Special Paper
방송공학회논문지 제26권 제7호, 2021년 12월 (JBE Vol. 26, No. 7, December 2021)
https://doi.org/10.5909/JBE.2020.26.7.844
ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

Class Specific Autoencoders Enhance Sample Diversity

Teerath Kumar[a)], Jinbae Park[a)], Muhammad Salman Ali[a)], A F M Shahab Uddin[a)], and Sung-Ho Bae[a)‡]

Abstract

Semi-supervised learning (SSL) and few-shot learning (FSL) have shown impressive performance even then the volume of labeled data is very limited. However, SSL and FSL can encounter a significant performance degradation if the diversity gap between the labeled and unlabeled data is high. To reduce this diversity gap, we propose a novel scheme that relies on an autoencoder for generating pseudo examples. Specifically, the autoencoder is trained on a specific class using the available labeled data and the decoder of the trained autoencoder is then used to generate N samples of that specific class based on N random noise, sampled from a standard normal distribution. The above process is repeated for all the classes. Consequently, the generated data reduces the diversity gap and enhances the model performance. Extensive experiments on MNIST and FashionMNIST datasets for SSL and FSL verify the effectiveness of the proposed approach in terms of classification accuracy and robustness against adversarial attacks.

Keywords: Pseudo examples, semi-supervised learning, few shot learning, adversarial attacks

I. Introduction

Deep learning neural networks have shown impressive performance in multiple do- mains i.e., image classification[1-5], text classification[7,6,8,9] and audio classification[10-12]. Supervised learning methods purely rely on labeled data and getting labeled data is time-consuming and tedious task [36,37] and labeled data is rarely available in large groups however, unlabeled data is available on a large scale [36,37,40,26,27].

One of the most popular techniques used for assigning pseudo labels on unlabeled data is Semi-Supervised Learning (SSL). In SSL, the model is first trained on the limited available labeled data and the trained model is then used to generate pseudo labels for the unlabeled data. The pseudo labeled data is then combined with the labeled data to retrain the model. If the level of diversity gap between the labeled and unlabeled data is low and the pseudo labels

a) Department of Computer Science and Engineering, Kyung Hee University
‡ Corresponding Author : Sung-Ho Bae
E-mail: shbae@khu.ac.kr
Tel: +82-31-201-2593
ORCID: https://orcid.org/0000-0002-3389-1159
※ This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (2018R1C1B3008159).
· Manuscript received October 8, 2021; Revised December 6, 2021; Accepted December 6, 2021.

of unlabeled data are of high confidence, the model can offer impressive performance[15-17].

However, SSL based techniques are not successful when the diversity of the dataset is high, resulting in incorrect pseudo labeling which in turns degrades the model accuracy. Few-shot learning (FSL) is another type of machine learning approach to learn from limited samples. It classifies new data with supervised information when limited training samples are available. Similar to the SSL, the problem of data diversity gap also exists in FSL[34]. To address the above mentioned problem, we propose a novel way to generate diverse samples that reduces the diversity gap, thereby improving the model accuracy for both SSL and FSL. We first train an autoencoder for a specific class and pass random noise to the decoder for generating samples of the respective class where the noise is sampled from a normal distribution. Using this method we can generate N samples for a respective class. These generated samples when combined with the original labeled data can improve the model performance for SSL and FSL as well as increase the robustness against adversarial attacks. Our main contributions in this work are as follows:

- We propose a novel way of generating new samples using class specific autoencoders which are highly capable of producing diverse samples to reduce the diversity gap of the labeled and unlabeled data
- Our approach improves the performance over previous SSL and FSL approaches
- Samples generated by our approach show more robustness against adversarial attacks
- We perform extensive experiments on MNIST and Fashion MNIST dataset to evaluate the effectiveness of the proposed approach

II. Related Work

1. Autoencoder

An autoencoder is a kind of neural network that is designed to encode input into a compact and meaningful representation, then decode it so that the reconstructed input

