Complex Intell. Syst. (2016) 2:221–234
DOI 10.1007/s40747-016-0024-6

ORIGINAL ARTICLE

A novel training algorithm for convolutional neural network

Alwin Anuse[1] · Vibha Vyas[2]

Received: 13 March 2016 / Accepted: 19 August 2016 / Published online: 13 September 2016

Abstract Many machine learning softwares are available which help the researchers to accomplish various tasks. These software packages have various conventional algorithms which perform well if the training and test data are independent and identically distributed. However, this might not be the case in the real world. The training data may not be available at one time. In the case of neural networks, the architecture has to be retrained with new data that are made available subsequently. In this paper, we present a novel training algorithm which can avoid complete retraining of any neural network architecture meant for visual pattern recognition. To show the utility of the algorithm, we have investigated the performance of convolutional neural network (CNN) architecture for a face recognition task under transfer learning. The proposed training algorithm may be used for enhancing the utility of machine learning software by providing researchers with an approach that can reduce the training time under transfer learning.

Keywords Convolutional neural network · Minimum change principle · Variance · Trainable weights · Transfer learning

Introduction

Machine learning algorithms aim at building a model from example inputs in order to make data-driven decisions or predictions. Applications such as face recognition, spam filtering, and recommendation engines which use large dataset uses machine learning. Google uses machine learning to identify and deindex webspam. Various machine learning software such as Weka, Java Neural Network Framework Neuroph, Scikit Learn, Open NN Multiple Back propagation exists that assists researchers in solving complex problems. These packages have conventional algorithms [1–9] for image analysis, machine learning and data mining that assume training and test data have the same distribution. In many real-world applications, this may not hold, for example, if one has to detect users current location using previously collected Wi-Fi data. It is expensive to calibrate Wi-Fi data in large-scale environment as the user needs to label extensive collection of Wi-Fi signal at each location. Knowledge transfer or transfer learning may be useful in saving significant efforts in labeling data [10]. Transfer of knowledge from a related task that has already been learned to a new task which shares some of the commonality is transfer learning. Basics of transfer learning are well explained in [11]. Transfer learning aims to solve the problem when the training and test data are different. Transfer learning approaches like instance transfer, feature representation transfer, parameter transfer, and relational knowledge transfer are discussed in [12–16]. Transfer learning finds its motivation in the fact that human beings can intelligently apply knowledge acquired previously to solve the new problem faster or with better solutions. NISP-95 workshop on "Learning to Learn" had a special session that discussed the fundamental motivation behind transfer Learning. The workshop was focused on the need for lifelong machine learning techniques that retain and reuse previously acquired knowledge [12,17,18]. Thus, the machine learning software packages should provide some simple and automatic/semiautomatic setting for users dealing with transfer learning tasks.

✉ Alwin Anuse
alwin.anuse@mitpune.edu.in

[1] Department of E&TC, Maharashtra Institute of Technology, Kothrud 38, Pune, India

[2] Department of E&TC College of Engineering, Pune, India

Multilayer feedforward neural networks are been effectively used in machine learning. They can be used to approximate complex nonlinear functions from high-dimensional input data. The performance of multilayer perceptron (MLP) depends on the underlying feature extraction method used [19]. The choice of feature extraction algorithm and features used for classification is often empirical, and therefore, it is suboptimal. One can directly use the training algorithm to find the best feature extractors by adjusting the weights. However, when the input dimension is high (image processing application), the number of connections, the number of free parameters increases because each hidden unit is fully connected to the input layer. This may lead to a network that overfits the data as the neural network would have a too high complexity. The input patterns are to be well aligned and normalized while presenting to such type of MLP leading to no built in variance with respect to local distortions and translations [20]. Various neural network classifiers are explained in [18,21–26]. A convolutional neural network(CNN) tries to solve the problems of MLP by extracting local features and combining them subsequently to perform the detection or recognition. CNN and neocognitron are the neural network architectures which are meant for visual pattern recognition. These architectures have integrated feature extraction and classification layers. However, in the literature, no work has been reported which focuses on neural networks (meant for visual pattern recognition) equipped with transfer learning without making changes in the architecture.

The contributions of this paper include the following:

1. Novel training algorithm for CNN architecture under transfer learning task. Three-phase training algorithm is proposed for the same. Phase I is a conventional phase in which CNN is trained with the conventional methods in [27–29]. Phase II is a knowledge transfer in which knowledge acquired from new training samples is transferred into the architecture with minimum changes in the free parameters (weights) of the neural network architecture. Phase III is a weight update for transfer learning phase. Phase II and III are the new steps added to the existing, i.e., conventional algorithm of CNN. These phases avoid complete retraining of CNN when new training data are available subsequently once the CNN is trained with old data. The proposed algorithm may enhance the utility of any machine learning software by reducing the training time for transfer learning.
2. A training method which can be used for any neural network architecture meant for visual pattern recognition under transfer learning.
3. The novel dataset that is unique and meant to advance the research on face recognition under transfer learning.
4. Minimum change principle is been proposed which can be used to train a neural network under transfer learning.

The remainder of this paper is organized as follows. Section “Related work” throws light on the work done in the area of transfer learning and deep learning. The aim of the proposed work is to equip CNN (deep learning network) with transfer learning framework. This section explains various ongoing applications in field of transfer learning and deep learning. Section “The framework of transfer learning” explains the transfer learning framework used in this research. Framework of transfer learning is applied to principal component analysis (PCA) to derive the projection matrix. Section “Convolutional neural networks” explains the architecture of CNN followed by the proposed training algorithm for the CNN architecture. Section “Comparison of traditional algorithm (conventional) with proposed algorithm” explains the comparison of traditional algorithm with the proposed algorithm. Section “Dataset” describes the dataset that is used in this research. Section “Experiments, parameter settings, and observations” throws light on the experiments performed on the CNN architecture. This section also explains various parameter setting in the algorithm followed by the observations. Section “Conclusion” brings ahead the conclusion of this research work.

Related work

In the last few years, visual recognition community has shown a growing interest in transfer learning algorithms [30,31]. Transfer subspace learning (TSL) is effectively used in understanding kin relationships in the photo [32]. Classification under covariate shift is been solved by transfer learning [33]. Features with meta-features that can be used in prediction task is studied in [34]. Building classifiers for text classification by extracting positive examples from unlabeled examples for improving performance of the system are highlighted in [10]. Transfer subspace learning that can reduce time and space cost is proposed in [35]. Enhanced subspace clustering algorithms [36,37] are used to handle complex data and to improve clustering results. Cross-domain discriminative locally linear embedding (CDLLE) can be used to reduce the human labeling efforts for social image annotation problem [38]. Robust framework against noise in the transfer learning setting is proposed in [39]. Semisupervised clustering algorithm with domain adaptation and the constraint knowledge with transferred centroid regularization is proposed in [40]. Xiaoxin Yin et al. have proposed [41] efficient classification across multiple database relations. Performance improvement is seen when transfer learning is used in medical image segmentation followed by classification [42]. Low-resolution face images are matched with the high-resolution gallery images using transfer learning which improved cross-resolution face matching [43]. Transfer learning using Bayesian model was used in [44] for face verification application. Ensemble-based transfer learning was used in text classification [45]. Knowledge was transferred between text and images using matrix factorization approach by Zhu et al. [46]. Geng et al. used domain adaptation metric learning for face recognition and web image annotation [47]. Server-based spam filter learned from public sources was designed and applied to individual users with the help of transfer learning [48].

In recent years due to its state-of-the art performance in many research domains, deep learning has attracted attention of academic community. Companies like Google, Facebook and Apple who collect and analyze massive amounts of data are putting forward lot of deep learning-related projects that happens to be the prime motivation behind this research. Deep learning challenges and perspectives are well explained in [49]. Weilong Hou et al. have done blind quality assessment via deep learning [50]. Shuhui Bu et al. for the first time applied deep learning for 3D shape retrieval [51]. Traffic flow prediction and deep learning approach is been proposed in [52]. Object tracking in blurred videos using blurred videos and deep image representations is proposed by Jianwei Ding et al. [53]. Adaptively learn representation that is more effective for the task of vehicle color recognition using spatial pyramid deep learning is given by Chuanping Hu et al. [54]. Deep learning is also been used to grade nuclear cataracts [55]. Deep learning is been widely used in medical image processing for segmentation, classification and registration [56–61], image denoising [62] and multimodal learning [63]. Deep learning is proved to give robust image representation for single training sample per person in face recognition task [64]. Corey Kereliuk et al. did music content analysis with deep learning [65]. Land use classification [66], scene classification [67] and visual tracking [68] applications work well with deep learning architectures. Impact of deep learning on developmental robotics is explained in [69]. Multi-label image annotation is been achieved using semisupervised deep learning [70]. Financial signal representation is done in [71] using deep neural networks. Pipeline for object detection and segmentation in the context of volumetric image parsing is proposed using marginal space deep learning [72]. Deep learning is also been used in indoor localization that reduces the location error compared with the three existing methods [73]. Convolutional neural networks (CNN), a very popular deep learning network is used in almost all the applications since it is believed to be one of the most appropriate networks for modeling images [74]. CNN are used for image classification [75], pose estimation [76], face recognition [77] and modeling texts [78–84].

