

Research Article

Nonnegative Matrix Factorizations Performing Object Detection and Localization

G. Casalino,¹ N. Del Buono,² and M. Minervini³

¹ *Dipartimento di Informatica, Università di Bari, Via E. Orabona 4, I-70125 Bari, Italy*

² *Dipartimento di Matematica, Università di Bari, Via E. Orabona 4, I-70125 Bari, Italy*

³ *Computer Science and Engineering Ph.D Division, Institute for Advanced Studies Lucca (IMT),
Piazza S. Ponziano 6, 55100 Lucca, Italy*

Correspondence should be addressed to N. Del Buono, delbuono@dm.uniba.it

Received 18 October 2011; Revised 3 March 2012; Accepted 16 March 2012

Academic Editor: Cezary Z. Janikow

Copyright © 2012 G. Casalino et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We study the problem of detecting and localizing objects in still, gray-scale images making use of the part-based representation provided by nonnegative matrix factorizations. Nonnegative matrix factorization represents an emerging example of subspace methods, which is able to extract interpretable parts from a set of template image objects and then to additively use them for describing individual objects. In this paper, we present a prototype system based on some nonnegative factorization algorithms, which differ in the additional properties added to the nonnegative representation of data, in order to investigate if any additional constraint produces better results in general object detection via nonnegative matrix factorizations.

1. Introduction

The notion of low dimensional approximation has played a fundamental role in effectively and efficiently processing and conceptualizing huge amount of data stored in large sparse matrices. Particularly, subspace techniques, such as Singular Value Decomposition [1], Principal Component Analysis (PCA) [2], and Independent Component Analysis [3], represent a class of linear algebra methods largely adopted to analyze high dimensional dataset in order to discover latent structures by projecting it onto a low dimensional space. Generally, a subspace method is characterized by learning a set of base vectors from a set of suitable data templates. This vector spans a subspace which is able to capture the essential structure of the input data. Once the subspace has been found (during the off-line learning phase), the detection of a new sample can be accomplished (in the so-called on-line detection phase) by projecting it on the subspace and finding the nearest neighbor of templates projected onto this subspace. These methods have found efficient applications in several areas of information retrieval, computer vision, and pattern recognition, especially in the

fields of face identification [4, 5], recognition of digits and characters [6, 7], and molecular pattern discovery [8, 9].

However, pertinent information stored in many data matrices are often nonnegative (examples are pixels in images, the probability of a particular topic appearing in a linguistic document, the amount of pollutant emitted by a factory, and so on [10–15]). During the analysis process, taking into account this nonnegativity constraint could bring some benefits in terms of interpretability and visualization of large scale data, while maintaining the physical feasibility more closely. Nevertheless, classical subspace methods describe data as a combination of elementary features involving both additive and subtractive components; hence, they are not able to guarantee the conservation of nonnegativity.

The recent approach of low-rank nonnegative matrix factorization (NMF) becomes particularly attractive to obtain a reduced representation of data by using additive components only. This idea has been motivated in a couple of ways. Firstly in many applications (e.g., by the rules of physics) one knows that the quantities involved cannot be negative. Secondly, nonnegativity has been argued for based on the intuition that

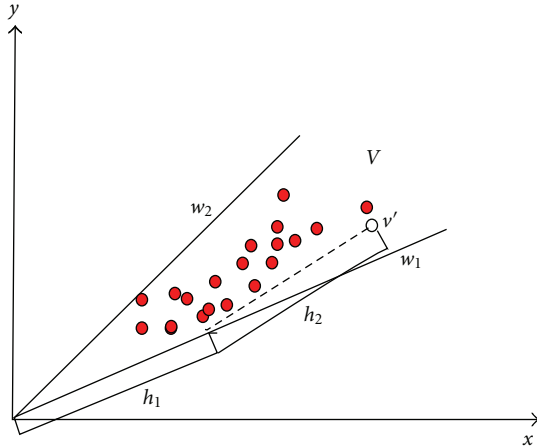


FIGURE 1: Nonnegative matrix factorization as conical coordinate transformation: illustration in two dimensional space.

parts are generally combined additively (and not subtracted) to form a whole; moreover, psychological and physiological principles assume that humans learn objects part-based. Hence, the nonnegativity constraints might be useful for learning part-based representations [16].

In this paper, we investigate the problem of performing “generic” object detection in images using the framework of NMF. By performing “generic” detection, we mean to detect, inside a given image, classes of objects, such as any car, any face, rather than finding a specific object (class instance), such a particular car, or a particular face.

Generally, object detection task is accomplished by comparing object similarities to a small number of reference features which can be expressed in holistic (global) or sparse (local) terms and then adopting a learning mechanism to identify regions in the feature space that correspond to the object class of interest. Among subspace techniques, PCA constitutes an example of approach which adopts global descriptors related to the variance of the image space (the so-called eigenfaces) to visually represent a set of given face images [17]. Other holistic approaches are based on global descriptors expressed by color, texture histogram, and global image transformations [18]. On the other hand, local features have been proved to be invariant regarding noise, occlusion or pose view and they are also supported by the theory of “recognition-by-components” introduced in [19]. The most adopted features of local type are Gabor features [20], wavelet features [21], and rectangular features [22]. Some approaches using part-based representation were proposed in [23, 24], but they present the drawback of requiring manually defined object parts and vocabulary of parts to represent object in the target class. More recently, automatic extraction of parts possessing high information contents in terms of local signal change has been illustrated in [25] together with a classifier based on a sparse representation of patches extracted around interesting points in the image.

The nonnegativity constraints of NMF make this subspace method a promising technique to automatically extract parts describing the structure of object classes. In fact, these

localized parts can be added in a purely additive way (with varying combination coefficients) to describe individual objects and could be used as learning mechanism to extract interpretable parts from a set of template images. Moreover, making use of the concept of distance from the subspace spanned by the extracted parts, NMF, could be also adopted as learning method to detect when an object is present or not inside a given image.

An interesting example of part-based representation of the original data can be found in the context of image articulation libraries. Here, NMFs are able to extract realistic parts (limbs) from image depicting stick figures with four limbs with different articulations. However, it should be pointed out that the existence of such a part-based representation heavily depends on the objects itself [26].

The firstly proposed NMF algorithms (the multiplicative and additive updated rules presented in [11]) have been applied in the fields of face identification to decompose a face image into parts reminiscent of features such as lips, eyes, and nose. More recently, comparisons between other nonnegative part-based algorithms (such as nonnegative sparse coding and local NMF) have been presented in the context of facial features, learning, demonstrating a good performance in term of detection rate by using only a small number of bases components [27]. A preliminary comparison on three NMF algorithms (classical multiplicative NMF [11], local NMF [28], and discriminant NMF [29]) has been illustrated in [30] on the recognition of different object color images. Moreover, results on the influence of additional constraints on NMF, such as the sparseness proposed in [31], have been presented in [32] for various dimensions of subspaces generated for object recognition tasks (particularly, face recognition and handwritten digits identification).

Here, we investigate the problem of performing detection of single objects in images using different NMF algorithms, in order to inquire if the representation provided by the NMF framework can effectively produce added value in detecting and locating objects inside images. The problem to be explored here can be formalized as follows. Given a collection of template images representing objects of the same class, that is a group of objects which may differ slightly from each other visually but correspond to the same semantic concept, for example, cars, digits, and faces, we would like to understand if NMF is able to provide some kind of local feature representations which can be used to individuate objects in test images.

The rest of the paper is organized as follows. The next section describes the mathematical problem of computing nonnegative matrix factorization and reviews some of the algorithms proposed in the literature and adopted to learn such a matrix decomposition model. These algorithms will constitute the core of an object detection prototype system based on the learning via NMF, proposed in Section 3 together with a brief description of its off-line and on-line learning phases. Section 4 presents experimental results illustrating the properties of the adopted NMF learning algorithms and their performance in detecting objects in real images. Finally, Section 5 concludes with a summary and possible directions for future work.



FIGURE 2: Example of a sliding window moving across a test image.

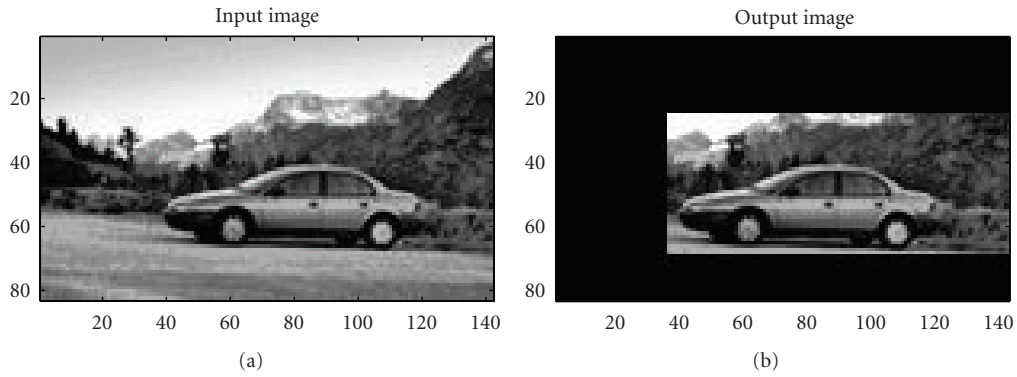


FIGURE 3: Example of output provided by the prototype system during the on-line detection phase.

2. Mathematical Background and Algorithms

The problem of finding a nonnegative low dimensional approximation of a set of data templates stored in a large dimension data matrix $V \in \mathbb{R}_+^{n \times m}$ can be stated as follows.

Given an initial dataset expressed by a $n \times m$ matrix V , where each column is an n -dimensional nonnegative vector of the original database (m vectors), find an approximate decomposition of the data matrix into a basis matrix $W \in \mathbb{R}_+^{n \times r}$ and an encoding variable matrix $H \in \mathbb{R}_+^{r \times m}$, both having nonnegative elements, such that $V \approx WH$.

Generally the rank r of the matrices W and H is much lower than the rank of V (usually it is chosen so that $(n + m)r < nm$). Each column of the matrix W contains a base vector of the spanned (NMF) subspace, while each column of H represents the weights needed to approximate the corresponding column in V by means of the vectors in W .

