This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Prototype-Incorporated Emotional Neural Network

Oyebade K. Oyedotun, *Member, IEEE*, and Adnan Khashman, *Senior Member, IEEE*

***Abstract*— Artificial neural networks (ANNs) aim to simulate the biological neural activities. Interestingly, many "engineering" prospects in ANN have relied on motivations from cognition and psychology studies. So far, two important learning theories that have been subject of active research are the prototype and adaptive learning theories. The learning rules employed for ANNs can be related to adaptive learning theory, where several examples of the different classes in a task are supplied to the network for adjusting internal parameters. Conversely, the prototype-learning theory uses prototypes (representative examples); usually, one prototype per class of the different classes contained in the task. These prototypes are supplied for systematic matching with new examples so that class association can be achieved. In this paper, we propose and implement a novel neural network algorithm based on modifying the emotional neural network (EmNN) model to unify the prototype- and adaptive-learning theories. We refer to our new model as "prototype-incorporated EmNN". Furthermore, we apply the proposed model to two real-life challenging tasks, namely, static hand-gesture recognition and face recognition, and compare the result to those obtained using the popular back-propagation neural network (BPNN), emotional BPNN (EmNN), deep networks, an exemplar classification model, and k-nearest neighbor.**

***Index Terms*— Emotional neural network (EmNN), face recognition, hand-gesture recognition, neural network, prototype learning.**

I. Introduction

MACHINE intelligence is a field that aims to achieve various tasks such as face recognition [1], speaker identification [2], natural language processing [3], and document segmentation [4] based on motivations from the human cognition processing [5], [6]. Inasmuch as these tasks are somewhat "trivial" for humans, machines strive to perform competitively [7], [8]. It is the hope that we can grossly simulate machines with "thinking" or processing capabilities such that performance on the aforementioned tasks can be achieved with reasonably high accuracy. More important is that machines can boast of intelligence when such systems have the capability to learn (or adapt internal parameters) and upgrade its performance over time based on available experiential knowledge [9], [10]; this is analogous to learning in humans [11], [12]. However, for us to breakthrough in machine intelligence and vision, we must first understand the basis of learning and visual processing in humans [13], [14]. Unfortunately, there exist a number of different schools of thought on how (object recognition) learning is achieved in humans, with two considerably important and actively researched theories of learning in humans being the prototype and adaptive-learning theories [15], [16]. The proposed model in this paper draws engineering inspiration from both learning theories to realize improved learning. We give a sufficient discussion on the two theories which give insight into the remaining sections within this work.

Manuscript received March 21, 2016; revised July 7, 2016, December 24, 2016, January 25, 2017, and April 7, 2017; accepted July 10, 2017. *(Corresponding author: Oyebade K. Oyedotun.)*

O. K. Oyedotun is with the Interdisciplinary Center for Security, Reliability and Trust (SnT), University of Luxembourg, L-1855 Luxembourg, Luxembourg, and also with the European Center of Research and Academic Affairs (ECRAA), 99010 Lefkosa, Turkey (e-mail: oyebade.oyedotun@uni.lu).

A. Khashman is with Final International University, 99370 Girne, Turkey, and also with ECRAA, 99010 Lefkosa, Turkey (e-mail: adnan.khashman@ecraa.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2017.2730179

A. Prototype-Learning Theory

The prototype-learning theory suggests that learning is achieved using the prototypes (representative examples) of different objects [17], [18]. Generally, for a particular object, one or very few number of prototypes is (or are) stored in the memory. When tasked with identifying a new object, the memory is scanned for the prototype that matches the most the new object; the class of the retrieved prototype is associated with the new object [19], [20]. This approach can be seen as grossly a "store-and-retrieve" memory-based system.

Many research works have described and exploited the concept of prototypes learning to develop novel machine-learning algorithms with motivating results. Zeithamova [21] described a prototype as 'a concise representation for an entire group (category) of entities, providing means to anticipate hidden properties and interact with novel stimuli based on their similarity to prototypical members of their group'. Chang *et al.* [22] in their work described, implemented, and applied adaptive prototype-learning systems to machine-learning problems. In the same work, various criteria such as generalized condensed nearest neighbor, k-means (KM) clustering, and fuzzy c-means clustering were used to select prototypes from the training data, which constituted a subset of the training data. It was noted that using the extracted prototypes rather than the whole training data significantly sped up training. At testing time, the categories of new samples were evaluated based on the already collected prototypes. The idea behind the learning algorithm is to select the smallest number of prototypes which give the optimal representation for the different classes contained in the training data; obtained results in this work were shown to be highly competitive with some other famous learning algorithms such as the conventional k-nearest neighbor (k-NN) and support vector machine. In another related work, a two-stage-based generalized prototype framework was described; data dimensionality reduction was achieved in the first stage via projection onto a line, while a thresholding stage was used for the discrimination of projected data distribution [23], i.e., class labels.

Although prototype-learning theory is effective in its "pure" form, one of its major flaws is operation under real life constraints such as occlusion and noise (incompleteness); systems based solely on this theory falter significantly [24], [25]. For example, consider the task of recognizing a cat with one missing limb; such an unforeseen situation may largely motivate the system to output a wrong class, considering that the prototype (or very few prototypes) stored do not depict that a cat may have three limbs. A strong point of concern on this theory is that the human ability on objects identification remains relatively strong even under such aforementioned constraints [26]–[28]. Furthermore, it is obvious that attempts to have all unforeseen situations as prototypes is quite infeasible because it is practically impossible to envisage all possible variations in objects of the same class; also, memory constraint is another major setback. Hence, other learning theories aim to account for such high human performances considering the aforementioned constraints.

B. Adaptive-Learning Theory

The adaptive-learning theory suggests that learning is achieved by using several examples of the different objects to be learned to adjust model internal parameters [29], [30]; this is as against the prototype-learning theory. Also, the whole objects may not be stored in the memory (as in the prototype theory); important features which differentiate objects of one category from others are extracted or learned and stored in the memory [31]. New objects are identified using learned features retrieved from the memory [32], [33]; this approach is more robust to real life constraints such as occlusion and noise. Note that adaptive-learning theory is synonymous with connectionism approach and explanation for learning.

Particularly, neural networks are at the center of learning in machines; these networks store experiential knowledge as interconnection weights [34]. They compose massively interconnected artificial neurons, which are stacked as layers. Features from examples are extracted in a phase referred to as training [35]. Generally, a large database of examples is used for training these networks, and networks' performances get better with more training examples. Once a neural network is trained, new examples can be identified by simulating the trained network; identification is achieved using learned features (stored as weights). Hence, it is very possible to still recognize a cat with say one missing limb since other preserved features that infer that the presented object is a cat are used. It will be seen that neural networks considerably implement the adaptive-learning theory [36], [37].

C. Multiprototype Learning Versus Exemplar Learning

In this section, we aim to clarify the delicate similarity between multiprototype and exemplar learning; this should eliminate arousing confusion further into the work. Also, we provide motivational basis for multiprototype learning.

First, it is important to note that many studies consider that prototype and exemplar learning are situated at extreme opposite ends of the learning spectrum [38]. The exemplar-learning theory strictly assumes that all observed examples so far are used for making inference on new examples. A suitable scenario to consider for exemplar learning is k-NN algorithm, where strictly all memorized examples per class are "recruited" for making inference on new examples. It is important to note that there are no reference examples in k-NN, as all available examples per class are "equally" used (or consequential) for performing inference.

Conversely, prototype-learning theory assumes that only one example per class is used for performing inference on new examples. Also, a prototype may be a single abstracted representation or an explicit central tendency example for a class [22], [39]. Going further, several studies perhaps observing that in many situations, inferring the classes of new examples from only one class representative example (prototype) may be misleading, hence reconsider prototype learning such that more than one representative example per class can be used for making inference. In other related works, chorus of prototypes and multiprototypes were proposed for more robust and meaningful learning [40]–[45]. In any situation, one unanimous position among researchers is that only in exemplar learning is it assumed that all available examples are used for inference. One obvious scenario for prototype learning is the KM clustering. Originally, only one prototype per class was considered for performing inference. However, with concerns on robustness issues, new research works have posited the benefit of multiprototypes per class for performing inference [46]–[48]. Hence, it can be considered that for prototype learning, even with the availability of several examples per class, only one or more representative examples per class are used for making inference, with other examples per class being strictly of no consequence for performing inference [22]. Nevertheless, we refrain from over-exploiting the extended prototype learning conception with multiprototypes per class by not proposing a "greedy" model that relies on too many prototypes per class. In this work, we have limited the number of prototypes per class to a maximum of 5, irrespective of the number of available examples per class. However, we leverage on both prototype and adaptive learning for building the proposed model within this research. For example, Schyns [49] described in his work the motivation for such a combined learning scheme.

