International Journal on Advanced Science Engineering Information Technology

Vol.7 (2017) No. 6
ISSN: 2088-5334

Convolutional Neural Networks and Deep Belief Networks for Analysing Imbalanced Class Issue in Handwritten Dataset

A'inur A'fifah Amri[#], Amelia Ritahani Ismail[#1], and Abdullah Ahmad Zarir[#]

[#]*Department of Computer Science, Kulliyyah of Information and Communication Technology, International Islamic University Malaysia, P.O. Box 10, 50728 Kuala Lumpur, Malaysia.*

E-mail: [1]amelia@iium.edu.my

***Abstract*— Imbalanced class is one of the trials in classifying materials of big data. Data disparity produces a biased output of a model regardless how recent the technology is. However, deep learning algorithms such as convolutional neural networks and deep belief networks have proven to provide promising results in many research domains, especially in image processing as well as time series forecasting, intrusion detection, and classification. Therefore, this paper will investigate the effect of imbalanced data discrepancy of classes in MNIST handwritten dataset using convolutional neural networks and deep belief networks. Based on the experiment conducted, the results show that although the algorithm is suitable for multiple domains and have shown stability, the imbalanced distribution of data still able to affect the overall performance of the models.**

***Keywords*— Convolutional neural network; deep belief network; imbalanced class.**

I. INTRODUCTION

Imbalanced class in a dataset occurs when the data instances are not depicted evenly among the parameters or classes. The majority class of the dataset is when the class has the most instances. The class with the least data instances is called the minority class. Therefore, when performing classification tasks with an imbalanced dataset can cause an overfitting. Overfitting is a result of accuracy bias due to overwhelming data values in one class compared to absent values of another class. The model might return a high accuracy result, but the majority class also influences the output.

The approach that will be focused on this paper is a review on the effects of imbalanced class in a handwritten data set towards deep learning algorithms. Deep learning is an example of machine learning collection that is recently introduced to solve complex, high-level abstract and heterogeneous data sets, especially image and audio data. There are several types of deep learning designs, namely deep neural network (DNN), deep belief network (DBN), recurrent neural network (RNN), convolutional neural network (CNN) and convolutional deep belief networks (CDBN). This paper will focus on two deep learning algorithms, which are CNN and DBN. CNN is organized in one or more convolutional layers with fully connected layers at the end of it. CNNs are used in computer vision and acoustic modeling for automatic speech recognition (ASR). DBN is a representation that is probabilistic and generative. It consists of several layers of hidden units. It is made up of layers of basic learning modules of Restricted Boltzmann Machine (RBM).

This paper is arranged in the following order. Section 2 presents the definitions of imbalanced data, the effects of imbalanced data has for classification tasks and the implementation of any deep learning algorithms used to counter this problem. Section 3 reviews the basic concepts and utilizations of CNN and DBN algorithms respectively. Section 4 explains the experimental setup of data imbalance classification using CNN and DBN and elucidates the preliminary result. Conclusions are described in Section 5.

Encouraging results have been received upon the application of deep learning algorithms in text recognition [1], audio classification [2] and even abstract high-level domains such as emotional recognition [3]. However, these are applied for data that are distributed evenly. Not many imbalanced data problems have been solved using a deep learning method.

According to some papers [4-7], imbalanced class in a dataset refers to the inequality of data dispensation between the classes. The class that has the most training values is termed 'majority class' and the class that has the least or most missing data values are called the 'minority class' [5].

Minority data class is a realistic problem experienced in data mining because of most of the time; data are scarce, despite its importance. The examples of minority classes in real-world problem are credit fraud detection [8] and cancer anomaly diagnosis [6,8]. It can be expensive if the new data needs labelling [9]. Unfortunately, most algorithms devised shown stable and promising performance when using balanced data in classification exercises but showed otherwise when imbalanced data is used [4]. Minority class prediction is presumed to achieve an expensive level of error as compared to classes with many instances and its testing instances are often wrongly classified as well [10].

