On Line Emotion Detection Using Retrainable Deep Neural Networks

Dimitrios Kollias, Athanasios Tagaris and Andreas Stafylopatis
School of Electrical and Computer Engineering
National Technical University of Athens
Zografou Campus 15780, Athens, Greece
andreas@cs.ntua.gr

***Abstract*—This paper presents a new methodology for detecting deterioration in performance of deep neural networks when applied to on line visual analysis problems and enabling fine-tuning, or retraining, of the network to the current data characteristics. Pre-trained deep neural networks which have a satisfactory performance on the problem under study constitute the basis of the approach, with efficient transfer learning being performed whenever drift is detected in network operation. The method is applied and validated on the problem of emotion detection using line facial expression analysis based on a dimensional emotion representation.**

Keywords—deep neural networks, transfer learning, retraining, drift detection, emotion recognition, on line facial expression analysis

I. INTRODUCTION

Dealing with neural network learning and generalization has been a major topic of research for the last thirty years [1], [2]. [3], [4], [5], [6]. Recently, big interest has been paid to deep learning techniques and neural networks [7-9] in various application areas, including speech analysis [10-13], natural language processing [14] and image analysis and recognition [15-19]. Currently deep architectures constitute the state-of-the-art in all those disciplines and multiple efforts have managed to improve the values of existing performance measures. Deep Convolutional Neural Networks (DCNNs), Deep Belief Networks (DBNs) and Recurrent Neural Networks (RNN) are specific architectures that either standalone, or in combination, are appropriately trained to generate rich internal representations and produce high performances in all these application domains.

Of equal importance has been the area of machine learning in non-stationary and changing environments. Detection of the drift in such cases is of great importance for machines that have the ability to adapt to changes in their environment, through fine tuning, or retraining to the new circumstances. Fine tuning, or re-training should be implemented, by retaining the former knowledge of the neural network, while focusing on the new information gathered from the current environment. Developing active, or passive approaches which are able to detect drift and adapt the network architecture in slowly varying environmental conditions has also attracted much research interest in the past [20], [21]. Some other approaches focus on drift detection and neural network adaptation. A variety of passive, or active methodologies have been developed for detecting drift, e.g., by using Gaussian modelling and searching for changes in obtained models' mean and standard deviation. These techniques have been applied in many multimedia analysis applications.

In this paper, we investigate the problem of on-line emotion detection based on facial expression analysis. Recognition of user emotional states (categorical, dimensional, facial action unit based) in-the-wild, can be based on various input modalities (facial expressions, facial landmarks, hand and body gestures, speech and audio, physiological cues). It constitutes a significant problem, that has recently been the topic of many research Challenges (such as EmotiW 2013-2016 [22], Av+EC 2014-2016 [23], FERA2015 [24]), Conference Workshops and Special Sessions (CVPR2015-2016, ICCV2015, ECCV2016).

The state-of-the-art in emotion detection is currently based on using pre-trained deep neural networks which are adapted to new environments, where only a small amount of training data is available, through transfer learning, without overfitting. Transfer learning is achieved by using deep networks - previously trained with large image datasets (even of generic objects) - and by performing fine-tuning, i.e., retraining, of all or of parts of the networks using the provided small dataset. In the above-mentioned Challenges, the winner methods have used networks pre-trained with the FER-2013 and the Toronto Face Dataset [25], [26], [27]. In [27], two transfer learning problems were tested: fine-tuning the general ImageNet CNN [19] to the FER-2013 one, and fine-tuning the FER-2103 to a new small dataset (SFEW). Moreover, in [25] and [27], CNNs are pre-trained with the above mentioned databases and fine-tuning is performed by extending the training of these CNNs using the SFEW training dataset and validating its performance with the SFEW validation dataset. In [26] the parameters of all convolutional layers of the pre-trained DCNN are frozen and only the parameters of the fully connected layers are updated during fine-tuning using the SFEW datasets.

A new method is proposed in this paper for detecting drift and retraining deep neural networks for emotion detection when handling time-varying visual data with facial

expressions. This approach includes a drift detection part, informing the network that a fine-tuning or re-training procedure has to be performed. The updating of the network weights is then implemented using a variant of the gradient projection method, which uses both former and current information.

