another problem this paper focuses on is face recognition in images and videos. We classify many methods proposed in the literature into two groups: the ones that do not use deep learning and the ones that do. For the first group (also named "shallow" methods), they start by extracting a representation of the face image using hand-crafted local image descriptors such as SIFT, LBP, HOG [9][12][32]; then they aggregate such local descriptors into an overall face descriptor by using a pooling mechanism, for example the Fisher Vector [14][22].

This work is concerned mainly with deep architectures which currently reach the state-of-the-art performance. The idea of such methods is to use a CNN feature extractor with parameters learned by composing several linear and non-linear operators. One of the representative methods for this approach is DeepFace [28]. This method uses a deep CNN trained to classify faces using a dataset of 4 million examples of 4000 persons. The goal of training is to minimize the distance between congruous pairs of faces (i.e. portraying the same identity) and maximize the distance between incongruous pairs, a form of metric learning. The authors later extended this work in [29], by increasing the size of the dataset to 10 million persons and 50 images per person. They proposed a bootstrapping strategy to select identities to train the network and showed that by controlling the dimensionality of the fully connected layer the generalisation of the network can be improved.

The DeepId series of papers by Sun et al. [24][26][27][25], extensions of the DeepFace, each of which improves the performance on LFW and YFW incrementally and steadily. A number of new ideas were introduced by incorporating over this series of papers, including: using multiple CNNs [26], a Bayesian learning framework [6] to train a metric, multi-task learning over classification and verification [24], different CNN architectures which branch a fully connected layer after each convolution layer [27], and very deep networks [25]. Compared to DeepFace, DeepID does not use 3D face alignment, but a simpler 2D affine alignment and trains on combination of CelebFaces [26] and WDRef [6]. However, the final model in [25] is quite complicated involving around 200 CNNs.

Recently, a research from Google [20] trains a CNN using a massive dataset of 200 million face identities and 800 million image face pairs. Their triplet-based loss considers two congruous (*a,b*) and a third incongruous face *c* in comparison. Differently from other metric learning approaches, their goal is to make *a* closer to *b* than *c*; comparisons are always relative to a pivot face. In training this loss is applied at multiple layers, not just the final one.

In this paper, we follow the VGG-Face descriptor network [15] which designs a procedure that is able to assemble a large-scale dataset, with small label noise, whilst minimizing the amount of manual annotation involved. They use weaker classifiers to rank the data presented to the annotators for reranking. They also show that a deep CNN can achieve results comparable to the state-of-the-art with appropriate training without any special techniques.

In other to apply in a new task (instance search) and data domain, instead of using the activation of the last layer, we propose to use the feature extracted from one of the fully connected layers with a linear classifier (e.g support vector machine with linear kernel) to train face model for the query person. To further improve the performance of the instance search system, especially in the case that the target person turns his/her back to the camera, we propose to combine person tracking with scene tracking.

3 Proposed framework

This section describes our proposed framework and its configurations. Our proposed system includes four main modules: BOW based retrieval, location learning for verification, face learning for recognition and final fusion. Figure 2 sketches out the work flow of main components in our INS system. Given a compound query topic including person and location examples, our goal is to rank video shots containing that combination. Each example is a video frame of location or person captured at a specific point of view as shown in Figure 1. In our framework, instead of using all frames of a video shot, we perform key frame extraction at 5 frames per second for saving computational cost.

For simplicity of notation, we only consider a set of query examples and key frames of a shot in the video dataset. Other shots are processed similarly. Firstly, for each location example, we extract local features using Hessian-Affine detector and rootSIFT descriptor, then quantize using a codebook trained on video database. In order to reduce the effect of noisy features given by irrelevant persons, we remove all visual words inside bounding boxes detected by a person detector. In this paper, we use Faster RCNN[19] with pre-trained network on PASCAL VOC 2007 to find person regions. Each frame of location is finally represented by a BOW feature vector L_k with tf-idf weighting scheme. For each person example, we only use the information detected by face detector since the target person may change clothes by the time. Each face bounding box is described by a CNN based descriptor and represented by a feature vector F_p.

