2018 24th International Conference on Pattern Recognition (ICPR)
Beijing, China, August 20-24, 2018

Data Augmentation via Latent Space Interpolation for Image Classification

Xiaofeng Liu[1,2,3†], Yang Zou[1†], Lingsheng Kong[2], Zhihui Diao[2], Junliang Yan[2], Jun Wang[2,3*]
Site Li[3], Ping Jia[2], Jane You[5]

[1] Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, USA
[2] Changchun institute of optical precision machinery and physics, Chinese academy of sciences, CAS, Changchun, China
[3] University of Chinese Academy of Sciences, Beijing, China
[4] Mellon Institute of Science, Carnegie Mellon University, Pittsburgh, USA
[5] Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China

{liuxiaofeng, yzou2, site}@cmu.edu; {konglingsheng, diaozh, jly, jiap}@ciomp.ac.cn; wangjun114@mails.ucas.ac.cn; csyjia@comp.polyu.edu.hk

Abstract— **Effective training of the deep neural networks requires much data to avoid underdetermined and poor generalization. Data Augmentation alleviates this by using existing data more effectively. However standard data augmentation produces only limited plausible alternative data by for example, flipping, distorting, adding noise to, cropping a patch from the original samples. In this paper, we introduce the adversarial autoencoder (AAE) to impose the feature representations with uniform distribution and apply the linear interpolation on latent space, which is potential to generate a much broader set of augmentations for image classification. As a possible "recognition via generation" framework, it has potentials for several other classification tasks. Our experiments on the ILSVRC 2012, CIFAR-10 datasets show that the latent space interpolation (LSI) improves the generalization and performance of state-of-the-art deep neural networks.**

Keywords-classification; data augmentation; vicinal risk minimization; inter-class sampling

I. INTRODUCTION

Over the last decade Deep Neural Networks have produced unprecedented performance on a number of tasks, given sufficient data. They have been demonstrated in variety of domains [1] spanning from image classification [2-6], machine translation [7], natural language processing [1], speech recognition [8], and synthesis [9], learning from human play [10]. These neural networks share the commonalities that they are trained to minimize the loss function which normally designed as their average error over the training data (mini-batch), a learning rule also known as the Empirical Risk Minimization (ERM) principle [11]. There is a classical result in learning theory [12] tells us that the convergence of ERM is guaranteed as long as the size of the learning machine (*e.g.*, the neural network) does not increase with the number of training data. Here, the size of a neural network is measured in terms of its number of parameters.

Data augmentation [13] is the choice to train on similar but different examples, which applying a small mutation in the original training data and synthetically creating new samples, is widely used to virtually increase the amount of training data (*e.g.* [14], [15]). It is also formalized by the Vicinal Risk Minimization (VRM) principle [16]. In VRM, human knowledge is required to describe a vicinity or neighborhood around each example in the training data. Then, additional virtual examples can be drawn from the vicinity distribution of the training examples to enlarge the support of the training distribution. For instance, when performing image classification, it is common to define the vicinity of one image as the set of its horizontal reflections, slight rotations, and mild scalings. Use of data augmentation is mostly the norm for winning image classification contests. While data augmentation consistently leads to improved generalization [13], the procedure is dataset-dependent, and thus requires the use of expert knowledge. Furthermore, data augmentation assumes that the examples in the vicinity share the same class, and does not model the vicinity relation across examples of different classes.

Recently works [17,18] extend the training distribution by incorporating the prior knowledge that pixel-level or feature-level linear interpolations of the samples should lead to linear interpolations of the associated target labels. However, for the high-dimensional data (*e.g.*, images), this assumption is invalid. Considering the inhomogeneous distribution of samples in a high-dimensional manifold, the interpolations of samples and targets may not align. A solution is to construct a latent feature space with uniform distribution and then using the assumption. The latent space can be sampled for feature-level data augmentation and to generate observable data values for pixel-level data augmentations.

