HIERARCHICAL INVARIANT SPARSE MODELING FOR IMAGE ANALYSIS

Leah Bar — Tel Aviv University

Guillermo Sapiro — University of Minnesota

ABSTRACT

Sparse representation theory has been increasingly used in signal processing and machine learning. In this paper we introduce a hierarchical sparse modeling approach which integrates information from the image patch level to derive a mid-level invariant image and pattern representation. The proposed framework is based on a hierarchical architecture of dictionary learning for sparse coding in a cortical (log-polar) space, combined with a novel pooling operator which incorporates the Rapid transform and max pooling to attain rotation and scale invariance. The invariant sparse representation of patterns here presented- can be used in different object recognition tasks. Promising results are obtained for three applications – 2D shapes classification, texture recognition and object detection.

Index Terms— Feature extraction, dictionary learning, sparse coding, hierarchical models, invariant representation

1. INTRODUCTION

Feature extraction is an essential pre-processing step for pattern recognition and machine learning problems, where the ultimate goal is to capture the internal and significant structure of the pattern for further processing and analysis. While low-level descriptors represent local characteristics of the pattern, mid-level features address their structured configuration, where as the complexity of the feature increases, so are their invariance and compactness properties [1].

Leading recognition models consist of coding local features, spatial pooling and classification. Boureau *et al.* [2] had recently presented a comparative study of different architectures and combinations of these modules, reporting impressive classification results on several benchmarks. Their results state that sparse coding over learned dictionaries paired with max pooling operator outperform other module combinations (this is further supported by the results in recent PASCAL competitions). Nevertheless, sparse codes are not inherently invariant under transformations such as translation, scaling, and rotation. Yang *et al.* [3], suggested a translation-invariant sparse coding supervised classification algorithm. The sparse codes serve as the local features, and a pyramid is built by the max pooling operator over growing sub-regions. This algorithm, however, was designed for relatively small distortions.

In this paper we introduce a framework for representing mid-level rotation and scale invariant features. While rotation and scaling are planar transformations, invariant features are calculated over sub-regions of the image/object, since small regions on the surfaces of 3D objects are approximately planar. Our scheme is composed of sparse codes with learned dictionaries over a conformal (log-polar) mapping of the data, such that rotated and scaled patterns are converted into shifted patterns in the new space. The sparse codes are then integrated via a novel pooling operator which incorporates max pooling and Rapid transform. First, we calculate scale invariant features within sub-regions of the image by means of a single level max pooling operator. The shifts due to rotations are dealt with via the Rapid transform. We then integrate the invariant features associated with different sub-regions of the image via a hierarchical structure, and feed them into an SVM classifier. A preliminary version of the suggested algorithm was recently introduced by the authors [4]. Nevertheless, the pooling hierarchical architecture here proposed is significantly different (one dictionary with max pooling over regions vs. two dictionaries with not such pooling in the previous version), and yields improved experimental results (e.g. in the case of texture classification, 25 vs. 4 classes with $\sim 90\%$ accuracy).

Our approach is closely related to [5], where log-polar images were represented by invariant wavelet packets. Yet, in our work we incorporate a hierarchical architecture and learned dictionaries, which were proved to outperform pre-defined bases. Our work is also related to [3] in the sense that we adopted their sparse codes max pooling operator along the hierarchy. In contrast with the recent works on combining deep learning with sparse coding, here the dictionary is learned at a single level. The suggested approach is generic and suited for data of very different nature. We demonstrate it with three different computer vision recognition problems: (transformed) digits classification, recognition of textures having multiple viewpoints and non-rigid deformations, and object detection on real images. For each example we compare with state-of-the-art algorithms designed for the specific task. Promising preliminary results verify the potential of the proposed framework.

2. BACKGROUND AND NOTATIONS

In sparse modeling, 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 $x \cong \mathbf{D}\alpha$. 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

$$\mathcal{F}(\alpha; \mathbf{D}) = \|x - \mathbf{D}\alpha\|_2^2 + \lambda\|\alpha\|_1, \tag{1}$$

where $\lambda \in \mathbb{R}$ is a regularization constant, and the code α is the minimizer of the functional. This formulation and its variants are often referred to as *basis pursuit* or *Lasso* [6]. The optimization in this work was carried out using the LARS [7] algorithm.