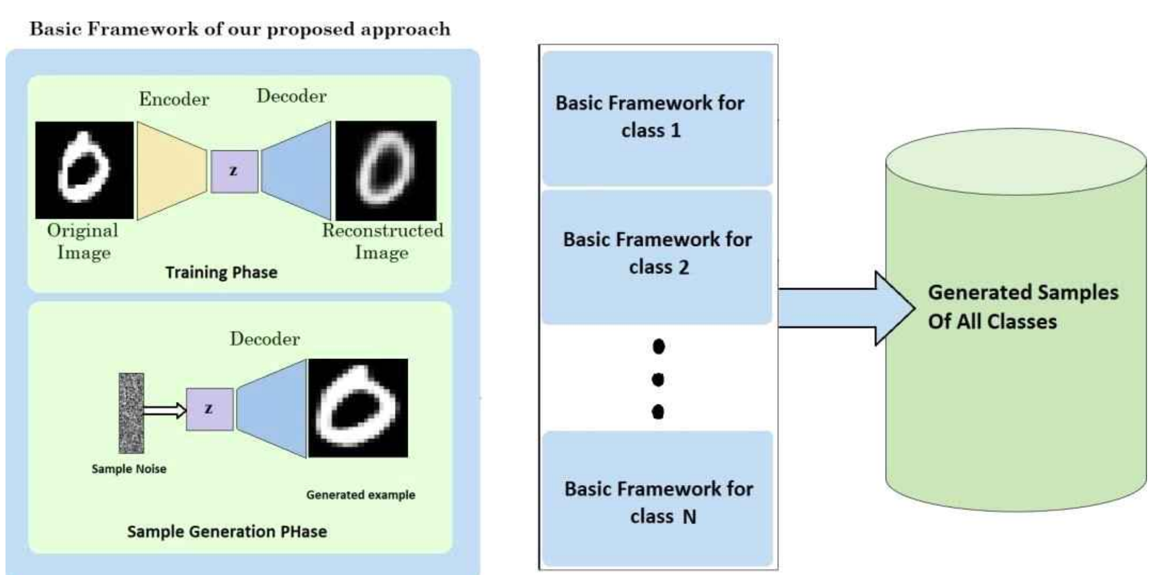

Fig. 1. Two-Step architecture, Step1: We train an auto-encoder on a selected class, Step 2: Sample random noise N times and pass it to the decoder to get new N samples of the respective class, Repeat the process for remaining classes

is as close to the original as possible[13]. Autoencoders are used for various purposes i.e., data compression [13], super resolution[14], data generation[13] and many more. In order to bridge the diversity gap between labeled and unlabeled data, several studies employed data augmentation[19-23]. A few works also exploited autoencoder for data augmentation[40,24,5,27]. Tachibana et. al.[40] divided the latent representation of autoencoder into two parts; one is used for style learning and the other for content learning. During training, the autoencoder is regularized by learning a prior distribution on both parts to make them separate. Jorge et. al.[24] suggests that any kind of transformation on data is problem independent. And based on that the author applied transformation on latent space rather than on data and created samples that are helpful in semi-supervised learning.

Elbattah et. al.[25] explored variational autoencoder (VAE) for image-based augmentation for eye-tracking data. Recently, semi-supervised based encoder-decoder[27] architecture has been introduced that can generate new samples using Gaussian noise for a given mean and standard deviation. Using this architecture data is generated which is then combined with already labeled data for training the classifier. However, their performance is limited due to not having more diversity. In contrast, the proposed approach generates the noisy samples to increase diversity, which improves performance.

2. Semi-supervised learning

SSL is defined as a method to utilize the unlabeled data by exploiting the limited amount of labeled data. Numerous different methodologies have been proposed to increase the performance of SSL based methods[41,30,18]. Lee et. al.[41] used both labeled and unlabeled data simultaneously to train the model in a supervised manner. Pseudo-labels to unlabeled data are assigned by picking maximum probability predictions. Arazo et. al.[30] generated soft pseudo-labels using a trained network and showed that naive pseudo-labeling easily overfits the wrong pseudo-labels due to confirmation bias. Sohn et. al.[18] produced pseudo-labels based on model predictions on unlabeled images that are weakly augmented. The pseudo-label is only considered for a specific image if the model makes a high-confidence prediction. After pseudo-labeling, the model is trained to predict pseudo-label of the same image that is strongly augmented.

3. Few-Shot Learning

When only labeled data is available in a limited quantity, FSL is used to effectively train a model. Various techniques have been introduced[35,31,32,26] to get impressive performance using FSL. Asadulaev et. al.[35] trained a network for each class separately and then distilled it to a small network and used a linear function for both large and small networks. Hariharan et. al.[32] proposed a two phases approach. Where the model learns representations of features on the classes having a large number of training examples, and then in the low-shot learning phase, the model learns using only a few number of examples. Kimura et. al.[26] used a large model as a reference, trained on a few number of samples, and then knowledge distillation is performed from large to small model.

Table 1. Autoencoder architecture used in experiments

layer type	dimensions	comments
input layer	784	
fully connected	500	relu activation
fully connected	500	relu activation
fully connected	2000	relu activation
latent layer fully connected	10	
fully connected	2000	relu activation
fully connected	500	relu activation
fully connected	500	relu activation
output layer	784	sigmoid activation

Previous works[13,19,27] used autoencoder to generate all class(es) samples during training as data augmentation and it is difficult to distinguish the samples of specific class,

```
Algorithm: Training-Samples-Generation(X,Y,N)
Input: X : selected samples only
       Y : labels of the selected samples
       N: Number of samples per class to generate
Output: Generated data
Output=[ ]
for Choose i ∈ UniqueClasses(y) do
    model = Autoencoder() // Creating autoencoder
    Train(Autoencoder, X_i) // Training autuecnoder for class i samples
    GeneratedSamples = GenerateSamples(Autoencoder, N) // Generate N
        samples of class i
    Output = Output ∪ GeneratedSamples
```

Algorithm 1: Algorithm to generate samples per class

so we use autoencoder as data augmentation in a very novel way to generate the samples that are easy to distinguish and helps in improving the performance.