The proposed work contrasts clearly from a concurrent work on deep learning and transfer learning in following ways:

1. Support vector machine (SVM) is extensively used in transfer learning methods. Most of the transfer learning algorithms are developed only for specific model that makes it difficult to use it for other models and restrict the applicability. To the best of author's knowledge and the data available from literature, the first attempt made to equip deep neural network with transfer learning framework was by Mingsheng Long et al. [85]. In their framework they have modified the architecture of CNN. However, the research work proposed in this paper is for the conventional CNN architecture. A novel training algorithm under transfer learning is proposed without changing the architecture of CNN. There was also an attempt to equip shallow neural network with transfer learning [52]. The authors of this paper also acknowledge the work of Fan Zhang et al., in their work they have suggested a neural network ensemble training to improve prediction accuracy at the expense of increased trainable parameters [67]. In short the proposed algorithm is generic and can be used for any deep learning architecture.
2. The transfer learning task is demonstrated with applications like medical image segmentation, text classification [86], web image annotation, face recognition, etc. Various standard datasets like Yale Face database, the Facial Recognition Technology (FERET) and Labeled Faces in the Wind (LFW) exists for doing the experimentation on recognition of faces. The face images in these datasets are acquired with various poses, illumination [87] and expressions, etc. No dataset of face exists which has face images acquired at different distances. The authors of this paper have made their own dataset which may be used by researchers working on a problem of face recognition at a distance. The details of the dataset is explained in Sect. "Dataset".

The framework of transfer learning

Given m training samples with x as a input and t as a target for classification task, $T = \{(X_1, t_1), (X_2, t_2), (X_m, t_m)\}$, n testing samples, $U = \{(X_{m+1}), (X_{m+2}), \ldots (X_{m+n},)\}$. The samples are drawn from a high-dimensional space R^{D}. The subspace learning algorithm finds a low-dimensional space R^{d}. A linear function $y = W^{\mathrm{T}}X$, where $W \epsilon R^{D\times d}$ and $y \epsilon \mathrm{R}^{\mathrm{d}}$ will find a low-dimensional space R^{d}. The objective function can be

$$W = \arg\min E(W) \tag{1}$$

subject to constraints $W^{\mathrm{T}}W = I$. The objective function $E(W)$ reduces the classification error. Equation (1) performs well if training and testing samples are independent and identically distributed. However, in practice, this might not be true. The distribution of training samples P_T and that of testing samples P_U may be different. Under such conditions, the learning framework given by equation fails. To solve this problem, the Bregman divergence-based regularization $D_W(P_T||P_U)$ which measures the distribution difference of samples in a projected subspace W is used [17]. Equation (1) is modified for transfer learning. The new framework for transfer learning is given as (2)

$$W = \arg\ \min E(W) + \rho D_W(P_T||P_U) \tag{2}$$

With constraints, e.g., $W^T\ W = I$. In (2) ρ is the regularization parameter that controls the trade-off between $E(W)$ and $D_{\mathrm{W}}(P_{\mathrm{T}}||P_{\mathrm{U}})$. The solution of (2) can be obtained by the gradient descent algorithm and is given by

$$W(\mathrm{new}) = W(\mathrm{old}) - \alpha(\partial\ E(W)/\partial W) + O, \tag{3}$$

where $O = \rho\ \frac{\partial D_W(P_T||P_U)}{\partial W}/\partial W$ and α is the learning rate.

Framework of TSL applied to principal component analysis (PCA)

Principal component analysis (PCA) projects the high-dimensional data to lower dimensional space by capturing maximum variance [88]. PCA projection matrix maximizes the trace of the total scatter matrix

$$W = arg\ max\ tr(W^T RW) \tag{4}$$

Subject to $\mathrm{WW}^T = I$. R is the autocorrelation matrix of training samples. $E(W)$ of PCA is given by (5)

$$E(W) = -tr(W^T RW) \tag{5}$$

$$\partial E(W)/\partial W = -2RW \tag{6}$$

By substituting (5) and (6) into (3), we can obtain the projection matrix W for transfer learning. The detailed procedure to get the solution of (3) is given in [2].

Convolutional neural networks

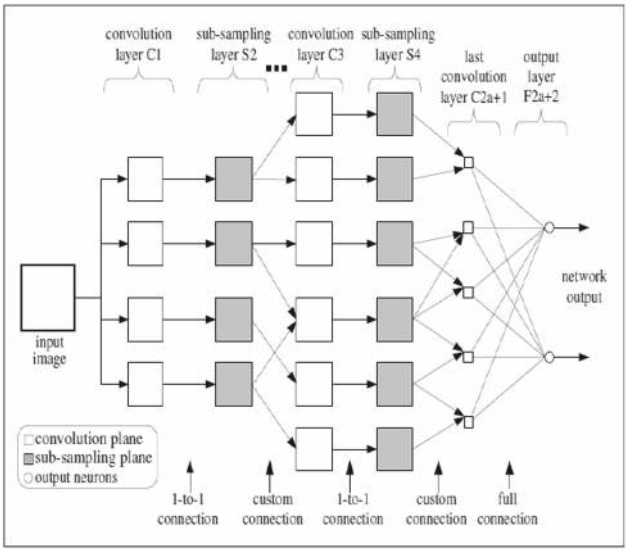


Fig. 1 Basic convolutional architecture for face recognition

Figure 1 shows convolutional neural network for face recognition task. The input plane receives images. The input is 74 × 74 pixel image. Layer C_1 is a convolutional layer with six feature maps. Each unit in each feature map has a connection to the 11 × 11 neighborhood in the input. The size of the feature map is 64 × 64. C_1 contains six kernels of size 11 × 11 and six biases, so the total number of trainable weights is 732.

Layer S_2 is a subsampling layer with six feature maps of size 32 × 32. Each unit in feature map has a connection to a 2 × 2 neighborhood in the corresponding feature map of C_1. Layer S_2 has no trainable weights.

Layer $C_3$is a convolutional layer consisting of 16 feature maps, i.e., 16 kernels of size 11 × 11 and sixteen biases which result in 1952 trainable weights.

Layer S_4 is a subsampling layer with 16 feature maps of size 22 × 22. The S_4 layer has no trainable parameters.

Layer C_5 is a convolutional layer consisting of 120 feature maps, i.e., 120 kernels of size 11 × 11 and 120 biases which result in 14,640 trainable weights.

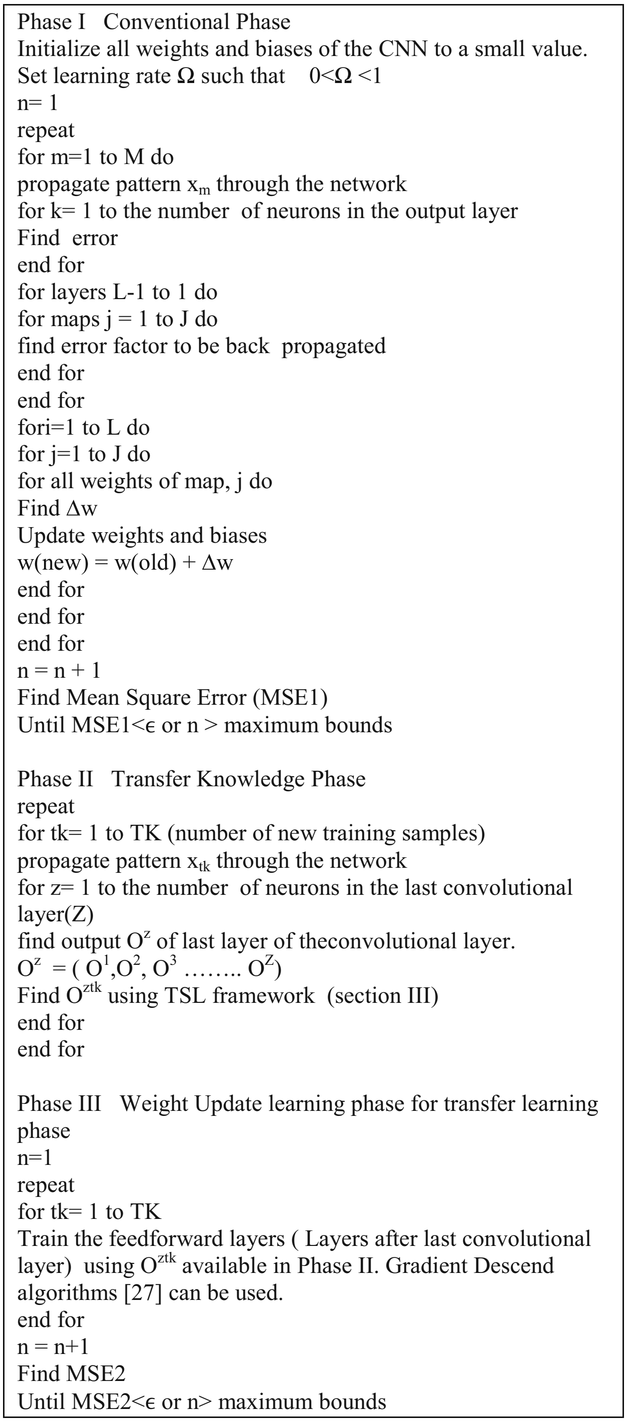
Layer F_6 contains 84 units, and the output layer consists of 25 units for solving a classification problem of 25 users or 50 units for solving a classification problem of 50 users. Trainable parameters for layer F_6 are 10,080. Trainable parameters for output layer are 2100 for 25 users and 4200 for 50 users classification problem.