The imbalanced class could cause poor classification prototypes [6, 7]. The algorithm that performs on a balanced dataset will not perform as good when using an imbalanced dataset [4], regardless how stable the algorithm model is. In an experiment, an imbalanced multimedia data set was used as input for the CNN model [5]. The paper compared the result with a balanced multimedia dataset. Even though the balanced dataset managed to steadily decrease its error rate, the analysis finds that the error rate achieved when an uneven dataset was used, the error rate was unsteady and kept oscillating. In a research paper [6], the author investigates the effect of uneven data allocation towards the performance of SVM. The review described that data disparity generate in a "high false negative rate". Another paper [11] modified kNN algorithms to counter the effect of imbalanced data to the algorithm. Bootstrapping is often used to improve the algorithm performance when imbalanced data is used [6, 9].

A convolutional neural network *(*CNN) consists of one or more convolutional layers [4, 5, 14], alternating with subsampling layers and by the end of the network, optionally, a fully connected MLP [4]. Basically, CNN architecture must consist of convolutional, pooling and fully connected [5]. The convolutional layers are responsible for feature extraction and is called feature map [4, 5, 14] and sometimes feature detection [15]. After convolutional layer, it is often paired up with a pooling layer that will perform a pooling function based on the inputs it received from the previous convolutional layer [4-7]. The pooling layer, also known as a subsampling layer, will alternate with a convolutional layer because it computes the statistics of the convolutional layer. The pooling layer or "downsampling layer" has a few layer alternatives such as min-pooling and max-pooling depending on the problem-solving context. As the name suggests, the pooling function will decrease the input pixels it received from its preceding convolutional layer [5] and carry on until the end of the network. At the end of the series of alternation, a fully connected MLP will be added and works as a classification module for the network [4]. This layer will receive all neurons from its previous layers whether they are convolutional or pooling and connect them with its own neurons [5].

However, the implementation of convolutional and subsampling layers in a CNN plus the method of the network training differs in every CNN [16]. It's all depends on the context of problems that are attempted to solve.

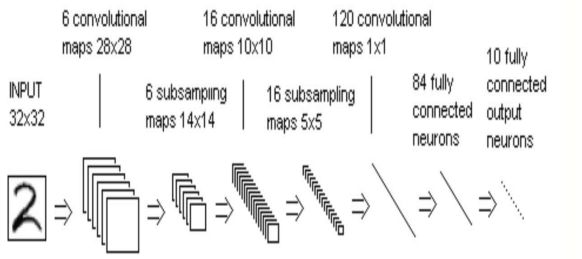

Fig. 1. An example of a simple CNN architecture

Figure 1 shows an example of CNN network architecture. An input of a handwritten digit is given to the network, and it will proceed to the convolutional layers. In this example, there are 6 convolutional maps that use 28 x 28 pixels. Then, the aggregated statistics from the feature maps will pass to the 6 pooling layers or subsampling layers. The subsampling layers will calculate the lower resolution presentation of the previous convolutional layer [14]. Next, it will pass down to 16 convolutional maps and the pixels kept lowering down when it continue to another 16 subsampling maps and 120 convolutional maps. At the end of the network, the neurons will be connected to a fully connected MLP, which provides classification as its output.

To understand DBN, the concept of Restricted Boltzmann Machine (RBM) must first be explained. The architecture of RBM is it consists of a bidirectional connection between hidden layers and visible layers. This feature allows the weight to be connected exclusively and allows deeper extraction between the neurons. RBM is a probabilistic model [2] and a bipartite, undirected graphical structure. It has an ensemble of double-barreled hidden random units h of dimension K, and a group of (binary or real-valued) visible random variables v of dimension D. A weight matrix ($W \in RD \times k$) illustrates the symmetric links between these two layers [17]. Two main RBM often used are Bernoulli, where visible and hidden layers are binary, and Gaussian is where the visible units are allowed to use real number values [3].

