2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Color Image Processing based on Nonnegative Matrix Factorization with Convolutional Neural Network

Thanh Xuan Luong, Bo-Kyeong Kim
Computational NeuroSystems Laboratory
Korea Advanced Institute of Science and Technology
Daejeon 305-701, Republic of Korea
{sputnikav, bokyeong1015}@gmail.com

Soo-Young Lee
Department of Electrical Engineering
Korea Advanced Institute of Science and Technology
Daejeon 305-701, Republic of Korea
sy-lee@kaist.ac.kr

***Abstract*— Although Nonnegative Matrix Factorization (NMF) has been widely known as an effective feature extraction method, which provides part-based representation and good reconstruction, there were relatively few researches using NMF for color image processing. Particularly, many studies are now using Convolutional Neural Network (CNN) in combined with Auto-Encoder (AE) or Restricted Boltzmann Machine (RBM) for learning features of color images. In this paper, we explore the ability of NMF to handle color images. Especially, a new method using NMF to learn features in CNN is proposed. In our experiments conducted on CIFAR-10, NMF shows the feasibility for reconstruction and classification of color images. Furthermore, unlike edge- or curve- shaped features learned by AE and RBM in CNN, our method provides dot- shaped features. These new types of features could be considered as basic building blocks in the lowest level of constructing images. Our results demonstrate that NMF is capable of being a supporting tool for CNN in learning features.**

Keywords—Nonnegative Matrix Factorization; Convolutional Neural Network; color image processing; CIFAR-10

I. INTRODUCTION

Numerous techniques have been developed for various applications of image processing. In the field of image reconstruction and classification, neural network is one of the most effective methods. Particularly, convolutional neural network (CNN), inspired by biological visual model [1], has shown groundbreaking results on popular image dataset such as MNIST [2] – [4] or CIFAR-10 [4], [5]. Many previous researches utilized auto-encoder (AE) and restricted Boltzmann machine (RBM) to learn features [6]–[8]. Using these feature extractors, optimal latent feature representation could be achieved by minimizing reconstruction error and modeling the probabilistic distribution of given data. The features learned by AE or RBM usually have edge- or curve-shapes in CNN. Motivated by the assumption that features in the lowest level for constructing image would be helpful for reconstruction, we combined CNN approach with nonnegative matrix factorization (NMF), which is a strong tool for extracting features in the context of machine learning.

NMF has been broadly used for processing gray images. Nonetheless, not many researches exploit this tool to treat color images [9], [10]. Since colors can deliver important information for understanding images and recognizing objects, we focus on study of color images using NMF in this work. In [9], a color histogram was defined and analyzed by NMF to classify color images. However, applying NMF to color images themselves could provide intuitive representation for analysis and reconstruction. In [10], NMF was directly used to encode color channels for face recognition. In this work, although the authors handled 3 colors independently and separately, processing all color channels simultaneously maybe more reasonable based on the generation of images.

In our paper, standard NMF and non-smooth NMF (nsNMF), a variant algorithm with sparseness constraint, are explored to handle color images. In difference from [9] and [10], we consider the efficiency of using 2 different methods: processing color channels separately and simultaneously. Moreover, to deal with scale and rotation variant images, CNN is applied and combined with NMF in learning features. To show advantages of NMF as a feature extractor of color images, features and reconstruction errors from NMF are compared to those from AE.

The outline of this paper is as follows. In Section II, the background knowledge about nsNMF and CNN is introduced. Then, Section III explains the detailed methods of our work. The experimental result and analysis are presented in section IV. Finally, Section V delivers conclusion and future work.

II. BACKGROUND

A. Non-Smooth Nonnegative Matrix Factorization

The standard NMF algorithm factorizes a non-negative $M \times N$ data matrix X into two non-negative matrices, a $M \times R$ basis matrix W and a $R \times N$ feature coefficient matrix H [11]. Here, *M*, *N*, and *R* are the input dimension, the number of samples, and the feature dimension to be reduced by NMF, respectively. The columns in basis matrix and feature matrix denote part-based building blocks and coefficients to explain how these blocks are linearly added to represent original data samples,

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (Ministry of Science, ICT & Future Planning) (2013-063906)

respectively. Cost function of NMF is set as the square of Euclidean distance between original matrix X and reconstructed matrix WH in equation (1).

