

SEVEN: Deep Semi-supervised Verification Networks

Vahid Noroozi*, Lei Zheng*, Sara Bahaadini†, Sihong Xie**, Philip S. Yu*

*Department of Computer Science, University of Illinois at Chicago, IL, USA

†Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL, USA

** Computer Science and Engineering Department, Lehigh University, Bethlehem, PA, USA

{vnoroo2,lzheng21}@uic.edu, sara.bahaadini@u.northwestern.edu, sxie@cse.lehigh.edu, psyu@uic.edu

Abstract

Verification determines whether two samples belong to the same class or not, and has important applications such as face and fingerprint verification, where thousands or millions of categories are present but each category has scarce labeled examples, presenting two major challenges for existing deep learning models. We propose a deep semi-supervised model named SEmi-supervised VErification Network (SEVEN) to address these challenges. The model consists of two complementary components. The generative component addresses the lack of supervision within each category by learning general salient structures from a large amount of data across categories. The discriminative component exploits the learned general features to mitigate the lack of supervision within categories, and also directs the generative component to find more informative structures of the whole data manifold. The two components are tied together in SEVEN to allow an end-to-end training of the two components. Extensive experiments on four verification tasks demonstrate that SEVEN significantly outperforms other state-of-the-art deep semi-supervised techniques when labeled data are in short supply. Furthermore, SEVEN is competitive with fully supervised baselines trained with a larger amount of labeled data. It indicates the importance of the generative component in SEVEN.

1 Introduction

Different from traditional classification tasks, the goal of verification tasks is to determine whether two samples belong to the same class or not, without predicting the class directly [Chopra *et al.*, 2005]. Verification tasks arise from applications where thousands or millions of classes are present with very few samples within each category (in some cases just one). For example, in face and signature verification, faces and signatures of a person are considered to belong to a class. While there can be millions of persons in the database, very few examples for each person are available. In such applications, it is also necessary to handle new classes without

the need to train the model from the scratch. It is not trivial to address such challenges with traditional classification techniques.

Motivated by the impressive performance brought by deep networks to many machine learning tasks [LeCun *et al.*, 2015; Bahaadini *et al.*, 2017; Zheng *et al.*, 2017], we pursue a deep learning model to improve existing verification models. However, deep networks require a large amount of labeled data for each class, which are not readily available in verification. There are semi-supervised training methods for deep network to tap on the large amount of unlabeled data. These semi-supervised methods usually have separate learning stages [Sun *et al.*, 2017; Nair and Hinton, 2010]. They first pre-train a model using unlabeled data and then fine-tune the model with labeled data to fit the target tasks. Such two-phase methods are not suitable for verification. First, the large number of classes and the lack of data (be it labeled or unlabeled) within each category prohibit us from any form of within class pre-training and fine-tuning. Second, if we pool data from all categories for pre-training, the learned features are general but not specific towards each category, and the later fine-tuning within each category may not be able to correct such bias due to the lack of labeled data.

To address such challenges, we propose Deep SEmi-supervised VErification Networks (SEVEN) that consists of a generative and a discriminative component to learn general and category specific representations from both unlabeled and labeled data simultaneously. We cross the category barrier and pool unlabeled data from all categories to learn salient structures of the data manifold. The hope is that by tapping on the large amount of unlabeled data, the structures that are shared by all categories can be learned for verification.

SEVEN then adapts the general structures to each category by attaching the generative component to the discriminative component that uses the labeled data to learn category-specific features. In this sense, the generative component works as a regularizer for the discriminative component, and aids in exploiting the information hidden in the unlabeled data. On the other hand, as the discriminative component depends on the structures learned by the generative component, it is desirable to inform the generative component about the subspace that is beneficial to the final verification tasks. Towards this end, instead of training the two components separately or sequentially, SEVEN chooses to train the two com-

ponents simultaneously and allow the generative component to learn more informative general features.

We evaluate SEVEN on four datasets and compare it to four state-of-the-art semi-supervised and supervised algorithms. Experimental results demonstrate that SEVEN outperforms all the baselines in terms of accuracy. Furthermore, it has shown that by using very small amount of labeled examples, SEVEN reaches competitive performance with the supervised baselines trained on a significantly larger set of labeled data.

The rest of this paper is organized as follows. In Section 2 we give an overview of the related works. In Section 3 we present SEVEN in detail. Section 4 gives the experimental evaluation and analysis of the proposed model, followed by a conclusion.

