HIERARCHICAL DICTIONARY LEARNING FOR INVARIANT CLASSIFICATION

Leah Bar and Guillermo Sapiro

Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, USA

ABSTRACT

Sparse representation theory has been increasingly used in the fields of signal processing and machine learning. The standard sparse models are not invariant to spatial transformations such as image rotations, and the representation is very sensitive even under small such distortions. Most studies addressing this problem proposed algorithms which either use transformed data as part of the training set, or are invariant or robust only under minor transformations. In this paper we suggest a framework which extracts sparse features invariant under *significant* rotations and scalings. The algorithm is based on a hierarchical architecture of dictionary learning for sparse coding in a cortical (log-polar) space. The proposed model is tested in supervised classification applications and proved to be robust under transformed data.

Index Terms— Sparse models, dictionary learning, hierarchy, log-polar, invariance, classification

1. INTRODUCTION

Sparse signal models over learned dictionaries were proved to be very powerful in recent years in the fields of image processing, speech processing, and machine learning. Sparse representations have the advantage of capturing inherent structures of the signal and demonstrate relative robustness to (additive) noise. In their standard form, these compact representations are not invariant under transformations such as translation, scaling, and rotation. Kavukcuoglu *et al.* [1] learn locally-invariant feature descriptors by pooling the sparse coefficients across overlapping windows. Yet, their algorithm is designed for relatively small distortions. Shift-invariant dictionary learning was investigated by [2, 3]. The idea in both papers is to train the dictionaries on many possible shifted versions of the signal (see also [4] for related ideas extending the popular SIFT descriptor to the affine case). This approach may be computational expensive, and in addition, several transformations may lead to impractical implementation. Huang *et al.* [5] simultaneously recover the sparse representation of the target image and the geometrical (translation/affine) transformation between the target and model images. The transformations are approximated by first order Taylor expansion and therefore have again the limitation of being relatively small.

Invariance can be approached by biologically-inspired architectures. Typically, the extraction of local features is followed by spatial pooling which is classically modeled as a *hierarchy* of increasingly complex structures. These ideas led to extensive research and algorithms. Serre *et al.* [6], for example, suggested a scale and position tolerant feature detector based on the alternation between template matching and a maximum pooling operator. Ranzato *et al.* [7] suggested a hierarchical feature extraction algorithm which is invariant under small shifts and distortions.

In this paper we introduce a framework for dictionary learning and sparse feature extraction, which is invariant under *significant* ordinary rotation and scaling transformations. In the proposed method, we integrate the ideas of sparse representation theory and hierarchical structures. By using a special conformal (log-polar) mapping of the data, rotated and scaled patterns are converted into shifted patterns in the new space on which we operate for learning the dictionary and adding hierarchy. Our approach is closely related to [8], where log-polar images were represented by invariant wavelet packets. Yet, in our work we incorporate a *hierarchical architecture* that is designed to also eliminate the effects of translations in this space. As we demonstrate, a hierarchical approach performs better than a one-layered one. Moreover, we learn dictionaries instead of using predefined wavelets, following the recent results in the literature clearly showing that such learned dictionaries often outperform off-the-shelf ones.

The suggested approach is general and suited for data of very different nature. The method was particularly tested with two applications: in the first example, we classified handwritten digits with only aligned training patterns and transformed tested patterns with significant rotations and scalings. Next, classification of texture images with large variability of scaling and rotations was performed. Promising results support the stability and robustness of the suggested approach.

2. BACKGROUND AND NOTATIONS

In sparse modeling representation, a signal $x \in \mathbb{R}^n$ is represented as a linear combination of basis column vectors $\mathbf{d}_j \in \mathbb{R}^n$ (atoms) which form a dictionary $\mathbf{D} \in \mathbb{R}^{n \times K}$, such that $x = \mathbf{D}\alpha$. The vector $\alpha \in \mathbb{R}^K$ is assumed to be sparse, meaning that the number of non-zero elements is much smaller than K. A dictionary can be overcomplete, $K \gg n$, as is often used in restoration algorithms. Most classification algorithms on the other hand, use undercomplete ($K < n$) dictionaries. Given a dictionary $\mathbf{D}$ and a signal x, the ℓ_1 sparse coding problem is given by

$$\hat{\alpha} = \arg\min_{\alpha} \|x - \mathbf{D}\alpha\|_2^2 + \lambda\|\alpha\|_1, \tag{1}$$

where $\lambda \in \mathbb{R}$ is a regularization constant. This formulation and its variants are often referred to as *basis pursuit* or *Lasso* [9]. The optimization in this work was carried out using the Lars [10] algorithm which we denote as $(\alpha) \leftarrow \mathbf{Lars}(x, \mathbf{D})$.