$$J_{NMF} = \frac{1}{2MN}\|X - WH\|^2 \quad (1)$$

The basic update rule is derived from the gradient descent to minimize cost function defined above. Assigning learning rate, which updates the basis matrix and feature coefficient matrix non-negatively, we come up with multiplicative update rule as in (2) and (3)

$$H_{rn} \Leftarrow H_{rn}\frac{(W^T X)_{rn}}{(W^T WH)_{rn}} \quad (2)$$

$$W_{mr} \Leftarrow W_{mr}\frac{(XH^T)_{mr}}{(WHH^T)_{mr}} \quad (3)$$

In the field of feature learning, imposing sparsity constraints has been largely used [3], [12], [13]. The efficiency of exploiting sparsity is justified by the fact that natural images represent only a small part of the image space. Thus, sparsity allows us to effectively extract necessary information in terms of features.

To introduce sparse characteristic to the basis matrix, we used nsNMF in [14]. The nsNMF was designed to force smoothness on the feature matrix H and thus provide non-sparseness or sparsity on the basis matrix. The multiplicative update rule is slightly modified by adding (4) before (2) and (3).

$$H \Leftarrow SH \quad (4)$$

Here, $S = (1-\theta)eye(\mathrm{R}) + \frac{\theta}{R}ones(\mathrm{R})$ is the smoothing matrix and parameter θ, which is ranged from 0 to 1, determines degree of smoothing. As θ is closed to 1, H is smooth, and W becomes sparse to compensate the loss of sparseness in H .

B. *Convolutional Neural Network*

For realistic input images with huge size, CNN could be introduced for preventing a redundancy in parameters by forcing each extracted feature to be global, i.e. to span the whole field of vision [15], [16]. CNN assumes a stationary property in the images. It means the statistics of each small part or sub-patch in image stay similar also for other regions. Based on the stationary assumption, features are learned from the sub-patches that are randomly extracted from input samples. In previous researches, single layer perceptron, AE, or RBM are usually used for learning features, as shown in [17] – [19].

After learning features, CNN mainly consists of 2 steps: convolution and pooling. We firstly learn the features over small patches randomly sampled from the original image set. This learning process gives a *k*-dimensional feature vectors $f(\mathrm{x})$. Then, convolution is done to detect local features, resulting in a feature map. As illustrated in Fig. 1 a, given an *n-by-n* image with 3 color channels RGB, we extract sub-images of size *w-by-w* and convolve them with *k* learned features to form a *k-by-(n-w+1)-by-(n-w+1)* representation of images.

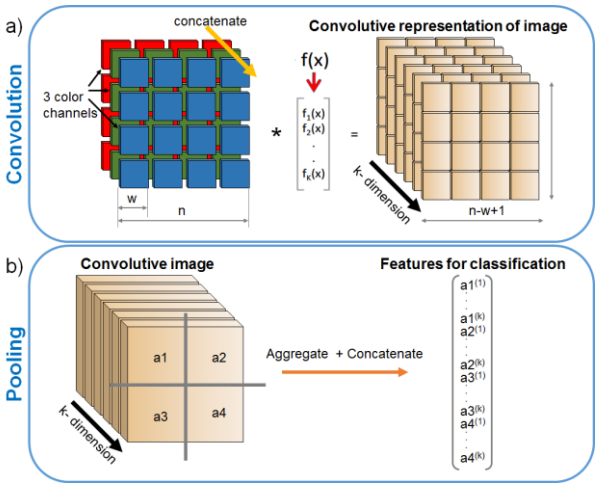


Fig. 1. Flow chart of CNN
a) convolution operation and b) pooing operation

One could utilize feature maps themselves from convolution step for classification. However, the convolution brings about a dramatically increased dimension of convolved features with a computational challenge. For example, let consider an instance image of size 32x32 pixels, convolution results in a map of size (32-8+1).(32-8+1) = 625, which is again multiplied with 100-dimensional features to form convolutive representation. Consequently, for 1 image sample, we need a classifier with about six-hundred dimensional input, which is practically expensive. To overcome this problem, aggregation operation called pooling is used. Pooling implies calculating maximum or average value over a range of neighbor sub-patches. Fig. 1 b shows pooling scheme over 4 non-overlapping regions. Here, a1, a2, a3 and a4 are feature vectors after pooling over 4 quadrants of image [17]. After aggregation, these vectors are concatenated to form features of reduced size.

III. METHODS

A. *Datasets: CIFAR-10*

The CIFAR-10 dataset [20] consists of 32x32 color images in 10 classes (airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck), with 6000 images per class. There are 50000 training images and 10000 test images in total.