The paper is composed of the following Sections. Description of the adaptation procedure is presented in Section II, while the deep neural network adaptation algorithm and the retraining decision mechanism are presented in Sections III and IV, respectively. The Emotion Detection problem and the experimental set up are described in Section V. Application of the proposed approach for emotion detection is described in Section VI. The forthcoming work and the conclusions are described in Section VII.

II. Problem formulation

The problem we are dealing with can be described as follows: Let a deep neural network learn to classify a training set, say S_b, consisting of input and corresponding desired output vectors. Let $y(i)$ be the output of the network when it receives a new data sample at its input, $i = 1,2,...$, not included in the network training data set. Whenever this data refers to a change in the environment, or context of classification, a fine-tuning, or network adaptation procedure should be executed. Assuming that w_b contains the network weights before adaptation and w_a the network weights after adaptation has been performed.

A new training set S_t is obtained from the new environment, normally including a small number of data, e.g. equal to the batch size of deep learning algorithms. In interactive applications, especially in human computer interaction, the user is assumed to be asked and provide the correct classification for this set of data. The new network weights w_a are then computed by minimizing the following error criterion:

$$E_a = E_{t,a} + nE_{f,a} \tag{1}$$

where $E_{t,a}$ denotes the error performed over training set S_t, i.e. over current knowledge and $E_{f,a}$ is the corresponding error performed over training set S_b, i.e. over former deep neural network knowledge. Parameter *n* is a weighting factor representing the relative significance of the current training set with respect to the former one.

In fact, the initially trained deep neural network classifier is supposed to provide a coarse approximation of the correct classification, regardless of the changes of the environment, or of the context of interaction. Possible misclassifications produced by this classifier are corrected by the retrained network, which yields the classification of the current operational context.

The aspects of network retraining, which are analyzed next, are the specific retraining algorithm, as well as the mechanism for deciding activation of retraining, or equivalently, for detecting drift in the operational, user, or context model. The latter is based on performance analysis of the initially trained network applied on the current data.

III. Deep neural network retraining

As already mentioned in the introduction, the current state-of-the-art in transfer learning between deep convolutional neural networks either performs fine-tuning of the pre-trained networks, using data from the new environment, or freezes (mainly the convolutional) part of the network and retrains (mainly the fully connected) rest of it.

In our approach we minimize (1) by assuming that a small perturbation of network weights (all, or the weights of the fully connected part) is enough to achieve good classification performance in the current conditions. Consequently

$$w_a = w_b + \Delta w \tag{2}$$

with Δw being a small weight increment. This assumption permits linearization of the nonlinear activation neuron function, using a first-order Taylor series expansion.

Moreover, to stress the importance of current data in the minimization of (1), we replace the term $E_{t,a}$ on its right hand side by the constraint that the actual network outputs, after retraining, $z_a(i)$, are equal to the desired ones, i.e.,

$$z_a(i) = d(i), \text{ for all data } i \text{ in } S_t \tag{3}$$

It can be shown [29] that, through linearization, the solution of (3) with respect to weight increments Δw is equivalent to the solution of a set of linear equations:

$$c = A\Delta w \tag{4}$$

with matrix *A* being computed in terms of previously trained weights, while the elements of vector *c* are defined as follows:

$$c(i) = d(i) - z_b(i), \text{ for all data } i \text{ in } S_t \tag{5}$$

and $z_b(i)$ denotes the outputs of the initially trained network, when this is applied to the data in S_t.

The size of vector *c* is smaller than the number of unknown weights Δw, thus many solutions exist for (4). To ensure uniqueness, however, we adopt the additional requirement that the solution causing a minimal degradation of the previous network knowledge should be selected. This is equivalent to minimizing the absolute difference of the errors produced by the networks with current and previous weights over data in S_b. It has been shown [30] that this difference takes the form

$$E_S = (\Delta w)^T K^T K (\Delta w) \tag{6}$$

where the elements of matrix K are expressed in terms of the previous network weights w_b and the training data in S_b.