Proposed three-phase training algorithm for CNN architecture using transfer learning approach

Figure 2 shows the proposed three-phase training algorithm.

Comparison of traditional algorithm (conventional) with proposed algorithm

Phase I of the algorithm is the training of CNN C and S Layers. Supervised learning is used to train C and S layers. A gradient descend method is used to update the weights in



```
Phase I   Conventional Phase
Initialize all weights and biases of the CNN to a small value.
Set learning rate $\Omega$ such that   $0<\Omega<1$
n= 1
repeat
for m=1 to M do
propagate pattern $x_m$ through the network
for k= 1 to the number  of neurons in the output layer
Find  error
end for
for layers L-1 to 1 do
for maps j = 1 to J do
find error factor to be back  propagated
end for
end for
fori=1 to L do
for j=1 to J do
for all weights of map, j do
Find $\Delta$w
Update weights and biases
w(new) = w(old) + $\Delta$w
end for
end for
end for
n = n + 1
Find Mean Square Error (MSE1)
Until MSE1<$\epsilon$ or n > maximum bounds

Phase II   Transfer Knowledge Phase
repeat
for tk= 1 to TK (number of new training samples)
propagate pattern $x_{tk}$ through the network
for z= 1 to the number  of neurons in the last convolutional
layer(Z)
find output $O^z$ of last layer of theconvolutional layer.
$O^z$  = ( $O^1$,$O^2$, $O^3$ …….. $O^Z$)
Find $O^{ztk}$ using TSL framework  (section III)
end for
end for

Phase III   Weight Update learning phase for transfer learning
phase
n=1
repeat
for tk= 1 to TK
Train the feedforward layers ( Layers after last convolutional
layer)  using $O^{ztk}$ available in Phase II. Gradient Descend
algorithms [27] can be used.
end for
n = n+1
Find MSE2
Until MSE2<$\epsilon$ or n> maximum bounds
```

Fig. 2 Proposed three-phase training algorithm for CNN architecture under transfer learning approach

Table 1 Proposed database

	Proposed database
Source	COEP and MIT Pune
Purpose	Designed for studying the problem of transfer subspace learning
Number of subjects	50
Number of images/videos	20,000
Static/videos	Static
Single/multiple faces	Single
Gray/color	Color
Resolution	640 × 480 and 2816 × 2112
Face pose	Frontal view
Facial expression	Neutral
Illumination	Controlled illumination
Ground truth	Identification of subjects under transfer subspace learning

all the layers. Phase II of the algorithm is only used when new training samples are available. The issue is to incorporate the information available from the new samples into the trained network. This issue is solved by the Phase II step of the algorithm. In this phase output, O^z of the last CNN layer is tapped and reweighted or updated using Eq. (3) to get new vector O^{ztk} for each training sample. In Phase III step of the algorithm, layer F_6 is trained with O^{ztk} as training vectors for the classification task.

Traditional/conventional algorithm which is used to train CNN has following two steps:

1. Conventional phase: this phase is same as conventional phase of proposed algorithm. MSE1 is the performance index used in this phase. This phase is used to train feature extraction layers of CNN (C_1, C_3 and C_5).
2. Weight updating phase: output of C_5 layer which is also called as features is used to train F_6 and output layer. Weight modification is done by using all the samples in the training data set. MSE2 is the performance goal used in this step.

In the proposed algorithm, we have tapped the output of C_5 layer and reweighted the same using Eq. 3. The output features are reweighted till the distribution difference between old training and new samples is reduced. In the phase III part of the proposed algorithm, these reweighted features are used to train F_6 and output layer. When the new training set is made available, the proposed algorithm does not disturb the trained CNN layers (C_1, C_3 and C_5). However, the new information is incorporated into the network by modifying weights on F_6 and output layer. This is proposed minimum change principle.

Dataset

To the best of our knowledge, there is no public dataset constructed with a large number of samples for performing experiments on face recognition under transfer subspace

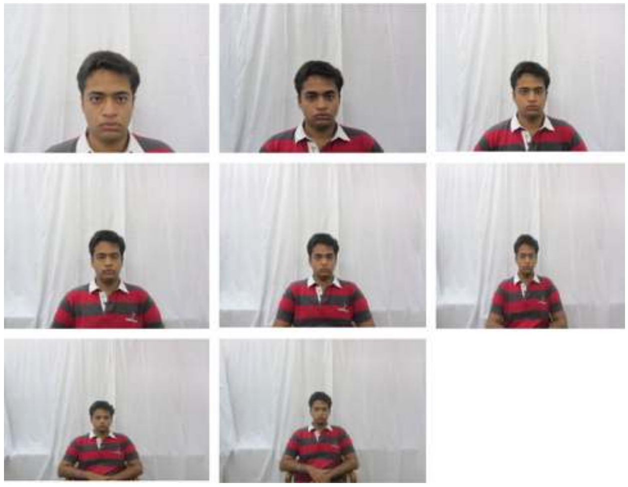


Fig. 3 Sample images from database (scale 1–scale 8)

learning. The distance between subject and camera is varied, and the camera position is shifted while preparing the database. The distance is varied in steps of 15 cm.We refer to distance of 15 cm as scale 1 (S1), 30 cm as scale 2 (S2) and 120 cm as scale 8 (S8). Similarly, shift of 5 cm at S1 as S1sh1, shift of 10 cm at S1 as S1sh2, rotation of 5° at S7 as S7r1 and rotation of 10° as S7r2, etc. Camera positions were shifted by 5 and 10 cm at scale S1 and S2. The camera was rotated with 5° and 10° of inclination at scale S7 and S8. The images were collected in an illumination controlled environment. For maintaining a level of consistency throughout the database, the same physical setup was used in each photography session. Because the equipment had to be reassembled for each session, there was some minor variation in images collected on different dates. The proposed database was collected in 10 sessions between December 2012 and June 2013.

The database contains 20,000 images that include 50 subjects. For every subject 25 images per scale, at four shifts and two angles were taken (total 400 images per subject). The details of the database are shown in Table 1. Figure 3 shows some sample images in database.

Experiments, parameter settings, and observations

Basic CNN architecture was trained using the conventional/traditional algorithm as per the steps discussed in Sect. “Comparison of traditional algorithm (conventional) with proposed algorithm” with the samples from developed database. Table 2 shows the results for 25-user system. Table 3 shows the results for 50-user system. 10 images per user per scale were used for training and 15 images per user per scale were used for testing. It was observed that when the training and testing samples are from the same scale, the classification rate seems to be high as compare to the testing samples from different scales.