2 Related Work

SEVEN can serve as a metric learning algorithm that is commonly employed in verification. The goal of metric learning is to learn a distance metric such that samples in any negative pair are far away and those in any positive pair are close. Many of the existing approaches [Sun *et al.*, 2017; Oh Song *et al.*, 2016; Zagoruyko and Komodakis, 2015; Hu *et al.*, 2014] learn a linear or nonlinear transformation that maps the data to a new space where the distance metric satisfies the above requirements. However, these methods do not address the large number of categories with scarce supervision information.

One of the earliest works in neural network-based verification is proposed by Bromley *et al.* for signature verification [Bromley *et al.*, 1993]. The proposed architecture, named Siamese networks, uses a contrastive objective function to learn a distance metric with Convnets. Similar approaches are employed for many other tasks such as face verification or re-identification [Koch, 2015; Sun *et al.*, 2014; Wolf, 2014]. It is worthy to mention that all these works are supervised and do not exploit unlabeled data.

Great interest in deep semi-supervised learning has emerged in applications where unlabeled data are abundant but obtaining labeled data is expensive or not feasible [Li *et al.*, 2016; Hoffer and Ailon, 2016; Rasmus *et al.*, 2015; Kingma *et al.*, 2014; Lee, 2013]. However, most of such approaches are designed for classification. To the best of our knowledge, there exists no deep semi-supervised learning to address the above two challenges in verification.

A key difference between SEVEN and most of the previous semi-supervised deep networks lies in the way that unlabeled and labeled data are exploited. Lee [Lee, 2013] has presented a semi-supervised approach for classification tasks called Pseudo-Label based on self-training scheme. It predicts the labels of unlabeled samples by training the model with the available labeled samples. Then they bootstrap the model with the highly confident labeled samples. This approach is prone to error because it may reinforce wrong predictions especially in problems with low confident estimation.

A more common semi-supervised approach is to pre-train a model with unlabeled samples and then the learned model is fine-tuned using the labeled samples. For example, [Nair

and Hinton, 2010] have pre-trained a Restricted Boltzmann Machine (RBM) with noisy rectified linear units (NReLU) in the hidden layers, then they used the learned weights to initialize and train a siamese network [Chopra *et al.*, 2005] in a supervised way. The problem with pre-training based approaches is that the supervised part of the algorithm can ignore or lose what the model has learned in the unsupervised step. Another problem with pre-training based approaches is that they still need enough labeled examples for the fine-tuning step.

Recently, some works have tried to alleviate such problems by performing the learning process from all the labeled and unlabeled data in a joint manner for *classification* tasks [Li *et al.*, 2016; Hoffer and Ailon, 2016; Maaløe *et al.*, 2016; Rasmus *et al.*, 2015]. They make the unsupervised model involved in the learning as a regularizer for the supervised model. It should be considered that all such techniques are designed for classification tasks and can not handle the cases mentioned in the introduction such as the few samples per each class and the high number of classes.

Another line of work that handles a large number of categories is extreme multi-label learning [Xie and Philip, 2017]. The most popular assumption is that all classes have sufficient amount of labeled data, and this is clearly different from our problem setting. Recently, there are methods focusing on predicting the tail labels [Jain *et al.*, 2016], but they are proposed for traditional classification task and can not handle new classes in the test data.

3 Proposed Model

3.1 Problem Formulation

The training set is represented as $\mathcal{X} = \{(x_1^i, x_2^i)\}_{i=1}^N$, where (x_1^i, x_2^i) is a pair of training samples $x_j^i \in \mathbb{R}^m$, $N = L + U$ is the total number of training pairs consisting of L labeled and U unlabeled pairs. The relation, i.e., label set denoted by $\mathcal{Y} = \{y^i | y^i \in \{pos, neg\}\}_{i=1}^L$ specifies the relation between the samples of each pair. A positive relation indicates that two samples of the pair belong to the same class and a negative relation indicates the opposite. The relations for the unlabeled pairs are unknown.

Our goal is to learn a nonlinear function $r_{\theta_e}(x_1, x_2) : \mathbb{R}^m \times \mathbb{R}^m \rightarrow \{pos, neg\}$ parameterized by θ_e that predicts the relation between the two data samples x_1 and x_2 . In other words, function $r_{\theta_e}(x_1, x_2)$ verifies if two samples are similar or not.

We define $r_{\theta_e}(\cdot, \cdot)$ based on the distance of x_1 and x_2 estimated by a metric distance function as:

$$r_{\theta_e}(x_1, x_2) = \begin{cases} neg & \text{if } d_{\theta_e}(x_1, x_2) > \tau \\ pos & \text{if } d_{\theta_e}(x_1, x_2) \leq \tau \end{cases} \quad (1)$$

where $d_{\theta_e}(\cdot, \cdot)$ is the metric distance function and threshold τ specifies the maximum distance that samples of a class are allowed to have. We define a nonlinear embedding function $f_{\theta_e}(\cdot)$ that projects data to a new feature space and $d_{\theta_e}(x_1, x_2) = \|f_{\theta_e}(x_1) - f_{\theta_e}(x_2)\|_2$ is the Euclidean distance between x_1 and x_2 in the new space. An arbitrary distance function can be also used instead of the Euclidean distance.

