Global Journal of Computer Science and Technology: F
Graphics & vision
Volume 17 Issue 2 Version 1.0 Year 2017
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Towards Arabic Alphabet and Numbers Sign Language Recognition

By Ahmad Hasasneh & Sameh Taqatqa
Palestine Ahliya University

Abstract- This paper proposes to develop a new Arabic sign language recognition using Restricted Boltzmann Machines and a direct use of tiny images. Restricted Boltzmann Machines are able to code images as a superposition of a limited number of features taken from a larger alphabet. Repeating this process in deep architecture (Deep Belief Networks) leads to an efficient sparse representation of the initial data in the feature space. A complex problem of classification in the input space is thus transformed into an easier one in the feature space. After appropriate coding, a softmax regression in the feature space must be sufficient to recognize a hand sign according to the input image. To our knowledge, this is the first attempt that tiny images feature extraction using deep architecture is a simpler alternative approach for Arabic sign language recognition that deserves to be considered and investigated.

Keywords: component; arabic sign language recognition, restricted boltzmann machines, deep belief networks, softmax regression, classification, sparse representation.

GJCST-F Classification: I.5, I.7.5

Towards Arabic Alphabet and Numbers Sign Language Recognition

Ahmad Hasasneh α & Sameh Taqatqa σ

Abstract- This paper proposes to develop a new Arabic sign language recognition using Restricted Boltzmann Machines and a direct use of tiny images. Restricted Boltzmann Machines are able to code images as a superposition of a limited number of features taken from a larger alphabet. Repeating this process in deep architecture (Deep Belief Networks) leads to an efficient sparse representation of the initial data in the feature space. A complex problem of classification in the input space is thus transformed into an easier one in the feature space. After appropriate coding, a softmax regression in the feature space must be sufficient to recognize a hand sign according to the input image. To our knowledge, this is the first attempt that tiny images feature extraction using deep architecture is a simpler alternative approach for Arabic sign language recognition that deserves to be considered and investigated.

Keywords: component; arabic sign language recognition, restricted boltzmann machines, deep belief networks, softmax regression, classification, sparse representation.

I. Introduction

Sign language continues to be the best method to communicate between the deaf and hearing impaired. Hand gestures enable communication between deaf people during their daily lives rather than speaking. In our society, Arabic Sign Language (ArSL) is only known for deaf people and specialists, thus the community of deaf people is narrow. To help people with normal hearing communicate effectively with the deaf and the hearing-impaired, numerous systems have been developed for translating diverse sign languages from around the world. Several review papers have been published that discuss such systems and they can be found in [1]–[7].

Generally, the process of ArSL recognition (ArSLR) can be achieved through two main phases: detection and classification. In stage one, each given image is pre-processed, improved, and then the regions of interest (ROI) is segmented using a segmentation algorithm. The output of the segmentation process can thus be used to perform the sign recognition process. Indeed, accuracy and speed of detection play an important role in obtaining accurate and fast recognition process. In the recognition stage, a set of features (patterns) for each segmented hand sign is first extracted and then used to recognize the sign. These features can be used as a reference to understand the differences among the classes.

Recognizing and documenting of ArSL have only been paid attention recently, where few attempts have investigated and addressed this problem, see for example [8]–[11]. The question of ArSL recognition is therefore a major requirement for the future of ArSL. It facilitates the communication between the deaf and normal people by recognizing the alphabet and numbers signs of Arabic sign language to text or speech. To achieve that goal, this paper proposes a new Arabic sign recognition system based on new machine learning methods and a direct use of tiny images.

The rest of the paper is organized as follows. Section2 presents the current approaches to Arabic alphabet sign language recognition (ArASLR). Section 3 describes the proposed model for ArASLR. Conclusions and future works are presented in section 4.

II. Current Approaches

Studies in Arabic sign language recognition, although not as advanced as those devoted to other scripts (e.g. Latin), have recently shown interest [8]–[11]. We have also seen that current research in ArSLR has only been satisfactory for alphabet recognition with accuracy exceeding 98%. Isolate Arabic word recognition has only been successful with medium-size vocabularies (less than 300 signs). On the other hand, continuous ArSLR is still in its early stages, with very restrictive conditions.

Current approaches on sign language recognition usually falls into two major approaches. The first one is sensors based approaches, which employs sensors attached to the glove. Look-up table software is usually provided with the glove to be used for hand gesture recognition. Recent sensors based approaches can be found, for instance, in [11]–[14]. The second approaches, vision-based analysis, are based on the use of video cameras to capture the movement of the hand that is sometimes aided by making the signer wear a glove that has painted areas indicating the positions of the fingers and the wrist then use those measurements in the recognition process. Image-based techniques exhibit a number of challenges. These include: lighting conditions, image background, face and hands segmentation, and different types of noise.

