LEARNING DEEP COMPACT DESCRIPTOR WITH BAGGING AUTO-ENCODERS FOR OBJECT RETRIEVAL

Haiyun Guo, Jinqiao Wang and Hanqing Lu

National Laboratory of Pattern Recognition, Institute of Automation
Chinese Academy of Sciences, Beijing, China, 100190
{haiyun.guo, jqwang, luhq}@nlpr.ia.ac.cn

ABSTRACT

Content based object retrieval across large scale surveillance video dataset is a significant and challenging task, in which learning an effective compact object descriptor plays a critical role. In this paper, we propose an efficient deep compact descriptor with bagging auto-encoders. Specifically, we take advantage of discriminative CNN to extract efficient deep features, which not only involve rich semantic information but also can filter background noise. Besides, to boost the retrieval speed, auto-encoders are used to map the high-dimensional real-valued CNN features into short binary codes. Considering the instability of auto-encoder, we adopt a bagging strategy to fuse multiple auto-encoders to reduce the generalization error, thus further improving the retrieval accuracy. In addition, bagging is easy for parallel computing, so retrieval efficiency can be guaranteed. Retrieval experimental results on the dataset of 100k visual objects extracted from multi-camera surveillance videos demonstrate the effectiveness of the proposed deep compact descriptor.

Index Terms— Object retrieval, bagging, auto-encoder

1. INTRODUCTION

Nowadays, millions of surveillance cameras have been installed in public areas, producing vast amounts of video data every day. Manually browsing and indexing visual objects in such a large scale video dataset is a time-consuming and boring task for viewers. Content based object retrieval is an effective tool for content management and safety monitoring. However, different from the traditional content based image retrieval (CBIR), object retrieval focuses only on the object of interest, the surrounding pixels irrelevant to the object are usually regarded as background noise and will hamper the retrieval performance. Besides, most of the query objects are usually of tiny size, even imprecise. Furthermore, there are many complex conditions such as illumination change, pose variation, and viewpoint shift existing in the video dataset, which increases the difficulty of object retrieval.

Learning an efficient content descriptor plays a critical role in many visual recognition tasks. Over the past decades, various hand-crafted image descriptors have been proposed in computer vision field. Some researchers mainly utilized global descriptors for object retrieval. For example, Mitrea *et al.* [1] made use of color naming histograms and color moments in multiple instance based object retrieval. However, these global features describe the whole image, thus involving background noise. Some other researchers tended to local descriptors such as SIFT [2], HOG [3], and GLOH [4], all of which are based on feature point detection. However, the query object in surveillance videos is usually not large enough to extract adequate feature points, so local descriptors cannot represent the object well.

In comparison to hand-crafted descriptors, deep features learned by deep models such as deep auto-encoder, deep belief net (DBN) and convolutional neural network (CNN) have been verified [5, 6] to be more effective to capture rich semantic information. Teng *et al.* [7] utilized auto-encoders to map color images into short binary codes for object retrieval. The retrieval results may have similar edge patterns but are not semantically related enough. Razavian *et al.* [8] used CNN features to conduct visual instance retrieval, and outperformed other state-of-the-art methods. However, the deep features in [8] are real-valued and rather high dimensional, thus slowing down the retrieval speed.

Therefore, to efficiently retrieve visual object of interest across large-scale surveillance videos, we propose an effective deep compact descriptor with bagging auto-encoders. To be specific, we utilize the CNN pre-trained on ILSVRC12 dataset and fine-tuned on the surveillance video dataset to abstract deep features. This kind of deep features can abstract rich semantic information from visual object and effectively suppress the background noise and disturbance [6]. To accelerate retrieval speed, we adopt auto-encoder to map the high dimensional real-valued CNN features into short binary codes. For one thing, binary codes are very cheap to store and fast to compare, thus greatly boosting retrieval efficiency. For another, as a nonlinear generalization of principal components analysis (PCA), auto-encoder works much better as a tool to reduce the dimensionality of data [9]. Since auto-encoder is an instable model [10], we take advantage of

