Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 12, No. 2, 2024

Data enhancement method based on cyclic generation adversarial network

Shiling Li, Tianhong Wang [a]

Faculty of Mathematics & Science, Integral University, Sicuan 621010, China
[a] wangtianlan09@gmail.com

Abstract: This study introduces a novel data augmentation technique employing Cycle Generative Adversarial Networks (CycleGAN) to mitigate the challenges posed by the paucity of image datasets in deep learning domains. Through the adept training of a CycleGAN model, this method substantially enriches image datasets, thereby enhancing the efficiency of deep learning models in target detection tasks. Distinct from conventional approaches, our strategy incorporates advanced activation functions, relative discriminators, and residual connections, which collectively foster greater image diversity and mitigate mode collapse, all while maintaining a low computational overhead. Evaluations conducted on diverse datasets, including MNIST, Synthetic Aperture Radar (SAR), and medical blood cell images, demonstrate the method's superior augmentation capabilities compared to traditional Deep Convolutional GAN (DCGAN) techniques, underscoring its efficacy and potential utility in preprocessing for deep learning applications.

Keywords: GAN; Data Augmentation; Object detection; CycleGAN.

1. Introduction

The ascension of deep learning technologies has underscored the critical role of data volume and quality in dictating model performance. Traditional image data augmentation methods, mainly encompassing simple geometric transformations and color adjustments, often result in images closely resembling their originals, thus adding limited informational value and modestly enhancing model generalization. The advent of Generative Adversarial Networks (GANs) [1] and their variants has sparked innovative solutions to data augmentation challenges, showcasing extensive applicability in areas like object detection [2], text recognition [3-4], super-resolution reconstruction [5-6], and effectively addressing high-dimensional data sampling issues, significantly influencing generative technology advancement. Nonetheless, GANs are susceptible to mode collapse, a training dilemma where models repetitively generate identical samples, stymying further learning. To surmount these hurdles, researchers have introduced several GAN variants: the Least Square Generative Adversarial Network (LSGAN) [7] by Mao et al., which enhances model stability by substituting cross-entropy loss with least squares loss, albeit with limitations in precisely quantifying real versus generated data disparities. The Wasserstein Generative Adversarial Network (WGAN) [8] by Arjovsky et al., employing Earth-Mover (EM) distance over traditional GAN's JS divergence, addresses some LSGAN limitations, yet grapples with weight clipping challenges and the necessity for improved sample stability. The Spectrally Normalized Generative Adversarial Network (SNGAN) [9] by Takeru et al., overcoming WGAN's 1-Lipschitz constraint issues through spectral normalization, boosts model steadiness. Furthermore, Radford et al.'s Deep Convolutional Generative Adversarial Network (DCGAN) [10] is favored for image data augmentation due to its exceptional generative quality. Rafael et al. [12] merged DCGAN with style transfer techniques, substantially raising Parkinson's disease EMG data recognition rates. Despite the prowess of GANs and their iterations in various domains, their application in enhancing object detection tasks remains underexploited. This paper employs the Cycle Generative Adversarial Network (CycleGAN) framework, aiming to ameliorate the diversity and quality of generated images, thus significantly supporting the training and generalization capabilities of deep learning models.

2. Materials and methods

2.1. Data Augmentation

Data augmentation enhances the original dataset through transformations, distortions, and other manipulations to produce novel data samples. In image processing, it's a prevalent strategy aimed at enriching training data diversity, thereby boosting deep learning models' generalization capabilities and robustness.

Image data augmentation techniques can be categorized based on their operational methods and characteristics. The primary categories include:

Geometric Transformations involve creating new image samples by applying translations, rotations, scaling, and flipping operations. This approach enables the simulation of images from varying angles, scales, and positions, enhancing data diversity.

Geometric Transformations [13] involve creating new image samples by applying translations, rotations, scaling, and flipping operations. This approach enables the simulation of images from varying angles, scales, and positions, enhancing data diversity.

Color Transformations generate new images by adjusting parameters such as hue, saturation, and brightness. This simulates different lighting conditions, thereby improving model robustness to changes in illumination.

Color Transformations [14] generate new images by adjusting parameters such as hue, saturation, and brightness. This simulates different lighting conditions, thereby improving model robustness to changes in illumination.