3.2 Model Description

Our proposed model consists of discriminative and generative components. The model learns a non-linear function for each component. For the discriminative component, the nonlinear embedding function $f_{\theta_e}(\cdot)$ is learned to yield “discriminative” and “informative” representation. In a discriminative feature space, similar samples are mapped close to each other while dissimilar pairs are far from each other. Such property is crucial for a good metric function. The generative component of the model is designed to exploit the information hidden in the unlabeled data. The desired representation should keep the salient structures shared by all categories as much as possible. We define a probabilistic framework of the problem along with the discriminative and generative modelings of our algorithm.

The conditional probability distribution of the relation variable y given the i^{th} pair can be estimated as:

$$p(y^i|x_1^i, x_2^i) = 1 - \tanh(d_{\theta_e}(x_1, x_2)) \quad (2)$$

which can be written as the following.

$$p(y^i|x_1^i, x_2^i) = \frac{2}{1 + \exp(2d_{\theta_e}(x_1, x_2))} \quad (3)$$

Here we use a \tanh function to map the distance between samples to $[0, 1]$. However, any monotonic increasing function $u(\cdot)$ which gives $u(0) = 1$ and $u(\infty) = 0$ can be also used for this purpose.

We define \tilde{p} as the ground truth distribution to be approximated by p in (3) as $\tilde{p}(y^i|x_1^i, x_2^i) = 1$ if $y_i = \text{pos}$, and $\tilde{p}(y^i|x_1^i, x_2^i) = 0$ otherwise. In the rest of the paper, the conditional distributions $p(y^i|x_1^i, x_2^i)$ and $\tilde{p}(y^i|x_1^i, x_2^i)$ are denoted by p_i and \tilde{p}_i , respectively. Due to the probabilistic nature of such distributions, we approximate \tilde{p} with p by minimizing the Kullback-Leibler divergence between them and introduce the following discriminative loss function $\mathcal{L}_{\mathcal{D}}(\mathcal{X}, \mathcal{Y}; \theta_e)$ defined over all the labeled pairs as:

$$\mathcal{L}_{\mathcal{D}}(\mathcal{X}, \mathcal{Y}; \theta_e) = \sum_{i=1}^L l_d(x_1^i, x_2^i; \theta_e) = \sum_{i=1}^L KL(\tilde{p}_i || p_i) \quad (4)$$

where $l_d(x_1^i, x_2^i; \theta_e)$ denotes the discriminative loss for the i^{th} pair, and $KL(\tilde{p}_i || p_i)$ denotes the KL-divergence between \tilde{p}_i and p_i . $KL(\tilde{p}_i || p_i)$ can be substituted by $H(p_i, \tilde{p}_i) - H(\tilde{p}_i)$ where $H(p_i)$ specifies the entropy of p_i , and $H(p_i, \tilde{p}_i)$ defines the cross entropy between p_i and \tilde{p}_i . Considering that the loss function is optimized with a gradient based optimization approach and $H(\tilde{p}_i)$ is a constant with respect to the parameters, we simplify the discriminative loss function as:

$$l_d(x_1^i, x_2^i; \theta_e) = -I\{y_i = \text{pos}\} \log(p_i) - I\{y_i = \text{neg}\} \log(1 - p_i) \quad (5)$$

where $I\{\cdot\}$ is the identity function. The loss function becomes equivalent to the cross entropy over p_i and \tilde{p}_i . It penalizes large distance (similarity) between samples from the same (different) class to make the new space discriminative. $\mathcal{L}_{\mathcal{D}}$ attains its minimum when $p_i = \tilde{p}_i$ over all the labeled pairs.

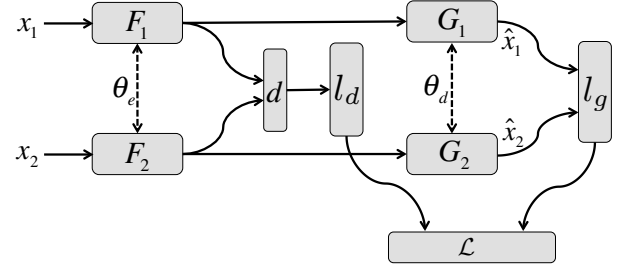


Figure 1: The overall architecture of SEVEN.