Consider a set of m signals $\mathbf{X} = [x_1, \ldots, x_m] \in \mathbb{R}^{n \times m}$. The dictionary $\mathbf{D}$ and coefficients set $\alpha = [\alpha_1, \ldots, \alpha_m] \in \mathbb{R}^{K \times m}$ are given by

$$\hat{\mathbf{D}}, \hat{\alpha} = \arg\min_{\mathbf{D},\alpha} \sum_{i=1}^{m} \|x_i - \mathbf{D}\alpha_i\|_2^2 + \lambda\|\alpha_i\|_1. \tag{2}$$

The optimization is performed by alternate minimization w.r.t. α and $\mathbf{D}$. We denote the dictionary learning process as $(\mathbf{D}, \alpha) \leftarrow \mathbf{TrainDictionary}(\mathbf{X})$. Detailed description of both algorithms can be found for example in [11].

3. CLASSIFICATION VIA SPARSE CODING

Let $\mathbf{X}^{train}$ be a labeled training set, and $\mathbf{X}^{test}$ the unlabeled testing set. Our goal is to learn a classifier based on $\mathbf{X}^{train}$ which is robust under possibly transformed data in $\mathbf{X}^{test}$. We begin by presenting a simple classification algorithm based on a sparse reconstructive model, and continue with introducing the invariant hierarchy-based approach.

The procedure which we refer to as STL, follows for example the *Self-taught Learning via Sparse Coding* algorithm [12] (see also [13]). The idea is to learn a dictionary from an unlabeled dataset. Then the sparse coding coefficients obtained when coding elements of the labeled dataset serve as features which are fed into an SVM classifier.[1] In our implementation, the dictionary was trained with the aligned labeled data. New data is then classified with the learned dictionary and a linear SVM.[2]

Algorithm **STL**

1. $(\mathbf{D}, \alpha) \leftarrow$ **TrainDictionary**$(\mathbf{X}^{train})$ in image space.
2. Learn a classifier $\mathcal{C}$ by a linear SVM based on α.
3. $(\beta) \leftarrow$ **Lars**$(\mathbf{X}^{test}, \mathbf{D})$
4. Classify the set β by $\mathcal{C}$.

This algorithm is very effective in the case that the training and testing sets are aligned. Even though there are state-of-the-art algorithm which have preferable performance, e.g., [13], they are based on discriminative dictionary learning models in the sense of a modified version of Equation (2). In our approach on the other hand, we use a simpler reconstructive one which is based on (2). Extending the framework here presented to such discriminative models is part of our ongoing efforts.

4. HIERARCHICAL DICTIONARY LEARNING

Two main approaches are widely used to deal with invariant features in frameworks as the one here proposed. One strategy is to train the system with as many transformed pattern as possible. Alternatively, invariant features with much smaller training sets can be extracted. In the proposed method, we follow this second approach, and the invariant characteristics are implicitly captured. Input images are first transformed by a conformal mapping such that rotations and/or scaling are reduced to horizontal and/or vertical translations. Dictionaries are then trained with this data in a hierarchical fashion: The outcome associated with grouped sub blocks from one layer serve as the input to a new layer of learned dictionaries. Finally, translation invariance is accomplished by a further special hierarchical transform.

4.1. Log-Polar Mapping

Images can be represented in different spaces. Fischer [14] originally suggested that the transformation of the visual field into its neural representation is approximated by a complex logarithmic mapping $W = log(Z)$, where $Z = a + ib$ (a and b are the spatial coordinates in the image domain) and $W = \xi + i\eta$ are complex numbers that define the retinal and cortical spaces respectively. The mapping is given by: $\xi = \log\sqrt{a^2 + b^2}$ and $\eta = \tan^{-1}(b/a)$. Radial lines in the cartesian domain are mapped into vertical lines in the cortical space, and concentric circles are mapped into horizontal lines in the cortical space (Fig. 1). Rotations are therefore converted into cyclic translation along the η axis, while scalings are converted into translation along the ξ axis, Fig. 2.