To improve the classification rate for samples belonging to cross-scale, we trained the CNN network with Proposed algorithm discussed in Sect. “Proposed three-phase training algorithm for CNN architecture using transfer learning

Table 2 Recognition rates in % for CNN algorithm trained with conventional algorithm for 25 users

Testing images	Training images															
	*S*1	S1sh1	S1sh2	S2	S2sh1	S2sh2	S3	S4	S5	S6	S7	S7r1	S7r2	S8	S8r1	S8r2
S1	**96**	64	20	0	8	8	8	32	76	76	88	88	92	84	88	84
S1sh1	44	**92**	4	20	12	12	16	36	48	52	60	56	36	64	68	36
S1sh2	16	48	**96**	8	12	12	8	28	44	52	56	56	32	56	56	36
S2	4	4	12	**100**	56	48	20	16	24	20	24	28	16	16	12	4
S2sh1	24	16	16	36	**100**	56	20	72	88	84	88	84	76	80	84	68
S2sh2	36	20	16	48	52	**100**	12	44	68	92	88	92	80	88	96	80
S3	12	8	8	8	12	12	**96**	24	88	12	16	8	12	24	20	76
S4	36	20	48	20	28	20	28	**100**	56	44	52	40	48	44	44	40
S5	28	20	28	20	40	24	12	20	**96**	40	8	12	12	0	12	20
S6	12	24	20	24	24	16	0	8	28	**100**	52	40	44	24	20	20
S7	20	24	16	24	20	20	4	4	20	44	**100**	68	48	60	64	40
S7r1	16	20	20	32	28	44	16	4	16	36	68	**100**	64	48	52	32
S7r2	12	16	12	0	4	0	12	8	8	52	60	56	**96**	24	24	44
S8	0	4	8	0	4	4	8	4	8	4	60	32	16	**100**	84	40
S8r1	16	16	20	16	28	24	8	4	4	16	68	44	40	76	**96**	72
S8r2	44	16	12	0	4	0	12	8	8	52	60	56	96	24	24	**100**

Bold values denote the classification rate for same scale training and testing samples

Table 3 Recognition rates in % for CNN algorithm trained with conventional algorithm for 50 users

Testing images	Training images															
	*S*1	S1sh1	S1sh2	S2	S2sh1	S2sh2	S3	S4	S5	S6	S7	S7r1	S7r2	S8	S8r1	S8r2
S1	**92**	22	13	12	8	17	12	8	20	4	41	29	50	30	21	90
S1sh1	28	**94**	47	14	14	12	11	11	30	2	47	44	42	38	36	88
S1sh2	16	26	**89**	13	15	15	9	13	27	3	55	48	46	50	49	85
S2	8	5	5	**96**	38	9	4	4	9	2	69	36	32	48	37	81
S2sh1	4	6	4	24	**88**	38	4	6	11	3	52	24	33	41	30	71
S2sh2	4	6	3	23	62	**69**	2	7	7	4	55	31	29	36	33	73
S3	21	2	28	7	6	6	**91**	5	5	2	18	11	13	12	25	21
S4	22	23	72	6	11	22	8	**88**	9	3	10	2	3	6	8	4
S5	28	28	10	27	24	36	9	7	**97**	24	4	2	11	3	6	2
S6	21	24	14	17	50	58	9	4	33	**97**	29	21	23	12	8	2
S7	9	21	14	13	14	10	18	4	11	31	**97**	46	34	36	29	12
S7r1	26	20	16	15	17	14	24	6	10	24	47	**97**	29	19	37	10
S7r2	21	11	12	18	13	14	19	2	14	41	38	24	**97**	21	20	24
S8	18	17	18	14	12	9	37	5	6	30	40	22	19	**96**	71	20
S8r1	28	10	18	15	13	10	40	7	6	14	31	26	20	59	**94**	25
S8r2	22	11	17	10	12	17	33	3	8	15	16	12	33	31	43	**98**

Bold values denote the classification rate for same scale training and testing samples

Table 4 Recognition rates in % for CNN algorithm trained with proposed three phase algorithm for 25 users:

Testing images	Training images															
	S1	S1sh1	S1sh2	S2	S2sh1	S2sh2	S3	S4	S5	S6	S7	S7r1	S7r2	S8	S8r1	S8r2
S1	**84**	84	84	84	84	84	84	84	84	84	84	84	84	84	84	84
S1sh1	92	**92**	92	92	92	92	92	92	92	92	92	92	92	92	92	92
S1sh2	88	88	**88**	88	88	88	88	88	88	88	88	88	88	88	88	88
S2	80	80	80	**80**	80	80	80	80	80	80	80	80	80	80	80	80
S2sh1	84	84	84	84	**84**	84	84	84	84	84	84	84	84	84	84	84
S2sh2	80	80	80	80	80	**80**	80	80	80	80	80	80	80	80	80	80
S3	82	82	82	82	82	82	**82**	82	82	82	82	82	82	82	82	82
S4	84	84	84	84	84	84	84	**84**	84	84	84	84	84	84	84	84
S5	88	88	88	88	88	88	88	88	**88**	88	88	88	88	88	88	88
S6	80	80	80	80	80	80	80	80	80	**80**	80	80	80	80	80	80
S7	84	84	84	84	84	84	84	84	84	84	**84**	84	84	84	84	84
S7r1	80	80	80	80	80	80	80	80	80	80	80	**80**	80	80	80	80
S7r2	80	80	80	80	80	80	80	80	80	80	80	80	**80**	80	80	80
S8	82	82	82	82	82	82	82	82	82	82	82	82	82	**82**	82	82
S8r1	80	80	80	80	80	80	80	80	80	80	80	80	80	80	**80**	80
S8r2	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	**80**

Bold values denote the classification rate for same scale training and testing samples

approach". We skipped the phase I part of the algorithm as the initial layers are already been trained by traditional algorithm. The aim here is to incorporate new information into the network that is available from new samples. Table 4 shows the results for 25-user system using proposed three-phase training algorithm. Table 5 shows the results of 50-user system using proposed three-phase algorithm. CNN was trained, using traditional algorithm for 3000 epochs with a learning rate of 0.05, 0.5 and 0.8. Best results were obtained with learning rate of 0.5. Figure 4 shows the plot of MSE1 versus iterations.Performance plot of traditional algorithm and the proposed algorithm, for training F_6 and output layer with MSE2 as performance index is been shown in Figs. 5 and 6, respectively. Weight modification phase under transfer learn-

Table 5 Recognition rates in % for CNN algorithm trained with proposed three phase algorithm for 50 users:

Testing images	Training images															
	S1	S1sh1	S1sh2	S2	S2sh1	S2sh2	S3	S4	S5	S6	S7	S7r1	S7r2	S8	S8r1	S8r2
S1	**65**	65	65	65	65	64	65	65	64	64	65	65	65	64	65	63
S1sh1	65	**66**	65	65	65	65	65	65	65	67	66	64	65	65	64	63
S1sh2	64	55	**64**	65	64	68	65	64	67	64	63	65	65	59	64	65
S2	66	66	61	**66**	66	67	65	65	64	67	65	65	66	66	65	65
S2sh1	64	66	66	65	**65**	65	66	65	63	65	64	65	64	65	64	65
S2sh2	66	66	67	66	65	**65**	65	66	65	63	63	62	64	62	64	63
S3	65	65	65	66	65	65	**67**	64	65	67	63	64	65	66	67	63
S4	66	65	65	65	64	67	64	**65**	65	66	64	65	63	65	63	55
S5	65	66	66	64	64	64	66	62	**67**	63	65	63	63	55	64	67
S6	65	65	62	66	65	65	65	66	65	**65**	66	65	66	65	64	64
S7	65	64	63	67	66	64	54	64	65	64	**62**	64	64	64	69	68
S7r1	66	65	66	65	65	66	64	66	65	63	66	**63**	65	64	66	63
S7r2	66	66	63	64	66	65	65	65	62	65	62	63	**64**	64	63	64
S8	68	65	66	64	65	65	65	64	65	59	65	65	64	**66**	64	62
S8r1	65	66	62	67	66	66	67	62	63	62	65	64	63	64	**65**	65
S8r2	59	66	64	68	65	66	65	67	62	64	65	64	66	65	62	**62**

Bold values denote the classification rate for same scale training and testing samples