The NMF is actually a conical coordinate transformation: Figure 1 provides a graphical interpretation in a two dimensional space. The two basis vectors w_1 and w_2 describe a cone which encloses the dataset V . Due to the nonnegative constraint, only points within this cone can be reconstructed through linear combination of these basis vectors:

$$v' = (w_1, w_2) \cdot (h_1, h_2)^T. \quad (1)$$

The factorization of $V \approx WH$ presents the disadvantages concerning the lack of uniqueness of its factors. For example, if an arbitrary invertible matrix $A \in \mathbb{R}^{r \times r}$ such that the two matrices $W' = WA$ and $H' = A^{-1}H$ are positive semidefinite can be found, then another factorization $V \approx W'H'$ exists. Such a transformation is always possible if A is an invertible nonnegative monomial matrix (a matrix is called monomial if there is exactly one element different from zero in each row

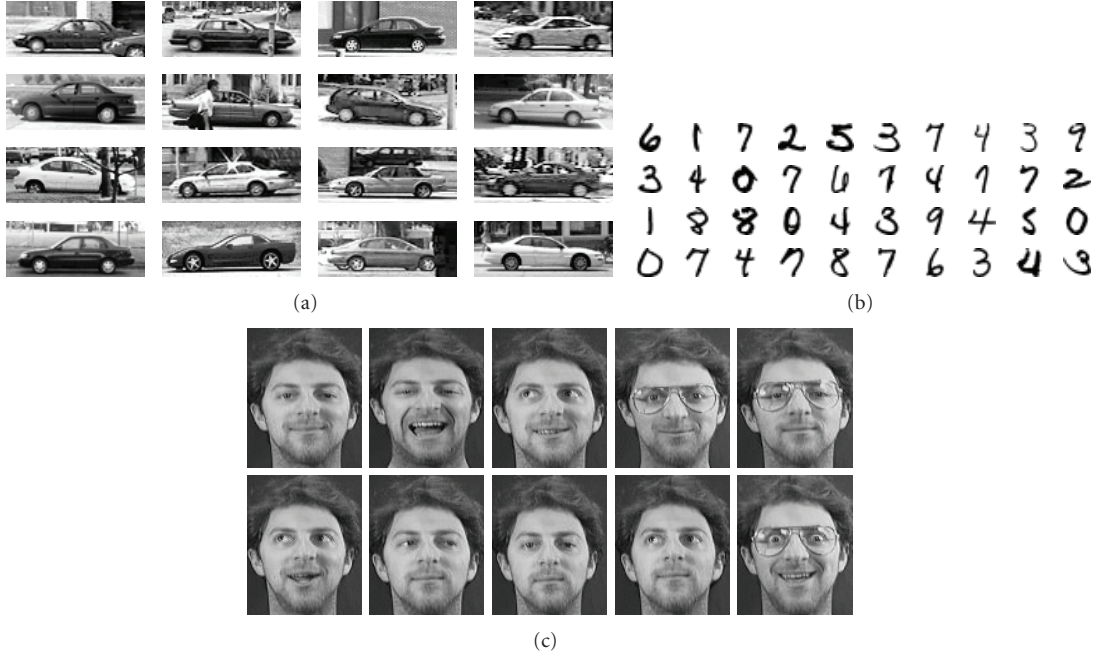


FIGURE 4: Examples of car images from (a) the CarData dataset, (b) USPS dataset, (c) ORL dataset.

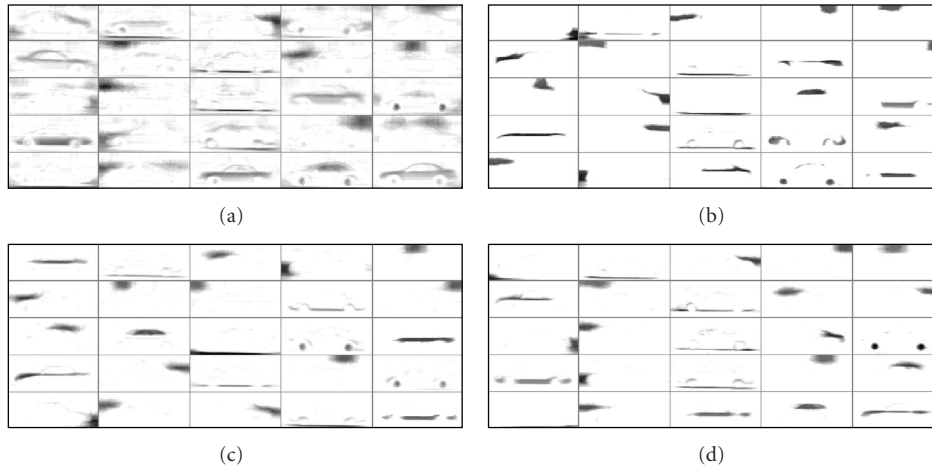


FIGURE 5: Illustration of the learnt bases (with $r = 20$) of the CarData dataset obtained via (a) NMF, (b) LNMF, (c) NMFsc, and (d) DLPP.

and column). However, if A is a nonnegative monomial matrix, in this case, the result of this transformation is simply a scaling and permutation of the original matrices [33].

An NMF of a given data matrix V can be obtained by finding a solution of a nonlinear optimization problem over a specified error function. Two simple error functions are often used to measure the distance between the original data V and its low dimensional approximation WH : the sum of squared errors (also known as the squared Euclidean distance), which leads to the minimization of the functional:

$$\|V - WH\|^2 \quad (2)$$

subject to the nonnegativity constraints over the elements W_{ij} and H_{ij} , and the generalized Kullback-Leibler divergence to the positive matrices:

$$\text{Div}(V \| WH) = \sum_{ij} \left(V_{ij} \log \left(\frac{V_{ij}}{(WH)_{ij}} \right) - V_{ij} + (WH)_{ij} \right), \quad (3)$$

subject to the nonnegativity of matrices W and H .

2.1. Classical Algorithm. The most popular approach to numerically solve the NMF optimization problem is the multiplicative update algorithm proposed in [11]. Particularly, it can be shown that the square Euclidean distance measure (2)

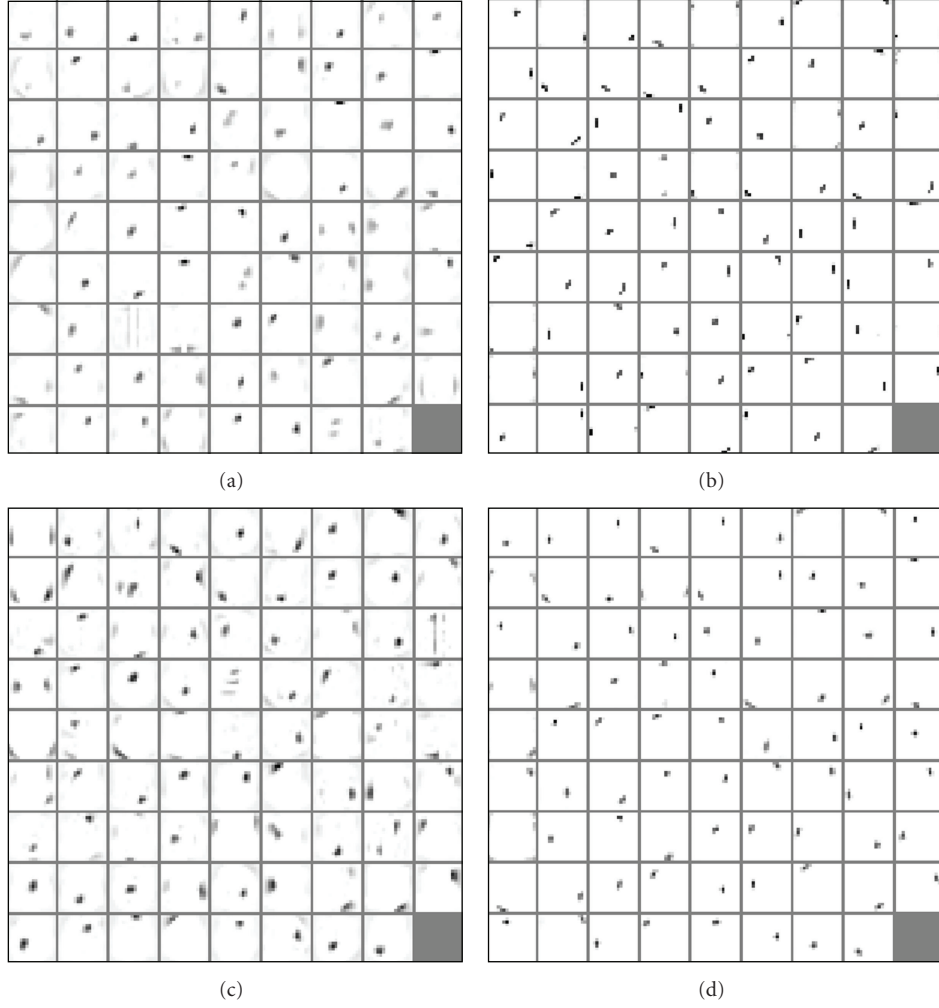


FIGURE 6: Illustration of the learnt bases (with $r = 80$) of the USPS dataset obtained via (a) NMF, (b) LNMF, (c) NMFsc, and (d) DLPP.

TABLE 1: Algorithm performances when applied to CarData, USPS, and ORL dataset, respectively. Reported values refer to the lowest and highest values of the factor rank r as previously described.

CarData						
Rank	20					
Method	MSE	Time	orth(W)	MSE	Time	orth(W)
NMF	$2.441e^9$	275	$8.7411e^4$	$1.457e^9$	453	$4.4435e^5$
LNMF	$2.404e^{10}$	292	4.9734	$2.373e^{10}$	472	10.2373
NMFsc	$2.559e^9$	695	$6.7818e^9$	$1.422e^9$	1265	$1.5825e^9$
DLPP	$2.664e^9$	2271	1.5627	$1.657e^9$	2591	3.3221
USPS						
Rank	80					
Method	MSE	Time	orth(W)	MSE	Time	orth(W)
NMF	$1.297e^4$	397	$2.8166e^4$	$3.031e^3$	847	$1.2142e^5$
LNMF	$1.331e^5$	374	6.6387	$1.609e^4$	1427	6.4695
NMFsc	$1.318e^4$	777	$5.2854e^4$	$5.568e^3$	1409	$2.7761e^4$
DLPP	$1.507e^4$	637	3.4077	$1.249e^3$	1144	3.2623
ORL						
Rank	20					
Method	MSE	Time	orth(W)	MSE	Time	orth(W)
NMF	$1.027e^9$	496	$1.5701e^5$	$5.413e^8$	705	$6.0577e^5$
LNMF	$3.104e^{10}$	556	4.4656	$3.080e^{10}$	781	8.8920
NMFsc	$1.425e^9$	1362	$1.0762e^{10}$	$6.183e^8$	2164	$2.2674e^9$
DLPP	$1.323e^9$	14824	1.7690	$8.145e^8$	15278	3.4647

