Convolutional Sparse Coding-based Image Decomposition

He Zhang
he.zhang92@rutgers.edu
Vishal M.Patel
http://www.rci.rutgers.edu/~vmp93

Electrical and Computer Engineering
Rutgers, The State University of New Jersey NJ, USA

Abstract

We propose a novel sparsity-based method for cartoon and texture decomposition based on Convolutional Sparse Coding (CSC). Our method first learns a set of generic filters that can sparsely represent cartoon and texture type images. Then using these learned filters, we propose a sparsity-based optimization framework to decompose a given image into cartoon and texture components. By working directly on the whole image, the proposed image separation algorithm does not need to divide the image into overlapping patches for leaning local dictionaries. Extensive experiments show that the proposed method performs favorably compared to state-of-the-art image separation methods.

1 Introduction

In many practical applications such as biomedical imaging, remote sensing, biometrics and astronomy, images can be modeled as superpositions of cartoon (i.e. piecewise smooth) and texture structures [11, 20]. For instance, in remote sensing, a synthetic aperture radar image can be modeled as a superposition of the ground reflectivity field (cartoon) with multiplicative speckle (texture) [2, 14]. Similarly, detecting cracks on concrete structures requires one to model the image as a superposition of background texture with a crack component [8]. In these applications, a common task is to separate such an image into two individual images - one containing the cartoon part and the other containing the texture part.

In recent years, methods based on sparse representation and ℓ_1-minimization have been developed to deal with this problem. In particular, an approach called Morphological Component Analysis (MCA) was proposed in [17] for separating different geometrical components from a given image under the assumption that an image is the linear mixture of several morphological components. In this method, it is assumed that different morphological components are sufficiently distinct and that each one can be sparsely represented using a specific dictionary but not in the other ones. The performance of MCA depends on the dictionaries chosen for representing cartoon and texture components. In practice, dictionaries corresponding to the Discrete Cosine Transform (DCT) or the Discrete Sine Transform (DST) are used to represent the texture component as their atoms are oscillatory in nature and dictionaries corresponding to wavelet, curvelet, shearlet or contourlet are used to represent the piecewise smooth component as they represent geometric features such as edges well. The

Pages 125.1-125.11
DOI: https://dx.doi.org/10.5244/C.30.125

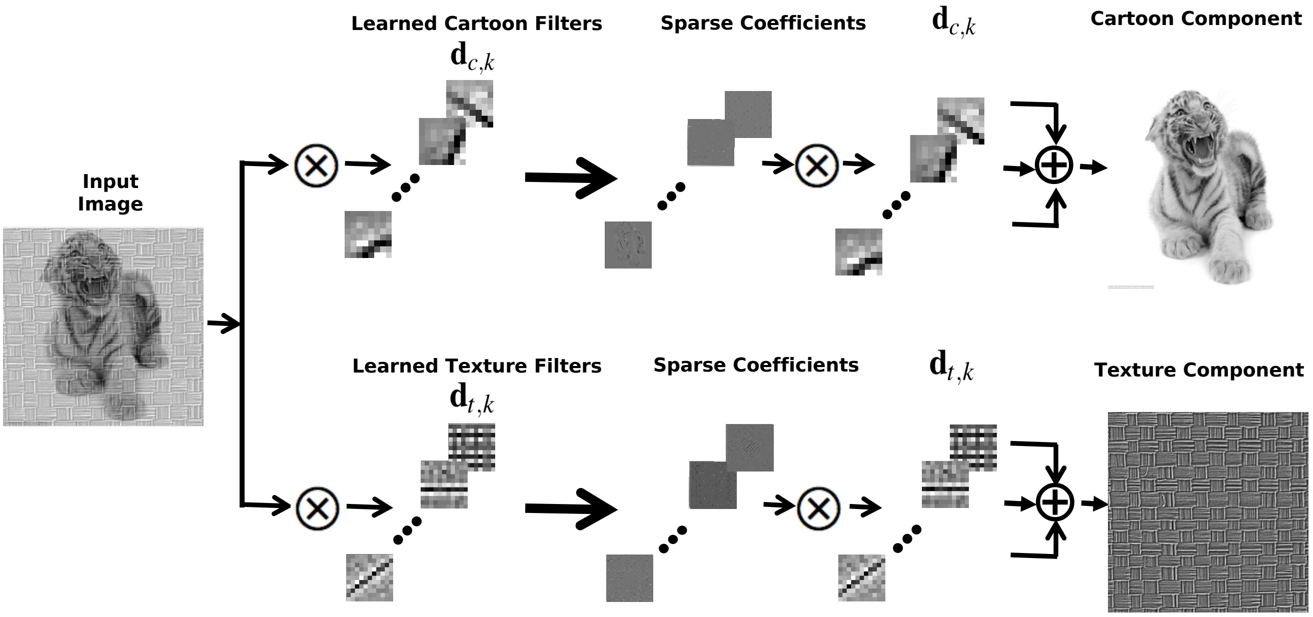


Figure 1: An overview of the proposed CSCD method for image separation.

MCA algorithm has been very successful in separating various components in many practical applications [3, 14, 17, 19]. However, one of the limitations of this approach is that complicated textures found in many practical applications can not be modeled by DCT or DST dictionaries. As a result, it tends to produce a poor separation.

It has been observed that learning a dictionary directly from training rather than using a predetermined dictionary such as DCT or wavelet, usually leads to better representation and hence can provide improved results in many image processing and classification problems [1, 7]. One such dictionary learning-based method for image separation was proposed in [15]. However, this method only learns local dictionaries for the texture component and uses predetermined global dictionaries such as wavelet or curvet for the cartoon component. One of the limitations of this method is that it is computationally very expensive and extremely slow [15]. Furthermore, most dictionary leaning approaches are patch-based and features learned with these methods often contain shifted versions of the same features [7]. To deal with this issue, Convolutional Sparse Coding (CSC) methods have been introduced in which shift invariance is directly modeled in the objective [5, 10, 22, 24]. CSC has been demonstrated to have important applications in a wide range of computer vision and image processing problems [9, 13, 23].