In this work, we propose prototype-incorporated emotional neural network (PI-EmNN) that is based on the emotional back-propagation (EmBP) learning algorithm [50]; in Section II, we explain that the model in [50], and therefore, our proposed model possesses no human emotions, but artificially simulated emotional signals for improving learning. The contribution of this paper is that we integrate the learning power obtainable from conventional neural network (based on adaptive learning: conventional network weights learning), emotional features (based on artificially simulated processing of global perceptions of presented training input patterns: emotional weights learning), and a novel prototype learning scheme (motivated by prototype learning: prototype weights learning).

In order to demonstrate that improved learning is realized with the novel neural network model described within this work, we apply the developed network model (PI-EmNN)

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

to two important vision-based recognition tasks, static hand-gesture recognition, and face recognition. Although the proposed model within this work should suffice on some other learning tasks, we find vision-based tasks more suited as applications since it is easier to evoke emotional responses based on visual stimuli. For vision-based tasks, the input data which are processed images have structured internal representation; that is, neighboring data (pixel) values have strong local correlation. Generally, vision-based recognition systems use image pixel values for learning; this is quite consistent with human visual processing. This contrast with feature extraction-based systems in which important features (e.g., texture and shape metrics) are first extracted from images, after which a classifier is then trained on such extracted features. It is considered that vision-based recognition systems are more naturally plausible than the feature extraction approach; hence, they readily evoke emotional responses during learning.

The Thomas Moeslund's static hand-gesture database [51] and ORL (AT&T) face database [52] have been used to train and test the proposed model within this work.

II. Proposed Neural Network Model

In this paper, we aim to fuse both prototype- and adaptive-learning theories in a neural network. Alternatively, we consider our approach as incorporating prototype learning in neural networks (since neural networks are traditionally based on adaptive-learning theory).

Here, we advance on an earlier published work which describes a novel neural network model referred to as the emotional neural network (EmNN) [50]. The EmNN has been used successfully with motivating performance in different applications [53]–[55]. The main idea behind the EmNN is such that neural network can simulate two important emotional features in learning such as anxiety and confidence. This is analogous to humans where anxiety is high for newly encountered tasks and confidence quite low (this happens at the start of training); conversely, anxiety decreases for familiar tasks, while confidence increases (this happens as training progresses). We emphasize that machines do not have human physiology and hence cannot feel in the same way humans do. Even after so much progress in machine intelligence and cognitive studies, learning in machine is quite far from learning in humans. Nevertheless, signals flow through machines; therefore, we can artificially simulate emotions in machines just as learning itself. In this context, the EmNN possesses no real (or human) but artificial emotions (anxiety and confidence) which rely on nonprocessing emotional neurons (denoted as M: orange in Fig. 1) which feed the hidden and output layers with the average values (global perceptions) of presented training patterns, referred to as Y_{PAT} [50]. Also, the weight interconnections of the emotional neurons are updated during learning.

In this paper, we further incorporate two additional nonprocessing neurons, P (prototype neuron) and C (correlation neuron), feeding both the hidden and output layers, i.e., shown as gray in Fig. 1. The prototype neuron P supplies the normalized prior prototype class label of the presented input pattern (attributes) to the hidden and output layers of the network. While, the correlation neuron supplies the correlation coefficient of the presented input pattern (attributes) with the selected prototype to the network. One of the major motivations for implementing prototype learning in neural network is such that overall learning can be quickly guided toward a solution space composing good local minima based on important prior knowledge of training data labels incorporated into the network. The weights of conventional neurons in the network are updated using the conventional back-propagation (BP) algorithm (shown without color in Fig. 1); while the weights of added nonprocessing neurons P, C, and M are updated using the EmBP algorithm. The proposed PI-EmNN model is shown in Fig. 1, and a brief discussion on the added neurons P, C, and M is given in the following. Note that the B in Fig. 1 represents the conventional bias neuron.

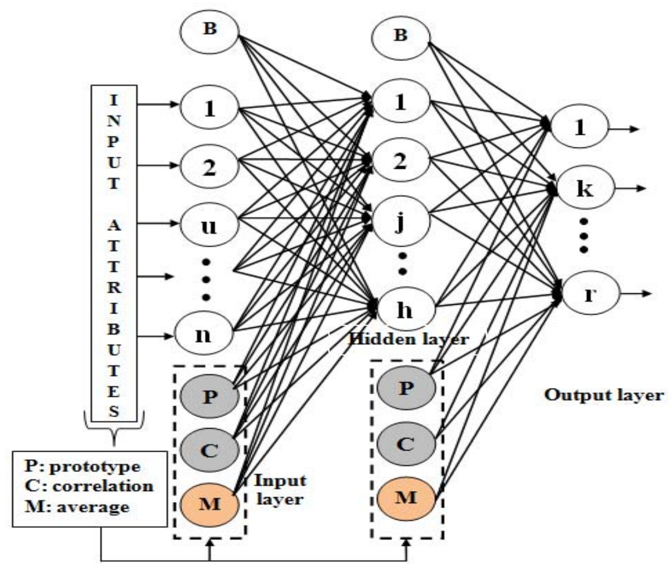

Fig. 1. PI-EmNN.

A. Prototype Neuron (P)

1) One Prototype Per Class Approach: The training data is scanned and one example per class for all the classes in the task is randomly selected to form the prototypes. Hence, for a task composed of r classes, a number of r prototypes are selected; each selected example per class is referred to as the prototype (representative example) of that class. Therefore, a prototype database is built for the task (from the training data). During training, each presented pattern is compared with all the r prototypes (prototype database); a distance metric is used to obtain the closet prototype to the presented input pattern (attributes). The prototype with lowest distance metric is selected as the associated prototype of the presented input pattern; the class of the selected prototype is coded as P. The Euclidean distance has been used in this work as the distance metric for obtaining the associated prototypes and therefore classes of presented input patterns. We assume that features are standardized. The Euclidean distance metric is defined in [56]

$$d = \sqrt{\sum_{u=1}^{n} \left(x_u - x_u^p\right)} \tag{1}$$

where x_u is the input attribute with index u from the presented input pattern, x_u^p is the input attribute with index u from the prototype, and n is the dimensionality of both the presented input patterns and prototypes.

The normalization of the selected class label l is achieved using (2). The normalized class labels of selected prototypes are supplied to the prototype neuron P

$$P = \frac{l}{r} \tag{2}$$

where

$$1 \leq l \leq r. \tag{3}$$

2) Multiprototypes Per Class Approach: In this approach, a number of prototypes per class z are randomly extracted from the training data. Each input pattern presented is compared (using the Euclidean distance metric) with the prototypes of each class; this makes the selection of associated prototype to which the presented input pattern belongs to more robust that is probability of associating the correct prototype and therefore class with the input pattern increases. In this work, we implement a voting system for selecting the associated prototype for presented input patterns; hence, z is chosen to have odd values. Also, we observe that a rough heuristic for determining z is the strength of variations observable in the training examples for each class. Note that for both one prototype and multiprototypes per class approaches, selected prototypes are examples sampled from the whole available training examples with replacement.

B. Correlation Neuron (C)

Here, we obtain the strength of association of a presented input pattern with the selected prototype. The Pearson's correlation coefficient is used to obtain the degree of relation between the selected prototype and the presented input pattern. The Pearson's correlation coefficient value ranges from -1 to $+1$. Also, we consider selected prototypes as independent variables and the input pattern (attributes) as dependent variables. Equation (4) describes Pearson's correlation coefficient R [57], where x_u is the input attribute with index u from the presented input pattern, x_u^p is the input attribute with index u from the prototype, and n is the dimensionality of both the presented input patterns and prototypes

$$R = \frac{\sum_{u=1}^{n} \left((x_u - \bar{x}_u)\left(x_u^p - \bar{x}_u^p\right) \right)}{\sqrt{\sum_{u=1}^{n} (x_u - \bar{x}_u)^2 \sum_{u=1}^{n} \left(x_u^p - \bar{x}_u^p\right)^2}}. \tag{4}$$

Since, neural networks typically accept input values in the range 0 to 1, the correlation coefficients are transformed into the range 0 to 1 and supplied to C (correlation neuron). More important is that we leverage on the transformation requirement to obtain another highly important statistic used to measure the certainty of predictions made from a certain model, which is referred to as the coefficient of determination, R^2 (square of Pearson's correlation coefficient) [58]. In this work, for the sake of compactness and intuition, we denote R^2 with C, as seen in (5) [59]. C expresses the strength of the linear association between the prototype (model) class data and the presented input data [60]. Its values are supplied to the correlation neuron

$$C = R^2. \tag{5}$$

It is noted that another important impact of the correlation neuron is that it supports (reinforces) the evidence of correctly associated prototype class supplied to P and dampens the effect of incorrectly associated prototype class supplied to P. For correctly associated prototype class and where presented input pattern is quite similar to the selected prototype, the coefficient of determination is high, i.e., close to 1. Conversely, where the associated prototype class is incorrect, the correlation coefficient is low, i.e., close to 0.

