Face representation using multiple codebook learning: three thin books are better than a thick one

*[1]Suwon Lee
[1]*Dept. Computer Science, KAIST, Korea, leesuwon@kaist.ac.kr*

Abstract

Encoding face images is an essential step in face recognition tasks. One of the most widely used face encoding methods is codebook-based. In codebook-based encoding, it is common to use a thicker codebook for greater discrimination by increasing the code number. However, this approach is more vulnerable to quantization errors as the code number is increased. Instead of using a thicker codebook, we employ multiple codebooks with fewer code words that are trained from randomly sampled sets of low-level feature vectors. Although thin codebooks may have less discriminative power, they are less vulnerable to quantization errors. Moreover, the collaboration between diverse codebooks enhances the discriminative power of the final representation by enabling codebooks to complement each other in increasing the redundant coverage between them. Our experiments on the labeled faces in the wild dataset show that the proposed method yields high performance in completing face-recognition tasks while proving satisfactory performance in low-level face representation.

Keywords: *Face recognition, Face representation, Face description, Multiple codebooks, Codebook learning*

1. Introduction

Face recognition has recently received significant amount of interest because of its challenging nature and the increasing demands for its real-world applications, such as automatic face annotation from video streams [1, 2] or photos [3]. The most essential process of face recognition includes face representation as the primary step. Through this step, the images of the faces are represented in a single vector format; a similarity score is then computed between vectors to make a decision. To represent a face image in vector format, the intensity magnitude of each pixel is not directly used because its value is very sensitive to environment changes, such as illumination, shadowing, and noise. Instead, the value is encoded into the other one by considering the relations between each pixel and its neighboring pixels.

In encoding-based face representation, which is one of the most widely used face-representation methods, the face images are encoded using local encoders to be robust against environmental changes such as illumination, viewpoint, and noise. Two types of approaches for the local encoders exist: handcrafted-based and codebook-based. In the handcrafted-based approach, face images are manually encoded using local descriptors such as local binary patterns (LBP) [4]. Although they have been proven effective to be used as local encoders in several face-recognition tasks, they tend to generate similar encoding results from different faces while degrading the discriminative power of the final representation. However, in the codebook-based approach, the face images are encoded using pre-trained encoders namely codebooks such as local quantized patterns (LQP) [5] and learning-based (LE) descriptor [6]. To obtain a more discriminative representation, it is common to use a thicker codebook by increasing the number of codes. However, a thicker codebook is more vulnerable to quantization errors as the number of codes is increased.

To address the above challenges, we propose a novel encoding method based on multiple codebooks. A single codebook with many code words (a thick book) is not suitable for dividing a vector space

---

* Corresponding Author
Received: Nov. 3, 2017, Revised: Dec. 4, 2017, Accepted: Dec. 21, 2017

spanned by low-level feature vectors. This is because the vectors are extracted from an intra-domain as a face and are not sufficiently distinctive from each other. Accordingly, the single codebook causes quantization errors when a new low-level feature vector is encoded by the codebook. Rather than the single codebook, therefore, we employ multiple codebooks with fewer code words (several thin books) that are trained from randomly sampled sets of low-level feature vectors. Although thin codebooks may have less discriminative power, they are less vulnerable to quantization errors. Moreover, the collaboration between diverse codebooks enhances the discriminative power of the final representation by enabling codebooks to complement each other in increasing the redundant coverage between them. We experimentally demonstrate that the proposed multi-codebook-based encoding method provides more discriminative representation than a single codebook-based method and that our face representation achieves comparable performance on the labeled faces in the wild (LFW) dataset.

2. Codebook-based face representation

When a codebook is trained, the vector space is divided into a number of partitions. Each partition is represented by an independent code. At the encoding step, a discrete value of the closest label of the code is assigned to the low-level feature vector, which is extracted on each pixel as quantization when generating an encoded image. A histogram of the encoded image $I(x,y)$ is computed as follows.

$$H_i = \sum_{x,y} f\{I(x,y) = i\}, \; i = 0, \ldots, N-1, \tag{1}$$

where N is the number of total code words, and $f\{X\}$ is 1 if X is true; 0 otherwise. In a single-codebook-based representation, each discrete value is counted into the corresponding bin of the histogram. Therefore, different quantization results of the low-level feature vectors directly lead to different bin assignments, regardless of their distances in the vector space. For example, two low-level feature vectors are considered identical if they are assigned to the same code word. However, two vectors assigned to different code words are considered completely different even if they are very close in the vector space. This strict quantization rule increases the quantization errors, particularly when using a large number of codes.

