# Virtual Adversarial Training for Semi-supervised Verification Tasks

Vahid Noroozi
*Department of Computer Science*
*University of Illinois at Chicago*
Chicago, IL, USA
vnoroo2@uic.edu

Sara Bahaadini
*Department of Computer Science*
*Northwestern University*
Evanston, IL, USA
sara.bahaadini@u.northwestern.edu

Lei Zheng
*Department of Computer Science*
*University of Illinois at Chicago*
Chicago, IL, USA
lzheng21@uic.edu

Sihong Xie
*Computer Science and Engineering Department*
*Lehigh University*
Bethlehem, PA, USA
sxie@cse.lehigh.edu

Philip S. Yu
*Department of Computer Science*
*University of Illinois at Chicago*
Chicago, IL, USA
psyu@uic.edu

***Abstract*—The goal in verification tasks is to determine the similarity of two samples or verifies if they belong to the same category or not. In this paper, we propose a semi-supervised embedding technique for verification tasks using deep neural networks. The proposed model exploits the unlabeled data by making the model robust to the perturbation of the input with virtual adversarial training. It increases the generalization of the embedding function and prevents overfitting which are crucial in verification tasks. The proposed algorithm, named VerVAT, is evaluated on several verification tasks and compared with state-of-the-art algorithms. Experiments show the effectiveness of VerVAT especially in cases where limited labeled data is available.**

***Index Terms*—Verification Task, Semi-supervised Learning, Deep Representation Learning, Virtual Adversarial Training.**

## I. Introduction

The task of estimating the similarity of two objects is called verification task. It has important applications such as face verification [1], signature verification [2], and learning sentence similarity [3]. Most of the suggested models for verification tasks are based on embedding learning techniques. It makes them applicable for some other tasks such as classification problems in scenarios with a large number of classes and limited or skewed number of samples for each class [3], [4].

Deep learning models have shown great and promising performance in many applications recently [5] but most of the successes are in supervised tasks where a large amount of labeled data is available. Labeling data can be very expensive or not feasible in some cases while unlabeled data are abundant for many problems. In such applications, semi-supervised learning can be an effective solution. While some works have been done on training neural networks in semi-supervised setting for classification problems, to the best of our knowledge, limited works have been done on deep semi-supervised verification tasks. In [6], a semi-supervised model, called SEVEN, is proposed which combines a supervised loss with an unsupervised one to handle unlabeled data. It showed promising results compared to the baselines. However the unsupervised part of SEVEN is based on auto-encoders. One of the drawbacks of auto-encoding approach is that the decoder part doubles the size of the network.

In this paper, we propose a semi-supervised embedding model for verification tasks. The proposed algorithm, named VerVAT, benefits from Virtual Adversarial Training (VAT) [7] to exploit the unlabeled data. Adversarial training may refer to different categories of machine learning algorithms. These algorithms are used for a variety of problems such as Generative Adversarial Networks (GAN) [8], adversarial examples [9], and adversarial loss optimization [10]. VAT is adopted from the adversarial training [9] technique originally proposed for increasing the robustness of neural networks toward adversarial examples. VAT has shown promising performance for semi-supervised classification tasks [11] where the distributions of train and test data are similar. But to the best of our knowledge, it has never been applied to embedding learning problems where classes of the training and test data can be different. We are the first to adopt this idea and propose a semi-supervised learning model for verification tasks through the introduction of an objective function based on virtual adversarial training. The proposed objective function is a combination of a discriminative part which imposes separation between various classes and a VAT based part which exploits the underlying structure of the unlabeled data. Virtual adversarial loss also helps the model to avoid overfitting and to have a smoother embedding function.

The proposed model can also be used in other tasks such as extreme classification where there exists a large number of classes in the order of thousands or millions. One common example of such tasks is face recognition where there may exist millions of classes with few samples for each class. In such settings, traditional neural networks for classification suffer from long tail problem and overfitting [12].