Noise Addition introduces random noises like Gaussian

noise or salt and pepper noise to images to mimic real-world noise conditions. This technique enhances model resilience against noise interference.

Noise Addition [15] introduces random noises like Gaussian noise or salt and pepper noise to images to mimic real-world noise conditions. This technique enhances model resilience against noise interference.

Adversarial Example Generation leverages Generative Adversarial Networks (GAN) to expand datasets by creating challenging adversarial examples. These samples aid in refining model learning on edge and anomaly cases, thus enhancing generalizability.

Adversarial Example Generation leverages Generative Adversarial Networks (GAN) to expand datasets by creating challenging adversarial examples. These samples aid in refining model learning on edge and anomaly cases, thus enhancing generalizability.

2.2. Principles of Recurrent Generative Adversarial Networks

2.2.1. Deep Convolutional Generative Adversarial Networks

Building on the concept of GANs, Radford et al. [10] innovated by integrating Convolutional Neural Networks (CNNs), a staple of supervised learning, into the unsupervised learning framework of generative adversarial networks, resulting in the Deep Convolutional Generative Adversarial Network (DCGAN). This integration significantly enhances the stability of GAN training. A DCGAN comprises a generator (G) and a discriminator (D), which compete to reach a Nash equilibrium.

The architecture of the DCGAN is depicted in Fig. 1. Typically, the hidden variable z is random noise following a Gaussian distribution, fed into the generator to produce synthetic data G(z). The discriminator evaluates G(z) against real data, making a binary judgment and providing feedback to the generator. The optimization process for the discriminator mirrors that of a binary classification challenge.

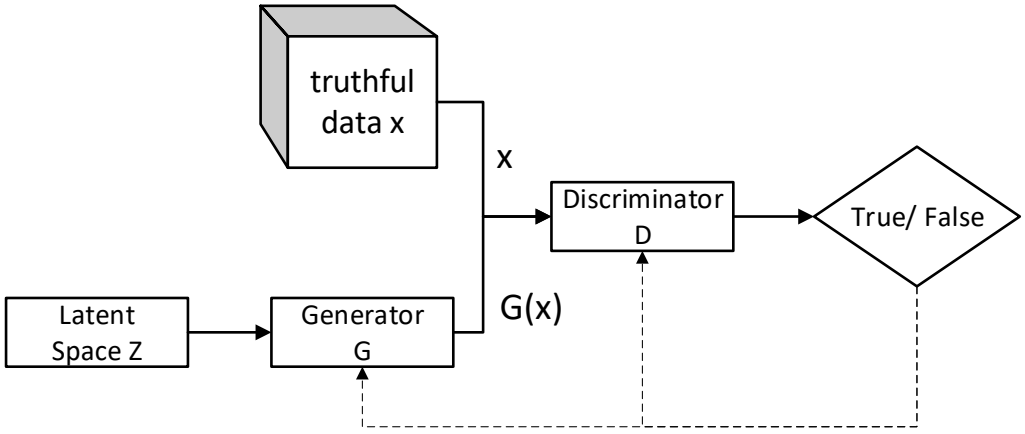

Fig. 1 DGAN model structure diagram

The interaction between G and D is akin to a binary minimax game, with the objective function defined as follows:

$$\min_G \max_D \{V(D,G)\} = E_{x\sim P_{data}(x)}[\log(D(x))] + E_{z\sim P_g(z)}[\log(D(G(z)))]$$

where $V(D,G)$ Is the cross-entropy loss commonly used in binary classification tasks, $P_{data}(x)$ Is the real data distribution, $P_g(z)$ Is the noise distribution,D(x) indicates that x originates from real data, and G(z) represents the generated samples after random noise passes through the generator. During training, the discriminator D aims to maximize the discrimination accuracy, and the generator G aims to minimize the maximum value of the discrimination accuracy of the discriminator D, i.e

$$\min_G \max_D \{V(D,G)\}$$

The training proceeds in two steps:

The first step fixes the generative model, trains the discriminator, and distributes from the real data $P_{data}(x)$ Sample x in, from the prior distribution $P_g(z)$ Sample z in,Generate data through the generator χ ,x and χ input discriminator are trained such that the objective function V is maximized.