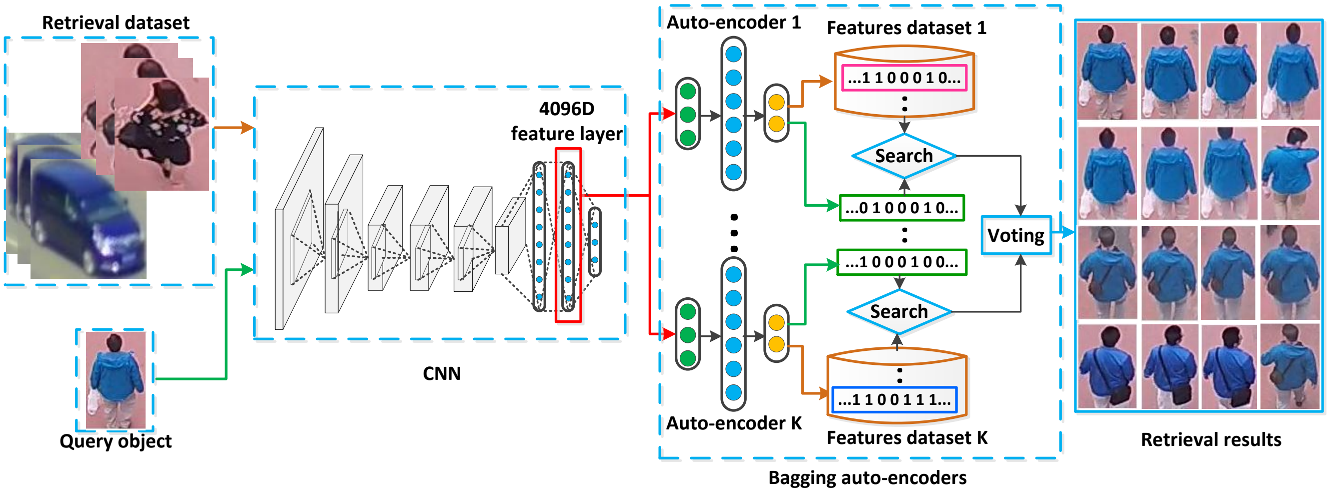

Fig. 1. The flowchart of the proposed object retrieval approach.

a bagging strategy [11] to effectively reduce the generalization error and improve the retrieval accuracy. Specifically, we train multiple auto-encoders using bootstrap replicates of training set and aggregate all the models to obtain the final retrieval results. It is worth mentioning that bagging is easy for parallelization, so the retrieval efficiency is guaranteed.

2. DEEP COMPACT DESCRIPTOR LEARNING

As illustrated in Figure 1, deep compact descriptor learning is divided into two stages: CNN based feature extraction and bagging auto-encoders based feature compression. To begin with, we pre-train CNN on ILSVRC dataset and fine-tune it on the surveillance video dataset. Then deep features can be extracted from FC7 layer of CNN and used to train auto-encoders. Afterwards we make bootstrap replicates of the auto-encoder training set and use them to train multiple auto-encoders respectively. So we obtain multiple descriptors for each object in parallel and adopt the average similarity value of each pair of objects to get the final retrieval results.

2.1. CNN based feature extraction

CNN was first introduced by LeCun [12] in the early 1990's, and has demonstrated astounding performance at challenging tasks such as image classification and object detection. Deep features learned with CNN have been verified to be able to abstract rich semantic information and inherently filter background noise. Therefore, we propose to use CNN to learn visual representations for object retrieval. We adopt the exact CNN architecture specified by Zeiler and Fergus [6] and pre-train the model on ILSVRC12 dataset with the help of Caffe [13]. Since fine-tuning a generic CNN on target dataset can significantly improve the performance, we fine-tune the CNN by retraining the last two fully-connected layers to classify the input images into three classes: person, vehicle and backgrounds. The retraining dataset consists of 255k object blocks extracted from multi-camera surveillance videos. As shown in Figure 2, we extract the 4096-dimensional real-valued deep features from the penultimate layer of CNN. During the CNN model fine-tuning and feature extraction, all the input images are resized to a fixed resolution of 225 × 225 and normalized to have zero mean value. As for the image preprocessing in CNN pre-training phase, we follow the practice in [5].

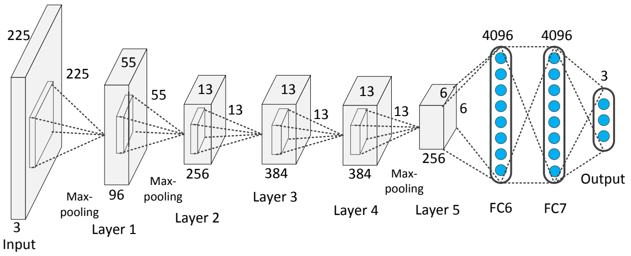

Fig. 2. The structure of CNN.