We have evaluated VerVAT on three different verification tasks. In two of them, the training and test samples are drawn

978-9-0827-9703-9/19/$31.00 

from disjoint classes which can not be handled easily by most of the traditional classification techniques for neural networks. In all of the experiments, the proposed algorithm achieves better results in terms of accuracy compared to the baselines. It shows the effectiveness of virtual adversarial training for semi-supervised embedding learning.

## II. Problem Formulation

We define the training data as a set of pairs consisting of two samples. The items of a pair can belong to the same class to form a positive pair or belong to different classes to from a negative pair. If the class information for at least one of the items of a pair is missing or not available, the pair's label is considered as unknown. The training set is represented as $\mathcal{D} = \left\{(x_1^i, x_2^i)\right\}_{i=1}^{N}$, where $(x_1^i, x_2^i)$ is a pair of training samples. $x_1^i \in \mathbb{R}^m$ is the first item of the $i^{\text{th}}$ pair and $x_2^i \in \mathbb{R}^m$ is the second item. The total number of pairs is indicated by $N$. The label set is defined as $\mathcal{L} = \{y^i | y^i \in \{p, n, u\}\}_{i=1}^{L}$ where $p$, $n$, and $u$ denote the positive, negative and unknown label, respectively.

We want to learn a parametric and highly nonlinear function that can verify whether two samples are similar or not. To be more specific, the goal of the model is to learn function $v(x_1, x_2; \Theta)$ to predict the relation between $x_1$ and $x_2$. It is defined based on the distance of $x_1$ and $x_2$ as

$$v(x_1, x_2; \Theta) = \begin{cases} p & \text{if} \quad d(f(x_1;\Theta), f(x_2;\Theta)) \leq \tau \\ n & \text{if} \quad d(f(x_1;\Theta), f(x_2;\Theta)) > \tau \end{cases} \quad (1)$$

where $d(.,.)$ is an arbitrary distance function. Function $f(x;\Theta)$ is a highly nonlinear function parameterized by $\Theta$ which maps a sample to a new space where distances can get estimated by a simple distance function like Euclidean or cosine distance. The threshold $\tau$ specifies the maximum distance that samples of a positive pair are allowed to have from each other. Samples farther than this threshold are considered to be from different classes with negative relation.

## III. Proposed Algorithm

### A. Model Architecture

The overall architecture of the proposed model is illustrated in Fig. 1. The input pair is given to two neural networks denoted as $F_1$ and $F_2$ with shared weights and parameters $\Theta$ like Siamese networks [2]. Siamese networks are widely used in similarity learning [3], [4], [13], embedding learning [14], [15], verification [2], [16], [17], and retrieval [18].

They should project the input samples to a new discriminative space where samples with positive relation are close to each other and samples with negative relation are far from each other. As the weights of $F_1$ and $F_2$ are shared, both subnetworks define the same nonlinear mapping function, denoted by $f(.;\Theta)$. To make the new representation to have such a discriminative property, a layer is added at the top of the networks $F_1$ and $F_2$ that calculates the distance between the two input samples in the new space denoted by $d(.,.)$. Function $d$ can be an arbitrary distance metric such as Euclidean or cosine distance in the new subspace. This

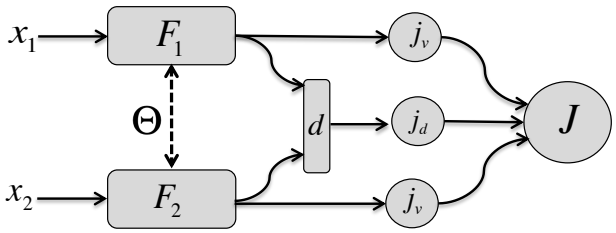


Fig. 1: The schematic representation of VerVAT. $F_1$ and $F_2$ are the neural networks with shared weights, and the circle shapes denote the loss functions.

function can be considered as a metric distance function which networks $F_1$ and $F_2$ are supposed to learn it. These networks are ConvNets built with convolutional layers, max-pooling, and a fully connected layer as the last layer.