Motivated by the success of CSC methods in learning adaptive and efficient representations, we present an image separation method based on CSC. Using multiple cartoon and texture training images, we first learn the convolutional filters corresponding to these components. Then, using the learned filters, we develop an MCA type of algorithm to separate the texture and cartoon components from a given image. Figure 1 gives an overview of the proposed Convolutional Sparse Coding-based image Decomposition (CSCD) method.

Rest of the paper is organized as follows. In Section 2, we give a brief background on sparsity-based image separation and convolutional sparse coding. Details of the proposed CSCD algorithm are given in Section 3. Experimental results are presented in Section 4 and Section 5 concludes the paper with a brief summary and discussion.

2 Background

In this section, we give a brief background on sparsity-based image separation and convolutional sparse coding.

Image Separation. Let $\mathbf{y}$ be a lexicographically ordered vector of size N^2 corresponding to an image $Y \in \mathbb{R}^{N\times N}$. Assume that $\mathbf{y}$ is a superposition of two different images

$$\mathbf{y} = \mathbf{y}_c + \mathbf{y}_t, \tag{1}$$

where $\mathbf{y}_c$ and $\mathbf{y}_t$ are the cartoon or piecewise smooth component and the texture component of $\mathbf{y}$, respectively. We assume that $\mathbf{y}_c$ is sparse in a dictionary represented in a matrix form as $\mathbf{D}_c \in \mathbb{R}^{N^2\times M_c}$, and similarly, $\mathbf{y}_t$ is sparse in a dictionary represented in a matrix form as $\mathbf{D}_t \in \mathbb{R}^{N^2\times M_t}$. The dictionaries $\mathbf{D}_c$ and $\mathbf{D}_t$ are chosen such that they provide sparse representations of piecewise smooth and texture components, respectively. That is, we assume there are coefficient vectors $\mathbf{x}_c \in \mathbb{R}^{M_c}$ and $\mathbf{x}_t \in \mathbb{R}^{M_t}$ so that $\mathbf{y}_c = \mathbf{D}_c\mathbf{x}_c$ and $\mathbf{y}_t = \mathbf{D}_t\mathbf{x}_t$. The sparsity assumption means that when the coefficients are ordered in magnitude, they decay rapidly. Then, one can estimate the components $\mathbf{y}_c$ and $\mathbf{y}_t$ via $\mathbf{x}_c$ and $\mathbf{x}_t$ by solving the following optimization problem [17]

$$\hat{\mathbf{x}}_c, \hat{\mathbf{x}}_t = \arg\min_{\mathbf{x}_c,\mathbf{x}_t} \frac{1}{2}\|\mathbf{y} - \mathbf{D}_c\mathbf{x}_c - \mathbf{D}_t\mathbf{x}_t\|_2^2 + \lambda_c\|\mathbf{x}_c\|_1 + \lambda_t\|\mathbf{x}_t\|_1 + \beta TV(\mathbf{D}_c\mathbf{x}_c), \tag{2}$$

where TV is the total variation (i.e. sum of the absolute variations in the image) and for an N-dimensional vector $\mathbf{x}$, $\|\cdot\|_q$ denotes the ℓ_q-norm, $0 < q < \infty$, defined as $\|\mathbf{x}\|_q = \left(\sum_{i=1}^N |x_i|^q\right)^{\frac{1}{q}}$. Here, λ_c, λ_t and β are positive regularization parameters. The two components are the corresponding representations of the two parts and can be obtained by $\hat{\mathbf{y}}_c = \mathbf{D}_c\hat{\mathbf{x}}_c$ and $\hat{\mathbf{y}}_t = \mathbf{D}_t\hat{\mathbf{x}}_t$. Various methods have been developed in the literature to obtain the solution of (2) [3, 17].

Convolutional Sparse Coding. In CSC, given a set of M training samples $\{\mathbf{y}_m\}_{i=1}^M$, the objective is to learn a set of convolutional filters $\{\mathbf{d}_k\}_{i=1}^K \in \mathbb{R}^{P\times P}$ by solving the following optimization problem

$$\begin{aligned} &\arg\min_{\mathbf{d},\mathbf{x}} \quad \frac{1}{2}\sum_{m=1}^{M}\left\|\mathbf{y}_m - \sum_{k=1}^{K}\mathbf{d}_k * \mathbf{x}_{m,k}\right\|_2^2 + \lambda\sum_{m=1}^{M}\sum_{k=1}^{K}\|\mathbf{x}_{m,k}\|_1 \\ &\text{subject to} \quad \|\mathbf{d}_k\|_2^2 \le 1 \quad \forall k \in \{1,\cdots,K\}, \end{aligned} \tag{3}$$

where $\mathbf{x}_{m,k} \in \mathbb{R}^{N\times N}$ are the sparse coefficients that approximate the data $\mathbf{y}_m$ when convolved with the corresponding filters $\mathbf{d}_k$ of fixed support. Here, $*$ represents the 2-D convolution operator and λ is a positive regularization parameter. Several methods have been proposed in the literature for solving the above optimization problem. For instance, [5] introduced a Fourier domain Alternating Direction Method of Multipliers (ADMM) [4] framework for solving the CSC problem. In [10], a flexible framework that can tackle proper boundary conditions was proposed for solving the CSC optimization problem. More recently, [22], [21] developed an efficient method that jointly uses the space and Fourier domains to solve the CSC problem. In this paper, we adapt the method proposed in [22] for learning the convolutional filters due to its simplicity and efficiency.

