Learning from Synthetic Data Using a Stacked Multichannel Autoencoder

Xi Zhang*, Yanwei Fu‡, Shanshan Jiang†, Leonid Sigal‡ and Gady Agam*
*Illinois Institute of Technology, Chicago, Illinois 60616
Email: {xzhang22, sjiang20}@hawk.iit.edu
†Illinois Institute of Technology, Chicago, Illinois 60616
Email: agam@iit.edu
‡Disney Research, Pittsburgh, PA 15213
Email: {yanwei.fu, lsigal}@disneyresearch.com

***Abstract*—Learning from synthetic data has many important and practical applications, An example of application is photo-sketch recognition. Using synthetic data is challenging due to the differences in feature distributions between synthetic and real data, a phenomenon we term *synthetic gap*. In this paper, we investigate and formalize a general framework – Stacked Multichannel Autoencoder (SMCAE) that enables bridging the synthetic gap and learning from synthetic data more efficiently. In particular, we show that our SMCAE can not only transform and use synthetic data on the challenging face-sketch recognition task, but that it can also help simulate real images, which can be used for training classifiers for recognition. Preliminary experiments validate the effectiveness of the framework.**

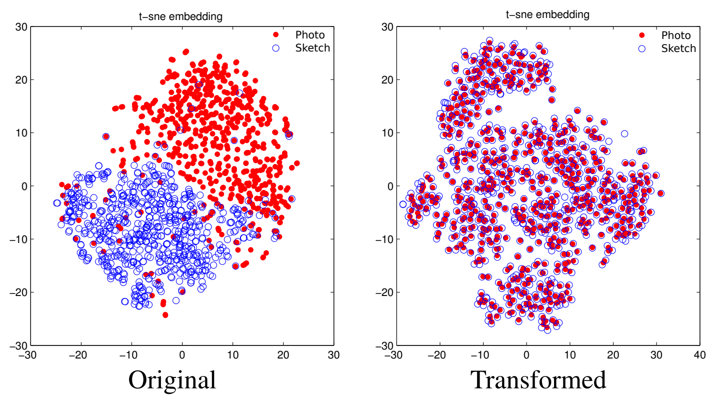

Fig. 2. t-SNE visualization [23] of the distribution of Histogram of Oriented Gradients (HOG) features in the data in CUFSF dataset [29], [31]. Left: synthetic gap is observed between photo and sketch features; Right: the synthetic gap is bridged by our SMCAE.

I. Introduction

Modern supervised learning algorithms need plenty of data to help train classifiers. More data with higher quality is always desired in real-world applications; but sometimes, it is beneficial to turn to synthetic data. For example, to help identify criminals, many criminal investigations can only rely on a synthetic face sketch rather than a facial photograph of a suspect which may not be available. Such synthetic face data is normally drawn by an expert based on descriptions of eyewitnesses and/or victim(s). Several photo-sketch examples are shown in Fig. 1. In this application, recognition based on synthetic data is very crucial.

Directly using synthetic data in a learning algorithm is unfortunately very challenging since synthetic data is different from real data at least to some extent, e.g. exaggerated facial shapes in sketch images in Fig. 1 as compared with real images. As a result, the feature distributions of synthetic data may be shifted away from those of real data as illustrated in Fig. 2. We term such shift in distributions as *synthetic gap*. Synthetic gap is largely caused by the generating process of synthetic data: whereas the synthetic data are generated by replicating principal patterns such as eyes, mouth, nose and hairstyle, rather than replicating every detail of real data. The synthetic gap is a major obstacle in using synthetic data in recognition problems, since synthetic data may fail to simulate potentially useful patterns of real data which are important to a successful recognition. To solve this problem, we associate synthetic data with real data, and jointly learn from them in a Stacked Multichannel Autoencoder (SMCAE) which can help bridge the synthetic gap by transforming characteristics of synthetic data to better simulate real data.

This paper addresses the problem of learning a mapping from synthetic data to real data. Specifically, we propose a novel framework – SMCAE. The training process of SMCAE facilitates the bridging of the synthetic gap between the real and the synthetic data by learning how to transform: (1) synthetic to real data and (2) real to real data. In (2) the model learns most essential 'characters' and 'patterns' of real data, while in (1) it learns how to augment the synthetic data to best reproduce the distribution of real data. Because the two tasks are learned simultaneously, with shared parameters, the essential 'characteristics' learned in (2) help to regularize results in (1) and vice versa as we will illustrate in the Handwritten Digit experiments.