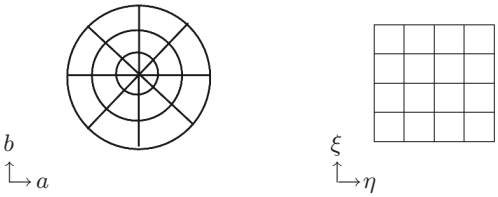

Fig. 1. **Left:** retinal space. **Right:** cortical space.

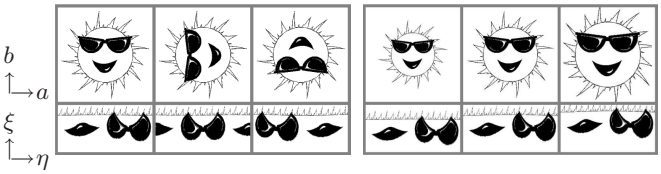

Fig. 2. **Left:** Rotations in the retinal space (*top*) are converted into cyclic shifts in the cortical space (*bottom*). **Right:** Scalings in the retinal space (*top*) are converted into shifts along the vertical axis in the cortical space (*bottom*).

4.2. Rapid Transform

We now briefly describe a non-linear hierarchical transformation which is invariant under cyclic permutation. It will be used later as we describe the proposed algorithm. The *Rapid transform*, presented in the frame below, was suggested by Reitboeck and Brody [15], and was widely used in pattern recognition algorithms, e.g., [16]. Let $\mathbf{U}$ be a vector of $M = 2^n$ elements. Then, the output vector $\mathbf{V} \leftarrow$**Rapid**$(\mathbf{U})$ is invariant under cyclic translations in the sense that for every translation $t \in [0, M]$, **Rapid**$(\mathbf{U}(i + t) \bmod M) =$ **Rapid**$(\mathbf{U}(i))$ (the proof can be found in [15]).

$\mathbf{V} \leftarrow$**Rapid**$(\mathbf{U})$

1. Let $\mathbf{U}(i)$ elements of a vector, $i = 1, \ldots, M$, $M = 2^n$, $\mathbf{V}^0 = \mathbf{U}$.
2. for $s = 1$ to n
3. $\quad \mathbf{V}^s(2i-1) = \left|\mathbf{V}^{s-1}(i) + \mathbf{V}^{s-1}(i + M/2)\right|$
4. $\quad \mathbf{V}^s(2i) = \left|\mathbf{V}^{s-1}(i) - \mathbf{V}^{s-1}(i + M/2)\right|$.

4.3. Hierarchical Invariant Algorithm (HIA)

Based on the previous sections, we describe now the proposed algorithm. Let $I^{train}[k] \in \mathbb{R}^{h' \times w'}$, $k = 1, \ldots, N^{train}$, be a set of training images, and $L^{train}[k] \in \mathbb{R}^{h \times w}$ the corresponding log-polar mapping. The dimension of the original image is not necessarily identical to the dimension of the log-polar one, since angular/radial resolution in the cortical space may be controlled. In the case of shapes images (like digits), the origin of the polar coordinate system is determined by the center of mass of the shape, otherwise the origin is the center of the image. Every log-polar image $L^{train}[k]$ is divided into $H_p \times W_p$ overlapping patches $x_i \in \mathbb{R}^{\sqrt{n} \times \sqrt{n}}$ (shown in Fig. 3). Let us now concatenate the whole patches from all training images to $\mathbf{X}^{train} = [\ldots, x_i[k], \ldots] \in \mathbb{R}^{n \times H_p W_p N^{train}}$.

The first layer dictionary $\mathbf{D}_1$ of size K_1 is now calculated based on the training set $\mathbf{X}^{train}$. The dictionary learning procedure yields also the training coefficients set $\alpha_1 \in \mathbb{R}^{K_1 \times H_p W_p N^{train}}$ corre-

[1] LIBSVM package: http://www.csie.ntu.edu.tw/ cjlin/libsvm/

[2] We used a linear SVM classifier in all the experiments reported in this paper, since we focus on the separability of the feature space.