Since location and person examples are independent, we can compute two rank lists independently. However, BOW model could perform in large-scale video data, we use location features to retrieve rank lists as the first step, then use face features for later reranking. Top K retrieved

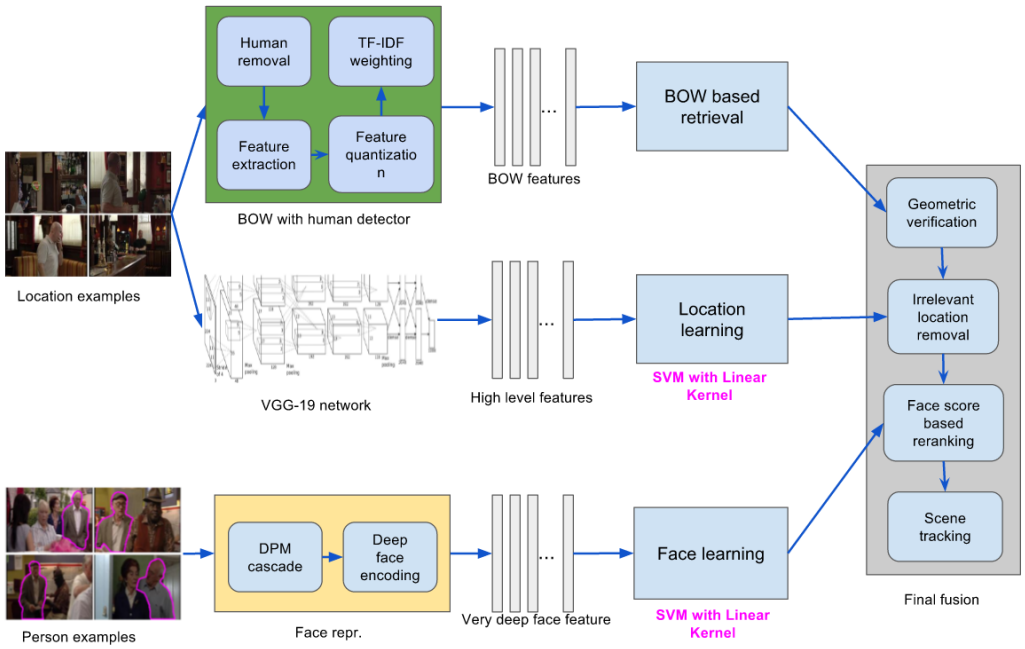

Figure 2: Framework overview.

shots based on S_{BOW} similarity score are then used for the reranking stage. Note that, BOW model is a non-structured model which does not take into account the spatial relationship between visual words. To remove irrelevant shot, we combine both RANSAC based algorithm and learning based approach for high level feature vector produced by a very deep CNN network VGG-19.

The second part of our system is person recognition based reranking. A person example includes a color image and its mask which helps the system to separate interested person from irrelevant objects. In this case, we only focus on face feature since the target person changes the cloths over time. We use a face detector and face descriptor to extract representative feature of the query person. After this stage, each person is represented by a set of deep feature vectors. A typical way to compare face features is using symmetric distance or similarity score. In this approach, each component of a feature vector is processed evenly. However, this vector is a high-level feature which describes many parts of a face. Some of them are important and some are not. Hence, we propose to use a linear classifier to learn the weights of a face feature, then compute the similarity score between the face model and a video shot.

Finally, we propose a final fusion step in which, it takes into account all components of the system including: BOW based location search, CNN based irrelevant location removal, face based reranking and scene tracking. In a video scene that contains a person and a location, both of them are not always shown perfectly: a person may change their head pose in multiple directions while a location may change points of view by the time. However, query examples of face and location are limited and incomplete. To propagate the score of positive shot, we inherit that value for the next scenes with a multiplication factor.

