Hierarchical Sparse Autoencoder Using Linear Regression-based Features in Clustering for Handwritten Digit Recognition

Hai T. Phan
Advanced Program in Computer Science, University Of Science
Ho Chi Minh City, Vietnam
pthai@apcs.vn

An T. Duong
University of Science
Ho Chi Minh City, Vietnam
0912019@student.hcmus.edu.vn

Nam Do-Hoang Le
John von Neumann Institute, Vietnam National University
Ho Chi Minh City, Vietnam
nam.le.ict@jvn.edu.vn

Son T. Tran
University of Science
Ho Chi Minh City, Vietnam
ttson@fit.hcmus.edu.vn

***Abstract*—Recently, handwritten digit recognition using higher level features has got more promising results than conventional ones using intensity values, where the higher level features are considered as features of simple strokes in images. Although the state-of-the-art performance is very impressive, there is still room to improve better in both accuracy and computation complexity. In this paper, we propose a new feature based on linear regression to extract geometrical characteristics of handwritten digits. The linear regression-based features are utilized to cluster set of digit image in preprocessing. After that, each set of clustered digit images is inputted a hierarchical sparse autoencoder to extract higher level features automatically. Our method result achieves error rates lower than that of conventional method in the most of cases. The experiment shows that the efficiency of data clustering can get promising results.**

Keywords - Higher level features, sparse autoencoder, hierarchical sparse autoencoder, handwritten digit recognition, linear regression-based features.

I. Introduction

Recognition algorithms provide some knowledge for computer perception and feature extraction plays an important role of recognition system. Some achievements of feature extraction such as SIFT [11], or HOG [2] and MFCCs [6] are utilized in most applications of computer vision and pattern recognition popularly. For instances, SIFT and HOG are utilized for image data and MFCCs is the features extraction of sound data. In pattern recognition, one of traditional researches is handwritten digit recognition. Many approaches such as K-Nearest neighbors, neural networks, and support vector machine are applied to classify representations from image data. Nevertheless, these approaches cannot obtain expected error rates without preprocessing [8]. Research [10] presents results of handwritten digit recognition by using state-of-the-art techniques on some popular databases of CENPARMI, CEDAR and MNIST. The meaning is the behaviors of performance in features extraction and classification techniques on those well-known databases. These approaches use representation of raw data for recognition. Recently, some researches [5], [17] focused on training deep, multi-layered networks proposed the methods using higher representation from raw data images and they have been recorded a significant improvement. Higher level features of data images are representations of input image by using simple strokes. Here higher representations can be learned by a system where the knowledge of experts in some specific parts in not necessary [14], [3]. This is the reason why they are considered to higher-level features. Moreover, based on these properties, higher-level features can be applied for any kind of data including images [9], [14], audio [3], and texts [15], [16]. There are some proposed methods [13], [14], [15] using higher level features learning. However, these approaches do not take promising results for specific characteristics of raw data. In research of Olshausen [13], he proposed sparse coding algorithm to show that the ability of higher representations level learning from input signals is the simple-cells receptive fields in primary visual cortex of mammalian brain. Moreover, Honglak Lee presented method based on iteratively solving two convex optimization problems with high-dimensional images [14], [9] to get an efficient solving algorithm, or Jame Martens [12] proposed the improvement of sparse coding using weights optimizing. In this paper, we propose an efficient feature applied for clustering in preprocessing and utilize those results to cluster training images. Then the clustered sets of training image are inputted in hierarchical sparse autoencoder to solve handwritten digit recognition. The structure of paper is presented as follows.

Section II presents our proposed methods. Hierarchical sparse autoencoder is described in Section II-A. In Section II-B, we present a new feature based on linear regression-based features method (LR-based features). Those features will be automatically clustered into characteristic sets by using hierarchical sparse autoencoder. Section III describes experiments and results of our methods to show the efficiency of linear regression-based features method by making comparisons between conventional methods [5], K-Nearest Neighbor (KNN) methods [8] and some methods mentioned in [17]. Section IV is the part of discussions.