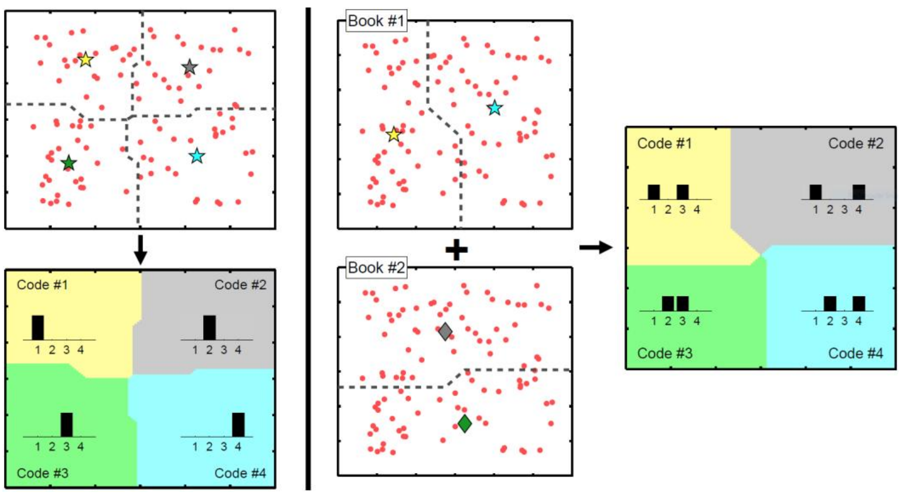

Figure 1. Single codebook (left) versus multiple codebooks (right)

3. Multi-codebook-based face representation

A single codebook comprises a thick book (e.g., 1 book of 256 code words), whereas multiple codebooks comprise several thin books (e.g., 4 books of 64 code words). In the encoding step, when generating M encoded images, multiple codebooks yield different quantization results. After the encoding step, a histogram of the encoded images $I_j(x, y)$ is computed as follows.

$$H_{i,j} = \sum_{x,y} f\{I_j(x, y) = i\},\ i = 0, \ldots, K-1,\ j = 0, \ldots, M-1, \tag{2}$$

where M is the number of codebooks, and K is the number of code words in each codebook. This can be additionally achieved by combining the separately computed histograms of each encoded image into a single histogram. When the histogram is computed, each codebook handles a different range of bins. Thus, multiple codebooks produce K^M different vector representations (1M for 4 books of 32 code words) for a low-level feature vector (i.e., $N = M \times K$), whereas a single codebook produces N different vector representations. By combining the multiple encoding results, more detailed vector representations are produced under more tolerant quantization rules than by using only one encoding result. Figure 1 shows the benefits of the multi-codebook-based representation. In a vector space partitioned by a single codebook, any two low-level feature vectors located in different partitions are considered different when they are encoded into a histogram; the distance between them is always one. Because the number of partitions in a vector space and the number of bins in a histogram are the same, each partition is matched to a corresponding bin. When a low-level feature vector is encoded into a histogram, only the corresponding bin of the histogram has a single response based on the location of the feature vector between the partitions. However, in a vector space partitioned by multiple codebooks, the codebooks have different partitioning boundaries; when they are combined, more combinations of partitions are generated. In the multi-codebook-based representations, if two low-level feature vectors belong to adjacent partitions (i.e., they share at least one partitioning boundary) in a vector space, the distance between them becomes less than one because they share at least one code in a specific codebook. Similarly, multiple codebooks produce more detailed representations under more tolerant quantization rules.

Multiple codebooks are designed to be thin (i.e., with a small number of code words) and diverse. Although the codebooks represented by a small number of code words are less discriminative in representing an entire vector space, they are less prone to quantization errors. Moreover, the combination of diverse codebooks enhances the final discriminative power in that they complement each other toward increasing the redundant coverage between the codebooks. The codebooks are generally trained using unsupervised subspace learning techniques such as the random projection tree and k-means clustering. We chose k-means clustering as our default algorithm for training the multiple codebooks because it is the simplest and the most intuitive algorithm. Although multiple codebooks based on the k-means clustering are described and evaluated in this study, multiple codebooks can be trained using any unsupervised subspace learning technique.