Thus, the retraining problem is reduced to minimization of (6) subject to constraints (3) and the constraint for small weight increments. A variety of methods can be used for this minimization. We have adopted the gradient projection method, which starts from a feasible point and moves in a direction which decreases E_S and satisfies the above constraints [30].

IV. Drift Detection and Network Retraining

Detection of drift in the statistical, or cognitive characteristics of a monitored, or analyzed event, will permit the neural network to respectively change its structure, or its interconnection weight values, so as to better fit the new context, or the new user interaction, or the new environmental conditions.

It would be valuable, if the network generated some cues, based on which it would be possible to detect when its performance deteriorated, i.e., its output would be much different from the desired one. However, as can be seen in (5), vector c expresses the difference between the desired and the actual network outputs based on w_b. Consequently, if the norm of vector c increases, network performance deviates from the desired one and retraining should be activated.

Let us assume that the N th retraining phase of the network classifier has been completed. If the classifier is then applied to the current data, it is expected to provide classification results of good quality. The difference $c(0)$ between the output of the retrained network and that produced by the initially pre-trained network at the first batch of data used for retraining constitutes an estimate of the level of improvement achieved by the retraining procedure:

$$c(0) = \sum_{batchsize} (y(0) - z_t(0))^2 \quad (7)$$

Let $c(k)$ denote the respective difference at the k-th instance of retraining data:

$$c(k) = \sum_{batchsize} (y(k) - z_t(k))^2 \quad (8)$$

It is anticipated that the level of improvement expressed by $c(k)$ will be close to that of $c(0)$ as long as the classification results are satisfactory. If they start differing considerably, then this comes in general from a change in the user or scene characteristics. It is, therefore, the quantity

$$a(k) = | c(k) - c(0) | \quad (9)$$

which is computed and compared to a threshold, expressing the maximum tolerance, to detect a drift and activate the network retraining procedure.

V. The On-Line Emotion Detection Problem

A. Dimensional Emotion Analysis

The 2-D valence-arousal space provides an emotion representation which has recently attracted much interest in the field of emotion recognition. Valence refers to whether the person expresses some positive, or negative feeling on its surroundings, whether they refer to other people, objects, or scenes. Arousal refers to the interest that a person shows towards a specific scene, or interaction, in which he or she participates.

This 2-D valence – arousal space is illustrated in Fig. 1 which reflects two studies, by Whissell [31] and by Plutchik [32]. The space can be represented by a circle on a computer screen, split into four quadrants by the two main axes. The vertical axis represents Activation, running from very active to very passive and the horizontal axis represents evaluation, running from very positive to very negative. It reflects the popular view that emotional space is roughly circular. The center of the circle marks a sort of neutral default state.

Fig. 1. The valence – arousal space

B. The On-Line Emotion Detection Dataset

In the following we adopt this representation for the on-line emotion detection problem. Our main focus is on analyzing facial expressions in video, detecting the underlying emotion of the person appearing in the consecutive scenes.

As has been already said, many researchers have recently investigated the ability of deep neural networks to analyze user behavior, especially in-the-wild, i.e. in uncontrolled environments, for recognition of emotions based on user facial expressions and other modalities. It is, however, known that analyzing emotion based on facial expressions is very much human dependent, in the sense that different humans express their emotions through different ways. Consequently, small or large adaptation of a deep neural network trained to extract

human behavior is necessary when testing its performance in a video in which different individuals appear.

Two different databases have been used in the paper for testing respective neural networks: the FER2013 database and the naturalistic database generated by Queen's University of Belfast (QUB) in the EC FP5 IST ERMIS project [28] and further extended into the EC FP6 IST HUMAINE Network of Excellence and EC FP7 SEMAINE Network. Characteristic frames of these databases are shown in Fig. 2a-c.

Fig. 2a. Frames of the FER Database

Fig. 2b. Frames of a specific user of the QUB Database

Fig. 2c. Frames of the ERMIS Database

In the FER dataset case, we generated a dimensional labeling, by using the already categorically labeled frames and the 2-D representation shown in Fig. 1

In the case of the QUB/ERMIS datasets, it should be added that they have been created by engaging participants to emotional dialogue, so facial expressions in these video sequences are not acted and extreme, but are mostly naturalistic.