C. Emotional Neuron (M)

The emotional neurons M supply input averages (global perceptions) of presented input patterns to the model. Each input pattern presented is averaged and supplied to the model using

$$M = Y_{\text{PAT}} = \frac{\sum_{u=1}^{n} x_u}{n}. \tag{6}$$

D. PI-EmNN Activations Computations and Weights Update

From Fig. 1, it is seen that the output of any neuron in the network is contributed by the conventional network input, bias input, prototype neuron input, correlation neuron input, and emotional neuron input, i.e., forward—pass computation. Furthermore, for weights update (back—pass computation), the conventional and bias neurons weights are updated using the conventional BP algorithm; the prototype, correlation, and emotional neuron weights are updated using EmBP algorithm. The inspiration for updating the prototype and correlation neuron weights using the EmBP algorithm is such that the network can also simulate emotional responses on the prototype and correlation neurons weights. The network is constrained to show anxiety and confidence on the prototype and correlation neurons weights as training progresses; note that the anxiety parameter (μ) decreases while the confidence parameter (k) increases. This is quite consistent with learning in humans, where we have low confidence and high anxiety at the beginning of learning an unfamiliar task, but over time (with training) anxiety reduces and confidence increases. In prototype learning, where only 1 or extremely few samples are taken as prototypes for learning, it therefore becomes more important and motivating that the network can show anxiety and confidence based on exposure to vast training data. The prototype and correlation neurons can be seen as also supplying a global representation of the different presented training patterns to the proposed neural network model, analogous to the emotional neuron; albeit, based on a different and novel approach (see Sections II-A–II-C). Furthermore, for situations where prototypes associated with presented input patterns can be sometimes wrong; it then becomes reasonable that the prototype and correlation neurons weights are updated using the EmBP algorithm such that confidence can increase over correctly associated prototypes while anxiety decreases over wrongly associated prototypes as training progresses. In the subsequent sections of this paper, it is shown that the described and novel weights update scheme performs as expected; an overall improved learning experience is observed based on the recognition tasks considered.

The output of any hypothetical hidden neuron j denoted as A_j is computed from

$$A_j = f\left(\sum_{u=1}^{n} w_{\mathrm{ju}} x_u + w_{\mathrm{jb}} b_j + w_{\mathrm{jp}} P + w_{\mathrm{jc}} C + w_{\mathrm{jm}} Y_{\mathrm{PAT}}\right) \quad (7)$$

where x_u is the input attribute indexed by u, w_{ju} is the weight interconnection from input neuron u to hidden neuron j, w_{jb} is the weight interconnection from input bias neuron to hidden neuron j, b_j is the bias of the hidden layer, w_{jp} is the weight interconnection from prototype neuron P to hidden neuron j, w_{jc} is the weight interconnection from correlation neuron C and hidden neuron j, w_{jm} is the weight interconnection from input emotional neuron M to hidden layer neuron j, Y_{PAT} is the global average of the input pattern, and f is the activation function.

The output (activation) of any hypothetical output neuron k denoted A_k is obtained using

$$A_k = f\left(\sum_{j=1}^{h} w_{\mathrm{kj}} A_j + w_{\mathrm{kb}} b_k + w_{\mathrm{kp}} P + w_{\mathrm{kc}} C + w_{\mathrm{km}} Y_{\mathrm{PAT}}\right) \quad (8)$$

where A_j is the output of hidden neuron j, w_{kj} is the weight interconnection from hidden neuron j to output neuron k, w_{kb} is the weight interconnection from hidden bias neuron to output neuron k, b_k is the bias of the output layer, w_{kp} is the weight interconnection from prototype neuron P to output neuron k, w_{kc} is the weight interconnection from correlation neuron C to output neuron k, w_{km} is the weight interconnection from hidden emotional neuron M to output layer neuron k, Y_{PAT} is the average of the input pattern, and f is the activation function

$$E_i = \frac{1}{2}\sum_{p=1}^{N}\sum_{k=1}^{r}(T_k - O_k)^2. \quad (9)$$

Equation (9) describes the mean squared error (MSE) cost function at iteration i, denoted E_i; where, T_k and O_k are the target and actual outputs, respectively; r is the number of output neurons; p and N are the index and total number of training patterns, respectively.

The global average of all presented patterns during training is denoted Y_{AVPAT} and calculated using

$$Y_{\mathrm{AVPAT}} = \frac{1}{N}\sum_{p=1}^{N} Y_{\mathrm{PAT}}. \quad (10)$$

The anxiety (μ_i) and confidence (k_i) coefficients at iteration i are obtained using

$$\mu_i = Y_{\mathrm{AVPAT}} + E_i. \quad (11)$$

$$k_i = \mu_0 - \mu_i. \quad (12)$$

Note that the confidence parameter k is 0 at the start of training and increases as training progresses, while the anxiety u is highest at the start of training and gradually decreases as training progresses.

1) Hidden-Output Layer Weights Update:

1) The weights of the conventional hidden-output layer neurons are updated using

$$w_{\mathrm{kj}}(i+1) = w_{\mathrm{kj}}(i) + \eta \Delta_k A_j + \beta[\delta w_{\mathrm{kj}}(i)] \quad (13)$$

where η is the learning rate, Δ_k is the output layer error signal, A_j is the output of hidden neuron j, β is the momentum rate, δw_{kj} is the previous weight change, and i is the iteration index.

2) The output error signal, Δ_k, is calculated using (14), where function f is taken as log-sigmoid for (7) and (8)

$$\Delta_k = A_k(1 - A_k)(T_k - A_k). \quad (14)$$

3) The weights of hidden-output layer bias neurons are updated using

$$w_{\mathrm{kb}}(i+1) = w_{\mathrm{kb}}(i) + \eta \Delta_k A_b + \beta[\delta w_{\mathrm{kb}}(i)] \quad (15)$$

where A_b is set to 1; δw_{kb} is the previous weight change for the hidden-output bias neuron.

4) The hidden-output layer prototype neuron weights are updated using (16); δw_{kp} is the previous weight change for the hidden-output prototype neuron

$$w_{\mathrm{kp}}(i+1) = w_{\mathrm{kp}}(t) + \mu \Delta_k P + k[\delta w_{\mathrm{kp}}(i)]. \quad (16)$$

5) The weights of the hidden-output layer correlation neuron are updated using

$$w_{\mathrm{kc}}(i+1) = w_{\mathrm{kc}}(t) + \mu . \Delta_k C + k[\delta w_{\mathrm{kc}}(i)] \quad (17)$$

where δw_{kc} is the previous weight change for the hidden-output correlation neuron.

6) The hidden-output layer emotional neuron weights are updated using

$$w_{\mathrm{km}}(i+1) = w_{\mathrm{km}}(t) + \mu \Delta_k Y_{\mathrm{PAT}} + k[\delta w_{\mathrm{km}}(i)] \quad (18)$$

where μ and k are the anxiety and confidence coefficients; δw_{km} is the previous weight change for the hidden-output emotional neuron.

2) Input-Hidden Layer Weights Update:

1) The weights of the conventional input-hidden layer neurons are updated using

$$w_{\mathrm{ju}}(i+1) = w_{\mathrm{ju}}(i) + \eta \Delta_j A_u + \beta[\delta w_{\mathrm{ju}}(i)] \quad (19)$$

where Δ_j is the error signal to the hidden layer, A_u is the input to hidden neuron j, δw_{ju} is the previous weight change, and Δ_j is calculated using

