Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 3, 2021, Pages. 8753 - 8762
Received 16 February 2021; Accepted 08 March 2021.

Wasserstein GANs for Generation of Variated Image Dataset Synthesis

[1]Kalpana Devi Bai. Mudavathu , [2]Dr. M V P Chandra Sekhara Rao

[1]Research Scholar, Department of CSE, Acharya Nagarjuna University, Guntur, AP, India.

[2]Professor, Department of CSE, RVR & JC College of Engineering, Guntur, AP, India.

[1]dkalpananaik@gmail.com,2manukondach@gmail.com

Abstract

Deep learning networks required a training lot of data to get to better accuracy. Given the limited amount of data for many problems, we understand the requirement for creating the image data with the existing sample space. For many years the different technique was used to develop data fixtures to improve modelling and training efficiently with the advent of GAN we were now able to get close to real data. However, the standard GANs require a lot of effort in training and not cost-efficient. A more practical way of training GANs is Wasserstein GAN. They can be used for efficient generating for data taking the training sample space. Better representation GANs with WGANS solved the problem of learning a probability density. In this paper, we now use WGANs for classification training data given the sample data. We intend to deal with the following objectives. (a)To consider the sample space for the training data to mock with WGAN. (b)To build WGAN in combination with the network for classification and evaluating the models' performance. (c)To compare WGAN and standard GAN for knowing the increased accuracy of the classifier.

Keywords: *WGANs, GANs Deep Learning, Machine Learning, Image Synthesis*

1. Introduction and Literature Review

Generative Adversarial Networks had produced state of the art results in several generative tasks for replicating data such as image generation, human language understanding, music composition and many more. These were first introduced by Ian Goodfellow back in 2014, in his research titled "Generative Adversarial Nets" [1]. Later, several architectures were introduced and experimented with GANs to make generative tasks more productive. The core logic behind GANs is inspired by game theory; hence, these are sometimes referred to as zero-game network architecture.

Unlike prevalent Deep Learning models like Convolutional Neural Networks (CNNs) as well as Recurrent Neural Networks (RNNs), GANs are built using two different architectures which are known as Generator Network (Z) and Discriminator Network (X). The objective of a network generator is to generate novel data, while the discriminator network classifies if the data is since the training data (real data) or the fake ones. Zero-sum logic is utilized for training these two adversarial networks, until the discriminator model is deceived, meaning

Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 3, 2021, Pages. 8753 - 8762
Received 16 February 2021; Accepted 08 March 2021.

the generator model is generating almost-real examples. In this way, the networks are trained alternatively to produce real-world data. In Fig.1, we have the GAN architecture illustrated in the original research by Ian Goodfellow.

As presented in Fig.1, the training data is sent into discriminator, and it can be considered as a simple CNN based classifier that trains on real data and classifies if a given image is real or a fake. On the other hand, random noise is sent into the generator to yield a fake image that's close to the real image. In this way, these networks are well trained to compete with all other on the other hand. Mathematically speaking, generator Z is enhanced to replicate the tangible data dissemination by creating indications that are problematic for the discriminator X to recognize from actual signals. In the interim, X is augmented to discriminate real signals and counterfeit signals engendered by Z.

