Visual information encoding for face recognition: sparse coding vs vector quantization

Dimche Kostadinov Sviatoslav Voloshynovskiy Maurits Diephuis
University of Geneva
Computer Science Department, Stochastic Information Processing Group
7 Route de Drize, Geneva, Switzerland
dimche.kostadinov@unige.ch svolos@unige.ch maurits.diephuis@unige.ch

Abstract

In this paper, we investigate the problem of visual information encoding for face recognition. We consider two models of information encoding based on sparse coding and vector quantization and compare their performance and computational complexity. The optimal solution is considered from the point of view of the best achievable classification accuracy by minimizing the probability of error under a given class of distortions. The results from the computer simulations confirm that our approach achieves similar performance with state-of-the-art sparse coding based image classification methods but with the considerably lower complexity.

1 Introduction

Visual information classification is of great practical interest in many multimedia and security applications. Traditionally, human face recognition is considered to be a reference application for testing different recognition frameworks. The main reasons for the interest in automatic human face recognition systems are the wide range of real world practical applications such as identification, verification, posture/gesture recognition, social network linking and multi-modal interaction.

In the past, Nearest Neighbour (NN) [2] and Nearest Feature Subspace (NFS) [7] have been used for classification. NN classifies the query image by only using its Nearest Neighbour. It utilizes the local structure of the training data and is therefore easily affected by noise. NFS approximates the query image by using all the images belonging to an identical class, using the linear structure of the data. Class prediction is achieved by selecting that class of images that minimizes the reconstruction error. NFS might fail in the case that classes are highly correlated to each other. Certain aspects of these problems can be overcome by Sparse Representation based Classification (SRC) [9]. However, on the other hand Qinfeng at al. [8] argue that the lack of sparsity in the data means that the compressive sensing approach cannot be guaranteed to recover the exact signal and therefore that sparse approximations may not deliver the desired robustness and performance. It has also been shown [1] that in some cases, the locality of the dictionary codewords is more essential than the sparsity. An extension of SRC, denoted Weighted Sparse Representation based Classification (WSRC) integrates the locality structure of the data into a sparse representation in a unified formulation.

In the most favourable case, when the training and observation models are known, one can design an optimal encoding/representation and a classifier that minimizes the classification error. However, in many applications the training and observation models are unknown or highly non-stationary and one only has a few training samples. In such a set-up, the recognition system basically learns the classifier in a "blind" way using only the available distorted training samples and expects that the observation model will exhibit similar behavior to the training model.

Most of the recent classification frameworks mainly rely on the discriminative nature of the sparse representation to perform classification. Accuracy not withstanding, it remains an open question whether or not this family of sparse methods attains the best trade-off for memory and computational complexity.

Therefore, considering the case in which the training and observation models are unknown, we focus on the problem of visual information encoding under prior ambiguity. In our formulation, the considered problem is closely related to both machine learning and coding. It should be pointed out that an alternative way of visual information encoding is based on the *bag-of-features* (BoF) approach. We will proceed with the generalized consideration of the BoF approach for multiple levels of multi-resolution image representation. Since the core of this representation is based on vector quantization we will refer to this approach as vector classification based recognition.

Practically, we consider and compare a type of face recognition system based principally on sparse coding and a type based on vector quantization. Both approaches are evaluated in terms of their classification accuracy for a certain range of distortions and in their computational and memory requirements.

This paper is organized as follows. Section 2 gives the basic problem formulation. In Section 3, we describe the sparse representation based recognition model, whereas the vector quantization method is introduced in Section 4. The results of the computer simulations for both methods are analysed in Section 5. Finally Section 6 concludes the paper.

Notation: We use capital bold letters to denote real valued matrices, $\mathbf{W} \in \Re^{M \mathrm{x} N}$, small bold letters to denote real valued vectors: $\mathbf{x} \in \Re^{M}$. We use sub and upper indexed vectors to denoted a single realization out of many from a given distribution e.g. $\mathbf{x}_i(m) \in \Re^{M}$, where m denotes the sample from some distribution. We denote an element of a vector as x. The estimate of $\mathbf{x}$ is denoted as $\hat{\mathbf{x}}$. All vectors have finite length, explicitly defined where appropriate.