$$\Delta_j = A_j(1 - A_j)\sum_{j=1}^{r} w_{kj} \Delta_k. \quad (20)$$

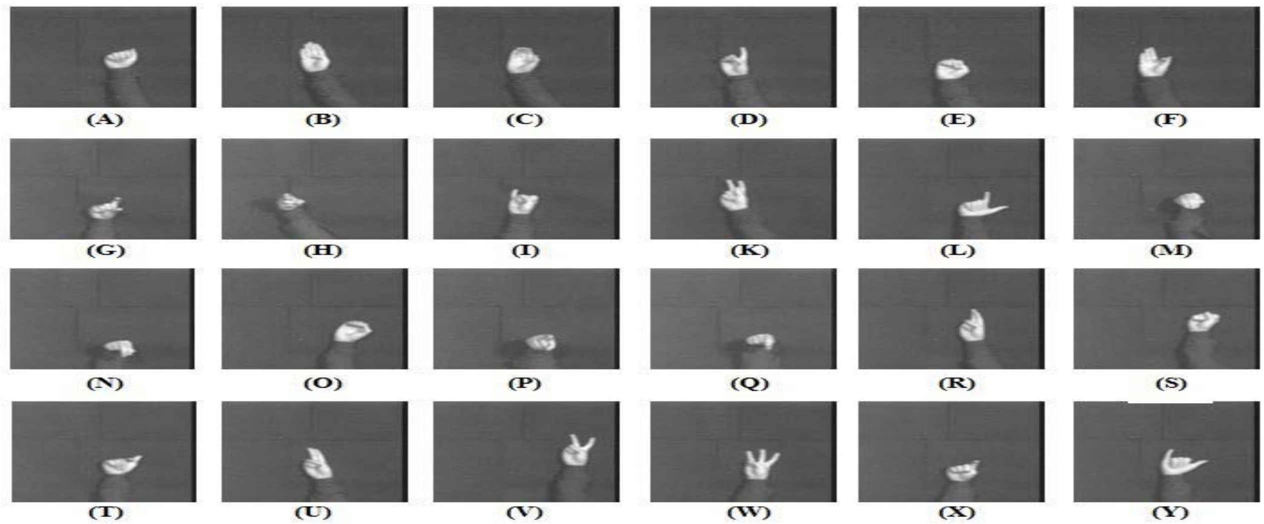

Fig. 2. Samples of the 24 unprocessed static hand-gestures from the Thomas Moeslund's database [51].

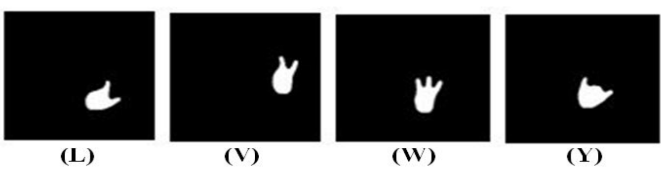

Fig. 3. Samples of preprocessed hand gestures.

2) The weights of the input-hidden layer bias neurons w_{jb} are updated using (21); where δw_{jb} is the previous weight change

$$w_{\mathrm{jb}}(i+1) = w_{\mathrm{jb}}(i) + \eta \Delta_j A_b + \beta[\delta w_{\mathrm{jb}}(i)]. \quad (21)$$

3) The weights of the input-hidden layer prototype neuron are updated using

$$w_{\mathrm{jp}}(i+1) = w_{\mathrm{jp}}(i) + \mu \Delta_j P + k[\delta w_{\mathrm{jp}}(i)]. \quad (22)$$

4) The weights of the input-hidden layer correlation neuron are updated using

$$w_{\mathrm{jc}}(i+1) = w_{\mathrm{jc}}(i) + \mu \Delta_j C + k[\delta w_{\mathrm{jc}}(i)]. \quad (23)$$

5) The weights of the input-hidden layer emotional neuron are updated using

$$w_{\mathrm{jm}}(i+1) = w_{\mathrm{jm}}(i) + \mu \Delta_j Y_{\mathrm{PAT}} + k[\delta w_{\mathrm{jm}}(i)]. \quad (24)$$

One important highlight of (16)–(18) and (22)–(24) used to update the prototype, correlation, and emotional weights is that as the anxiety coefficient reduces and confidence parameter increases, the network is made to pay less attention to the error signal, but more attention to the previous weight changes. This novel weights update scheme for the prototype, correlation, and emotional neurons can be considered an extra inertia to the network during learning, and motivates the network against convergence to poor local minima. Also, note that emotional parameters are self-taught during training.

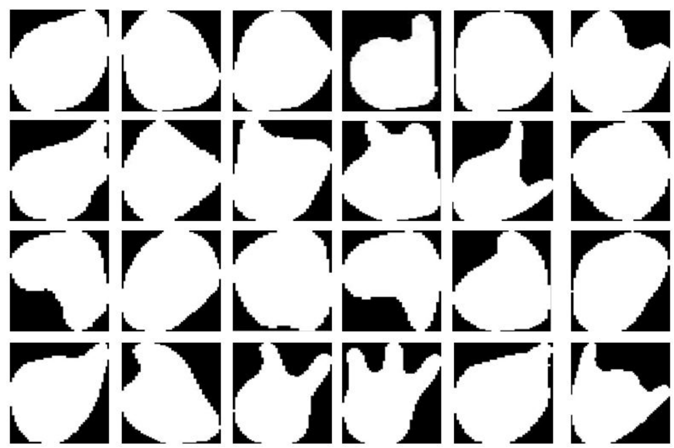

Fig. 4. Samples of the 24 segmented hand gestures.

III. Database Analysis and Processing

A. Static Hand-Gesture Database

The database used in this work is obtained from the Thomas Moeslund's static hand-gesture database [51]. The database contains the 24 static hand-gestures for the American sign language; the two alphabets which are missing are the non-static sign languages ("J" and "Z"). The sample images of unprocessed hand gestures are shown in Fig. 2. It is obvious that for a more reasonable training (reducing redundant information in the training data); the hand gestures in Fig. 2 should be segmented. Critical to the segmentation stage is the image preprocessing stage described in the following.

1) Image Preprocessing: The images are preprocessed by conversion to binary (black and white). The binary conversion is achieved by thresholding the images at 0.5 gray level. Furthermore, the binary images are filtered with a median filter of size 10×15; the filtering denoises the images prior to the segmentation. Some samples of hand gestures converted to binary and with the median filtering applied on them are shown in Fig. 3.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Fig. 5. ORL face database [52].

2) Image Segmentation: The images are segmented by running an algorithm on the preprocessed images; the algorithm extracts the white pixels contained in the images in a form of bounding box. The 24 segmented hand-gesture sample images are shown in Fig. 4.

The static hand-gestures recognition database contains 2040 samples; all images are rescaled to 32×32 pixels (this reduces training computational requirements). In this work, we have taken 1440 samples ($\sim$70%) as the training data and 600 samples ($\sim$30%) as the testing data. The training-to-testing data ratio has been chosen such as not to bias learning, i.e., using too many training samples, but few testing samples.

B. Face Database

The ORL (AT &T) face database is used in this work for training and testing the proposed model. The face database contains frontal poses of 40 different subjects with moderate variations in presentations; this makes recognition more challenging. Also, the same subjects are captured with varying facial expressions, and some subjects have glasses on in some of the images, while the same subjects are without glasses in other images; these variations make the recognition task even further challenging. The database contains ten sample images (in grayscale) per subject; hence, a total of 400 sample images (in grayscale) for the 40 different subjects. The sample images of the 40 different subjects are shown in Fig. 5.

For training the proposed model in this work, five images per subject (200 images for the 40 subjects: 50% of available data) are used, while the remaining five images per subject (200 images for the 40 subjects: remaining 50% of available data) are used for testing the trained models. Furthermore, the images are rescaled to 32×32 pixels, which reduce computational requirements. The training-to-testing data ratio has been chosen such that comparative analysis with an earlier work can be achieved with ease.

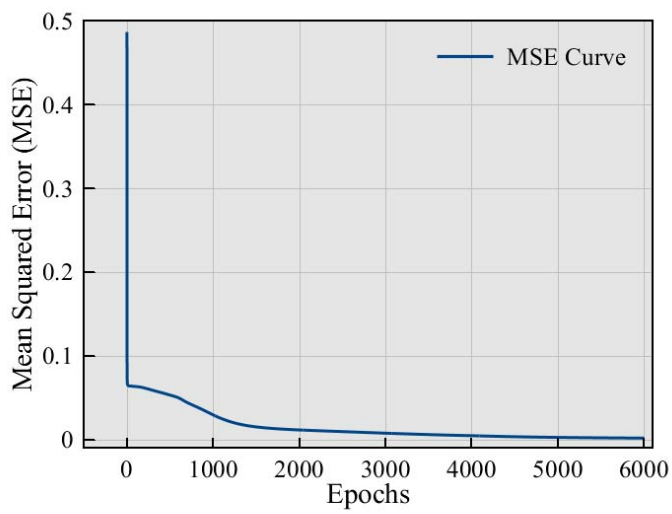

Fig. 6. Learning curve for PI-EmNN3 for gesture recognition.

IV. PI-EmNN Application to Recognition Tasks

In this section, we describe the training of the proposed network model (PI-EmNN) for the static hand-gesture recognition and face recognition tasks. Furthermore, PI-EmNNs are trained using different number of prototypes per class, i.e., 1, 3, and 5 prototypes per class for both static hand gesture and face recognition tasks. This allows the observation of learning experience and performance of the PI-EmNNs with different number of prototypes per class as described in Section II.

The databases used in the training of the networks for the aforementioned tasks are described in Section III. Also, for comparative analysis, the conventional back-propagation neural network (BPNN), EmNN, deep neural models, and k-NN are trained for the considered tasks.