3 Proposed Approach

Following the mixture model in (1), given $\mathbf{y}$ our goal is to separate it into $\mathbf{y}_c$ and $\mathbf{y}_t$. Assume that we have already learned the convolutional filters corresponding to $\mathbf{y}_c$ and $\mathbf{y}_t$ by solving the CSC problem (3) for the cartoon and the texture components separately. That is, we have learned $\{\mathbf{d}_{c,k}\}_{k=1}^{K_c}$ and $\{\mathbf{d}_{t,k}\}_{k=1}^{K_t}$ such that $\mathbf{y}_c = \sum_{k=1}^{K_c} \mathbf{d}_{c,k} * \mathbf{x}_{c,k}$ and $\mathbf{y}_t = \sum_{k=1}^{K_t} \mathbf{d}_{t,k} * \mathbf{x}_{t,k}$, where $\mathbf{x}_{c,k}$ and $\mathbf{x}_{t,k}$ are the sparse coefficients that approximate $\mathbf{y}_c$ and $\mathbf{y}_t$ when convolved with the filters $\mathbf{d}_{c,k}$ and $\mathbf{d}_{t,k}$, respectively. We propose to estimate $\mathbf{y}_c$ and $\mathbf{y}_t$ via $\mathbf{x}_{c,k}$ and $\mathbf{x}_{t,k}$ by solving the following CSC-based optimization problem

$$\begin{aligned}\hat{\mathbf{x}}_{c,k}, \hat{\mathbf{x}}_{t,k} = \arg\min_{\mathbf{x}_{c,k},\mathbf{x}_{t,k}} \frac{1}{2}\left\| \mathbf{y} - \sum_{k=1}^{K_c} \mathbf{d}_{c,k} * \mathbf{x}_{c,k} - \sum_{k=1}^{K_t} \mathbf{d}_{t,k} * \mathbf{x}_{t,k} \right\|_2^2 + \lambda_c \sum_{k=1}^{K_c} \left\|\mathbf{x}_{c,k}\right\|_1 \\ + \lambda_t \sum_{k=1}^{K_t} \left\|\mathbf{x}_{t,k}\right\|_1 + \beta TV\left(\sum_{k=1}^{K_c} \mathbf{d}_{c,k} * \mathbf{x}_{c,k} \right).\end{aligned} \tag{4}$$

Once, $\mathbf{x}_{c,k}$ and $\mathbf{x}_{t,k}$ are estimated, the two components can be obtained by $\hat{\mathbf{y}}_c = \sum_{k=1}^{K_c} \mathbf{d}_{c,k} * \hat{\mathbf{x}}_{c,k}$ and $\hat{\mathbf{y}}_t = \sum_{k=1}^{K_t} \mathbf{d}_{t,k} * \hat{\mathbf{x}}_{t,k}$.

3.1 Optimization

For simplicity we discard the TV part in (4) for the discussion given here. The resulting optimization problem can be solved iteratively over $\mathbf{x}_{c,k}$ and $\mathbf{x}_{t,k}$.

Update step for $\mathbf{x}_{c,k}$. In this step, we assume that $\mathbf{x}_{t,k}$ is fixed. As a result, the following problem needs to be solved

$$\hat{\mathbf{x}}_{c,k} = \arg\min_{\mathbf{x}_{c,k}} \frac{1}{2}\left\| \mathbf{y} - \sum_{k=1}^{K_c} \mathbf{d}_{c,k} * \mathbf{x}_{c,k} - \sum_{k=1}^{K_t} \mathbf{d}_{t,k} * \mathbf{x}_{t,k} \right\|_2^2 + \lambda_c \sum_{k=1}^{K_c} \left\|\mathbf{x}_{c,k}\right\|_1 . \tag{5}$$

Since, $\mathbf{x}_{t,k}, \mathbf{d}_{t,k}$ and $\mathbf{d}_{c,k}$ are fixed, (5) is essentially a sparse coding problem which can be solved using the DFT-based ADMM algorithm presented in [21].

Update step for $\mathbf{x}_{t,k}$. For a fixed $\mathbf{x}_{c,k}$, we have to solve the following problem to obtain $\mathbf{x}_{t,k}$

$$\hat{\mathbf{x}}_{t,k} = \arg\min_{\mathbf{x}_{t,k}} \frac{1}{2}\left\| \mathbf{y} - \sum_{k=1}^{K_c} \mathbf{d}_{c,k} * \mathbf{x}_{c,k} - \sum_{k=1}^{K_t} \mathbf{d}_{t,k} * \mathbf{x}_{t,k} \right\|_2^2 + \lambda_t \sum_{k=1}^{K_t} \left\|\mathbf{x}_{t,k}\right\|_1 . \tag{6}$$

Again, this problem can be solved using the ADMM method presented in [21].

TV **correction.** Once the sparse coefficients have been estimated, we apply the TV correction on the recovered cartoon component. We replace the TV correction term by a redundant Haar wavelet-based shrinkage estimate as this seems to give the best results. The shrinkage is applied as follows

$$\hat{\mathbf{y}}_c = \mathbf{H}\mathcal{S}_\beta\left(\mathbf{H}^T\left(\sum_{k=1}^{K_c} \mathbf{d}_{c,k} * \hat{\mathbf{x}}_{c,k}\right)\right), \tag{7}$$

where $\mathcal{S}_\beta(\mathbf{x}) = \mathrm{sign}(\mathbf{x})(|\mathbf{x}| - \beta)_+$ is the element-wise soft-thresholding operator with threshold β. This adjustment is applied only to the piecewise smooth component to control the

Algorithm 1: The CSCD Algorithm for Image Decomposition.

```
Input: $\{\mathbf{d}_{c,k}\}_{k=1}^{K_c}$, $\{\mathbf{d}_{t,k}\}_{k=1}^{K_t}$, $\mathbf{y}$, $\lambda_c$, $\lambda_t$, $L$
    Initialization
    for $i = 1 : L$
        Obtain $\hat{\mathbf{x}}_{c,k}$ by solving (5).
        Obtain $\mathbf{y}_c$ by applying the Haar threshold in (7).
        Use $\mathbf{y}_c$ to replace $\sum_{k=1}^{K_c} \mathbf{d}_{c,k} * \hat{\mathbf{x}}_{c,k}$ in (6).
        Obtain $\hat{\mathbf{x}}_{t,k}$ by solving (6).
    end for
    $\hat{\mathbf{y}}_c$=$\mathbf{y}_c$;
    $\hat{\mathbf{y}}_t = \sum_{k=1}^{K_{l,t}} \mathbf{d}_{t,k} * \hat{\mathbf{x}}_{t,k}$,
Output: $\hat{\mathbf{y}}_c$, $\hat{\mathbf{y}}_t$
```

ringing artifacts near the edges caused by the oscillations of the atoms in the dictionary $\{\mathbf{d}_{c,k}\}_{k=1}^{K_c}$. So the edges and contours are better kept in the cartoon part and highly oscillatory cannot be captured by the TV correction. The same adjustment was used in [16, 17] and the substitution was partially motivated by observing the connection between TV and the Haar wavelet given in [18].