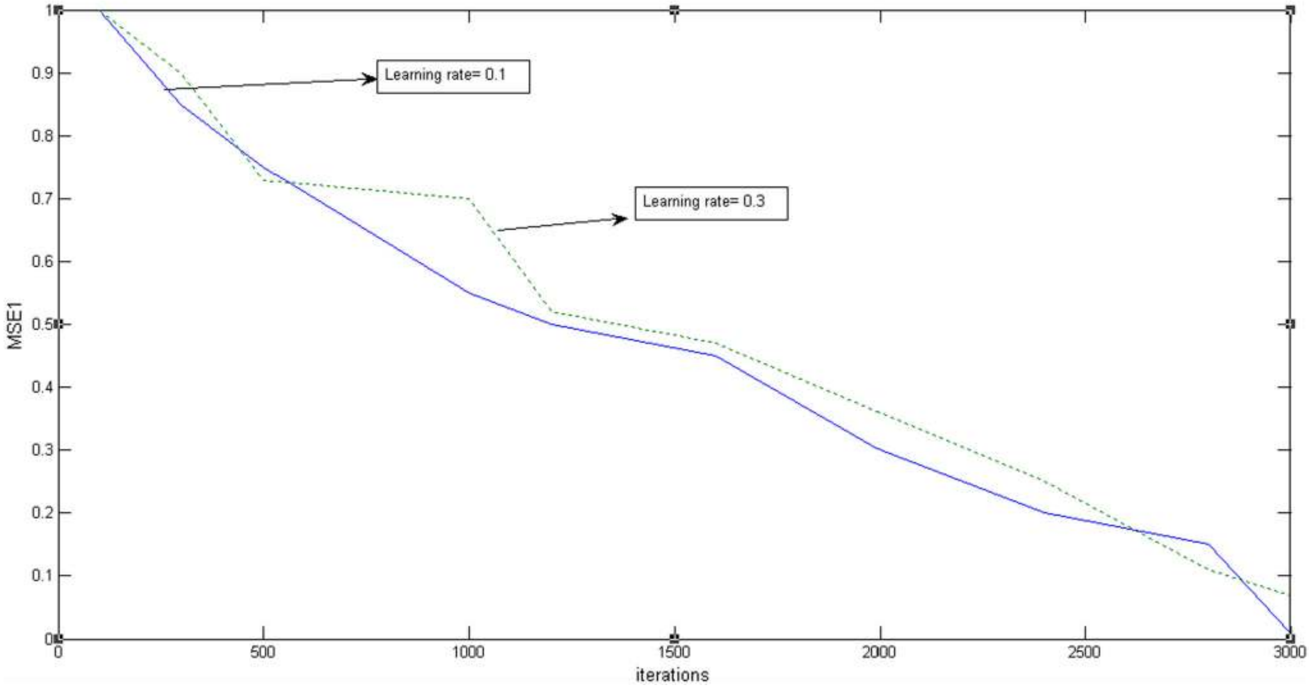


Fig. 4 Performance plot of MSE1 versus iterations (50-user system)

ing with proposed algorithm converges faster compare to traditional algorithms weight modification phase.

As shown in Table 6, basic CNN trained for 25 users has 29,504 trainable parameters. If the network has to incorporate knowledge from new training samples taken at different scales made available subsequently, it has to be retrained again with 29,504 trainable weights. However, the proposed algorithm avoids complete retraining by using minimum change principle, i.e., by updating weights in F_6 and output layer with which a relatively acceptable classification rate can be achieved with 12,180 trainable parameters. Same is true for a 50-user system with only 14,280 trainable weights in transfer learning task. 17,324 weights are not disturbed in the proposed three-phase training algorithm for CNN. For

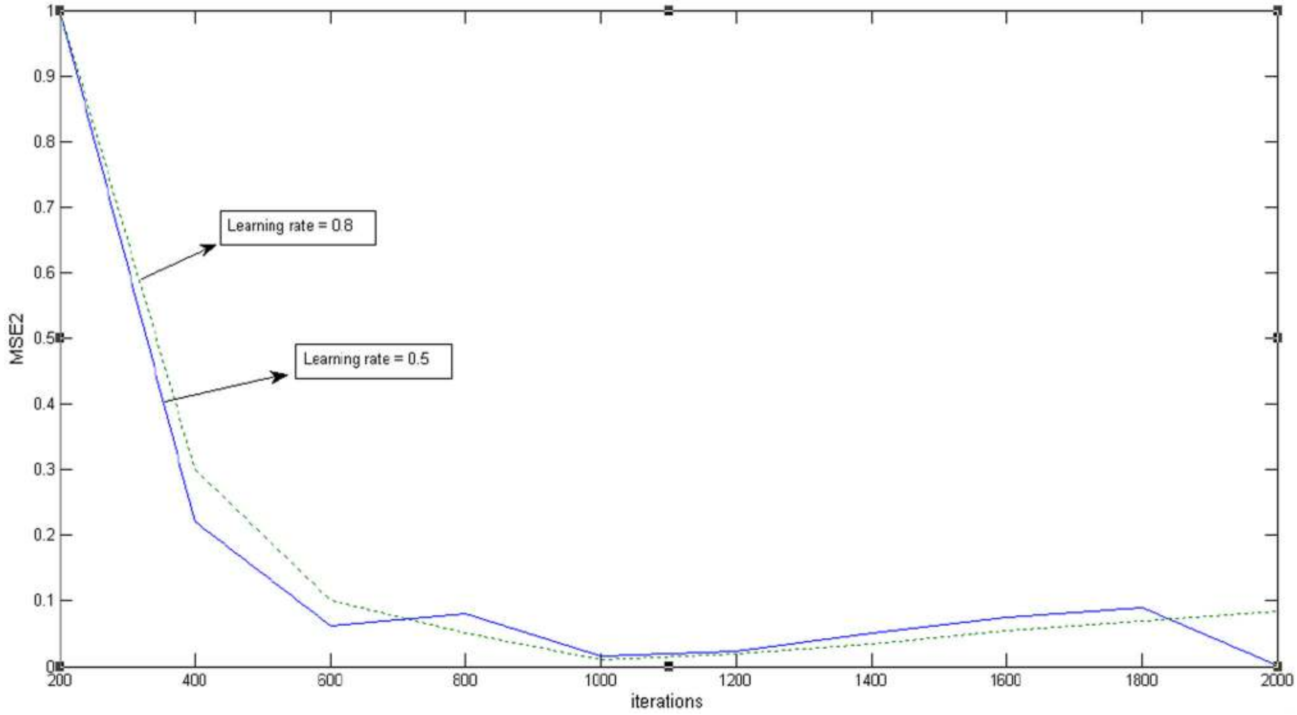


Fig. 5 Performance plot of MSE2 versus iterations for traditional algorithm (50-user system)

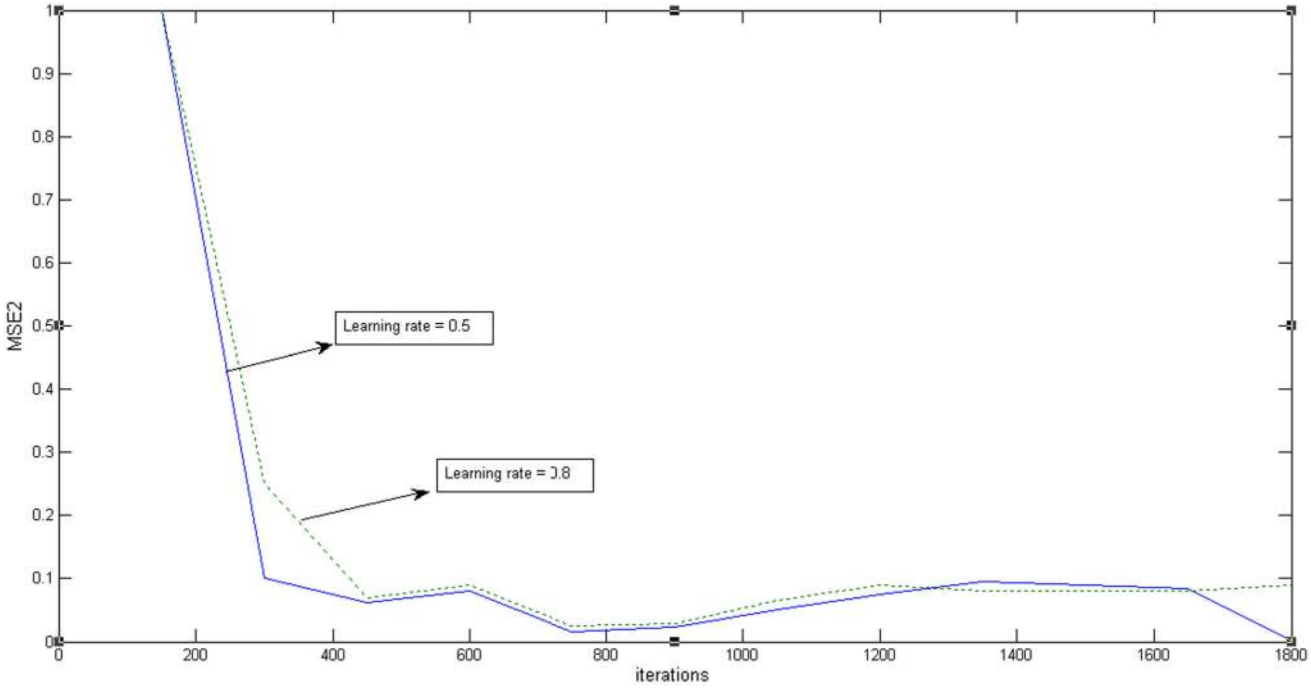


Fig. 6 Performance plot of MSE2 versus iterations for proposed algorithm (50-user system)

a 25-user system, the proposed algorithm gives an average classification rate of 80 percent for all scales in transfer subspace task and an average rate of 60 percent for a 50-user system. Figures 7 and 8 show the comparison of classification rates of conventional CNN algorithm with proposed algorithm when the training samples are from the scale 1. As seen from Tables 4 and 5, the same scale (training and testing with same scales) classification rates tend to drop with