2.2. Auto-encoder based feature compression

To accelerate object retrieval, we need to transform CNN descriptors into low-dimensional binary codes. As a nonlinear generalization of PCA, auto-encoder is a more effective data dimensionality reduction tool than PCA. Auto-encoder consists of an adaptive multilayer encoder network, which maps the high-dimensional data into low-dimensional code, and a similar decoder network, which recovers the data from the code. The whole auto-encoder is trained with gradient descent by minimizing the error between the original data and its reconstruction. As in [14], the whole training process of an auto-encoder consists of pre-training and fine-tuning. In the pre-training process, we create the encoder network by learning a stack of restricted boltzmann machines (RBM) with standard contrastive divergence specified in [15]. Each RBM has only one hidden layer and trained by using the hidden activities of the previous RBM as its input data. By stacking RBMs, we can obtain better initial weights for gradient descent. Then we unroll the encoder network to create a deep auto-encoder, which is initialized with the weights from the stack of RBMs and fine-tuned by backpropagating reconstruction errors. We train auto-encoders on the dataset of 100k CNN features, which are extracted from 100k randomly selected objects from CNN training set. The deep compact descriptors are obtained by rounding the outputs of the logistic units in the central code layer of auto-encoders to 0 or 1.

2.3. Bagging auto-encoders

As verified in [10], neural networks are instable models and perturbing the learning set can cause significant changes in the constructed model especially when the distribution of samples is uneven. Bagging is an efficient ensemble method for improving instable predication models, and is extensively applied to classification problems. As demonstrated in Figure 3, multiple new training sets are formed by making bootstrap replicates of auto-encoder training set and each of them is used to train an auto-encoder respectively. By reselecting the training set, bagging increases the difference in integration degree of different auto-encoders, thus improving the generalization ability of auto-encoders. Since we can obtain multiple descriptors for each object with the above auto-encoders, we can calculate multiple similarity values for each pair of objects. Then we obtain the final retrieval results by ranking the average similarity values.

Let $f(x, y)$ denotes the ground truth similarity value of object x and y, $S_i(x, y)$ denotes the similarity value calculated with the descriptor produced by i-th auto-encoder, and $S_l(x, y)$ denotes the average similarity value, then we have,

$$S_l(x, y) = E[S_i(x, y)]$$

Since $E[Z^2] \geq (E[Z])^2$, we can find that:

$$E[(f(x, y) - S_i(x, y))^2] = f^2(x, y) - 2f(x, y)E[S_i(x, y)]$$
$$+E[S_i^2(x, y)] \geq (f(x, y) - E[S_i(x, y)])^2$$

Finally, we can derive,

$$(f(x, y) - S_l(x, y))^2 \leq E[(f(x, y) - S_i(x, y))^2]$$

The above analysis guarantees the effectiveness of bagging auto-encoders for improving retrieval accuracy. Though training auto-encoders may take much time, bagging is very easy for parallel computing. Since object retrieval with multiple descriptors can also be conducted in parallel, the training and retrieval efficiency can both be guaranteed.

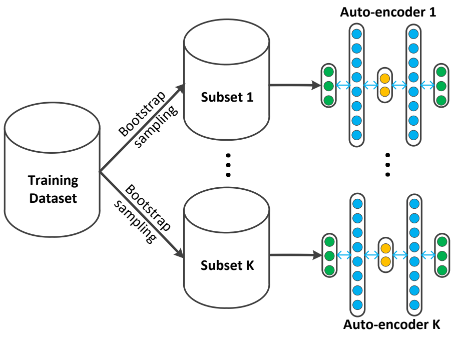

Fig. 3. The flowchart of bagging auto-encoders.

3. EXPERIMENTS

3.1. Experimental setting

Since there is no standard dataset for object retrieval in surveillance videos, we collect surveillance videos from HD cameras installed at residential entrances on the campus and build a dataset consisting of 355k object blocks by using background subtraction and object tracking [16]. The dataset covers various weather conditions, light changes, poses, and viewpoints. 100k objects of the dataset are used to evaluate the proposed retrieval approach. The remaining are employed to train CNN, of which 220k objects are training set, 5k are validation set and 30k are testing set respectively. A total of 300 queries, 100 vehicles and 200 persons, are selected to conduct object retrieval experiments.