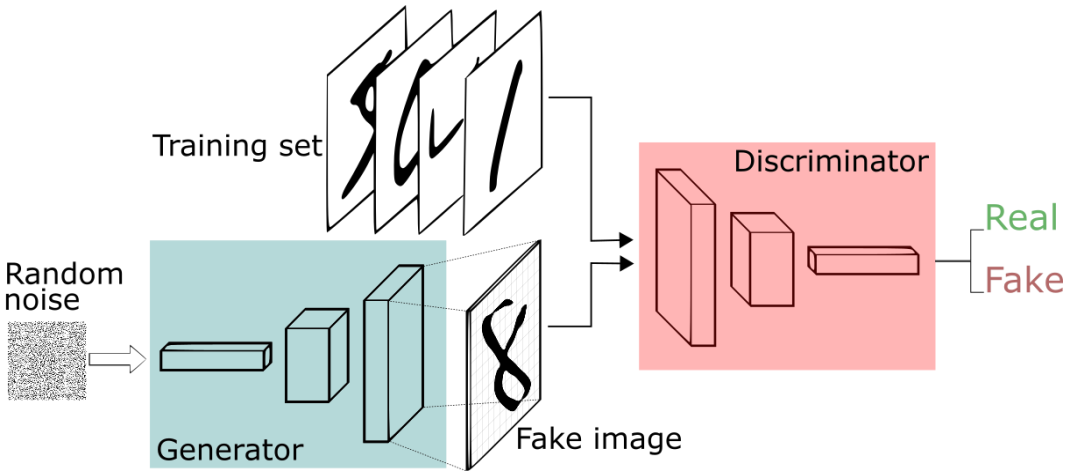

Figure 1. GAN Generator, Discriminator Architecture [1]

In broad-spectrum, the training procedure is a min-max a two-player game with an objective unbiased function given by the following equation. The backpropagation or the primary training loss function for GANs is derived using two simple equations. Considering *D(x)* as the discriminator output where x is the chance of getting a real output. The loss function's primary goal is to exploit the opportunity to distinguish authentic images as accurate and produced images as fake. I.e. the maximum possibility of the pragmatic data. Using the cross-entropy loss function *p log(q)*, the objective function of GAN become:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

In the above equation, **p** is the actual signal from the actual distribution. The variables pr and y are the noise vectors generated using the Gaussian otherwise uniform distribution. If X is trained to be optimum beforehand respectively Z parameter is rationalised, then the minimization value function is equivalent to the disseminations on p. But whenever the discriminator is inundated, this frequently consequences in a gradient vanishing delinquent. In repetition, Z is trained to maximize **E** used by equation 2, which can evade this exertion to selected

Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 3, 2021, Pages. 8753 - 8762
Received 16 February 2021; Accepted 08 March 2021.

magnitude [2]. Nevertheless, in some cases, equal this modern loss function might not give good results when a noble discriminator appears.

$$-\nabla_{\theta_g} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \rightarrow 0 \quad \textit{change to} \quad \nabla_{\theta_g} \log\left(D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)$$

2. Wasserstein Generative Adversarial Network

The Wasserstein Generative Adversarial Network, WGAN in short, is built on top of Generative Adversarial Networks (GANs). Few main advantages of WGANs include better training and fewer loss metrics correlating with the quality of generating images. However, to achieve this, several mathematical research was carried out to achieve it. WGANs only require only an insufficient adjustment to establish over a customary deep convolutional generative adversarial network or DCGAN. The WGANs are primarily used to engender synthetic samples. For the training process, when compared with original gans and taking the mathematical inspiration of min-max game and optimization, the goal of the loss function is to influence the Nash equilibrium [4], whichever poses the vanishing gradient problem [5]. Next, WGAN utilizes the Wasserstein distance as an alternative of the Jensen–Shannon (JS) discrepancy. This help in evaluating the distribution between the tangible samples and engendered samples. These hyperparameters make the training process faster, stable and efficient.

WGANs owed to the irregular JS mutation have certain additional popular detachments and alterations, and the GAN normally has an unsteady gradient once training the initiator Z. The Wasserstein distance [8] is capable to extent the dissimilarity among 2 frequencies. The Wasserstein distance WD(pr, pg) [11, 12] is distinct as the less cost to congregate the model distribution(pg) to the real distribution(pr). The Wasserstein loss can condense the gradient vanishing problem.