The target of the discriminator is:

$$V(G,D) = E_{x\sim P_{data}(x)}[\log(D(x))] + E_{z\sim P_g(x)}[\log(D(z))]$$

After a series of transformations, we obtain:

$$\max_D V(G,D) = -\lg 4 + 2JSD(P_{data}(x), P_g(x))$$

The problem of maximizing V(G,D) is transformed into solving the problem of maximizing JS divergence between real $P_{data}(x)$ and generated data $P_g(z)$.

In the second step, we need to solve the problem of training the discriminator and minimizing the generator under the premise that the value function of the discriminative network is maximized, that is, the problem of minimizing the divergence between the real data and the generated data. According to JS divergence, When $P_{data}(x) = P_g(z)$, the JS divergence takes its minimum value, at this time, the discriminator assigns $P_{data}(x)$ and $P_g(z)$ a score of 0.5, at this time, $\min_G V(G,D) = -\lg 4$. The generator and the discriminator are trained alternately, so that the data generated by the final generator is gradually close to the real data.

2.2.2. CycleGan

Expanding on the foundational concept of GANs, Luo Hao et al. introduced the Cycle Generative Adversarial Network (CycleGAN) [15] to address the challenge of training GANs in the absence of paired sample information. CycleGAN

architecture encompasses two dual GAN networks, each comprising a discriminator and a unidirectional generator.

The CycleGAN model consists of two sets of generator maps G:X→Y and F: Y→X and the associated discriminators DY and DX. DY prompts generator G to convert pictures in source domain X into a generated image that is indistinguishable from pictures in target domain Y. The basic model structure of CycleGAN is shown in Fig. 2

CycleGAN features two mappings, G:X→Y and F: Y→X, along with their corresponding discriminators, DY and DX. DY encourages the generator G to transform images from the source domain X into images that are indistinguishable from those in the target domain Y. Fig. 2 illustrates the basic structure of the CycleGAN model.

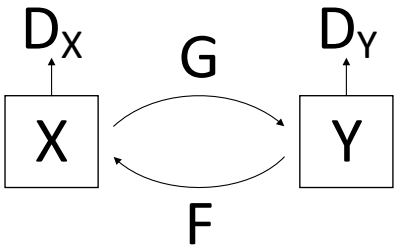

Fig. 2 Schematic diagram of CycleGAN model structure

The essence of CycleGAN lies in its cycle-consistency loss, akin to the content loss in style transfer, ensuring the preservation of original image characteristics in the generated images. The CycleGAN loss function is twofold, comprising adversarial loss and cycle-consistency loss.

The adversarial loss for CycleGAN mirrors that of the classical GAN framework but is applied twice due to the model's dual generator-discriminator setup, yielding two adversarial loss components:

$$L_{GAN}(G, D_Y, X, Y) = E_{y\sim Pdata(y)}[\log(D_Y(y))] + E_{x\sim Pdata(x)}[\log(1-D_Y(G(x)))]$$

$$L_{GAN}(F, D_X, X, Y) = E_{x\sim Pdata(x)}[\log(D_X(x))] + E_{y\sim Pdata(y)}[\log(1-D_X(G(y)))]$$

G(x) denotes the image transformed by generator G from domain X to resemble images in domain Y.

F(y) signifies the image translated by generator F from domain Y to mimic images in domain X. Fig. 3 depicts the cycle-consistency loss, incorporating both the forward cycle-consistency loss F(G(x))≈x and the reverse cycle-consistency loss G(F(y))≈y, expressed as follows:

$$L_{cyc}(G, F) = E_{x\sim Pdata(x)}\left[\|F(G(x)) - x\|\right] + E_{y\sim Pdata(y)}\left[\|G(F(y)) - y\|\right]$$

The CycleGAN loss amalgamates these elements to form its composite objective function.

$$Loss_{full} = Loss_{GAN}(G, D_x) + Loss_{GAN}(F, D_y) + \lambda Loss_{cycle}$$

Training CycleGAN involves iterative refinement where, in the initial phase, generators are trained with fixed discriminators to challenge the discriminators' ability to differentiate between synthetic and real data.