Table 6 Trainable weights in CNN

Algorithm	Various layers of CNN							Total trainable weights
	C_1 trainable weights	S_2 trainable weights	C_3 trainable weights	S_4 trainable weights	C_5 trainable weights	F_6 trainable weights	Output layer trainable weights	
Conventional CNN algorithm (trained for 25 users)	732	–	1952	–	14,640	10,080	2100	29,504
Conventional CNN algorithm (trained for 50 users)	732	–	1952	–	14,640	10,080	4200	31,604
Proposed three-phase training algorithm for CNN (25 users)	–	–	–	–	–	10,080	2100	12,180
Proposed three-phase training algorithm for CNN (50 users)	–	–	–	–	–	10,080	4200	14,280

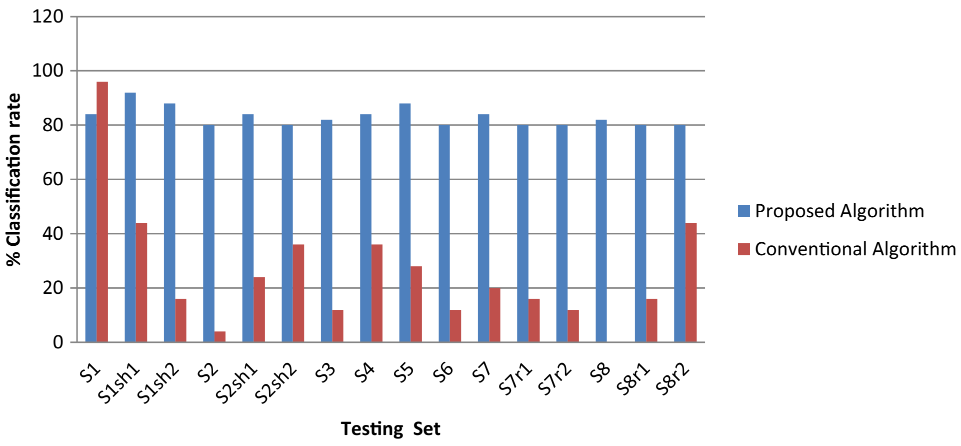


Fig. 7 Comparison of classification rates of conventional CNN algorithm with proposed algorithm when the network was trained with scale S1 samples (25 users)

the proposed algorithm. This implies that there is a negative transfer that hinders the percentage classification rate at same scales. The negative transfer happens if the sources of data are too dissimilar [89].

We experimented CNN with 52 × 52, 60 × 60, 74 × 74 and 84 × 84 input size. The CNN was trained using proposed algorithm with learning rate values of 0.05, 0.5 and 0.8. The best results were obtained with an input size of 74 × 74 and learning rate of 0.5. In Eq. 3 there are two parameters α and ρ. setting higher value of α allows more information to be transferred. However, very high value of α makes most of the elements of O^{ztk} zero which is not suitable for transfer of information. Larger the value of ρ, the distribution between source and target domains will be small. However, very small value of ρ will result in less transfer of information. We heuristically determined the value of α and ρ, by varying α and keeping ρ constant and vice versa. Variation of classification accuracies with different values of α and ρ are shown in Figs. 9 and 10.

Various researchers have tackled different applications of transfer learning on SVM architecture. We have proposed a generic training algorithm which can be used for any deep learning network having feature extraction and classification layer integrated. Also the application of face recognition at a distance is novel. As a result there is no data available in the literature with which the proposed work can be compared.

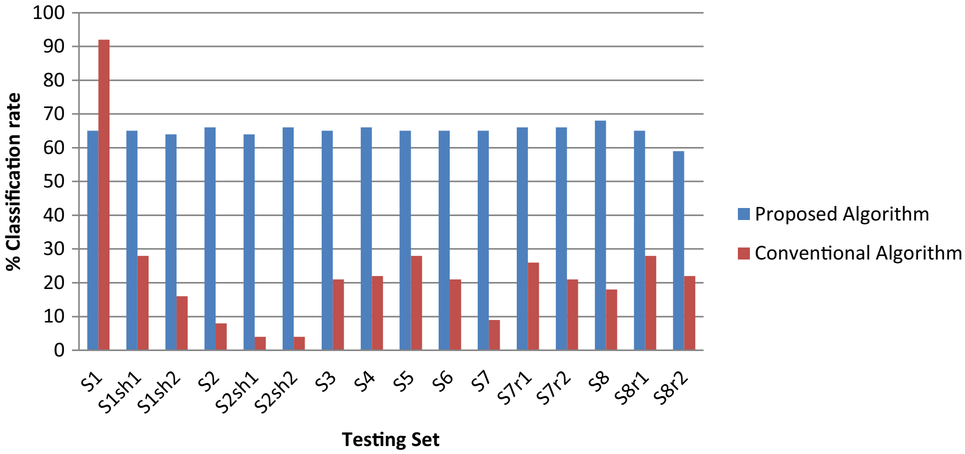


Fig. 8 Comparison of classification rates of conventional CNN algorithm with proposed algorithm when the network was trained with scale S1 samples (50 users)

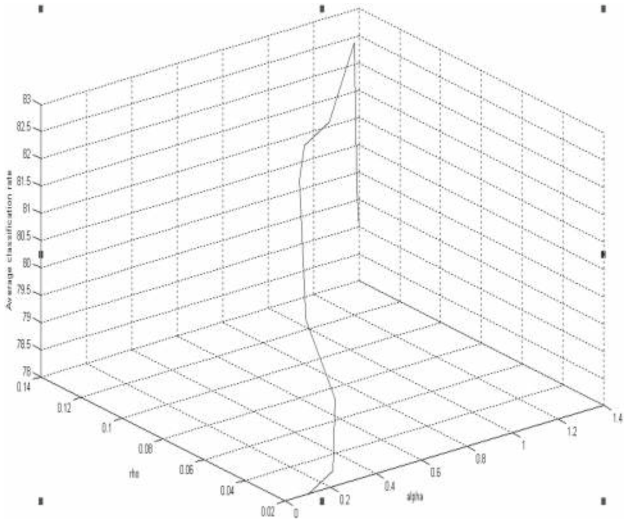


Fig. 9 Variation of classification accuracy as a function of rho (ρ) and alpha (α) (25-user system)

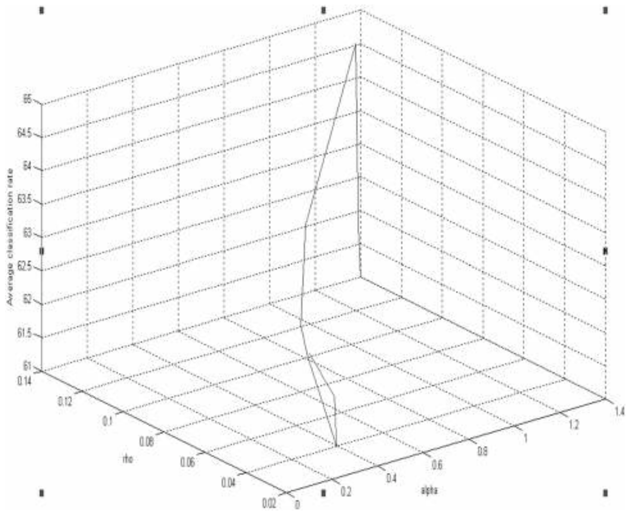


Fig. 10 Variation of classification accuracy as a function of rho (ρ) and alpha (α) (50-user system)

Hence authors of this paper have compared the proposed training algorithm with the existing traditional algorithm of CNN.

Conclusion

We have proposed a novel training algorithm that can be used to train any neural network architecture which is meant for visual pattern recognition. These networks have feature extraction and classification layers integrated into the architecture. In many applications training data are made available subsequently. In this situation neural networks like CNN and neocognitron are to be trained again with the new data. The proposed approach can be used in such situations. In this approach, one can tap the output of the last feature extraction layer and reweight the output in such a way that the distribution difference between the old and new training samples is reduced. We have shown the utility of the algorithm for the CNN architecture. However, the approach is generic and can be used for any neural network architecture which has feature extraction and classification layers integrated into one architecture.