Two experts have annotated the experimental image sequences by evaluating both facial movements and emotional states of the user in two separate phases. During the first phase, they annotated the activated Action Units (AUs) [33], after taking into account their relation with the movements of facial features. During the second phase, they evaluated the emotional state (positive or negative) of the user throughout the video, thus, generating an initial, visual, annotation with respect to the 2D emotion representation.

The selected frames were the most 'facially expressive' (with large facial movements and a high confidence level of estimation), which were then extended to a larger dataset, by an AU similarity approach based on the results of our existing facial feature extraction system [34].

Since our target is on-line emotion detection, we selected scenes from the above labelled datasets and created composite video, including sequential appearance of scenes from the different datasets. It is on these video that we performed our experimental study on network retraining.

C. The deep neural network architecture

A Deep Convolutional Neural Network (DCNN) was the basis for analysis of incoming images of individuals from the above databases in this work.

We adopted the architecture used in [35] and shown in Fig. 3, since the specific emotion recognition approach used transfer learning from pre-trained networks, based on the "AlexNet" ImageNet reference model [19], fine-tuning them with appearance images and with extracted facial geometrical features from the FER database. The networks included convolutional and pooling layers. The last two stages are fully connected feedforward layers, with the last stage consisting of softmax units, which output the probabilities for each network output. The activation function used in the inner network layer is the rectified linear unit (ReLU) function. The network has 5 outputs corresponding to the four quadrants of the 2-D emotional space shown in Fig. 1 and the (0,0) neutral state.

In the following, we formed the composite video, including scenes from the above databases and we investigated the ability of the proposed approach to detect the drift when moving from one scene to the next and use the deep convolutional neural network to perform on-line emotion detection based on analysis of the facial expressions of the acting persons.

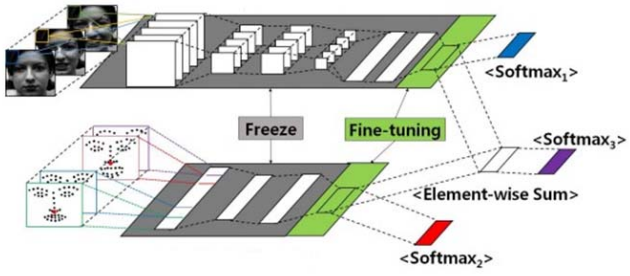

Fig. 3. Fine-tuning of pre-trained deep networks using facial appearance and geometrical images

It should be mentioned that it is possible to use a Deep Recurrent Neural Network [36] instead of the DCNN in the proposed approach, without changing the presented methodology.

The composite video includes frames of standard PAL format, which are first processed through a face detection algorithm and the resulting facial images are cropped and presented as inputs to the DCNN, with the corresponding label.

VI. Experimental Study

The ability of the methodology proposed in this paper for on-line retraining of deep neural networks for emotion detection has been tested on video sequences which contain scenes from the three different datasets presented in the former Section.

First we have pre-trained a deep convolutional neural network, as the one shown in Fig. 3 based on the FER database. The training procedure for the DCNNs closely followed that of [35], using stochastic gradient descent with hyperparameters (momentum=0.9, weight decay=0.0005, initial learning rate=0.001, dropped by a factor of 10 following every 10 epochs of training).

We have constructed a long video sequence including consecutive scenes from the above datasets. The first 2000 frames came from the FER dataset, so the pre-trained DCNN (let us call it *Net0*) had a very good performance on them. We needed, however, a change detection mechanism to indicate when a new scene started. In particular the following scene includes facial expression activity from the QUB dataset, as shown in Fig. 2b. In a real case scenario, the user shown in Fig. 2b would like to use the pre-trained network (*Net0*) in his own environment.

Detecting the drift between the different parts of the video can be made using techniques which perform change detection [20-21], by testing the statistical distribution of data. For efficiency, we can use aggregate motion vectors between consecutive frames for motion estimation. In this way we can detect big changes in the pixel distributions of each frame, therefore detecting consecutive scenes coming from different datasets. Fig. 4a shows such a case, where the aggregated motion vector activity is different, between a scene taken from the QUB dataset and another scene taken from the ERMIS dataset.