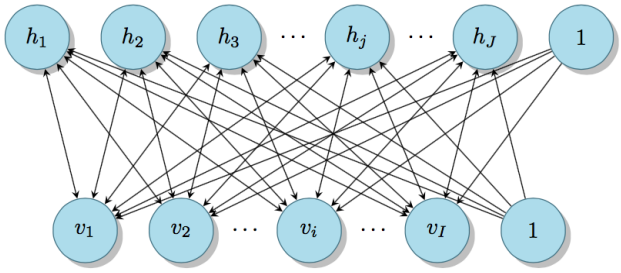

Fig. 2. Example of RBM architecture schematic design [17]

Figure 2 above presents the schematic design of RBM architecture. RBM is made up of stochastic visible units and stochastic hidden units that are connected to each other [13].

A deep belief network (DBN) is likelihood and prompt representation consists of several layers of hidden units composed of basic learning modules. DBN is made up of heaped Restricted Boltzmann Machine (RBM) used greedily as shown in Figure 3. However, such feature results in DBN to be computationally expensive and time-consuming

because the number of layers DBN needs to go through is a lot.

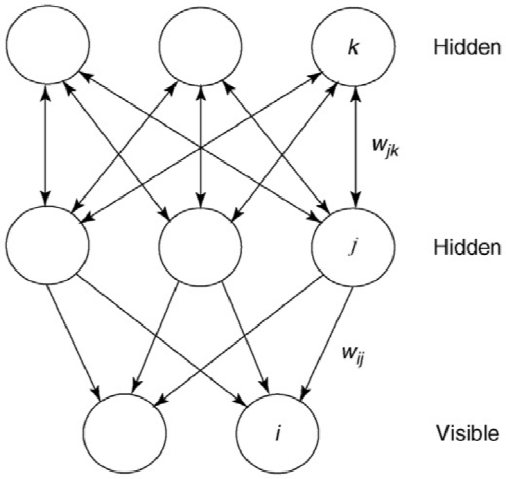

Fig. 3. A stacked RBM or known as DBN [18]

According to Le & Provost [3], training a DBN is not cost effective in terms of computation because pre-training took 11 minutes per epoch and fine-tuning takes up 10 minutes per epoch. DBN is performed in emotion recognition domain [3] by learning high-level features. Face verification domain can also be tackled using DBN, despite the usage of CNN, the hybrid algorithm aims to achieve robustness in verifying similarities of different faces [19]. DBN is also used to replicate natural images [20] by learning multiple layers of unlabeled data.

II. MATERIAL AND ALGORITHM

For this experiment, an imbalanced data set that is suitable for classification task is selected. Then, the source codes of CNN and DBN are modified to suit the data set. The data set will be executed by both source codes. Then, the preliminary results of both CNN and DBN are recorded and further evaluated. Many of the experiments used MNIST handwritten data set as a benchmark [1, 16, 17].

The experimental data set used in this experiment is MNIST handwritten digit data set. It is downloaded from the website [21]. The dataset is preprocessed and consists of 4 files, 2 training files, and 2 testing files.

The training set consists of 60000 examples. The test set has 10000 examples. Since the objective of this paper is to review data disparity and algorithms' performance, the data has been modified to a smaller size but imbalanced. The labels' values are 0 to 9. Pixels are organized row wise. The values are between 0 and 255. 0 signifies background (white), while 255 denotes foreground (black). The images were centred in a 28x28 image. Data distribution is described in Table 1 below together with their percentages.

TABLE I
DATA DISTRIBUTION OF MNIST DATASET

Labels	Number of data	Imbalance Percentage (%)
0	500	100
1	45	9
2	150	30
3	250	50
4	150	30
5	35	7
6	25	5
7	200	40
8	350	70
9	15	3

The algorithms for both CNN and DBN are detailed out in Algorithm 1 and Algorithm 2. This algorithm is based on the previous discussions on CNN and DBN.