The preprocessing is done by rescaling pixel values to the range from 0 to 1 and concatenating all columns for each color channel. Thus a single image can be represented as 3 vectors $X_j^c \in \Re^{1024}$ where $c \in \{\text{red, green, blue}\}$ denotes a corresponding color channel and $j \in \{1, \ldots, \mathrm{N}\}$ denotes the sample index. Notice that whitening methods are not capable to be applied in our experiment due to nonnegative nature of NMF.

B. *nsNMF without CNN architecture*

To treat color images, we construct 3 data matrices $X^c = [X_1^c \ldots X_N^c] \in \Re^{1024\times N}$. Without CNN, 2 types of architectures are considered to explore the effect of dealing information from 3 color channels separately

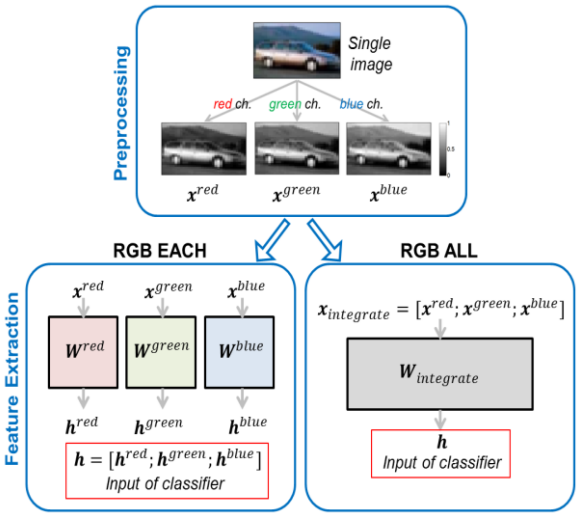


Fig. 2. Method using nsNMF without CNN

(called 'RGB EACH') or simultaneously (called 'RGB ALL').

As shown in Fig. 2, the RGB EACH method has 3 encoding models for red, green, and blue color channels. It supposes that 3 colors may have an independent relationship to form complete image. For each color channel c, X^c is decomposed to $W^c \in \Re^{1024\times R}$ and $H^c \in \Re^{R\times N}$ via nsNMF. After that, by concatenating 3 feature coefficient matrices in column-wise direction, $H = [H^{red}; H^{green}; H^{blue}] \in \Re^{3R\times N}$ is formed.

The RGB ALL uses a stacked data matrix from 3 color channels, $X = [X^{red}; X^{green}; X^{blue}] \in \Re^{3072\times N}$ as illustrated in Fig. 2. Then nsNMF learns $W \in \Re^{3072\times R}$ and $H \in \Re^{R\times N}$. For both RGB EACH and RGB ALL, each column of H is used as an input vector of classifier. Notice that the dimension of columns of H is 3 R for RGB EACH and R for RGB ALL. To achieve a fair comparison, we set the feature dimension of RGB ALL to be 3 times bigger than RGB EACH. For example, if 20 is used as the feature dimension in RGB EACH, then 60 is used in RGB ALL. For compactness, the denoted feature dimensions in following experimental results are regarded to the case of RGB EACH.

In our experiment, the parameter θ in nsNMF is set to be 0.8 after trying several values. For classification, we used k-Nearest Neighbor (k-NN) and soft-max classifier. In k-NN classifier, the Euclidean distance is used as a measure, and optimal number of neighbors is selected from the set $k \in [1, 3, 5, ..., 15]$. To train soft-max classifier, weight decaying cost is added to original cost using maximum likelihood.

C. nsNMF with CNN architecture

In order to learn R features, we applied nsNMF to 200,000 sub-patches of size 8x8 that are randomly sampled from 50,000 training images. All 3 color channels are exploited simultaneously to reach the highest efficiency in the same fashion of RGB ALL as in Fig. 3.