$$W(pr,pg)=\inf\gamma\sim\Pi(pr,pg)\mathbb{E}(x,y)\sim\gamma[\|x-y\|]$$

In the above equation M (pr, pg) represents the customary of altogether joint distributions γ (p,q) whose marginals are correspondingly pr then pg. Spontaneously, γ (p,q) denotes how abundant ‘‘mass’’ consumes to be elated from p to q so has to transmute pr into pg . The Wasserstein distance WD(pr, pg) then is the ‘‘cost’’ of the optimum transportation plan. The WGAN value function is erected by Kantorovich Rubinstein duality [9] to the below equation, which optimizes the loss.

$$\min Z \max X\in S \; Ex\sim pr\,[Z(p)] - Ey\sim pg\,[X(q)] \text{ [10]}$$

The outcome for WGAN value function is a critical function whose gradient interrelated to the input performs improved than GAN, making it stress-free to optimise the generator. Where S is the customary of 1-Lipschitz functions, the outcome of the WGAN value function is a critical function whose gradient interrelated to the input behaves restored than GAN, making it stress-free to optimise the generator. Furthermore, WGAN has the idyllic property of having its value function linked to sample mass. WGAN staples the weights of the criticiser keen on a compact space [t, t] to impose the Lipschitz limit on critics [13]. For any n, the set of functions that mollify this

Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 3, 2021, Pages. 8753 - 8762
Received 16 February 2021; Accepted 08 March 2021.

restriction is a subcategory of the n-Lipschitz functions that depend on t and the essential architecture.

3. Loss Function for Wasserstein

The DCGAN considers the discriminator as a simple binary classification model. This helps the network to forecast the prospect if a prearranged image is tangible or not. Unlike normal GANs, the hyperparameters for WGAN are quite different. For training purpose, the binary cross-entropy function is utilized for both generator and discriminator networks [14]. In this research, we'll be utilizing the Wasserstein loss function, and it reassures the discriminator to envisage a score of how tangible or fake a prearranged input appearances, and at the same time, it transmutes the part of the discriminator as a critic from a classifier to predict the score of real or fakeness of a given input image. If the score's large, it means the generator has to still train. It's minimum then the generator is able to produce new images. On top of this, we also provided a custom function that estimates the average score for a real or fake image. With the respective SGD (Stochastic Gradient Descent) optimization algorithm, we can proliferate the output labels using the mean score. For example, we consider -1 to be a real image and 1 for a fake image that has no effect. Using this, we can predict if the given input image is real or fake by observing the boundaries of the loss function. The following is practically implemented on PyTorch, a neural network framework based on Python.

4. Effective techniques to achieve faster convergence of GAN

In this section, we'll be discussing five effective practises to accomplish quicker conjunction of GAN training.

Feature Mapping: This technique suggests optimizing the discriminator to examine whether the generator's output competitions the tangible samples' expected statistics. In such a consequence, the innovative loss function is demarcated as follows.

$$\|\mathbb{E}x\sim prf(x)-\mathbb{E}z\sim pz(z)f(G(z))\|22 \text{ [11]}$$

Where f(x) is an statistical feature, such as mean or median.

Minibatch Discrimination: The difference between the training data points in one batch and independent processing batch is calculated by the discriminator itself. For each batch, the closeness of samples is approximated using the following mathematical expression: c(xi,xj). This summarises one data point by summing up the clones with the other models on the same batch, $o(xi)=\sum jc(xi,xj)$. Lastly, o(xi) additional to the response model.

Historic Averaging: For both models, add $\|\Theta-1t\sum ti=1\Theta i\|2$ into the loss function, where is the model parameter and I is the configuration of the

Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 3, 2021, Pages. 8753 - 8762
Received 16 February 2021; Accepted 08 March 2021.

parameter at the previous training time i. If the training speed changes too drastically in time, this addition piece penalises it [15].

Smoothing one-sided labels: Rather than providing labels 1 and 0, the discriminator is fed with soft values. For example, 0.9 and 0.1 this help to condense network defencelessness.

Virtual Batch Normalization (VBN): Every data samples are standardised based on immovable batch size. These are often referred to as the "reference batch", as the data is within the minibatch. This is defined in the initial process of training a WGAN.