Algorithm 1 Convolutional Neural Network

```
trainSize ⇐ [500, 1000, 1500]
testSize ⇐ 200
maxEpoch ⇐ 5000
        while N < maxEpoch do
                if n < trainSize + 1 then
                        trainData ⇐ TrainingError
                        trainLabels ⇐ TrainingError
                        return Training Error
                else
                        testData ⇐ OutputData
                        testLabels ⇐ OutputData
                end if
end while
return OutputData
```

Algorithm 2 Deep Belief Network

```
trainSize ⇐ [500, 1000, 1500]
testSize ⇐ 200
maxEpoch ⇐ 5000
unsupervisedLearningRate ⇐ 0.01
supervisedLearningRate ⇐ 0.05
momentumMax ⇐ 0.95
while N < maxEpoch do
        if n < trainSize + 1 then
              net ⇐ db.DBN()
             return TrainingError
        else
            testData ⇐ OutputData
            testLabels ⇐ OutputData
            return OutputData
        end if
end while
```

III. RESULTS AND DISCUSSION

This section presents the result of the training error of the algorithms, CNN and DBN using the imbalanced dataset as an input. The maximum epochs for all the networks are 5000. However, the training size for the neurons is varied from 500, 1000 and 1500. The training error is captured throughout the experiments for analysis.

As presented in figure 4, CNN is stuck at a local maximum. It is unable to learn from imbalanced data. The maximum error of CNN when training size is 500 is 72.3853

and the minimum error is 3.0986. In Figure 5, DBN training error increases and then kept decreasing and converging. It remained constant at its local minimum by epoch 2000. The maximum error of DBN when training size is 500 is 2.1131 and the minimum error is 1.9186. This shows that DBN is able to learn and predict for imbalanced data.