We highlight two main contributions of this paper: (1) To the best of our knowledge, this is the first attempt to address the problem of *synthetic gap*, by demonstrating that the synthetic data could be used to improve the performance on a recognition task. (2) We propose a Stacked Multichannel Autoencoder (SMCAE) model to bridge the *synthetic gap* and jointly learn from both real and synthetic data.

II. Related Work

Transfer Learning aims to extract the knowledge from one, or more, source tasks and apply it to a target task. Transfer learning can be used in many different applications, such as

Fig. 1. Examples of face photos and sketches. Data comes from the CUFSF dataset [29], [31].

web page classification [19] and zero-shot classification [14]. A more detailed survey of transfer learning is given by [16]. Our method is a specific form of transfer learning, termed domain adaptation [6], [30], [32]. Nonetheless, different from previous domain adaptation approaches, we assume the the *synthetic gap* is caused by the shift in feature distribution of synthetic data from real data and so we assume that the main 'characters' and 'patterns' strongly co-exist in both the synthetic and real data. Our SMCAE is thus developed based on this assumption.

Autoencoder is a special type of a neural network where the output vectors have the same dimensionality as the input vectors [27]. Autoencoder with its different variants [9], [11], [2], [18] was shown to be successful in learning and transferring shared knowledge among data source from different domains [5], [8], [10], and thus benefit other machine learning tasks. Our framework borrows the idea of autoencoder to jointly learn two different and yet related tasks: mapping synthetic to real data; and real to real data. It is worth noting that in [20], a multimodal autoencoder with structure similar to ours is proposed. Their multimodal autoencoder put two normal autoencoders together by sharing a hidden layer. In their structure, data at input end and output end are fully symmetric and each modal of data occupy one branch of the antuencoder. In contrast to their structure, the proposed SMCAE composes the structure of both normal autoencoder and denoising autoencoder. With this composition, one branch of SMCAE is capable exploring intrinsic features of data in one domain, and another branch of SMCAE is going to transfer data from one domain to another domain using features discovered from both branches. The structure of SMCAE could be easily expanded to more branches to compensate more complicated multi-task learning problems. Our experiments show that our SMCAE is better than other autoencoders in this regard.

Learning from synthetic templates. Some recent works of learning from synthetic data [24], [25], [4] mostly generate synthetic data either by applying a simple geometric transformation or adding image degradation to real data. To help offline recognition of handwritten text [24], [25], a perturbation model combined with morphological operation is applied to real data. To enhance the quality of degraded document [4], degradation models such as brightness degradation, blurring degradation, noise degradation, and texture-blending degradation, were used to create a training dataset for a handwritten text recognition problem. These methods did not address the synthetic gap problem, and thus have been limited to a small performance improvements by using synthetic data. In [17], computer graphics 3D models are used to ease training data generation. To simulate pedestrian in a picture, authors track volunteers pose from multiple views and human bodies are reshaped using a morphable 3D human model. The reshaped picture of human bodies later are composed with real world backgrounds. The same idea has been adopted in [21] where in addition to render a 3D model to simulate an object in a real scene, features extracted from synthetic data are adapted to better train an object detector.

III. Stacked Multichannel Autoencoder (SMCAE)

We propose the SMACE model to learn a mapping from synthetic and real data. To learn this mapping, the SMCAE model is formulated as a stacked structure of multichannel autoencoders which facilitates an efficient and flexible way of jointly learning from both synthetic and real data. The structure and configuration of the SMCAE is illustrated in Fig. 3. Specifically, we set the left and right tasks in two channels of the SMCAE respectively. The *left task*, as illustrated in left channel of Fig. 3, takes synthetic data as input and real data as reconstruction target; while the *right task* of the right channel in Fig. 3 uses real data in both input and reconstruction target. All between-layer connections that are colored in gray are shared by tasks of the two channels. The SMCAE structured in this way attempts to transform synthetic data to real data in *left task* using representation learned from real data in *right task*.

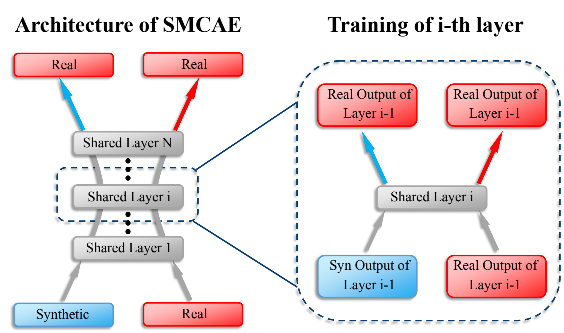

Fig. 3. (left) Illustration of the SMCAE: black edges between two layers are linked to and shared by two tasks; red and blue links are separately connected to the left and right task respectively. (right) A zoom-in structure of SMCAE with single hidden layer.

