A Hybrid Face Recognition Approach Using GPUMLib

Noel Lopes[1,2] and Bernardete Ribeiro[1]

[1] CISUC - Center for Informatics and Systems of University of Coimbra, Portugal
[2] UDI/IPG - Research Unit, Polytechnic Institute of Guarda, Portugal
noel@ipg.pt, bribeiro@dei.uc.pt

Abstract. We present a hybrid face recognition approach which relies on a Graphics Processing Unit (GPU) Machine Learning (ML) Library (GPUMLib). The library includes a high-performance implementation of the Non-Negative Matrix Factorization (NMF) and the Multiple Back-Propagation (MBP) algorithms. Both algorithms are combined in order to obtain a reliable face recognition classifier. The proposed approach first applies an histogram equalization to the original face images in order to reduce the influence from the surrounding illumination. The NMF algorithm is then applied to reduce the data dimensionality, while preserving the information of the most relevant features. The obtained decomposition is further used to rebuild accurate approximations of the original data (by using additive combinations of the parts-based matrix). Finally, the MBP algorithm is used to build a neural classifier with great potential to construct a generalized solution. Our approach is tested in the Yale face database, yielding an accuracy of 93.33% thus demonstrating its potential. Moreover, the speedups obtained with the GPU greatly enhance real-time implementation face recognition systems.

Keywords: GPU computing, Non-Negative Matrix Factorization, Multiple Back-Propagation, Hybrid systems, Face Recognition.

1 Introduction

Face recognition has many potential applications in various distinct areas, such as military, law-enforcement, anti-terrorism, commercial and human-computer interaction [8]. Typically, solving this problem involves several phases: (*i*) segmentation of the faces, (*ii*) extraction of relevant features from the face regions, (*iii*) recognition and (*iv*) verification [10]. In this paper, we concentrate on the last phases, leaving out the segmentation phase. Over the past decades, face recognition has become an increasingly important area of research, attracting researchers from pattern recognition, neural networks, image processing, computer vision, Machine Learning (ML) and psychology among others [8,10]. However, this is still a very challenging and complex problem, because the appearance of the individuals is affected by a numerous factors (e.g. illumination conditions, facial expressions, usage of glasses) and current systems are still no match for

I. Bloch and R.M. Cesar, Jr. (Eds.): CIARP 2010, LNCS 6419, pp. 96–103, 2010.

the human perception system [10]. A detailed survey on existing techniques and methods for face recognition can be found in Zhao et al. [10].

In this paper, we propose a hybrid method for the face recognition problem, relying on a Graphics Processing Unit (GPU) Machine Learning Library (GPUMLib) high-performance implementation of Non-Negative Matrix Factorization (NMF) and Multiple Back-Propagation (MBP) algorithms.

NMF is an unsupervised technique for discovering a parts-based representation of objects [11]. Essentially, it decomposes a matrix, containing only non-negative coefficients, into the product of two matrices (also containing non-negative coefficients), usually with reduced ranks, thus reducing the number of characteristics in the database, while preserving the relevant information that allows for the reconstruction of the original data. Since negative coefficients are not allowed, the original data is reconstructed through additive combinations of the parts-based factorized matrix representation, which is consistent with psychological and physiological evidence for parts-based representations in the brain [2]. On the other hand, MBP is a neural networks supervised algorithm, being able to offer potentially greater generalization capabilities than the well known Back-Propagation (BP) algorithm [3]. Over time, neural networks have proven to solve complex problems in many different domains (e.g. pattern recognition, image processing, intelligent control and time series prediction systems) [7]. By combining both algorithms, we are able to take advantage of the superior generalization capabilities of neural networks, while retaining the possibility of having a parts-based representation of the facial images provided by the NMF algorithm.

The remainder of this paper is organized as follows: The next section covers the GPUMLib implementation of the NMF and MBP algorithms. Section 3 details the proposed hybrid method for face recognition. Section 4 analyses the results obtained with the referred method for the Yale face database. Finally, in section 5 conclusions and future work are addressed.