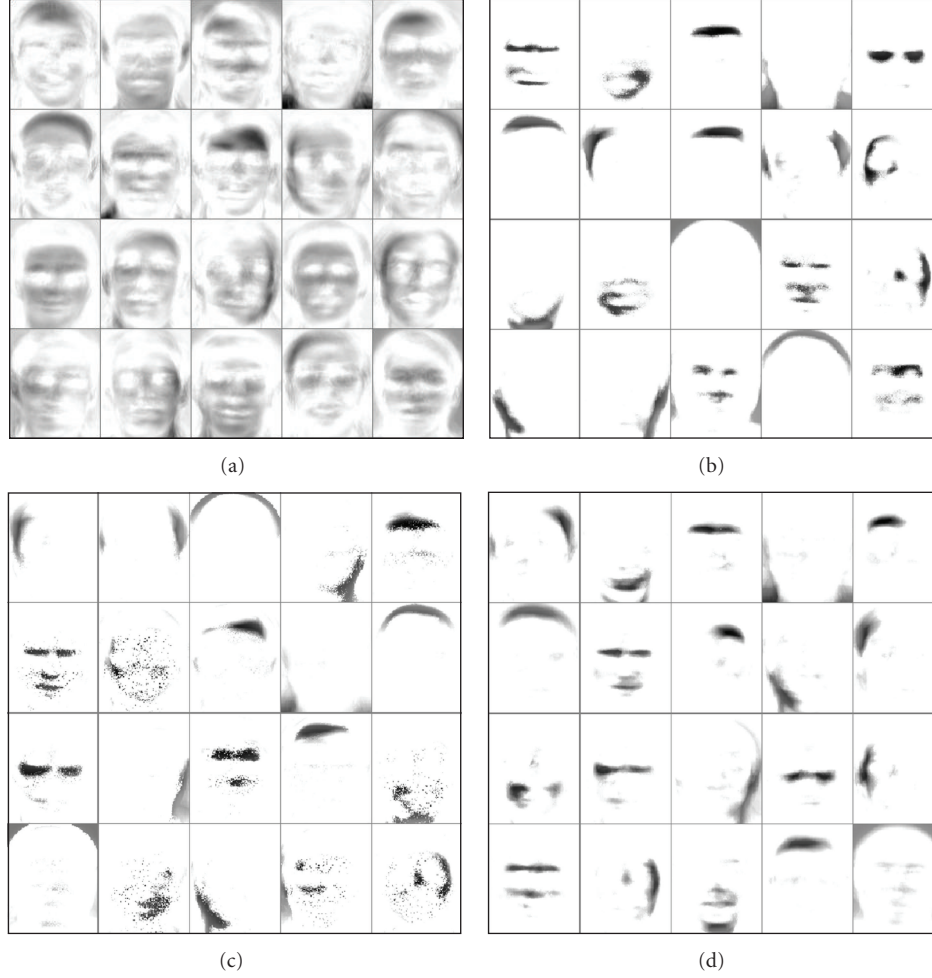


FIGURE 7: Illustration of the learnt bases (with $r = 20$) of the OPS dataset obtained via (a) NMF, (b) LNMF, (c) NMFsc, and (d) DLPP.

```

Initialize nonnegative matrices  $W^{(0)}$  and  $H^{(0)}$ 
While Stopping criteria are not satisfied do
     $W \leftarrow W \odot (VH^T) \oslash (WHH^T)$ 
     $H \leftarrow H \odot (W^T V) \oslash (W^T WH)$ 
end while
{ $\odot$  and  $\oslash$  denotes the Hadamard product, that is the element-wise matrix
multiplication and the element-wise division, respectively}

```

ALGORITHM 1: The Lee and Seung multiplicative update rules (NMF).

```

Initialize nonnegative matrices  $W^{(0)}$  and  $H^{(0)}$ 
While Stopping criteria are not satisfied do
     $H \leftarrow H \odot (VH^T) \oslash (WHH^T)$ 
     $W \leftarrow W \odot ((VH^T) \oslash (WW^T VH^T))^{(1/2)}$ 
end while
{ $\odot$  and  $\oslash$  denotes the Hadamard product and the element-wise division,
respectively and  $(\cdot)^{(1/2)}$  denotes the element-wise square root operation}

```

ALGORITHM 2: NMF with orthogonal constraint on W .

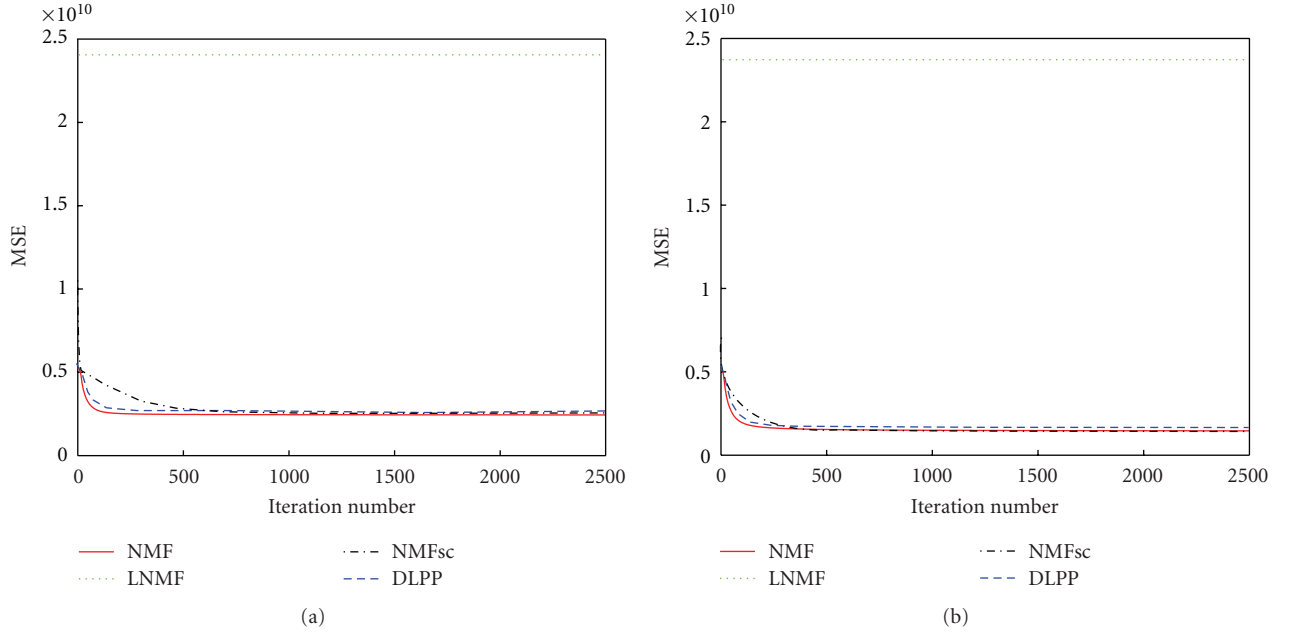


FIGURE 8: Behavior of the MSE during the learning iterations for the CarData dataset ((a) rank value $r = 20$, (b) rank value $r = 115$).

TABLE 2: Algorithm performances in terms of *recall* and *precision* when applied to CarData with factor ranks $r = 20$ and $r = 110$. Bold fonts indicate the highest values of precision and recall.

$r = 20$					
Method	TP	FP	Recall	Precision	F-measure
NMF	103	67	0.52	0.61	0.56
LNMF	92	78	0.46	0.54	0.5
NMFsc	106	64	0.53	0.62	0.57
DLPP	37	133	0.19	0.22	0.2
$r = 110$					
Method	TP	FP	Recall	Precision	F-measure
NMF	112	58	0.56	0.66	0.61
LNMF	86	85	0.43	0.5	0.46
NMFsc	110	60	0.55	0.65	0.59
DLPP	21	93	0.11	0.18	0.13

TABLE 3: Algorithm performances in terms of recall and precision when applied to Usps with factor ranks $r=80$ and $r=220$. Bold fonts indicate the highest values of precision and recall.

$r = 80$					
Method	TP	FP	Recall	Precision	F-measure
NMF	2602	98	0.96	0.96	0.96
LNMF	2457	243	0.91	0.91	0.91
NMFsc	2615	85	0.97	0.97	0.97
DLPP	658	2042	0.24	0.24	0.24
$r = 220$					
Method	TP	FP	Recall	Precision	F-measure
NMF	2602	98	0.96	0.96	0.96
LNMF	1708	1042	0.63	0.63	0.63
NMFsc	2603	97	0.96	0.96	0.96
DLPP	2195	505	0.81	0.81	0.81

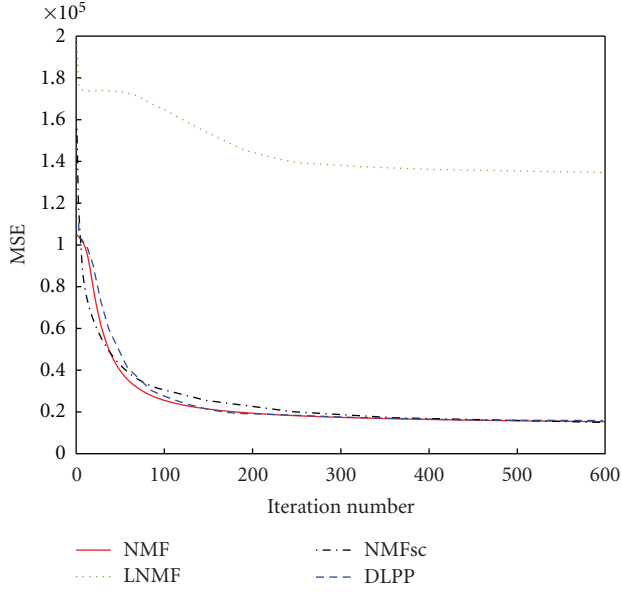


FIGURE 9: Behavior of the MSE during the initial 600 iterates in the learning phase on the USPS dataset.

TABLE 4: Algorithm performances in terms of recall and precision when applied to ORL with factor ranks $r = 20$ and $r = 80$.

$r = 20$					
Method	TP	FP	Recall	Precision	F-measure
NMF	80	0	1	1	1
LNMF	80	0	1	1	1
NMFsc	80	0	1	1	1
DLPP	40	40	0.5	0.5	0.5
$r = 80$					
Method	TP	FP	Recall	Precision	F-measure
NMF	80	0	1	1	1
LNMF	80	0	1	1	1
NMFsc	80	0	1	1	1
DLPP	41	39	0.51	0.51	0.51

is nonincreasing under the iterative updated rules described in Algorithm 1.

Lee and Seung update rules can be interpreted as a diagonally rescaled gradient descent method (i.e., a gradient descent method using a rather large learning rate). It has been proved that the above algorithm converges into a local minimum. Other techniques, such as alternating nonnegative least squares method or bound-constrained optimization algorithms, such as projected gradient method, have also been used when additional constraints are added to the nonnegativity of the matrices W or H [34–36].

2.2. NMF Algorithms with Orthogonal Constraints. Differently to other subspace methods, the learned basis vectors in NMF are not orthonormal to each other. Different modifications of the standard cost functions (2) and (3) have been proposed to include further constraints on the factors W and/or H , such as orthogonality or sparsity.

As concerning the possibility of making the bases or the encoding matrices closer to the Stiefel manifold (the Stiefel manifold is the set of all real $l \times k$ matrices with orthogonal columns $\{Q \in \mathbb{R}^{l \times k} \mid Q^T Q = I_k\}$, being I_k the $k \times k$ identity matrix) (which means that vectors in W or H should be orthonormal to each other), two different update rules have been proposed in [37] to add orthogonality on W or H , respectively. Particularly, when one desires that matrix W is as close as possible to the identity matrix of conformal dimension (i.e., $W^T W \approx I_r$), the multiplicative update rule (1) can be modified as described in Algorithm 2 (see [38] for details).