A. Problem setup

We first illustarte the setup of a single layer in each channel of our SMCAE. For a single channel of our SMCAE is basically an autoencoder [7][26]. Assume an input dataset with n instances $X = \{x_i\}_{i=1}^{n}$ where $x_i \in \mathbb{R}^m$. To encode the input data, we have $h_e(x_i) = f(W_e^j x_i + b_e^j)$ where $f(\cdot)$ is a sigmoid function and $\theta_e = \{W_e^j, b_e^j\}$, $W_e^j \in \mathbb{R}^{k\times m}, b_e^j \in \mathbb{R}^k$ is a set of encoding parameters in j-th layer. In contrast, the decoding process is defined as $h_d(x_i) = f(W_d^j h_e(x_i) + b_d^j)$ with the decoding parameters $\theta_d = \{W_d^j, b_d^j\}$, $W_d^j \in \mathbb{R}^{m\times k}, b_d^j \in \mathbb{R}^m$ and the encoded representations $h_e(x_i)$.

To minimize the reconstruction error, we have

$$J(\theta_e, \theta_d) = \frac{1}{n}\sum_{i=1}^{n}(h_d(x_i) - x_i)^2 + \lambda W^j \tag{1}$$

where $W^j = (\sum_k \sum_m (W_e^j)^2 + \sum_m \sum_k (W_d^j)^2)/2$ is a weight decay term added to improve generalisation of the autoencoder and λ leverages the importance of this term. To avoid learning the identity mapping in the autoencoder, a regularisation term $\Theta = \sum_{i=1}^{k} \delta \log \frac{\delta}{\hat{\delta}_i} + (1-\delta)\log \frac{1-\delta}{1-\hat{\delta}_i}$ that penalizes over-activation of the nodes in the hidden layer is added[1]. $\hat{\delta}_i$ is an averaged activation of all nodes in the hidden layer and is computed as: $\hat{\delta}_i = \frac{1}{k}\sum_{i=1}^{k} h_e(x_i)$. Thus the objective of single channel is updated to:

$$J(\theta_e, \theta_d) = \frac{1}{n}\sum_{i=1}^{n}(h_d(x_i) - x_i)^2 + \lambda W^j + \rho\Theta \tag{2}$$

where ρ controls sparsity of representation in hidden layer.

B. The SMCAE model

The structure of the SMCAE model is extended from an autoencoder so that it can simultaneously deal with tasks in both the left and right channels. Specifically, we use the notation $\langle \mathfrak{i}{:}X,\ \mathfrak{o}{:}X\rangle$ to denote the configuration of input data (short for $\mathfrak{i}$) and reconstruction target at the output layer (short for $\mathfrak{o}$) in one channel of SMCAE. We thus label the tasks in the left and right channels of SMCAE as $\langle \mathfrak{i}{:}X_s,\ \mathfrak{o}{:}X_r\rangle^L$ and $\langle \mathfrak{i}{:}X_r,\ \mathfrak{o}{:}X_r\rangle^R$ individually, where $\langle\cdot\rangle^L$ and $\langle\cdot\rangle^R$ indicate the left and right channel branch of SMCAE. X_s, X_r stand for synthetic and real data respectively. The tasks in the two channels share the same parameters θ_e in all hidden layers which enforces the autoencoder to learn common structures of both tasks. At the output layer, we divide the SMCAE into two separate channels with their own parameters θ_d^L and θ_d^R.

Our target is to minimize the reconstruction error of the two tasks of SMCAE together while taking into account the balance between two channels. The new objective function of SMCAE is thus,

$$E = J^L(\theta_e, \theta_d^L) + J^R(\theta_e, \theta_d^R) + \gamma\Psi \tag{3}$$

We add $\Psi = \frac{1}{2}(J^L(\theta_e, \theta_d^L) - J^R(\theta_e, \theta_d^R))^2$ as a regularisation term to balance the learning rate between the two channels.