2 GPU Machine Learning Library

GPUMLib is an open source ML library, aiming to provide machine learning researchers and practitioners with a high-performance library by taking advantage of the GPU enormous computational power [5]. Currently, the GPUMLib fully implements the BP, MBP and NMF algorithms. Furthermore an implementation of the Radial Basis Functions (RBF) neural networks is being developed and the implementation of other ML algorithms is being planned. The library is released under the GNU General Public License and its source code, documentation and examples can be obtained at `http://gpumlib.sourceforge.net/`.

2.1 Non-Negative Matrix Factorization

Given a matrix $V \in \mathbb{R}_{+}^{n \times m}$ containing only non-negative coefficients ($V_{ij} \geq 0$) and a pre-specified positive integer, $r < min(n, m)$, NMF finds two matrices, with non-negative coefficients, $W \in \mathbb{R}_+^{n\times r}$ and $H \in \mathbb{R}_+^{r\times m}$ whose product approximates V (as closely as possible):

$$V \approx WH \ . \tag{1}$$

The value of r is generally chosen to satisfy $(n+m)r < nm$, so that the approximation WH can be viewed as a compressed form of the original data [9].

The non-negativity constrains imposed to the elements of W and H are compatible with the intuitive notion of combining parts to form a whole, which is how NMF learns a parts-based representation [11]. In our case, each column of V represents a human face. Thus, the basis elements of W, may contain facial features, such as eyes, noses and lips [1].

In order to measure the quality of the approximation defined in (1) the Euclidean distance can be used:

$$\|V - WH\|^2 = \sum_{ij} (V_{ij} - (WH)_{ij})^2 \ . \tag{2}$$

Minimizing (2) subject to the constrains $W_{ij} \geq 0$ and $H_{ij} \geq 0$ leads to an optimization problem that can be solved by iteratively applying (3) and (4) until a good approximation is found:

$$H_{a\mu} \leftarrow H_{a\mu} \frac{(W^T V)_{a\mu}}{(W^T WH)_{a\mu}} \ , \tag{3}$$

$$W_{ia} \leftarrow W_{ia} \frac{(VH^T)_{ia}}{(WHH^T)_{ia}} \ . \tag{4}$$

To determine the speedups granted by the GPU parallel implementation of the NMF algorithms, the face database #1 from the MIT Center for Biological & Computational Learning, available at `http://cbcl.mit.edu/cbcl/software-datasets/FaceData2.html`, was used. This database includes a total of 2429 face images of $19 \times 19 = 361$ pixels. Thus, matrix V is composed by 361 rows and 2429 columns. Figure 1 presents the speedups yielded by an NVIDIA GTX 280 GPU (relatively to a Core 2 Quad Q9300 2.5 GHz CPU), considering the Euclidian distance metric and the multiplicative update rule described earlier (according to the value of r chosen) [6].

2.2 Multiple Back-Propagation

MBP is a generalization of the BP algorithm that can be used to train Multiple Feed-Forward (MFF) networks. Together, MFF networks and the MBP algorithm form an architecture that is in most cases preferable to Feed-Forward (FF) networks trained with the BP algorithm [3]. MFF networks are obtained by integrating two FF networks: a main network and a space network. The main network contains selective activation neurons that differ from standard ones by an importance factor. This parameter is adjusted according to the pattern presented to the

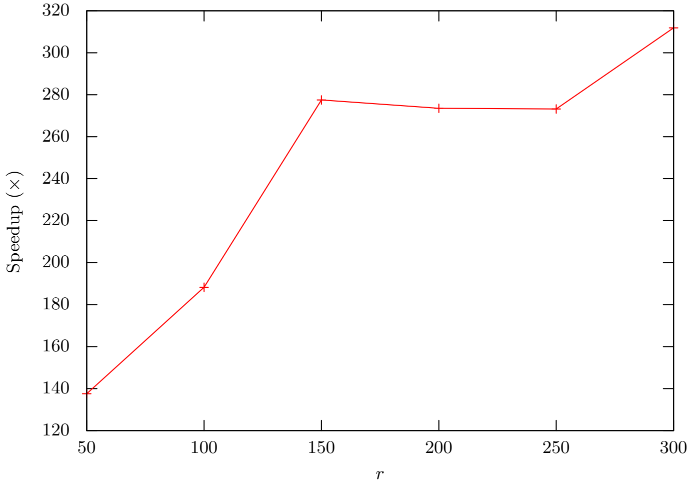

Fig. 1. Speedups provided by the GPU relatively to the CPU, for the NMF algorithm

network, enabling fine-tuning in a given set of patterns. In other words, it is responsible by the network response to specific *stimulus* while ignoring the rest. The importance factors are determined by the space network, which receives the same inputs as the main network. Both networks will thus function in a collaborative manner and must be trained together [3].