3.1 Location search

In the first stage of the system, we retrieve top K shots that is similar to the location examples using BOW model with local feature. In this paper, we use the state-of-the-art configuration of BOW framework that have been used for image retrieval. Local features of each key frame of a shot are extracted using Hessian-affine detector and rootSIFT feature descriptor. Each feature is represented by a 128-dimensional vector. All features gathered from database video frames are clustered using approximate K-Means algorithm (AKM) with a very large number of codewords. Since the limitation of hardware computation, only 100 million features are randomly sampled to train 1 million codewords. These features are then quantized using the codebook with hard-assignment strategy. Finally, each video frame is represented by a very sparse BOW feature vector using tf-idf (term frequency-inverse document frequency) weighting scheme. Because the rank list only counts video shots not video frames, we aggregate all BOW vectors of frames of a shot into a single one for compact representation and fast retrieval. Using the following encoding scheme, frame j-th of i-th video shot is represented by a BOW feature vector $S_{i,j}$. We accumulate all vectors of a shot into a single one using average pooling:

$$S_i = \frac{1}{n}\sum_{j=1}^{n} S_{i,j} \quad (1)$$

where, n is number of key frames of the shot.

Feature vectors of video shots are then transferred to build an inverted index which helps to significantly boost

(a) Location example with irrelevant persons (b) Person example with irrelevant context

Figure 3: Two images illustrate a location example (the left-hand side one) and a query person example (the-right hand side one). For the location example, there may have some irrelevant persons (marked by yellow boundaries) whose noisy visual words take part in the BOW feature vector of the frame. For the person example (marked by magenta boundary), face feature is one of the most important features for retrieving.

the speed of retrieval. The similarity between the i-th shot and the given location is computed by the following formula:

$$LS_i = \frac{1}{n'} \sum_{k=1}^{n'} asym(L_k, S_i) \quad (2)$$

where, n' is the number of query examples and $asym$ is an asymmetrical similarity score[34].

Top K shots returned by BOW model are then reranked in the next steps. One important parameter in this initial step is K, the threshold for selecting top ranked shots. By observing the z-score normalized distance of all query examples, we found that they have the same distribution as shown in Figure 4. Intuitively, we fixed the cut off threshold for top K shots is -2.5.

The main assumption of BOW model is that two similar objects share significant number of local patches that can be matched against each other. The chosen query examples are often captured in perfect views due to the meticulousness of user while database frames are not always. When changing point of view significantly, local feature based BOW model gives bad retrieval performance. To be more robust with point of view, we represent each video frame by a high-level feature vector derived from a fully connected layer of CNN network. We use a very deep pre-trained network, i.e. VGG-19, and remove the last layer which commonly used for classification task. Video frames are re-sized and normalized before transferring to the feed forward network. The output of the network is a 4096 dimensional feature vector representing the whole video frame. Comparing two video frames is equivalent to comparing their representing feature vectors. However, using symmetric metric such as Euclidean distance (L_2) may result in low accuracy since all components of a feature vector have the same role. In fact, for each location, some of the components are important. A learning method is proposed to magnify the role of these key components.

3.2 Face feature learning for reranking

The second part of the query is person examples. Face recognition is a very popular approach to identify a person. Faces are detected using DPM cascade detector [32] applied in maximum 5 key frames per shot. Then, face feature vector are extracted using VGG-Face descriptor, a CNN based network[15]. Particularly, each face image is represented by a 4096 dimensional deep feature vector.