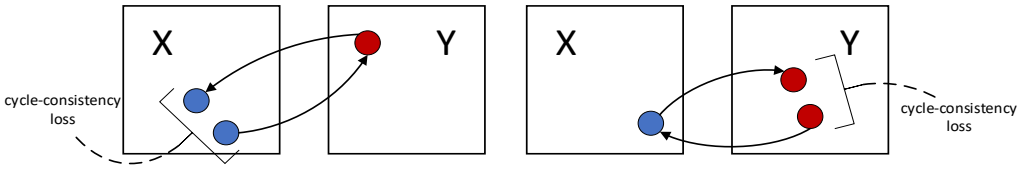

Fig. 3 Schematic diagram of cycle-consistency loss

2.3. CycleGAN Data Augmentation

The data augmentation process utilizing Cycle Generative Adversarial Networks (CycleGAN) comprises several key steps:

2.3.1. Train the CycleGAN model

Initially, images from two distinct feature domains, denoted as X and Y, are compiled. The CycleGAN model is then trained with these datasets, enabling it to assimilate the characteristic features from each domain.

2.3.2. Image feature transformation

Upon completing the CycleGAN model's training, it becomes possible to transfer features from domain X to domain Y. This process ensures the preservation of spatial image data positions while transforming embedded feature data, thereby enriching the training dataset for object detection models with a broader spectrum of features.

2.3.3. CycleGAN Data Augmentation

Integrating the CycleGAN model with conventional image augmentation techniques—hereafter referred to as CycleGAN-based augmentation—as depicted in Fig. 4, enhances the dataset. This entails initially cropping the image randomly, guiding the CycleGAN model for feature transformation, and subsequently splicing the image randomly. Such augmentation enhances the model's adaptability and generalization capacity to novel visual representations.

Fig. 4 CycleGAN data enhancement diagram

3. Experiment and Analysis

3.1. Experimental environment

The experimental setup in this study operates on a Windows 10 platform, equipped with 64GB RAM and an NVIDIA GeForce RTX 3090 Ti 24GB GPU. This configuration furnishes substantial computational power, facilitating the experiments. Utilizing the PyTorch deep learning framework, we trained and evaluated the recurrent generative adversarial networks and object detection models, ensuring both training efficiency and result reliability.

3.2. Dataset preparation

For this investigation, the VOC dataset, encompassing 20 diverse object categories such as humans, vehicles, and animals, was selected as the primary dataset. Its rich visual information underpins the model training process. Leveraging the VOC dataset, we curated three distinct sub-datasets of varying sizes (10, 100, and 1000 images) to support further experimental analyses.

3.3. Evaluation indicators

To quantitatively assess the enhancement effect of the proposed data augmentation method, this study utilizes various metrics: Precision, Class Mean Pixel Accuracy (MPA), and Mean Intersection over Union (MIoU), all derived from a Confusion Matrix.

A Confusion Matrix is a tool used for the assessment of classification model performance, delineating the correspondence between model predictions and actual labels. It's instrumental in supervised learning, particularly for binary and multiclass classification challenges, facilitating a detailed understanding of model accuracy through four distinct outcomes: True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). These terms are defined as follows:

True Positives (TP): Instances correctly identified as Positive.

False Positives (FP): Instances falsely labeled as Positive, actually belonging to the Negative class.

True Negatives (TN): Instances accurately categorized as Negative.

False Negatives (FN): Instances wrongly classified as Negative, despite being Positive.

Precision is calculated by the ratio of correctly predicted positive observations to the total predicted positives:

$$Percision = \frac{p_{ii}}{\sum_{j=0}^{n} P_{ij}}$$

Class Mean Pixel Accuracy (MPA) represents the average pixel accuracy across all object classes within an image:

$$MPA = \frac{1}{n+1}\sum_{i=0}^{n}\frac{p_{ij}}{\sum_{j=0}^{n} p_{ij}}$$

Mean Intersection over Union (MIoU) is the average of the ratio between the intersection and the union of the predicted and actual values for each class, calculated as follows:

$$MIoU = \frac{1}{n+1}\sum_{i=0}^{n}\frac{p_{ii}}{\sum_{j=0}^{n} p_{ij} + \sum_{j=0}^{n} p_{ij} - p_{ii}}$$

Where p_{ii} is the number of correctly classified pixels, p_{ij} is the number of pixels that should have belonged to class i but were classified as class j, and n is the number of classes.