The regularization term of Ψ is a novel contribution of our SMCAE. Basically, Ψ penalizes a situation where the difference of learning errors between two channels are large. Since in the configuration of the SMCAE the data at the input and output end of two channels are not symmetric, the learning error resulted by optimizing learning process in two channels are very different. Having Ψ in our objective will prevent from a situation where the optimization of one channel dominates the entire SMCAE so as to help SMCAE to better leverage the learning process and find a compromising balance between two channels. For importance of Ψ in our objective, we show the learning results of setting different γ for Ψ in Fig. 7

The minimization of Eq. 3 is achieved by back propagation and stochastic gradient descent using a Quasi-Newton method – LBFGS. In the SMCAE, with balance regularization added to the objective, the only difference as opposed to sparse autoencoder is the gradient computation of unknown parameters θ_e and θ_d^L, θ_d^R. We clarify these differences in the following equations:

$$\begin{aligned}
\nabla_{W_e^j} E &= \frac{\partial J^L}{\partial W_e^j} + \frac{\partial J^R}{\partial W_e^j} + \gamma(J^L - J^R)(\frac{\partial J^L}{\partial W_e^j} - \frac{\partial J^R}{\partial W_e^j}) \\
\nabla_{b_e^j} E &= \frac{\partial J^L}{\partial b_e^j} + \frac{\partial J^R}{\partial b_e^j} + \gamma(J^L - J^R)(\frac{\partial J^L}{\partial b_e^j} - \frac{\partial J^R}{\partial b_e^j})
\end{aligned} \tag{4}$$

and

$$\begin{aligned}
\nabla_{W_d^L} E &= \frac{\partial J^L}{\partial W_d^L} + \gamma(J^L - J^R)\frac{\partial J^L}{\partial W_d^L} \\
\nabla_{b_d^L} E &= \frac{\partial J^L}{\partial b_d^L} + \gamma(J^L - J^R)\frac{\partial J^L}{\partial b_d^L} \\
\nabla_{W_d^R} E &= \frac{\partial J^R}{\partial W_d^R} + \gamma(J^L - J^R)(-\frac{\partial J^R}{\partial W_d^R}) \\
\nabla_{b_d^R} E &= \frac{\partial J^R}{\partial b_d^R} + \gamma(J^L - J^R)(-\frac{\partial J^R}{\partial b_d^R})
\end{aligned} \tag{5}$$

We train a SMCAE in a greedy manner where one layer gets trained at a time. The configuration for training one layer of SMCAE is shown in Fig. 3(right). The output of a trained layer is then sent as input to the next layer for training. A fine-tuning is implemented to the entire stacked structure once all layers are trained. Thus, after SMCAE has been trained, to transform new synthetic data, the data is sent to the left channel of the SMCAE $\langle \mathfrak{i}{:}X_s,\ \mathfrak{o}{:}X_r\rangle^L$. We take output of this process as transformed synthetic data.

C. Competitors

As shown in Fig. 4, we compare the SMCAE configuration to three alternative configurations: (1) SMCAE-II which places two separate channels on the structure, i.e. $\langle \mathfrak{i}{:}X_s,\ \mathfrak{o}{:}X_s\rangle^L$ and $\langle \mathfrak{i}{:}X_r,\ \mathfrak{o}{:}X_r\rangle^R$. (2) Stacked autoencoder type-I (SAE-I) which merges the tasks in a single channel stacked autoencoder, with the configuration of :$\langle \mathfrak{i}{:}X_sX_r,\ \mathfrak{o}{:}X_rX_r\rangle$. (3) Stacked autoencoder type-II (SAE-II) which simply transforms source data to target data, and configures as: $\langle \mathfrak{i}{:}X_s,\ \mathfrak{o}{:}X_r\rangle$.

Compared with SAE-I and SAE-II, our two channel structures endow more flexibility. Critically, the single channel models force synthetic data to fit real data, which causes synthetic data to lose information and become less useful for recognition. In contrast, SMCAE can explore 'characters' and 'patterns' common in both synthetic and real data. Intrinsically, SMCAE first encodes both synthetic and real data into common hidden layers which model common information useful for recognition. Then the decoding process transforms the synthetic data to better simulate real data. Although SMCAE-II has the same two branches in the structure, it does not learn such transformation between synthetic data and real data.

IV. Experiments and Results

We first compare SMCAE on the challenging task of face-sketch recognition [29], [31] using the CUFSF dataset. We

[1] δ is a sparsity parameter and is empirically set to 0.05 in all our experiments.