II. Our Proposed Methods

A. Hierarchical Sparse Autoencoder

Instead of training raw data like previous works, we approach hierarchically to take advantage of specific characteristics of data. Our method is inspired by Alexanders work in neuroscience. It shows that human brains have many different cortex areas to process the perceived outside world information. There are some specific cortex areas being active

to process the perceived signals. The interesting thing is that the brains active the same cortex areas when the signals are highly correlated with each other [7]. Furthermore, our observation also shows that handwritten digits are formed from primitive strokes such as straight and curve strokes. And some of handwritten digits have the correlated characteristics. In particular, the percentage of straight strokes to compose 1, 4 and 7 is more dominating than the percentage of curve strokes. In contrast, curve strokes are the key ingredient to compose 0, 2, 6 and 8 rather than straight strokes. Our approach has three main steps:

- Specific characteristics clustering.
- Higher representations learning.
- Double supervised learning.

The first step is specific characteristics clustering. A characteristic can be described as a d-dimensional vector. Let $\xi_1, \xi_2, , \xi_k$ denote k correlated characteristic sets. Particularly, we can have ξ_i containing characteristics which are highly correlated with each other. In other word, the average distance of a d-dimensional characteristic vector in ξ_i to other characteristic vectors in ξ_i is smaller than the average distance of this vector to characteristic vectors in ξ_j $(i \neq j)$. Thus, characteristics in ξ_i can be seen as specific characteristics for learning higher representations better. More specific in handwritten digits data, ξ_1 can be characteristic set containing characteristics which reveal that straight strokes are the key ingredient to compose digits. And it can be curve strokes are the key ingredient to compose digits for ξ_2. As a result, handwritten digit samples, which have characteristics belonging to the same correlated characteristic set ξ_i, will also belong to the same set described as follow

$$E_i = \{x \in \mathbb{R}^n | \ \min \delta\left(\varphi(x), \xi_i\right)\} \tag{1}$$

where E_i is an entity set containing samples have characteristics which belong to ξ_i. x is a sample of our dataset. $\varphi(x) : \mathbb{R}^n \to \mathbb{R}^d$ is a mapping function which maps a sample to a characteristic (this function is described in detail in Section II-B). And $\delta\left(\varphi(x), \xi_i\right)$ can be computed as follow

$$\delta\left(\varphi\left(x\right), \xi_i\right) = \frac{1}{|\xi_i|} \sum_{j}^{|\xi_i|} distance\left(\varphi\left(x\right), c_j\right) \tag{2}$$

where c_j is a d-dimensional characteristic vector belonging to ξ_i. The distance $\left(\varphi\left(x\right), c_j\right)$ is a distance function such as Manhattan, Euclidean and Correlation. In our approach, we use K-means algorithm for this clustering step.

The second step is higher representations learning on entity sets. Suppose at the first step the dataset has been already divided into k entity sets corresponding to k correlated characteristic sets. At this step, we train separately each entity set with one sparse autoencoder to learn weight matrices of that sparse autoencoder. These weight matrices are used to compute higher representations of raw data. Higher representations of training data are used as features to train classifiers at supervised learning phase.

The third step is double supervised learning to recognize class labels. At the first level of supervised learning, higher representations in each cluster from the second step are inputted into one softmax regression model. In our work, we use softmax regression as classifiers after sparse autoencoder because softmax regression and sparse autoencoder can be combined to become a deep learning model. The output of softmax regression is probabilistic vectors instead of class labels of input data. A probabilistic vector shows the probability for an input data to belong to each class. These vectors are then concatenated with the corresponding higher representation vectors of input data to form the features for the next level of supervised learning. The purpose of using probabilistic vectors of softmax regression is to orient for prediction or augment information for the next learning level. At the second learning level, the concatenated features are trained with SVM using 1-vs-all method. Here we apply the 1-vs-all approach instead of one-vs-one method because it helps to reduce the number of SVM classifiers and to avoid the ambiguity of outputs by using the voting of SVM results. Therefore, it will need $k \times l$ SVM models at maximum for this training step (l is the number of classes). The complete process can be interpreted as the visualization in Fig. 1 where the raw data means training image, linear regression-based feature is a method to extract regression lines in skeleton image, the entity set is the clustered sample data based on the results of K-mean algorithm applied for normalized histogram of new features.