2 Problem Formulation

The face recognition system consists of two stages: *enrolment* and *identification*.

At the enrollment stage, the photos from each subject are acquired and organized in the form of a codebook. We will assume that the recognition system should recognize K subjects. The photos of each subject i, $1 \leq i \leq K$, are acquired under different imaging conditions such as lighting, expression, pose, etc., which will represent the variability of face features and serve as intra-class statistics. We will also assume that the frontal face images are aligned to the same scale, rotation and translation using common computer vision features.

Thus each subject i is defined by $\mathbf{x}_i(m) \in \Re^{N}$ vectors representing a concatenation of aligned image columns with $1 \leq m \leq M$, where M represents the number of training images per subject that we assume to be the same for all subjects. The samples from all subjects are arranged into a codebook represented by a matrix:

$$\mathbf{W} = [\mathbf{x}_1(1), ..., \mathbf{x}_1(M), ..., \mathbf{x}_i(1), ..., \mathbf{x}_i(M), ..., \mathbf{x}_K(1), ..., \mathbf{x}_K(M)] \in \Re^{N \mathrm{x}(K*M)}. \quad (1)$$

At the recognition stage, a probe or query $\mathbf{y} \in \Re^{N}$ is presented to the system. The system should identify the subject i as accurate as possible based on $\mathbf{y}$ and $\mathbf{W}$. It is also assumed that $\mathbf{y}$ always corresponds to one of the subjects represented in the database. If it is not a case, a rejection option is integrated into the final decision.

3 Sparse Representation Based Recognition

In this section, face recognition is considered as a classification problem where the classifier should produce a decision in favour of some class i whose codebook codewords produce the most accurate approximation of the probe $\mathbf{y}$. One important class of approximations is represented by a *sparse linear approximation* [9], where the probe $\mathbf{y}$ is approximated by $\hat{\mathbf{y}}$ in the form of:

$$\hat{\mathbf{y}} = \mathbf{W}\boldsymbol{\alpha}\,, \tag{2}$$

where $\boldsymbol{\alpha} \in \Re^{M\text{x}K}$ is a sparse coding vector. The coding vector $\boldsymbol{\alpha}$ weights the codebook codewords gathered for all classes to favour the contribution of codewords corresponding to the correct class $\hat{i}$. The model of approximations can be represented as:

$$\mathbf{y} = \hat{\mathbf{y}} + \mathbf{r}\,, \tag{3}$$

where $\mathbf{r} \in \Re^N$ is the residual approximation error vector.

For each class i, let $\boldsymbol{\delta}_i : \Re^{K*M} \rightarrow \Re^{K*M}$ be a function that selects the coefficients associated with the ith class. For $\boldsymbol{\alpha} \in R^{K*N}$, $\boldsymbol{\delta}_i(\boldsymbol{\alpha})$ is a new vector whose only non-zero entries are the entries in $\boldsymbol{\alpha}$ that are associated with class i.

Then the probe $\mathbf{y}$ is classified based on the approximation $\hat{i}$ that minimizes the L_p-norm of the residual error vector between $\mathbf{y}$ and $\hat{\mathbf{y}}_i$:

$$\hat{i} = \arg\min_{1\leq i\leq K} \|\mathbf{W}\boldsymbol{\delta}_i(\boldsymbol{\alpha}) - \mathbf{y}\|_p. \tag{4}$$

Equation (4) corresponds to the minimum L_p distance classification, where for $p = 2$ one has the Euclidean distance and for $p = 1$ one has the Manhattan distance. A natural extension to (4) that might be considered is a bounded distance decoding (BBD) rule:

$$\hat{i} = \{i \in \{1, \cdots, M\} : \|\mathbf{W}\boldsymbol{\delta}_i(\boldsymbol{\alpha}) - \mathbf{y}\|_p \leq \eta N\}\,, \tag{5}$$

where $\eta \geq 0$. The BDD rule is useful when the classifier should reject probes that are unrelated to the database. In the general case, the BDD will produce a list of candidates that satisfy the above condition. To have only one unique $\hat{i}$ on the list, the parameter η should be chosen accordingly. Geometrically in the L_p space, it means that the L_p spheres with radius η around each approximate centroid for each class should not overlap, thus producing a unique classification.