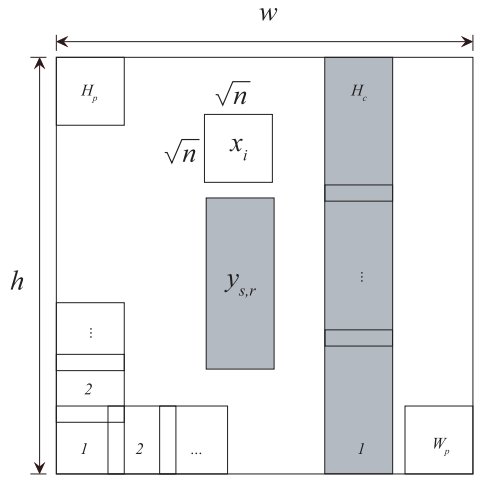

Fig. 3. Hierarchical structure of blocks. The white $\sqrt{n} \times \sqrt{n}$ patches are used in the first layer of the hierarchy, while the shadowed sub-columns are used in the second one.

sponding to the sparse code. We are now ready for the second layer of the hierarchy. Motivated by capturing the most representative structures of the data, each patch is now replaced by its most prominent atom $\mathbf{d}_l$, e.g., the atom which has the maximum α.

Let us now group some of such atoms to a unit $y_{s,r}$ which forms a sub-column. A full column accommodates H_c sub-columns, and there are total of $W_p \times H_c$ overlapping sub-columns per image (shadowed blocks in Fig. 3). Once again, we concatenate the whole sub-columns from the training images to $\mathbf{Y}^{train} = [\ldots, y_{s,r}[k], \ldots]$, $s = 1, \ldots, H_c$, $r = 1, \ldots, W_p$, $k = 1, \ldots, N^{train}$. The dictionary $\mathbf{D}_2$ of size K_2 and coefficients set α_2 are now calculated based on $\mathbf{Y}^{train}$. This process can be repeated in higher hierarchical levels, yet in the current study we simplified the model by using only two levels.

In the next two steps we obtain the desired invariance properties. From now on, we process every image k separately. Let $\alpha_2^{s,r}$ be the coefficients vector associated with sub-column $y_{s,r}$:

$$\begin{array}{cccc} \alpha_2^{H_c,1} & & \cdots & \alpha_2^{H_c,W_p} \\ \vdots & & & \vdots \\ \alpha_2^{2,1} & & & \\ \alpha_2^{1,1} & \alpha_2^{1,2} & \cdots & \alpha_2^{1,W_p} \\ \alpha_2^{1} & \alpha_2^{2} & \cdots & \alpha_2^{W_p} \end{array}$$

Scale invariance is accomplished by summing the coefficients over a column, such that $\alpha_2^r = \sum_{s=1}^{H_c} \alpha_2^{s,r}$ (shadowed row). Clearly, the sum of the coefficients is invariant under their permutations, and the vertical translations (due to scalings) are canceled.

Every image k is now represented by W_p vectors. These arrays are now fed into the Rapid transform, where every element $U(r)$ is represented by α_2^r. The outcome of the rapid transform is denoted by $\tilde{\alpha}^r$. As was explained in Section 4.2, the transformed vector is invariant under cyclic translations, and the coefficients $\tilde{\alpha}^r$ are therefore rotation invariant.

The last step is learning the SVM classifier $\mathcal{C}$. One option is to train $W_p N^{train}$ sets of $\tilde{\alpha}^r \in \mathbb{R}^{K_2}$. The other option is to group all the coefficients associated to image k, meaning that we train N^{train} sets of $[(\tilde{\alpha}^1)^T, \ldots, (\tilde{\alpha}^{W_p})^T] \in \mathbb{R}^{K_2 W_p}$. The whole learning algorithm is summarized in the HIA algorithm frame.

Given a new testing data set, invariant features $\tilde{\beta}^r$ are calculated by the above procedure using the learned D_1 and D_2. Classification is then based on grouped/non grouped $\tilde{\beta}^r$ and SVM.

The IA algorithm (see frame below) was designed to evaluate the significance of the hierarchical approach. The algorithm is similar to HIA except that the second dictionary learning stage is omitted. Experimental results support the superiority of the hierarchical model.