4. Understanding and Controlling Gradients and Weights on Generative Adversarial Networks

4.1. Experiments

We initially have trained the Standard GAN with the datasets MNIST and FashionMNIST. We have then compared the loss, time and other. Eventually then we took a route towards tweaking parameters. The same datasets are training with Wasserstein GANs and the results are compared.

4.2. Training Standard GAN with an MNIST and FMNIST Datasets

The time taken for the WGAN to train itself is almost 1hour 18 minutes. Standard GANs took a lot of time to converge on both generators and discriminator for both the data sets. Datasets that are created out of standard image generation are mostly noisy until hundred epoch and started evolving proving that to be harder to train.

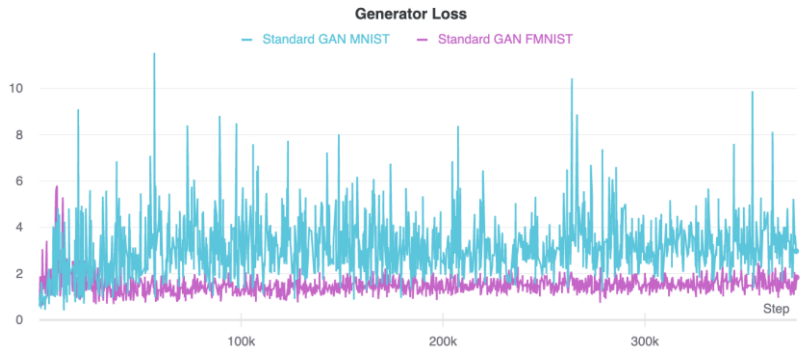

Figure 2. Generator Loss on MNIST and FMNIST

Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 3, 2021, Pages. 8753 - 8762
Received 16 February 2021; Accepted 08 March 2021.

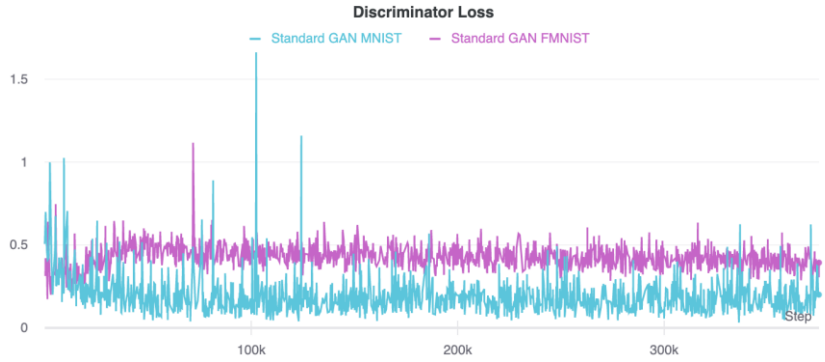

Figure 3. Discriminator Loss on MNIST and FMNIST

5. Training Wasserstein GAN with an MNIST and FMNIST Datasets

The time taken for WGAN when compared to standard GAN to generate images is much lesser. The standard GAN approximately takes 52 minutes to finish 200 epoch of training about 21 minutes less than the time that is taken by the Standard GAN on both the datasets. Within a few from the first epoch the took less time to converge on both discriminator and generator the WGAN have generated much clearer images than the standard GAN right from the first epoch.