### B. Loss Function

We propose to impose two main characteristics on the new subspace to be learned by networks $F_1$ and $F_2$. First of all, the new subspace obtained from these two networks should be discriminative so that samples from different classes are separable easily. Samples from the same class should be close to each other, and samples from different classes should be far from each other. This property makes the similarity prediction performed by function $d(.,.)$ easier. However, the discriminative property is not enough for semi-supervised settings where the relation of some pairs are not available.

To fully exploit the information of all data, we impose the unsupervised constraint. Another challenge in training neural networks is overfitting especially when the distribution of the classes in the test and train are different [11]. To address this problem, we propose to adopt the idea of virtual adversarial training (VAT) to regularize the training process (will be explained in detail in Section III-B2). Both properties are imposed by a unified loss function as

$$\mathcal{J}(X, Y; \Theta) = (1-\alpha)\mathcal{J}_{\mathcal{D}}(X, Y; \Theta) + \alpha\mathcal{J}_{\mathcal{V}}(X; \Theta) + \beta \left\|\Theta\right\|_2 \quad (2)$$

where $\mathcal{J}_{\mathcal{D}}(X, Y; \Theta)$ indicates the supervised loss for labeled data, and $\mathcal{J}_{\mathcal{V}}(X; \Theta)$ is the unsupervised loss for all data which imposes the adversarial training loss on the learned function. Parameter $\beta$ controls the regularization term $\|\Theta\|$ which is imposed on all the weight parameters of the network. Regularization to prevent overfitting is important especially in cases where the distribution of train and test data are not similar. Parameter $\alpha$ is the weighting parameter that controls the trade-off between the supervised and unsupervised part of the loss.

*1) Discriminative Space:* The discriminative part of the loss function, $\mathcal{J}_{\mathcal{D}}(X, Y; \Theta)$ is estimated for the $L$ labeled pairs as:

$$\mathcal{J}_{\mathcal{D}}(X, Y; \Theta) = \sum_{1 \leq i \leq L} j_d(x_1^i, x_2^i; \Theta) \quad (3)$$

where $j_d(x_1, x_2)$ indicates the discriminative loss for the pair $(x_1, x_2)$ in the new subspace. It can be defined with a contrastive loss function as:

$$j_d(x_1^i, x_2^i; \Theta) = I\{y^i = p\} d(f(x_1; \Theta), f(x_2; \Theta))^2 + I\{y^i = n\} max\{0, m - d(f(x_1; \Theta), f(x_2; \Theta))\}^2 \quad (4)$$

where $I\{.\}$ is the identity function. Function $d$ measures the distance of two samples in the new space. We used Euclidean distance as the distance function. It penalizes the distance between positive samples and also the similarity between negative ones. This loss pushes the positive samples close to each other in the space while pushes negative samples far from each other. It makes the new representation space discriminative. Parameter $m$ specifies a margin which prevents the loss function to push negative pairs further than $m$.

*2) Virtual Adversarial Training:* In order to exploit the information in the unlabeled data we adopt Virtual Adversarial Training (VAT) [7] to our embedding learning model. VAT is inspired from the defense techniques which are used to increase the robustness of neural networks toward adversarial attacks. It tries to minimize the change in the output of a neural network when its input is perturbed locally. It regularizes the embedding space and increases the generalization of the learned subspace while there exists limited labeled data. VAT has shown to be effective for semi-supervised learning [11].

The loss $\mathcal{J}(X, Y; \Theta)$ for all the pairs including unlabeled and labeled samples is defined as:

$$J_v(X; \theta) = \sum_{1 \le i \le N} \sum_{j=1}^{2} j_v(x_j^i; \theta) \quad (5)$$

where $j_v(x_j^i; \theta)$ estimates the VAT loss for sample $x_j^i$. It is defined as the following to minimize the greatest change in the embedding space for sample $x_j^i$.