We compare the proposed approach with two traditional retrieval methods: Raw and Local Sensitive Hashing (LSH) [17]. Raw is to use the cascaded $32 \times 32 \times 3$ pixel values of the resized input image to represent the object. LSH is employed to encode 32×32 resized object block into 1024-bit code. In our approach, there are three parameters to set: the code length of deep compact descriptor b, the set of bootstrap training set p and the number of bagging models n. We set $b = \{512, 1024\}$, $p = 100,000$ and $n = \{2, 4\}$ respectively. For a fair comparison, all the real-valued features are measured by cosine similarity distance and the binary descriptors are by Hamming distance. Mean average precision (MAP) is used to measure the retrieval performance of different methods.

3.2. Experimental results and analysis

In Table 1 and Table 2, "CNN-Real" means directly employing the 4096-dimensional real-valued CNN features to conduct object retrieval. "CNN-Au512b" and "CNN-Au1024b" denotes encoding CNN features into 512-bit and 1024-bit codes respectively. "CNN-BagAu512b(2)" represents bagging auto-encoders to learn deep compact descriptor, the number in the round bracket denotes the number of model-

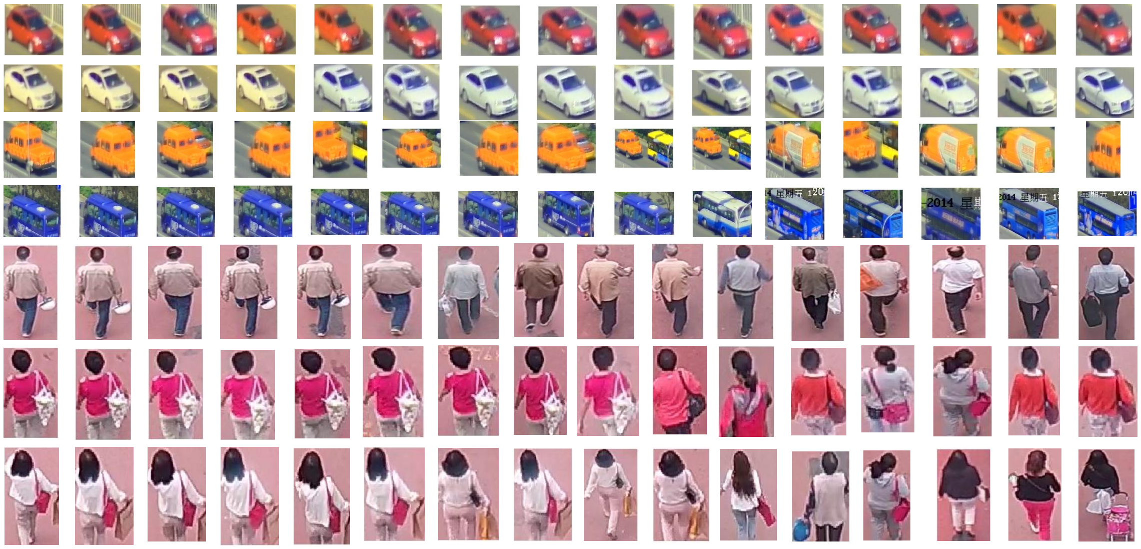

Fig. 4. Selected examples of object retrieval with deep compact descriptor. The first object of each row is the query.

s used in bagging. Compared to Raw and LSH [17], our CNN-BagAu512b(4) and CNN-BagAu1024b(4) improve about 9% and 14% for person retrieval respectively, about 5% and 8% for vehicle retrieval respectively, which verifies the superiority of the proposed deep compact descriptor. Compared to CNN-Au512b and CNN-Au1024b, our CNN-BagAu512b(4) and CNN-BagAu1024b(4) improve about 2% for person retrieval, about 1% for vehicle retrieval, which shows bagging auto-encoders indeed enhances the retrieval accuracy. With the increase of the number of auto-encoders, the retrieval performance increases slightly both for person and vehicle retrieval. In addition, though bagging auto-encoders decreases the retrieval accuracy slightly, it greatly shortens the average retrieval time, from more than 10s taken by CNN-Real to 1.67s by CNN-BagAu1024b(4),0.782s by CNN-BagAu512b(4) respectively. All the experiments are conducted on Intel i5-2400 CPU 3.1 GHz with 20 GB memory. Figure 4 shows the retrieval results returned with "CNN-BagAu1024b(4)", which demonstrates the effectiveness of the proposed deep compact descriptor.