After this stage, each person is represented by a set of deep feature vectors $\{F_1, F_2, ..., F_m\}$ where m is the number of face examples. We perform similarly to each frame of a video shot. $S_{F_{i,j,k}}$ represents feature vector extracted from a face of a person in a video frame. A natural way to compute the similarity between a person and a shot is to take the minimum distance between all pairs of face feature vectors. The distance formula is given as following:

$$FS_i = \min_{l,j,k} L_2(F_l^*, S_{F_{i,j,k}}^*)$$

where F_l^* and $S_{F_{i,j,k}}^*$ are normalized vector of F_l and $S_{F_{i,j,k}}$, L_2 is Euclidean distance metric.

Although this feature is designed to work with L_2 distance metric, there is a big gap in performance. This could be explained that a face feature vector does not have the same weight for all components. With each face, the weights of components are different. Therefore, we propose to learn these features by a large margin classifier with a linear kernel. Each face candidate of a frame of a shot after transferred to the classifier will be scored by a value. Positive values indicate positive example, and vice versa.

In this paper, we use Support Vector Machine (SVM) with linear kernel to train face features of the target person. Positive features are chosen from the query examples while negative ones are from the last 50 persons of the initial rank list returned from L_2 distance based approach. After training with SVM algorithm, the target person is represented

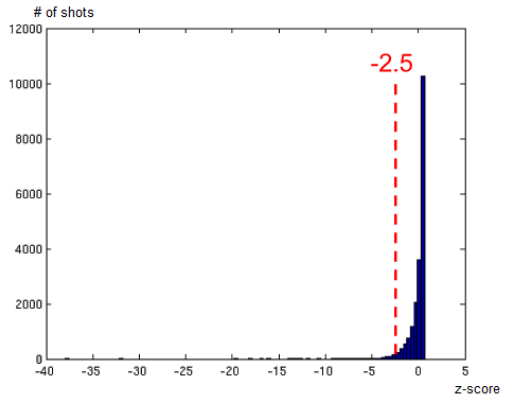

Figure 4: Distribution of z-score normalized distance.

by a single model M.

3.3 Final fusion

This is our main contribution module which leverages the power of BOW model, deep features and machine learning. At first, the rank list returned by BOW based location search is then used as the input of geometric verification step. Visual words of each database video frame is then verified using RANSAC algorithm. The number of inliers represents the similarity between a video frame and query location. The output of geometric verification step is the input of the irrelevant location removal step. Using classifier learned from location examples, we classify each video frame of a shot using linear kernel approach. The output score of a shot is the average of all decision values of frames in that shot. We remove shots which have negative decision values and transfer the remained ones to the next step. In the face based reranking step, we use the face model learned from query examples to recognize persons of a video shot. The output score of shot i-th in this step is the maximum decision value of all frames that belong to:

$$score_i = \max_{j,k} svm(M, S^*_{F_{i,j,k}})$$

where M is the face model, $S^*_{F_{i,j,k}}$ is normalized vector of $S_{F_{i,j,k}}$ and svm is the linear classifier. If $score_i > 0$, it means that there is at least one frame containing the query person in shot i-th and vice versa.

The final step of our system is scene tracking. To deal with cases that the target person appears in a shot but his face is unclear, we transfer the decision value from the last positive shot to the next ones with small decreasing. Note that, we only apply scene tracking to shots which have negative decision values. Assume that two consecutive shots i-th and $i+1$-th have scores $score_i > 0$ and $score_{i+1} \leq 0$. We update $score_{i+1} = \frac{1}{2} score_i$. We also update for the maximum 5 shots with the same factor. The output of this step is the rank list after sorting final score values in descending order.

4 Experiment

4.1 Dataset

To demonstrate the advantage of the proposed method on different types of query, we used TRECVID Instance Search (INS) datasets for evaluation. We used the TRECVID INS benchmarks in year 2016 which was released by NIST. For experimentation, we name this dataset as INS2016.