Different orthogonal NMF algorithms have been derived using directly the true gradient in Stiefel manifold [38, 39] and imposing the orthogonality between nonnegative basis vectors in learning the decomposition.

An interesting issue, strictly tied with the computation of the orthogonal NMF, when the adopted cost function is the generalized KL-divergence, is the connections with some probabilistic latent variable models. Particularly in [40], it has been pointed out that the objective function of a probabilistic latent semantic indexing model is the same of the objective function of NMF with an additional orthogonal constraint. Moreover, when the encoding matrix H is required to possess orthogonal columns, it can be proved that orthogonal NMFs are equivalent to the K-means clustering algorithm [40, 41].

2.3. NMF Algorithm with Localization Constraints. NMF algorithms optimizing a slight variation of the KL-divergence (3) can be adopted to yield a factorization which reveals local features in the data, as proposed in [28]. Particularly, local nonnegative matrix factorization uses the error function:

$$\sum_{ij} \left(V_{ij} \log \left(\frac{V_{ij}}{(WH)_{ij}} \right) - V_{ij} + (WH)_{ij} + \alpha U_{ij} \right) - \beta \sum_i Q_{ii}, \quad (4)$$

where $\alpha, \beta > 0$ are constants, and $U = W^T W$ and $Q = HH^T$. The function (4) is the KL-divergence (3) with three additional terms designed to enforce the locality of the basis features. Particularly, the modified objective function (4) attempts to minimize the number of basis components required to represent the dataset V and the redundancy between different bases, trying to make them as orthogonal as possible. Moreover, it maximizes the total activity on each component, that is, the total squared projection coefficients summed over all training data, so that only bases containing the most important information should be retained. The iterative update rules derived by the error function (4) are described in Algorithm 3.

It has been proved that the update rules in Algorithm 3 decrease monotonically the objective function (4) to a local minimum.

2.4. NMF Algorithm with Sparseness Constraints. NMF algorithms can be extended to include the option to control

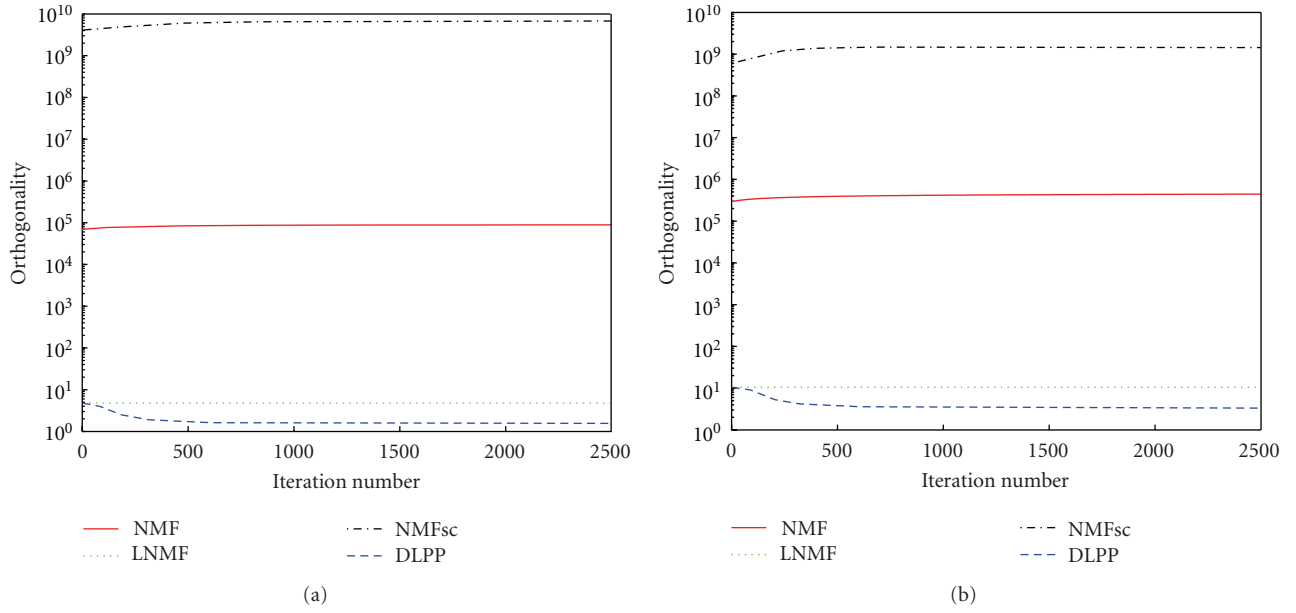


FIGURE 10: Behaviour of the orthogonality error for matrix W during the learning iterations on the CarData dataset: (a) rank value $r = 20$, (b) rank value $r = 115$.

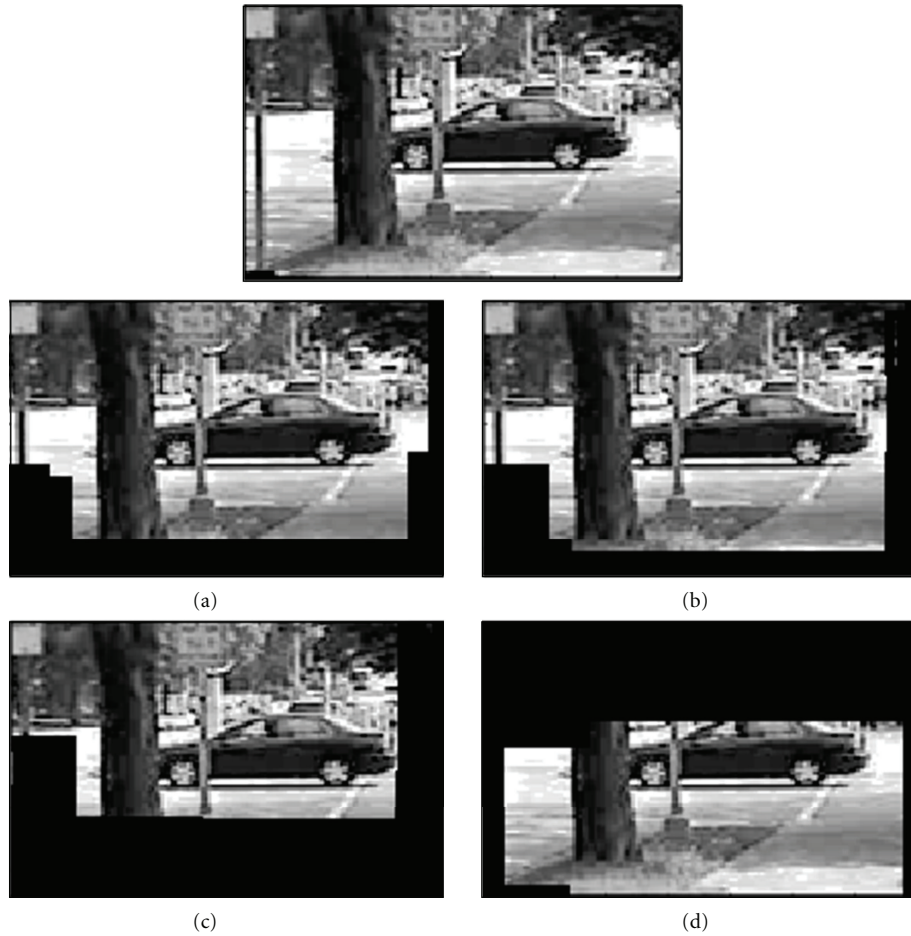


FIGURE 11: Output of the on-line detection phase after learning the CarData dataset: query image on the top, (a) NMF, (b) LNMF, (c) NMFsc, and (d) DLPP. The off-line phase has been performed with $r = 110$.

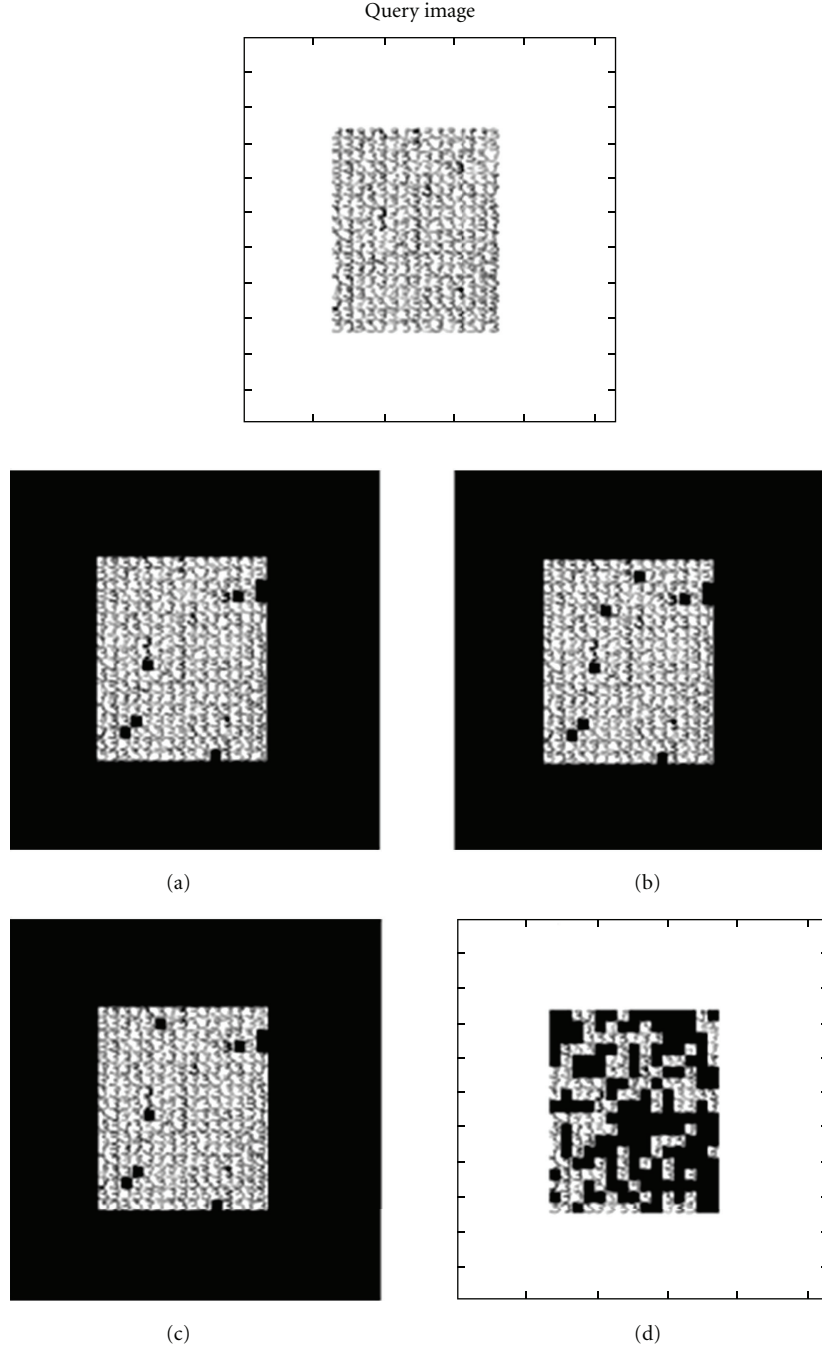


FIGURE 12: Output of the on-line detection phase after learning the USPS dataset: query image on the top, (a) NMF, (b) LNMF, (c) NMFsc, and (d) DLPP. The off-line phase has been performed with $r = 80$.

sparseness explicitly in order to discover parts-based representations that are qualitatively better than those given by standard NMF, as proposed in [31]. Particularly, to quantify the sparseness of a generic given vector $x \in \mathbb{R}^k$, the following relationship between the 1-norm and the Euclidean norm (in the original Hoyer's paper the terminology L_1 -norm and L_2 -norm is adopted) has been adopted:

$$\text{sparseness}(x) = \frac{\sqrt{k} - (\|x\|_1)/(\|x\|_2)}{\sqrt{k} - 1}. \quad (5)$$

Function (5) assumes values in the interval $[0, 1]$, where 0 indicates the minimum degree of sparsity obtained when all the elements x_i possess the same absolute value, while 1 indicates the maximum degree of sparsity, which is reached when only one component of the vector x is different from zero. This measure can be adopted to impose a desired degree of sparseness on vectors in W and/or the encoding coefficient vectors in H , depending on the specific application the nonnegative decomposition is seeking for.