Training time of any neural network increases with the increasing number of samples. If the training samples are not available at one time, then the situation demands retraining. Many machine learning softwares do not have provision to avoid retraining. The proposed algorithm can increase the utility of any machine learning software by giving an user a method with which by doing few disturbances in the trainable parameters transfers the new information into the architecture. By this approach the training time can be reduced under transfer learning task.

We have proposed a novel three-phase training algorithm for CNN under transfer learning that gives a constant average classification rate. With the proposed framework one has to disturb only 60 percent weights in the architecture for incorporating the knowledge available from the new training samples. We proposed minimum change principle, as per that one has to disturb few weights to transfer knowledge. The work may be extended by (1) reducing a negative transfer of knowledge; (2) coming up with information theoretic measure of the information transfer.

References

1. Schurmann J (1978) A multifont word recognition system for postal address reading. IEEE Trans Comput C-27(8):721–732
2. Jain AK, Duin RPW, Mao J (2000) Statistical pattern recognition: a review. IEEE Trans Pattern Anal Mach Intell 22(1):4–33
3. Ghosh D, Dube T, Shivaprasad AP (2010) Script recognition-a review. IEEE Trans Pattern Anal Mach Intell 32(12):2142–2161
4. Arica N, Yarman-Vural FT (2001) An overview of character recognition focussed on off-line handwriting. IEEE Trans Syst Man Cybern Part C 31(2):216–233
5. Oh AS, Lee JS, Suen CY (1999) Analysis of class separation and combination of class-dependent features for handwriting recognition. IEEE Trans Pattern Anal Mach Intell 21(10):1089–1094
6. Bakkre MJ, Rahman MT, Bhuiyan MA (2009) The enhanced face recognition using binary patterns of Gabor features. In: Proc. IEEE TENCON, pp 1–5
7. Al-Mubaidand SH, Umair A (2006) A new text categorization technique using distributional clustering and learning logic. IEEE Trans Knowl Data Eng 18(9):1156–1165
8. Baralis E, Chiusano S, Garza P (2006) A lazy approach to associative classification. IEEE Trans Knowl Data Eng 20(2):156–171
9. Gandomi AH, Roke DA (2015) Assessment of artificial neural network and genetic programming as predictive tools. Adv Eng Softw 88:63–72
10. Pan SJ, Yang Q, Hu DH (2008) Transfer learning for wifi-based indoor localization. In: AAAI 2008 workshop on transfer learning for complex task, Illinois, USA, 13–17 July 2008
11. Sinno P, Qiang Y (2010) A survey on transfer learning. IEEE Trans Knowl Data Eng 22:1345–1359
12. Pan SJ, Yang Q (2010) A survey on transfer learning. IEEE Trans Knowl Data Eng 22(10):1345–1359
13. Hido K, Sugiyama M (2009) A least squares approach to direct importance estimation. J Mach Learn Res 10:1391–1445
14. Glorot X, Bordes A, Bengio Y (2011) Domain adaptation for large scale sentiment classification: a deep learning approach. In: Proceedings of the 28th International conference on Machine Learning. Bellevue, WA, USA, pp 513–520
15. Evgeniou T, Pontil M (2004) Regularized multi task learning. In: Proceedings of the tenth ACM International Conference on Knowledge discovery and Data Mining, pp 109–117. doi:10.1145/1014052.1014067
16. Li F, Pan SJ, Jin O, Yang Q, Zhu X (2012) Cross domain co-extraction of sentiment and topic lexicons. In: ACL '12 proceedings of the 50th Annual Meeting of the Association for computational Linguistics: Long papers, vol 1, pp 410–419
17. Si S, Tao D, Geng B (2010) Bregman divergence-based regularization for transfer subspace learning. IEEE Trans Knowl Data Eng 22(7):929–942
18. Daume MH III, Marcu D (2006) Domain adaptation for statistical classifiers. J Artif Intell Res 26:101–126
19. The CS, Lim CP (2007) An artificial neural network classifier design based on variable kernel and non-parametric density estimation. Springer Neural Process Lett 27:137–151. doi:10.1007/s11063-007-9065-6
20. Wood J (1996) Invariant pattern recognition: a review. Pattern Recognit 29(1):1–17
21. Frank T, Kraiss KF, Kuhlen T (1998) Comparative analysis of fuzzy ART and ART-2A network clustering performance. IEEE Trans Neural Netw 9(3):544–559
22. Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient–based learning applied to document recognition. In: Proc. IEEE, pp 2278–2324
23. Fasel B, Gatica-Perez D (2006) Rotation–invariant neoperceptron. In: Proc. ICPR, vol 3, pp 336–339. doi:10.1109/ICPR.2006.1020
24. Yan QC, Damper RI, Nixon MS (1997) On neural network implementations of k-nearest neighbor pattern classifiers. IEEE Trans Circuits Syst 44(7):622–629
25. Granger E, Connolly JF, Sabourin R (2008) A comparision of fuzzy ARTMAP and gaussian ARTMAP neural networks for incremental learning. In: Proc. IJCNN, pp 3305–3308
26. Yaghini M, Khoshraftar MM, Fallahi M (2013) A hybrid algorithm for artificial neural network training. Eng Appl Artif Intell 26:293–301
27. Tivive FHC, Bouzerdoum A (2005) Efficient training algorithms for a class of shunting inhibitory convolutional neural networks. IEEE Trans Neural Netw 16(3):541–556
28. Neubauer C (1998) Evaluation of Convolutional Neural Networks for Visual Recognition. IEEE Trans. Neural Networks 9(4):685–696
29. Fan J, Xu W, Wu Y, Gong Y (2010) Human tracking using convolutional neural networks. IEEE Trans Neural Netw 21(10):1610–1623
30. Lawrence N, Platt J (2004) Learning to learn with the informative vector machine. In: Proc. Machine Learning. doi:10.1145/1015330.1015382
31. Jie L, Tommsai T, Caputo B (2011) Multiclass tansfer learning from uncontrained priors. In: International Conference on Computer Vision, pp 1863–1870. doi:10.1109/ICCU.2011.6126454
32. Xia S, Shao M, Luo J, Fu Y (2012) Understanding kin relationships in a photo. IEEE Trans Multimed 14(4):1046–1056
33. Bickel S, Bruckner M, Scheffer T (2007) Discriminative learning for differing training and test distributions. In: Proc. Machine Learning, pp 81–88. doi:10.1145/1273496.1273507
34. Lee SI, Chatalbashev V, Vickrey D, Koller D (2007) Learning a meta–level prior for feature relevance from multiple related tasks. In: Proc. Machine Learning, pp 489–496. doi:10.1145/1273496.1273558
35. Yang S, Lin M, Hou C, Zhang C, Wu Y (2012) A general framework for transfer sparse subspace learning. Springer Neural Comput Appl 21:1801–1817. doi:10.1007/s00521-012-1084-1
36. Kelvin Sim,VivekanandGopalkrishnan,ArthurZimek andGao Cong, “A Survey on Enhanced Subspace Clustering,” Springer. Data Min Knowl Disc vol 26 10.1007/s10618 -012-0258-x pp. 332 - 397, February 2012
37. Dai W, Yang Q, Xue GR, Yu Y (2008) Self taught clustering. In: Proc. Machine Learning. doi:10.1145/1390156.1390182