The generalized solution of the approximation problem (2) under the constraint of sparsity of vector $\boldsymbol{\alpha}$ as a constrained optimization problem was considered in our previous work [5]:

$$\hat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\text{argmin}}(\boldsymbol{\phi}(\mathbf{W}\boldsymbol{\alpha} - \mathbf{y}) + \lambda\boldsymbol{\psi}(\mathbf{G}\boldsymbol{\alpha}))\,, \tag{6}$$

where $\boldsymbol{\phi}(.)$ is the penalty function corresponding to the prior distribution of the residual vector, $\boldsymbol{\psi}(.)$ is a regulizer corresponding to the prior distribution of the approximation coefficients $\boldsymbol{\alpha}$ and λ corresponds to the Lagrangian multiplier. The matrix $\mathbf{G}$ is the regularization matrix, where a simple selection of the regularizer $\mathbf{G}$ corresponds to the identity matrix $\mathbf{G} = \mathbf{I}$. A diagonal form of $\mathbf{G}$ might also be used to enforce linear locality constraints [6].

Note that this problem formulation, depending on the penalty function and the regularization term, considers the cases of hard and soft encoding (also global and local as in [5]).

4 Multilevel Vector Quantization based Recognition

In this section, we consider an alternative model of classification based on *multilevel vector quantization* (MVQ). The proposed approach has a certain similarity with BoF methods and convolutional deep learning neural networks (CNN). The image is partitioned on overlapping or non-overlapping blocks. The main idea behind the proposed method is to learn a codebook of centroids $\mathcal{C}_j^\ell = \{\mathbf{c}_{1,j}^\ell, \cdots, \mathbf{c}_{K_c,j}^\ell\}$ for each block j, $1 \leq j \leq B_\ell$, where B_ℓ is the number of blocks at each level of decomposition ℓ, $1 \leq \ell \leq L$ and K_c stands for the number of centroids chosen to be the same for all blocks and all levels. The different levels correspond to the different block sizes used for the image partitioning. The decomposition of images on local blocks of different sizes is explained by: (a) the necessity to cope with the non stationary nature of distortions that are approximated by stationary ones using local decompositions and (b) the multilevel decomposition should take the relationship between local coefficients into account, similar to CNN methods. The overall goal of the proposed method is to achieve a competitive classification accuracy together with an acceptable memory storage and complexity.

The MVQ based classification consists of three main steps: (a) codebook generation, (b) block encoding using the basis vectors of the generated codebook and (c) classification.

4.1 Codebook generation

Given the training data $\mathbf{x}_i(m)$ for all subjects $1 \leq i \leq K$ with $1 \leq m \leq M$ training samples per subject, each image is partitioned on B_ℓ blocks of size 2×2 and 3×3 corresponding to the levels of decomposition $\ell \in \{1, 2\}$, respectively. Therefore, each block of a training image $\mathbf{x}_i(m)$ is denoted as $\mathbf{x}_{i,j}^\ell(m)$. The trained codebook for each block j at the level ℓ consists of a set of K_c centroids $\mathcal{C}_j^\ell = \{\mathbf{c}_{1,j}^\ell, \cdots, \mathbf{c}_{K_c,j}^\ell\}$, learned with the k-means algorithm.

4.2 Encoding

Given a set of training samples $\mathbf{x}_i(m)$ for all subjects $1 \leq i \leq K$ with $1 \leq m \leq M$, represented with a multilevel block decomposition, each block is assigned to the nearest centroids using a k-NN or ϵ-NN strategy (bounded distance decoding) *, represented by the list:

$$\mathcal{L}(\mathbf{x}_{i,j}^\ell(m)) = \{w \in \{1, \cdots, K_c\} : d(\mathbf{x}_{i,j}^\ell(m), \mathbf{c}_{w,j}^\ell) \leq \epsilon L_\ell\}, \tag{7}$$

where $1 \leq w \leq K_c$, L_ℓ is the block total size at level ℓ and $\epsilon \geq 0$.