For the past six years (2010-2015) the instance search task has tested systems on retrieving specific instances of objects, persons and locations. They share the same collection of test videos with a master shot reference. Currently, new query type will be tested by asking systems to retrieve specific persons in specific locations. The dataset contains approximately 244 video files extracted from the BBC EastEnders program with totally 300 GB in storage and 464 hours in duration. Each query topic of INS2016 consists of two set of examples: location and person. For the person set, each example includes an image and corresponding mask to delimit the target entity with others. For location set, only image examples are provided. This INS dataset is very challenging due to the variety in query types: from indoor to outdoor location, unclear to clear person.

Evaluation Protocol. There are 30 query topics or pairs of person-location and about 470 thousand video shots in this challenge. The system must return top 1000 shots that are most similar to each given topic. The ground truth files for each query are created manually and provided by TRECVID organization. To evaluate the performance of each method, we use the mean average precision (MAP) as a standard measurement. Although some evaluations of intermediate results such as location search when combining deep features and BOW are expected, there already has some reports about the performance of state-of-the-art systems on individual query of last year challenges[13]. Therefore, in this paper, we only take care about the performance of compound query.

4.2 Retrieval performance and visualization

In this section, we discuss some quantitative results of our method evaluated against the ground truth gathered from the TRECVID INS 2016. For ease of observation, we use the following abbreviations with descriptions:

- Avg-Fusion: normalized scores of person and location fusion.
- L_2-Reranking: using our framework, after geometric verification step, we rerank the initial top K list using L_2 distance for face features. The similarity score of a frame is the opposite number of min-min distance between face examples and all face detected in frames of a shot. The similarity score of a frame is the opposite number of that distance value. We use mean function for all similarity scores of frames in a shot to represent

Table 1: Comparison between average fusion and reranking methods.

Run	MAP
Avg-Fusion	15.6
L_2-Reranking	18.9

the final similarity (average pooling) (similar to other methods in the experiment).

- CNN-Loc+L_2-Reranking: similar to L_2-*Reranking* but we augment the CNN based location reranking step after geometric reranking step.
- Linear Kernel: similar to the baseline *CNN-loc+L_2-Reranking* but we use linear kernel to learn face model of the query person and compute similarity score with candidate faces.
- Linear Kernel+scene tracking: similar to the Linear Kernel, but we also apply scene tracking to deal with frames that face of target person is not detected.

4.3 Average fusion for person-location query

In many systems, average fusion is one of the simple and effective methods to improve the retrieval performance. However, for compound queries such as location-person, average fusion is not good as face based reranking method as shown in Table 1. It can be explained that, the scores of each target location and person are independent and incomparable. Moreover, frames with very clear and recognizable faces often have large proportions in appearance but less information about the context scene. Hence, a frame has higher score in recognizing a face may have lower score or low rank for a location, and vice versa.

4.4 Deep feature for location reranking

In this section, we want to illustrate that, deep feature for reranking improves the performance pretty much even for rich-texture query object such as location. The experimental result is shown in Table 2. Past state-of-the-art systems of TRECVID showed that, for rich-textured object such as location, local feature based BOW model is one of the most suitable choices. However, in case of real life videos, the proportion of location evidences is very small. Using CNN features of the query location, the system has more information to keep scenes that seems to be removed by the cut-off threshold in geometric verification step.

4.5 Face feature learning and scene tracking

Table 3 summarizes the results of using our different methods, measuring their relative performance in terms of the MAP score. From the table, we can see that the first proposed method (Linear Kernel) performs much better than the baseline one which only uses L_2 distance (CNN-Loc+L_2-Reranking), showing a gain in the MAP from 19.8% to 25.9%. Moreover, with scene tracking step, the final performance is significantly boosted from 25.9% to 50.6%.

Table 2: Comparison of retrieval systems with and without high-level feature reranking.

Run	MAP
L_2-Reranking	18.9
CNN-Loc+L_2-Reranking	19.8

Table 3: Experimental results on different configurations for TRECVID INS 2016.