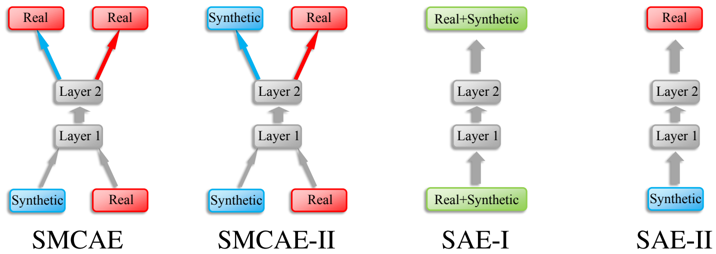

Fig. 4. Illustration of the compared configurations: SMCAE, SMCAE-II, SAE-I and SAE-II.

show that SMCAE is better than alternative configurations. To further validate the efficacy of our framework, we train SMCAE on handwritten digit images and generate synthetic data to simulate real images. We show that the synthetic data can help train classifiers for recognition.

Dataset. We conduct our experiments on two different datasets: (1) The CUFSF dataset [29], [31] containing the photos and sketches of 1194 people with lighting variations. We employ the standard split defined in [29], [31] which selects 500 persons as the training set, and the remaining 694 persons as the testing set. (2) handwritten digits dataset[2] (HWDUCI) containing 5620 instances in total in which 3823 samples are used for training and 1917 samples are used for testing. The handwritten digits from 0 to 9 in this dataset are collected from 43 people: 30 contributed to the training set and the other 13 to the test set. For all experiments, we empirically set the number of hidden layers in SMCAE to two and each layer has 1000 nodes. The same settings are used to make SMCAE, SMCAE-II, SAE-I and SAE-II more comparable.

Evaluation Metrics. We report the following metrics when they are available: (1) F1-score, which is defined as $F1 = 2\cdot(Precision \cdot Recall)\,/\,(Precision + Recall)$. (2)Receiving Operator Characteristic (ROC) curves and VR@0.1%FAR which is the performance of Verification Rate (VR) at 0.1% False Acceptance Rate (FAR). VR@0.1%FAR is a standard evaluation metric and proposed in [29]. (3) Rank-1 recognition accuracy.

Features.(1) Similar to [13], in the CUFSF dataset we use Histogram of Oriented Gradients (HOG). To further reduce the computational cost, the resolution of all photos and sketches is reduced to 50×50. So the cell size of HOG features is set to 3. (2)The HWDUCI dataset uses HOG features with cell size 3.

Classifiers. For CUFSF dataset, nearest-neighbor search with Euclidean metric is used in retrieving the most similar photo to the query sketch. In the handwritten digit classification, a Support Vector Machine (SVM) with RBF kernel[3] is used in the experiments.

A. Results on the CUFSF dataset

In all experiments on this dataset, HOG features of sketch images are first transformed by the SMCAE and then used as queries. We first compare the results of photo-sketch matching using HOG feature transformed by SMCAE, SMCAE-II, SAE-I and SAE-II. The results are reported as ROC curve starting with VR@0.1%FAR. The dissimilarity between a photo and a sketch is computed as the Euclidean distance between descriptors.

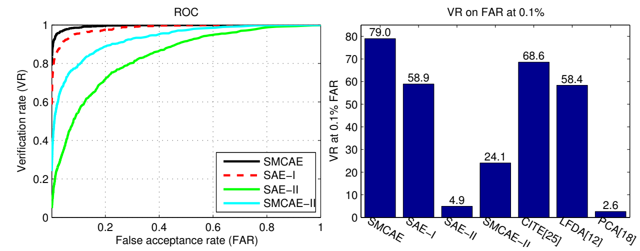

Fig. 5. Results on CUFSF dataset. Left: ROC curve of different methods; Right: VR@0.1%FAR of different methods.

The ROC curves and VR@0.1%FAR are shown in Fig. 5. Clearly, the proposed SMCAE achieves the highest results on AUC values and VR@0.1%FAR accuracy and significantly outperforms the alternative configurations. Note that we also report the state-of-the-art approaches of VR@0.1%FAR including LFDA [13], CITE [31] and classic eigenfaces(PCA)[22]. It is worth noting that in some of previous works, a better result could be obtained by combining multiple features. For example in [31], multiple CITE features generated by a random forest are used to batter matching photos and sketches. Here, to enable a comparison with more fairness, we focus our comparison on matching results obtained by using uncombined feature only.