TABLE I
TRAINING HYPER-PARAMETERS FOR STATIC HAND-GESTURE RECOGNITION

Network	PI-EmNN1 (1-prototype)	PI-EmNN3 (3-prototypes)	PI-EmNN5 (5-prototypes)	EmNN	BPNN
Number of training samples	1,440	1,440	1,440	1,440	1,440
Activation function	Sigmoid	Sigmoid	Sigmoid	Sigmoid	Sigmoid
Number of hidden neurons	30	30	30	30	30
Learning rate (η)	0.0007	0.0007	0.0007	0.0007	0.0007
Momentum rate (β)	0. 002	0.002	0.002	0.002	0.002
Epochs	6,000	6,000	6,000	6,000	6,000
Training time (secs)	564.7	586.49	584.16	544.31	504.47
Mean Squared Error (MSE)	0.0035	**0.0030**	0.0077	0.0057	0.0058
Anxiety coefficient (μ)	0.0321	**0.0097**	0.0108	0.0123	–
Confidence coefficient (k)	0.2798	**0.4946**	0.3427	0.4201	–

* Using a 2.0GHz (Dual core) PC with 3GB of RAM, Windows 7 OS and MATLAB programming environment

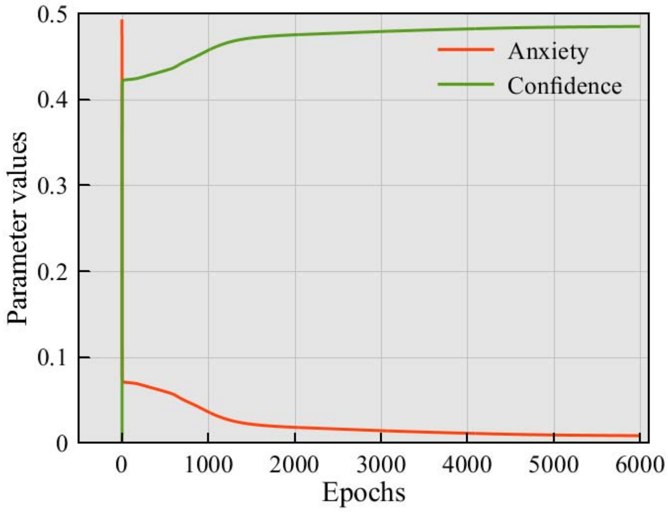

Fig. 7. PI-EmNN3 learning curve for emotional parameters.

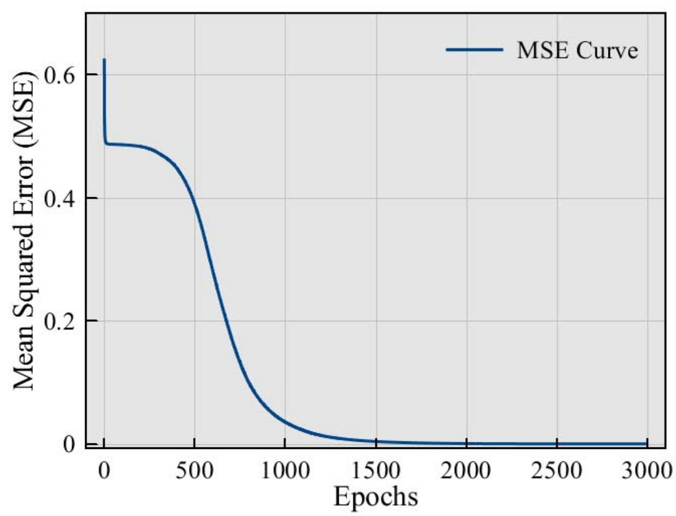

Fig. 8. Learning curve for PI-EmNN5 for face recognition.

A. Static Hand-Gesture Recognition

Several experiments are carried out to determine the hyper-parameters for the networks; these parameters are presented in Table I. Input images are all of size 32×32 pixels. The input layer has 1024 neurons (the size of input images), the output layer has 24 neurons (the number of output classes in the task); the number of hidden neurons was obtained heuristically as 30 during the training phase. Note that the sigmoid activations in Table I are the logistic function type. Table I shows that the PI-EmNN with three prototypes per class (PI-EmNN3) achieved the lowest MSE of 0.0030.

Also, it is observed that PI-EmNN3 achieved the lowest anxiety coefficient (μ), 0.0097, and highest confidence coefficient (k), 0.4946, at the end of training. The learning curve for PI-EmNN3 is shown in Fig. 6. The curve describing the learning of emotional parameters, anxiety coefficient (μ), and confidence coefficient (k) are shown in Fig. 7. As expected, it is observed that the anxiety coefficient drops as training progresses, while the confidence coefficient increases.

B. Face Recognition

The ORL face database described in Section III is used for the training of the PI-EmNNs. Networks are of 1024 input neurons (since input images are of size 32×32 pixels) and 40 output neurons; the suitable number of hidden neurons was obtained heuristically as 70 during training. Furthermore, the training hyper-parameters for the different networks are shown in Table II. Note that the activations functions shown in Table II are of the logistic function type. Also, it can be observed that the conventional EmNN and BPNN are not shown in Table II; the aforementioned networks are not considered for training in this work as results for comparative analysis with the proposed network models are obtained from an earlier work [50]. From Table II, it can be observed

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE II
TRAINING HYPER-PARAMETERS FOR FACE RECOGNITION

Network	PI-EmNN1 (1-prototype)	PI-EmNN3 (3-prototypes)	PI-EmNN5 (5-prototypes)
Number of training samples	200	200	200
Activation function	Sigmoid	Sigmoid	Sigmoid
Number of hidden neurons	70	70	70
Learning rate (η)	0.0089	0.0089	0.0089
Momentum rate (β)	0. 002	0.002	0.002
Epochs	3,000	3,000	3,000
Training time (secs)	643.8	657.2	668.5
Mean Squared Error (MSE)	0.0038	0.0025	**0.0017**
Anxiety coefficient (μ)	0.0054	0.0050	**0.0049**
Confidence coefficient (k)	0.6187	0.6221	**0.6253**

* Using a 2.0GHz (Dual core) PC with 3GB of RAM, Windows 7 OS and MATLAB programming environment

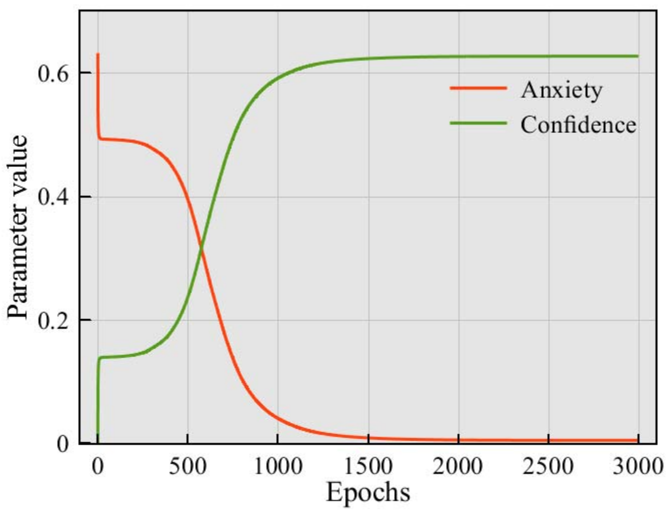

Fig. 9. PI-EmNN5 learning curve for emotional parameters.

that PI-EmNN5 achieved the lowest MSE on training. The error learning curve for PI-EmNN5 is shown in Fig. 8. Also, the learning curve of emotional parameters for PI-EmNN5 is shown in Fig. 9. It can be observed that for PI-EmNN5; the anxiety level has decreased to 0.0049 but confidence level has increased to 0.6253 at the end of training.

It is noteworthy to restate that emotional parameters, anxiety (μ), and confidence (k) are not set before training. The parameters are self-taught during training; detailed descriptions relating to how these parameters are learned during training are provided in Section II. Note that emotional parameters are updated progressively at the end of each iteration, and only the final values for the emotional parameters are reported, i.e., Tables I and II.

V. RESULTS, COMPARISON, AND DISCUSSION

For both tasks considered in this work, the trained networks are simulated with the training data. Also, the trained networks are simulated with the test data; this allows the observation of the generalization capability of the trained networks. The performances of the networks are assessed based on achieved recognition rates. The recognition rates (classification accuracy) of the models are obtained using

$$\text{Recognition rate} = \frac{\lambda_{\text{corrects}}}{\lambda_{\text{total simulated samples}}} \tag{25}$$

where $\lambda_{\text{corrects}}$ is the number of correctly classified samples and $\lambda_{\text{total simulated samples}}$ is the total number of simulated samples.