The MBP algorithm implementation in GPUMLib[1] has been tested with several benchmarks and real-world applications demonstrating the impressive speedups (up to 180×) that in some problems are responsible for reducing the running time from weeks to hours [4].

3 Face Recognition Based on a Hybrid Classifier

The proposed method consists of four steps. In the first step the facial images are pre-processed in order to reduce the influence of the different ambient illumination conditions. In the next step the parts-based representation of the faces dataset are obtained with the NMF algorithm. In the following step, using this information, a classifier is trained with the MBP algorithm. Finally, its performance is evaluated on the test data to validate the designed classifier. Eventually, the previous steps can be repeated with different configurations, until a classifier that meets the goals expectations is found.

3.1 Image Pre-processing

In order to reduce the influence of the surrounding illumination, an histogram equalization can be applied to the face images. This method improves the contrast

[1] The Multiple Back-Propagation software, available at `http://dit.ipg.pt/MBP`, shares the same GPU implementation and has been extensively tested and widely used by neural networks researchers and practitioners.

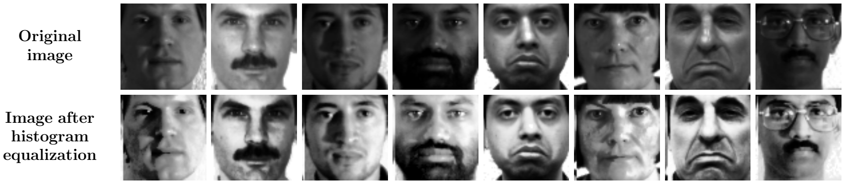

Fig. 2. Yale face database images before and after the histogram equalization

of the images by changing its gray levels [11]. Figure 2 depicts some of the Yale face database images before and after the histogram equalization has been applied.

3.2 Parts-Based Representation of the Images Yale Data Base

Once the contrast of the images is enhanced, the NMF algorithm is then applied to the training database. As said before, this step is used in order to reduce the data dimensionality and to gather the main characteristics of the individuals faces. Thus, we build the $V \in \mathbb{R}_+^{n\times m}$ matrix by placing one image in each column of the matrix. Therefore the number of rows will be equal to the number of pixels of the images. As a result we will end up with the matrix $W \in \mathbb{R}_+^{n\times r}$ containing r parts-based faces representations and the matrix $H_{train} \in \mathbb{R}_+^{r\times m}$ containing the respective codification that must be added in order to obtain the correct approximations of the original m images. For the test database, the process is similar. First we must build a $V_{test} \in \mathbb{R}_+^{n\times m'}$ matrix. Then the NMF algorithm is applied to the new matrix. However, this time the parts-based matrix, W, must remain invariable. Thus, only the matrix $H_{test} \in \mathbb{R}_+^{r\times m'}$ gets updated. Once this process is completed, matrix H_{test} will contain the codification of the parts-based images that must be added in order to obtain the approximations of the m' images in the test database. Figure 3 shows the sequence of steps needed to obtain the desired matrices.

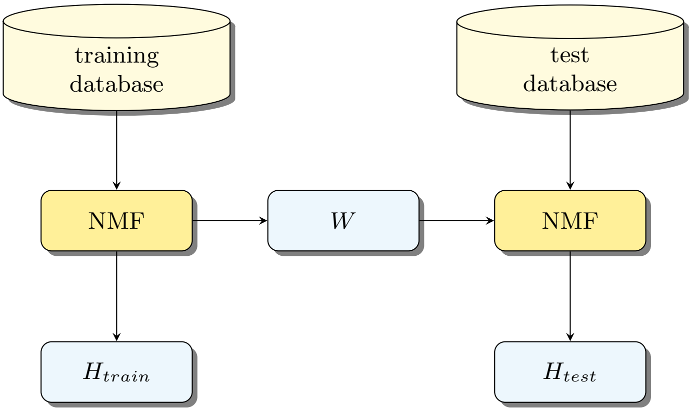

Fig. 3. Applying the NMF algorithm to the training and test databases

3.3 Building and Evaluating the Classifier

As soon as the matrices are computed, the H_{train} matrix can be used to train a MFF network with the MBP algorithm. The quality of the resulting neural network can then be asserted by using H_{test} matrix.