† contribute equally *corresponding author

National Key R&D Plan 2016YFB0501003
The National Natural Science Foundation of China 61308099
Hong Kong Government General Research Fund GRF 152202/14E
PolyU Central Research Grant G-YBJW
Youth Innovation Promotion Association, CAS (2017264)
Innovative Foundation of CIOMP, CAS (Y586320150)
China Scholarship Council 201704910715

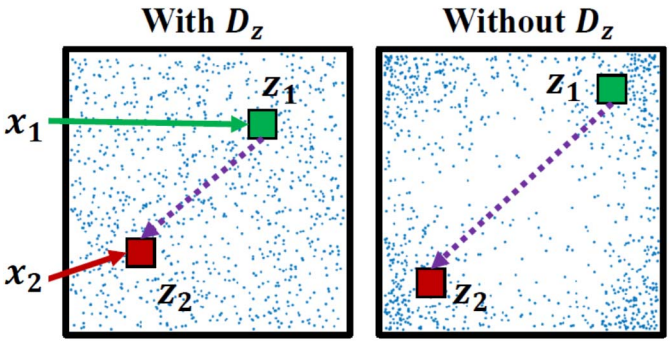

Figure 2. Effect of *Dis*, which forces *z* to a uniform distribution. For simplicity, *z* is illustrated in a 2-D space. Blue dots indicate *z*'s mapped from training faces through the encoder. With *Dis*, the distribution of *z* will approach uniform. Otherwise, *z* may present "holes". The rectangular points denote the corresponding *z* mapped from the input faces *x*1 and *x*2, and the dotted arrow indicates the traversing from *z*1 to *z*2. The intermediate points along the traversing are supposed to generate a series of plausible morphing samples from *x*1 to *x*2. Without *Dis*, the learned *z* presents a sparse distribution along the path of traversing, causing the generated sample to look unreal.

There are several possible model types to import the uniform distribution prior to latent space. Two popular generative models for image data are the Variational Autoencoder (VAE [19]) and the Generative Adversarial Network (GAN [20]). VAEs use the framework of probabilistic graphical models with an objective of maximizing a lower bound on the likelihood of the data. GANs instead formalize the training process as a competition between a generative network and a separate discriminative network. Though these two frameworks are very different, both construct high dimensional latent spaces that can be sampled to generate images resembling training set data. Moreover, these latent spaces are generally highly structured and can enable complex operations on the generated images by simple vector space arithmetic in the latent space [21].

Normally, the GAN schemes are not well-matched to supervised recognition tasks. The GAN-generated results are expected to align with the central part of the data distribution. For instance, the style transfer as an inner-class transfer method do generate several samples with the same class label but with different appearance or background environment. However, those inner-class samples are generated according to the distribution of training set to be similar as them. Moreover, those deformation are added by a very shallow network (*i.e.*, decoder), which may not informative for a much deeper recognition network. Thus, the generated samples usually cannot support the network to adjust the boundary. [21] employed the GAN for hard negative generation for metric learning [22,23]. In here, we propose to utilize the GAN for more general classification tasks as a kind of data augmentation or vicinal risk minimization method.

Generative models are often evaluated by examining samples from the latent space. Techniques frequently used are random sampling and linear interpolation. But often these can result in sampling the latent space from locations very far outside the manifold of probable locations. In high dimensional space, even with a uniform distribution most points lie on a thin shell in the unit cube. We utilize both the uniform distribution prior to avoid the "hole" of dead zone.

The three major contributions in this paper are: 1) We propose a novel framework to augment the dataset using the inter-class interpolation of latent feature representations. 2） uniform distribution are imposed as the prior to avoid "hole" effect of the inter-class interpolation via the adversarial autoencoder network. 3) we explore the linear interpolation on the ILSVRC 2012 and CIFAR-10 datasets.

II. Proposed Method

Our latent space interpolation (LSI) has followed two useful principles when sampling the latent space of a generative model. The first is to avoid sampling from locations that are highly unlikely given the prior of the model. This technique is very well established - including being used in the original VAE paper which adjusted sampling through the inverse CDF of the Gaussian to accommodate the Gaussian prior [24]. The second principle is to recognize that the dimensionality of the latent space is often artificially high and may contains dead zones that are not on the manifold learned during training. This has been demonstrated for VAE models [25] and implies that simply matching the model's prior will not always be sufficient to yield samples that appear to have been drawn from the training set.