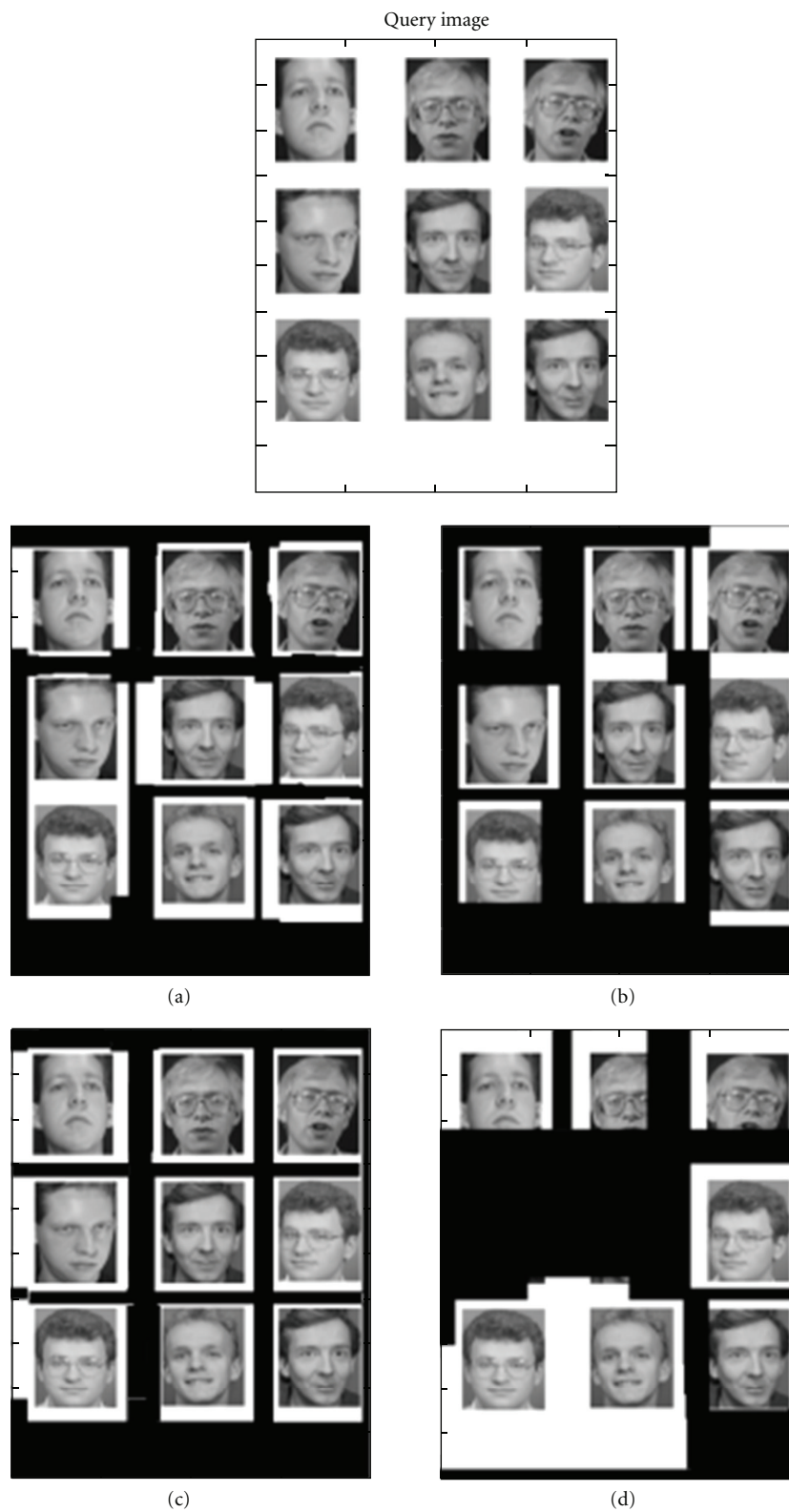


FIGURE 13: Output of the on-line detection phase after learning the ORL dataset: query image on the top, (a) NMF, (b) LNMF, (c) NMFsc, (d) and DLPP. The off-line phase has been performed with $r = 20$.

Initialize nonnegative matrices $W^{(0)}$ and $H^{(0)}$
While Stopping criteria are not satisfied **do**
 $H \leftarrow (H \odot (W^\top (V \oslash WH)))^{(1/2)}$
 $W \leftarrow W \odot ((V \oslash WH)H^\top)$
 $W \leftarrow W \text{diag}(\|W_{*1}\|_1, \|W_{*2}\|_1, \dots, \|W_{*r}\|_1)^{-1}$
end while
 { \odot and \oslash denotes the Hadamard product and the element-wise division, respectively, $(\cdot)^{(1/2)}$ denotes the element-wise square root operation, $\text{diag}(\|W_{*1}\|_1, \|W_{*2}\|_1, \dots, \|W_{*r}\|_1)$ indicates the $r \times r$ diagonal matrix whose diagonal elements are the 1-norm of the column basis vectors in W }

ALGORITHM 3: Local nonnegative matrix factorization update rules.

To compute NMF with sparseness constraints, a projected gradient descent algorithm has been developed. This algorithm essentially takes a step in the direction of the negative gradient of the cost function (2) and subsequently projects onto the constraint space, that is, the cone of nonnegative matrices with a prescribed degree of sparseness ensured imposing that sparseness $(W_i) = s_W$ and sparseness $(H_i) = s_H$, where W_i and H_i are the i th column of W and H , respectively, and s_W and s_H are the desired sparseness. The update rules used to compute W and H are described in Algorithm 4.

It should be observed that when the sparsity constraint is not required by W or H , the update rules are those provided by Algorithm 1 (the interested readers can be addressed to [31] for further details on this algorithm).

3. Object Detection System Based on NMF

In this section, we schematically present an object detection prototype system based on the learning via NMF. The working flow of the prototype system can be roughly divided in two main phases: the off-line learning phase and the on-line detection phase (mainly devoted to the object location activity).

The off-line learning phase consists in preparing the training image data and then learning a proper subspace representation of them. To be compliant to the format of the data matrix V (in order to obtain one of its nonnegative factorizations), each given $p \times q$ training image has to be converted into a pq -dimensional column vector (stacking the columns of the image matrix into a single vector) and then inserted as a column of the matrix V . It should be observed that this vector representation of an image data presents the drawback of losing the spatial relationship between neighborhood pixels inside the original image.

Once the image training matrix $V \in \mathbb{R}_+^{n \times m}$ is formed (now being $n = pq$), its NMF can be computed by applying one of the following algorithms:

- (i) the Lee and Seung multiplicative update rule (indicated by NMF and described in Algorithm 1) [11],

- (ii) NMF with orthogonal additional constraint on the basis matrix W (indicated by DLPP and described in Algorithm 2) [37],
- (iii) local NMF (indicated by LNMF and described in Algorithm 3) [28],
- (iv) NMF with sparseness additional constraint (indicated by NMFsc and described in Algorithm 4) [31].

Once the bases and the encoding matrices have been obtained using one of the previous algorithms, the on-line detection phase can be started. In particular, for each test sample image q , the distance from the subspace spanned by the learned basis matrix W is computed by means of the following formula:

$$\text{dist}(W, q) = \|q - WW^\top q\|_2. \quad (6)$$

The value distance $\text{dist}(W, q)$ is then compared with a fixed threshold ϑ , which is adopted to positively recognize the test image q as *known* object. Particularly, the decisional rule which can be easily derived is

$$\text{"IF } \text{dist}(W, q) \leq \vartheta \text{ THEN } q \text{ is labelled as known object and the object is located inside"} \quad (7)$$

Since the dimensions of the test image are bigger than those of the training images, we adopt a common approach to detect rigid object such as faces or cars [42]. Particularly, a frame of the same dimensions of training images (i.e., a window-frame of $p \times q$ pixels) is slid across the test image in order to locate the subregions of the test image which contain *known* objects. To reduce computational costs, started from the left-up corner of the test image, the sliding frame is moved in steps of size 5 percent of the test image, first in the horizontal and in the vertical direction (as shown in Figure 2).

The detection threshold is relevant to label each query image as object belonging or not to the subspace representation of the training space. Lowering the threshold increases the correct detections, but also increases the false positives; raising the threshold would have the opposite effect. To overcome this weakness, a preliminary detection phase can


```

Input: positive constants :  $\mu_W > 0, \mu_H > 0$ 
Choose an appropriate  $Projection(\cdot)$  to ensure the degree of sparseness
Initialize nonnegative matrices  $W^{(0)} H^{(0)}$ 
while Stopping criteria are not satisfied do
     $W_{ij} \leftarrow W_{ij} - \mu_W((WHH^T)_{ij} - (XH^T)_{ij})$ 
     $W \leftarrow Projection(W)$ 
     $H \leftarrow H_{ij} - \mu_H((W^T WH)_{ij} - (W^T X)_{ij})$ 
     $H \leftarrow Projection(H)$ 
end while
{ $\mu_W > 0$  and  $\mu_H > 0$  are positive constants representing the step size
of the algorithm and  $Projection(\cdot)$  indicates the appropriate projection
operator}

```

ALGORITHM 4: NMF with space constraints.

be performed in order to determine a range $[d, D]$ used to fix a default threshold value as follows:

$$\vartheta_{\text{default}} = d + (D - d) * 0.1. \quad (8)$$

The multiplicative factor 0.1 has been derived empirically. Although the simple mechanisms adopted to estimate the threshold value could cause the drawback, the proposed system identifies something also when it deals with images which do not contain any object of interest. Different estimation methods of the default threshold could be adopted to increase the detection rate; however, we delayed such aspect to a more detailed analysis to be tackled in a future work of ours. Figure 3 provides an example of the results obtained after the on-line detection phase: the picture on the left represents the test image, while the picture on the right represents a copy of the test image in which black pixels identify those pixels belonging to sliding windows which have not been identified as *known* objects.