Ⅲ. Method

We divide our approach into two phases: (i) training phase; and (ii) sample generation phase. In the training phase, an equal number of labeled samples are randomly taken from all of the classes. After that M number of class specific autoencoders are trained where the number of classes Ci = 1, 2, ...,M. On the other hand, in the sample generation phase, the decoders are first detached from the autoencoders. Then N random noises are passed to each of the decoders to generate the subsequent N number of samples from each of the classes. Fig. 1 presents the overall architecture of the proposed approach and algorithm 1 explains the algorithm of the proposed approach.

1. Training Phase

We utilize the autoencoder following the state-of-the-art work as shown in Table 1. The encoder comprises 5 layers with the dimension of 784, 500, 500, 2000 and 10 and the decoder comprises 5 layers with the dimension of 10, 2000, 500,500 and 784. We use stochastic gradient descent for training[43] with Mean Square Error (MSE) as a loss function as shown in Eq. 1

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2 \tag{1}$$

where y_i is the original input and $\tilde{y}_i$ is reconstructed input and n is the number of input used in a batch.

Table 2. CNN used for SSL. Taken from [27]

layer type	dimensions	Comments
input layer	(28, 28, 1)	
CNN	(28, 28, 64) 64 are filters	kernel 3 × 3; stride 1
BN	(28, 28, 64) 64 are filters	with relu activation
CNN	(28, 28, 64) 64 are filters	kernel 3 × 3; stride 1
BN	(28, 28, 64) 64 are filters	with relu activation
CNN	(14, 14, 64) 64 are filters	kernel 3 × 3; stride 2
BN	(14, 14, 64) 64 are filters	with relu activation
Flatten	12544	
fully connected	512 (Connections)	
BN	512	with relu activation
Dropout	512	dropout rate = 0.5
fully connected	10 (Classes)	softmax activation; class prediction

Table 3. CNN used for FSL. Taken from [26]

layer type	dimensions	comments
input	input layer	(28, 28, 1)
CNN	(28, 28, 16) 16 are filters	kernel 3 × 3; stride 2
BN	(28, 28, 16) 16 are filters	with pRelu activation
Dropout	512	dropout rate = 0.5
CNN	(28, 28, 16) 16 are filters	kernel 3 × 3; stride 1
BN	(28, 28, 16) 16 are filters	with pRelu activation
Dropout	Dropout	dropout rate = 0.5
CNN	(28, 28, 32)	kernel 3 × 3; stride 1
BN	(28, 28, 32)	with pRelu activation
Dropout	Dropout	dropout rate = 0.5
fully connected	1000 (Connections)	
BN	(28, 28, 132)	with pRelu activation
Dropout	DropOut	dropout rate = 0.5
fully connected	10 (Classes)	softmax

2. Sample Generation Phase

Once the autoencoder is trained for a specific class, we detach the decoder part of the network. Then we pass N random noise to the decoder to generate N number of samples from the respective class. Where the random noise is sampled from a Gaussian distribution with mean $\mu = 0$ and standard deviation $\sigma = 0.1$, reason of using mean and standard deviation values is that they are standard to use, as follows:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{2}$$

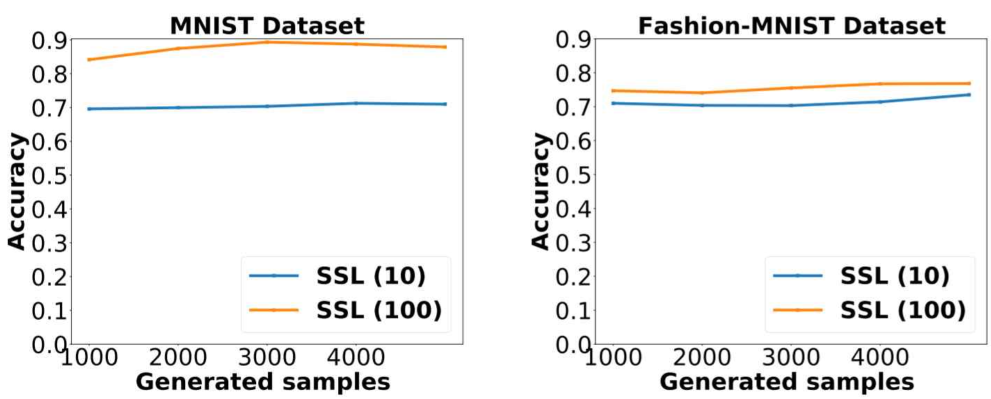

Fig. 2. Number of samples generated vs accuracy for each class on MNIST and FashionMNIST datasets

Table 4. Performance of the proposed method compared to the previous approaches. Best and 2nd best results are respectively shown in blue and red color