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 the minimizers of

$$\mathcal{G}(\mathbf{D}, \alpha) = \sum_{i=1}^{m} \|x_i - \mathbf{D}\alpha_i\|_2^2 + \lambda\|\alpha_i\|_1. \tag{2}$$

The optimization is performed by alternate minimization w.r.t. α and $\mathbf{D}$. Detailed description of both algorithms can be found for example in [8].

3. SPARSE AND INVARIANT MID-LEVEL FEATURE EXTRACTION

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 patterns 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 rotation and scaling are reduced to horizontal and vertical translations respectively. Dictionaries are then trained with this data, and the sparse codes are pooled to form the invariant representation of a pattern within sub-regions of the image . The features in the different sub-regions are then pooled again and form an integrated global representation of the image and/or pattern.

3.1. Log-Polar Mapping

Images can be represented in different spaces. Fischer [9] 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 log-polar 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 log-polar space, and concentric circles are mapped into horizontal lines in the log-polar space. Rotations are therefore converted into cyclic translations along the η axis, while scalings are converted into translations along the ξ axis, Fig. 1. We will denote the log-polar transform as $\mathcal{L} : \mathbb{R}^{h' \times w'} \rightarrow \mathbb{R}^{h \times w}$, where h and w represent the scale and angular resolutions respectively, while $h' \times w'$ are the original image dimensions.

3.2. 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* $\mathcal{R}$, presented below, was suggested by Reitboeck and Brody [10], and was widely used in pattern recognition algorithms, e.g., [11]. Let $\mathbf{U}$ be a vector of $M = 2^n$ elements. Then, the output vector $\mathbf{V} = \mathcal{R}(\mathbf{U})$ is invariant under cyclic translations in the sense that for every translation $t \in [0, M]$, $\mathcal{R}(\mathbf{U}(i + t) \bmod M) = \mathcal{R}(\mathbf{U}(i))$ (the proof can be found in [10]).

Algorithm $\mathbf{V} = \mathcal{R}(\mathbf{U})$ (Rapid Transform)

---

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

3.3. Hierarchical Invariant Algorithm (HIA)

We are now ready to introduce our hierarchical invariant feature extraction algorithm. Let $\mathbf{Y} \in \mathbb{R}^{h' \times w'}$ be the input image. We decompose $\mathbf{Y}$ into $H_b \times W_b$ overlapping blocks such that $Y^{mn} := R_{mn}\mathbf{Y}$, where R_{mn} is an operator that extracts the mn block. We next apply the log-polar transform to every block such that

$$X^{mn} := \mathcal{L}\left(Y^{mn}\right) \in \mathbb{R}^{h \times w}.$$

[1] These elements are generic, they will later on become vectors.

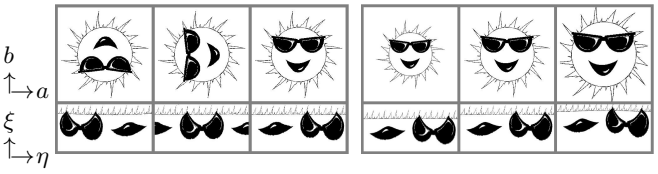

Fig. 1. **Left:** Rotations in the retinal space (*top*) are converted into cyclic shifts in the cortical space (*bottom*). **Right:** Scalings in the retinal space (*top*) are converted into shifts along the vertical axis in the cortical space (*bottom*).

Now, every block X^{mn} is divided into $H \times W$ overlapping patches such that $x_{ij}^{mn} := R_{ij}X^{mn} \in \mathbb{R}^{\sqrt{n} \times \sqrt{n}}$ (Fig. 2). A dictionary $\mathbf{D} \in \mathbb{R}^{n \times K}$ is trained using (2) with the patches x_{ij}^{mn} of the training set. The sparse codes $\alpha_{ij}^{mn} \in \mathbb{R}^K$ are then given by

$$\alpha_{ij}^{mn} = \arg\min_{\bar{\alpha}_{ij}^{mn}} \left\|x_{ij}^{mn} - \mathbf{D}\bar{\alpha}_{ij}^{mn}\right\|_2^2 + \lambda\|\bar{\alpha}_{ij}^{mn}\|_1,$$