Every time new data is gathered to be used by the classifier, a new H matrix (containing the codification of the parts-based images that approximate the new data) needs to be created by using the NMF algorithm. Although only the H matrix needs to be computed as the parts-based matrix W remains constant, the time consumed in the process can prevent this method from being used in real-world applications. Thus the GPU implementation of the NMF algorithm is crucial for this method to be used in real-world scenarios.

4 Results and Discussion

To test the proposed method we used the Yale face database, consisting of 165 gray-scale face images (with 64 × 64 pixels) of 15 individuals. Each individual appears in 11 images, representing different facial expressions (happy, normal, sad, sleepy, surprised, wink) or configurations (center-light, left-light, right-light, with glasses, without glasses). In order to build the training dataset we randomly choose eight images of each person. The remainder three images were placed on the test dataset. Thus, the training dataset contains 120 images and the test dataset 45. An histogram equalization was applied to all the images.

Thereafter, the NMF algorithm was applied to the training dataset in order to determine the parts-based matrix, W, representation of the faces and the matrix H_{train} that will later be used to create (train) a classifier. The number of parts-based images (r) was chosen to be 45, so that each individual can potentially have three part-based images. In practice, because NMF is an unsupervised algorithm there is no guarantee that each individual will have three parts-based images associated or that the parts-based images will not end up being sharing by several individuals. Figure 4 shows the first 40 images of W. Once the matrix W was computed, we used the NMF algorithm again, this time on the test dataset in order to obtain the H_{test} matrix, necessary to assert the quality of the resulting neural network classifiers.

Finally, in order to train the neural networks, we developed a preliminary Autonomous Training System (ATS) that actively searches for better solutions, by adjusting the topology of the networks. The ATS makes use of the GPUMLib parallel implementation of the MBP algorithm. Essentially, after training a neural network the ATS evaluates its performance and compares it with the best performance achieved so far. The results of the performance comparison are then used to determine the number of hidden neurons of a new neural network and the process is repeated until a predefined specified stopping criteria is meet. For the problem at hand, the ATS took less than 16 hours to train a total of 100000 MFF networks. Figure 5 shows the number of networks trained by the ATS according to number of hidden neurons. The best network (with 12 hidden neurons) presents an accuracy of 93.33% on the test dataset and of 100% on

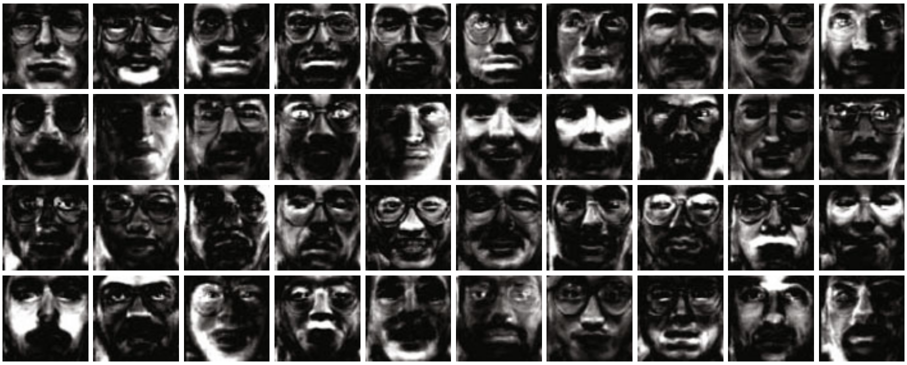

Fig. 4. Parts-based faces representations (W)