Run	MAP
Linear Kernel + scene tracking	50.6
Linear Kernel	25.9
CNN-Loc+L_2-Reranking	19.8

Also, note that the scene tracking step not only keeps the high precision but also improves the recall compared to Linear Kernel method. Because there are many cases that the target persons do not put their faces in front of the camera, hence many shots are lost in the final rank list. By using scene tracking, the total recall of the retrieval system is improved surprisingly. This can be observed on the precision-recall curves as shown in Figure 5 where the curve of *Linear Kernel+scene tracking* is significantly higher than the other ones.

To show the efficiency of the proposed method compared to the baseline system, we visualize the rank list returned from the systems. The query topic is given in Figure 1. Top six shots returned from the system using L_2 distance and Linear Kernel classifier are visualized in Figure 6. Each row shows the key frames of a shot of a rank list. When using L_2 distance, the precision is very low, that is the reason why top six rank list of the baseline contains many irrelevant shots marked by red bounding boxes. Using Linear Kernel classifier, the precision of the system is improved significantly, hence the ratio of relevant shots is very high.

5 Conclusion

Inspired by recent successes of deep learning techniques, in this paper, we attempt to leverage the powerful of deep feature in instance search task. We aim to use deep feature as a tool for reranking the location search result by bridging the semantic gap made by BOW model. Moreover, to search for more difficult object which is deformable and could be

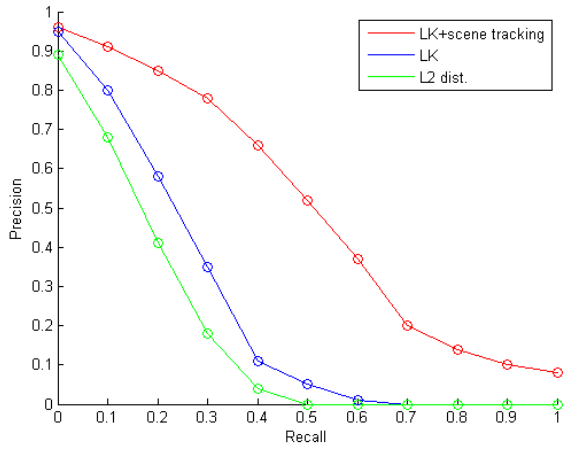

Figure 5: Precision recall curves when conducting experiment on TRECVID INS 2016.

captured in different environments, we propose to apply a machine learning approach to learn deep features extracted from human face detected in video frame. In particular, we investigate a framework of combining BOW model and deep learning based feature with application to instance search task with a new type of query topic: a specific person in a specific location. By conducting experiments on a large-scale dataset, we proved that our proposed method significantly improves the performance of retrieval.

In future work, we will investigate on advanced deep learning techniques such as retraining network with new data generated from query examples. We also evaluate the retrieval systems on other diverse datasets for more in-depth empirical studies.

Acknowledgement

The video frames from BBC Eastenders video used in this document are programme material copyrighted by BBC.

This research is funded by Vietnam National University HoChiMinh City (VNU-HCM) under grant number B2017-26-01.

References

[1] R. Arandjelović and A. Zisserman. Three things everyone should know to improve object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, pages 2911–2918, Washington, DC, USA, 2012.

[2] R. Arandjelović and A. Zisserman. All about VLAD. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1578–1585, 2013.

[3] A. Babenko and V. S. Lempitsky. Aggregating deep convolutional features for image retrieval. *CoRR*, abs/1510.07493, 2015.

[4] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky. Neural codes for image retrieval. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, pages 584–599. Springer International Publishing, Cham, 2014.

[5] Y. Cao, C. Wang, Z. Li, L. Zhang, and L. Zhang. Spatial-bag-of-features. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3352–3359, June 2010.

[6] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun. Bayesian face revisited: A joint formulation. In *Proceedings of the European Conference on Computer Vision - Volume Part III*, ECCV'12, pages 566–579, Berlin, Heidelberg, 2012. Springer-Verlag.