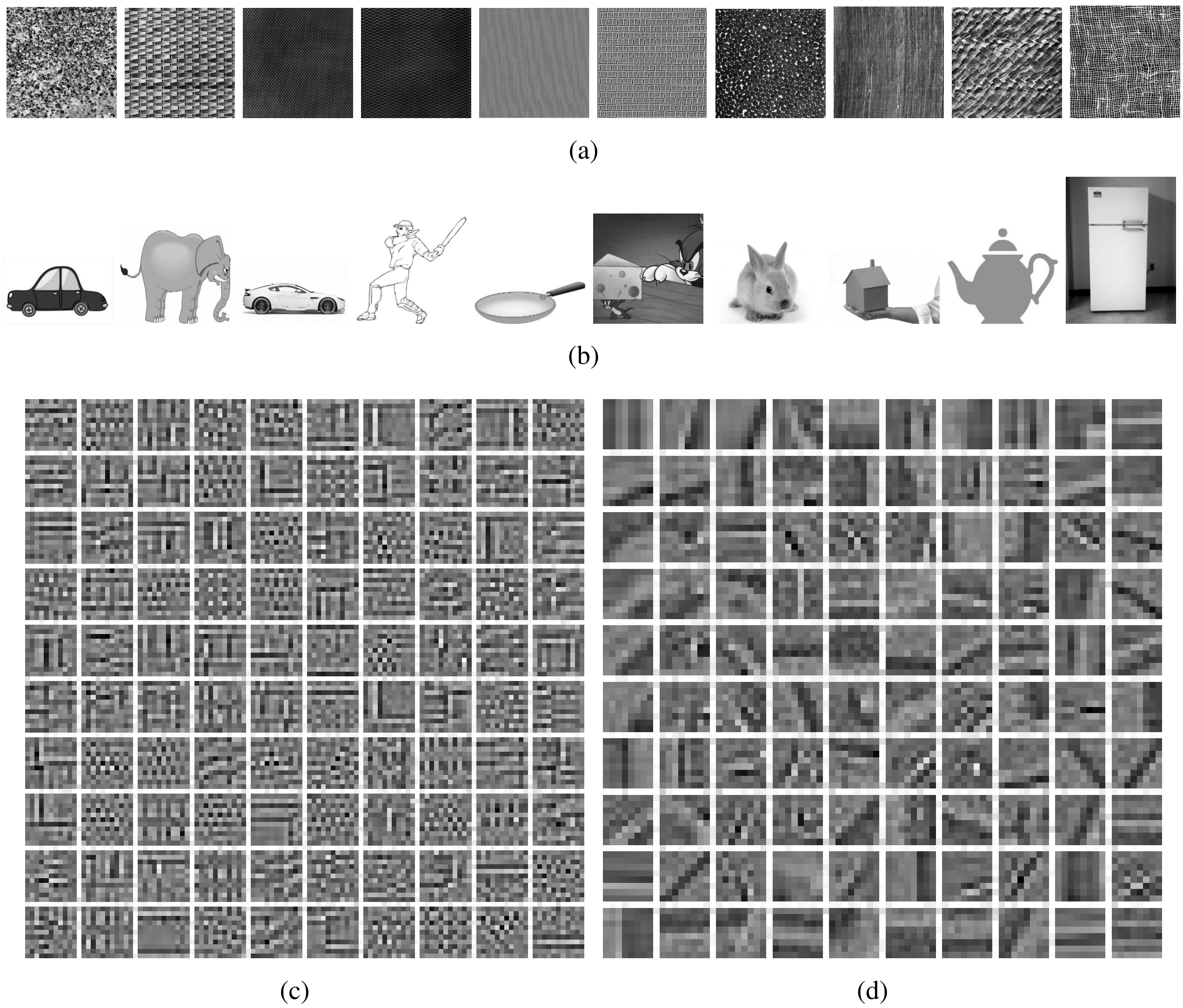
The overall CSCD algorithm is summarized in the Algorithm 1. Here, λ_c and λ_t are the changing parameters corresponding to the two parts, L is the total iteration number, $\mathbf{y}$ is the input image to be separated and $\hat{\mathbf{y}}_c$ and $\hat{\mathbf{y}}_t$ are the estimated cartoon component and the texture component, respectively. The Haar shrinkage value β in (7) is $3\sigma_{\mathbf{y}_c}$, where $\sigma_{\mathbf{y}_c}$ is the standard deviation of the noise estimated using a median estimator on the finest scale of the Haar wavelet coefficients of $\sum_{k=1}^{K_c} \mathbf{d}_{c,k} * \hat{\mathbf{x}}_{c,k}$ [6].

4 Experimental Results

In this section, we present the results of our proposed CSCD algorithm for image separation and compare them with the sparsity-based MCA method [17], adaptive dictionary lending-based MCA (A-MCA) method [15], and a recent stat-of-the-art Block Nuclear Norm (BNN) based image separation method [12]. In these experiments, we use the Peak Signal to Noise Ratio (PSNR) to measure the performance of the routines tested. For the MCA method, wavelet and local DCT dictionaries are used to represent the cartoon and the texture components, respectively. For the A-MCA method, we use a curvelet dictionary for sparsely representing the cartoon component and learn a local patch-based texture dictionary using the images shown in Figures 2 (a) to represent the texture component.