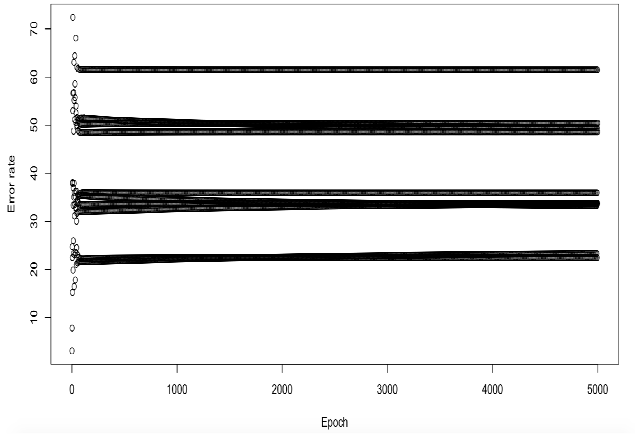

Fig. 4. CNN training error when training size is 500

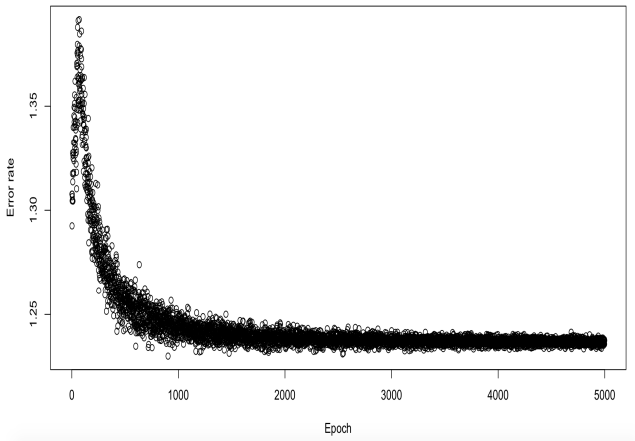

Fig. 5. DBN training error when training size is 500

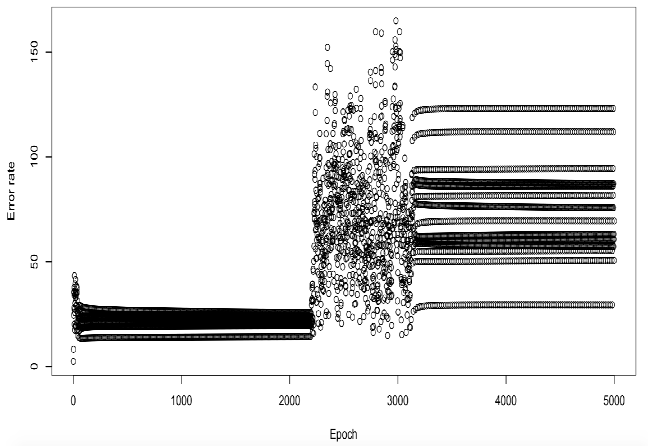

Fig. 6. CNN training error when training size is 1000

In figure 6, the error rate in CNN is stuck in a local minimum, but after epoch 2000 the error rate scattered. After epoch 3000, the error rate stuck at the local maximum. This shows that CNN does not learn from imbalanced data at all. The maximum error of CNN when training size is 1000 is 164.8928 and the minimum error is 2.5014. In figure 7, the error rate in DBN decreases and begins to converge between epoch 1000 and 2000. The error rate decreases faster compared to when the training size was 500.This portrays the ability of DBN to learn from imbalanced data. The maximum error of DBN when training size is 1000 is 1.3914 and the minimum error is 1.23.

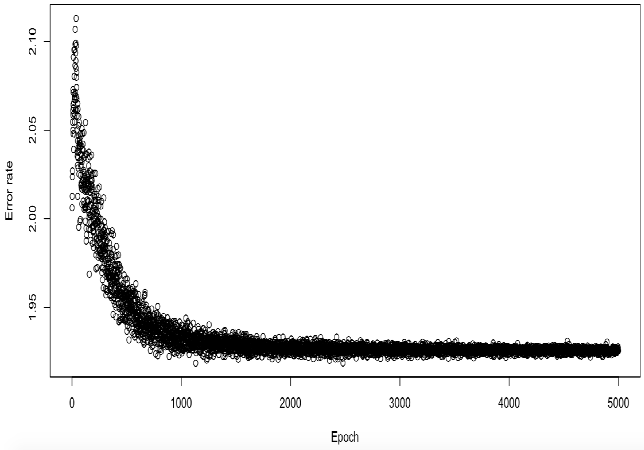

Fig. 7. DBN training error when training size is 1000

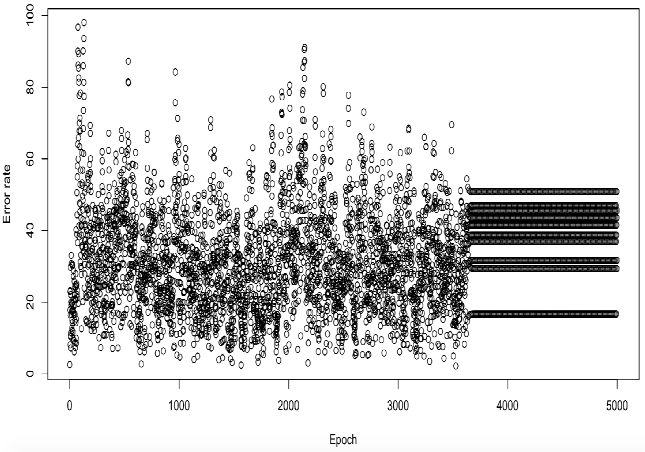

Fig. 8. CNN training error when training size is 1500

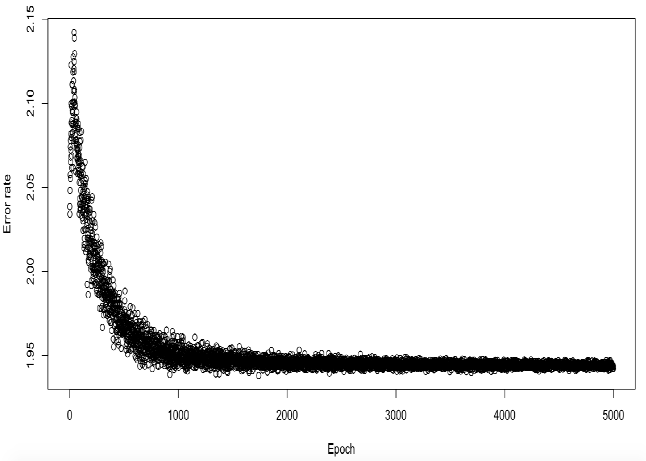

Fig. 9. DBN training error when training size is 1500

In figure 8, the error rate of CNN is scattered until it reaches between epoch 3000 and 4000. The error rate converges and stuck at the local maximum until the final epoch. CNN is still unable to learn to predict from imbalanced data despite the different training size. The maximum error of CNN when training size is 1500 is 98.05599 and the minimum error is 2.2024. In figure 9, the

error rate of DBN begins to converge after epoch 1000 and became constant until epoch 5000. The maximum error of DBN when training size is 1500 is 2.1421 and the minimum error is 1.9382. This shows that the larger the training size, the error merged much faster.