Author α σ: Information Technology Department Palestine Ahliya University Bethlehem, West Bank, Palestine.
e-mails: ahasasneh@paluniv.edu.ps, sameh@paluniv.edu.ps

Among of image-based approaches, some authors [15] introduced a method for automatic recognition of Arabic sign language alphabet. For feature extraction, Hus moments were used followed by support vector machines (SVMs) to perform the classification process. A correct recognition rate of 87% was achieved. Other authors in [16] developed a neuro-fuzzy system. The proposed system includes five main steps: image acquisition, filtering, segmentation, and hand outline detection, followed by feature extraction. Bare hands were considered in the experiments, achieving a recognition accuracy of 93.6%. In [17], the authors proposed an adaptive neuro-fuzzy inference system for alphabet sign recognition. A colored glove was used to simplify the segmentation process, and geometric features were extracted from the hand region. The recognition rate was improved to 95.5%. In [18], the authors developed an image-based ArSL system that does not use visual markings. The images of bare hands are processed to extract a set of features that are translation, rotation, and scaling invariant. A recognition accuracy of 97.5% was achieved on a database of 30 Arabic alphabet signs. In [19], the authors used recurrent neural networks for alphabet recognition. A database of 900 samples, covering 30 gestures performed by two signers, was used in their experiments. The Elman network achieved an accuracy rate of 89.7%, while a fully recurrent network improved the accuracy to 95.1%. The authors extended their work by considering the effect of different artificial neural network structures on the recognition accuracy. In particular, they extracted 30 features from colored gloves and achieved an overall recognition rate of 95% [20].

A recent paper reviews the different systems and methods for the automatic recognition of Arabic sign language can be found in [7]. It highlights the main challenges characterizing Arabic sign language as well as potential future research directions. Recent works on image-based recognition of Arabic sign language alphabet can be found in [9], [10], [21]–[25]. In particular, Naoum et al. [9] proposes an ArSLR using KNN. To achieve good recognition performance, they proposed to combine this algorithm with a glove based analysis technique. The system starts by finding histograms of the images. Profiles extracted from such histograms are then used as input to a KNN classifier. Mohandes [10] proposes a more sophisticated recognition algorithm to achieve high performance of ArSLR. The first attempt to recognize two-handed signs from the Unified Arabic Sign Language Dictionary using the CyberGlove and SVMs to perform the recognition process. PCA is used for feature extraction. The authors in [21] proposed an Arabic sign language alphabet recognition system that converts signs into voice. The technique is much closer to a real-life setup; however, recognition is not performed in real time. The system focuses on static and simple moving gestures. The inputs are color images of the gestures. To extract the skin blobs, the YCbCr space is used. The Prewitt edge detector is used to extract the hand shape. To convert the image area into feature vectors, principal component analysis (PCA) is used with a K-Nearest Neighbor Algorithm (KNN) in the classification stage. Furthermore, the authors in [22] and [23] proposed a pulse-coupled neural network (PCNN) ArSLR system able to compensate for lighting nonhomogeneity and background brightness. The proposed system showed invariance under geometrical transforms, bright background, and lighting conditions, achieving a recognition accuracy of 90%. Moreover, the authors in [24] introduced an Arabic Alphabet and Numbers Sign Language Recognition (ArANSLR). The phases of the proposed algorithm consists of skin detection, background exclusion, face and hands extraction, feature extraction, and also classification using Hidden Markov Model (HMM). The proposed algorithm divides the rectangle surrounding by the hand shape into zones. The best number of zones is 16 zones. The observation of HMM is created by sorting zone numbers in ascending order depending on the number of white pixels in each zone. Experimental results showed that the proposed algorithm achieves 100% recognition rate.

On the other hand, new systems for facilitating human machine interaction have been introduced recently. In particular, the Microsoft Kinect and the leap motion controller (LMC) have attracted special attention. The Kinect system uses an infrared emitter and depth sensors, in addition to a high resolution video camera. The LMC uses two infrared cameras and three LEDs to capture information within its interaction range. However, the LMC does not provide images of detected objects. The LMC has recently been used for Arabic alphabet sign recognition with promising results [25].

After presenting the different existing image-based approaches that have been used to achieve ArASLR, we have noted that these approaches generally include two main phases of coding and classification. We have also seen that most of the coding methods are based on hand-crafted feature extractors, which are empirical detectors. By contrast, a set of recent methods based on deep architectures of neural networks give the ability to build it from theoretical considerations.