A. Static Hand-Gesture Recognition

The trained networks for the static hand-gesture recognition in Section IV-A are simulated with the training data and testing data. The achieved recognition rates for the trained networks, PI-EmNN1, PI-EmNN3, PI-EmNN5, EmNN, and BPNN are presented in Table III. It can be seen that stacked denoising auto encoder (SDAE) achieved the highest recognition rate on training data (i.e., 99.44%); PI-EmNN3 (with three prototypes per class) follows SDAE, outperforming other models including BPNN, EmNN, k-NN, and convolutional neural network (CNN). Also, in Table III, it can be seen that k-NN (with $k = 7$) achieved the highest recognition rate on the test data (i.e., 95.83%), slightly outperforming PI-EmNN3. More important is that on the test data, PI-EmNN3 and PI-EmNN5 outperform the other models including EmNN, BPNN, k-NN (with $k = 15$), SDAE, and CNN. Particularly, for the PI-EmNN models, it is observed that PI-EmNN3 achieved the highest recognition rate on the testing data (94.33%). PI-EmNN5 follows PI-EmNN3 in performance, while PI-EmNN1 slightly lags the EmNN in performance. It is observed that the incorporation of prototype knowledge into the conventional emotional BPNN improved the overall learning experience of the proposed model for the task; that is, the PI-EmNN3 and PI-EmNN5 outperform models which rely essentially on adaptive learning (i.e., BPNN, EmNN, SDAE, and CNN) and a model which rely solely on exemplar learning

TABLE III
RECOGNITION RATE (%) FOR STATIC HAND-GESTURE RECOGNITION

Model	Train data	Test data
PI-EmNN1 (1-prototype)	96.60	90.33
PI-EmNN3 (3-prototypes)	**99.24**	**94.33**
PI-EmNN5 (5-prototypes)	97.71	93.00
EmNN	96.39	90.83
BPNN	95.63	88.17
k-NN (with k=7)	**98.96**	**95.83**
k-NN (with k=15)	96.88	92.83
SDAE (4 hidden layers) [61]	99.44	92.83
CNN (4 hidden layers) [61]	98.13	91.33

TABLE IV
RECOGNITION RATE (%) FOR FACE RECOGNITION TASK

Model	Train data	Test data
PI-EmNN1 (1-prototype)	100	91.50
PI-EmNN3 (3-prototypes)	**100**	**93.50**
PI-EmNN5 (5-prototypes)	100	92.00
EmNN [50]	100	90.00
BPNN [50]	100	87.00
k-NN (with k=1)	100	91.00
k-NN (with k=3)	96.50	84.50
SDAE (3 hidden layers)	100	92.50
CNN (4 hidden layers)	99.50	83.00
SOM+CNN (4 hidden layers) [62]	**-**	**96.20**

(i.e., k-NN). This strengthens our aforementioned position that adaptive learning can benefit from prototype learning since our proposed model, PI-EmNN, relies on both.

B. Face Recognition

The trained networks in Section IV-B are simulated with both the training (200 samples) and testing data (200 samples).

Table IV shows the obtained recognition rates for the different models on both training and testing data. From all experiments performed in this work, it is observed that PI-EmNN3 (with three prototypes per class) outperforms EmNN,

BPNN, k-NN, SDAE, and CNN on testing data. For the SDAE, best performance was obtained with three hidden layers.

For the PI-EmNN models, it is observed that PI-EmNN5 (with five prototypes per class) follows PI-EmNN3 on performance, while PI-EmNN1 (with one prototype per class) slightly lags PI-EmNN5 on performance. In addition, we compare our models with models from another work [62] which employed the five images per subject for training and the other five images per subject for testing. Although a higher recognition rate was reported in [62], we note that they used self-organizing map (SOM) for explicit feature extraction and dimensionality reduction before classification with a CNN. This truly reflected on computational requirements in their experiments. For example, the SOM + CNN model was reported to have required 4 h for training; the explicit feature extraction via SOM alone took 100 000 weights updates for the ordering phase and 50 000 weights updates for the fine-adjustment phase. Conversely, our proposed model (PI-EmNN) did not employ any explicit feature extraction as observed in [62], required only 3000 epochs and a maximum training time of 670 s, see Table II. Also, it is well-known that CNNs are data "hungry" if they are to yield competitive performances. Therefore, we note that the explicit feature extraction and simultaneous data dimensionality reduction could be responsible for the improvement in the performance of CNN from 83% as obtained in our experiment in this work to 96.2% as reported in [62], see Table IV. In fact, [62] acknowledged the poor performance of CNN without the employed explicit feature extraction.

One significant finding on the proposed model is that the best performances are obtained when three prototypes per class are used; for both recognition tasks, the PI-EmNN with three prototypes per class yielded the best performances. In this paper, we experiment with 0, 1, 3, and 5 prototypes per class; note that the zero prototype per class can be seen as equivalent to the conventional EmNN presented in an earlier work [50]. The proposed models with five prototypes per class are found to follow the three prototypes per class models on performance.

Also, the one prototype per class models are found to have almost the same performance with the zero prototype per class models (EmNNs) and slightly lag the five prototypes per class models. For the static hand gesture and face recognition tasks, Fig. 10 shows the achieved recognition rates on the testing data for the proposed models based on the number of prototypes per class, i.e., from Tables IV and V. From Fig. 10, it can be observed that the proposed model (PI-EmNN) peaks for the static hand gestures and face recognition tasks with three prototypes per class, after which the performance of the model begins to decrease.

We conjecture that the initial increase in performance is associated with the prior knowledge acquired by the network based on the incorporation of prototype learning. Also, with three prototypes per class, the model is exposed to a more robust prototype learning based on the voting criterion described in Section II-A. In the one prototype per class scenario, insufficient or incorrect prior knowledge due to wrong associations of prototypes with other class labels is as a result of using only one prototype per class to determine the class labels of the training data; this may negatively impact learning as is seen in the static hand-gesture recognition task, where PI-EmNN1 slightly lags EmNN (zero prototype per class PI-EmNN) on testing recognition rate. Conversely, increasing the number of prototypes per class beyond 3 begins to introduce too many variations in the samples of possible prototypes; hence, the possibility that a presented input pattern will be wrongly associated with one of the

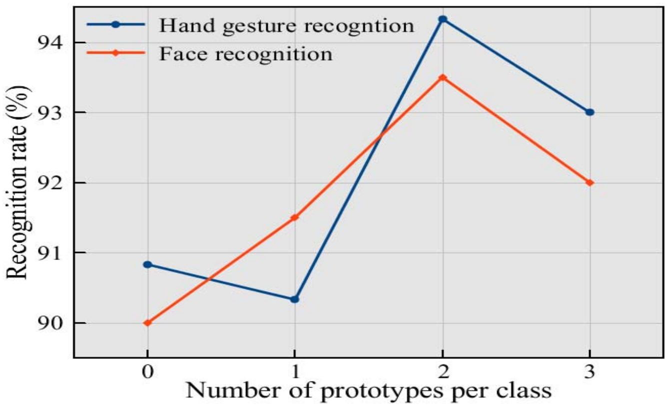

Fig. 10. PI-EmNN recognition rates on testing data against the number of prototypes per class.

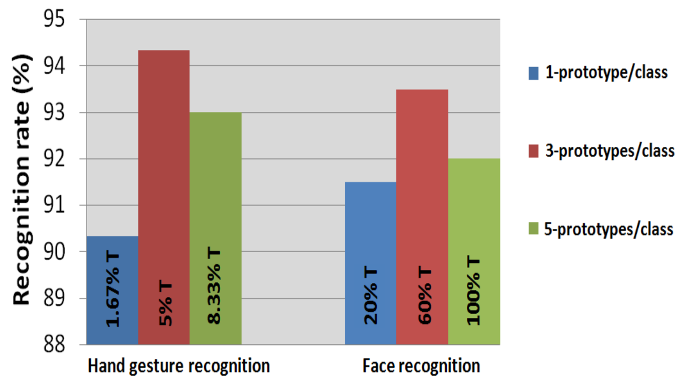

Fig. 11. PI-EmNN testing recognition rates as a function of the percentage of training data T selected as prototypes.

TABLE V
TEN-FOLD CROSS-VALIDATION RECOGNITION RATE (%) ON TEST DATA

Model/recognition task	Hand gesture	Face
PI-EmNN1 (1-prototype)	98.14	97.25
PI-EmNN3 (3-prototypes)	**99.12**	**98.00**
PI-EmNN5 (5-prototypes)	98.92	97.25
BPNN	97.94	96.75
EmNN	98.04	97.25
k-NN	98.58 (k=7)	97.50 (k=1)
k-NN	97.16 (k=15)	95.25 (k=3)
SDAE	98.58	96.50
CNN	98.87	95.75

prototypes of another class may again increase. Interestingly, Reisinger and Mooney reported similar performance in their work on multiprototype learning [45]. We posit that based on prototype learning, the acquired prior knowledge on class labels is useful in quickly guiding the PI-EmNNs toward good local minima in solution space. In fact, various pretraining schemes for deep neural networks are also somewhat hinged on this premise [63].