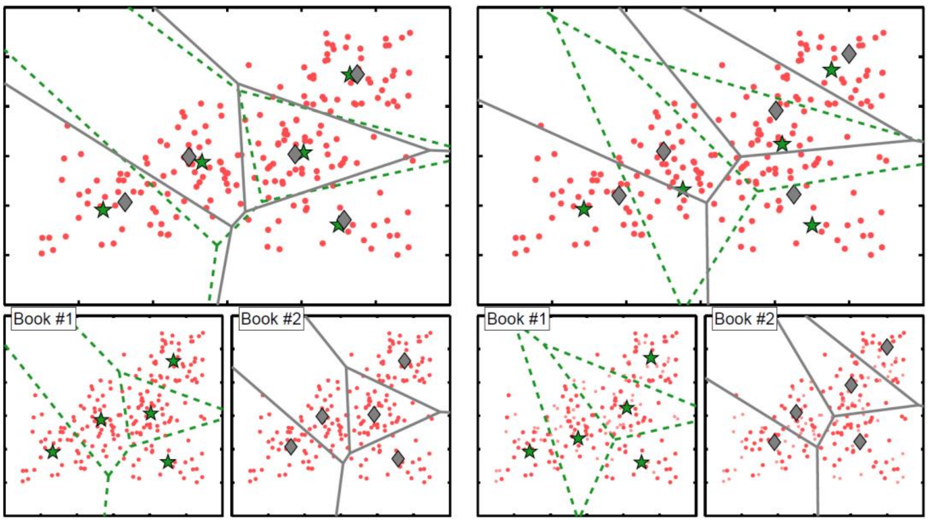

Figure 2. Multi-codebook learning by random seeds (left) versus bootstrapping (right)

To generate diverse codebooks, we exploit the benefits of the randomness in the learning stage. Some codebook-learning algorithms have randomness in their processes, yielding different code words between the codebooks whenever the algorithms are re-initialized for multiple codebooks. In addition, k-means clustering exhibits randomness by selecting random initial seeds when it is initialized. With M independent trials, different M codebooks are generated. However, as shown in Figure 2, the code words from different codebooks tend to be similar because the k-means clustering tends to place the code words around densely distributed training samples, even when the algorithm is re-initialized with different random initial seeds. The similarity of the code words reduces the redundant coverage among the codebooks and limits the advantages otherwise offered by multiple codebooks.

For better diversity, we introduce additional randomness in training the samples so that the codebooks are trained over different training sets. Thus, we exploit a bootstrap scheme, whereby M slightly different training sets are generated from a training set. In the bootstrap, N_B instances are randomly drawn with a replacement from a training set of size N_B. In this training set, some instances are drawn more than once; however, certain instances are not drawn at all because of the replacement of the sampling. The probability that a certain instance is not drawn after N_B draws is $(1-1/N_B)^{N_B} \approx e^{-1} = 0.368$. This implies that a codebook is trained for approximately 63.2% of the training samples; the samples vary depending on the codebooks. Bootstrapping is commonly used to train multiple base-learners from a training set. During the testing, the base learners are aggregated for better prediction results. This has been applied to both classification and regression problems. However, in our approach, the bootstrap is applied to the quantization problem wherein multiple codebooks are trained via bootstrapping; furthermore, they are aggregated for more tolerant quantization. To investigate how diverse codebooks are generated using the bootstrap scheme, we define the diversity of the codebooks as follows.

$$\sum_m \left(\sum_k D(c_k^m)\right),\ m = 0, \ldots, M-1,\ k = 0, \ldots, K-1, \tag{3}$$

where $D(C)$ is the distance between the code word C and its closest code word of the other books. This is reasonable for comparing the diversity of the two codebooks because the code words from more diverse codebooks tend to have higher $D(C)$. In our experiment, using (3), we compared the diversity of two types of multiple codebooks (with and without bootstrapping) by varying M and K from 2 to 8 and from 16 to 512, respectively. As expected, the codebooks trained with the bootstrapping were consistently more diverse than that without the bootstrapping under the condition that the two codebooks were trained with the same M and K. When $M = 4$ and $K = 64$, the bootstrapping helped in increasing the diversity by approximately 1.6 times.

4. Experimental results

We evaluate the proposed multiple codebook approach by comparing it to a single codebook through a face verification test in terms of the LFW benchmark [7, 8]. The LFW dataset provides four different versions of face images, including the original and three different types of images aligned according to different alignment methods. Of these methods, we used the aligned version of the LFW dataset [9] in all experiments. For a fair comparison, the same sampling pattern (a center with eight neighbors) was used to extract low-level feature vectors; the number of total code words N was fixed as 256. In the multiple-codebook learning process, the number of code words in each codebook K was determined using the number of codebooks M. These were trained using k-means clustering with a bootstrapping initialization with M = 2, 4, and 8. We applied principal component analysis (PCA) compression to the high-dimensional vector produced using each method to obtain a low-dimensional vector as the final representation. The mean classification accuracy over ten-fold cross-validation under different PCA dimensions was plotted for evaluation. The 256-code LBP was also tested for better comparisons.