where $i = 1, \ldots, HH_b$, and $j = 1, \ldots, WW_b$. In the next two

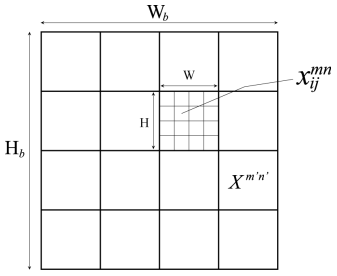

Fig. 2. Blocks and patches setting

steps we obtain the desired invariance properties. Scale invariance is accomplished by the first type of hierarchy: the max pooling operator along the columns (recall that the log-polar shifts scalings in this direction). Let $\{\gamma_i\}_{i=1}^N \in \mathbb{R}^M$ a set of N vectors, each with M elements, where γ_{ij} stands for vector i at index j. Following [3], let us define the max pooling operator as

$$\mathcal{P}_{max}^N\{\gamma_i\}(j) := \max\left\{|\gamma_{1j}|, |\gamma_{2j}|, \ldots, |\gamma_{Nj}|\right\}, \quad j = 1, \ldots, M.$$

Then, the scale-invariant feature vectors per block take the form

$$\beta_j^{mn} := \mathcal{P}_{max}^H\{\alpha_i^{mn}\}(j) \in \mathbb{R}^K, \;\; j = 1, \ldots, W.$$

$$\begin{array}{cccc}
\alpha_{H1}^{mn} & & \cdots & \alpha_{HW}^{mn} \\
\vdots & & & \vdots \\
\alpha_{21}^{mn} & & & \\
\alpha_{11}^{mn} & \alpha_{12}^{mn} & \cdots & \alpha_{1W}^{mn} \\
\hline
\beta_1^{mn} & \beta_2^{mn} & \ldots & \beta_W^{mn}
\end{array}$$

Clearly, the maximal coefficients associated with overlapping patches along the columns are invariant under their permutations, and the vertical translations (due to scalings) are discarded (thereby obtaining scale invariance).

Every block X^{mn} is now represented by $\beta_j^{mn}, j = 1, \ldots, W$ vectors. These vectors are now fed into the Rapid transform, where every element $U(j)$ is represented by $\beta_j^{mn} \in \mathbb{R}^K$. The outcome of the Rapid transform, denoted by $\tilde{\beta}_j^{mn}$, is therefore now both scale

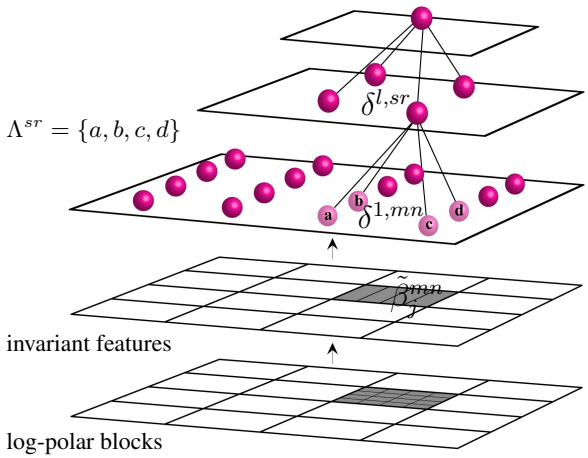

Fig. 3. Algorithm sketch for hierarchical invariant feature extraction.

and rotation invariant since the cyclic horizontal translations in the log-polar space (due to rotations) are now eliminated.

Finally, the feature vector associated with block X^{mn} is given by the concatenation of the scale and rotation invariant features, obtaining $\delta^{1,mn} = [\tilde{\beta}^{mn}_1, \ldots, \tilde{\beta}^{mn}_W] \in \mathbb{R}^{KW}$. The superscript 1 designates the bottom level of the hierarchical structure. The feature vector in level l of the (second-type) hierarchy is given by the max pooling of the corresponding features in the level below it,

$$\delta^{l,sr} = \mathcal{P}^{\#\Lambda_{sr}}_{max}\{\delta^{l-1,\Lambda^{sr}}\},$$

where Λ^{sr} is the index set of the cells in level $l-1$ that cover cell sr in level l (Fig. 3). The feature set of all blocks at level l is denoted by $\{\delta^l\}$, and the final set that is fed into the SVM classifier is composed of one or multiple levels e.g. $\mathcal{F} = \{\delta^1\}$, or $\mathcal{F} = \{\delta^1, \delta^2, \delta^3\}$.