38. Si S, Tao D, Wang M, Chan KP (2010) Social image annotation via cross-domain subspace learning. Springer Multimed Tools Appl 56:91–108. doi:10.1007/s11042-010-0567-2
39. Huy TN, Tong B, Shao H, Suzuki E (2012) Transfer learning by centroid pivoted mapping in noisy environment. Springer J Intell Inf Syst. doi:10.1007/s10844-012-0226-3
40. Tong B, Shao H, Chou BH, Suzuki E (2012) Linear semi-supervised projection clustering by transferred centroid regularization. Springer J Intell Inf Syst 39:461–490. doi:10.1007/s10844-012-0198-3
41. Yin X, Han J, Yu PS (2006) Efficient classification across multiple database relations: a crossmine approach. IEEE Trans Knowl Data Eng 18(6):770–783
42. Van Opbroek A, Ikram MA, Vernooij MW, De Bruijne M (2015) Transfer learning improves supervised image segmentation across imaging protocols. IEEE Trans Med Imaging 34(5):1018–1030
43. Bhatt HS, Singh R, Vatsa M, Ratha NK (2014) Improving cross-resolution face matching using ensemble-based co-transfer learning. IEEE Trans Image Process 23(12):5669
44. Cao X, Wipf D, Wen F, Duan E, Sun J (2013) A practical transfer learning algorithm for Face verification. In: Proc. Int conf comput. Vis, pp 3208–3215. doi:10.1109/ICCU.2013.398.3201
45. Zhao P, Hoi SC (2010) OTL: a framework of onlinetransfer learning. In: Proc. Int. cof. Machine learning, pp 1231–1238. doi:10.1145/2505515.2505603
46. Zhu Y et al (2011) Heterogenous transfer learning for image classification. In: Proc. AAA I conf. Artificial intelligence, pp 1304–1309
47. Geng B, Tao D, Xu C (2011) DAML: domain adaptation metric learning. IEEE Trans Image Process 20(10):2980–2989
48. Deng Z, Choi KS, Jiang Y (2014) Generalized hidden-mapping ridge regression, knowledge-leveraged inductive transfer learing for neural networks, fuzzy systems and kernel methods. IEEE Trans Cybern 44(12):2585–2599
49. Chen XW, Lin X (2014) Big data deep learning: challenges and perspectives. IEE Access 2:514–525
50. Hous W, Gao X, Tao D, Li X (2015) Blind image quality assessment via deep learning. IEEE Trans Neural Netw Learn Syst 26(6):1275–1286
51. Bu S, Liu Z, Han J, Wu J, Ji R (2014) Learning high-level feature by deep belief networks for 3-D model retrieval and recognition. IEE Trans Multimed 16(8):2154–2167
52. Lv Y, Duan Y, Kang W, Li Z, Wang FY (2015) Traffic Flow Prediction With Big Data: A Deep Learning Approach. IEEE Trans Intell Transp Syst 16(2):865–812
53. Ding J, Huang Y, Liu W, Haung K (2016) Severely Blurred object Tracking by learning deep image representations. IEEE Trans Circuits Syst Video Technol 26(2):319–331
54. Hu C, Bai X, Qi L, Chen P, Xue G, Mei L (2015) Vehicle color recoginition with spatial pyramid deep learning. IEEE Trans Intell Transp Syst 16(5):2925–2934
55. Gao X, Lin S, Wong TY (2015) Automatic features learning to grade nuclear "cataracts based on deep learning". IEEE Trans Biomed Eng 162(11):2693–2701
56. Ciresan DC (2013) Mitosis detection in breast cancer histology images with deep neural networks. In: Proc. Int. conf. med. Image comput. Comput. Assisted intervention, pp 411–418
57. Habibzadeh M et al (2013) White blood cell differential counts using convolutional neural network for low resolution images. In: Proc. Int. conf. Med. Image Comput. Comput. Assisted intervention. LNCS, vol 7895, pp 263–274
58. Wu G et al (2013) Unsupervised deep feature learning for deformable registration of MR brain images. In: Proc. Int. conf. Med. Image comput. Comput. Assisted intervevtion. LNCS, vol 8150, pp 649–656
59. Liao S et al (2013) Representation learning: a unified deep learning framework for automatic prostate MR segmentation. In: Proc. Int. conf. Med. Image Comput. Comput. Assisted Intervention. LNCS, vol 8150, pp 254–261
60. Prasoon A et al (2013) Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network. In: Proc. Int. conf. Med. Image Comput. Comput. Assisted intervention. LNCS, vol 8150, pp 246–253
61. Brosch T, Tam R (2013) Manifold learning of brain MRIs by deep learning. In: Proc. Int. conf. Med. Image Comput. Comput. Assisted intervention. LNCS, vol 8150, pp 633–640
62. Vincent P, Larochella H, Lajoie I, Bengio Y, Manzagol PA (2010) Stacked denoising autoencoders:learning useful representations in a deep network with a local denoising criterion. J Mach Learn Res 11:3371–3408
63. Ngiam J et al (2011) Multimodal deep learning. Proc. ICML, Bellevue, WA, USA, pp 689–696
64. Shenghua G, Yuting Z, Kui J, Lu J, Yingying Z (2015) Single sample face recognition via learning deep supervised autoencoders. IEEE Trans Inf Forensics Secur 10(10):2108–2118
65. Kereliuk C, Sturm BL, Larsen J (2015) Deep learning and music adversaries. IEEE Trans Multimed 17(11):2059–2071
66. Luus FPS, Salmon BP, Van den Bergh F, Maharaj BTJ (2015) Multiview deep learning for land use classification. IEEE Geosci Remote Sens Lett 12(12):2448–2452
67. Zhang F, Du B, Zhang L (2016) Scene classification via a gradient boosting randomconvolutional network framework. IEEE Trans Geosci Remote Sens 54(3):1793–1802
68. Hu J, Lu J, Tang YP Deep metric learning for visual tracking. IEEE Trans Circuits Syst VideoTechnol. doi:10.1109/TCSVT.2015.2477936
69. Sigvad O, Droniou A (2016) Towards deep decelopmental learning. Auton. Mental Dev, IEEE Trans. doi:10.1109/TAMD.2015.2496248
70. Wu F, Wang Z, Zhang Z, Yang Y, Luo J, Zhu W, Zhuang Y (2015) Weakly semi-supervised deep learning for multi-label image annotation. IEEE Trans Big Data 1(3):109–122
71. Deng Y, Bao F, Kong Y, Ren Z, Dai Q (2016) Deep direct reinforcement learning for financial signal representation and trading. IEEE Trans Neural Netw Learn Syst. doi:10.1109/TNNLS.2016.2522401
72. Ghesu FC, Krubasik E, Georgescu B, Singh V, Zheng Y, Hornegger J, Comaniciu D (2016) Marginal space deep learning: efficient architecture for volumetric image parsing. IEEE Tans Med Imaging 35(5):1217–1228
73. Wang X, Gao L, Mao S, Pandey S (2016) CSI-based Finger printing for indoor localization: a deep learning approach. IEEE Trans Vehicular Technol. arXiv:1603.07080v1
74. He Y, Xiang S, Kang C, Wang J, Pan C Cross-model retrievalvia deep and bidirectional representation learning. IEEE Trans Multimed. doi:10.11.1109/Tmm.2016.2558463
75. Krizhevsky A, Sutskever I, Hinton G (2012) ImageNet classification with deep Convolutional neural networks. In: Advances in Neural Information Processing Systems, pp 1106–1114
76. Tompson J, Jain A, LeCun Y, Bregler C (2014) Joint training of a convolutional network and a graphical model for human pose estimation. In: Advances in Neural Information Processing Systems, pp 1799–1807. arXiv:1406.2984v2
77. Sun Y, Chen Y, Wang X, Tang X (2014) Deep learning face representation by joint identification-verification. In: Advances in Neural Information Processing Systems, pp 1988–1996. arXiv:1406.4773v1
78. Kim Y (2014) Convolution neural networks for sentence classification. In: Conference on Empirical Methods in Natural Language Processing, pp 1746–1751. arXiv:1406.5882 [CS.CL]
79. Blunsom P, Grefenstette E, Kalchbrenner N (2014) A convolutional neural network for modelling sentences. In: Annual Meeting of the Association for computational Linguistics, pp 1–8

80. Dos Santos C, Gatti M (2014) Deep convolutional neural networks for sentiment analysis of short texts. In: Interntional conference on Computational Linguistics, Dublin, Ireland, 23–29 August 2014, pp 69–78
81. Johnson R, Zhang T (2014) Effective use of word order for text categorization with convolutional neural networks. arXiv:1412.1058 (preprint)
82. Zhang X, LeCun Y (2015) Text understanding from scratch. arXiv:1502.01710 (preprint)
83. Hu B, Lu Z, Li H, Chen Q (2014) Convolutional neural network architectures for matching natural language sentences. In: Advances in Neural information processing systems, pp 1367–1375
84. Lu Z, Li H (2013) A deep architectures for matching short texts. In: Advances in Neural information processing systems, pp 1367–1375
85. Long M, Wang J, Cao Y, Sun J, Yu PS (2016) Deep Learning of transferable representation for scalable Domain Adaptation. IEEE Trans Knowl Data Eng 99. doi:10.1109/TKDE.2016.2554549
86. Fung GPC, Yu JX, Lu H, Yu PS (2006) Text classification without negative examples revisit. IEEE Trans Knowl Data Eng 18(1):6–20
87. Adini Y, Moses Y, Ullman S (1997) FaceRecognition: the problem of compensating for changes in illumination direction. IEEE Trans Pattern Anal Mach Intell 19(7):721–732
88. Hotelling H (1933) Analysis of a complex of statistical variables into principal components. J Educ Psychol 24:417–441
89. Dyer KB, Capo R, Poilkar R (2014) COMPOSE: a semisupervised learning framework for initially labeled nonstationary streaming data. IEEE Trans Neural Netw Learn Syst 25:12–26