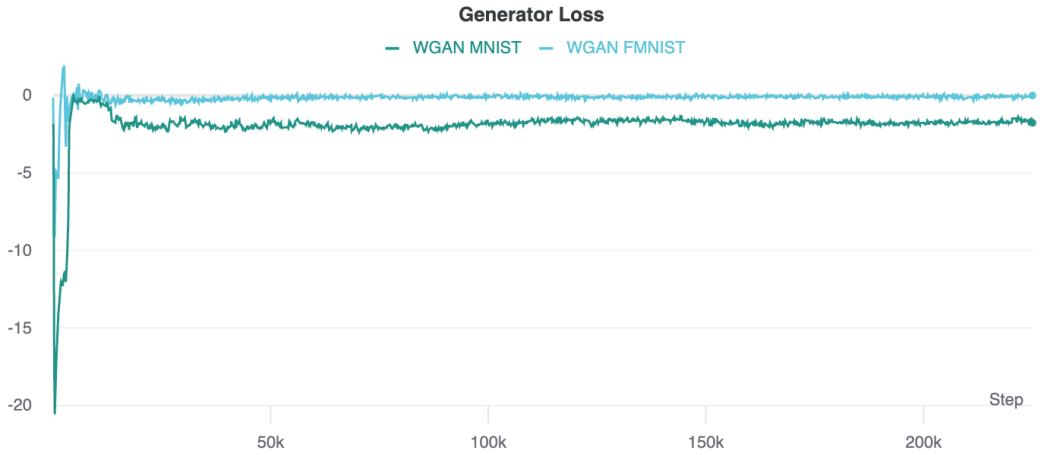

Figure 4. Generator Loss on MNIST and FMNIST for WGAN

Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 3, 2021, Pages. 8753 - 8762
Received 16 February 2021; Accepted 08 March 2021.

Figure 5. Discriminator Loss on MNIST and FMNIST for WGAN

6. Changes in Architectures and Training

In order to understand what's the reason behind faster convergence and better image generation it's all about the weight clipping in training that WGAN introduces. The image data that is synthesized are more clear than that of have been made on standard GAN. Having control on the gradients as well as weights have clearly shown results thereby, increasing the accuracy on the classifier. Below is the training proposed training snippet,

```
for epoch in epochs:
    for image in real_images
        valid = Tensor(0, image_size)
        fake = Tensor(1, image_size)
        real_images = Tensor(real_images)
        z = latent_space_variables
        generated_images = generator(z)
        generator_loss =
adversarial_loss(discriminator(generated_images),
valid)

        backprop(generator_loss)
        stepoptimizer
        real_loss = adversarial_loss
discriminator(real_images), valid)
        fake_loss =
adversarial_loss(discriminator(generated_images),
fake)

        discriminator_loss = real_loss + fake_loss / 2

        backprop(discriminator_loss)
        generated_images
 endfor
```

```
for epoch in epochs
    for image in images
        real_images = Tensor(image)
        z = latent_space_variables
        fake_images = generator(z)
        real_images_mean =
mean(discriminator(real_images))
        fake_images_mean =
mean(discriminator(fake_images))
        discriminator_loss =
real_images_mean +
fake_images_mean
        backprop(loss)
        stepoptimizer
        for probabilities in
discriminatoroptions
            clamp(-paramaters,
+paramaters)
        endfor
        if i % option_critic is 0
            optimizer
            generated_images =
generate_images(z)
            generator_loss = -
mean(discriminator(generated_images))
            generator_backward
            optimizer_step
```

Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 3, 2021, Pages. 8753 - 8762
Received 16 February 2021; Accepted 08 March 2021.

endfor	endif
	endfor
	endfor

6. Results

The below are the results of the data that is trained on both standard GAN and Warrenstein GAN as shown below the data that is being generated by the WGAN and has been much better efficient.