After that, the learned features are convolved with each input image to make R feature maps of size 25x25. Then

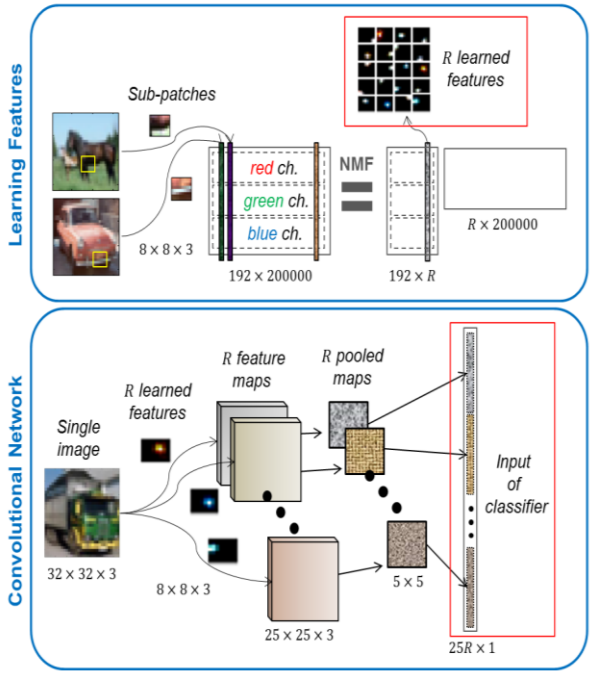


Fig. 3. Method using nsNMF with CNN

mean-pooling is done over 5x5 region without overlapping to construct R pooled maps of size 5x5. For classification using soft-max classifier, $25R$-dimensional vector is formed for each image by concatenating the pooled maps. For this experiment, we desire to visualize the change of extracted features and classification error by varying the number of features R and the degree of sparseness θ.

IV. Results

In order to measure performance of nsNMF for color image processing, we compare it to AE, which is one of popular feature extractors, in terms of reconstruction. The first experiment tests nsNMF to process color channels separately and simultaneously. The second experiment shows development of features and advantages of nsNMF combined with CNN architecture.

A. nsNMF without CNN architecture

1) Reconstruction

The reconstruction error of one sample image is calculated in (5) for RGB EACH and in (6) for RGB ALL.

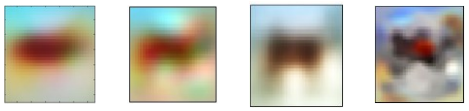
$$E_{RGB\_EACH} = \frac{1}{3}\sum_c \left(\left\| x^c - W^c h^c \right\|^2 \Big/ \left\| x^c \right\|^2 \right) \quad (5)$$

$$E_{RGB\_ALL} = \left\| x - Wh \right\|^2 \Big/ \left\| x \right\|^2 \quad (6)$$

An example of reconstruction for sample "horse" is depicted in Table I. For each feature dimension, the reconstructed image and error are shown. Fig. 4 shows mean reconstruction error over whole samples in CIFAR-10.

As expected, the reconstruction error decreases as we enlarge feature dimension. The overall performance is better for the case of RGB ALL compared to RGB EACH. It demonstrates extracting information of 3 color channels simultaneously is useful in reconstruction of color images.

TABLE I
EXAMPLE OF RECONSTRUCTION FOR A SINGLE SAMPLE IN CIFAR-10

Feature Dim.	RGB EACH AE	RGB EACH NMF	RGB ALL AE	RGB ALL NMF
Original Sample				
20				
	0.0917	0.0632	0.0766	0.0662
50				
	0.0623	0.0436	0.0491	0.0350
100				
	0.0364	0.0270	0.0468	0.0257
200				
	0.0263	0.0174	0.0338	0.0159
500				
	0.0155	0.0051	0.0237	0.0048

More importantly, reconstruction results of NMF are better than AE for all feature dimensions. Both AE and NMF are similar in that they learn features to minimize reconstruction cost, whereas the difference comes out from non-negative constraints in NMF. Our results show non-negativity in NMF is beneficial to development of meaningful features for reconstruction.

2) Classification

The classification performance as a function of feature dimension is illustrated in Fig. 5. Although our results are better than the chance level of 10-class problem, they are still lower than other researches [17]. Here, we note that

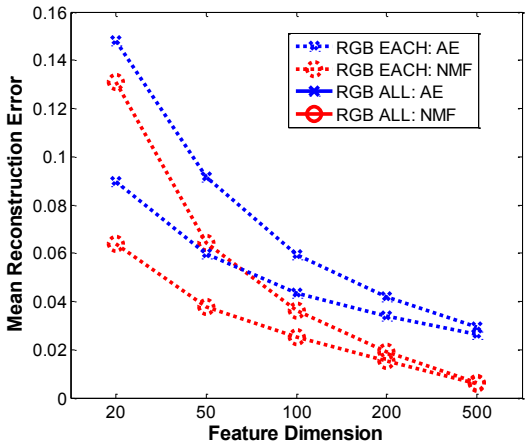


Fig. 4. Mean reconstruction error of CIFAR-10 using auto-encoder and nsNMF without CNN