ArSLR therefore requires projecting images onto an appropriate feature space that allows an accurate and rapid classification. Contrarily to these empirical methods mentioned above, new machine learning methods have recently emerged which strongly related to the way natural systems code images [26]. These methods are based on the consideration that natural image statistics are not Gaussian as it would be if they have had a completely random structure [27]. The auto-similar structure of natural images allowed the evolution to build optimal codes. These codes are made of statistically independent features and many different methods have been proposed to construct them from image datasets. Imposing locality and sparsity constraints in these features is very important. This is probably due to the fact that any simple algorithms based on such constraints can achieve linear signatures similar to the notion of receptive field in natural systems. Recent years have seen an interesting interest in computer vision algorithms that rely on local sparse image representations, especially for the problems of image classification and object recognition [28]–[32]. Moreover, from a generative point of view, the effectiveness of local sparse coding, for instance for image reconstruction [33], is justified by the fact that an natural image can be reconstructed by a smallest possible number of features. It has been shown that Independent Component Analysis (ICA) produces localized features. Besides it is efficient for distributions with high kurtosis well representative of natural image statistics dominated by rare events like contours; however the method is linear and not recursive. These two limitations are released by DBNs [34] that introduce nonlinearities in the coding scheme and exhibit multiple layers. Each layer is made of a RBM, a simplified version of a Boltzmann machine proposed by Smolensky [35] and Hinton [36]. Each RBM is able to build a generative statistical model of its inputs using a relatively fast learning algorithm, Contrastive Divergence (CD), first introduced by Hinton [36]. Another important characteristic of the codes used in natural systems, the sparsity of the representation [26], is also achieved in DBNs. Moreover, it has been shown that these approaches remain robustness to extract local sparse efficient features from tiny images [37]. This model has been successfully used in [32] to achieve semantic place recognition. The hope is to demonstrate that DBNs coupled with tiny images can also be successfully used in the context of ArASLR.

III. Proposed Model

The methodology of this research mainly includes four stages (see figure 1) which can be summarized as follows: 1) data collection and image acquisition, 2) image pre-processing, 3) feature extraction and finally 4) gesture recognition.

a) Description of the Database

The alphabet used for Arabic sign language is displayed in Figure 2, left [38], will be used to investigate the performance of the proposed model. In this database, the signer performs each letter separately. Mostly, letters are represented by a static posture, and the vocabulary size is limited. In this section, several methods for image-based Arabic sign language alphabet recognition are discussed. Even though the Arabic alphabet only consists of 28 letters, Arabic sign language uses 39 signs. The 11 additional signs represent basic signs combining two letters. For example, the two letters “ال” are quite common in Arabic (similar to the article “the” in English). Therefore, most literature on ArASLR uses these basic 39 signs.

b) Image Pre-processing

The typical input dimension for a DBN is approximately 1000 units (e.g. 30x30 pixels). Dealing with smaller patches could make the model unable to extract interesting features. Using larger patches can be extremely time-consuming during feature learning. Additionally the multiplication of the connexion weights acts negatively on the convergence of the CD algorithm. The question is therefore how could we scale the size of realistic images (e.g. 300x300 pixels) to make them appropriate for DBNs?

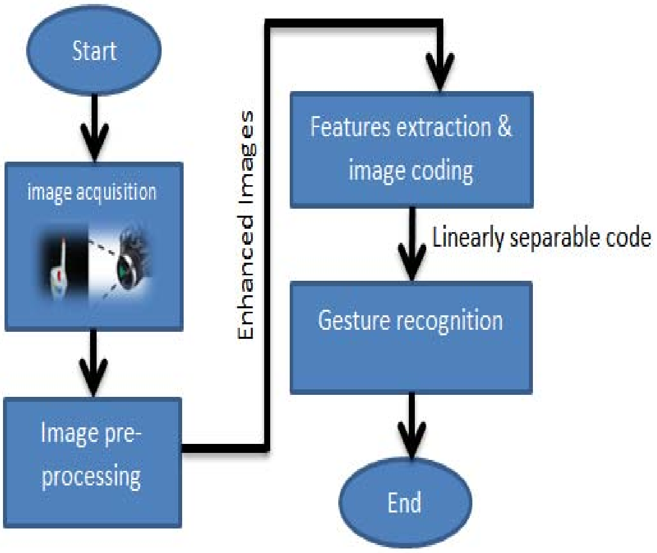

Figure 1: Proposed model

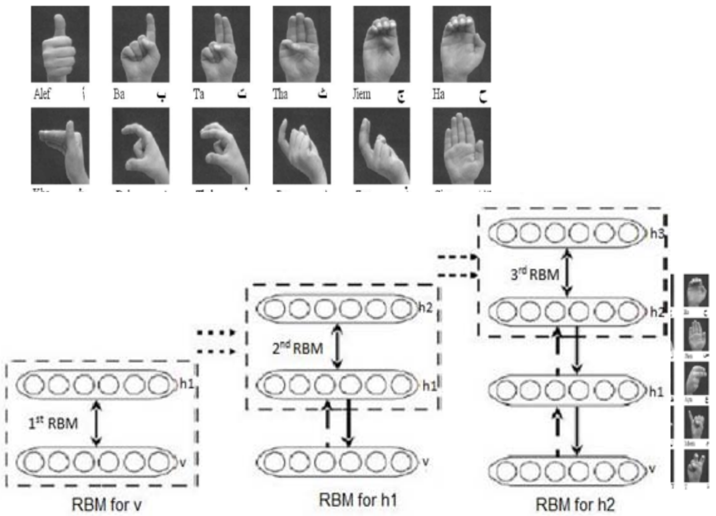

Figure 2: Left: Original Arabic sign language alphabet. Right: The corresponding tiny images of Arabic sign language alphabet. One can see that, despite the size reduction, these small images remain fully recognizable