5. EXPERIMENTAL RESULTS

The proposed algorithms were tested with two different databases: handwritten digits and textures from multiple view points. In both algorithms we used undercomplete dictionaries which are known to be effective in classification tasks. For fair comparisons, dictionary sizes were manually optimized. All the experiments reported in this section share the same dictionary sizes. For the STL algorithm, $K = 64$. For both hierarchical levels in the suggested algorithm, the dictionary sizes were $K_1 = 256$ and $K_2 = 256$. As was explained before, the data for the STL algorithm was given in cartesian (image) space, while the data for the HIA algorithm was given in log-polar (cortical) space.

Algorithm **HIA**

1. $(\mathbf{D}_1, \alpha_1) \leftarrow$ **TrainDictionary**$(\mathbf{X}^{train})$ in log-polar space.
2. $x_i^{train} \leftarrow \mathbf{d}_{\Lambda_i}$, where $\Lambda_i = \arg\max_l \alpha_{1,i}(l)$.
3. Group atoms to sub-columns $y_{s,r}$.
4. $\mathbf{Y}^{train} = [y_{1,1}[1], \ldots, y_{s,r}[N^{train}]]$
5. $(\mathbf{D}_2, \alpha_2) \leftarrow$ **TrainDictionary**$(\mathbf{Y}^{train})$.
6. For each image k
7. Sum over columns: $\alpha_2^r = \sum_s \alpha_2^{s,r}$ (scale invariance).
8. $\tilde{\alpha}^r \leftarrow$ **Rapid** (α_2^r) (rotation invariance).
9. end
10. Learn a classifier $\mathcal{C}$ based on (non-grouped) $\tilde{\alpha}^r$, or grouped set per image $\{\tilde{\alpha}^r\}$.

Algorithm **IA** The same as HIA. Skip stages 2-5 and substitute $\alpha_2 \leftarrow \alpha_1$ in 7,8.

The re-sampled USPS [17] dataset contains 4649 training images and 4649 testing images of size 16×16 which were centered in a 24×24 matrix. Following [13], the whole image served as a patch in the STL implementation. As for the HIA algorithm, the resolution of the log-polar images was increased to 40×40, and patches of 10×10 with overlap of 8 pixels were used. The invariant features were grouped such that the SVM classifier was trained with 4649 vectors.

In the first experiment (Table 1), we trained both systems with aligned digits and then classified the aligned testing set. As expected, the STL algorithm performs a little better than the HIA (first row). This could be explained by the fact that HIA incorporates lots of overheads and interpolations during the log-polar mapping. Next, we trained the dictionaries with random rotated digits in an angles range of $[-50°, 50°]$. The testing set was randomly rotated as well. Classification results in this case were very close (second row), which makes sense due to the fact that STL learns different angles possibilities. Lastly, we added a scaling effect of $\pm 20\%$ digit size. Since the parameter K was fixed in all cases, a better result was obtained

Data Set	θ [°]	Scale	STL	HIA	IA
10 digits	0	1	**95.8**	93.7	91.4
	± 50	1	**89.8**	**88.7**	80.8
	± 50	± 0.2	83.8	**88.0**	78.1

Table 1. Classification accuracy for the digits data [%]. Both training and testing samples are randomly transformed.

Data Set	θ [°]	Scale	STL	HIA	IA
10 digits	± 50	1	59.1	**86.0**	76.9
	± 50	± 0.2	55.6	**83.6**	73.3
3 Textures	-	-	76.4	**94.7**	82.9
4 Textures	-	-	75.6	**91.2**	80.4

Table 2. Classification accuracy for the digits and texture data [%]. Only testing samples are randomly transformed for the digits (such transformations are natural in the texture dataset).

for the HIA algorithm (third row). The STL dictionary was not rich enough to accommodate as many combinations of scales and rotations in the learning set. On the other hand, the HIA dictionary learned invariant features and therefore performed better.

The next experiment is summarized in Table 2. In this case, dictionaries were learned from aligned digits only, and testing images were randomly rotated and scaled. Fig. 4 illustrates few samples from both sets. The superiority of the suggested algorithm is clear even when compared to IA algorithm. The rotation range was selected such that there would be no confusion between the digits 6 and 9. Experiments which exclude the digit 9 yield classification accuracy of 77.5% (versus 36.4% using STL) with the whole rotation range of $[0°, 360°]$.