However, this methodology cannot be used to detect changes in the same scene. Fig. 4b shows the motion vector activity in the QUB video scene, which includes a variety of facial expressions and underlying emotions, as the ones shown in Fig. 2b.

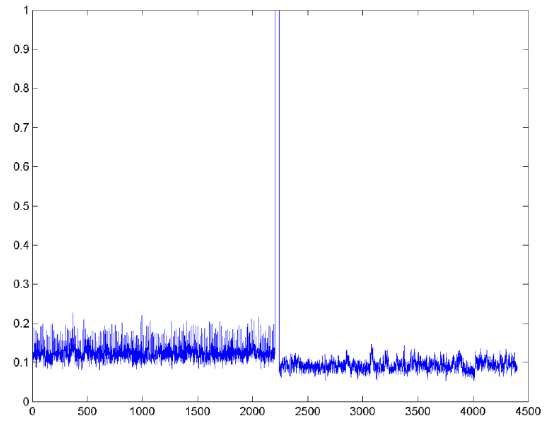

Fig. 4a. Motion activity in and between different scenes

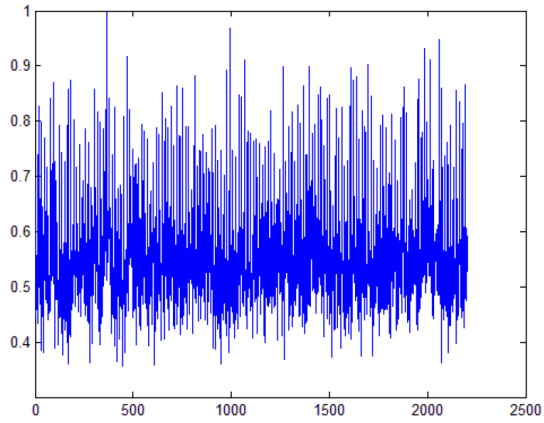

Fig. 4b. Motion activity in the QUB scene

By detecting the change in environment of use, *Net0* should be re-trained using data from the new user's environment and respective labels provided, in an interactive way, by the user, or the environment.

Let us assume, that, in this case, we have the time and ability to interact with the user and collect mini-batches (10 frames each) with many different facial expressions of him/her. Then we can retrain *Net0* using the method presented in Section III. Implementing this, a new network, say, *Net1*, results, which we apply to the following frames of the video for emotion detection. We were able to get an increase in performance, compared to that of *Net0,* on this QUB scene frames, of 20%.

In the following, we examine the on-line emotion detection problem, where it is possible to get a response by the user, regarding his or her emotional state, shown in only one (the currently processed) frame or mini-batch of frames. To illustrate this case, let us assume that another scene, with another person, as the one in Fig. 2c, follows the two former (FER and QUB) scenes.

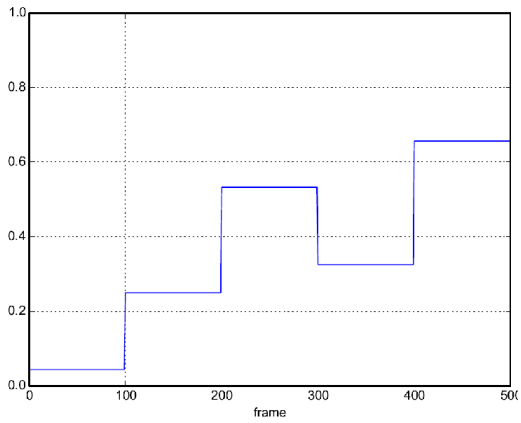

Fig. 5a. Criterion (9) for retraining no 1 (frame 100)

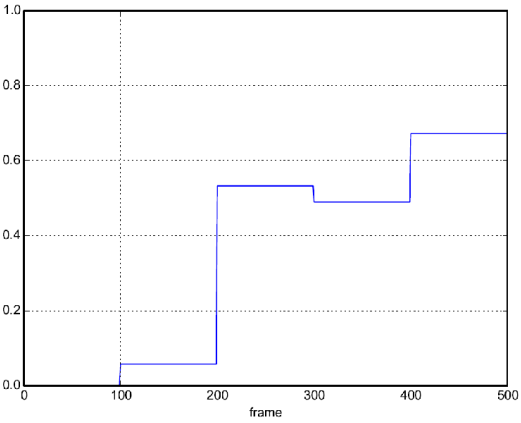

Fig. 5b. Criterion (9) for retraining no 2 (frame 200)