Three solutions can be envisioned. The first one is to select random patches from each image as done in [39], the second is the use of convolutional architectures, as proposed in [40], and the last one is to reduce the size of each image to a tiny image as proposed in [37]. The first solution extracts local features and the characterization of an image using these features can only be made using BoWs approaches we wanted to avoid. The second solution shows the same limitations as the first one and additionally gives raise to extensive computations that are only tractable on Graphics Processing Unit architectures. Features extraction using random patches is irrespective of the spatial structures of each image [41]. In the case of structured scenes like the ones used in semantic place recognition these structures bear an interesting information.

Besides, tiny images have been successfully used in [37] for classifying and retrieving images from the 80-million images database developed at MIT. Torralba in [37] showed that the use of tiny images combined with a DBN approach led to code each image by a small binary vector defining the elements of a feature alphabet that can be used to optimally define the considered image. The binary vector acts as a bar-code while the alphabet of features is computed only once from a representative set of images. The power of this approach is well illustrated by the fact that a relatively small binary vector largely exceeds the number of images that have to be coded even in a huge database (2256 ≈ 1075). So, for all the se reasons we have chosen image reduction.

On the other hand, natural images are highly structured and contain significant statistical redundancies, e.g. their pixels have strong correlations [42], [43]. Removing these correlations is known as whitening. It has been shown that whitening is a mandatory step for the use of clustering methods in object recognition [44]. Whitening being a linear process and it does not remove the higher order statistics present in the data.

As a consequence, as proposed by [37] and [32], after color conversion and image cropping, the image size is reduced to 42x24 as shown in figure 1. The final set of tiny images is centered and whitened in order to eliminate order 2 statistics. Consequently the variance in equation 6 will be set to 1. Contrarily to [37], the 42x24 = 1008 pixels of the whitened images will be used directly as the input vector of the network for features extraction purpose.

c) *Features Extraction*

Next the feature extraction stage comes. This stage is the most significant stage which is based on using a new unsupervised machine learning model DBNs. DBNs are probabilistic generative models composed of multiple RBMs layers of latent stochastic variables. The latent variables typically have binary values. They correspond to hidden units or feature detectors. The input variables are zero-mean Gaussian activation units and are often used to reconstruct the visible units. As shown in figure 3, the top two layers have undirected, symmetric connections between them and they form the weights or the features. These features are extracted using the principle of energy function minimization according to the quality of the image reconstruction. It has been shown that features extracted by DBNs are more promising for image classification than hand-engineered features [32], [45], [46]. So, we hope that, due to the statistical independence of the features and their sparse nature, learning in the feature space will become linearly independent, greatly simplifying the way we will learn to classify the signs.

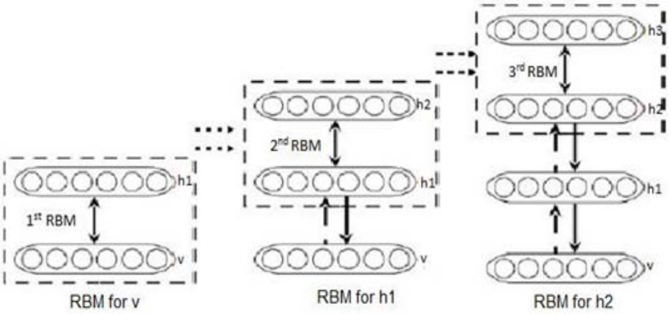

Figure 3: Stacking Restricted Boltzmann Machines (RBM) to achieve Deep Belief Network. This figure also illustrates the layer-wise training of a DBN

1) Gaussian-Bernoulli Restricted Boltzmann Machines

Unlike a classical Boltzmann Machine, a RBM is a bipartite undirected graphical model, $\theta = \{w_{ij}, b_i, c_j\}$ linking, through a set of weights w_{ij} between visible and hidden units and biases $\{b_i, c_j\}$ a set of visible units v to a set of hidden units h [27]. For a standard RBM, a joint configuration of the binary visible units and the binary hidden units has an energy function, given by:

$$E(\boldsymbol{v}, \boldsymbol{h}; \theta) = -\sum_{i,j} v_i h_j w_{ij} - \sum_i b_i v_i - \sum_j c_j h_j \qquad (1)$$

Probabilities of the state for a unit in one layer conditional to the state of the other layer can therefore be easily computed. According to Gibbs distribution:

$$P(\boldsymbol{v}, \boldsymbol{h}; \theta) = -\frac{1}{Z(\theta)} exp^{-E(\boldsymbol{v},\boldsymbol{h},\theta)} \qquad (2)$$

where $Z(\theta)$ is a normalizing constant.