In the second example we used a texture database [18] with high variability of scaling and viewpoints within each class (Fig. 5). The database contains 40 images of size 480×640 for every class. We used 25 images for training and 15 images for testing. For the STL algorithm, optimal patches size was 20×20 with an overlap of 12 pixels. As for the HIA algorithm, patches of 50×50 with 42 pixels overlap were used. Features vectors in this case were not grouped and every sub-image was classified independently. The results are presented in the bottom panel of Table 2. Once again, the robustness of the algorithm is verified by the tolerance under significant geometric transformations.

6. CONCLUSIONS

In this paper, we showed that a hierarchical approach to dictionary learning, combined with a cortical (log-polar) transform, plays a significant role in automatic invariant features extraction. The suggested algorithm demonstrated very promising results in the case of transformed pattern classification. In future work we would like to study additional transformations, such as affine, and also the introduction of this framework in discriminative dictionary learning [13].

Acknowledgments: The authors would like to thank Julien Mairal for the dictionary learning and Lars code, and Ignacio Ramirez and Federico Lecumberry for their extensive help. Prof. Dario Ringach inspired in part this work by asking questions about hierarchy in learning. Work partially supported by ONR, NGA, NSF, DARPA, and ARO.

Fig. 4. Samples from the training set (*left*) and transformed testing set (*right*).

Fig. 5. Samples from a textures database. Every column represents a different class.

7. REFERENCES

[1] K. Kavukcuoglu, M. Ranzato, R. Fergus, and Y. LeCun, "Learning invariant features through topographic filter maps," in *Proc. IEEE CVPR*, 2009.

[2] B. Mailh, S. Lesage, R. Gribonval, F. Bimbot, and P. Vandergheynst, "Shift-invariant dictionary learning for sparse representations: Extending k-svd," in *Proc. EUSIPCO*, 2008.

[3] B.V. Gowreesunker and A. H. Tewfik, "A shift tolerant dictionary training method," in *Proc. SPARS*, 2009.

[4] J. M. Morel and G. Yu, "ASIFT: A new framework for fully affine invariant image comparison," *SIAM Journal on Imaging Sciences*, vol. 2, 2009.

[5] J. Huang, X. Huang, and D. Metaxas, "Simultaneous image transformation and sparse representation recovery," in *Proc. IEEE CVPR*, 2008.

[6] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, "Robust object recognition with cortex-like mechanisms," *IEEE Trans. PAMI*, vol. 29, pp. 411–426, 2007.

[7] M. Ranzato, F. j. Huang, Y. Boureau, and Y. LeCun, "Unsupervised learning of invariant feature hierarchies with applications to object recognition," in *Proc. IEEE CVPR*, 2007.

[8] C. M. Pun and M. C. Lee, "Log-polar wavelet energy signatures for rotation and scale invariant texture classification," *IEEE Trans. PAMI*, vol. 25, pp. 590–603, 2003.

[9] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society B.*, vol. 58, pp. 267–288, 1996.

[10] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Annals of Statistics*, vol. 32, pp. 407–499, 2004.

[11] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *ICML*, 2009.

[12] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: Transfer learning from unlabeled data," in *ICML*, 2007.

[13] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Supervised dictionary learning," in *Adv. NIPS*, 2009.

[14] B. Fischer, "Overlap of receptive field centers and representation of the visual field in the optic tract," *Vision Res*, vol. 13, pp. 2113–2120, 1973.

[15] H. J. Reitboeck and T. P. Brody, "A transformation with invariance under cyclic permutation for applications in pattern recognition," *Information and Control*, vol. 15, pp. 130–154, 1969.

[16] H. J. Reitboeck and J. Altman, "A model for size and rotation-invariant pattern processing in the visual system," *Biological Cybernetics*, vol. 51, pp. 113–121, 1984.

[17] J. J. Hull, "A database for handwritten text recognition research," *IEEE Trans. PAMI*, vol. 16, pp. 550 – 554, 1994.

[18] S. Lazebnik, C. Schmid, and J. Ponce, "A sparse texture representation using local affine regions," *IEEE Trans. PAMI*, vol. 27, pp. 1265–1278, 2005.