CNN's error range is bigger compared to the error range of DBN. The error scale of DBN is from 1 to 2.5, whereas the error scale of CNN is from 2 to 165. This shows that CNN has more error compared to DBN. High level of error shows a low level of predictive accuracy.

The accuracy rate of CNN for all training set is 0.1. It is constant despite the varying training size. The accuracy rate of DBN for all training set is 20 out of 200, which is 0.1. It is also constant for all despite the different training size. To further analyze the performance of CNN, we further explore the convergence rate of the algorithm with balance dataset. Therefore, for each label, there will be 500 instances. From the experiment conducted, it seems that even though with balanced datasets, CNN is unable to converge and stuck at a local minimum. The outcomes are presented in Figure 10 and Figure 11.

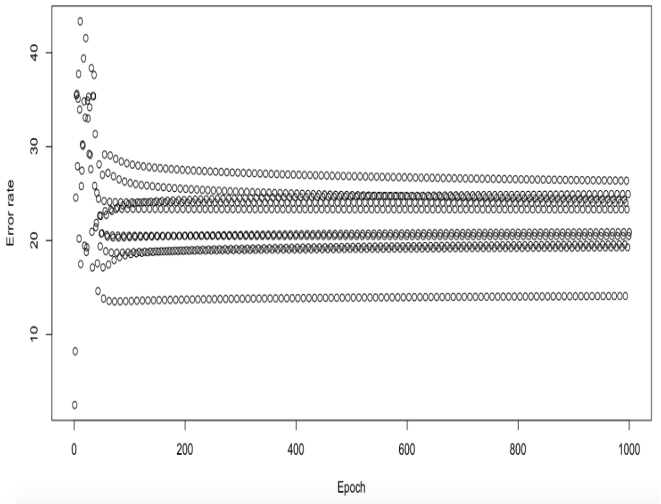

Fig.10. CNN training error when training size is 1000

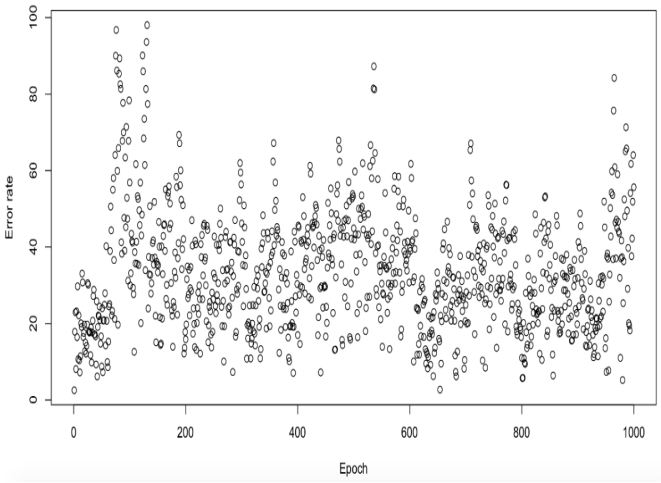

Fig.11. CNN training error when training size is 1500

IV. CONCLUSIONS

As a conclusion, it seems that DBN can learn from imbalanced data set but a lengthy processing time to eventually converge. A usual method to reduce the effects of imbalanced dataset is usually to modify or manipulate the data itself whether by oversampling or under-sampling. However, in the study of algorithm or model modification to minimize the data skew due to data imbalance is yet to be explored thoroughly. Future work shall include the methods to improve models when imbalanced data is used.