Thus after marginalization:

$$P(\boldsymbol{h}, \theta) = \sum_{\boldsymbol{v}} P(\boldsymbol{v}, \boldsymbol{h}; \theta) \qquad (3)$$

it can be derived [47] that the conditional probabilities of a standard RBM are given as follows:

$$P(h_j = 1 | \boldsymbol{v}; \theta) = \sigma\left(c_j + \sum_i w_{ij} v_i\right) \qquad (4)$$

$$P(v_i = 1 | \boldsymbol{h}; \theta) = \sigma\left(b_i + \sum_j w_{ij} h_j\right) \qquad (5)$$

where $\sigma(x) = 1/(1 + e^{-x})$ is the logistic function.

Since binary units are not appropriate for multi-valued inputs like pixel levels, as suggested by Hinton [48], in the present work visible units have a zero-means Gaussian activation scheme:

$$P(v_i = 1 | \boldsymbol{h}; \theta) = \mathrm{N}\left(b_i + \sum_j w_{ij} h_j , \sigma^2\right) \qquad (6)$$

In this case, the energy function of Gaussian-Bernoulli RBM is given by:

$$E(\boldsymbol{v}, \boldsymbol{h}; \theta) = -\sum_i \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_j c_j h_j - \sum_{i,j} \frac{v_i}{\sigma_i} h_j w_{ij} \qquad (7)$$

2) Learning RBM Parameters

One way to learn RBM parameters is through the maximization of the model log likelihood in a gradient ascent procedure. The partial derivative of the log-likelihood for an energy-based model can be expressed as follows:

$$\frac{\partial}{\partial\theta} L(\theta) = -\langle\frac{\partial E(\boldsymbol{v}, \theta)}{\partial\theta}\rangle_{data} + \langle\frac{\partial E(\boldsymbol{v}, \theta)}{\partial\theta}\rangle_{model} \qquad (8)$$

where $\langle\ \rangle_{model}$ is an average with respect to the model distribution and $\langle\ \rangle_{data}$ an average over the sample data. The energy function of a RBM is given by:

$$E(\boldsymbol{v}, \theta) = log \sum\nolimits_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h};\theta)} \qquad (9)$$

and

$$\partial E(\boldsymbol{v}, \theta)/\partial\theta = \sum\nolimits_{\boldsymbol{h}} p(\boldsymbol{h}|\boldsymbol{v}; \theta) \partial E(\boldsymbol{v}, \boldsymbol{h}; \theta)/\partial\theta \qquad (10)$$

Unfortunately, computing the likelihood needs to compute the partition function, , that is usually intractable. However, Hinton [28] proposed an alternative learning technique called Contrastive Divergence (CD). This learning algorithm is based on the consideration that minimizing the energy of the network is equivalent to minimize the distance between the data and a statistical generative model of it. A comparison is made between the statistics of the data and the statistics of its representation generated by Gibbs sampling. Hinton [36] showed that usually only a few steps of Gibbs sampling (most of the time reduced to one) are sufficient to ensure convergence. For a RBM, the weights of the network can be updated using the following equation:

$$w_{ij} \leftarrow w_{ij} + \eta\left(\langle v_i^0 h_j^0 \rangle - \langle v_i^n h_j^n \rangle\right) \tag{11}$$

where η is the learning rate, v^0 corresponds to the initial data distribution, h^0 is computed using equation 4, v^n is sampled using the Gaussian distribution in equation 6 and with n full steps of Gibbs sampling, and h^n is again computed from equation 4.

3) Layerwise Training for Deep Belief Networks

A DBN is a stack of RBMs trained in a greedy layer-wise and bottom-up fashion introduced by [34]. The first model parameters are learned by training the first RBM layer using the contrastive divergence. Then, the model parameters are frozen and the conditional probabilities of the first hidden unit values are used to generate the data to train the higher RBM layers. The process is repeated across the layers to obtain a sparse representation of the initial data that will be used as the final output.

d) Gesture Recognition

Assuming that the non-linear transform operated by DBN improves the linear separability of the data, a simple regression method will be used to perform the classification process. To express the final result as a probability that a given sign means one thing, we normalize the output with a softmax regression method. According to maximum likelihood principles, the largest probability value gives the decision of the system. The classification process will also be investigated using a more sophisticated classifier, a SVM classification method instead of softmax regression. In case of comparable results; this will underline that the DBN computes a linear separable signature of the initial data.