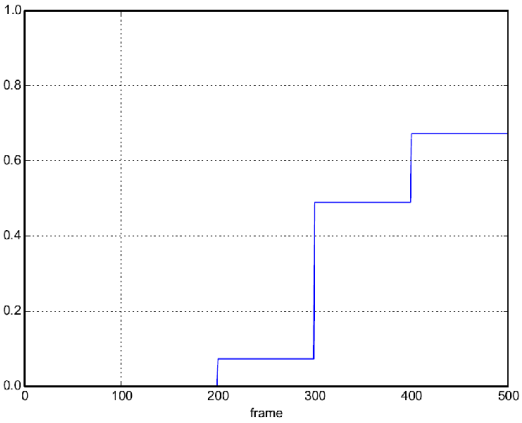

Fig. 5c. Criterion (9) for retraining no 3 (frame 300)

It should be reminded that we already have two networks detecting emotion in the video frames, *Net0* and *Net1*. As mentioned earlier in the paper, we expect that *Net0* will have a satisfactory performance on this new data; the performance of *Net1*, which was trained to match specifically the QUB scene data, is expected to deteriorate. Thus, criterion (9), based on (7)-(8) was used to detect the need for retraining.

Fig. 5a shows the normalized average value of (9) in 500 frames of the video used in our experiments. The first 100 frames belong to the QUB scene, to which *Net0* and *Net1* are applied, while the following 400 frames belong to the ERMIS dataset, showing, in sequence, expressions of 4 different emotional states of the user, as the ones shown in Fig. 2c. As expected in these 100 frames, (9) provides a small value, while in the following frames, its value increases, indicating the need for retraining the network.

Following detection of this change in (9), we collected the first mini-batch of frames (i.e., frames 101-110) with their labels and retrained *Net1*, as described in Section III. We have chosen to retrain *Net1* and not *Net0* in this case, because we know that scenes which resemble already processed ones will appear later in the generated video. The resulting network, say *Net2*, is then applied to the rest of the video, while (9) is used to compare its performance to *Net0*. Fig. 5b shows the value of criterion (9) between the performances of these two networks.

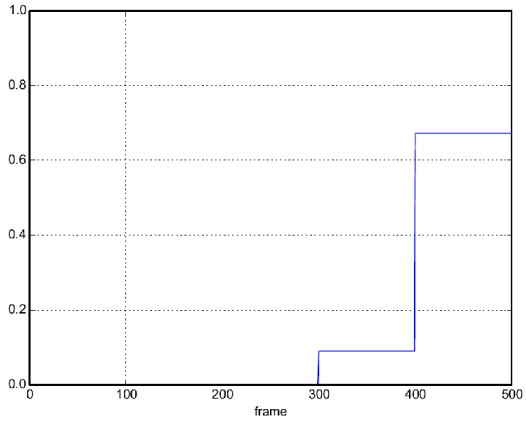

Fig. 6a. Criterion (9) for retraining no 4 (frame 400)

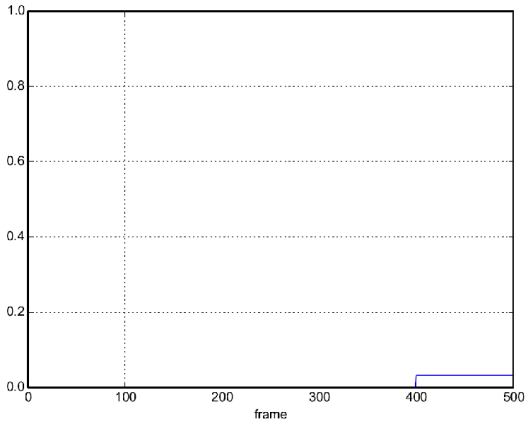

Fig. 6b. Criterion (9): no need to retrain until frame 500

It can be seen that (9) provides a small value until frame 200, where a new expression of the user in Fig. 2c is included, to which *Net2* has not been trained. Consequently, the need for retraining is indicated by the criterion (9) and a new retraining phase is performed, collecting, again, the first mini-batch of frames (i.e., frames 201-210). The procedure is continued with the resulting retrained network, *Net3*.