$$j_v(x_j^i; \theta) = g(f(x_j^i; \theta), f(x_j^i + r_{adv}; \theta))$$
$$r_{adv} = \underset{r; \|r\|_2 < \varepsilon}{\arg\max}\, g(\, f(x_j^i; \theta), f(x_j^i + r; \theta)) \quad (6)$$

where $g$ is a non-negative function which measures the distance between its two inputs, and $\varepsilon$ is a small positive number. We selected Euclidean distance function as the distance function $g$. Vector $r_{adv}$ is the adversarial perturbation which specifies the direction in the input space which produces the maximum difference in the embedding space. By minimizing this loss function, the sensitivity of the output embedding space to the input perturbation is minimized. There exists no closed form to calculate the vector $r_{adv}$, but it can get approximated by

$$r_{adv} = \varepsilon \frac{g}{\|g\|} \quad (7)$$

where

$$g = \nabla_r d(f(x_j^i; \theta), f(x_j^i + r; \theta))$$
$$r \sim N(0, \frac{\varepsilon}{\sqrt{D_x}} I) \quad (8)$$

Vector $r$ is a random noise vector added to the input of the neural networks to create the perturbation. It is drawn from a normal distribution $N$. $D_x$ is the dimension size of the inputs, and $I$ is an identity matrix with the dimension of $D_x$.

The gradient vector $g$ can get computed by back-propagation on the network. More details on this approximation can be found in [7].

The whole model is trained using backpropagation with respect to the loss function in Equation 2. Given a set of $N$ pairs, we optimize the model by Adam [20] optimization technique over shuffled mini-batches. Batch normalization [21] technique is also applied after each convolutional layer to normalize the output of each layer.

## IV. Experiments

### A. Datasets

We evaluate the proposed algorithm on the following datasets:

**Labeled Faces in the Wild (LFW)** [22]: It is a database of face photographs designed for evaluating face verification or recognition tasks. It contains 2200 pairs of face images consisting of 1100 positive and 1100 negative pairs for verification tasks. Positive pairs are images from the same person, while negative pairs are from different persons. There are 500 positive and 500 negative pairs in the test set. Due to the small size of the training data, we use 5-fold validation in the validation process for estimating the best parameters.

**BiosecurID-SONOF (SONOF)** [23]: We use a subset of this dataset comprising signatures collected from 132 users. It contains 16 signatures for each user. All images are normalized and resized to $80 \times 80$. Users are randomly divided into two groups of 100 and 32 for the training and test purposes.

**US Postal Service (USPS)** [24]: USPS dataset contains 9298 handwritten digits automatically scanned from envelopes by the US Postal Service. It has 10 classes. All images are normalized 16 x 16 grayscale. We divided the samples randomly into $7900/1398$ for training and test. After the pairing process, we will have 7900 and 1398 pairs for training, and test. We used 5-fold cross-validation for estimating the best values for the parameters. All images are resized to $64 \times 48$.

Dataset LFW is originally built for verification tasks, and its train and test samples are already in the form of positive and negative pairs, but the rest are mostly used for image classification tasks. We make these datasets in pairs so that they can be used for verification. The pairing process is as follows. First, we split the training data randomly into labeled and unlabeled sets with the specified ratio. Then, each sample gets paired with another sample randomly. The other sample is selected from the same class with the probability of $0.5$, otherwise from a different class to have equal number of positive and negative pairs. The pairs are selected from their own corresponding set, labeled or unlabeled. Test or validation samples are not divided into labeled and unlabeled sets like training set, but they just get paired with a similar process. The classes in the training and test samples are disjoint in SONOF

TABLE I: Performance of different methods on LFW, SONOF, and USPS in terms of accuracy.