Training images shown in Figures 2 (a) and Figure 2 (b) are used to learn the convolution filters $\{\mathbf{d}_{t,k}\}_{k=1}^{K_t}$, $\{\mathbf{d}_{c,k}\}_{k=1}^{K_c}$, respectively using the CSC method proposed in [22]. The corresponding learned filters are shown in Figures 2 (c) and Figures 2 (d) for the texture and the cartoon components, respectively. From Figures 2 (c), one can see that these filters are oscillatory in nature and they do a good job in capturing the patterns of the training textures (Figures 2 (a)). Similarly, from Figures 2 (d), we observe that the learned filters look similar to those found in a Gabor dictionary. Also, they capture domain specific information found in cartoon type images such as edges.

Figures 3 and 4 show the original images and the decomposed images corresponding



(a)

(b)

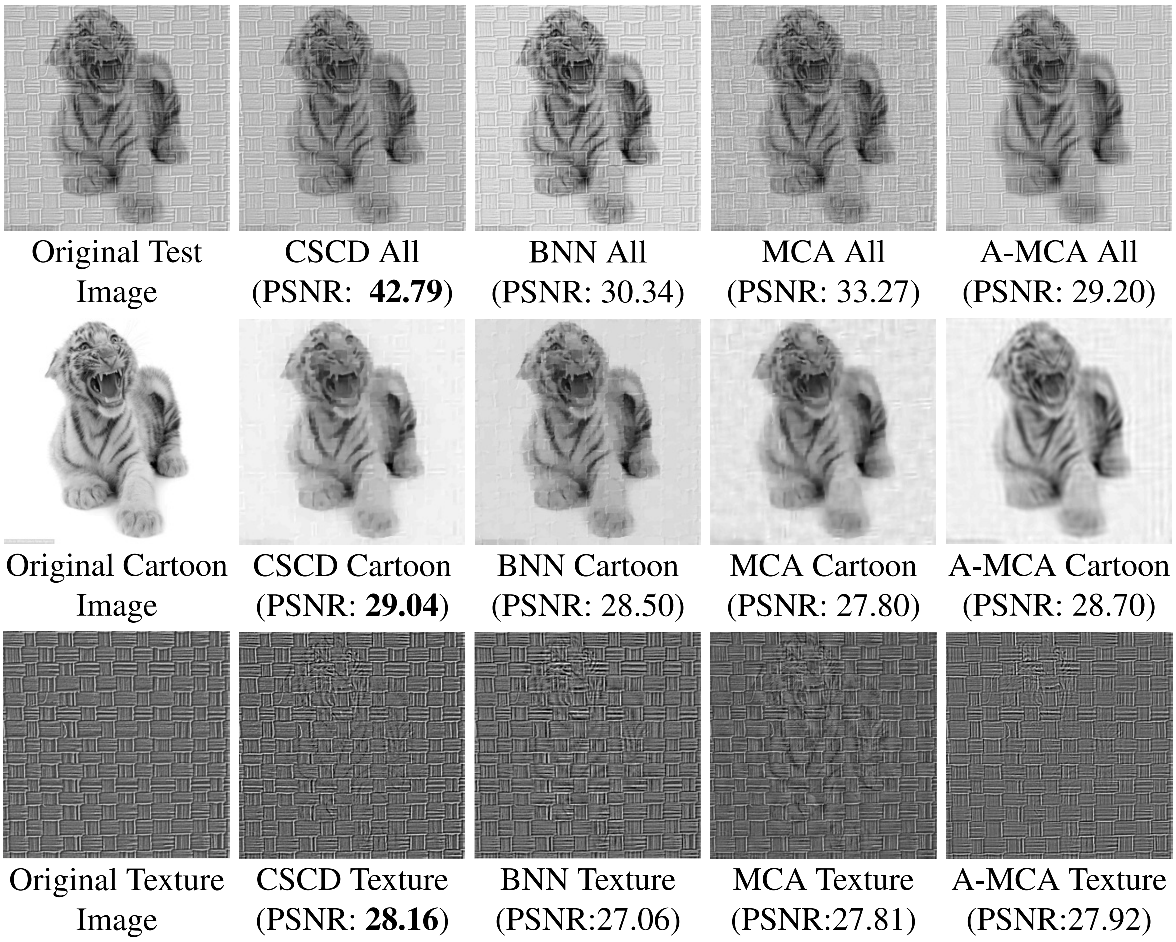
(c)

(d)

Figure 2: (a) Training texture images used for learning a set of texture filters $\{\mathbf{d}_{t,k}\}_{k=1}^{K_t}$. (b) Training cartoon images used for learning a set of cartoon filters $\{\mathbf{d}_{c,k}\}_{k=1}^{K_c}$. (c) Learned texture filters $\{\mathbf{d}_{t,k}\}_{k=1}^{K_t}$. (d) Learned cartoon filters $\{\mathbf{d}_{c,k}\}_{k=1}^{K_c}$.

to different methods. All the test images were excluded from the images used to learn the convolution filters. All the parameters are empirically determined. We fix $\lambda_t = \max(0.55 - 0.05 * i, 0.005)$ and $\lambda_c = \max(0.55 - 0.05 * i, 0.005)$ during the ith iteration for the following two cases. In both of these figures, the first column shows the original test image, original cartoon image and original texture image. Second column shows the results corresponding to our CSCD method. Third, fourth and fifth columns show the results corresponding to the BNN, MCA and A-MCA methods, respectively.