3.4. Model training and testing

This research focuses on the impact of data augmentation techniques on the efficacy of object detection models. The models selected for examination are:

SSD (Single Shot MultiBox Detector)
YOLOv3
YOLOv5
YOLOv8

The experimental procedure is bifurcated into two segments to scrutinize the effects of data augmentation on model performance:

Training with the Original Dataset: Here, no data augmentation is applied, serving as a baseline for comparison against the augmented dataset's outcomes.

Training with Augmented Dataset: This segment employs the CycleGAN-based augmentation to enrich the dataset, intending to simulate varied input scenarios and enhance model generalization and robustness.

Through the above experimental design, the specific effects of data augmentation techniques on the performance of object detection models on datasets of different sizes will be obtained.

3.5. Analysis and evaluation of experimental results

Initially, the SSD model served as the foundation for investigating the impact of CycleGAN-based data augmentation on the model's target detection capabilities, with experiments conducted on datasets of two different sizes. By employing accuracy as the performance metric, it was observed that the models augmented with CycleGAN data showed enhanced performance compared to those without augmentation. The outcomes of these experiments are detailed in Table 1.

Further analysis involved comparative experiments using YOLOv3, YOLOv5, and YOLOv8 models across datasets comprising 10, 100, and 1000 images. The findings indicate a substantial improvement in model generalizability on smaller datasets through CycleGAN augmentation, attributed primarily to the increased diversity of training samples and a reduced risk of overfitting. Notably, for datasets containing merely 10 and 100 images, augmentation facilitated the model's recognition of an expanded range of features, thereby improving test set performance. For the 1000-image dataset, while augmentation still boosted performance, the extent of improvement was less pronounced, suggesting that ample data volume already enables sufficient feature representation learning, rendering the additional diversity from augmentation less impactful on model performance.

The metric mAP(50-95) serves as an indicator of the object detection model's precision, with higher values denoting greater accuracy. Trends depicted in Figure 6 reveal that CycleGAN data augmentation not only accelerates model convergence but also markedly enhances performance across all dataset sizes.

Table 1. Accuracy of SSD models with/without data enhancement at different datasizes

Datasize	With Preprocessing	Without Preprocessing
100	0.39808	0.63613
1000	0.73937	0.75474

Table 2. Indicator data of yolo model under different data set sizes

Model	Precision		MIoU	
	With Preprocessing	Without Preprocessing	With Preprocessing	Without Preprocessing
Yolov3_10	0.91180	0.90391	0.95227	0.82872
Yolov3_100	0.97543	0.96801	0.92464	0.88536
Yolov3_1000	0.95875	0.95368	0.86458	0.86060
Yolov5_10	0.95277	0.93038	0.79036	0.16026
Yolov5_100	0.95985	0.92984	0.92050	0.81800
Yolov5_1000	0.92541	0.89979	0.83607	0.80352
Yolov8_10	0.92733	0.95940	0.87483	0.34069
Yolov8_100	0.96968	0.94509	0.92953	0.81394
Yolov8_1000	0.95761	0.94215	0.86846	0.82614

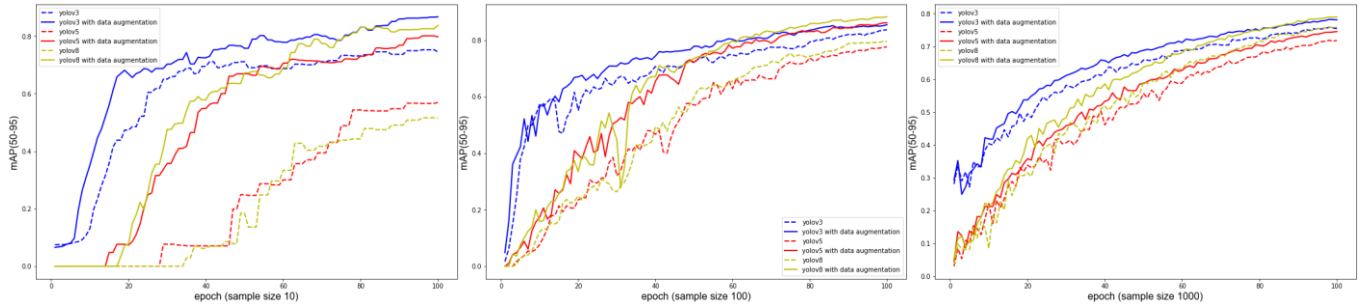

Fig. 5 On datasets of different magnitude, model mAP(50-95) indicators - training batch plots