| Dataset | LFW | | | | SONOF | | | | USPS | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| # of labeled pairs | 110 | 880 | 1760 | All | 160 | 640 | 1280 | All | 40 | 160 | 800 | All |
| PCA | - | - | - | 64.5 | - | - | - | 67.6 | - | - | - | 70.9 |
| DDML [1] | 51.5 | 61.9 | 64.8 | 71.1 | 58.5 | 72.5 | 82.9 | 86.1 | 69.0 | 75.7 | 80.8 | 92.7 |
| Pseudo-label [19] | 52.0 | 53.9 | 57.9 | 70.1 | 53.8 | 63.2 | 80.5 | 84.5 | 70.1 | 57.9 | 78.3 | **93.3** |
| Autoencoder-Siamese | 55.1 | 63.5 | 64.2 | 66.0 | 61.9 | 70.4 | 78.8 | 82.1 | 72.2 | 77.6 | 82.9 | 93.0 |
| SEVEN [6] | 61.2 | 65.7 | 67.0 | 68.7 | 72.7 | 79.3 | 84.1 | 85.3 | 76.2 | 80.2 | 82.8 | 93.1 |
| VerVAT | **61.6** | **68.6** | **72.6** | **73.5** | **82.9** | **83.45** | **85.6** | **87.7** | **78.2** | **84.5** | **84.9** | 93.0 |

and LFW datasets, while in USPS dataset, classes are common between the test and train.

### *B. Baselines*

Handling new classes in the test data is a common case in verification tasks, while it is a great challenge for most of the traditional classification techniques based on neural networks. Therefore, we adopted some of the deep semi-supervised techniques to verification networks to be used as our baselines.

**Principle Component Analysis (PCA)**: It as an unsupervised feature learning technique which does not need any label information. The distance between samples after applying the PCA transformation is considered as the similarity of two samples. A threshold is selected for each dataset based on the performance on the training data to find the relation between two samples.

**Pseudo-Label** [19]: It is a semi-supervised approach for training deep neural networks. It initially trains a supervised model with the labeled data. Then in each epoch, it predicts the labels of the unlabeled samples with the trained model, and then adds the ones with high confidence to the labeled samples to continue training. The model was proposed and evaluated for classification tasks. We followed the same approach to train a Siamese network [2].

**Discriminative Deep Metric Learning (DDML)** [1]: It uses the architecture of Siamese networks [2] with a modified version of the contrastive loss function. It is a supervised approach and does not use unlabeled pairs.

**Autoencoder-Siamese**: It pre-trains an autoencoder in an unsupervised manner. Then, its encoder part is fine-tuned with labeled pairs in a Siamese network [2] structure.

**SEVEN** [6]: It is a model based on neural networks specifically proposed for semi-supervised verification tasks. This model used auto-encoding and generative models to handle unlabeled data and prevent overfitting problem while our algorithm benefits virtual adversarial training to exploit the information in the unlabeled data.

### *C. Performance Evaluation*

The performance of VerVAT and all baselines are presented in Table I. The results are reported for a different number of labeled pairs and the best accuracy for each case is depicted in bold. The performance is reported 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 data. The last column of each section indicates the case where all the training pairs have label information. As PCA is a fully unsupervised method, no label information is used for this baseline, and just one performance is reported for each dataset.

Most of the parameters of baselines are selected based on the accuracy metric using cross-validation. USPS is divided into training and validation sets because is has enough samples, but LFW and SONOF are validated with 5-fold validation. After finding the best values for parameters with 5-fold validation, the whole training data is used for training.

All the neural networks are trained for $250$ epochs with Adam [20] optimizer and the best model with the lowest loss is selected as the final model. The pre-training phase of training for both Pseudo-Label and Autoencoder-Siamese is performed for $150$ epochs. The batch size is set to $512$ for all the experiments. Margin parameter $m$ is set to $1$.

As can be seen, VerVAT outperforms other baselines in terms of accuracy in cases with limited number of labeled pairs. The difference in performance compared to other baselines is more significant for the lower number of labeled pairs. It verifies empirically the effectiveness of the proposed approach of addressing the problem of limited labeled data.

One of the drawbacks of SEVEN and Autoencoder-Siamese is that they use autoencoders. Their encoders should incorporate most of the unnecessary detail of the image data into the hidden representations so that the decoder can reconstruct the original input. Such representations contain unnecessary information for the goal task and can affect the performance of the verification task while VerVAT benefits VAT to exploit the unlabeled data and does not have this limitation.