The encoding results in the generation of an encoding vector: $\mathbf{D}_{\mathbf{x}}{}_{i,j,w}^{\ell}(m) = (d_{x}{}_{i,j,1}^{\ell}(m), \cdots, d_{x}{}_{i,j,K_c}^{\ell}(m)) \in \{0,1\}^{L \times K \times B_\ell \times K_c \times M}$, with $d_{x}{}_{i,j,w}^{\ell}(m) = 1$ for $w \in \mathcal{L}(\mathbf{x}_{i,j}^\ell(m))$.

The final stage of encoding includes the pooling of results from all training samples to the final index that is accomplished based on max-pooling (MAXP) or alternatively average-pooling (AVGP):

$$\text{MAXP: } d_{x}{}_{i,j,w}^{\ell} = \max_{1 \leq m \leq M} d_{x}{}_{i,j,w}^{\ell}(m), \text{ AVGP: } d_{x}{}_{i,j,w}^{\ell} = \sum_{m=1}^{M} d_{x}{}_{i,j,w}^{\ell}(m). \tag{8}$$

*Due to the space limitation, we will proceed with the ϵ-NN only.

The main idea behind this particular form of max pooling is to capture all centroids for a given block and level of decomposition that might represent a subject under various observation distortions. In fact, if the observation model were stationary and known, the representative centroids could be computed analytically.

The final stage of encoding includes the generation of an inverted file look up table where each block $1 \leq j \leq B_\ell$ has a set of centroids centroid $w \in \{1, \cdots, K_c\}$ at the level $\ell \in \{1, 2\}$ containing the indices of corresponding subjects $i \in \{1, \cdots, K\}$. This look up table is very sparse and efficient for memory storage.

4.3 Classification

The classification consists of two stages: (a) block decoding and (b) final fusion.

4.3.1 Block decoding

The goal of block decoding is to find the ϵ-NN centroids corresponding to each block of observation image $\mathbf{y}$:

$$\mathcal{L}(\mathbf{y}_j^\ell) = \{w \in \{1, \cdots, K_c\} : d(\mathbf{y}_j^\ell, \mathbf{c}_{w,j}^\ell) \leq \epsilon L_\ell\}, \tag{9}$$

where $1 \leq w \leq K_c$, L_ℓ is the block total size at level ℓ and $\epsilon \geq 0$.

The block decoding results in the generation of activation vector:

$$\mathbf{D}_{\mathbf{y}_{j,w}}^{\ell} = (d_{y_{j,1}}^{\ell}, \cdots, d_{y_{j,K_c}}^{\ell}). \tag{10}$$

It is important to note that the activation vector might be considered as *hard decoding*, when its elements are assigned 0 or 1, if the above condition is satisfied, or *soft decoding*, when its element correspond to the reliability or likelihood of observing some centroid $\mathbf{c}_{w,j}^\ell$ given the block $\mathbf{y}_j^\ell$, for $w \in \mathcal{L}(\mathbf{x}_{i,j}^\ell(m))$. In the case of soft decoding, the reliable centroids will obtain weights closer to 1 and non-reliable closer to 0. It is also remarkable that in the case of reliable decoding, the list $\mathcal{L}(\mathbf{y}_j^\ell)$ will be sparse indicating that the reliable centroid(s) is(are) found. Otherwise, all elements of this list will have identical weights. Therefore, one can use the notion of sparsity to estimate the reliability of the produced estimate. We refer the interested readers to [4] for more details.

4.3.2 Final fusions

The final decision can be produced at each level of decomposition ℓ that would correspond to more conservative recognition architectures or it can be obtained as a result of fusion from several levels.