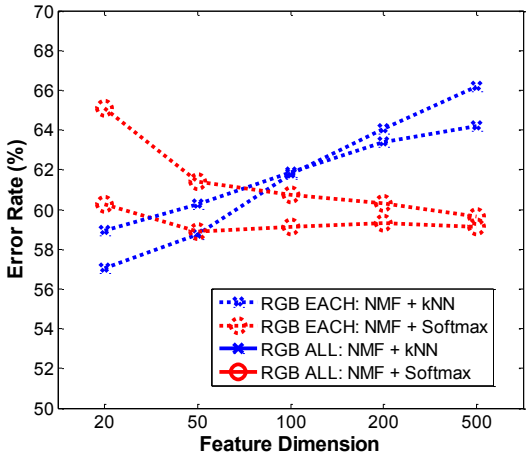


Fig. 5. Classification performance of CIFAR-10 using nsNMF without CNN

there is a tendency depending on the type of classifier. When feature dimension becomes larger, the error rate increases with k-NN classifier. In contrast, the classification performance is improved with soft-max classifier. To refine this classification outcome, in the next experiment we combine our feature extractors with CNN, an efficient architecture for dealing with realistic and complex image sets such as CIFAR-10.

B. nsNMF with CNN architecture

1) Learned features

The features are learned from 200,000 sub-patches by standard NMF with $\theta = 0$ and nsNMF with $\theta = 0.8$. These features are illustrated in Table II. For comparison, in the left-sided column, we show features learned by standard AE. Except for the case of 20 learned features, randomly selected 49 features are plotted.

Table II shows that color variety is integrated in each extracted feature. In CNN-AE, edge-shaped features are observed. Meanwhile, both NMF and nsNMF provide dot-shaped features because non-negative constraint forces to extract basic building blocks in the lowest level for images. Also, as increasing the number of features, the dots become small and positioned in different locations. Furthermore, higher sparseness of nsNMF results in the strongly localized features. In contrary, features with low sparseness show blurred dots with relatively large size.

2) Classification

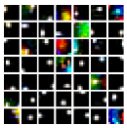
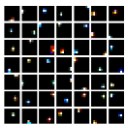
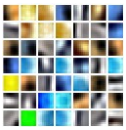
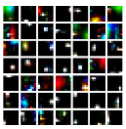
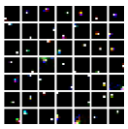
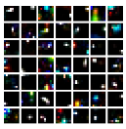
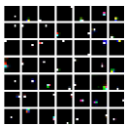
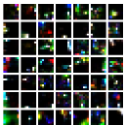
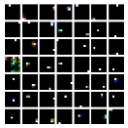
The classification results using standard NMF and nsNMF with CNN are shown in Fig. 6. Compared to Fig. 5 without CNN, better recognition performances are achieved. It implies the advantage of using CNN approach for CIFAR-10, which contains scale and rotation variant images.

The error of standard NMF keeps going down as increasing the number of features. However, when using nsNMF with $\theta = 0.8$, there is an optimal number of features with that architecture reaches its best performance.

V. CONCLUSION

A framework for features extraction and image classification has been designed using nsNMF and CNN architecture. Our work was conducted on a CIFAR-10, a challenging dataset of color images. The experimental

TABLE II
LEARNED FEATURES OF CIFAR-10
USING AUTO-ENCODER AND NSNMF WITH CNN

# Learned Features[a]	CNN-AE	CNN-NMF	
		$\theta = 0$	$\theta = 0.8$
20			
50			
100			
200			
500			

[a]The term '# Learned Features' denotes the number of learned features.

results have been shown in the aspect of features, reconstruction, and classification performance.

In the first experiment, two approaches to work with color images are compared. We observed that simultaneous processing of 3 color channels provides better reconstruction and recognition results. This experiment also shows the major advantage of nsNMF over AE for reconstruction of original color images. However, without CNN architecture, it has been hard to handle scale and rotation variant image in CIFAR-10 for classification. In the second experiment, we used CNN where nsNMF learned features from lots of sub-patches. Features are then visualized along with the ones learned by AE. In contrast to features learned by AE, NMF features are dot-shaped and highly-localized due to its non-negative nature. The classification accuracy is improved with CNN compared to the model without CNN.