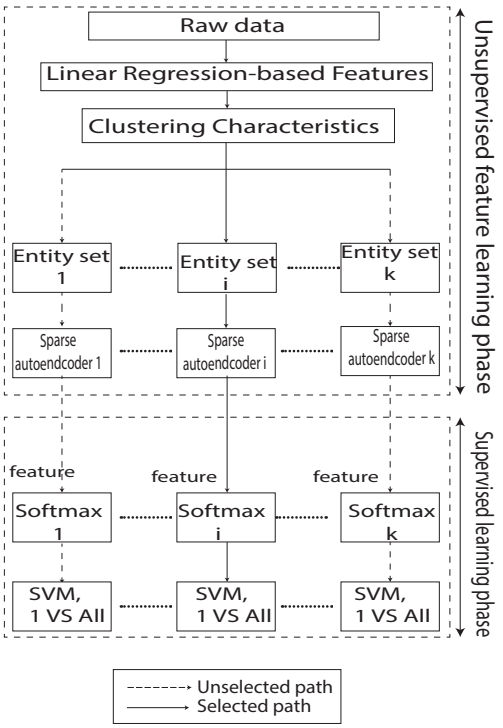
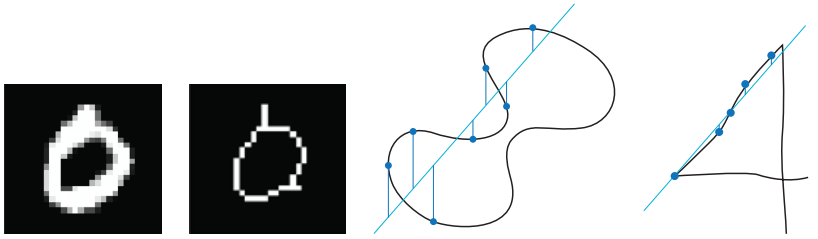


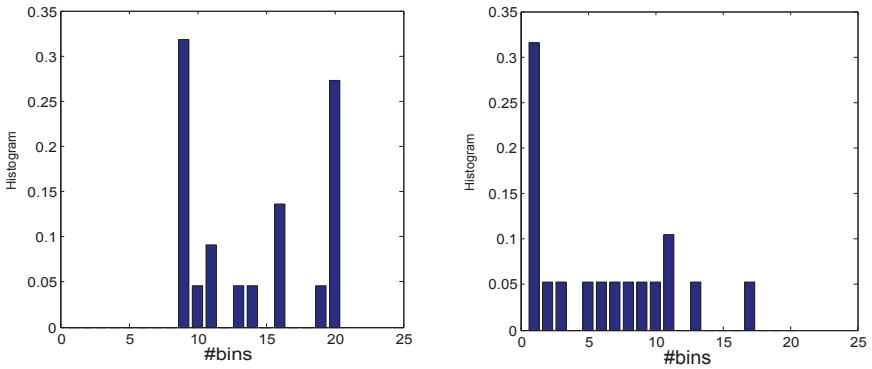
Fig. 1. Hierarchical Sparse Autoencoder

Those 3 steps described above focus on training phase of this approach. The testing phase here follows the flow of the model; however, it is quite different from the training phase. Concretely, a new testing sample can be assigned into 2 clusters at maximum. Of course the sample will belong to the cluster which has the minimum distance from the cluster centroid to the sample. In addition, the sample can also belong to one more cluster if the distance from it to the centroid of that cluster is not greater than 1.5 times the



(a) Original image (b) skeleton image (c) Curve instance (d) Straight instance

Fig. 2. Results of skeleton, and regression.