4. Experimental Results

This section presents some experimental evaluation of the object detection/localization approach developed in the previous section. The prototype system is evaluated on single-scale images (i.e., images containing objects of the same dimension of the training data). After a brief description of the data sets adopted in the off-line training phase, some comparisons of the above-mentioned NMF algorithms are reported. Our primary concern is on the qualitative evaluation of the different algorithms in order to assess when additional constraints on basis matrix (such as sparseness and orthogonality) and/or different number of bases images (explicitly represented by the rank r) can produce better results in object detection.

All the numerical results have been obtained by Matlab 7.7 (R2008b) codes implemented on an Intel Core 2 Quad Q9400 processor, 2.66 GHz with 4 GB RAM. The execution time of each algorithm has been computed by the build in Matlab functions tic and toc.

In order to test the object detection prototype system based on the illustrated NMF algorithms, three image datasets have been adopted: CarData, USPS, and ORL.

The exploited datasets represent three different typologies of objects: cars, handwritten digits, and faces, respectively. Figure 4 illustrates some training images from the adopted datasets.

The CarData training set contains 550 gray scale training images of cars of size 100×40 pixels, while the test set is composed by 170 single-scale test images, containing 200 cars at roughly the same scale as in the training images. The USPS dataset contains normalized gray scale images of handwritten digits of size 16×16 pixels, divided into a training set of 7291 images and a test set of 2007 images including all digits from 0 to 9. A preprocessing of USPS has been applied to rescale pixel values from the range $[-1, 1]$ to the range $[0, 1]$. Figure 4 illustrates some training images from the adopted datasets. The ORL dataset contains gray scale images of faces of 40 distinct subjects. Each image is of size 92×112 pixels and has been taken against a dark homogeneous background with the subjects in an upright, frontal position, with slight left/right out-of-plane rotation. We use the first 8 images of each subject for the training set and the remaining 2 images for the test set.

4.1. Experimental Setup. The off-line learning phase has been run for different values of the rank r (representing the number of bases images) and with selected degree of sparsity imposed to NMFsc algorithm (particularly, the sparsity parameters in NMFsc have been fixed as $s_W = 0.5$ and $s_H = []$). As previously observed, we are interested in assessing the existence of any qualitative difference between the NMF learning algorithms in the context of generic object detection. In fact, the rank value r represents the dimensionality of the subspace spanned by the matrix W : an increase in its value can be interpreted as an information gain with respect to the original dataset. On the other hand, large values of r could introduce some redundancy in the basis representation of the dataset, nullifying the benefits provided by the part-based representation of the NMF. The algorithms have been trained on each dataset for various values of rank (CarData: $r = 20, 110$, USPS: $r = 80, 220$, ORL: $r = 20, 80$). We report the results related to the lowest and the highest rank values for each dataset. For the benefit of comparison, the same stopping criteria has

been adopted for all NMF learning algorithms (i.e., the algorithms stop when the maximum number of iterations, set to 2500, is reached). Moreover, the results reported in the following sections represent the average values obtained over ten different random initializations of the nonnegative initial matrices $W^{(0)}$ and $H^{(0)}$. Note that, for each trial, the same initial matrices randomly generated (with proper dimensions with respect to the adopted dataset) have been used for all the algorithms.

The algorithms have been compared in terms of final approximation error, computed by $\text{MSE}(W, H) = \|V - WH\|_2$, execution time (indicating the number of seconds required by each algorithm to complete the learning phase) and degree of orthogonality of W , measured by $\text{orth}(W) = \|W^T W - I\|_F$. This latter measure has been added for highlighting when additional constraints (in the specific case the orthogonality of the basis factor) provide better results in the detection phase.

4.2. Results of the Off-Line Learning Phase. This section reports the results obtained at the end of the off-line training phase for all the three chosen image datasets. Table 1 reports the MSE, the execution time, and the degree of orthogonality of W , when the algorithms are trained on the chosen datasets. For each dataset, the results obtained for the initial value and the final value of the rank are reported. These results are related to the lower and the higher subspace approximation of each dataset.

Figure 5 illustrates the part-based representation of CarData dataset learned by the adopted algorithms. For the benefit of appreciating some visual difference between the obtained bases, we plot the bases only for the smaller value $r = 20$. Analogously, Figures 6 and 7 report the bases representation of USPS (with rank value $r = 80$) and ORL dataset (with rank value $r = 20$), respectively. Algorithm NMF learns global representation of either set of face car and face image, while it provides local representation of handwritten digits. LNMF, DLPP, and NMFsc algorithms, instead, learn localized image parts some of which appear to roughly correspond to parts of faces, parts of cars, part of digit marks. Essentially, the NMF algorithms select a subset of the pixels which are simultaneously active across multiple images to be represented by a single bases vector.

As an example, Figure 8 illustrates the behavior of the MSE during the learning phase on the CarData dataset, with rank values $r = 20$ and $r = 115$, respectively. It should be observed that after some iterations all algorithms converge to similar values of the MSE. The LNMF algorithm presents a larger value of the MSE just because this algorithm is based on the KL-divergence cost function so it provides a rougher approximation of the dataset in term of MSE. To better appreciate the rate of convergence of all algorithms, Figure 9 reports the behavior of the MSE during the initial 600 iterates in the learning phase associated with the USPS dataset, with rank value $r = 80$. A behavior similar to that depicted in Figures 8 and 9 is shown for all the other datasets and for different values of the rank r .

As concerning the degree of orthogonality of the matrix W learned by each algorithm, Figure 10 reports the semilog

plot of the orthogonality error for W during the learning iterations on the CarData dataset (with the rank values $r = 20$ and $r = 115$, resp.). It should be observed that both LNMF and DLPP produce a matrix W possessing a discrete degree of orthogonality. On the other hand, since NMF and NMFsc do not incorporate any additional constraint, they preserve or sometimes deteriorate the degree of orthogonality of the initial matrix W_0 . Similar plots for the orthogonal error can be depicted for the matrices obtained using the USPS and ORL dataset, respectively.

4.3. Results of the On-Line Detection Phase. Once the bases and the encoding matrices have been obtained at the end of the learning phase, we are ready to enter the on-line detection and localization phase in order to assess a qualitative analysis of the considered algorithms (by means of the prototype system). To measure the performance of the NMF-based object detection/localization system, we are interested in knowing how many of the objects it detects and how often the detection it makes is false. Particularly, the two quantities of interest are the number of correct detections and the number of false detections: the former should be maximized while the latter quantity has to be minimized. As we have already observed in Section 3, the decisional rule (7), which allows to identify a test image as *known* object, is dependent on the detection threshold ϑ . Opportunely varying the threshold ϑ , a different tradeoff between correct and false detections can be reached. This tradeoff can be estimated considering the *recall* and the *precision*. The recall is the proportion of objects that are detected, the precision is the fraction of corrected detected objects among the total number of detection made by the system. Denoting by TP the number of true positive, TF the number of false positive, nP and nF the total number of positives and negatives in the dataset, respectively, the performance measures are $\text{Recall} = TP/nP$ and $\text{Precision} = TP/(TP + FP)$, and the number of false detections can be computed as $1 - \text{Precision}$. It should be pointed out that precision-recall is a more appropriate measure than the common ROC curve, since this metric is designed for binary classification tasks, not for detection tasks [25].

The evaluation results have been obtained by manually determining the location of the windows containing interesting objects. Tables 2, 3 and 4 report the performance results for Cardata, USPS, and ORL, respectively, when different values of the dimensionality r of the subspace dataset approximation are adopted. NMF algorithms evidence some differences in terms of recall and precision, particularly NMF and NMFsc provide better results than LNMF and DLPP. The performance of the latter algorithms is also quite bad on the ORL face dataset, which represents one of the easiest database in terms of recognition.

Figure 11 reports the results obtained after the on-line phase on a car test example. The picture on the top illustrates the query image; the remaining pictures provide the positive pixels provided by (a) NMF, (b) LNMF, (c) NMFsc, and (d) DLPP, respectively (trained with $r = 110$).

Figure 12 illustrates the results obtained after the on-line phase on a handwritten digit test example. The picture on