[7] O. Chum, M. Perdoch, A. Mikulik, and J. Matas. Total recall ii: Query expansion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 889–896, Los Alamitos, CA, USA, 2011.

[8] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *IEEE International Conference on Computer Vision*, 2007.

[9] R. G. Cinbis, J. Verbeek, and C. Schmid. Unsupervised Metric Learning for Face Identification in TV Video. In *ICCV 2011 - International Conference on Computer Vision*, pages 1559–1566, Barcelona, Spain, Nov. 2011. IEEE.

[10] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *Proceedings of the European Conference on Computer Vision: Part I*, ECCV '08, pages 304–317, Berlin, Heidelberg, 2008. Springer-Verlag.

[11] H. Jégou and A. Zisserman. Triangulation embedding and democratic aggregation for image search. In *CVPR - International Conference on Computer Vision and Pattern Recognition*, Columbus, United States, June 2014.

[12] C. Lu and X. Tang. Surpassing human-level face verification performance on lfw with gaussian face. In *Proceedings of the AAAI Conference on Artificial Intelligence*, AAAI'15, pages 3811–3819. AAAI Press, 2015.

[13] P. Over, J. Fiscus, G. Sanders, D. Joy, M. Michel, G. Awad, A. Smeaton, W. Kraaij, and G. Quénot. Trecvid 2014 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2014*. NIST, USA, 2014.

Figure 6: Result visualization of query from Figure 1. a) Top 6 rank list using L_2 distance. b) Top 6 rank list using Linear Kernel classifier.

[14] O. M. Parkhi, K. Simonyan, A. Vedaldi, and A. Zisserman. A compact and discriminative face track descriptor. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, IEEE, 2014.

[15] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *British Machine Vision Conference*, 2015.

[16] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[17] J. Philbin, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *In CVPR*, 2008.

[18] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: An astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, CVPRW '14, pages 512–519, Washington, DC, USA, 2014. IEEE Computer Society.

[19] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Neural Information Processing Systems (NIPS)*, 2015.

[20] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[21] X. Shen, Z. Lin, J. Brandt, S. Avidan, and Y. Wu. Object retrieval and localization with spatially-constrained similarity measure and k-nn re-ranking. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3013–3020, June 2012.

[22] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman. Fisher Vector Faces in the Wild. In *British Machine Vision Conference*, 2013.

[23] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1470–1477, Oct. 2003.

[24] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *Proceedings of the International Conference on Neural Information Processing Systems*, NIPS'14, pages 1988–1996, Cambridge, MA, USA, 2014. MIT Press.

[25] Y. Sun, D. Liang, X. Wang, and X. Tang. Deepid3: Face recognition with very deep neural networks. *CoRR*, abs/1502.00873, 2015.

[26] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *Proceedings of the IEEE Conference on Computer Vision*

and Pattern Recognition, CVPR '14, pages 1891–1898, Washington, DC, USA, 2014. IEEE Computer Society.

[27] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. *CoRR*, abs/1412.1265, 2014.

[28] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[29] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Web-scale training for face identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[30] G. Tolias and Y. S. Avrithis. Speeded-up, relaxed spatial matching. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 1653–1660, 2011.

[31] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li. Deep learning for content-based image retrieval: A comprehensive study. In *Proceedings of the ACM International Conference on Multimedia*, MM '14, pages 157–166, New York, NY, USA, 2014. ACM.

[32] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *in Proc. IEEE Conf. Comput. Vision Pattern Recognition*, 2011.

[33] W. Zhang and C.-W. Ngo. Searching visual instances with topology checking and context modeling. In *Proceedings of the ACM Conference on International Conference on Multimedia Retrieval*, ICMR '13, pages 57–64, New York, NY, USA, 2013. ACM.

[34] C. Zhu, H. Jegou, and S. Satoh. Query-adaptive asymmetrical dissimilarities for visual object retrieval. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, pages 1705–1712. IEEE, 2013.