(a) Histogram of 8 (b) Histogram of 4

Fig. 3. Histogram results.

minimum distance to all clusters. After the new sample is tested with SVM models, the output of a SVM model is also a probabilistic vector. The number of probabilistic vectors is equal to the number of clusters that the sample belongs to. Finally, the maximum probability of all probabilistic vectors decides the class label of the testing sample.

B. Linear Regression-based Features

We present a new feature based on linear regression. From observations of digits (from 0 to 9) in some popular fonts as Times New Romans or Arial, we can see that there are three main kinds of characteristic. They are straight-line-segment digits of $\{1, 4, 7\}$, curve digits of $\{0, 3, 6, 8, 9\}$ and both straight and curve digits of $\{2, 5\}$. However, it is really difficult to classify the ten numbers of hand written digits based on these three kinds of characteristic. Our approach is based on statistical learning, where feature extraction must be considered carefully. Here we apply linear regression to approximate sampling points in skeleton image of sample data. As we know that linear regression constructs the best-fitting straight line for skeleton image of sample data by fitting a linear model. In one skeleton image, we have many regression lines and a set of mean square error (MSE) corresponding to them. It means that we can generally define characteristic sets of straight line segment, curve, and both straight-line segment and curve in the structure of ten digits by applying a cluster model to all histograms of MSE in one skeleton image.

Before applying linear regression, we take the skeleton of raw data. Fig. 2 b) shows the result of skeleton image. With a skeleton image, original characteristic of digit image is reserved. In this case, for the leftmost corner pixel of skeleton image, we find 8-neighbor connected path takes priority with respect to counterclockwise. For each pixel in connected path, we consider n consecutive entries which are observed data for each regression line. For instances, if connected path have m pixels then there are $m - n$ regression lines. Here we assume that the value of n is always less than that of m.

To simplify notation for linear regression in this case, we denote the equivalent optimization problem

$$minimize_\theta \, MSE = \frac{1}{n} \sum_{i=1}^{n} \left(Y_i - \widehat{Y}_i \right)^2 \tag{3}$$

where X, Y are two vectors of coordinate points in the skeleton image, θ is a vector of coefficients, and n is the number of entries of observed data. After using least mean square method, we can calculate the values of parameter θ. The interpolation value of line regression is described by the following

$$\widehat{Y}_i = \theta_0 + \theta_1 X_i \tag{4}$$

In practice, we set up $(n = \frac{size(X)}{2})$. We can see that mean square error (MSE) of regression lines in digits written by curve strokes is always greater than that of one digit written by straight strokes. This can help make distinct characteristic of digits. From that, for each raw image data, we obtain set of mean square errors represents for characteristics of digits. Here if m is the number of pixels in skeleton image, we will have $m - n$ values of MSE corresponding to $m - n$ regression lines in skeleton image. After linear regression calculation, we map the set of MSE in each skeleton image into histogram space

having the the same number of bins. It helps to eliminate the difference of the number of regression lines in each skeleton image. After that, the value of histogram is normalized to $[0..1]$ by dividing the total value of accumulation in its bins. Let H_i denote the histogram of MSE set for i^{th} sample data, ε_i denote the normalized histogram corresponding to H_i.

$$\varepsilon_i = \frac{H_i}{\sum_j H_i(j)} \tag{5}$$

We will cluster $\varepsilon = \{\varepsilon_1, \varepsilon_2, ...\}$ into K classes of characteristic set. The results of clustering by K-mean algorithm on MSE histogram is utilized to cluster raw sample data into the characteristic set ξ and it will be inputed to hierarchical sparse autoencoder as mentioned in Section II-A. In practice, the value of K is set up in the range of $[2, .., 5]$ to select the best performance in test database. The number of bins is calculated on MSE set is 20 with $width = 0.05$ and choose $min = 0$. $\varepsilon \in \mathbb{R}^{20\times 1}$ and is presented in Fig. 3, which describes two instance of histogram corresponding to Fig. 2(c) and Fig. 2(d). The output of this approach is sample ξ. Section III shows the efficiency of Linear Regression-based Features on hierarchical sparse autoencoder.