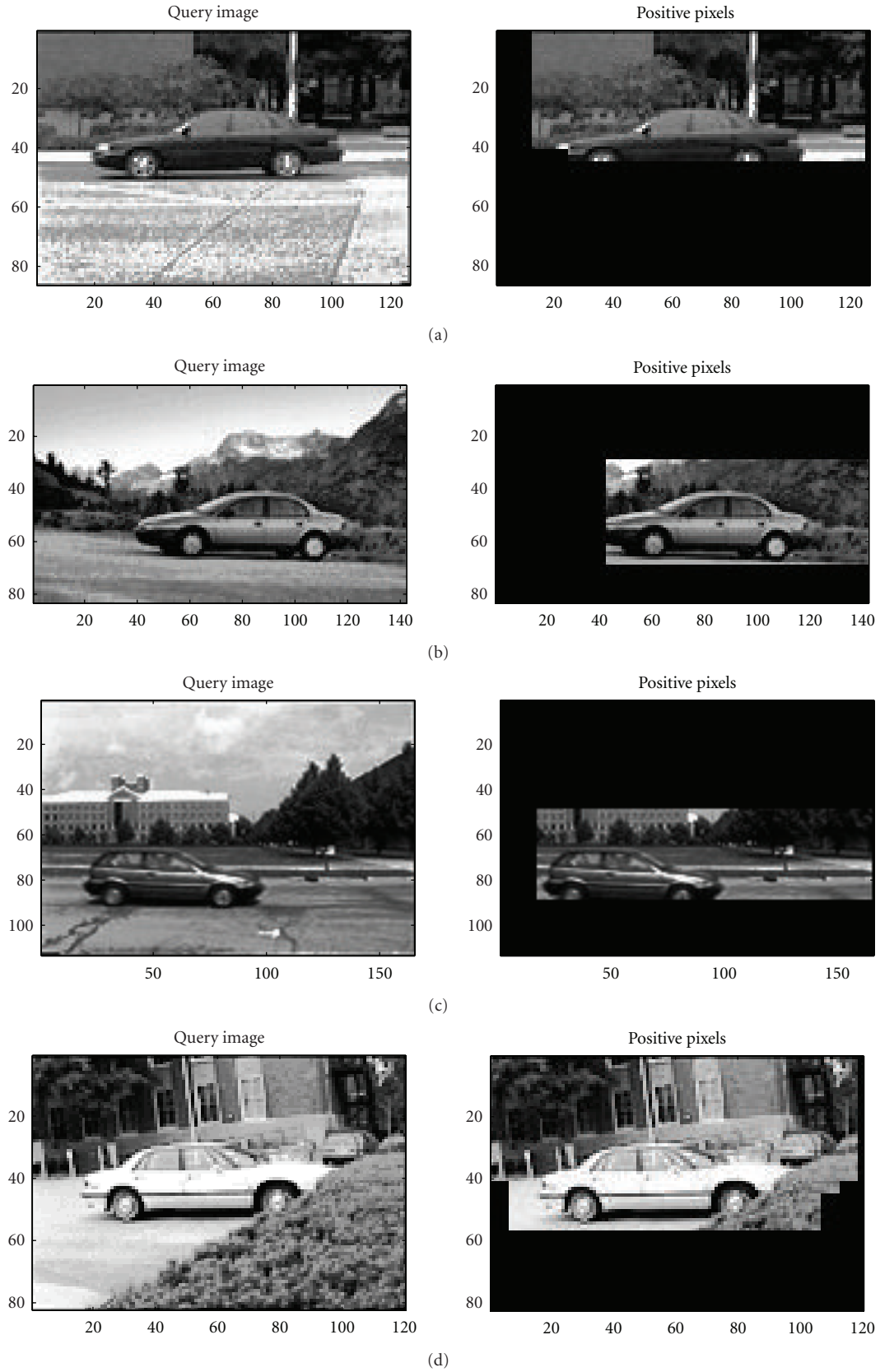


FIGURE 14: Output of the on-line detection phase after learning the CarData dataset: (a) NMF, (b) LNMF, (c) NMFsc, and (d) DLPP with $r = 110$ and $\vartheta = 2.6e^3$.

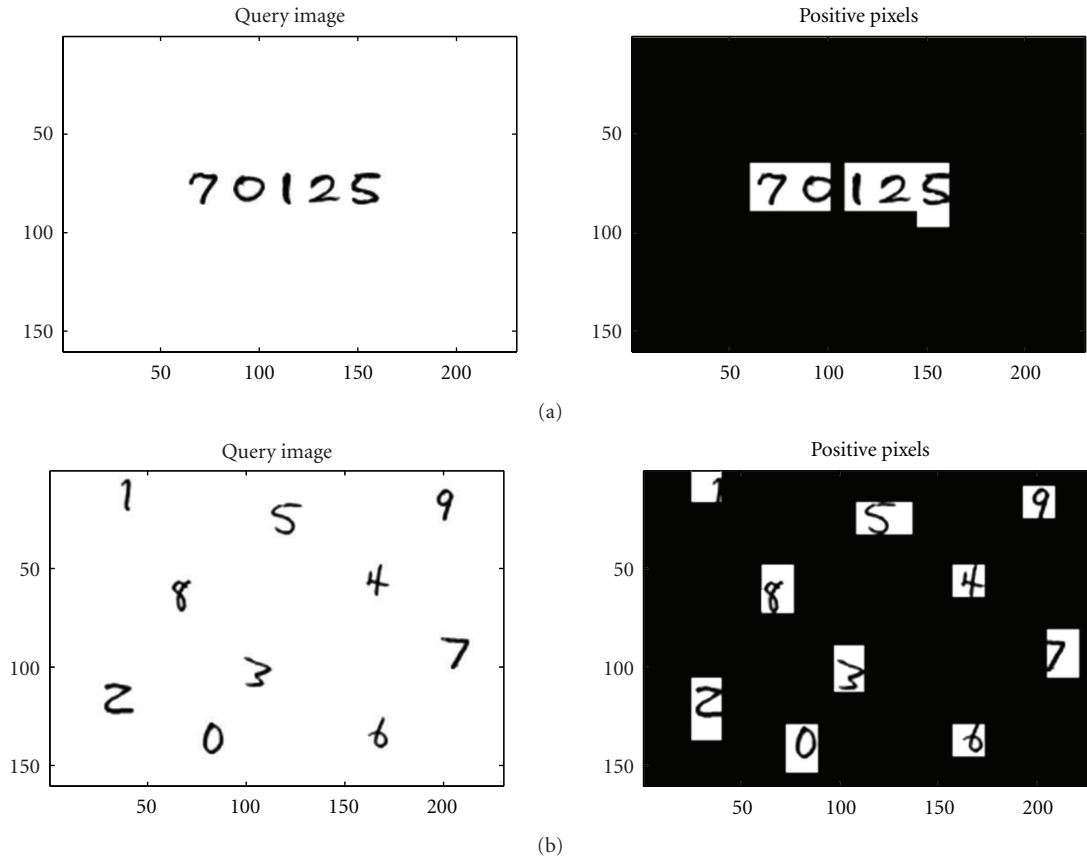


FIGURE 15: Output of the on-line detection phase after learning the USPS dataset: (a) NMF, (b) NMFsc with $r = 80$ and $\vartheta = 1.0e^8$.

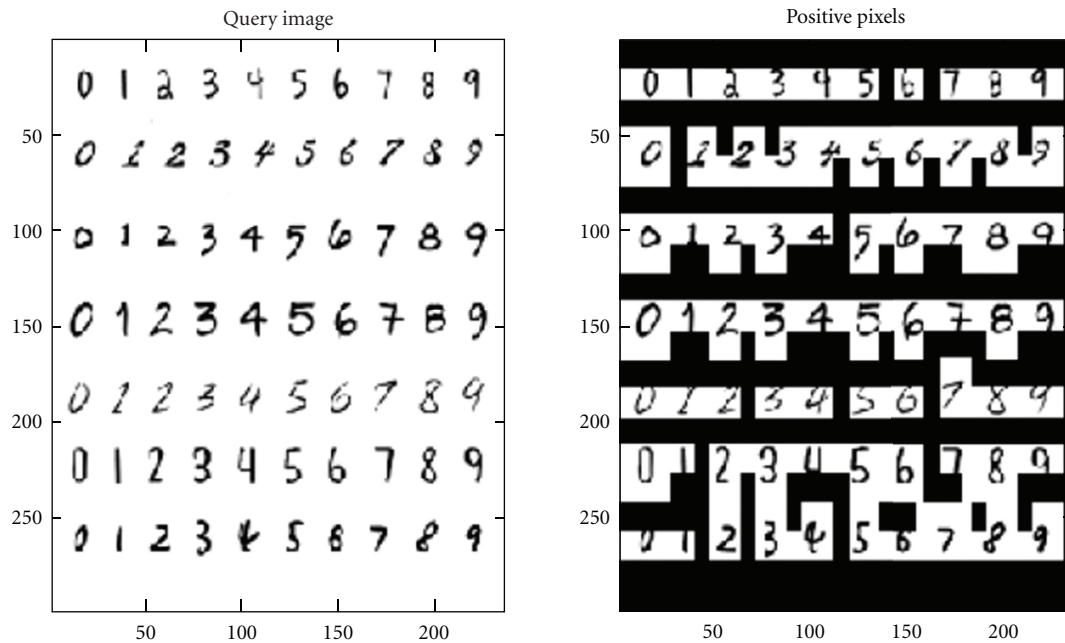


FIGURE 16: Output of the on-line detection phase on a white paper image presenting some handwritten digits. Test is made with NMFsc after learning the USPS dataset, with $r = 80$ and $\vartheta = 1.0e^8$.

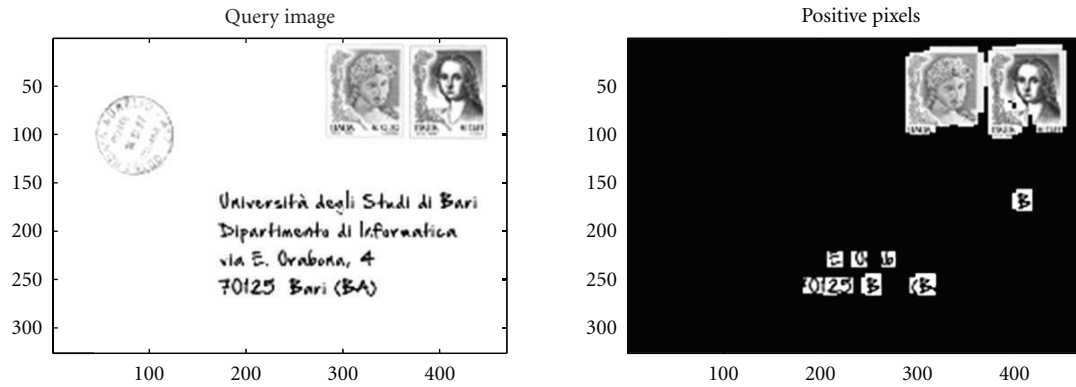
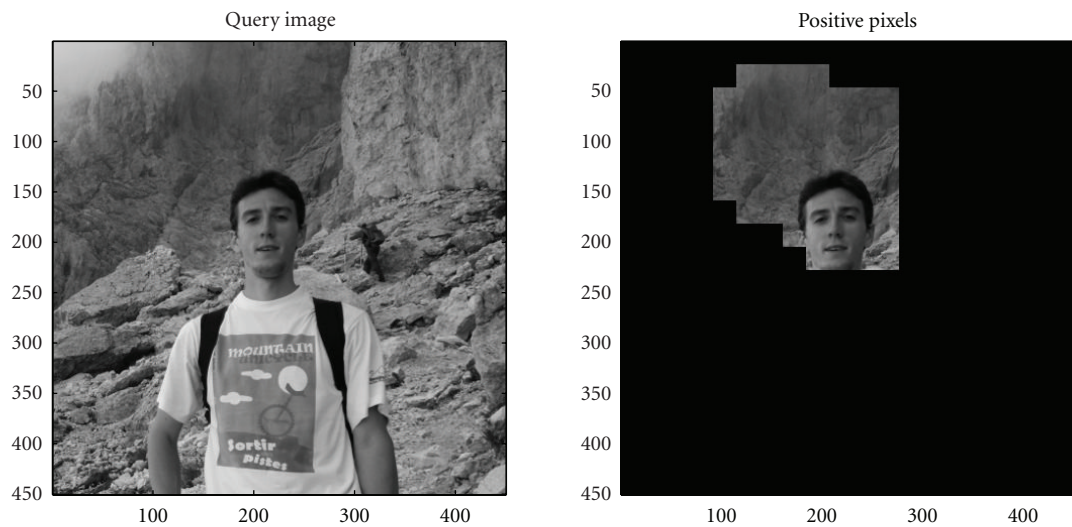
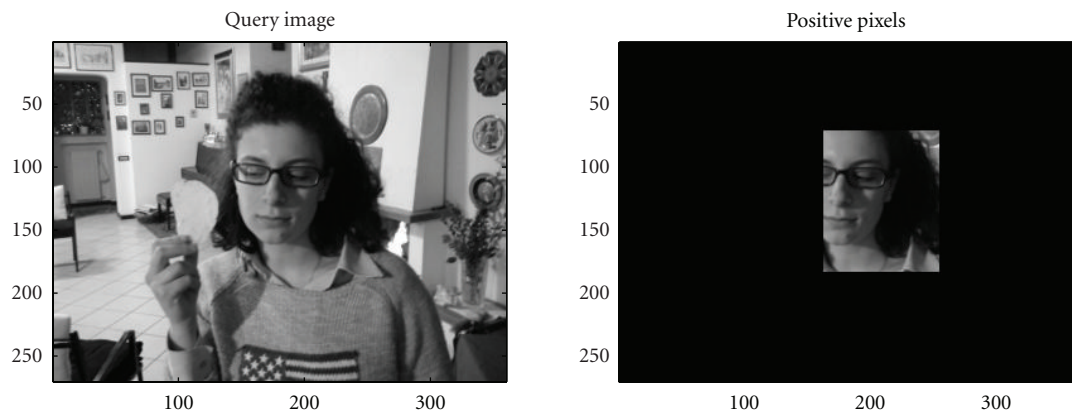


FIGURE 17: Output of the on-line detection phase on a letter envelope image presenting some handwritten digits. Test is made with LNMF after learning the USPS dataset, with $r = 80$ and $\vartheta = 2.3e^3$.