As can be seen from these figures, our method is able to separate the morphological components from the given images better than the other methods. In particular, experiments with the *Tiger+Texture* image shown in Figure 3, our method achieves the PSNR of 29.04 dB on the separated cartoon component compared to the PSNRs of 28.50, 27.80 and 28.70 corresponding to BNN, MCA and A-MCA methods, respectively. Similarly, our method obtains the PSNR of 28.16 dB for the texture component compared to the PSNRs of 27.06, 27.81 and 27.92 corresponding to BNN, MCA and A-MCA methods, respectively. Overall, our method achieves the PSNR of 42.79 dB when the two estimated components are combined compared to the PSNRs of 30.34 dB, 33.27 dB and 29.20 dB for the BNN, MCA and A-MCA methods, respectively. Similar PSNR performances are also observed in Figure 4 with



Original Test Image | CSCD All (PSNR: **42.79**) | BNN All (PSNR: 30.34) | MCA All (PSNR: 33.27) | A-MCA All (PSNR: 29.20)

Original Cartoon Image | CSCD Cartoon (PSNR: **29.04**) | BNN Cartoon (PSNR: 28.50) | MCA Cartoon (PSNR: 27.80) | A-MCA Cartoon (PSNR: 28.70)

Original Texture Image | CSCD Texture (PSNR: **28.16**) | BNN Texture (PSNR:27.06) | MCA Texture (PSNR:27.81) | A-MCA Texture (PSNR:27.92)

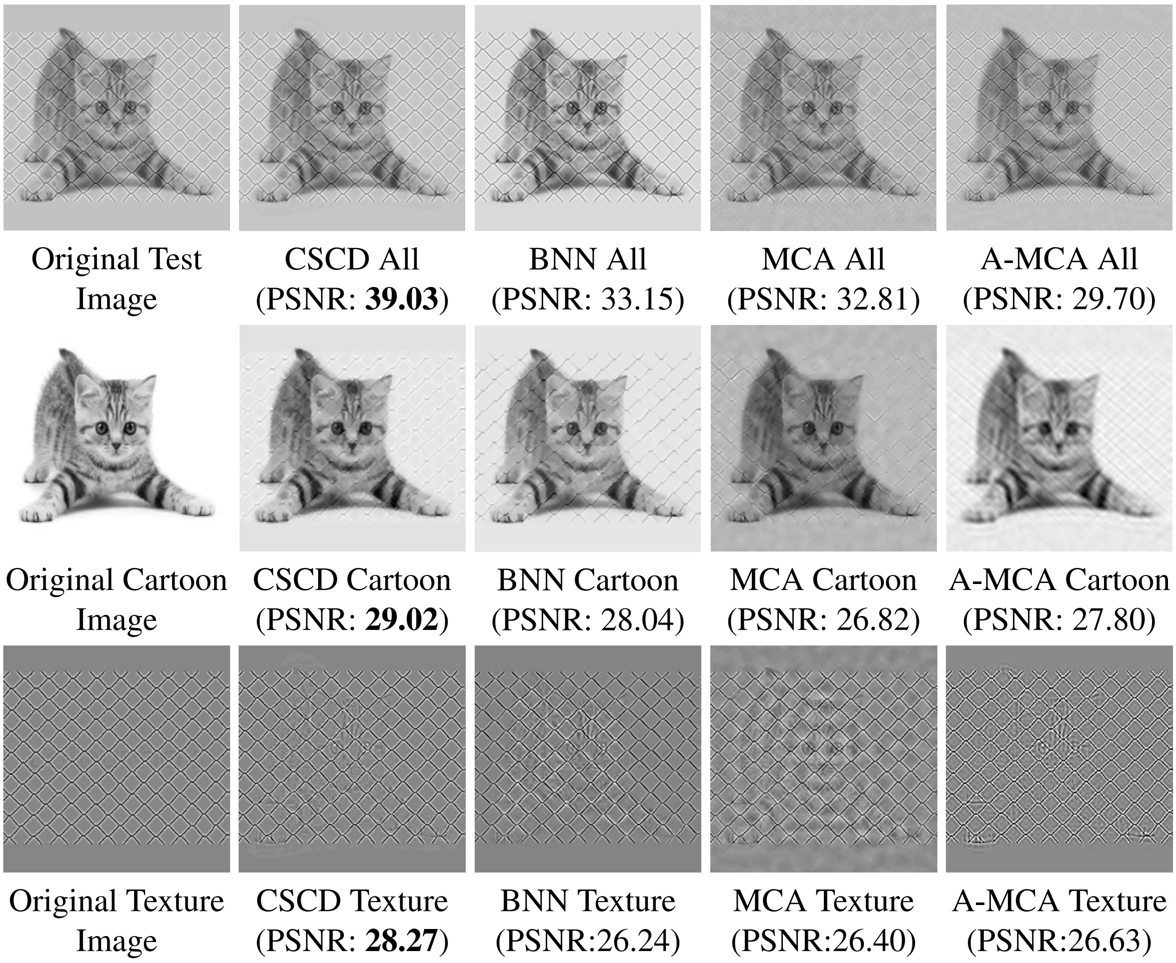
Figure 3: Image decomposition results on the *Tiger+Texture* image. We compare the performance of our method with that of BNN, MCA and A-MCA.

the experiments on the *Cat+Cage* image. These results clearly indicate that an improvement is achieved when a convolutional sparse coding method is used to separate morphological components of an image as can be seen by comparing the visual as well as PSNR results of our method with that of MCA, A-MCA and BNN in Figures 3 and 4.

Using the *Tiger+Texture* and *Cat+Cage* images, in Figure 5, we show the evolution of the objective function. Note that in Figure 5, the relative error decreases significantly after a few iterations and saturates around the sixth iteration, showing that the proposed method is efficient and requires less number of iterations compared to converge.

In the last set of experiments, we present an application of the proposed CSCD method in extracting the underlying fingerprint from a latent fingerprint. Latent fingerprints are among the most valuable and common types of physical evidence. Latent fingerprints obtained from crime scenes can be used as crucial evidence in forensic identification. However, matching latent fingerprints with the enrolled fingerprints is a difficult problem as latent fingerprints are often of poor quality with tiny outlets. In this experiment, we show that one can use the proposed CSCD method to extract the underlying fingerprint from a latent fingerprint, which can be then matched with the enrolled fingerprints. In this experiment, we only present the separation results as the matching of latent fingerprints is beyond the scope of this paper. We use the same learned cartoon filters as used in the previous experiments. However, we learn the texture filters corresponding to fingerprints, from a set of clean fingerprints. In this experiment, we set $\lambda_t = \max(0.5 - 0.05 * k, 0.15)$ and $\lambda_c = \max(0.55 - 0.05 * k, 0.05)$.