III. Experiments and Results

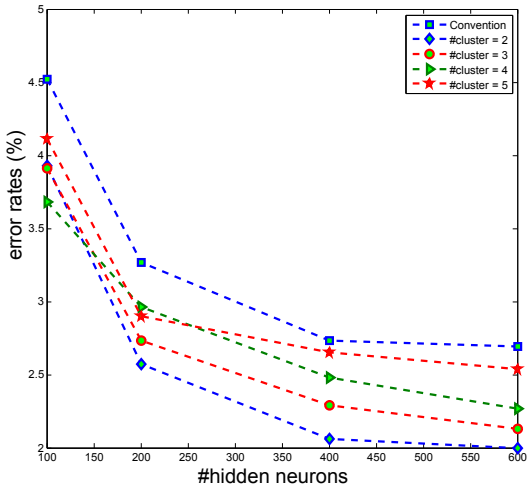
We conduct three experiments on MNIST and USPS [4] dataset to evaluate the performance of our approach on various distance functions. In experiments, 60000 and 7291 samples are used for training and 10000 and 2007 samples are used for testing on MNIST and USPS dataset respectively. In addition, our proposed methods are also evaluated the performance on many clustering ways to find out the most suitable clustering way for this problem. Meanwhile, the conventional method [5] just feeds raw data into a sparse autoencoder to learn features. Then, the features are used to recognize the digit at supervised learning phase. So it does not take the correlation of characteristics of data. Our purpose is to show the efficiency of using linear regression-based features in Hierarchical Sparse Autoencoder as the feature extraction of classification model. Therefore, two following experiments are utilized to evaluate the error rates of some baselines of Geoffrey Hinton [5], K-Nearest Neighbor of YanLecun [8] and Nonlinear learning [17] with two our proposed methods including hierarchical sparse autoencoder without clustering and one using linear regression-based features in preprocessing. Additionally, we also evaluate the efficiency of linear regression-based features on USPS dataset on the third experiment. All experiments were performed on a computer of Intel(R) with Core(TM) i5, CPU 650@ 3.20Hz 4.00GB RAM and 64-bit Operating system.

A. Hierarchical Sparse Autoencoder without clustering on MNIST dataset

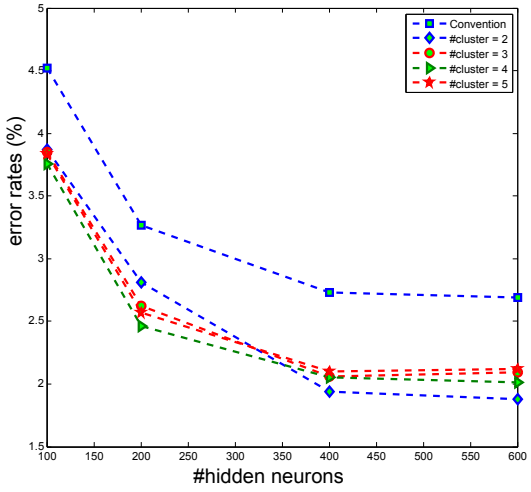
In this experiment, the purpose is to evaluate the performance of our method without linear regression-based features on Euclidean distance used for clustering characteristics. The number of hidden neurons of each autoencoder is in turn set at 100, 200, 400 and 600 neurons. K-means plays a role as a clustering algorithm to form correlated characteristic sets and entity sets. Hence, we in turn evaluate the possibility of clustering of specific characteristics by K-means. This can be evaluated by variously setting the number of clusters of K-means.

TABLE I. Error rates (%) of our approach using Euclidean distance to cluster data without linear regression-based features.