To alleviate the insufficiency of the unlabeled data for verification task, through generative modeling, we encourage the embedding function $f_{\theta_e}(\cdot)$ to learn the salient structures shared by all categories. We define a nonlinear function $g_{\theta_d}(\cdot)$ parametrized by θ_d to project back the samples from new representation obtained from $f_{\theta_e}(\cdot)$ to the original feature space.

The generative loss for the i^{th} pair (x_1^i, x_2^i) is defined as the reconstruction error between the original input and the corresponding reconstructed output as:

$$j_g(x_1^i, x_2^i; \theta_e, \theta_d) = \|g_{\theta_d}(f_{\theta_e}(x_1^i)) - x_1^i\|_2 + \|g_{\theta_d}(f_{\theta_e}(x_2^i)) - x_2^i\|_2 \quad (6)$$

where $g_{\theta_d}(f_{\theta_e}(x_j^i))$ indicates the reconstruction of the input of the x_j^i and is denoted by \hat{x}_j^i . The generative loss function $\mathcal{L}_{\mathcal{G}}$, over all pairs including labeled and unlabeled, is defined as:

$$\mathcal{L}_{\mathcal{G}}(\mathcal{X}, \mathcal{Y}; \theta_e, \theta_d) = \sum_{i=1}^{L+U} l_g(x_1^i, x_2^i; \theta_e, \theta_d) \quad (7)$$

We combine the generative and discriminative components into a unified objective function and write the optimization problem of SEVEN as:

$$\mathcal{L}(\mathcal{X}, \mathcal{Y}; \theta_e, \theta_d) = \sum_{i=1}^L l_d(x_1^i, x_2^i; \theta_e) + \alpha \sum_{i=1}^{L+U} l_g(x_1^i, x_2^i; \theta_e, \theta_d) + \beta (\|\theta_e\|_2 + \|\theta_d\|_2) \quad (8)$$

where $\|\theta_e\|$ and $\|\theta_d\|$ are the regularization terms on the parameters of the functions $f_{\theta_e}(\cdot)$ and $g_{\theta_d}(\cdot)$. The parameter β controls the effect of this regularization, parameter α controls the trade off between the discriminative and generative objectives.

3.3 Model Architecture and Optimization

We choose deep neural networks for parameterizing $f_{\theta_e}(\cdot)$ and $g_{\theta_d}(\cdot)$. The schematic representation of SEVEN is illustrated in Figure 1. The input pair is given to two neural networks denoted by F_1 and F_2 with shared parameters θ_e . They represent the discriminative component of the SEVEN (nonlinear embedding function $f_{\theta_e}(\cdot)$). They project the input samples to the discriminative feature space. A layer, denoted

by d , is added on top of the networks F_1 and F_2 that estimates the distance between the two samples of the input pair in the discriminative space.

It can be considered as the metric distance function $d_{\theta_e}(\cdot, \cdot)$ which networks F_1 and F_2 are supposed to learn. The final layers of F_1 and F_2 are connected to two other subnetworks denoted by G_1 and G_2 in Figure 1 with shared parameters θ_d . They model the generative component of SEVEN ($g_{\theta_d}(\cdot)$). They project back the samples to the original space. In other words, they can be considered as decoders for the encoders F_1 and F_2 . The outputs of G_1 and G_2 shown as \hat{x}_1 and \hat{x}_2 are the reconstructions of the corresponding inputs x_1 and x_2 .

Subnetworks F_1 and F_2 are ConvNets built with convolutional and max-pooling layers. G_1 and G_2 are made with transposed convolutional and upsampling layers which perform the reverse operations of convolutional and max-pooling layers, respectively. More detail of the transposed convolutional layer can be found in [Dumoulin and Visin, 2016]. The complete specifications of the models are presented in Table 1.

The whole model is trained using backpropagation with respect to the objective function in Equation 8. Given a set of N pairs, we optimize the model through an adaptive version of gradient descent called RMSprop [Dauphin *et al.*, 2015] over shuffled mini-batches.

We employ l_2 -regularization and dropout [Srivastava *et al.*, 2014] strategy to the convolutional and fully connected layers of the subnetworks to prevent overfitting. Batch normalization [Ioffe and Szegedy, 2015] technique is also applied after each convolutional layer to normalize the output of each layer. It can improve the performance in some cases. The training procedure of SEVEN is illustrated in Algorithm 1.

Algorithm 1: Training procedure of SEVEN

Input: Training set: $\mathcal{X} = \{(x_1^i, x_2^i)\}_{i=1}^N$, label set $\mathcal{Y} = \{y^i\}_{i=1}^L$, number of iterations T , and batch size m .