ACKNOWLEDGEMENT

This research is supported by the International Islamic University Malaysia under the Research Initiative Grants Scheme (RIGS): RIGS16-346-0510

REFERENCES

[1] Wang, Tao, Wu, David J, Coates, Adam, and Ng, Andrew Y. End-to-end text recognition with convolutional neural networks. In ICPR, pp. 3304–3308. IEEE, 2012.

[2] S. Dieleman, P. Brakel, B. Schrauwen, Audio-based Music Classification with a Pretrained Convolutional Network. International Society for Music Information Retrieval Conference (ISMIR), 669–674 (2011).

[3] D. Le, E. M. Provost, in 2013 IEEE Workshop on Automatic Speech Recognition and Understanding (IEEE, 2013; http://ieeexplore.ieee.org/document/6707732/), pp. 216–221.

[4] P. Hensman, D. Masko, The Impact of Imbalanced Training Data for Convolutional Neural Networks. PhD (2015) (available at https://www.kth.se/social/files/588617ebf2765401cfcc478c/PHensmanDMasko_dkand15.pdf).

[5] Yan, Y., Chen, M., Shyu, M.-L. & Chen, S.-C. Deep Learning for Imbalanced Multimedia Data Classification. in 2015 IEEE International Symposium on Multimedia (ISM) 483–488 (IEEE, 2015). doi:10.1109/ISM.2015.126

[6] Y. Liu, X. Yu, J. X. Huang, A. An, Combining integrated sampling with SVM ensembles for learning from imbalanced datasets. Information Processing and Management. 47, 617–631 (2011).

[7] Fernández, A., García, S. & Herrera, F. Addressing the classification with imbalanced data: Open problems and new challenges on class distribution. in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 6678 LNAI, 1–10 (2011).

[8] N. V. Chawla, N. Japkowicz, A. Kolcz, Editorial : Special Issue on Learning from Imbalanced Data Sets. ACM SIGKDD Explorations Newsletter. 6, 1–6 (2004).

[9] J. Berry, I. Fasel, L. Fadiga, D. Archangeli, Training Deep Nets with Imbalanced and Unlabeled Data. Proc. Interspeech, 1756–1759 (2012).

[10] G. Weiss, F. Provost, The effect of class distribution on classifier learning: an empirical study. Rutgers Univ (2001) (available at ftp://ftp.cs.rutgers.edu/http/cs/cs/pub/technical-reports/work/ml-tr-44.pdf).

[11] W. Liu, S. Chawla, in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (2011), vol. 6635 LNAI, pp. 345–356.

[12] V. Nair, G. Hinton, 3D Object Recognition with Deep Belief Nets. Advances in Neural Information Processing Systems, 1–9 (2009).

[13] A. Mohamed, D. Yu, L. Deng, Investigation of full-sequence training of deep belief networks for speech recognition. Interspeech, 2846–2849 (2010).

[14] O. Abdel-Hamid, L. Deng, D. Yu, in Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH (International Speech and Communication Association, 2013), pp. 3366–3370.

[15] M. Matsugu, K. Mori, Y. Mitari, Y. Kaneda, Subject independent facial expression recognition with robust face detection using a convolutional neural network. Neural networks : the official journal of the International Neural Network Society. 16, 555–9 (2003).

[16] D. C. Ciresan, U. Meier, J. Masci, Flexible, high performance convolutional neural networks for image classification. International Joint Conference on Artificial Intelligence, 1237–1242 (2011).

[17] N. Lopes, B. Ribeiro, J. Gonçalves, "Restricted Boltzmann machines and deep belief networks on multi-core processors", the 2012 International Joint Conference on Neural Networks, pp. 1-7, 2012.

[18] G. E. Hinton, Learning multiple layers of representation. Trends in Cognitive Sciences. 11 (2007), pp. 428–434.
[19] Sun, Y., Wang, X. & Tang, X. Hybrid deep learning for face verification. in ICCV 8828, 1489–1496 (2013).
[20] M. A. Ranzato, G. E. Hinton, Factored 3-Way Restricted Boltzmann Machines For Modeling Natural Images. Artificial Intelligence. 9, 621–628 (2010).
[21] Y. LeCun, C. Cortes, MNIST handwritten digit database. AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist (2010).