Furthermore, we consider how the generalization performances of the PI-EmNNs vary with the percentage of training data taken as prototypes. Fig. 11 shows the testing recognition rates of the PI-EmNNs for the hand gesture and face recognition tasks with the percentage of training data T taken as prototypes. This percentage can be calculated as $(P \times C/T) \times 100\%$, where P is the number of prototypes per class, C is the number of classes in the classification task, and T is the total number of training examples available for the task. Note that though we restricted the number of prototypes per class to a maximum of 5 irrespective of the number of available examples per class, a small training data set with respect to the number of classes can make the percentage of prototype data quite high; this is observable in the face recognition task. In this case, it is interesting to note that when the network is activated, only one of those prototypes (which could almost be identical to one of the exemplars) is provided so that learning remains unbiased in view of available data for training. It should be observed that for sufficiently large data sets such that the ratio C/T is extremely small; the percentage of prototype data will be quite small even at five prototypes per class, i.e., observable in the hand-gesture recognition task. Nevertheless, generalization performance did not strictly increase with the percentage of prototype data.

In order to further demonstrate the competitiveness of the proposed model, we perform additional experiments for the earlier recognition tasks using ten-fold cross-validation training scheme with the same model architectures and hyper-parameters as in the previous experiments. Table V shows the results obtained for the test data.

Particularly, for both recognition tasks considered in this work, we note that PI-EmNN3 outperforms all the other models based on ten-fold cross-validation results. The results are more interesting when one considers that the PI-EmNN models have only one hidden layer, but still outperforms both SDAE and CNN both with many hidden layers.

VI. CONCLUSION

This work builds on two fundamental-learning theories for explaining category generalization in humans, that is, prototype- and adaptive-learning theories. We share the idea that both the prototype and adaptive theories are valid for learning. Therefore, we propose that incorporating prototype knowledge into neural networks can be used to improve the overall learning experiences of such networks. The proposed neural network model has been applied to two challenging tasks in machine vision; namely, static hand-gesture recognition and face recognition. All the experiments performed in this work show that the incorporation of prototype learning into the EmNN improves overall learning and generalization. In addition, it is interesting that our proposed model which employs only one hidden layer and no convolution operations for feature learning achieves competitive performance against models with many hidden layers of features abstraction and convolution operations, i.e., deep neural networks.

Future work includes the application of the proposed model to a wider domain of machine vision and pattern recognition problems. The authors believe that using an expanded domain of problems, it is possible to investigate even further that the optimum number of prototypes per class obtained within this work holds for a broad range of visual tasks. Furthermore, the connection between prototype learning and semisupervised learning as obtains in deep networks is an interesting future research direction.

ACKNOWLEDGMENT

The authors would like to thank the Associate Editor and anonymous reviewers for their constructive criticisms and suggestions which helped improve this paper.

REFERENCES

[1] M. Uzair, A. Mahmood, and A. Mian, "Hyperspectral face recognition with spatiospectral information fusion and PLS regression," *IEEE Trans. Image Process.*, vol. 24, no. 3, pp. 1127–1137, Mar. 2015.
[2] Y. Hu, D. Wu, and A. Nucci, "Fuzzy-clustering-based decision tree approach for large population speaker identification," *IEEE Trans. Audio, Speech, Language Process.*, vol. 21, no. 4, pp. 762–774, Apr. 2013.
[3] E. M. Sibarani, M. Nadial, E. Panggabean, and S. Meryana, "A study of parsing process on natural language processing in Bahasa Indonesia," in *Proc. IEEE 16th Int. Conf. Comput. Sci. Eng. (CSE)*, Dec. 2013, pp. 309–316.
[4] O. K. Oyedotun and A. Khashman, "Document segmentation using textural features summarization and feedforward neural network," *Appl. Intell.*, vol. 45, no. 1, pp. 198–212, 2016.
[5] R. Khosrowabadi, C. Quek, K. K. Ang, and A. Wahab, "ERNN: A biologically inspired feedforward neural network to discriminate emotion from EEG signal," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 3, pp. 609–620, Mar. 2014.
[6] M. Rajah and A. McIntosh, "Overlap in the functional neural systems involved in semantic and episodic memory retrieval," *J. Cognit. Neurosci.*, vol. 17, no. 3, pp. 470–482, Mar. 2005.
[7] A. A. Hopgood, "Artificial intelligence: Hype or reality?" *Computer*, vol. 36, no. 5, pp. 24–28, May 2003.
[8] A. L. Von, M. Blum, and J. Langford, "Telling humans and computers apart automatically," *Commun. ACM*, vol. 47, no. 2, pp. 56–60, 2004.
[9] J. F. Mas and J. J. Flores, "The application of artificial neural networks to the analysis of remotely sensed data," *Int. J. Remote Sens.*, vol. 29, no. 3, pp. 617–663, 2008.
[10] J. M. Zurada, *Introduction to Artificial Neural Systems*. Saint Paul, MN, USA: West, 1992, pp. 2218–2295.
[11] G. G. Lendaris, "Adaptive dynamic programming approach to experience-based systems identification and control," *Neural Netw.*, vol. 22, nos. 5–6, pp. 822–832, 2009.
[12] J. Onians, "The role of experiential knowledge in the ultimate design studio: The brain," *J. Res. Pract.*, vol. 6, no. 2, p. 11, 2010.
[13] Y. Li, T. Sawada, L. J. Latecki, R. M. Steinman, and Z. Pizlo, "A tutorial explaining a machine vision model that emulates human performance when it recovers natural 3D scenes from 2D images," *J. Math. Psychol.*, vol. 56, no. 4, pp. 217–231, 2012.
[14] S. J. Dickinson, *Object Categorization: Computer and Human Vision Perspectives*. Cambridge, U.K.: Cambridge Univ. Press., 2009.
[15] S. R. Zaki, R. M. Nosofsky, R. D. Stanton, and A. L. Cohen, "Prototype and exemplar accounts of category learning and attentional allocation: A reassessment," *J. Experim. Psychol., Learn., Memory, Cognit.*, vol. 29, no. 6, pp. 1160–1173, 2003.
[16] J. L. Elman, "Learning and development in neural networks: The importance of starting small," *Cognition*, vol. 48, no. 1, pp. 71–99, Jul. 1993.
[17] M. Frixione and A. Lieto, "Prototypes vs exemplars in concept representation," in *Proc. Int. Conf. Knowl. Eng. Ontol. Develop. (KEOD)*, 2012, pp. 226–232.
[18] J. A. Hampton, "Testing the prototype theory of concepts," *J. Memory Lang.*, vol. 34, no. 5, pp. 686–708, 1995.
[19] Y. Azrieli and E. Lehrer, "Categorization generated by extended prototypes—An axiomatic approach," *J. Math. Psychol.*, vol. 51, no. 1, pp. 14–28, 2007.
[20] J. Hampton, I. Van Mechelen, R. S. Michalski, and P. Theuns, Eds., *Categories and Concepts: Theoretical Views and Inductive Data Analysis* (Cognitive Science Series). San Diego, CA, USA: Academic, 1993, pp. 67–95.
[21] D. Zeithamova, "Prototype learning systems," in *Encyclopedia of the Sciences of Learning*. New York, NY, USA: Springer, 2012, pp. 2715–2718.
[22] F. Chang, C.-C. Lin, and C.-J. Lu, "Adaptive prototype learning algorithms: Theoretical and experimental studies," *J. Mach. Learn. Res.*, vol. 7, pp. 2125–2148, Oct. 2006.
[23] A. B. A. Graf, O. Bousquet, G. Rätsch, and B. Schölkopf, "Prototype classification: Insights from machine learning," *Neural Comput.*, vol. 21, no. 1, pp. 272–300, 2009.
[24] L. R. Gleitman, A. C. Connolly, and S. L. Armstrong, "Can prototype representations support composition and decomposition," in *The Oxford Handbook of Compositionality*. New York, NY, USA: Oxford Univ. Press, 2012.
[25] J. van der Auwera and V. Gast, "Categories and prototypes," in *The Oxford Handbook of Language Typology*, J. J. Song, Ed. Oxford, U.K.: Oxford Univ. Press, 2011.
[26] J. S. Pan, N. Bingham, and G. P. Bingham, "Embodied memory: Effective and stable perception by combining optic flow and image structure," *J. Experim. Psychol., Hum. Perception Perform.*, vol. 39, no. 6, p. 1638, 2013.
[27] M. Meng and M. C. Potter, "Detecting and remembering pictures with and without visual noise," *J. Vis.*, vol. 8, no. 9, p. 7, 2008.
[28] C. Kim and P. Milanfar, "Visual saliency in noisy images," *J. Vis.*, vol. 13, no. 4, p. 5, 2013.
[29] S. Haykin, *Neural Networks and Learning Machines*, vol. 3. Upper Saddle River, NJ, USA: Pearson Education, 2009.
[30] G. E. Hinton, "Connectionist learning procedures," *Artif. Intell.*, vol. 40, nos. 1–3, pp. 185–234, 1989.
[31] J. N. Rouder and R. Ratcliff, "Comparing exemplar- and rule-based theories of categorization," *Current Directions Psychol. Sci.*, vol. 15, no. 1, pp. 9–13, 2006.
[32] C. J. Marsolek, "Abstractionist versus exemplar-based theories of visual word priming: A subsystems resolution," *Quart. J. Experim. Psychol. A, Hum. Experim. Psychol.*, vol. 57, no. 7, pp. 1233–1259, 2004.
[33] L. Karlsson, P. Juslin, and H. Olsson, "Exemplar-based inference in multi-attribute decision making: Contingent, not automatic, strategy shifts," *Judgment Decision Making*, vol. 3, no. 3, pp. 244–260, 2008.
[34] A. M. Hermundstad, K. S. Brown, D. S. Bassett, and J. M. Carlson, "Learning, memory, and the role of neural network architecture," *PLoS Comput. Biol.*, vol. 7, no. 6, p. e1002063, 2011.
[35] M. Gong, J. Liu, H. Li, Q. Cai, and L. Su, "A multiobjective sparse feature learning model for deep neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 12, pp. 3263–3277, Dec. 2015.
[36] R. S. Sutton and A. G. Barto, "Toward a modern theory of adaptive networks: Expectation and prediction," *Psychol. Rev.*, vol. 88, no. 2, p. 135, 1981.
[37] B. Widrow and M. A. Lehr, "30 years of adaptive neural networks: Perceptron, madaline, and backpropagation," *Proc. IEEE*, vol. 78, no. 9, pp. 1415–1442, Sep. 1990.
[38] T. Griffiths, K. Canini, A. Sanborn, and D. Navarro, "Unifying rational models of categorization via the hierarchical Dirichlet process," in *Proc. 29th Annu. Cognit. Sci. Soc.*, Nashville, TN, USA, Aug. 2007, pp. 323–328.
[39] T. Verbeemen *et al.*, "Beyond exemplars and prototypes as memory representations of natural concepts: A clustering approach," *J. Memory Lang.*, vol. 56, no. 4, pp. 537–554, 2007.
[40] S. Edelman, W. Vanpaemel, S. Pattyn, G. Storms, and T. Verguts, "Representation, similarity, and the chorus of prototypes," *Minds Mach.*, vol. 5, no. 1, pp. 45–68, 1995.
[41] J. C. Bezdek, T. R. Reichherzer, G. Lim, and Y. Attikiouzel, "Classification with multiple prototypes," in *Proc. 5th IEEE Int. Conf. Fuzzy Syst.*, vol. 1. Sep. 1996, pp. 626–632.
[42] Y. Cai, H.-F. Leung, and A. W.-C. Fu, "Multi-prototype concept and object typicality in ontology," in *Proc. 21st Int. FLAIRS Conf.*, Coconut Grove, FL, USA, May 2008, pp. 470–475.
[43] F. Aiolli and A. Sperduti, "Multiclass classification with multi-prototype support vector machines," *J. Mach. Learn. Res.*, vol. 6, pp. 817–850, May 2005.
[44] Y. Cai, H. Zhao, H. Han, R. Y. K. Lau, H.-F. Leung, and H. Min, "Answering typicality query based on automatically prototype construction," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol.*, vol. 1. Dec. 2012, pp. 362–366.
[45] J. Reisinger and R. J. Mooney, "Multi-prototype vector-space models of word meaning," in *Proc. Hum. Lang. Technol., Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, Jun. 2010, pp. 109–117.