LSI enjoys several desirable aspects of previous data augmentation and regularization schemes without suffering from their drawbacks. Like the method of [26], it does not require significant domain knowledge. [26] show that interpolation and extrapolation in feature space can improve generalization. However, their proposal only operates at the feature level and does not account for changes in the corresponding labels. Recent approaches have also proposed to regularize the output distribution of a neural network by label smoothing [27], or penalizing high-confidence softmax distributions [28]. These methods bear similarities with mixup in the sense that supervision depends on multiple smooth labels, rather than on single hard labels as in traditional ERM. However, the label smoothing in these works is applied or regularized independently from the associated feature values. Like label smoothing, the supervision of every example is not overly dominated by the ground-truth label. Unlike

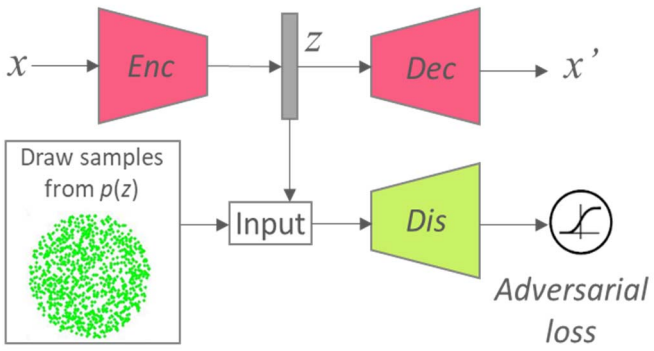

Figure 1. Architecture of an adversarial autoencoder. The top row is a standard autoencoder that reconstructs an image x from a latent code z. The bottom row diagrams a second network trained to discriminatively predict whether a sample arises from the hidden code of the autoencoder or from a sampled distribution specified (e.g., uniform distribution) by the user.

both of these approaches, the LSI transformation establishes a linear relationship between data augmentation and the supervision signal. We believe that this leads to a strong regularizer that improves generalization as demonstrated by our experiments. The linearity constraint, through its effect on the derivatives of the function approximated, also relates mixup to other methods such as Sobolev training of neural networks [29] or WGAN-GP [30].

When other types of data augmentation are employed in addition to our technique, we can apply them for each image before mixing them into the final image for training. The data augmentation incurs additional time to prepare the input image, but this can be done on the CPU while the GPU is executing the training through back propagation. Therefore, the additional execution time of the CPU does not visibly increase the total training time per image. Our current implementation uses a single data loader to obtain one minibatch, and then latent space interpolation is applied to the same minibatch after random shuffling. This strategy works equally well, while reducing I/O requirements.

A. Uniform distribution representations

Adversarial autoencoder (AAE) [31] can be treated as the combination of generative adversarial networks (GANs) [20] and variational autoencoder (VAE) [19], which maintains the autoencoder structure like the VAE but replaces the *KL*-divergence loss with a discriminative network, denoted by *Dis*, like in GAN. We illustrate the basic framework of AAE in Figure 1. Instead of generating images from random noise as in GAN, AAE utilizes the encoder part to learn the latent variables approximated on certain prior, making the style of generated images controllable. In addition, AAE better captures the data manifold compared to VAE. We denote the input as x, output as x', and the distribution of the training data as $p_{data}(x)$, then the distribution of z is q($z|x$). Assuming $p(z)$ is a prior distribution, and $z^* \sim p(z)$ denotes the random sampling process from $p(z)$. The min-max objective function can be used to train the *Enc* and *Dis*:

$$\min_{Enc} \max_{Dis} \mathbb{E}_{z^* \sim p(z)}[logDis(z^*)] + \mathbb{E}_{x \sim p_{data}(x)}[log(1 - Dis_z(z^*))] \quad (1)$$

As for the *Dec*, the L2 loss is used as the reconstruction loss as following:

$$L_{recon} = \|x - x'\|_2^2 \quad (2)$$

The *Dis* imposes a prior distribution (*i.e.*, uniform distribution) on z. Specifically, *Dis* aims to discriminate the z generated by encoder *Enc*. Simultaneously, *Enc* will be trained to generate z that could fool *Dis*. Such adversarial process forces the distribution of the generated z to gradually approach the prior. We use uniform distribution as the prior, forcing z to evenly populate the latent space with no apparent "holes". As shown in Figure. 2, the generated z's (depicted by blue dots in a 2-D space) present uniform distribution under the regularization of *Dis*, while the distribution of z exhibits a "hole" without the application of *Dis*. Exhibition of the "hole" indicates that the samples generated by interpolating between arbitrary z's may not lie on the real image manifold – generating unrealistic appearances. Note that our objective is not to generate photorealistic images as in GAN [20], but to offer more informative and discriminative training sample, the adversarial loss in pixel-level is dropped in here for faster training and processing.