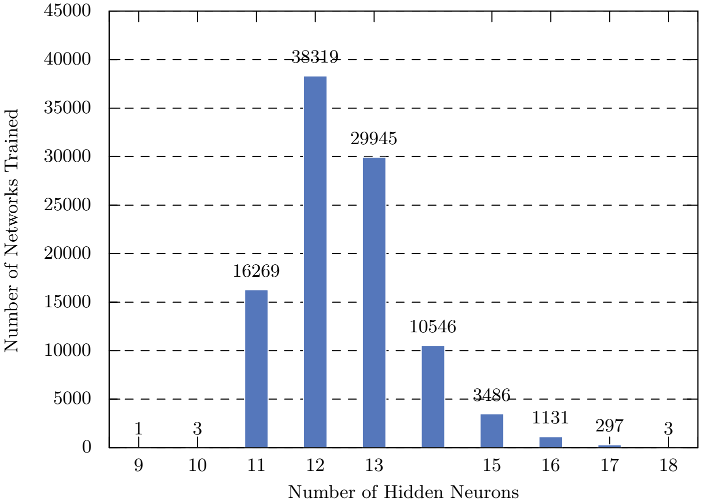

Fig. 5. Number of networks trained by the ATS, according to the number of neurons

the training dataset. Only three images (of different persons) on the test dataset were misclassified and among those one had 46.11% probability of belonging to the correct individual. Thus, the results obtained validate the proposed method, demonstrating its potential. Moreover the ATS approach is very promissing, since it was able to find high-quality solutions without any human-intervention (aside from the initial configuration).

5 Conclusions and Future Work

A hybrid method, combining the NMF and MBP algorithms, for for face recognition is proposed. The proposed approach was tested in the Yale face database, yielding an accuracy of 93.33%, thus demonstrating its viability and power. However a key factor for its success in real-world applications, may rely on the use of the GPU to support for high-performance implentations of the NMF algorithm, because when presenting new data to the classifier it is necessary to calculate the matrix containing the codification of the parts-based images that approximate the new data. Nevertheless, GPUMLib already presents a GPU implementation of both the NMF and MBP algorithms, making the proposed method very attractive for use in real-world scenarios. Future work will address determining the impact of changing the number of parts-based images (r) in the quality of the resulting solutions. Moreover, another line of work consists in comparing the proposed method with other reported methods in the literature.

References

1. Gillis, N., Glineur, F.: Using underapproximations for sparse nonnegative matrix factorization. Pattern Recogntion 43(4), 1676–1687 (2010)
2. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature 401, 788–791 (1999)
3. Lopes, N., Ribeiro, B.: An efficient gradient-based learning algorithm applied to neural networks with selective actuation neurons. Neural, Parallel & Scientific Computations 11(3), 253–272 (2003)
4. Lopes, N., Ribeiro, B.: GPU implementation of the multiple back-propagation algorithm. In: Corchado, E., Yin, H. (eds.) IDEAL 2009. LNCS, vol. 5788, pp. 449–456. Springer, Heidelberg (2009)
5. Lopes, N., Ribeiro, B.: GPUMLib: a new library to combine machine learning algorithms with graphics processing units. In: 10th International Conference on Hybrid Intelligent Systems, Atlanta, USA (2010)
6. Lopes, N., Ribeiro, B.: Non-negative matrix factorization implementation using graphic processing units. In: Fyfe, C., Tino, P., Charles, D., Garcia-Osorio, C., Yin, H. (eds.) IDEAL 2010. LNCS, vol. 6283, pp. 275–283. Springer, Heidelberg (2010)
7. Tang, H., Tan, K.C., Yi, Z.: Neural Networks: Computational Models and Applications. Studies in Computational Intelligence. Springer, New York (2007)
8. Wang, J., Zhang, B., Wang, S., Qi, M., Kong, J.: An adaptively weighted subpattern locality preserving projection for face recognition. Journal of Network and Computer Applications 33(3), 323–332 (2010)
9. Xu, B., Lu, J., Huang, G.: A constrained non-negative matrix factorization in information retrieval. In: IEEE International Conference on Information Reuse and Integration, IRI 2003, pp. 273–277 (2003)
10. Zhao, W., Chellappa, R., Phillips, P.J., Rosenfeld, A.: Face recognition: A literature survey. ACM Computing Surveys 35(4), 399–458 (2003)
11. Zilu, Y., Guoyi, Z.: Facial expression recognition based on NMF and SVM. International Forum on Information Technology and Applications 3, 612–615 (2009)