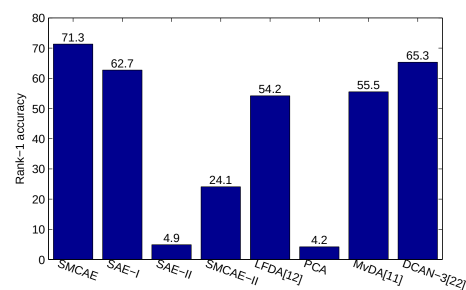

Fig. 6. Rank-1 accuracy of different methods on CUFSF dataset.

There are several reasons why our SMCAE outperform the other approaches. First, compared with SMCAE-II, the configuration of SMCAE involves a task that handles the transformation from synthetic to real data, and thus better eliminates the distance between them. Second, compared with SAE-I, rather than merging two tasks in a single channel SMCAE employs two channels to better clarify each task with the aim of reconstructing the main 'characters' and 'patterns' co-existing in both tasks. Thus synthetic data can be more easily transformed to real data with less error. Finally, SMCAE is better than SAE-II as SMCAE learns features of real data in task $\langle \mathfrak{i}{:}X_r,\ \mathfrak{o}{:}X_r\rangle^R$. These features will better compensate the difference between synthetic data and real data during the transformation.

We further validate the results by using Rank-1 recognition accuracy which is also reported in [12], [28]. The results are shown in Fig. 6. The methods of [12], [28] are comparable to our SMCAE. Method [12] employed a discriminant common

[2]collected from UCI machine learning repository (HWDUCI) [3].
[3]The parameters are cross-validated

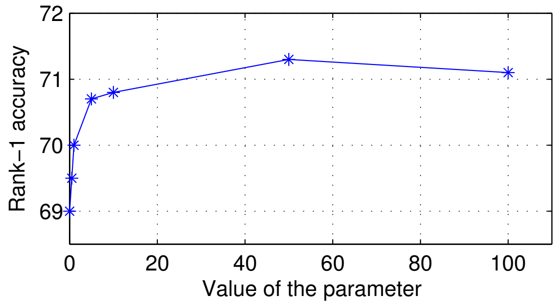

Fig. 7. Rank-1 accuracy by setting different value for γ in Eq 3. Rank-1 accuracy by setting γ equal to 0, 0.5, 1, 5, 10, 50, and 100 are shown in the figure.

subspace to maximize the between-class variations and minimize the within-class variations. Method [28] used a structure composed of two autoencoders. As can be seen Fig. 6, the SMCAE outperforms all other methods.

Parameter Validation in Eq. 3. To validate the significance of Ψ in Eq. 3. We set γ with different values and report the rank-1 accuracy in Fig 7. Particularly, when γ is 0, it takes 2 times longer for SMCAE to converge compared with $\gamma = 50$ used in this work, Further with $\gamma = 0$ the rank-1 accuracy is dropped by more than 2%. This validates the importance of term Ψ discussed in Sec. 3.2.

Qualitative results. Some qualitative results are shown in Fig. 8. It shows that a sketch HOG transformed by our SMCAE is more similar to the ground truth photo HOG.

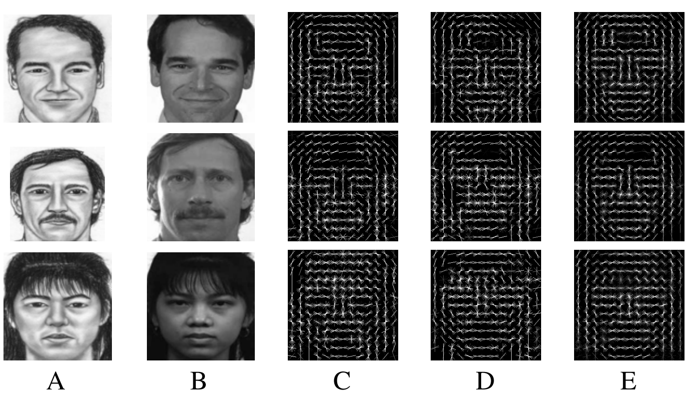

Fig. 8. Example of HOG features transformed by SMCAE. A: Sketch. B: Photo. C: Original sketch HOG. D: Photo HOG. E: Transformed sketch HOG.

B. Handwritten Digit Recognition

Generating synthetic data. A synthetic version of each real character is generated as a variant of a centralized model learned from real characters. The centralized model of digit is shaped by control points $C = \{c_i\}_{i=1}^{n}$ settled on the boundary of the digit. A technique called migration is used to locate corresponding control points on each real digit image. A synthetic digit image then could be generated by filling areas closed by the control points [4]. Examples of generated synthetic digits are shown in Fig. 9. To generate more synthetic data which is used to train the classifier once transformed by the trained SMCAE, we assume that locations of the control points follow a multivariate normal distribution $C \sim N(\mu, \Sigma)$ with μ and Σ estimated using control points on the synthetic digit images. For each digit, $3,000$ new synthetic images are generated by randomly drawing samples from $N(\mu, \Sigma)$.