Method		Conventional method [5]	Our method			
			#Clusters			
			2	3	4	5
#hidden neurons	100	4.52	**3.93**	**3.91**	**3.68**	**4.11**
	200	3.27	**2.57**	**2.73**	**2.96**	**2.90**
	400	2.73	**2.06**	**2.29**	**2.48**	**2.65**
	600	2.69	**2.00**	**2.13**	**2.27**	**2.54**



(a) Non-preprocessing



(b) Using linear regression-based features

Fig. 4. Visualization of error rates (%) of our methods

The result of the first experiment is shown in Table I. Our method on all clusters got the error rates lower than the conventional method. The results also show that when the number of hidden neurons increases, the error rates decrease on the conventional method and our method (without linear regression-based features) as well. However, it does not do the same thing for number of clusters. Concretely, the error rates

decrease from 2 clusters to 3 clusters and then they increase to 5 clusters. This can be seen clearly as visualization in Fig. 4(a). With the lower error rates in most cases, we can have the most suitable number of clusters for Euclidean distance is 2.

B. Hierarchical Sparse Autoencoder using linear regression-based features in preprocessing on MNIST dataset

In the second experiment, we use linear regression-based features for hierarchical sparse autoencoder. The number of hidden neurons and the number of clusters are evaluated same with the experiment III-A. As shown in Table II is the error rates of this experiment, they also decrease when the number of hidden neurons increases like the experiment III-A.

TABLE II. ERROR RATES (%) OF OUR APPROACH USING EUCLIDEAN DISTANCE TO CLUSTER DATA USING LINEAR REGRESSION-BASED FEATURES IN PREPROCESSING FOR MNIST DATASET.

Method		Conventional method [5]	Our method			
			#Clusters			
			2	3	4	5
#hidden neurons	100	4.52	**3.86**	**3.85**	**3.75**	**3.84**
	200	3.27	**2.81**	**2.62**	**2.46**	**2.57**
	400	2.73	**1.94**	**2.06**	**2.05**	**2.10**
	600	2.69	**1.87**	**2.09**	**2.00**	**2.12**

Nevertheless, a large number of clusters do not decrease error rates more than the 2 clusters. This result is visualized in Fig. 4(b). According to the result, 2 clusters are the most suitable number of clusters when using linear regression-based features. Fig. 4 also presents the error rates of our proposed method without linear regression-based features is worse than using linear regression-based features. At the (cluster,neuron) values of (5,100), our method without linear regression based features is not quite efficient with error rate of 4.11%. However, error rate in the same case of the neuron number equal to 100 is still lower than conventional method.

In addition, this experiment shows the efficiency of using linear regression-based features in preprocessing. The error rates are lower than that of experiment III-A in the most of cases. Fig. 5 represents average error rates follow as cluster of 2, 3, 4, and 5 in experiment III-A and III-B.

From Fig. 5, we can see the efficiency of using linear regression-based features which can get more promising results. The error rates decrease to the best result in this experiment is 1.87% with #neurons = 600 at the cluster number of 2. The error rates of hierarchical sparse autoencoder without linear regression-based features increase more rapidly than that of using preprocessing.

As we can see in Fig. 6 is the visualization of trained weights of conventional method and our method. Obviously, the trained weights of our method are more correlated with each other in the same cluster and more distinct with ones in other clusters. Particularly, the visualization of trained weights in cluster 1 seems straight strokes and cluster 3 seems curve strokes. This helps higher representations extracted from sparse autoencoders are better than the old ones using conventional method. Because the trained weights of conventional method are more confusing rather than our method.