Output: Model’s parameters: Θ

$B = \lfloor \frac{|\mathcal{X}|}{m} \rfloor$; // number of batches

Randomly split the training set X into B batches;

for $t = 1, 2, \dots, T$ **do**

for $b = 1, 2, \dots, B$ **do**

Feedforward propagation of the b^{th} batch;

Calculate \mathcal{L}^b according to Equation (8);

Estimate gradients $\frac{\partial \mathcal{L}^b}{\partial \Theta_t}$ by back propagation;

Calculate Θ_{t+1} using RMSProp;

end

end

return Θ_T ;

4 Experiments

4.1 Datasets

We evaluate the proposed algorithm on the following four datasets.

MNIST: It is a dataset of 70000 grayscale images of handwritten digits from 0 to 9. We use the original split of 60000/10000 for the training and test sets. A uniform random noise of $[0, 1]$ is added to each pixel to make it noisy and more challenging.

US Postal Service (USPS) [Hull, 1994]: It is a dataset of 9298 handwritten digits automatically scanned from envelopes by the US Postal Service. All images are normalized to 16×16 grayscale images. We selected randomly 85% of the images for the training set.

Labeled Faces in the Wild (LFW) [Huang *et al.*, 2012]: It is a database of face images that contains 1100 positive and 1100 negative pairs in the training set, and 500 positive and 500 negative pairs in the test set. All images are resized to 64×48 .

BiosecurID-SONOF (SONOF) [Galbally *et al.*, 2015]: We use a subset of this dataset comprising signatures collected from 132 users, each user has 16 signatures. Signature images are normalized and converted to 80×80 grayscale images. We divided the users randomly into 100/32 for the training and test purposes.

In SONOF and LFW datasets, classes in the training and test samples are disjoint, while in MNIST and USPS classes are common between test and train sets. The samples of LFW are already in the form of pairs. For other datasets, we create the pairs by first splitting samples into two distinct sets for the training and test. We split the train set randomly into labeled and unlabeled samples. Then, each sample gets paired with two other samples randomly. One sample is selected from the same class to form a positive pair, and another one from a different class to form a negative pair.

4.2 Baselines

We compare the performance of SEVEN with the following baselines. It should be considered that we can not compare SEVEN with classification techniques because they are not usually designed to handle new classes in the test data which happens in verification applications. Since there are no other deep semi-supervised works for verification tasks, we adopt the common deep semi-supervised techniques to verification networks as our baselines.

Discriminative Deep Metric Learning (DDML) [Hu *et al.*, 2014]: They developed a deep neural network that learns a set of hierarchical transformations to project pairs into a common space by using a contrastive loss function. It is a supervised approach and can not use unlabeled data.

Pseudo-Label [Lee, 2013]: It is a semi-supervised approach for training deep neural networks. It initially trains a supervised model with the labeled samples. Then it labels the unlabeled samples with the current trained model before each iteration, and use the high confidence ones along with the labeled samples for training in the next iteration. We followed the same approach for training a siamese network [Bell and Bala, 2015] to extend their approach to the verification tasks.

Convolutional Autoencoder + Siamese Network (Pre-ConvSia): We pre-train a siamese network [Bell and Bala, 2015; Chopra *et al.*, 2005] with an convolutional autoencoder model [Masci *et al.*, 2011]. Then we fine-tune the network

Table 1: Models specifications for different datasets. BN: batch normalization, ReLu: rectified linear activation function, Conv: convolutional layer, TransConv: transposed convolutional layer, Upsampling: Upsampling layer, Dense Layer: fully connected layer, and Max-pooling: max-pooling layer.

MNIST and USPS	
Network F_i	Network G_i
MNIST: 28×28	Input 128×1 Dense Layer ReLu Reshape layer 2×2 Upsampling 5×5 TransConv (8) ReLu Dropout(0.5) 2×2 Upsampling 3×3 TransConv (1) Sigmoid Dropout(0.5)
USPS: 16×16	
3×3 Conv (8)	
ReLu	
2×2 Max-pooling	
Dropout(0.5)	
5×5 Conv (8)	
ReLu	
2×2 Max-pooling	
Dropout(0.5)	
Dense Layer (128×1)	
ReLu	
LFW and SONOF	
Network F_i	Network G_i
LFW: 64×48	Input 128×1 Dense Layer ReLu Reshape layer 3×3 TransConv (64) BN-ReLu Dropout(0.5) 2×2 Upsampling 3×3 TransConv (32) BN-ReLu Dropout(0.5) 2×2 Upsampling 3×3 TransConv (1) BN-Sigmoid Dropout(0.5)
SONOF: 100×100	
4×4 Conv (32)	
BN-ReLu	
Dropout(0.5)	
2×2 Max-pooling	
Dropout(0.5)	
3×3 Conv (64)	
BN-ReLu	
2×2 Max-pooling	
Dropout(0.5)	
3×3 Conv (128)	
BN-ReLu	
Dropout(0.5)	
Dense Layer (128×1)	
ReLu	

with labeled pairs. The network uses ConvNets as the underlying network for the modeling.