Table 1. Person retrieval results of different methods.

	MAP
Raw	0.4282
LSH [17]	0.4372
CNN-Au512b	0.5172
CNN-BagAu512b(2)	0.5246
CNN-BagAu512b(4)	0.5323
CNN-Au1024b	0.5549
CNN-BagAu1024b(2)	0.5725
CNN-BagAu1024b(4)	0.5764
CNN-Real	0.6122

Table 2. Vehicle retrieval results of different methods.

	MAP
Raw	0.3436
LSH [17]	0.3484
CNN-Au512b	0.3829
CNN-BagAu512b(2)	0.3926
CNN-BagAu512b(4)	0.3939
CNN-Au1024b	0.4110
CNN-BagAu1024b(2)	0.4183
CNN-BagAu1024b(4)	0.4213
CNN-Real	0.4321

4. CONCLUSIONS

In this paper, we propose to learn deep compact descriptor for object retrieval across large scale surveillance videos. We make use of CNN to extract efficient real-valued deep features. Then, to speed up retrieval, we utilize auto-encoders to encode deep features into low-dimensional binary codes. Furthermore, bagging strategy is adopted to reduce the generalization error of auto-encoders, thus boosting the retrieval accuracy. It is worth mentioning that bagging is an ideal tool for parallel computing, so the retrieval efficiency is guaranteed. Experimental results on 100k objects retrieval dataset testify the superiority of the proposed approach.

5. ACKNOWLEDGMENT

This work was supported by 863 Program 2014AA015104, and National Natural Science Foundation of China 61273034, and 61332016.

6. REFERENCES

[1] C.A. Mitrea, I. Mironica, B. Ionescu, and R. Dogaru, “Multiple instance-based object retrieval in video surveillance: Dataset and evaluation,” in *ICCP*. IEEE, 2014, pp. 171–178.

[2] D.G. Lowe, “Object recognition from local scale-invariant features,” in *CV*. Ieee, 1999, vol. 2, pp. 1150–1157.

[3] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *CVPR*. IEEE, 2005, vol. 1, pp. 886–893.

[4] K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors,” *PAMI*, vol. 27, no. 10, pp. 1615–1630, 2005.

[5] A. Krizhevsky, I. Sutskever, and G.E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *NIPS*, 2012, pp. 1097–1105.

[6] M.D. Zeiler and R. Fergus, “Visualizing and understanding convolutional neural networks,” *arXiv preprint arXiv:1311.2901*, 2013.

[7] Kezhen Teng, Jinqiao Wang, Min Xu, and Hanqing Lu, “Mask assisted object coding with deep learning for object retrieval in surveillance videos,” in *Proceedings of the ACM International Conference on Multimedia*. ACM, 2014, pp. 1109–1112.

[8] A.S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, “Cnn features off-the-shelf: an astounding baseline for recognition,” *arXiv preprint arXiv:1403.6382*, 2014.

[9] Geoffrey E Hinton and Ruslan R Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[10] Lean Yu, Shouyang Wang, and Kin Keung Lai, “A neural-network-based nonlinear metamodeling approach to financial time series forecasting,” *Applied Soft Computing*, vol. 9, no. 2, pp. 563–574, 2009.

[11] Leo Breiman, “Bagging predictors,” *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[12] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, and L.D. Jackel, “Backpropagation applied to handwritten zip code recognition,” *NC*, vol. 1, no. 4, pp. 541–551, 1989.

[13] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” *arXiv preprint arXiv:1408.5093*, 2014.

[14] A. Krizhevsky and G.E. Hinton, “Using very deep autoencoders for content-based image retrieval.,” in *E-SANN*. Citeseer, 2011.

[15] Geoffrey Hinton, “A practical guide to training restricted boltzmann machines,” *Momentum*, vol. 9, no. 1, pp. 926, 2010.

[16] Y. Zhang, J. Wang, W. Fu, H. Lu, and H. Xu, “Specific vehicle detection and tracking in road environment,” in *ICIMCS*. ACM, 2011, pp. 182–186.

[17] A. Andoni and P. Indyk, “Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions,” in *FSCS*, 2006.