In conclusion, our results demonstrate that NMF and nsNMF are capable of being a supporting tool for CNN in learning the features and classifying image data. Though edge- or curve-shaped features have been shown in previous works using AE or RBM, new type of dot-shaped features are drawn in our work.

Within the range of this paper, our proposed model is applied to one dataset with limited numbers of features. However, there is a room for improving classification performance in future works by applying a better NMF model, such as stacked NMF, to CNN or multi-layer CNN.

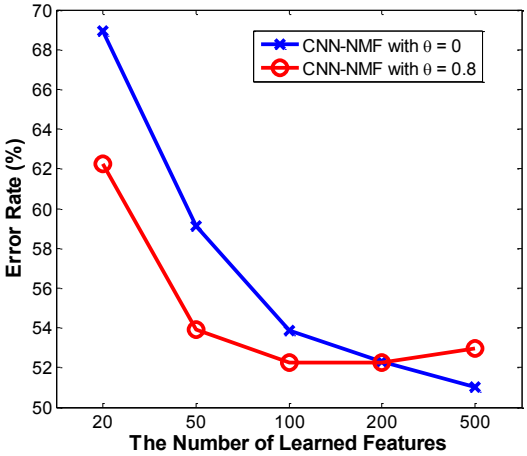


Fig. 6. Classification performance of CIFAR-10 using NMF with CNN

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (Ministry of Science, ICT & Future Planning) (2013-063906).

REFERENCES

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, pp. 2278-2324, 1998.

[2] K. Labusch, E. Barth, and T. Martinetz, "Simple method for high-performance digit recognition based on sparse coding," *IEEE Trans. Neural Networks*, vol. 19, pp. 1985-1989, 2008.

[3] M. A. Ranzato, F. J. Huang, Y. L. Boureau, and Y. Lecun, "Unsupervised learning of invariant feature hierarchies with applications to object recognition," in *Computer Vision and Pattern Recognition (CVPR), 2007 IEEE Conf.*, pp. 1-8.

[4] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conf.*, pp. 3642-3649.

[5] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1106-1114.

[6] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proc. 26th Annual International Conference on Machine Learning*, 2009, pp. 609-616.

[7] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *The Journal of Machine Learning Research*, vol. 9999, pp. 3371-3408, 2010.

[8] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *Artificial Neural Networks and Machine Learning–ICANN 2011*, 2011, pp. 52-59.

[9] D. Guillamet, B. Schiele, and J. Vitria, "Analyzing non-negative matrix factorization for image classification," in *Proc. 16th Int. Conf. Pattern Recognition*, 2002, pp. 116-119.

[10] M. Rajapakse, J. Tan, and J. Rajapakse, "Color channel encoding with NMF for face recognition," in *Int. Conf. Image Processing, 2004 (ICIP'04)*, 2004, pp. 2007-2010.

[11] D. D. Lee, H. S. Seung, "Algorithms for Non-negative Matrix Factorization," in *Advances in Neural Information Processing Systems 13,* 2001, pp. 556–562.

[12] J. Wright, Y. Ma, J. Marial, G. Sapiro, T. S. Huang, and S. Yan, "Sparse Representation for Computer Vision and Pattern Recognition," in *Proceedings of the IEEE*, vol. 98, pp. 1031–1044, 2010.

[13] R. Rigamonti, M. A. Brown, "Are sparse representation really relevant for image classification?," *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conf.*, pp. 1545–1552, 2011.

[14] A. P. Montano, J. M. Carazo, and K. Kochi, "Nonsmooth nonnegative matrix factorization (nsNMF)", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 3, pp.403–415, Mar. 2006.

[15] Y. LeCun, F. J. Huang, and L. Bottou, "Learning methods for generic object recognition with invariance to pose and lighting," in *Computer Vision and Pattern Recognition (CVPR), 2004 IEEE Conf.,* pp. 97–104

[16] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?," in *IEEE Int. Conf. Computer Vision 2009*, pp. 2146–2153.

[17] A. Coates, A. Y. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Int. Conf. Artificial Intelligence and Statistics (AISTATS)*, 2011, pp. 215-223

[18] S. Knerr, L. Personnaz, and G. Dreyfus, "Handwritten digit recognition by neural networks with single-layer training," *IEEE Trans. Neural Networks*, vol. 3, no.6, pp. 962–968, 1992.

[19] G. E. Hinton, R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, pp. 504–507, 2006.

[20] http://www.cs.toronto.edu/~kriz/cifar