Therefore, each block of the observation image $\mathbf{y}$ at each level ℓ produces the lists of image indices that are the most likely candidates for the corresponding blocks. Thus, the final decision is just a result of the most likely index $i \in \{1, \cdots, K\}$ selection that obtains the majority of votes. It should also be pointed out that each decision can be produced as a result of the largest similarity between the observation vector $d_{x_{i,j,w}}^{\ell}$ and $d_{y_{j,w}}^{\ell}$ that is estimated as:

$$\hat{i}^\ell = \max_{i \in \{1, \cdots, K\}} t_i^\ell, \text{ where: } t_i^\ell = \sum_{j=1}^{B_\ell} \sum_{w=1}^{K_c} d_{x_{i,j,w}}^{\ell} d_{y_{j,w}}^{\ell}. \tag{11}$$

The decision at the global level is produced as:

$$\hat{i} = \max_{i \in \{1,\cdots,K\}} t_i, \text{ where: } t_i = \sum_{\ell=1}^{L} \sum_{j=1}^{B_\ell} \sum_{w=1}^{K_c} d_{x_{i,j,w}}^{\ell} d_{y_{j,w}}^{\ell}. \tag{12}$$

5 Computer Simulations

In this section we present the results of the computer simulation. We compare the results of the accuracy using several sparse based representation classification models [5] versus MVQ with a list decoding model. In addition we indicate their computational complexity by measuring the average execution time for a single recognition.

The computer simulation is carried out on the publicly available Extended Yale B database for face recognition [3]. We use all images from this database cropped and normalized to 192x168 pixels. In our set up, the images from the dataset are rescaled to 10x12 pixels using nearest neighbour interpolation. In all of the computer simulations we use raw, basic, elementary image pixel values (block of image pixel values) as features. To be unbiased in our validation of the results we use 5-fold cross validation, where for a single validation for each subject, half of the images are selected at random for training and the remainder for testing.

All of the MVQ models for block j at the level ℓ use trained codebooks that consists of a set of K_c centroids $\mathcal{C}_j^\ell = \{\mathbf{c}_{1,j}^\ell, \cdots, \mathbf{c}_{K_c,j}^\ell\}$, learned with the k-means algorithm. The number of centroids at any level ℓ for any block j is 512.

Figure 1 shows the resulting accuracy of the MVQ method using one layer independent, overlapping 2×2, 3×3 and two layer joint, overlapping 2×2 and 3×3 blocks with hard and soft decoding, employing bounded distance decoding with different ϵ values. The parameter ϵ in equations (7) and (9) is chosen adaptively for each block based on the sparsity level ε as defined in [4].

Table 1 summarizes the best achievable accuracy of classification at different levels for the hard and soft decoding.

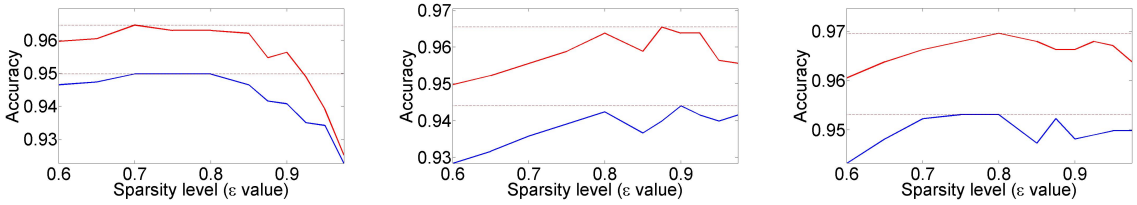
	2×2	3×3	Fused
Hard encoding	0.94	0.94	0.95
Soft encoding	0.96	0.97	0.97

Table 1: MVQ recognition accuracy.

Figure 2a gives a comparison of the accuracy of all methods deployed while Figure 2b shows the computation time needed for classifying a single image query.

In conclusion, the accuracy of MVQ based recognition using 2×2 and 3×3 overlapping blocks is 0.97 which is on par with the best sparse coding based recognition method denoted as " $*$ ", that is, the sparse approximation method that uses the L_1 norm as a penalty function and the L_1 norm as regulizer applied on overlapping blocks from [5].

It is also noteworthy that the MVQ method is between a factor 200 faster in recognition than competing sparse methods.