	MNIST		FashionMNIST	
Semi-supervised learning				
Models	100	1000	100	1000
CCNs	0.765±0.003	0.89 ± 0.02	0.66±0.01	0.803±0.001
CCNs(MS)	0.811 ± 0.006	0.945 ± 0.001	0.724±0.008	0.836±0.001
CNNs(Ours)	0.78± 0.019	0.89±0.006	0.725 ± 0.021	0.799 ± 0.014

Few-Shot learning	MNIST					FashionMNIST				
Models	10	20	50	100	200	10	20	50	100	200
NN	37.9	46.0	66.0	78.3	86.7	39.3	47.9	58.3	64.9	71.3
GP	39.9	51.6	64.6	73.2	80.0	44.6	52.4	59.9	65.7	71.4
Imt	43.5	51.2	67.7	78.1	86.1	43.6	50.9	60.0	67.3	72.5
Imt, opt	44.1	53.7	70.0	79.5	86.7	41.2	49.7	60.1	67.3	72.2
Imt, opt, fd	44.1	53.9	70.4	80.0	86.6	44.8	52.7	62.1	67.3	72.5
Our	46.3	54.3	59.4	67.4	76.4	48.20	56.10	58.8	65.8	69.49

where $p(x)$ is a random noise vector of dimension 1 x 10. The generated samples are then used to train SSL and FSL approaches.

Ⅳ. Experiments

1. Datasets

To check the effectiveness of our proposed approach, we perform experiments on MNIST[50] and FashionMNIST[51] datasets. MNIST is a 10 digits (0 to 9) dataset, which consists of 60000 training images and 10000 test images where each image has a size of 28x28 pixels. And FashionMNIST is a classification dataset for clothes and accessories. It also consists of 10 classes and contains 60000 training images and 10000 test images of size 28 × 28 pixels each.

2. Experimental Setup

We train the autoencoder with a learning rate of 0.1 for 250 epochs. For classification, we use 3 layers convolutional neural network (CNN) architecture for SSL as shown in Table 2 and for FSL we used 3 layers CNN as shown in Table 3. Reason for choosing different model architecture for SSL and FSL is only for fair comparison purposes. For both SSL and FSL, the model is trained for 100 epochs using a learning rate of 0.01 with Adadelta [33] optimizer. The results are reported on average of five runs accuracy. In the proposed method, N is a hyperparameter

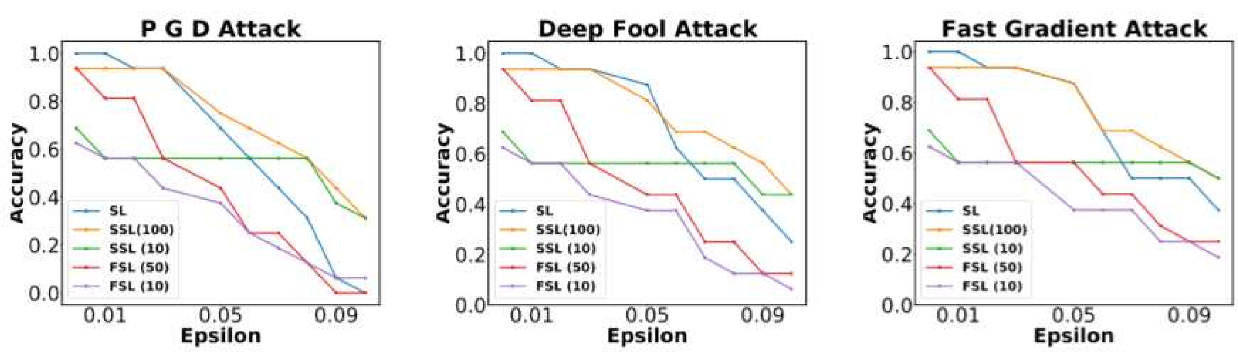

Fig. 3. Comparison of different adversarial attacks on MNIST dataset. We compare fully super- vised learning with SSL, and FSL using the proposed approach

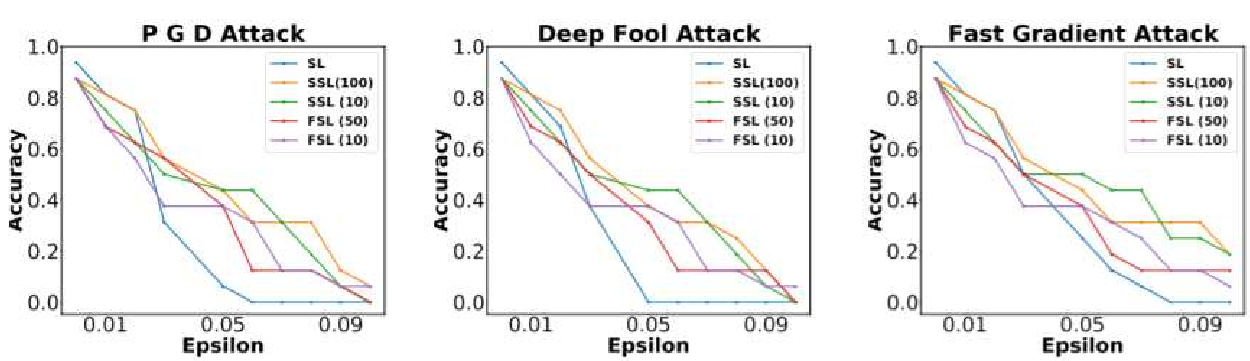

Fig. 4. Comparison of different adversarial attacks on the FashionMNIST dataset. We compare fully supervised learning with SSL, and FSL using the proposed approach