B. Linear interpolation

Frequently linear interpolation is used, which is easily understood and implemented. Interpolation is used to traverse between two known locations in latent space. Research on generative models often uses interpolation as a way of demonstrating that a generative model has not simply memorized the training examples [32]. In creative applications interpolations can be used to provide smooth transitions between two decoded images. Sampling from the latent vicinal distribution produces virtual representation-target pairs:

$$\hat{z} = \lambda z_i + (1 - \lambda) z_j \quad (3)$$

$$\hat{y} = \lambda y_i + (1 - \lambda) y_j \quad (4)$$

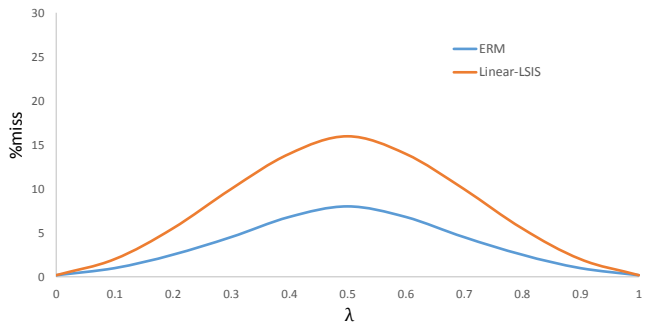

Figure 5. Prediction errors in-between training data. Evaluated at $\hat{z} = \lambda z_i + (1-\lambda) z_j$, a prediction is counted as a "miss" if it does not belong to $\{y_i, y_j\}$. The model trained with mixup has fewer misses.

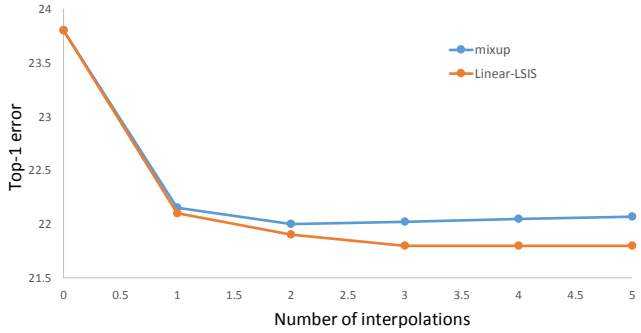

Figure 6. The relationship of the Top-1 eror rate (%) with the number of uniform-interpolation used as data augmentation in ILSVRC 2012 dataset with ResNet-50 in 200 epochs.

where (z_i, y_j) and (z_j, y_i) are two representation-target pairs drawn at random from the training data, and $\lambda \in [0,1]$. The implementation of linear interpolation is straightforward, and introduces a minimal computation. The difference with mixup [17] in here is that the linear interpolation manipulation is conducted in latent space which align the uniform distribution instead of the pixel-level. The one-hot vectors are used to label the original samples. The loss function for a generated sample are calculated as the weighted sum (via λ) of two cross-entropy losses corresponding to both of its original samples. When the λ=0.5, we can simply use two-hot vector to label the generated sample belongs to two of its original samples for faster processing.

III. Experiments

In this section, we investigate the effects of latent space inter-class sampling (LSI) data augmentation using various image classification tasks: ILSVRC 2012 and CIFAR-10 datasets. During the training, we also use the conventional data augmentations by extracting a patch from a random position of the input image and using random horizontal flipping as the baseline regardless of whether or not it was with our latent space interpolation. In this paper, we do not ensemble predictions in the results. For the validation set, we extracted the patch from the center position and fed it into the classifier without horizontal flipping. We implemented our algorithm using TensorFlow [42] as the framework.

TABLE I. Validation Errors on the Development Set of ILSVRC-2012 Dataset with Different Data Augmentation Methods.