[4]Please refer to supplementary material for details.

Fig. 9. Illustration of real digit images (upper row) and corresponding synthetic versions (lower row).

We compare our SMCAE with SMCAE-II, SAE-I, SAE-II, LeNet-5 [15] and the best results [1] reported on this data set. The classification performance is evaluated by F1-score. A Support Vector Machine (SVM) classifier with RBF kernel is used in the experiments. For SMCAE, SMCAE-II, SAE-I and SAE-II in the test, real training data together with transformed synthetic data are used to train the SVM.

As shown in Fig. 10 (left), the SVM classifier with our SMCAE is better than all the alternative methods. This validates the effectiveness of our framework in generating synthetic data to better help training a classifier.

To further demonstrate how transformed synthetic data improve the classification results, we conducted more evaluations by training classifiers using different combinations of training sets in Fig. 10 (right). Particularly, four combinations of training sets are used. First, to have a performance baseline of SVM, we trained the SVM using real data only. To investigate how much improvement we could obtain in classification using a SVM trained by transformed synthetic data, we compare a SVM trained by synthetic data and transformed synthetic data respectively. The best performance is obtained with a SVM trained by real data together with transformed synthetic data.

With more synthetic training data generated by SMCAE, we gain a large margin of improvement in the classification. We notice that we can get the same result (0.989) by using Transformed synthetic and Real+Transformed Synthetic separately in Fig. 10 (right), which highlights the effectiveness of SMCAE in transforming synthetic data to simulate real data.

Finally, it is interesting to evaluate how the amount of synthetic data affects the classification results. We increasingly add more transformed synthetic data (from 300 to 3,300 samples) when training the SVM. The classification results are reported in Fig. 11. The curve shows an ascending trend when adding more samples, which means that all transformed synthetic data added to this test are highly effective and useful in the classification.

V. Conclusion

In this paper we identify the synthetic gap problem. To solve this problem, we propose a novel Stacked Multichannel autoencoder (SMCAE) model. SMCAE has multiple channels in its structure and is an extension of a standard autoencoder. We show that SMCAE not only bridges the synthetic gap between real data and synthetic data, but also jointly learns from both real and synthetic data.

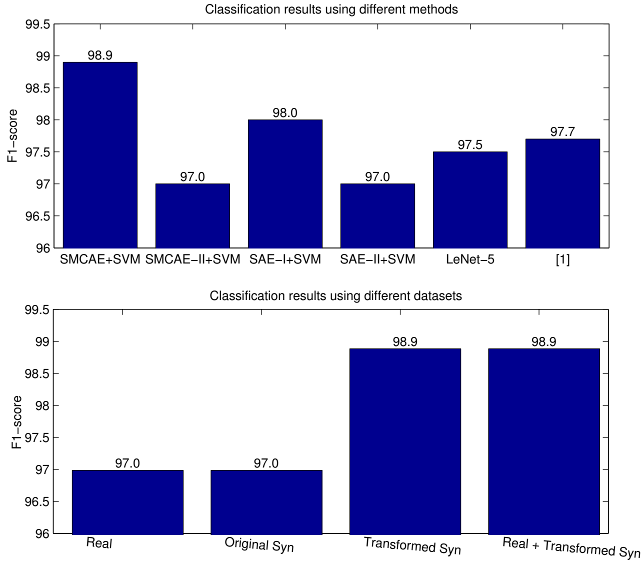

Fig. 10. Comparison of classification results in F1-score of different methods (left) and different training datasets (right). In the left figure, all the methods in the test used the same training dataset which combines original real data and transformed synthetic data.

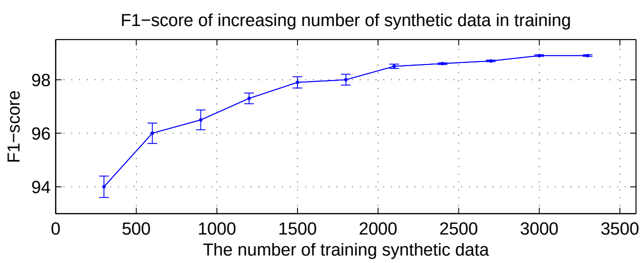

Fig. 11. F1-scores and corresponding standard deviation when increasing the number of synthetic data used in training is shown.