IV. Experimental Results

For this task, we have conducted an experiment using the pre-processed dataset (the tiny-normalized dataset) which are randomly sampled from the Arabic Alphabet dataset which contains 28 letters. A complete structure (1024-1024) of the first RBM layer was used for this case. Figure shows features extracted using the locally normalized data. These features remain sparse but cover a broader spectrum of spatial frequencies. An interesting observation is that they look closer to the ones obtained with convolutional networks [40] for which no whitening is applied to the initial dataset.

The features shown in figure 4 have been extracted by training the first RBM layer on 6000 normalized image patches (32x32 pixels) sampled from the Arabic Alphabet database. One can see that the extracted features represent most of the 28 signs of the letters. Some others are localized and correspond to small parts of the initial views, like edges and corners that can be identified as hand elements (i.e. they are not specific of a given sign). These features can thus be used to code the initial data to achieve the linear separability, which will greatly simplifies the recognition process.

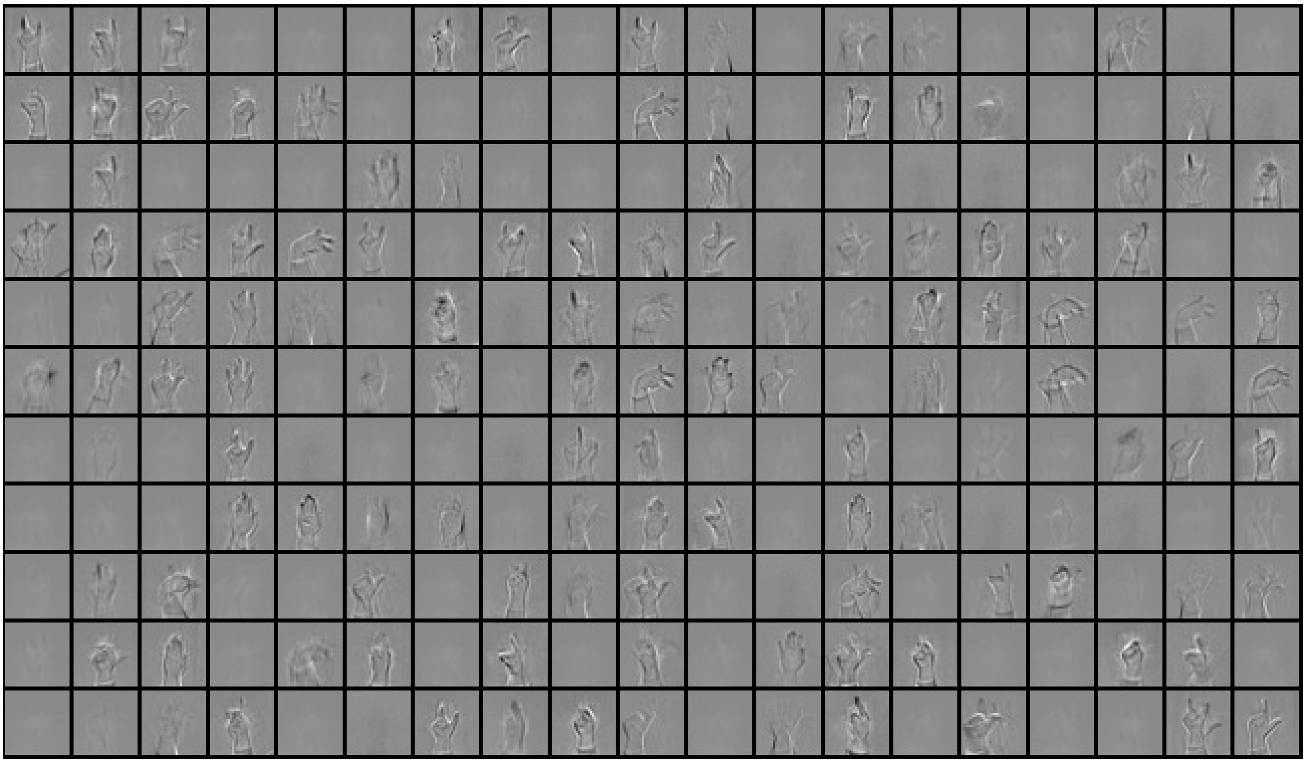

Figure 4: Learned over-complete natural image bases. Sample of the 1024 features learned by training the first RBM layer on normalized image patches (32x32) sampled randomly from gesture dataset. For this experiment, the training protocol is similar to the one proposed in [40] (300 epochs, a mini-batch size of 200, a learning rate of 0:02, an initial momentum of 0:5, a final momentum of 0:9, a weight decay of 0:0002, a sparsity target of 0:02, and a sparsity cost of 0:02).

V. Conclusions and Future Works

The aim of this paper is therefore to propose to use DBNs coupled with tiny images in a challenging image recognition task, view-based ArASLR. The expected results should demonstrate that an approach based on tiny images followed by a projection onto an appropriate feature space can achieve interesting classification results in an ArASLR task. Our hope is to get comparable results or even to outperform the results obtained in [10], [24] based on more complex techniques. In case of comparable results, this paper is thus offer a simpler alternative to the method recently proposed in [10], [24] based on cue integration and the computation of a confidence criterion in a HMM or a SVM classification approach.