[46] T. Long and L.-W. Jin, "A new simplified gravitational clustering method for multi-prototype learning based on minimum classification error training," in *Advances in Machine Vision, Image Processing, and Pattern Analysis*. Berlin, Germany: Springer, 2006, pp. 168–175.

[47] M. Liu, X. Jiang, and A. C. Kot, "A multi-prototype clustering algorithm," *Pattern Recognit.*, vol. 42, no. 5, pp. 689–698, 2009.

[48] A. N. Fengwei, T. Koide, and H. J. Mattausch, "A K-means-based multi-prototype high-speed learning system with FPGA-implemented coprocessor for 1-NN searching," *IEICE Trans. Inf. Syst.*, vol. E95.D, no. 9, pp. 2327–2338, Sep. 2012.

[49] P. G. Schyns, "A modular neural network model of concept acquisition," *Cognit. Sci.*, vol. 15, no. 4, pp. 461–508, 1991.

[50] A. Khashman, "A modified backpropagation learning algorithm with added emotional coefficients," *IEEE Trans. Neural Netw.*, vol. 19, no. 11, pp. 1896–1909, Nov. 2008.

[51] *Thomas Moeslund's Gesture Recognition Database—PRIMA*. Accessed on Jan. 12, 2016. [Online]. Available: http://www-prima.inrialpes.fr/FGnet/data/12-MoeslundGesture/database.html

[52] *ORL (AT & T) Face Database*. Accessed on Jan. 07, 2016. [Online]. Available: http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

[53] A. Khashman, "An emotional system with application to blood cell type identification," *Trans. Inst. Meas. Control*, vol. 34, nos. 2–3, pp. 125–147, 2012.

[54] A. Khashman, "Modeling cognitive and emotional processes: A novel neural network architecture," *Neural Netw.*, vol. 23, no. 10, pp. 1155–1163, 2010.

[55] A. Khashman, "Blood cell identification using emotional neural networks," *J. Inf. Sci. Eng.*, vol. 25, no. 6, pp. 1737–1751, 2009.

[56] L. Liberti, C. Lavor, N. Maculan, and A. Mucherino, "Euclidean distance geometry and applications," *SIAM Rev.*, vol. 56, no. 1, pp. 3–69, 2014.

[57] M. M. Mukaka, "A guide to appropriate use of correlation coefficient in medical research," *Malawi Med. J.*, vol. 24, no. 3, pp. 69–71, 2012.

[58] R. C. Quinino, E. A. Reis, and L. F. Bessegato, "Using the coefficient of determination R2 to test the significance of multiple linear regression," *Teach. Statist.*, vol. 35, no. 2, pp. 84–88, 2013.

[59] S. L. Crawford, "Correlation and regression," *Circulation*, vol. 114, no. 19, pp. 2083–2088, 2006.

[60] P. H. Young, "Generalized coefficient of determination," *J. Cost Anal. Manage.*, vol. 2, no. 1, pp. 59–68, 2000.

[61] O. K. Oyedotun and A. Khashman, "Deep learning in vision-based static hand gesture recognition," *Neural Comput. Appl.*, vol. 27, no. 3, pp. 1–11, 2016.

[62] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE Trans. Neural Netw.*, vol. 8, no. 1, pp. 98–113, Jan. 1997.

[63] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, Dec. 2010.

Oyebade K. Oyedotun (S'14–M'16) received the M.Sc. degree in electrical and electronic engineering from Near East University, Lefkosa, Turkey, in 2015. He is currently pursuing the Ph.D. degree with the Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Luxembourg City, Luxembourg.

Since 2015, he has been a Senior Member with the European Centre for Research and Academic Affairs, Lefkosa. His current research interests include machine learning, vision applications, and cognition modeling.

Adnan Khashman (M'91–SM'01) received the B.Eng. degree in electronic and communication engineering from the University of Birmingham, Birmingham, U.K., in 1991, and the M.S. and Ph.D. degrees in electronic engineering from the University of Nottingham, Nottingham, U.K., in 1992 and 1997, respectively.

During 1998–2014, he served as the Chairman of Computer Engineering Department, the Chairman of Electrical and Electronic Engineering Department, and the Dean of the Engineering Faculty at Near East University, Lefkosa, Turkey. From 2014 to 2015, he was the Founding Dean of Engineering Faculty and the Vice-Rector at the British University of Nicosia, Girne, Turkey. Since 2015, he has been the Founder and Director of the European Centre for Research and Academic Affairs, Lefkosa. He is currently a Professor with the Faculty of Engineering, Final International University, Girne. His current research interests include image processing, pattern recognition, emotional neural modeling, and intelligent systems.