that defines the number of samples to be generated by the decoder. It is worth noting that the model performance varies with the value of N. As a result, it is crucial to choose the optimal value of N. We generate samples ranging from 1000 to 5000 with an interval of 1000. For each SSL case, we choose the optimal value of N and repeat the experiments for five times. Optimal N for FashionMNIST and MNIST can be seen in Fig 2. In the case of FSL, we generate 10,000 samples by following state-of-the-art methods [26].

3. Results

We performed experiments for SSL and FSL, and for a fair comparison, we cited the results from relevant work using the same architecture for SSL and for FSL as shown in Table 2 and Table 3, respectively. In Table 4 for FSL results, we used abbreviation terms due to limited space, Imt=Imitation, opt=optimize, fd=fidelity. Here the term "imitation" represents the loss that is introduced to reduce overfitting between reference and target model during distillation; "optimize" that is used to select samples that increase the diversity of the data; and "fidelity" selects high confident samples based on their weights. Table 4 presents the performance comparison of the proposed method with other state-of-the-art methods where the term "CNNs" refers to the CNN that is trained on selected labels as a purely supervised learning and "SSL(MS)" refers to the semi-supervised learning with samples generated by autoencoder alongside mean and standard deviation layer. In SSL, 100 and 1000 labeled samples are used for both the MNIST and FashionMNIST datasets. The proposed method outperforms the FashionMNIST dataset on 100 samples cases and achieves competitive results on other cases. In FSL, 10, 20, 50, 100 and 200 supervised samples of MNIST and FashionMNIST are used. Our approach outperforms previous FSL technique in four cases 10 and 20 supervised samples of MNIST and FashionMNIST even previous FSL technique applied imitation, optimization and fidelity. For the rest of the cases, the proposed approach achieves competitive results.

Furthermore, we check the robustness of our approach against adversarial attacks on models trained using SSL and FSL. Here we apply three different adversarial attacks

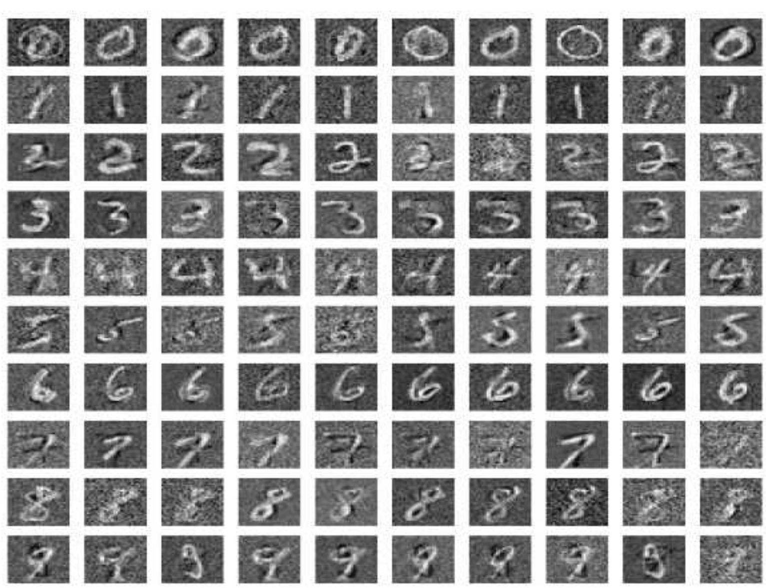

Fig. 5. Samples generated by class specific autoencoder

i.e., Projected Gradient Descent Attack (PGDA)[46,45], Fast Gradient Attack[46] and Deep Fool Attack that is a simple and fast gradient-based adversarial attack[46]. Fig. 3 and Fig. 4 present the performance comparison of the proposed method on MNIST and FashionMNIST datasets, respectively. In Fig. 3 and Fig. 4, SL indicates the supervised learning on the entire labeled dataset, SSL(n) indicates semi-supervised learning with n labeled samples per class and FSL(n) indicates model using few-shot learning trained with n samples per class. Both of the figures show that the performance of SL falls drastically with increased levels of attack indicated by the value of epsilon. In contrast, the performance of SSL and FSL using our approach is more stable as compared to SL. It is because samples generated by the proposed method include noisy examples as shown in Fig. 5, which in turn increase the robustness against adversarial attacks.