The first row of Figure 6 shows the input latent fingerprint and the learned texture (fingerprint) filters. The learned fingerprint filters show some characteristics unique to fingerprints.



Original Test Image

CSCD All (PSNR: **39.03**)

BNN All (PSNR: 33.15)

MCA All (PSNR: 32.81)

A-MCA All (PSNR: 29.70)

Original Cartoon Image

CSCD Cartoon (PSNR: **29.02**)

BNN Cartoon (PSNR: 28.04)

MCA Cartoon (PSNR: 26.82)

A-MCA Cartoon (PSNR: 27.80)

Original Texture Image

CSCD Texture (PSNR: **28.27**)

BNN Texture (PSNR:26.24)

MCA Texture (PSNR:26.40)

A-MCA Texture (PSNR:26.63)

Figure 4: Image decomposition results on the *Cat+Cage* image. We compare the performance of our method with that of BNN, MCA and A-MCA.

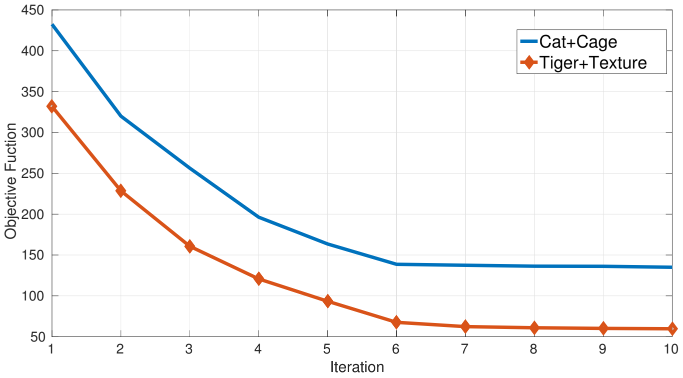
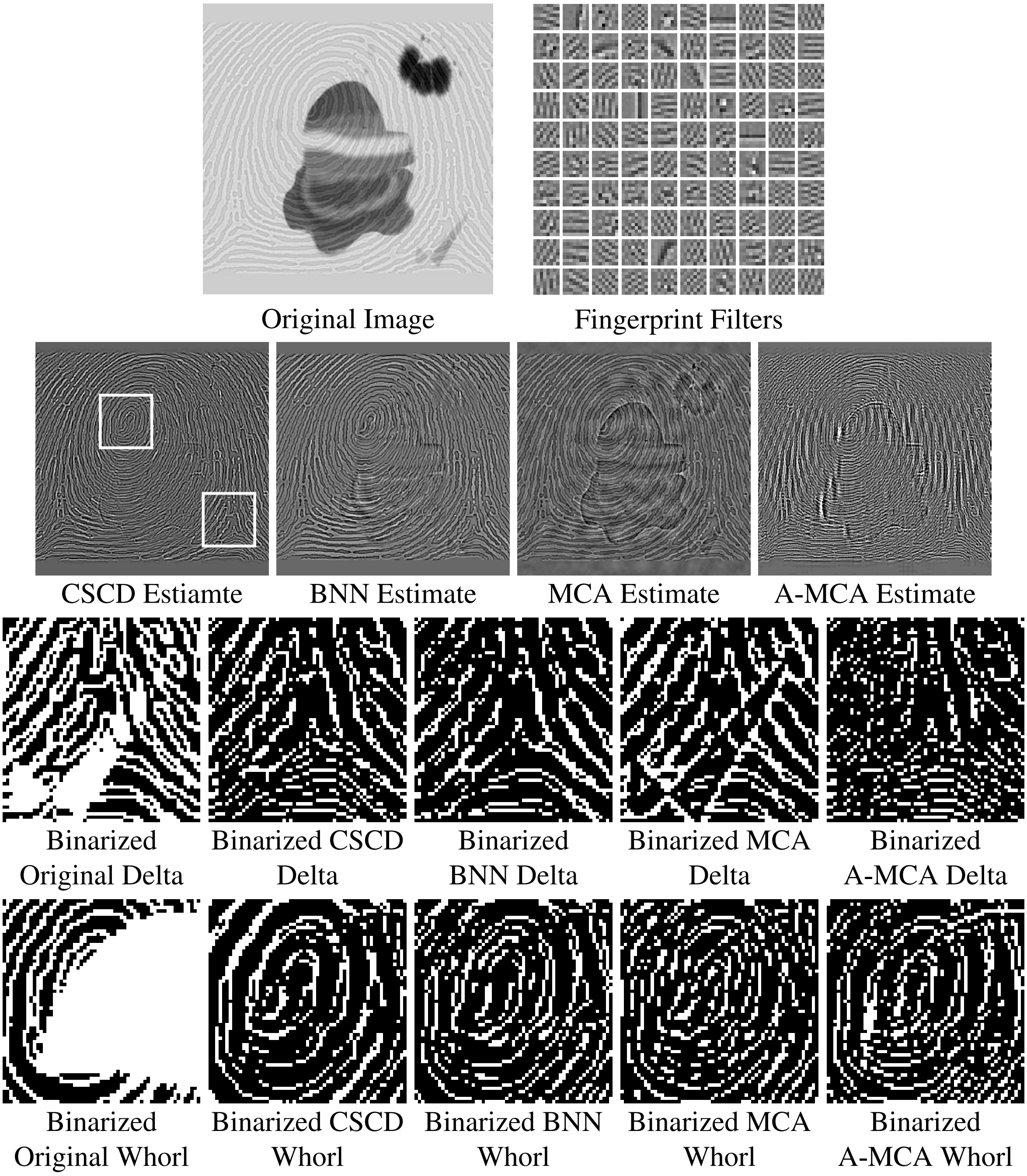


Figure 5: The objective function value as a function of iteration number for the experiments with the *Tiger+Texture* and *Cat+Cage* images.