4. EXPERIMENTAL RESULTS

The proposed algorithm is now tested with three datasets reflecting data with different nature. In the first example we present 2D shapes classification of handwritten digits with *significant* rotation and scale transformations. Next, we classify textures from multiple view points, and finally we perform object detection in real scenes. For each example we compare with state-of-the-art algorithms tailored to each specific task. We used a *linear* SVM classifier[2] in all the experiments reported in this paper, since we focus on the separability of the designed feature space. More advanced classifiers might lead to further improvements.

4.1. Handwritten Digits Recognition

The re-sampled USPS [16] database contains 4649 training images and 4649 testing images of size 16×16, centered in a 24×24 matrix, see Fig. 4. The resolution of the log-polar images was increased to 40×40 with $W(=16) \times H(=16)$ patches. There was one block ($W_b = 1, H_b = 1$), and one hierarchical level ($\mathcal{F} = \{\delta^1\}$) with patches of 10×10 with overlap of 8 pixels. The dictionary size was set to $K = 256$. In the first experiment, we trained the system with aligned digits and then classified the aligned testing set. We compared our results to two leading algorithms: SCSM (Standard Classification with Sparse Modeling), which follows for example the *Self-taught Learning via Sparse Coding* algorithm [13] (see also [12]), and MD (Multiple Dictionaries) [12, 14]. As expected, the SCSM and MD algorithms performed slightly better than the proposed HIA (first row in Table 1). This could be explained by the fact that HIA incorporates lots of overhead and interpolations during the log-polar mapping, not needed if the data will not have transformations. Next, we trained the dictionary with random rotated digits in an angle range of $[-50°, 50°]$ and scaling effect of $\pm 20\%$ digit size (second row). The testing set was randomly rotated and scaled as well. Classification results in this case are very close (second row) with a little advantage to MD. This makes sense due to the fact that SCSM and MD learn different possible angles and scales. The next experiment, which is the critical one, is summarized in row 3 of Table 1. In this case, the dictionary was learned from aligned digits only, and testing images were randomly rotated and scaled (Fig. 4). In this case, the proposed HIA algorithm significantly outperforms SCSM and MD, which verifies the proposed learned invariant representation of the digits.

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

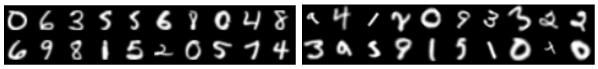

Fig. 4. Samples from the training set (*left*) and the transformed testing set (*right*),

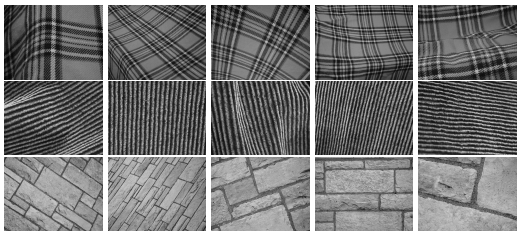

Fig. 5. Samples from the UIUC textures database.

4.2. Texture Recognition

Another application we tested is texture recognition. The UIUCTex dataset [15] contains 25 texture classes with 40 images per class. Textures are viewed under significant scale and viewpoint changes. Furthermore, the dataset includes non-rigid deformations, illumination changes, and viewpoint-dependent appearance variations. Fig. 5 shows three classes with five sample images per class, where the first and second rows depict folded fabrics. In these cases, the approximation of planar transformation can be used only for sub-blocks of the image. We used 20 images for training and 20 images for testing and two hierarchical levels such that the feature set was $\mathcal{F} = \{\delta^2\}$. Every image was partitioned into 4 blocks ($W_b = 2, H_b = 2$), where the block size was set to 170×170 with overlap of 30 pixels. The log-polar image within each block was divided into $W(=32) \times H(=16)$ patches, where patch size was 10×10 with overlap of 5 pixels. The dictionary size was set to $K = 441$. The results are shown in the last row of Table 1. Table 2 presents classification results of a flat (one) block, the outcome of level 1 (4 blocks, $\mathcal{F} = \{\delta^1\}$) and level 2 (4 blocks, $\mathcal{F} = \{\delta^2\}$), where a global max pooling was taken from the four blocks. The importance of the hierarchy is clearly observed. The proposed approach yields promising results even when compared to [15] that is based on local affine regions (LAR) and was specifically designed for texture classification. Poor results were obtained by SCSM and MD in this case (Table 1).