Our future work is to empirically investigate the proposed model to achieve Arabic sign language alphabet recognition. The first step is to code the initial dataset using the extracted features. Assuming that the non-linear transform operated by DBN improves the linear separability of the data, a simple regression method will be used to perform the classification process. The classification process will also be examined using a sophisticated classification techniques like SVM in order to investigate whether the linear separability is gained by DBN or not.

After investigating the classification results of the system, this research can be extended to investigate the recognition of further deaf sign groups, such as Arabic numbers, basic Arabic words. Also, this system could be developed to be provided as a web service used in the field of conferences and meetings attended by deaf people. Finally, it can be used in intelligent classrooms and intelligent environments for real time translation for sign language.

References Références Referencias

1. N. Pashaloudi and K. G. Margaritis, “Hidden markov model for sign language recognition: A review,” in in Proc. 2nd Hellenic Conf. AI, SETN-2002, Thessaloniki, Greece. IEEE, Apr. 1112, 2002, p. 343354.
2. L. Dipietro, A. M. Sabatini, and P. Dario, “A survey of glove-based systems and their applications,” Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol. 38, no. 4, pp. 461–482, 2008.
3. M. Moni et al., “Hmm based hand gesture recognition: A review on techniques and approaches,” in Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE

International Conference on. IEEE, 2009, pp. 433–437.

4. S. Kausar and M. Y. Javed, "A survey on sign language recognition," in Frontiers of Information Technology (FIT), 2011. IEEE, 2011, pp. 95–98.
5. Riad, H. K. Elmonier, S. Shohieb, A. Asem et al., "Signs world; deeping into the silence world and hearing its signs (state of the art)," arXiv preprint arXiv: 1203.4176, 2012.
6. P. K. Vijay, N. N. Suhas, C. S. Chandrashekhar, and D. K. Dhananjay, "Recent developments in sign language recognition: a review," Int J Adv Comput Eng Commun Technol, vol. 1, pp. 21–26, 2012.
7. M. Mohandes, M. Deriche, and J. Liu, "Image-based and sensor based approaches to arabic sign language recognition," Human-Machine Systems, IEEE Transactions on, vol. 44, no. 4, pp. 551–557, 2014.
8. N. S. M. Salleh, J. Jais, L. Mazalan, R. Ismail, S. Yussof, A. Ahmad, A. Anuar, and D. Mohamad, "Sign language to voice recognition: hand detection techniques for vision-based approach," Current Developments in Technology-Assisted Education, vol. 422, 2006.
9. R. Naoum, H. H. Owaied, and S. Joudeh, "Development of a new Arabic sign language recognition using k-nearest neighbor algorithm," Journal of Emerging Trends in Computing and Information Sciences, vol. 3, no. 8, 2012.
10. M. A. Mohandes, "Recognition of two-handed arabic signs using the cyberglove," Arabian Journal for Science and Engineering, vol. 38, no. 3, pp. 669–677, 2013.
11. SamirElons, M. Abull-ela, and M. F. Tolba, "Pulse-coupled neural network feature generation model for arabic sign language recognition," IET Image Processing, vol. 7, no. 9, pp. 829–836, 2013.
12. M. Mohandes and M. Deriche, "Arabic sign language recognition by decisions fusion using dempster-shafer theory of evidence," in Computing, Communications and IT Applications Conference (ComComAp), 2013. IEEE, 2013, pp. 90–94.
13. K. Assaleh, T. Shanableh, and M. Zourob, "Low complexity classification system for glove-based arabic sign language recognition," in Neural Information Processing. Springer, 2012, pp. 262–268.
14. H. Khaled, S. G. Sayed, E. S. M. Saad, and H. Ali, "Hand gesture recognition using modified 1$ and background subtraction algorithms," Mathematical Problems in Engineering, vol. 2015, 2015.
15. M. Mohandes, "Arabic sign language recognition," in International conference of imaging science, systems, and technology, Las Vegas, Nevada, USA, vol. 1, 2001, pp. 753–9.
16. O. Al-Jarrah and A. Halawani, "Recognition of gestures in arabic sign language using neuro-fuzzy systems," Artificial Intelligence, vol. 133, no. 1, pp. 117–138, 2001.
17. M. Al-Rousan and M. Hussain, "Automatic recognition of arabic sign language finger spelling," International Journal of Computers and Their Applications, vol. 8, pp. 80–88, 2001.
18. O. Al-Jarrah and F. A. Al-Omari, "Improving gesture recognition in the arabic sign language using texture analysis," Applied Artificial Intelligence, vol. 21, no. 1, pp. 11–33, 2007.
19. M. Maraqa and R. Abu-Zaiter, "Recognition of arabic sign language (arsl) using recurrent neural networks," in Applications of Digital Information and Web Technologies, 2008. ICADIWT 2008. First International Conference on the. IEEE, 2008, pp. 478–481.
20. M. Maraqa, F. Al-Zboun, M. Dhyabat, and R. A. Zitar, "Recognition of Arabic sign language (arsl) using recurrent neural networks," 2012.
21. E. E. Hemayed and A. S. Hassanien, "Edge-based recognizer for Arabic sign language alphabet (ars2v-arabic sign to voice)," in Computer Engineering Conference (ICENCO), 2010 International. IEEE, 2010, pp. 121–127.
22. S. Elons, M. Abull-ela, and M. Tolba, "Neutralizing lighting nonhomogeneity and background size in pcnn image signature for Arabic sign language recognition," Neural Computing and Applications, vol. 22, no. 1, pp. 47–53, 2013.
23. SamirElons, M. Abull-ela, and M. F. Tolba, "Pulse-coupled neural network feature generation model for arabic sign language recognition," IET Image Processing, vol. 7, no. 9, pp. 829–836, 2013.
24. Z. A. Mahmoud, M. H. Alaa, A. E.-R. S. Sameh, and M. S. Elsayed, "Arabic alphabet and numbers sign language recognition," International Journal of Advanced Computer Science and Applications (ijacsa), vol. 6, no. 3, 2015.
25. M. Mohandes, S. Aliyu, and M. Deriche, "Arabic sign language recognition using the leap motion controller," in Industrial Electronics (ISIE), 2014 IEEE 23rd International Symposium on. IEEE, 2014, pp. 960–965.
26. Olshausen and D. J. Field, "Sparse coding of sensory inputs," Current opinion in neurobiology, vol. 14, no. 4, pp. 481–487, 2004.
27. J. Field, "What is the goal of sensory coding?" Neural computation, vol. 6, no. 4, pp. 559–601, 1994.
28. 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, 2007. CVPR'07. IEEE Conference on. IEEE, 2007, pp. 1–8.