Model	***Method***	***Epochs***	***Top-1 Error***	***Top-5 Error***
ResNet-50	ERM[35]	90	23.5	-
	mixup[17]	90	23.3	6.6
	LSI	90	23.3	6.6
ResNet-101	ERM[35]	100	22.1	-
	mixup[17]	90	21.5	5.6
	LSI	90	21.4	5.4
ResNet-50	ERM	200	23.6	7.0
	mixup[17]	200	22.1	6.1
	LSI	200	22.1	6.0
ResNet-101	ERM	200	22.0	6.1
	mixup[17]	200	20.8	5.4
	LSI	200	20.7	5.2

A. ImageNet Classification

We evaluate the proposed LSI on the ILSVRC-2012 classification dataset [34], which contains 1.3 million training images and 50,000 validation images, from a total of 1,000 classes. For the training stage, we also follow standard data augmentation practices: scale and aspect ratio distortions, random crops, and horizontal flips [35]. During the testing, only the 224 × 224 central crop of each image is tested. We use LSI, mixup and ERM (with only standard data augmentations mentioned above) to train several state-of-the-art ImageNet-2012 classification models, and report both top-1 and top-5 error rates in Table 1.

For all the experiments in this section, we use data-parallel distributed training with a minibatch size of 1,024. We use the learning rate schedule described in [35]. Specifically, the learning rate is increased linearly from 0.1 to 0.4 during the first 5 epochs, and it is then divided by 10 after 30, 60 and 80 epochs when training for 90 epochs; or after 60, 120 and 180 epochs when training for 200 epochs. We also find that models with higher capacities and/or longer training runs are the ones to benefit the most from LSI.

B. CIFAR-10 Classification

For the CIFAR-10 dataset, we use three state-of-the-art networks: the Pre-Act ResNet-18 [36] as implemented in [37], WideResNet-28-10 [38] as implemented in [39] and DenseNet-BC-190 network [40] as implemented in [43]. We only interpolate

one generative sample for data augmentation as the same as mixup [17] for fare comparation. All models are trained on a single Nvidia TitanX GPU for 200 epochs on the training set with 128 examples per minibatch, and evaluated on the test set. Learning rates start at 0.1 and are divided by 10 after 100 and 150 epochs for all models except WideResNet. For WideResNet, we follow [38] and divide the learning rate by 10 after 60, 120 and 180 epochs. Weight decay is set to 10^{-4}. We do not use dropout in these experiments.

TABLE II. TEST ERROR OF THE CIFAR EXPREMENTS. ONLY ONE INTERPOLATION SAMPLE IS USED FOR MIXUP AND LSI.

Datasets	***Model***	***ERM***	***Mixup*** [17]	***LSI***
CIFAR-10	Pre-Act ResNet-18	5.6	3.9	3.8
	WideRes Net-28-10	3.8	2.7	2.7
	DenseNet-BC-190	3.7	2.7	2.6

We summarize our results in Table II. In CIFAR-10 classification problem, the models trained using LSI significantly outperform their analogues trained with ERM and outperform the pixel-level interpolation.

IV. CONCLUSION

This paper presents our new data augmentation technique named LSI. As a form of vicinal risk minimization, which trains on virtual examples constructed via interpolations of the features in a latent space with uniform distribution. Several methods are employed to avoid the dead zone in manifold of feature representations. We conduct our classification experiments on three popular real datasets, the ILSVRC 2012 and CIFAR-10 datasets and the recognition accuracy are improved on all of the test dataset with the state-of-the-art deep neural networks. Our technique is valuable for tasks with a limited number of samples, such as medical image classification tasks. In the future, we may extend this framework for more classification tasks and other machine learning tasks especially the training of Generative Adversarial Networks.

ACKNOWLEDGMENT

The careful works of the anonymous reviewers are greatly appreciated. We wish to thank Prof. B.V.K Kumar for his constructive suggestions and foresight supports.

REFERENCES

[1] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, and Gang Wang. Recent Advances in Convolutional Neural Networks. dec 2015. URL http://arxiv.org/abs/1512.07108..

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pp. 1097–1105, 2012.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. feb 2015. URL http://arxiv.org/abs/1502.01852.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. 2015. http://arxiv.org/abs/1512.03385.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In European Conference on Computer Vision, pp. 630–645. Springer, 2016.

[6] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. arXiv preprint arXiv:1608.06993, 2016.

[7] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144, 2016..

[8] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Processing Magazine, 29(6):82–97, 2012a.

[9] Yuxuan Wang, R. J. Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc V. Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous. Tacotron: A fully end-to-end text-to-speech synthesis model. CoRR, abs/1703.10135, 2017. URL http://arxiv.org/abs/1703.10135.