	Database		θ [°]	Scale	SCSM [12, 13]	MD [14]	HIA	LAR [15]
1	10 digits	train	0	1				
		test	0	1	96.6	97.7	95.7	
2		train	± 50	± 0.2				
		test	± 50	± 0.2	89.8	95.2	93.4	
3		train	0	1				
		test	± 50	± 0.2	50.4	50.9	87.4	
4	25 Textures				26.3	14.6	90.0	96.03

Table 1. Classification accuracy for several datasets in [%].

	One block	$l = 1$	$l = 2$
25 Textures	70.8	78.9	90.0

Table 2. Texture classification accuracy in [%].

	HIA	VOC [17]	Felzenszwalb *et al.* [18]
Cat	$0.163, 0.185$	$0.151, 0.160$	$0.188, 0.190, 0.236$
Dog	$0.136, 0.139$	$0.113, 0.118$	$0.151, 0.153, 0.185$

Table 3. Average precision of object detection.

4.3. Object Detection

The last application is object detection, where we tested our algorithm on PASCAL VOC2006 database [17]. We focused on Dog and Cat objects which are mostly characterized by special fur/hair texture with non-rigid shape (different pose, legs and head location). The Cat set consists of 386 training images and 388 testing images, while the Dog set consists of 365 training images and 370 testing images. The goal is to predict the bounding box of the object, where to be considered a correct detection the area of overlap a_o between the predicted bounding box B_p and the ground truth bounding box B_{gt} must exceed 50%, with $a_o = \text{area}(B_p \cap B_{gt})/\text{area}(B_p \cup B_{gt})$.

In the training phase we decomposed the (given) bounding box of the objects into blocks of size 120×120 with overlap of 10 pixels. The log-polar image within each block was divided into $W(=4) \times H(=2)$ patches, where patch size was 15×15 with overlap of 5 pixels. Blocks outside the bounding box were considered background. The dictionary size was set to $K = 225$. Depending on object size, the feature set consists of $\mathcal{F} = \{\delta^2, \delta^3, \delta^4\delta^5\}$, where $\delta^2, \delta^3, \delta^4$ pools 1×2, 2×2 and 3×3 blocks from level 1 respectively. δ^5 is the global max pooling from all blocks of level 1. In the testing phase, images were scanned by sliding blocks of different size. Every block was classified as object/background using a linear SVM . The bounding box of the union of connected detected blocks were marked as the predicted box (Fig. 6). Average precision using the suggested algorithm is $0.163, 0.185$, for the Cat and $0.136, 0.139$ for the Dog (using different combinations of sliding blocks size), which are comparable to reported results of detection algorithms[3] (Table 3).

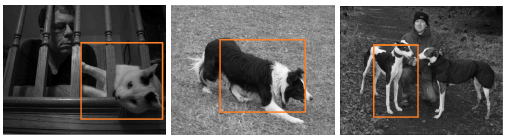

Fig. 6. Examples of detected objects

5. CONCLUSIONS

In this paper we developed a framework for sparse representation via a hierarchical architecture, where local features are integrated to capture the spatial structure of patterns of different types. Using these concepts we developed a rotation and scale invariant recognition algorithm achieving comparable to state-of-the-art results in a number of applications and standard datasets. In future work we would like to study additional invariant transformations and robustness to illumination changes.

[3]These results of precision-recall [17] are not available in subsequent PASCAL reports.

6. REFERENCES

[1] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *ICML*, 2009.

[2] Y. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *CVPR*, 2010.

[3] J. Yang, K. Yu, and T. Huang, "Supervised translation-invariant sparse coding," in *CVPR*, 2010.

[4] L. Bar and G. Sapiro, "Hierarchical dictionary learning for invariant classification," in *ICASSP*, 2010.

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

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

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

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

[9] 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.

[10] 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.

[11] 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.

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

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

[14] P. Sprechmann and G. Sapiro, "Dictionary learning and sparse coding for unsupervised clustering," in *ICASSP*, 2010.

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

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

[17] M. Everingham, A. Zisserman, C. Williams, and L. Van Gool, "The pascal visual object classes challenge 2006 (voc2006) results," .

[18] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *PAMI*, vol. 32, pp. 550 – 554, 2010.