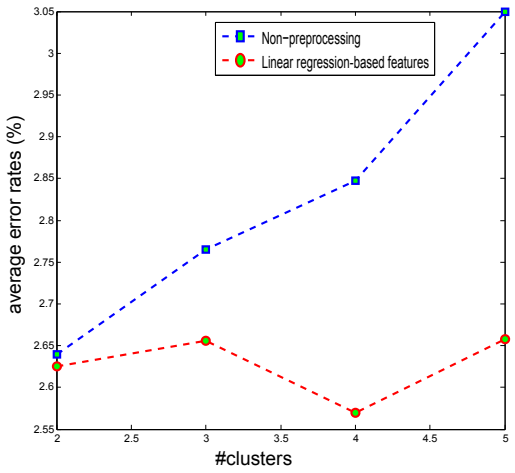


Fig. 5. Visualization of average error rates (%) in experiment III-A and III-B .

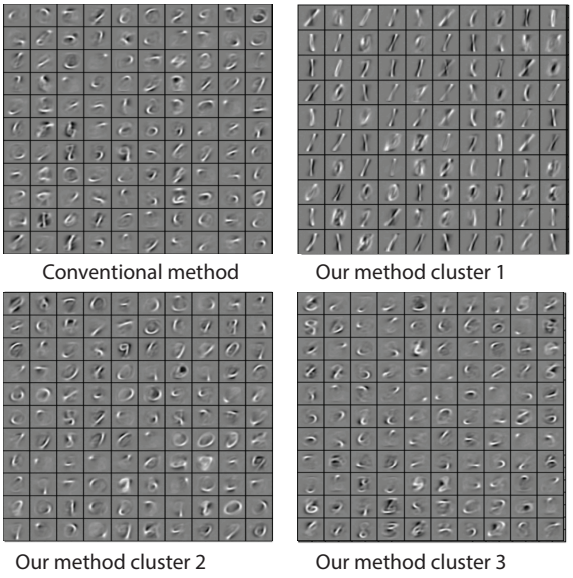


Fig. 6. Visualization of learnt weights of sparse autoencoder using original approach (a), cluster 1 of our approach (b), cluster 2 of our approach (c) and cluster 3 of our approach (d).

For comparison, we also provide several methods in two publications [8], [17]. These architectures obtain some specific benchmarks. Error rates of our proposed method (using LR-based features) get the lower error rate. Our propose hierarchical sparse autoecoder without preprocessing (Non LR-based features) does not achieve significant error rates with architectures of Linear SVM with local coordinate coding ($|C| = 4096$) and Sparse coding in the same context. Fig. 7 visualizes this point. In addition, Method of multi-column deep neural networks (MCDNN) [1] achieved the state-of-the-art result with an impressive error rate decreasing to 0.23 percentage on MNIST by using fast parallel programming power of graphics processing unit (GPUs).

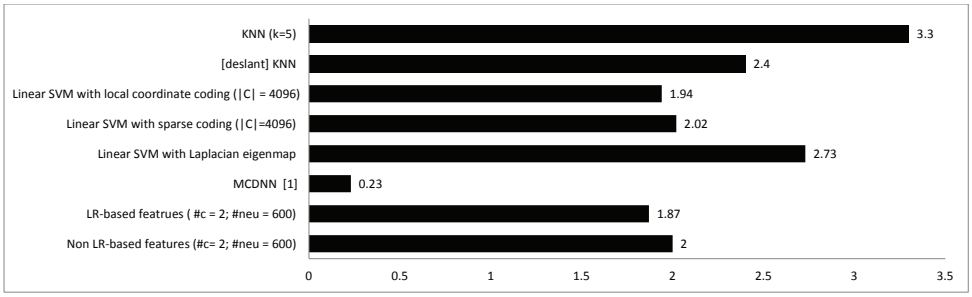


Fig. 7. Visualization of Error rates on test set (%) of several methods, [deslant] presents the method in context of deslanted version database mentioned in [8]. MCDNN [1] is multi-deep column neural networks achieved the current state-of-the-art results on MNIST and another datasets as well. LR-based features is linear regression-based features, which is our proposed method.