(a)



(b)

FIGURE 18: Output of the on-line detection phase after learning the ORL dataset. Test is made with NMFsc with $r = 20$ and $\vartheta = 2.4e^3$.

the top illustrates the query image, the remaining pictures provide the positive pixels provided by (a) NMF, (b) LNMF, (c) NMFsc, and (d) DLPP, respectively (trained with $r = 80$). As it can be noted the DLPP algorithm provides the worst result, since it locates all the background pixels around the digit images.

Figure 13 illustrates the results obtained after the on-line phase on a composited image with different ORL test images. Again, the picture on the top illustrates the query image, the remaining pictures provide the positive pixels provided by (a) NMF, (b) LNMF, (c) NMFsc, and (d) DLPP, respectively (trained with $r = 80$). Also in this case, the worst results are given by DLPP algorithm, which is not able to correctly locate all the ORL test images.

4.4. Qualitative Analysis in Natural Images. The following images illustrate the results obtained during the on-line detection phase for each considered algorithm with different query images. Particularly, Figure 14 provides an example of detection of a car inside some test images taken from the CarData test set.

Figure 15 illustrates the detection and location of some digit images inserted in a large scale image with white background while Figure 16 reports the detection/location results of some digit image written on a large white page. Figure 17 shows the detection of some handwritten digits presenting on an image of a real letter envelope. In the latter case, it could be observed that there are some false positive detections such as the two stamps and the letters in the address. This can be explained in the case of the stamps by considering their bigger dimension with respect to the sliding window and also the bases (see Figure 6) learnt by the algorithm, in the case of the letters by considering the inherent resemblance between some handwritten numbers and letters (such as “0” and “O,” “B” and “8,” “6” and “b”).

Figure 18 gives evidence of the capability of NMF algorithms to recognize human face inside two real world pictures which portrait human figures with different backgrounds; as it can be observed the adopted algorithm is able to recognize the presence of a face different from the training faces learnt in the off-line training phase. This represents a confirmation that the part-based representation provided by NMF can effectively produce added value in detecting and locating objects inside images.

5. Conclusions and Future Work

To summarize, we have presented a prototype framework for learning how to detect and locate “generic” objects in images using the part-based representation provided by nonnegative matrix factorization of a set of template images. Comparisons between different NMF algorithms have been presented, evidencing that different additional constraints (such as sparseness) could be more suitable to identify localized parts describing some structures in object classes. Our experiments on the well-known databases demonstrated that the proposed NMF-based prototype system is able to extract such interpretable parts from a set of training images

in order to use them in localizing similar object in real world image.

Future work could be undertaken to allow the elaboration of object images with different scales, to improve final localization (using, for instance, a repeated part elimination algorithm), and to apply different criteria and/or measures to identify when a test image does or not belong to the subspace of known objects.

Acknowledgment

The authors would like to thank the anonymous referees for their suggestions and comments, which proved to be very useful for improving the paper.

References

- [1] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, 3rd edition, 2001.
- [2] I. T. Jolliffe, *Principal Component Analysis*, Springer, 1986.
- [3] A. Hyvarinen, “Independent component analysis,” *Neural Computing Surveys*, vol. 2, 2001.
- [4] D. Guillaumet and J. Vitrià, “Non-negative matrix factorization for face recognition,” *Lecture Notes in Computer Science*, vol. 2504, pp. 336–344, 2002.
- [5] X. Sun, Q. Zhang, and Z. Wang, “Face recognition based on NMF and SVM,” in *Proceedings of the 2nd International Symposium on Electronic Commerce and Security*, pp. 616–619, 2009.
- [6] D. Guillaumet and J. Vitrià, “Evaluation of distance metrics for recognition based on non-negative matrix factorization,” *Pattern Recognition Letters*, vol. 24, no. 9–10, pp. 1599–1605, 2003.
- [7] W. Liu and N. Zheng, “Non-negative matrix factorization based methods for object recognition,” *Pattern Recognition Letters*, vol. 25, no. 8, pp. 893–897, 2004.
- [8] Y. Gao and G. Church, “Improving molecular cancer class discovery through sparse non-negative matrix factorization,” *Bioinformatics*, vol. 21, no. 21, pp. 3970–3975, 2005.
- [9] H. Kim and H. Park, “Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis,” *Bioinformatics*, vol. 23, no. 12, pp. 1495–1502, 2007.
- [10] P. Paatero and U. Tapper, “Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values,” *Environmetrics*, vol. 5, no. 2, pp. 111–126, 1994.
- [11] D. D. Lee and S. H. Seung, “Algorithms for non-negative matrix factorization,” in *Proceedings of the Advances in Neural Information Processing Systems Conference*, vol. 13, pp. 556–562, MIT Press, 2000.
- [12] M. Novak and R. Mammone, “Use of nonnegative matrix factorization for language model adaptation in a lecture transcription task,” in *Proceedings of the IEEE International Conference Acoustic, Speech and Signal Processing*, vol. 1, pp. 541–544, IEEE Computer Society, 2001.
- [13] M. Chu and R. J. Plemmons, “Nonnegative matrix factorization and applications,” *IMAGE, Bulletin of the International Linear Algebra Society*, vol. 34, pp. 2–7, 2005.
- [14] V. P. Pauca, J. Piper, and R. J. Plemmons, “Nonnegative matrix factorization for spectral data analysis,” *Linear Algebra and Its Applications*, vol. 416, no. 1, pp. 29–47, 2006.

- [15] Chen D. and Plemmons R., "Nonnegativity constraints in numerical analysis," in *Proceedings of the Symposium on the Birth of Numerical Analysis*, pp. 541–544, World Scientific Press, 2008.
- [16] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.
- [17] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–591, 1991.
- [18] B. Schiele and J. L. Crowley, "Recognition without correspondence using multidimensional receptive field histograms," *International Journal of Computer Vision*, vol. 36, no. 1, pp. 31–50, 2000.
- [19] I. Biederman, "Recognition-by-components: a theory of human image understanding," *Psychological Review*, vol. 94, no. 2, pp. 115–147, 1987.
- [20] T. S. Lee, "Image representation using 2d gabor wavelets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, pp. 959–971, 1996.
- [21] R. N. Strickland and H. I. Hahn, "Wavelet transform methods for object detection and recovery," *IEEE Transactions on Image Processing*, vol. 6, no. 5, pp. 724–735, 1997.
- [22] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1511–1518, December 2001.
- [23] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-based object detection in images by components," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 4, pp. 349–361, 2001.
- [24] S. Ullman, M. Vidal-Naquet, and E. Sali, "Visual features of intermediate complexity and their use in classification," *Nature Neuroscience*, vol. 5, no. 7, pp. 682–687, 2002.
- [25] S. Agarwal, A. Awan, and D. Roth, "Learning to detect objects in images via a sparse, part-based representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1475–1490, 2004.
- [26] D. Donoho and V. Stodden, "When does non-negative matrix factorization give a correct decomposition into parts?" in *Proceedings of the Neural Information Processing Systems*, vol. 16, pp. 1141–1149, 2003.
- [27] B. J. Shastri and M. D. Levine, "Face recognition using localized features based on non-negative sparse coding," *Machine Vision and Applications*, vol. 18, no. 2, pp. 107–122, 2007.
- [28] S. Z. Li, X. Hou, H. Zhang, and Q. S. Cheng, "Learning spatially localized, parts-based representation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 207–212, IEEE Computer Society, 2001.
- [29] I. Buciu and I. Pitas, "A new sparse image representation algorithm applied to facial expression recognition," in *Proceedings of the 14th IEEE Signal Processing Society Workshop of the Machine Learning for Signal Processing*, pp. 539–548, October 2004.
- [30] I. Buciu, "Learning sparse non-negative features for object recognition," in *Proceedings of the IEEE 3rd International Conference on Intelligent Computer Communication and Processing (ICCP '07)*, pp. 73–79, September 2007.
- [31] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.
- [32] D. Soukup and I. Bajla, "Robust object recognition under partial occlusions using NMF," *Computational Intelligence and Neuroscience*, vol. 2008, Article ID 857453, 14 pages, 2008.
- [33] A. Berman and R. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, Academic Press, 1979.
- [34] M. T. Chu, F. Diele, R. Plemmons, and S. Ragni, "Optimality, computation and interpretation of nonnegative matrix factorizations," Tech. Rep., NCSU, 2005.
- [35] M. T. Chu and M. M. Lin, "Low-dimensional polytope approximation and its applications to nonnegative matrix factorization," *SIAM Journal on Scientific Computing*, vol. 30, no. 3, pp. 1131–1155, 2007.
- [36] C.-J. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural Computation*, vol. 19, no. 10, pp. 2756–2779, 2007.
- [37] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix tri-factorizations for clustering," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '06)*, pp. 126–135, August 2006.
- [38] S. Choi, "Algorithms for orthogonal nonnegative matrix factorization," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '08)*, pp. 1828–1832, June 2008.
- [39] N. Del Buono, "A penalty function for computing orthogonal non-negative matrix factorizations," in *Proceedings of the 9th International Conference on Intelligent Systems Design and Applications (ISDA '09)*, pp. 1001–1005, December 2009.
- [40] C. Ding, T. Li, and W. Peng, "Nonnegative matrix factorization and probabilistic latent semantic indexing: equivalence, chi-square statistic, and a hybrid method," in *Proceedings of the 21st National Conference on Artificial Intelligence and the 18th Innovative Applications of Artificial Intelligence Conference (AAAI '06)*, pp. 342–347, July 2006.
- [41] C. Ding, X. He, and H. D. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering," in *Proceedings of the SIAM Data Mining Conference*, pp. 606–610, 2005.
- [42] K. Murphy, A. Torralba, D. Eaton, and W. Freeman, "Object detection and localization using local and global features," *Lecture Notes in Computer Science*, vol. 4170, pp. 394–412, 2006.