Autoencoder + Siamese Network (PreAutoSia): It is similar to PreConvSia, but uses MLP as the underlying network for the modeling. It is significantly faster in training compared to PreConvSia.

Principle Component Analysis (PCA): We use PCA as an unsupervised feature learning technique. The distance between samples in the new space learned by PCA indicates their relations. The threshold on the distance is selected for each dataset separately based on the performance on the training data.

4.3 Experimental Settings

The architectures of SEVEN for all datasets are presented in Table 1. All the parameters of SEVEN and also other baselines are selected based on a validation on a randomly selected 20% subset of the training data. The l_2 -regularization

parameter β is selected from $\{1e - 4, 1e - 3, 0.01, 0.1\}$ for each dataset separately. The parameter α that controls the trade-off between generative and discriminative objectives is selected from $\{0.01, 0.05, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0\}$. It is set to 0.05, 0.1, 0.05, and 0.2 for MNIST, LFW, USPS and SONOF, respectively. Parameter τ is set to 0.5 for all the four datasets.

All the neural network models are trained for 150 epochs. The pre-training is also performed for 150 epochs for the baselines which require pre-training. RMSProp optimizer is used for the training of all the neural networks with the default value $\lambda = 0.001$ recommended in the original paper.

4.4 Performance Evaluation

We report the performance in terms of accuracy which is the number of pairs in the test set verified correctly divided by the total number of pairs in the test set. The performance of SEVEN and all baselines are presented in Tables 2 and 3. The results are reported for different number of labeled pairs and the best accuracy for each case is depicted in bold. The last column indicates the case where all labeled training pairs are used. PCA is a fully unsupervised method, thus one performance is reported for each dataset.

As it can be seen from the tables, SEVEN outperforms other baselines in cases where a limited number of labeled pairs are used and the differences in performance are more significant where the number of labeled pairs is lower, and thus SEVEN can address the scarcity of labeled data better.

Algorithm DDML can give good performance when we have enough labeled data but its performance is significantly lower compared to SEVEN in cases with few labeled samples. DDML does not use the unlabeled data while other baselines benefit from the information hidden in the unlabeled data. By increasing the number of labeled pairs, the difference in accuracy decreases.

SEVEN outperforms all the semi-supervised baselines. One of the main advantages of SEVEN over other semi-supervised methods is that they perform supervised step after pre-training with unlabeled data is finished. This may cancel out some of the learned information from unlabeled data through a supervised process. There is no guarantee that the supervised process can benefit from the unsupervised learning [Rasmus *et al.*, 2015]. Among the semi-supervised baselines, Pseudo-Label not only gives worse results compared to SEVEN, but also it shows lower performance than PreConvSia and PreAutoSia in many cases. It can be related to the noise and error in estimating the labels for unlabeled pairs.

4.5 Model Analysis

We perform some experiments to analyze the effect of the different components of SEVEN. The performances of different variants of SEVEN are given in Table 4. The number of labeled pairs for each dataset is indicated in front of the name of the dataset. DisSEVEN indicates SEVEN with $\alpha = 0$ in Eq. (8) which disables the G_i networks and the generative aspect of the model. This variant does not consider the unlabeled data during the learning. GenSEVEN corresponds to a model that does not have the discriminate component. In other words, it does not have the contrastive layer and does

Table 2: Performance of different methods on LFW and SONOF in terms of accuracy.

Dataset	LFW						SONOF						
	# of labeled pairs	110	440	880	1320	1760	All	160	320	640	960	1280	All
SEVEN		61.2	64.1	65.7	66.3	67.0	68.7	72.7	74.6	79.3	83.1	84.1	85.3
PCA		-	-	-	-	-	64.5	-	-	-	-	-	67.61
DDML		51.5	54.2	61.9	63.8	64.8	71.1	58.5	67.7	72.5	78.4	82.9	86.1
Pseudo-label		52.0	52.2	53.9	57.4	57.9	70.1	53.8	59.9	63.2	71.0	80.5	84.5
PreConvSia		55.1	62.3	63.5	63.2	64.2	66.0	61.9	67.1	70.4	71.5	78.8	82.1
PreAutoSia		51.1	62.9	63.0	63.5	64.2	66.1	57.2	62.7	66.4	70.1	73.1	79.0

Table 3: Performance of different methods on MNIST and USPS in terms of accuracy.