C. Hierarchical Sparse Autoencoder using linear regression-based features in preprocessing on USPS dataset

We also try to test our proposed method with the same convention in experiment III-B for USPS dataset. The results is presented in Table III. Follow to the results of conventional

TABLE III. ERROR RATES (%) OF OUR APPROACH USING EUCLIDEAN DISTANCE TO CLUSTER DATA USING LINEAR REGRESSION-BASED FEATURES IN PREPROCESSING FOR USPS DATASET.

Method		Conventional method [5]	Our method			
			#Clusters			
			2	3	4	5
#hidden neurons	100	6.63	**6.02**	**6.56**	6.82	6.82
	200	6.23	**5.48**	**5.92**	6.32	6.42
	400	6.28	**5.43**	**5.92**	6.97	6.77
	600	6.08	**5.23**	**5.77**	6.57	6.57

method; we only achieve the lower error rates at clusters of 2 and 3. Hence, an efficient clustering in preprocessing like linear regression-based features approach can get much more promising results.

IV. CONCLUSION

As mentioned in Section III-B, our proposed methods can not get better result than MCDNN [1] because of more powerful computer system on GPUs and training deep and wide on receptive fields of convolutional winner-take-all neurons networks with large parameters of hundreds of maps per layer. Additionally, they combine many benchmark architectures like convolutional deep neural networks (DNN) and winner-take-all to get impressive results. However, the significant advantage of our proposed methods is simple methods (linear regression-based features and raw image classification) with the lower number of layer networks.

V. ACKNOWLEDGEMENT

This work is financially supported by National Foundation for Science and Technology Development of Vietnam (NAFOSTED).

REFERENCES

[1] Dan C. Ciresan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *CVPR*, pages 3642–3649. IEEE, 2012.

[2] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR (1)*, pages 886–893, 2005.

[3] Roger Grosse, Rajat Raina, Helen Kwong, and Andrew Y. Ng. Shift-invariance sparse coding for audio classification. *CoRR*, abs/1206.5241, 2012.

[4] Trevor Hastie, Robert Tibshirani, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction: with 200 full-color illustrations*. New York: Springer-Verlag, 2001.

[5] Geoffrey Hinton and Ruslan Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504 – 507, 2006.

[6] Lennig M. Hunt, M. and Mermelstein. Experiments in syllable-based recognition of continuous speech. In *In Proceedings of International Conference on Acoustics, Speech and Signal Processing*, pages 880–883, 1996.

[7] Alexander G. Huth, Jack L. Gallant An T. Vu, and Shinji Nishimoto. A continuous semantic space describes the representation of thousands of object and action categories across the human brain. *Neuron*, 2012.

[8] Yann LeCun, Lon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, volume 86, pages 2278–2324, 1998.

[9] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *ICML*, page 77, 2009.

[10] Cheng-Lin Liu, Kazuki Nakashima, Hiroshi Sako, and Hiromichi Fujisawa. Handwritten digit recognition: benchmarking of state-of-the-art techniques. *Pattern Recognition*, 36(10):2271–2285, 2003.

[11] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[12] James Martens. Deep learning via hessian-free optimization. In Johannes Fürnkranz and Thorsten Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 735–742, Haifa, Israel, June 2010. Omnipress.

[13] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

[14] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y. Ng. Self-taught learning: Transfer learning from unlabeled data. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, 2007.

[15] Patrice Simard, David Steinkraus, and John C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *ICDAR*, pages 958–962, 2003.

[16] Richard Socher, Eric H. Huang, Jeffrey Pennin, Andrew Y. Ng, and Christopher D. Manning. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In J. Shawe-Taylor, R.S. Zemel, P. Bartlett, F.C.N. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 801–809. 2011.

[17] Kai Yu, Tong Zhang, and Yihong Gong. Nonlinear learning using local coordinate coding. In Yoshua Bengio, Dale Schuurmans, John D. Lafferty, Christopher K. I. Williams, and Aron Culotta, editors, *NIPS*, pages 2223–2231. Curran Associates, Inc., 2009.