Figure 3 shows the results. The codebook-based representation achieved better performance than the handcrafted-based one. The performance was improved by using multiple codebooks, and the proposed multi-codebook-based representation outperformed the single-codebook-based representation for every

PCA dimension. When $M = 4$, the multi-codebook-based representation yielded the best performance. These results demonstrate that the proposed multiple-codebook-based encoding method is useful for face representation.

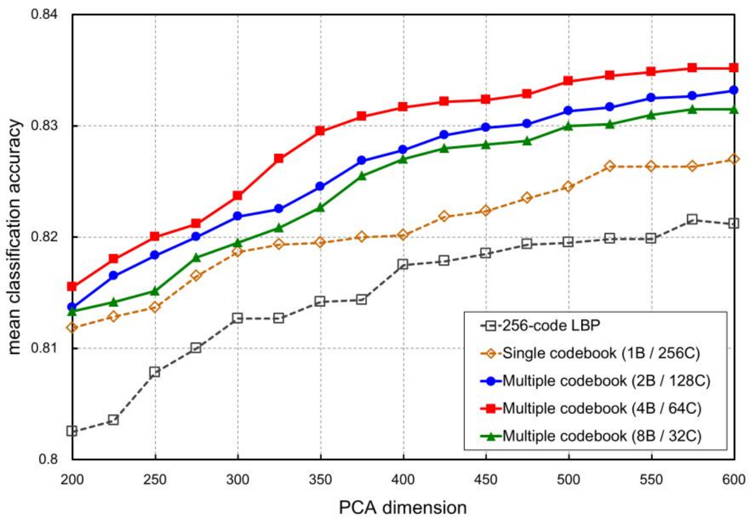

Figure 3. Evaluation of multiple codebooks

5. Conclusion

In this paper, we proposed a novel method for face representation based on multiple codebooks. In this method, diverse codebooks were trained and collaborated to encode the micro-structures of a given face. Our experimental results demonstrated that using diverse codebooks increases the final representation power and yields high performance in completing face-recognition tasks. We observed a trade-off between the codebook diversity and discriminative power. This suggests that we can further improve the face-recognition performance by learning multiple codebooks to achieve a sound trade-off between the codebook diversity and the discriminative power, which is part of our future work. We herein focused on low-level face representation; nevertheless, the proposed method can be easily applied to any face-recognition system and can be combined with previous methods to further improve the performance. We hope our work serves as an effective alternative approach for face representation.

6. References

[1] M. Everingham, J. Sivic, A. Zisserman, "Hello! My name is... Buffy–automatic naming of characters in TV video," in Proc. of the British Machine Vision Conference (BMVC), Edinburgh, United Kingdom, 2006, pp. 1–10.

[2] J. Sivic, M. Everingham, A. Zisserman, "Who are you?–Learning person specific classifiers from video," in Proc. of the IEEE International Conf. on Computer Vision and Pattern Recognition (CVPR), Miami, FL, 2009, pp. 1145–1152.

[3] J. Cui, F. Wen, R. Xiao, Y. Tian, X. Tang, "Easyalbum: an interactive photo annotation system based on face clustering and re-ranking," in Proc. of the ACM SIGCHI Conf. on Human Factors in Computing Systems, San Jose, CA, 2007, pp. 367–376.

[4] T. Ahonen, A. Hadid, M. Pietikainen, "Face description with local binary patterns: application to face recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence.,* Vol. 28, No. 12, pp. 2037–2041, Oct. 2006.

[5] S. U. Hussain, T. Napoleon, F, Jurie, "Face recognition using local quantized patterns," in Proc. of the British Machine Vision Conference (BMVC), Guildford, United Kingdom, 2012, pp. 1–11.

[6] Z. Cao, Q. Yin, X. Tang, J. Sun, “Face recognition with learning-based descriptor,” in Proc. of the IEEE International Conf. on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, 2010, pp. 2707–2714.
[7] G. B. Huang, H. Ramesh, T. Berg, E. Learned-Miller, “Labeled faces in the wild: a database for studying face recognition in unconstrained environments,” in *Dept. Comput. Sci., Univ. Massachusetts Amherst*, Amherst, MA, 2007, pp. 1–11.
[8] G. B. Huang, E. Learned-Miller, “Labeled faces in the wild: updates and new reporting procedures,” in *Dept. Comput. Sci., Univ. Massachusetts Amherst*, Amherst, MA, 2014, pp. 1–5.
[9] L. Wolf, T. Hassner, Y. Taigman, “Effective unconstrained face recognition by combining multiple descriptors and learned background statistics,” *IEEE Trans. Pattern Analysis and Machine Intelligence.,* Vol. 33, No. 10, pp. 1978–1990, Dec. 2011.