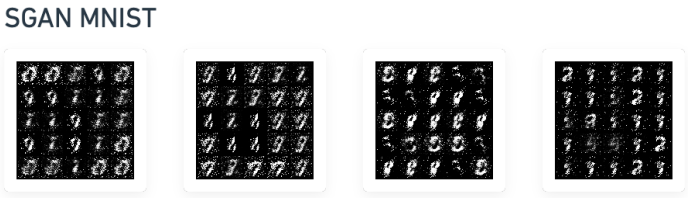

Figure 6. Results from SGAN on MNIST

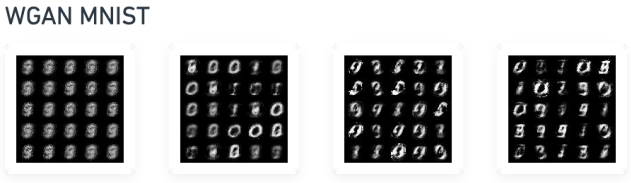

Figure 7. Results from WGAN on MNIST

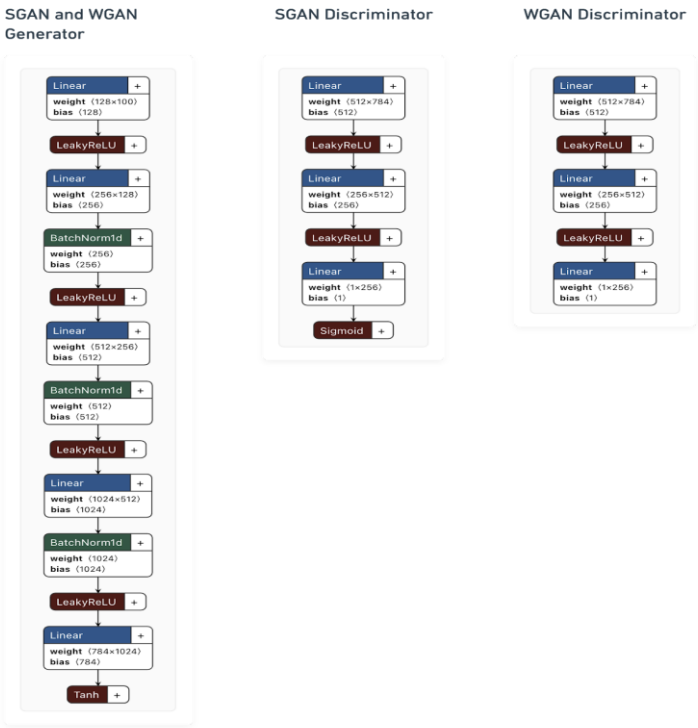

Figure 8. Architectures from the Training

Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 3, 2021, Pages. 8753 - 8762
Received 16 February 2021; Accepted 08 March 2021.

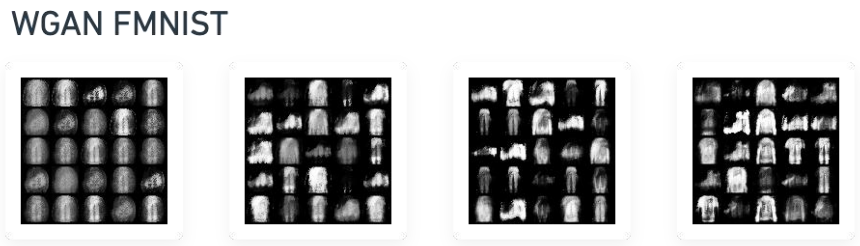

Figure 9. Results from WGAN on FMNIST

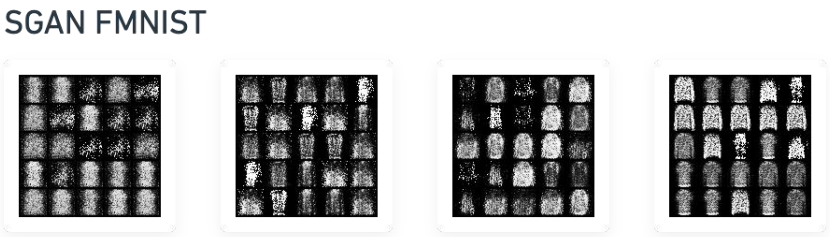

Figure 10. Results from SGAN on FMNIST

7. Results

This paper experimented on a more practical way of training GAN by analysing the gradients and making the models' training converge faster with Wasserstein GAN. For improving GAN training, one of the most vital for effective generation is data samples that have been observing gradients and resulting in weights during these models' training. Since GAN is the game between two neural networks, controlling both generator and discriminator gradients is likely to create better results. As we advance in the future of the work creating better datasets Loss function on the generators and discriminator may also play a crucial role.