## V. Conclusion

We presented a deep verification network that learns a distance metric for semi-supervised verification tasks whose training samples consists of negative, positive or unknown pairs. It exploits the unlabeled and labeled data in a joint manner to learn a discriminative feature space. The proposed model is the first verification model for semi-supervised setting which benefits from virtual adversarial training to learn a robust and smooth embedding space. The experiments demonstrated the effectiveness of the proposed algorithm. It outperforms state-of-the-art deep semi-supervised learning approaches for verification tasks on all the experimented datasets.

## Acknowledgment

This work is supported in part by NSF through grants IIS-1526499, IIS-1763325, and CNS-1626432.

## REFERENCES


[1] 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)*, 2014, pp. 1875–1882.

[2] 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*, vol. 7, no. 04, pp. 669–688, 1993.

[3] Jonas Mueller and Aditya Thyagarajan, "Siamese recurrent architectures for learning sentence similarity," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[4] Paul Neculoiu, Maarten Versteegh, and Mihai Rotaru, "Learning text similarity with siamese recurrent networks," in *Proceedings of the 1st Workshop on Representation Learning for NLP*, 2016, pp. 148–157.

[5] Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, and SS Iyengar, "A survey on deep learning: Algorithms, techniques, and applications," *ACM Computing Surveys (CSUR)*, vol. 51, no. 5, pp. 92, 2018.

[6] Vahid Noroozi, Lei Zheng, Sara Bahaadini, Sihong Xie, and Philip S Yu, "Seven: deep semi-supervised verification networks," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. AAAI Press, 2017, pp. 2571–2577.

[7] Takeru Miyato, Shin-ichi Maeda, Shin Ishii, and Masanori Koyama, "Virtual adversarial training: a regularization method for supervised and semi-supervised learning," *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[9] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy, "Explaining and harnessing adversarial examples," in *ICLR*, 2015.

[10] Rizal Fathony, Ashkan Rezaei, Mohammad Ali Bashiri, Xinhua Zhang, and Brian Ziebart, "Distributionally robust graphical models," in *Advances in Neural Information Processing Systems*, 2018, pp. 8344–8355.

[11] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow, "Realistic evaluation of deep semi-supervised learning algorithms," in *Advances in Neural Information Processing Systems*, 2018, pp. 3235–3246.

[12] Kashif Shah, Selcuk Kopru, and Jean David Ruvini, "Neural network based extreme classification and similarity models for product matching," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, 2018, vol. 3, pp. 8–15.

[13] Sean Bell and Kavita Bala, "Learning visual similarity for product design with convolutional neural networks," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, pp. 98, 2015.

[14] Sara Bahaadini, Neda Rohani, Aggelos K Katsaggelos, Vahid Noroozi, Scott Coughlin, and Michael Zevin, "Direct: Deep discriminative embedding for clustering of ligo data," in *25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 748–752.

[15] Sara Amini, Vahid Noroozi, Sara Bahaadini, S Yu Philip, and Chris Kanich, "Deepfp: A deep learning framework for user fingerprinting via mobile motion sensors," in *IEEE International Conference on Big Data*. IEEE, 2018, pp. 84–91.

[16] Gregory Koch, *Siamese neural networks for one-shot image recognition*, Ph.D. thesis, University of Toronto, 2015.

[17] 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)*. IEEE, 2005, vol. 1, pp. 539–546.

[18] Yonggang Qi, Yi-Zhe Song, Honggang Zhang, and Jun Liu, "Sketch-based image retrieval via siamese convolutional neural network," in *IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 2460–2464.

[19] 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*, 2013, vol. 3, p. 2.

[20] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[21] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[22] Gary Huang, Marwan Mattar, Honglak Lee, and Erik G Learned-Miller, "Learning to align from scratch," in *Advances in Neural Information Processing Systems*, 2012, pp. 764–772.

[23] 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*, vol. 48, no. 9, pp. 2921–2934, 2015.

[24] Jonathan J. Hull, "A database for handwritten text recognition research," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 16, no. 5, pp. 550–554, 1994.