Ⅴ. Conclusion

In this paper, we proposed a novel way of generating pseudo examples using an autoencoder that is trained on a limited amount of labeled data of a specific class. The trained decoder is then used as a generator to produce new samples based on random noise, sampled from a normal distribution. The generated samples are diverse in nature that reduces the diversity gap, thereby improving the model performance in both of the SSL and FSL approaches and outperforms the state-of-the-art methods. Specifically, the proposed method achieves the best classification accuracy of 89.0% and 72.5% on MNIST and FashionMNIST dataset, respectively when applied for SSL approach. Also, it achieves the best classification accuracy of 46.3% and 48.2% on MNIST and FashionMNIST dataset, respectively when applied in FSL approach. Furthermore, the proposed method helps to increase the model robustness against adversarial attacks compared to the supervised learning approach. One key disadvantage of the proposed approach is, it takes more training time during sample generation, as we are training autoencoder for each class separately, so it depends on the number of classes. Future work direction is how to make training faster?

References

[1] Krizhevsky, A., Sutskever, I. & Hinton, G. ImageNet classification with deep convolutional neural networks. Communications Of The ACM. 60, 84-90 (2017)

[2] Xin, M. & Wang, Y. Research on image classification model based on deep convolution neural network. EURASIP Journal On Image And Video Processing. 2019, 1-11 (2019)

[3] Ali, M., Iqbal, T., Lee, K., Muqeet, A., Lee, S., Kim, L. & Bae, S. ERDNN: Error-Resilient Deep Neural Networks With a New Error Correction Layer and Piece-Wise Rectified Linear Unit. IEEE Access. 8 pp. 158702-158711 (2020)

[4] Ali, M. & Bae, S. Soft Error Adaptable Deep Neural Networks. Proceedings Of The Korean Society Of Broadcast Engineers Conference. pp. 241-243 (2020)

[5] Khan, L., Saad, W., Han, Z., Hossain, E. & Hong, C. Federated learning for internet of things: Recent advances, taxonomy, and open challenges. IEEE Communications Surveys & Tutori- als. (2021)

[6] Jo, T. NTC (Neural Text Categorizer): neural network for text categorization. International Journal Of Information Studies. 2, 83-96 (2010)

[7] Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L. & Brown, D. Text classification algorithms: A survey. Information. 10, 150 (2019)

[8] Li, Q., Peng, H., Li, J., Xia, C., Yang, R., Sun, L., Yu, P. & He, L. A survey on text classifica- tion: From shallow to deep learning. ArXiv Preprint ArXiv:2008.00364. (2020)

[9] Ullah, I., Khan, S., Imran, M. & Lee, Y. RweetMiner: Automatic identification and catego- rization of help requests on twitter during disasters. Expert Systems With Applications. 176 pp. 114787 (2021)

[10] Sinha, H., Awasthi, V. & Ajmera, P. Audio classification using braided convolutional neural networks. IET Signal Processing. 14, 448-454 (2020)

[11] Park, J., Kumar, T. & Bae, S. Search of an Optimal Sound Augmentation Policy for Environ- mental Sound Classification with Deep Neural Networks. Proceedings Of The Korean Society Of Broadcast Engineers Conference. pp. 18-21 (2020)

[12] Kumar, T., Park, J. & Bae, S. Intra-Class Random Erasing (ICRE) augmentation for audio classification. Proceedings Of The Korean Society Of Broadcast Engineers Conference. pp. 244-247 (2020)

[13] Bank, D., Koenigstein, N. & Giryes, R. Autoencoders. ArXiv Preprint ArXiv:2003.05991. (2020)