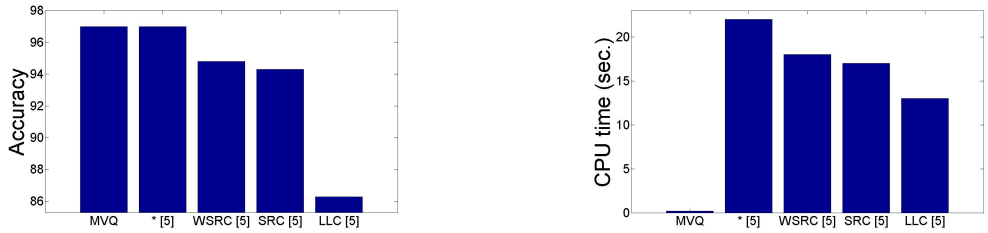


(a) MVQ using blocks of size 2×2

(b) MVQ using blocks of size 3×3

(c) MVQ using blocks of size 2×2 and 3×3

Figure 1: Accuracy of MVQ recognition with one layer independent, overlapping blocks of size 2×2, 3×3 and two layer joint, overlapping blocks of size 2×2 and 3×3 with hard and soft decoding (red and blue line respectively), employing bounded distance decoding with different ε values, computed for each block j.



(a) Comparison of the accuracy using different models

(b) The CPU running time using different models for a PC with an Intel Xeon CPU E5-16200 @ 3.6GH and 32GB RAM memory.

Figure 2: Comparison considering the best accuracy and computation time for the MVQ method (using overlapping blocks with sizes 2×2 and 3×3) and the methods *, WSRC, SRC and LLC

6 Future work

Future research will consider explore geometrically invariant coding strategies, and further link the MVQ framework with conventional convolutional neural networks. Furthermore, we will strive to optimize the encoding and list decoding strategies by incorporating reliability statistics.

7 Conclusions

In this paper we considered the face recognition problem from a both machine learning and information coding perspective, adopting an alternative way of visual information encoding. Our model of classification is based on *multilevel vector quantization* (MVQ), conceptually similar to BoF and CNN. The results from the computer simulations confirm that the MVQ based recognition model achieves an accuracy that is comparable to state-of-the-art sparse coding based image classification methods[5]. In addition the complexity in terms of processing time and memory of the MVQ model is significantly lower compared to other state-of-the-art methods based on sparse coding.

Acknowledgement

The work presented in this paper was supported by a grant from Switzerland through the Swiss Contribution to the enlarged European Union (PSPB-125/2010).

References

[1] Coates, A., Ng, A.Y.: The importance of encoding versus training with sparse coding and vector quantization. In: Proceedings of the 28th International Conference on Machine Learning (ICML-11). (2011) 921–928

[2] Cover, T., Hart, P.: Nearest neighbor pattern classification. Information Theory, IEEE Transactions on **13**(1) (1967) 21–27

[3] Georghiades, A.S., Belhumeur, P.N., Kriegman, D.J.: From few to many: Illumination cone models for face recognition under variable lighting and pose. IEEE Transactions on Pattern Analysis and Machine Intelligence **23** (2001) 643–660

[4] Hoyer, P.O., Dayan, P.: Non-negative matrix factorization with sparseness constraints. Journal of Machine Learning Research **5** (2004) 1457–1469

[5] Kostadinov, D., Voloshynovskiy, S., Ferdowsi, S.: Robust human face recognition based on locality preserving sparse over complete block approximation. In: Proceedings of SPIE Photonics West, Electronic Imaging, Media Forensics and Security V, San Francisco, USA (January, 23 2014)

[6] Lu, C.Y., Min, H., Gui, J., Zhu, L., Lei, Y.K.: Face recognition via weighted sparse representation. J. Vis. Comun. Image Represent. **24**(2) (February 2013) 111–116

[7] Shan, S., Gao, W., Zhao, D.: Face recognition based on face-specific subspace. International journal of imaging systems and technology **13**(1) (2003) 23–32

[8] Shi, Q., Eriksson, A., van den Hengel, A., Shen, C.: Is face recognition really a compressive sensing problem? 2013 IEEE Conference on Computer Vision and Pattern Recognition **0** (2011) 553–560

[9] Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust face recognition via sparse representation. Pattern Analysis and Machine Intelligence, IEEE Transactions on **31**(2) (2009) 210–227