29. J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009, pp. 1794–1801.
30. J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan, "Sparse representation for computer vision and pattern recognition," Proceedings of the IEEE, vol. 98, no. 6, pp. 1031–1044, 2010.
31. Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, 2010, pp. 2559–2566.
32. Hasasneh, E. Frenoux, and P. Tarroux, "Semantic place recognition based on deep belief networks and tiny images." in ICINCO (2). SciTePress, 2012, pp. 236–241.
33. K. Labusch and T. Martinetz, "Learning sparse codes for image reconstruction." in ESANN, 2010.
34. G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," Neural computation, vol. 18, no. 7, pp. 1527–1554, 2006.
35. P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," in Parallel Distributed Processing: Volume 1: Foundations, D. E. Rumelhart, J. L. McClelland et al., Eds. Cambridge: MIT Press, 1987, pp. 194–281.
36. G. E. Hinton, "Training products of experts by minimizing contrastive divergence," Neural computation, vol. 14, no. 8, pp. 1771–1800, 2002.
37. A. Torralba, R. Fergus, and Y. Weiss, "Small codes and large image databases for recognition," in Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. IEEE, 2008, pp. 1–8.
38. (2007) The arabic dictionary of gestures for the deaf. [Online].Available: http://www.menasy.com/arb%20Dicti onary%20for% 20the%20deaf%20 2.pdf
39. Krizhevsky, G. E. Hinton et al., "Factored 3-way restricted Boltzmann machines for modeling natural images," in International Conference on Artificial Intelligence and Statistics, 2010, pp. 621–628.
40. H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in Proceedings of the 26th Annual International Conference on Machine Learning. ACM, 2009, pp. 609–616.
41. M. Norouzi, M. Ranjbar, and G. Mori, "Stacks of convolutional restricted boltzmann machines for shift-invariant feature learning," in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009, pp. 2735–2742.
42. F. Attneave, "Some informational aspects of visual perception." Psychological review, vol. 61, no. 3, p. 183, 1954.
43. H. Barlow, "Redundancy reduction revisited," Network: computation in neural systems, vol. 12, no. 3, pp. 241–253, 2001.
44. Coates, A. Y. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in International conference on artificial intelligence and statistics, 2011, pp. 215–223.
45. G. E. Hinton, A. Krizhevsky, and S. D. Wang, "Transforming autoencoders," in Artificial Neural Networks and Machine Learning–ICANN 2011. Springer, 2011, pp. 44–51.
46. G. E. Hinton, "Deep belief networks," Scholarpedia, vol. 4, no. 5, p. 5947, 2009.
47. Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.
48. G. Hinton, "A practical guide to training restricted boltzmann machines," Momentum, vol. 9, no. 1, p. 926, 2010.

This page is intentionally left blank