[14] Zeng, K., Yu, J., Wang, R., Li, C. & Tao, D. Coupled deep autoencoder

for single image super-resolution. IEEE Transactions On Cybernetics. 47, 27-37 (2015)
[15] Zhai, X., Oliver, A., Kolesnikov, A. & Beyer, L. S4l: Self-supervised semi-supervised learn- ing. Proceedings Of The IEEE/CVF International Conference On Computer Vision. pp. 1476- 1485 (2019)
[16] Ebert, S. Semi-supervised learning for image classification. (2012)
[17] Luo, Y., Zhu, J., Li, M., Ren, Y. & Zhang, B. Smooth neighbors on teacher graphs for semi- supervised learning. Proceedings Of The IEEE Conference On Computer Vision And Pattern Recognition. pp. 8896-8905 (2018)
[18] Sohn, K., Berthelot, D., Li, C., Zhang, Z., Carlini, N., Cubuk, E., Kurakin, A., Zhang, H. & Raffel, C. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. ArXiv Preprint ArXiv:2001.07685. (2020)
[19] Shorten, C. & Khoshgoftaar, T. A survey on image data augmentation for deep learning. Journal Of Big Data. 6, 1-48 (2019)
[20] Wang, X., Wang, K. & Lian, S. A survey on face data augmentation for the training of deep neural networks. Neural Computing And Applications. pp. 1-29 (2020)
[21] Sinha, R., Lee, S., Rim, M. & Hwang, S. Data augmentation schemes for deep learning in an indoor positioning application. Electronics. 8, 554 (2019)
[22] Luo, C., Zhu, Y., Jin, L. & Wang, Y. Learn to augment: Joint data augmentation and network optimization for text recognition. Proceedings Of The IEEE/CVF Conference On Computer Vision And Pattern Recognition. pp. 13746-13755 (2020)
[23] Gontijo-Lopes, R., Smullin, S., Cubuk, E. & Dyer, E. Tradeoffs in Data Augmentation: An Empirical Study. International Conference On Learning Representations. (2020)
[24] Jorge, J., Vieco, J., Paredes, R., Sánchez, J. & Benedi, J. Empirical Evaluation of Variational Autoencoders for Data Augmentation. VISIGRAPP (5: VISAPP). pp. 96-104 (2018)
[25] Elbattah, M., Loughnane, C., Guérin, J., Carette, R., Cilia, F. & Dequen, G. Variational Au-toencoder for Image-Based Augmentation of Eye-Tracking Data. Journal Of Imaging. 7, 83 (2021)
[26] Kimura, A., Ghahramani, Z., Takeuchi, K., Iwata, T. & Ueda, N. Few-shot learning of neural networks from scratch by pseudo example optimization. British Machine Vision Conference 2018, BMVC 2018. (2019)
[27] Felix Berkhahn & Dominik Geißler , R. AUGMENTING VARIATIONAL AUTOEN-CODERS WITH SPARSE LABELS: A UNIFIED FRAMEWORK FOR UNSUPERVISED, SEMI-(UN)SUPERVISED, AND SUPERVISED LEARNING. ArXiv E-prints. (2019,11)
[28] Tachibana, R., Matsubara, T. & Uehara, K. Semi-supervised learning using adversarial net-works. 2016 IEEE/ACIS 15th International Conference On Computer And Information Sci- ence (ICIS). pp. 1-6 (2016)
[29] Haiyan, W., Haomin, Y., Xueming, L. & Haijun, R. Semi-supervised autoencoder: A joint approach of representation and classification. 2015 International Conference On Computa- tional Intelligence And Communication Networks (CICN). pp. 1424-1430 (2015)
[30] Arazo, E., Ortego, D., Albert, P., O'Connor, N. & McGuinness, K. Pseudo-labeling and confirmation bias in deep semi-supervised learning. 2020 International Joint Conference On Neural Networks (IJCNN). pp. 1-8 (2020)
[31] Wang, Y., Yao, Q., Kwok, J. & Ni, L. Generalizing from a few examples: A survey on few-shot learning. ACM Computing Surveys (CSUR), Vol.53, No.63, pp.1-34, May 2021.
[32] Hariharan, B. & Girshick, R. Low-shot visual recognition by shrinking and hallucinating features. Proceedings Of The IEEE International Conference On Computer Vision. pp. 3018- 3027 (2017)
[33] Zeiler, M. Adadelta: an adaptive learning rate method. ArXiv Preprint ArXiv:1212.5701.(2012)
[34] Jiang, S., Zhu, Y., Liu, C., Song, X., Li, X. & Min, W. Dataset Bias in Few-shot Image Recognition. ArXiv Preprint ArXiv:2008.07960. (2020)
[35] Arip Asadulaev, A. Interpretable Few-Shot Learning via Linear Distillation. ArXiv E-prints.(2019,10)
[36] Li, W., Wang, Z., Li, J., Polson, J., Speier, W. & Arnold, C. Semi-supervised learning based on generative adversarial network: a comparison between good GAN and bad GAN approach. ArXiv Preprint ArXiv:1905.06484. (2019)
[37] Kingma, D., Mohamed, S., Rezende, D. & Welling, M. Semi-supervised learning with deep generative models. Advances In Neural Information Processing Systems. pp. 3581-3589 (2014)
[38] Weston, J., Ratle, F., Mobahi, H. & Collobert, R. Deep learning via semi-supervised embedding. Neural Networks: Tricks Of The Trade. pp. 639-655 (2012)
[39] Li, Y., Pan, Q., Wang, S., Peng, H., Yang, T. & Cambria, E. Disentangled variational auto- encoder for semi-supervised learning. Information Sciences. 482 pp. 73-85 (2019)
[40] Tachibana, R., Matsubara, T. & Uehara, K. Semi-supervised learning using adversarial networks. 2016 IEEE/ACIS 15th International Conference On Computer And Information Science (ICIS). pp. 1-6 (2016)
[41] Lee, D. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. Workshop On Challenges In Representation Learning, ICML. 3 pp. 2 (2013)
[42] He, K., Zhang, X., Ren, S. & Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. Proceedings Of The IEEE International Conference On Computer Vision. pp. 1026-1034 (2015)
[43] Robbins, H. & Monro, S. A stochastic approximation method. The Annals Of Mathematical Statistics. pp. 400-407 (1951)
[44] Hussain, S. Resources for Urdu language processing. Proceedings Of The 6th Workshop On Asian Language Resources. (2008)
[45] Goodfellow, I., Shlens, J. & Szegedy, C. Explaining and harnessing adversarial examples.ArXiv Preprint ArXiv:1412.6572. (2014)
[46] Rauber, J., Brendel, W. & Bethge, M. Foolbox: A python toolbox to benchmark the robust- ness of machine learning models. ArXiv Preprint ArXiv:1707.04131. (2017)
[47] Plötz, T. & Fink, G. Markov models for offline handwriting recognition: a survey. Interna- tional Journal On Document Analysis