References

[1] Fevzi Alimoglu and Ethem Alpaydin. Combining multiple representations and classifiers for handwritten digit recognition. In *ICDAR*, 1997.

[2] Fares Alnajar, Zhongyu Lou, Jose Alvarez, and Theo Gevers. Expression-invariant age estimation. In *BMVC*, 2014.

[3] Kevin Bache and Moshe Lichman. UCI machine learning repository, 2013.

[4] Gungor Bal, Gady Agam, Ophir Frieder, and Gideon Frieder. Interactive degraded document enhancement and ground truth generation. In *Electronic Imaging 2008*. International Society for Optics and Photonics, 2008.

[5] Pierre Baldi. Autoencoders, unsupervised learning, and deep architectures. *Unsupervised and Transfer Learning Challenges in Machine Learning, Volume 7*, page 43, 2012.

[6] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Mach. Learn.*, 2010.

[7] Yoshua Bengio. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[8] Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. *Unsupervised and Transfer Learning Challenges in Machine Learning, Volume 7*, page 19, 2012.

[9] Minmin Chen, Zhixiang Xu, Kilian Q. Weinberger, and Fei Sha. Marginalized denoising autoencoders for domain adaptation. In *International Conference on Machine Learning*, 2012.

[10] Jun Deng, Zixing Zhang, Erik Marchi, and Bjorn Schuller. Sparse autoencoder-based feature transfer learning for speech emotion recognition. In *Affective Computing and Intelligent Interaction (ACII)*, 2013.

[11] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *ICML*, 2011.

[12] Meina Kan, Shiguang Shan, Haihong Zhang, Shihong Lao, and Xilin Chen. Multi-view discriminant analysis. In *ECCV*, 2012.

[13] Brendan F Klare, Zhifeng Li, and Anil K Jain. Matching forensic sketches to mug shot photos. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(3):639–646, 2011.

[14] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE TPAMI*, 2013.

[15] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[16] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE TKDE*, 2010.

[17] Leonid Pishchulin, Arjun Jain, Christian Wojek, Mykhaylo Andriluka, Thorsten Thormählen, and Bernt Schiele. Learning people detection models from few training samples. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1473–1480. IEEE, 2011.

[18] Adria Ruiz, Joost Van de Weijer, and Xavier Binefa. Regularized multi-concept mil for weakly-supervised facial behavior categorization. In *BMVC*, 2014.

[19] Kanoksri Sarinnapakorn and Miroslav Kubat. Combining subclassifiers in text categorization: A dst-based solution and a case study. *IEEE TKDE*, 2007.

[20] Nitish Srivastava and Ruslan R Salakhutdinov. Multimodal learning with deep boltzmann machines. In *Advances in neural information processing systems*, pages 2222–2230, 2012.

[21] Baochen Sun and Kate Saenko. From virtual to reality: Fast adaptation of virtual object detectors to real domains. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014.

[22] Matthew A Turk and Alex P Pentland. Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, pages 586–591. IEEE, 1991.

[23] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

[24] Tamás Varga and Horst Bunke. Effects of training set expansion in handwriting recognition using synthetic data. In *In 11th Conf. of the International Graphonomics Society*. Citeseer, 2003.

[25] Tamás Varga and Horst Bunke. Comparing natural and synthetic training data for off-line cursive handwriting recognition. In *Frontiers in Handwriting Recognition, 2004. IWFHR-9 2004. Ninth International Workshop on*, pages 221–225. IEEE, 2004.

[26] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

[27] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2011.

[28] Wen Wang, Zhen Cui, Hong Chang, Shiguang Shan, and Xilin Chen. Deeply coupled auto-encoder networks for cross-view classification. *arXiv preprint arXiv:1402.2031*, 2014.

[29] Xiaogang Wang and Xiaoou Tang. Face photo-sketch synthesis and recognition. *IEEE TPAMI*, 2009.

[30] Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. Feature hashing for large scale multitask learning. In *ICML*, 2009.

[31] Wei Zhang, Xiaogang Wang, and Xiaoou Tang. Coupled information-theoretic encoding for face photo-sketch recognition. In *CVPR*, pages 513–520. IEEE, 2011.

[32] Fan Zhu, Ling Shao, and Jun Tang. Boosted cross-domain categorization. In *British Machine Vision Conference*, 2014.