7. References

[1] Goodfellow I, Bengio Y, Courville A. Deep learning. Cambridge: The MIT Press; 2016.

[2] Creswell A, White T, Dumoulin V, Arulkumaran K, Sengupta B, Bharath AA. Generative adversarial networks: an overview. IEEE Signal Process Mag 2018;35(1):53–65.

[3] Arjovsky M, Chintala S, Bottou L. Wasserstein GAN. 2017:arXiv:1701.07875.

[4] Creswell A, White T, Dumoulin V, Arulkumaran K, Sengupta B, Bharath AA. Generative adversarial networks: an overview. IEEE Signal Process Mag 2018;35(1):53–65.

[5] Ratliff LJ, Burden SA, Sastry SS. Characterization and computation of local Nash equilibria in continuous games. In: Proceedings of the 51st Annual Allerton Conference on Communication, Control, and Computing; 2013 Oct 2–4; Monticello, IL, USA. New York: IEEE; 2006. p. 917–24.

Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 3, 2021, Pages. 8753 - 8762
Received 16 February 2021; Accepted 08 March 2021.

[6] Danihelka I, Lakshminarayanan B, Uria B, Wierstra D, Dayan P. Comparison of maximum likelihood and GAN-based training of real NVPs. 2017: arXiv:1705.05263.

[7] Yang Q, Yan P, Zhang Y, Yu H, Shi Y, Mou X, et al. Low dose CT Image denoising using a generative adversarial network with Wasserstein distance and perceptual loss. IEEE Trans Med Imaging 2017;37(6):1348–57

[8] F. Jia, Y. Lei, J. Lin, X. Zhou, and N. Lu, ‘‘Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data,’’ Mech. Syst. Signal Process., vol. 72, pp. 303–315, May 2016.

[9] C. Villani, ‘‘Optimal transport. Old and new,’’ J. Bus., vol. 4, p. 44, 2009.

[10] Y. Lei, Intelligent Fault Diagnosis and Remaining Useful Life Prediction of Rotating Machinery. London, U.K.: Heinemann, 2016.

[11] F. Jia, Y. Lei, S. Xing, and N. Lu, ‘‘Deep normalized convolutional neural network for imbalanced fault classification of machinery and its understanding via visualization,’’ Mech. Syst. Signal Process., vol. 110, pp. 349–367, Sep. 2018.

[12] Sahiner B, Pezeshk A, Hadjiiski LM, Wang X, Drukker K, Cha KH, Summers RM, Giger ML. Deep learning in medical imaging and radiation therapy. Med phys. 2018;46:e1–36.

[13] Kaji S. Image translation by CNNs trained on unpaired data. 2019. https://github.com/shizuo-kaji/UnpairedImageTranslation. Accessed 18 June 2019.

[14] Kaji S. Image translation for paired image datasets (automap + pix2pix). 2019. https://github.com/shizuo-kaji/PairedImageTranslation. Accessed 18 June 2019.

[15] C. Villani, ‘‘Optimal transport. Old and new,’’ J. Bus., vol. 4, p. 44, 2009.

Authors

Kalpana is a research scholar in the Department of Computer Science and Engineering at Acharya Nagarjuna University, Guntur, and Presently working as Assistant Professor at PSCMRCET, VIJAYAWADA affiliated to JNTUK, Kakinada. She has over years of 10 experience in teaching and worked in the fields of Imaging Processing, Data Mining, Artificial Intelligence.

M.V.P.Chandra Sekhara Rao , is a Professor in the Department of Computer Science and Engineering in R.V.R. & J.C. College of Engineering, Chowdavaram, Guntur. He has over the years of 24 experience in teaching. He completed his B.E and M.Tech in Computer Science & Engineering. He got Ph.D. from JNTU, Hyderabad and his research area is Data Mining. He has published 8 papers in international journals and presented a paper at an international conference.