And Recognition (IJDAR). 12, 269 (2009)
[48] Lee, C. & Leedham, C. A new hybrid approach to handwritten address verification. Interna- tional Journal Of Computer Vision. 57, 107-120 (2004)
[49] Ul-Hasan, A., Ahmed, S., Rashid, F., Shafait, F. & Breuel, T. Offline printed Urdu Nastaleeq script recognition with bidirectional LSTM networks. 2013 12th International Conference On Document Analysis And Recognition. pp. 1061-1065 (2013)
[50] LeCun, Y. The MNIST database of handwritten digits. Http://yann. Lecun. Com/exdb/mnist/. (1998)
[51] Xiao, H., Rasul, K. & Vollgraf, R. FashionMNIST: a novel image dataset for benchmarking machine learning algorithms. ArXiv Preprint ArXiv:1708.07747. (2017)
[52] Pu, Y., Gan, Z., Henao, R., Yuan, X., Li, C., Stevens, A. & Carin, L. Variational autoencoder for deep learning of images, labels and captions. Advances In Neural Information Processing Systems. pp. 2352-2360 (2016)

Introduction Authors

Teerath Kumar

- May. 2019 ~ : He is pursuing the M.S. degree leading to Ph. D. degree with Department of Computer Science and Engineering, Kyung Hee University, South Korea.
- ORCID : https://orcid.org/0000-0001-8769-4989
- Interests : Semi-Supervised Learning, Data Generation and Augmentation

Jinbae Park

- May. 2020 ~ : He is pursuing the M.S. degree with Department of Computer Science and Engineering, Kyung Hee University, South Korea.
- May. 2013 ~ Feb, 2020 : He received the bachelor's degree with Department of Computer Science and Engineering and Electronic Engineering from Kyung Hee University, South Korea.
- ORCID : https://orcid.org/0000-0003-3469-199X
- Interests : Deep Learning, Model Compression,Quantization, Pruning, Distillation

Muhammad Salman Ali

- May. 2018 ~ : He is pursuining the M.S. degree leading to Ph. D. degree with Department of Computer Science and Engineering, Kyung Hee University, South Korea.
- May. 2013 ~ Feb. 2017 : He received the bachelor's degree in computer science fromt the National University of Sciences and Technology (NUST), Islamabad, Pakistan.
- ORCID : http://orcid.org/0000-0002-8548-3827
- Interests : Interpretation of Deep Neural Networks. Image Processing, Video Coding, Machine Learning

Introduction Authors

A F M Shahab Uddin

- Sep. 2021 ~ : He has been a postdoctorate fellow with Machine Learning and Visual Comuting laboratory (MLVC), Kyung Hee University, South Korea.
- Sep. 2017 ~ Aug, 2021 : He received the Ph. D. degree with Department of Computer Science and Engineering from Kyung Hee University, South Korea.
- 2016 : He received the M.S degree with Information and Communication Engineering from Islamic University, Bangladesh.
- 2015 : He received the bachelor’s degree with Information and Communication Engineering from Islamic University, Bangladesh.
- ORCID : https://orcid.org/0000-0003-1074-0515
- Interests : Image Quality Assessment, Perceptual Image Processing, Deep Learning and Inverse Problems in Image Processing

Sung-Ho Bae

- Sep. 2017 ~ : He has been an Assistant Professor with Department of Computer Science and Engineering, Kyung Hee University, Yongin, South Korea.
- Jul. 2016 ~ Aug. 2017 : He was a Postdoctoral Associate with Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology (MIT), MA, USA.
- Feb. 2011 ~ Aug. 2016 : He received M.S. and Ph. D. degrees with Department of Electrical and Electronic Engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea.
- May. 2004 ~ Feb. 2011 : He received the B.S. degree with he Department of Computer Science and Engineering/Electronics from Kyung Hee University, South Korea.
- ORCID : https://orcid.org/0000-0002-3389-1159
- Interests : Adversarial Attack, Model Compression, Semi-supervised Learning and Data Augmentation, Explainable DNN, Neural Architecture Search for Hardware Implementation, International Standardization