Dataset	MNIST						USPS						
	# of labeled pairs	30	60	120	600	2400	All	40	80	160	300	800	All
SEVEN		75.5	76.9	79.8	84.8	90.7	96.8	76.2	77.3	80.2	80.7	82.8	93.1
PCA		-	-	-	-	-	65.84	-	-	-	-	-	70.96
DDML		61.1	65.9	75.7	84.0	90.4	96.8	69.0	71.8	75.7	75.9	80.8	92.7
Pseudo-label		59.8	67.9	76.8	83.2	89.3	95.2	70.1	57.4	57.9	77.2	78.3	93.3
PreConvSia		64.4	73.0	77.2	82.7	90.8	97.2	72.2	78.2	77.6	78.1	82.9	93.0
PreAutoSia		61.5	68.0	71.9	78.9	84.7	93.1	70.6	73.9	69.0	75.0	82.0	90.2

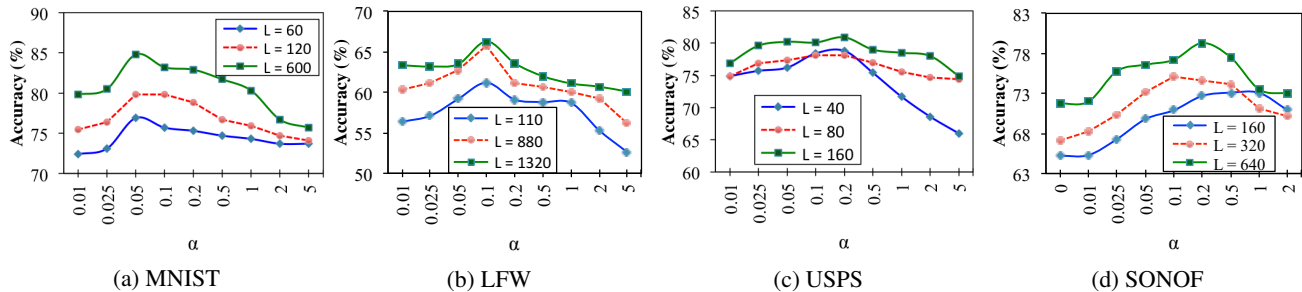


Figure 2: The accuracy of SEVEN for different values of parameter α for (a) MNIST, (b) LFW, (c) USPS and (d) SONOF.

not use the label information. SEVEN indicates the full variant of SEVEN with both generative and discriminative components. The variant SEVEN (MLP) is similar to the regular SEVEN, except that it uses fully connected layers instead of convolutional and transposed convolutional layers.

Among all the different variants, full SEVEN gives the best performance. It shows the effectiveness of both the generative and discriminative components. It also verifies the effectiveness of using the information hidden in the unlabeled data. The results show that the discriminative component has the broader impact compared to the generative component. SEVEN (MLP) gives weaker performance compared to SEVEN. It is mainly because of the capabilities of convolutional layers in modeling image data as it has also been shown by ConvNets in image processing applications.

4.6 Parameter Sensitivity

We analyze the effect of the parameter α in Equation (8) on the performance of SEVEN on all the four datasets. Parameter α of SEVEN controls the trade-off between the generative and discriminative aspects of the model. In Figure 2 the performance of SEVEN for different values of α is plotted. For each dataset, the performance is plotted for three different values of L (number of labeled pairs). There exists a trade-off between the two generative and discriminative aspects of

SEVEN on all of the four datasets. As it can be seen, the optimum value of this parameter is dependent to the dataset and also to the ratio of labeled data to some extent.

Table 4: Performance of SEVEN variants in accuracy.

Method	MNIST (120)	LFW (440)	USPS (80)	SONOF (320)
DisSEVEN	75.7	54.2	73.9	70.3
GenSEVEN	73.0	58.2	60.0	62.5
SEVEN	79.8	64.1	77.3	74.6
SEVEN (MLP)	73.1	60.0	77.3	70.9

5 Conclusion

Benefiting from the salient structures hidden in the unlabeled data and the ability of deep neural networks in nonlinear function approximation, we propose a semi-supervised deep SEmi-supervised VERification Network (SEVEN) for verification tasks. SEVEN benefits from both generative and discriminative modelings in a unified model. These two components are simultaneously trained which lead them to closely interact and influence each other. Extensive experiments demonstrate that SEVEN outperforms other state-of-the-art deep semi-supervised techniques in a wide spectrum of verification tasks. Furthermore, SEVEN shows competitive performance compared with fully supervised baselines that require a significantly larger amount of labeled data, indicating the important role of the generative component in SEVEN.