[10] C. Clark and A.J. Storkey. Training deep convolutional networks to play go. In Proceedings of 32nd International Conference on Machine Learning (ICML2015), 2015. (arxiv 2014)..

[11] V. N. Vapnik. Statistical learning theory. J. Wiley, 1998.

[12] V. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. Theory of Probability and its Applications, 1971.

[13] P. Simard, Y. LeCun, J. Denker, and B. Victorri. Transformation invariance in pattern recognitiontangent distance and tangent propagation. Neural networks: tricks of the trade, 1998.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. NIPS, 2012.

[15] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus. Intriguing properties of neural networks. ICLR, 2014.

[16] O. Chapelle, J. Weston, L. Bottou, and V. Vapnik. Vicinal risk minimization. NIPS, 2000.

[17] Zhang, H., Cisse, M., Dauphin, Y.N. and Lopez-Paz, D., 2017. mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412.

[18] Inoue, H., 2018. Data Augmentation by Pairing Samples for Images Classification. arXiv preprint arXiv:1801.02929.

[19] Doersch, C., 2016. Tutorial on variational autoencoders. arXiv preprint arXiv:1606.05908.

[20] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., 2014. Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672-2680).

[21] X. Liu, B.V.K Kumar, Y. Ge, C. Yang, J. You and P. Jia. 2018. Normalized face image generation with perceptron generative adversarial networks. In IEEE ISBA 2018.

[22] X. Liu, Y. Ge, C. Yang and P. Jia. Adaptive metric learning with deep neural networks for video-based facial expression recognition. Journal of Electronic Imaging, 27(1), p.013022.

[23] X. Liu, B.V.K Kumar, J. You, P. Jia. Adaptive Deep Metric Learning for Identity-Aware Facial Expression Recognition. In CVPRW 2017.

[24] Kingma, Diederik P. and Welling, Max. Auto-encoding variational Bayes. In Proceedings of the International Conference on Learning Representations, 2014.

[25] Makhzani, Alireza, Shlens, Jonathon, Jaitly, Navdeep, Goodfellow, Ian, Brendan, Frey. Adversarial Autoencoders. arxiv. abs/1511.05644. 2016.

[26] T. DeVries and G. W. Taylor. Dataset augmentation in feature space. ICLR Workshops, 2017.

[27] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[28] G. Pereyra, G. Tucker, J. Chorowski, Ł. Kaiser, and G. Hinton. Regularizing neural networks by penalizing confident output distributions. ICLR Workshops, 2017.

[29] W. M. Czarnecki, S. Osindero, M. Jaderberg, G. ´ Swirszcz, and R. Pascanu. Sobolev training for neural networks. NIPS, 2017.

[30] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved training of wasserstein gans. NIPS, 2017.

[31] Makhzani, Alireza, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. "Adversarial autoencoders." arXiv preprint arXiv:1511.05644 (2015).

[32] Radford , Alec, Metz , Luke, Chintala, Soumith. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. Advances in Neural Information Processing Systems, 2015.

[33] Shoemake, Ken. Animating rotation with quaternion curves. In ACM Siggraph, 19(3):245–254, 1985.

[34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet large scale visual recognition challenge. IJCV, 2015.

[35] P. Goyal, P. Doll´ar, R. Girshick, P. Noordhuis, L.Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He. Accurate, large minibatch SGD: Training imageNet in 1 hour. arXiv, 2017.

[36] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. ECCV, 2016.

[37] K. Liu, 2017. URL https://github.com/kuangliu/pytorch-cifar.

[38] S. Zagoruyko and N. Komodakis. Wide residual networks. BMVC, 2016.

[39] S. Zagoruyko and N. Komodakis, 2016. URL https://github.com/szagoruyko/wide-residual-networks.

[40] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. CVPR, 2017.

[41] X. Liu, L. Kong, Z. Diao and P. Jia. “Line-scan system for continuous hand authentication”. Optical Engineering, 56(3),. 033106-033106, 2017.

[42] X. Liu, Z. Li, L. Kong, Z. Diao, J. Yan, Y. Zou, C. Yang, P. Jia, J. You. “A joint optimization framework of low-dimensional projection and collaborative representation for discriminative classification”, In ICPR 2018.

[43] A. Veit, 2017. URL https://github.com/andreasveit.