In the second row of this figure, we show the results of image separation corresponding to different methods. It can be seen from this figure that our method is able to extract the underlying structure of the fingerprint better than MCA and A-MCA. This can also be seen by comparing the binarized extracted delta and whorl patterns in the last two rows of this figure. It is interesting to see that if the cartoon part is piecewise smooth, then our method and BNN can recover the shape of the fingerprint better than MCA since it uses local DCT to represent the texture component. Furthermore, as can be seen from the results of A-MCA, learning a local dictionary to represent the fingerprint textures does not produce good results. This experiment clearly shows the significance of our method compared to MCA and A-MCA



Original Image

Fingerprint Filters

CSCD Estiamte

BNN Estimate

MCA Estimate

A-MCA Estimate

Binarized Original Delta

Binarized CSCD Delta

Binarized BNN Delta

Binarized MCA Delta

Binarized A-MCA Delta

Binarized Original Whorl

Binarized CSCD Whorl

Binarized BNN Whorl

Binarized MCA Whorl

Binarized A-MCA Whorl

Figure 6: Fingerprint separation results. We compare the visual performance of our method with that of BNN, MCA and A-MCA based om recovered Whor and Delta part.

that use fixed and patch-based adaptive dictionaries, respectively.

5 Conclusion

In this paper, we presented a CSC-based image separation method for decomposing a given image into a texture and a cartoon component. Our method entails learning cartoon and texture filters directly from training examples. Using these learned filters, we proposed a sparsity-based optimization framework for image separation. Various experiments showed the significance of our CSC-based image separation method over the sparse representation-based methods that use global dictionaries and patch-based methods that use local dictionaries for image separation.

In the future, we will apply the proposed image separation method on various computer vision problems such as intrinsic image estimation and albedo estimation that require separating a specific component from a given image.

Aknowledgement

This work was supported by an ARO grant W911NF-16-1-0126.

References

[1] M. Aharon, M. Elad, and A. Bruckstein. k -svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, Nov 2006.

[2] Jean-François Aujol, Gilles Aubert, Laure Blanc-Féraud, and Antonin Chambolle. *Scale Space Methods in Computer Vision*, chapter Image Decomposition Application to SAR Images, pages 297–312. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003. ISBN 978-3-540-44935-5.

[3] J. Bobin, J. L. Starck, J. M. Fadili, Y. Moudden, and D. L. Donoho. Morphological component analysis: An adaptive thresholding strategy. *IEEE Transactions on Image Processing*, 16(11):2675–2681, Nov 2007.

[4] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[5] Hilton Bristow, Anders Eriksson, and Simon Lucey. Fast convolutional sparse coding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 391–398, 2013.

[6] D. Donoho, I. Johnstone, and Iain M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81:425–455, 1993.

[7] Michael Elad. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer, 1st edition, 2010. ISBN 144197010X, 9781441970107.

[8] Xavier Gibert, Vishal M. Patel, Demetrio Labate, and Rama Chellappa. Discrete shearlet transform on GPU with applications in anomaly detection and denoising. *EURASIP Journal on Advances in Signal Processing*, 2014:64, 2014.

[9] S. Gu, W. Zuo, Q. Xie, D. Meng, X. Feng, and L. Zhang. Convolutional sparse coding for image super-resolution. In *IEEE International Conference on Computer Vision*, pages 1823–1831, Dec 2015.

[10] Felix Heide, Wolfgang Heidrich, and Gordon Wetzstein. Fast and flexible convolutional sparse coding. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5135–5143. IEEE, 2015.

[11] Yves Meyer. *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations: The Fifteenth Dean Jacqueline B. Lewis Memorial Lectures*. American Mathematical Society, Boston, MA, USA, 2001. ISBN 0821829203.

[12] Shintaro Ono, Takamichi Miyata, and Isao Yamada. Cartoon-texture image decomposition using blockwise low-rank texture characterization. *Image Processing, IEEE Transactions on*, 23(3):1128–1142, 2014.

[13] Christian Osendorfer, Hubert Soyer, and Patrick van der Smagt. *Neural Information Processing: 21st International Conference, ICONIP 2014, Kuching, Malaysia, November 3-6, 2014. Proceedings, Part III*, chapter Image Super-Resolution with Fast Approximate Convolutional Sparse Coding, pages 250–257. Springer International Publishing, Cham, 2014.

[14] Vishal M Patel, Glenn R Easley, Rama Chellappa, and Nasser M Nasrabadi. Separated component-based restoration of speckled sar images. *IEEE Transactions on Geoscience and Remote Sensing*, 52(2):1019–1029, 2014.

[15] Gabriel Peyré, Jalal Fadili, and Jean-Luc Starck. Learning the morphological diversity. *SIAM Journal on Imaging Sciences*, 3(3):646–669, 2010.

[16] Hayden Schaeffer and Stanley Osher. A low patch-rank interpretation of texture. *SIAM Journal on Imaging Sciences*, 6(1):226–262, 2013.

[17] Jean-Luc Starck, Michael Elad, and David L Donoho. Image decomposition via the combination of sparse representations and a variational approach. *Image Processing, IEEE Transactions on*, 14(10):1570–1582, 2005.

[18] Gabriele Steidl, Joachim Weickert, Thomas Brox, Pavel Mrázek, and Martin Welk. On the equivalence of soft wavelet shrinkage, total variation diffusion, total variation regularization, and sides. *SIAM J. Numerical Analysis*, 42(2):686–713, 2004.

[19] S. Taheri, V. M. Patel, and R. Chellappa. Component-based recognition of faces and facial expressions. *IEEE Transactions on Affective Computing*, 4(4):360–371, Oct 2013.

[20] Luminita A. Vese and Stanley J. Osher. Modeling textures with total variation minimization and oscillating patterns in image processing. *Journal of Scientific Computing*, 19(1):553–572, 2003.

[21] Brendt Wohlberg. Efficient convolutional sparse coding. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7173–7177. IEEE, 2014.

[22] Brendt Wohlberg. Efficient algorithms for convolutional sparse representations. *IEEE Transactions on Image Processing*, 25(1):301–315, Jan 2016.

[23] M. D. Zeiler, G. W. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *International Conference on Computer Vision*, pages 2018–2025, Nov 2011.

[24] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Rob Fergus. Deconvolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2528–2535. IEEE, 2010.