Two more changes are detected in frames 300 and 400, corresponding to two new expressions of the user, resulting in two more network retraining phases and generation of two more networks, *Net4* and *Net5*. Fig. 6a and 6b show the values of (9) when they are respectively applied to the data.

VII. Conclusions

This paper presents a new methodology for retraining of deep neural networks when detecting emotion in video, using a mechanism for drift detection and a retraining algorithm. The approach has been applied to the problem of on-line emotion recognition based on facial expression analysis. Forthcoming work includes application of the methodology to real-life human computer interaction scenarios, especially in interactive applications, where user behavior analysis plays a very important role.

References

[1] T.-Y. Kwok and D.-Y. Yeung, "Constructive algorithms for structure learning in feedforward neural networks for regression problems", IEEE Trans. Neural Networks, vol. 8, pp. 630–645, 1997.

[2] R. Reed, "Pruning algorithms—A survey", IEEE Trans. Neural Networks, vol. 4, pp. 740–747, 1993

[3] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth, "Learnability and the Vapnik–Chervornenkis dimension", J. Assoc. Computing Machinery, vol. 36, pp. 929–965, 1989.

[4] B. Cheng and D. M. Titterington, "Neural networks: A review from a statistical perspective", Statist. Sci., vol. 9, pp. 2–54, 1994

[5] S. Haykin, Neural Networks: A Comprehensive Foundation. NewYork: Macmillan, 1994.

[6] V. N. Vapnik, The Nature of Statistical Learning Theory. New York: Springer-Verlag, 1995.

[7] Y. Bengio, Y, "Learning Deep Architectures for AI", Foundations and Trends in Machine Learning, vol 2, no. 1, pp. 1–127, 2009.

[8] G.E. Hinton, S. Osindero, Y.W. Teh, "A Fast Learning Algorithm for Deep Belief Nets", Neural Computation, vol. 18, no. 7, pp. 1527–1554, 2006.

[9] Y. LeCun, K. Kavukcuoglu, C. Farabet, "Convolutional Networks and Applications in Vision", Proc. ISCAS, pp. 253–256, Paris, France, 30 May-2 June 2010.

[10] N. Morgan, "Deep and wide: Multiple layers in automatic speech recognition", Audio, Speech, and Language Processing, IEEE Transactions on, vol. 20, no. 1, pp. 7–13, 2012.

[11] A. Mohamed, G.E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks". Audio, Speech, and Language Processing, IEEE Transactions on, vol. 20, no. 1, pp. 14–22, 2012.

[12] G. Sivaram and H. Hermansky, "Sparse multilayer perceptron for phoneme recognition", Audio, Speech, and Language Processing, IEEE Transactions on, vol. 20, no. 1, pp. 23–29, 2012.

[13] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Unsupervised learning of hierarchical representations with convolutional deep belief networks", Communications of the ACM, vol. 54, no. 10, pp. 95–103, 2011.

[14] M. Wollmer, M. Kaiser, F. Eyben, B. Schuller, G. Rigoll, "Lstm-modeling of continuous emotions in an audiovisual affect recognition framework", Image and Vision Computing vol. 31, no. 2, pp. 153–163, 2013

[15] H. Lee, C. Ekanadham, and A. Ng, "Sparse deep belief net model for visual area v2", Advances in neural information processing systems, vol. 20, pp. 873–880, 2008.

[16] Y. Tang and C. Eliasmith, "Deep networks for robust visual recognition", in International Conference on Machine Learning. Haifa, Israel, 21-24 June 2010.

[17] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Unsupervised learning of hierarchical representations with convolutional deep belief networks", Communications of the ACM, vol. 54, no. 10, pp. 95–103, 2011.