References

- [Bahaadini *et al.*, 2017] Sara Bahaadini, Neda Rohani, Scott Coughlin, Michael Zevin, Vicky Kalogera, and Aggelos K Katsaggelos. Deep multi-view models for glitch classification. *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2017.
- [Bell and Bala, 2015] Sean Bell and Kavita Bala. Learning visual similarity for product design with convolutional neural networks. *ACM Transactions on Graphics (TOG)*, 34(4):98, 2015.
- [Bromley *et al.*, 1993] Jane Bromley, James W Bentz, Léon Bottou, Isabelle Guyon, Yann LeCun, Cliff Moore, Eduard Säckinger, and Roopak Shah. Signature verification using a siamese time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(04):669–688, 1993.
- [Chopra *et al.*, 2005] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR’05)*, volume 1, pages 539–546. IEEE, 2005.
- [Dauphin *et al.*, 2015] Yann Dauphin, Harm de Vries, and Yoshua Bengio. Equilibrated adaptive learning rates for non-convex optimization. In *Advances in Neural Information Processing Systems*, pages 1504–1512, 2015.
- [Dumoulin and Visin, 2016] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016.
- [Galbally *et al.*, 2015] Javier Galbally, Moises Diaz-Cabrera, Miguel A Ferrer, Marta Gomez-Barrero, Aythami Morales, and Julian Fierrez. On-line signature recognition through the combination of real dynamic data and synthetically generated static data. *Pattern Recognition*, 48(9):2921–2934, 2015.
- [Hoffer and Ailon, 2016] Elad Hoffer and Nir Ailon. Semi-supervised deep learning by metric embedding. *arXiv preprint arXiv:1611.01449*, 2016.
- [Hu *et al.*, 2014] Junlin Hu, Jiwen Lu, and Yap-Peng Tan. Discriminative deep metric learning for face verification in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1875–1882, 2014.
- [Huang *et al.*, 2012] Gary Huang, Marwan Mattar, Honglak Lee, and Erik G Learned-Miller. Learning to align from scratch. In *Advances in Neural Information Processing Systems*, pages 764–772, 2012.
- [Hull, 1994] Jonathan J. Hull. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, 16(5):550–554, 1994.
- [Ioffe and Szegedy, 2015] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.
- [Jain *et al.*, 2016] Himanshu Jain, Yashoteja Prabhu, and Manik Varma. Extreme Multi-label Loss Functions for Recommendation, Tagging, Ranking & Other Missing Label Applications. In *SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2016.
- [Kingma *et al.*, 2014] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.
- [Koch, 2015] Gregory Koch. *Siamese neural networks for one-shot image recognition*. PhD thesis, University of Toronto, 2015.
- [LeCun *et al.*, 2015] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
- [Lee, 2013] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshop on Challenges in Representation Learning*, volume 3, page 2, 2013.
- [Li *et al.*, 2016] Yawei Li, Lizuo Jin, AK Qin, Changyin Sun, Yew Soon Ong, and Tong Cui. Semi-supervised auto-encoder based on manifold learning. In *International Joint Conference on Neural Networks (IJCNN)*, pages 4032–4039, 2016.
- [Maaløe *et al.*, 2016] Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. Auxiliary deep generative models. In *International Conference on Machine Learning (ICML)*, pages 1445–1453, 2016.
- [Masci *et al.*, 2011] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. *Artificial Neural Networks and Machine Learning (ICANN)*, pages 52–59, 2011.
- [Nair and Hinton, 2010] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning (ICML)*, pages 807–814, 2010.
- [Oh Song *et al.*, 2016] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4004–4012, 2016.
- [Rasmus *et al.*, 2015] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems*, pages 3546–3554, 2015.
- [Srivastava *et al.*, 2014] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- [Sun *et al.*, 2014] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation by joint identification-verification. In *Advances in Neural Information Processing Systems*, pages 1988–1996, 2014.
- [Sun *et al.*, 2017] Yao Sun, Lejian Ren, Zhen Wei, Bin Liu, Yanlong Zhai, and Si Liu. A weakly-supervised method for makeup-invariant face verification. *Pattern Recognition*, 2017.
- [Wolf, 2014] Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [Xie and Philip, 2017] Sihong Xie and S Yu Philip. Active zero-shot learning: a novel approach to extreme multi-labeled classification. *International Journal of Data Science and Analytics*, pages 1–10, 2017.
- [Zagoruyko and Komodakis, 2015] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4353–4361, 2015.
- [Zheng *et al.*, 2017] Lei Zheng, Vahid Noroozi, and Philip S Yu. Joint deep modeling of users and items using reviews for recommendation. In *10th ACM International Conference on Web Search and Data Mining (WSDM)*, pages 425–434. ACM, 2017.