[18] K. Sohn, D.Y. Jung, H. Lee, and A.O. Hero, "Efficient learning of sparse, distributed, convolutional feature representations for object recognition", in Computer Vision (ICCV), 2011 IEEE International Conference on CV, pp. 2643–2650, 2011.

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

[20] G. Ditzler, M. Roveri, C. Alippi, R. Polikar, "Learning in Mon-Stationary Environments: A Survey", IEEE Computational Intelligence Magazine, pp. 12-25, November 2011.

[21] C. Alippi, Intelligence for Embedded Systems. New York: Springer Verlag, 2014.

[22] A. Dhall, O. Ramana Murthy, R. Goecke, J. Joshi, T. Gedeon, " Video and image based emotion recognition challenges in the wild: Emotiw 2015", Proceedings of the 2015 ACM International Conference on Multimodal Interaction. pp. 423–426, 2015.

[23] F. Ringeval, B. Schuller, B., M. Valstar, S. Jaiswal, E. Marchi, D. Lalanne, R. Cowie, M. Pantic, "Av+ ec 2015: The first affect recognition challenge bridging across audio, video, and physiological data", Proceedings of the 5th International Workshop on Audio/Visual Emotion Challenge. pp. 3–8. ACM, 2015.

[24] M.F. Valstar, T. Almaev, J.M. Girard, G. McKeown, M. Mehu, M., L. Yin, M. Pantic, J.F. Cohn, "2015-second facial expression recognition and analysis challenge", Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on. vol. 6, pp. 1–8, 2015.

[25] B.K. Kim, H. Lee, J. Roh, S.Y. Lee, "Hierarchical committee of deep CNNS with exponentiallyweighted decision fusion for static facial expression recognition", Proceedings of the 2015 ACM International Conference on Multimodal Interaction. pp. 427–434, 2015

[26] Z. Yu, C. Zhang, "Image based static facial expression recognition with multiple deep network learning", Proceedings of the 2015 ACM International Conference on Multimodal Interaction, pp. 435–442, 2015.

[27] H.W. Ng, V.D. Nguyen, V. Vonikakis, S. Winkler, "Deep learning for emotion recognition on small datasets using transfer learning", Proceedings of the 2015 ACM International Conference on Multimodal Interaction. pp. 443–449, 2015.

[28] S. Ioannou, A. Raouzaiou, V. Tzouvaras, T. Mailis, K. Karpouzis, S. Kollias, "Emotion recognition through facial expression analysis based on a neurofuzzy network", Special Issue on Emotion: Understanding & Recognition, Neural Networks, Elsevier, vol. 18, no. 4, pp. 423-435, May 2005.

[29] A. Doulamis, N. Doulamis, S. Kollias, "On-Line Retrainable Neural Networks: Improving the Performance of Neural Networks in Image Analysis Problems", IEEE Trans. Neural Networks, vol. 11, no.1, pp. 137-156, January 2000.

[30] D. C. Park, M. A. EL-Sharkawi, and R. J. Marks II, "An adaptively trained neural network", IEEE Trans. Neural Networks, vol. 2, pp.334–345, 1991.

[31] C. Whissel, The dictionary of affect in language, emotion: Theory, research and experience. R. Plutchik and H. Kellerman, Eds., New York: Academic, 1989.

[32] R. Plutchik, Emotion: A psychoevolutionary synthesis. Harpercollins College Division, 1980.

[33] P. Ekman, W.V. Friesen, Facial action coding system, 1977.

[34] S. Asteriadis, P. Tzouveli, K. Karpouzis, S. Kollias, "Non-verbal feedback on user interest based on gaze direction and head pose", 2nd International Workshop on Semantic Media Adaptation and Personalization, London, United Kingdom, 17-18 December 2007.

[35] H. Jung, S. Lee, J. Yim, S. Park, J. Kim, "Joint fine-tuning in deep neural networks for facial expression recognition", Proceedings of the IEEE International Conference on Computer Vision. pp. 2983–2991, 2015.

[36] S. Ebrahimi Kahou, V. Michalski, K. Konda, R. Memisevic, C. Pal, " Recurrent neural networks for emotion recognition in video", Proceedings of the 2015 ACM International Conference on Multimodal Interaction. pp. 467–474, 2015.