4. Conclusion

This study introduces a novel data augmentation technique leveraging Cycle Generative Adversarial Networks (CycleGAN), which demonstrably enhances object detection model efficacy across various dataset sizes. Experimental results underscore that both YOLO and SSD models exhibit improved performance on augmented datasets, particularly noticeable within smaller datasets where the effect of data preprocessing is more profound.

The findings affirm CycleGAN's utility in data augmentation, especially for boosting training outcomes on limited-scale datasets. This methodology not only broadens image generation diversity but also effectively counters mode collapse, thereby bolstering training stability without a significant computational overhead.

Moreover, given the observed variances in data preprocessing effects across different models and dataset sizes, future investigations will delve into the underlying causes to refine our data augmentation approach, ensuring its broad applicability and efficacy in diverse application contexts.

Acknowledgements

This work was supported by the Southwest University of Science and Technology Student Innovation Fund Project (Grant No.: S202310619062) and the Southwest University of Science and Technology Student Innovation Training Program (Grant No.: CX23-079).

References

[1] Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., & Bengio, Y. (2014). Generative Adversarial Nets. Neural Information Processing Systems.

[2] Chen Guowei, Liu Lei, Guo Jiayi, Pan Zongxu, Hu Wenlong. Semi-supervised aircraft detection in remote sensing images based on generative adversarial networks [J]. Journal of University of Chinese Academy of Sciences, 2020, 37(4): 539-546.Han B-G, Lee JT, Lim K-T, Choi D-H. License Plate Image Generation using Generative Adversarial Networks for End-To-End License Plate Character Recognition from a Small Set of Real Images. Applied Sciences. 2020; 10(8):2780.

[3] Zhang Gui-mei, LONG Bang-yao, LU Fei-fei. Zero-shot text recognition combined with transfer guidance and bidirectional recurrent structure GAN [J]. Pattern Recognition and Artificial Intelligence, 2020, 33(12): 1083-1096. ZHANG Guimei, LONG Bangyao, LU Feifei. Zero-Shot Text Recognition Combining Transfer Guide and Bidirectional Cycle Structure GAN. , 2020, 33(12): 1083-1096.

[4] MIN Rui, Yang Xuezhi, DONG Zhangyu, et al. SAR image super-resolution reconstruction based on structure enhanced generative adversarial network [J]. Geography and Geo-Information Science, 2021, 37(2):47-53.

[5] Molahasani Majdabadi, M., Ko, SB. Capsule GAN for robust face super resolution. Multimed Tools Appl 79, 31205–31218 (2020).

[6] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang and S. P. Smolley, "Least Squares Generative Adversarial Networks," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017, pp. 2813-2821, doi: 10.1109/ICCV.2017.304. keywords: {Gallium nitride; Generators; Stability analysis; Entropy; Linear programming;Image resolution},

[7] Zhu Y, Jiang B, Jin H, Zhang M, Gao F, Huang J, Lin T and Wang X. (2024). Networked Time-series Prediction with Incomplete Data via Generative Adversarial Network. ACM Transactions on Knowledge Discovery from Data. 18:5. (1-25). Online publication date: 30-Jun-2024.

[8] Miyato, Takeru, Toshiki Kataoka, Masanori Koyama and Yuichi Yoshida. “Spectral Normalization for Generative Adversarial Networks.” ArXiv abs/1802.05957 (2018): n. pag.

[9] Radford, Alec, Luke Metz and Soumith Chintala. “Unsupervised Representation Learning with Deep

Convolutional Generative Adversarial Networks." CoRR abs / 1511.06434 (2015): n. pag.

[10] Anicet Zanini R, Luna Colombini E. Parkinson's Disease EMG Data Augmentation and Simulation with DCGANs and Style Transfer. Sensors (Basel). 2020 May 3;20(9):2605. doi: 10.3390/s20092605. PMID: 32375217; PMCID: PMC7248755.

[11] Zhu, Jun-Yan, Taesung Park, Phillip Isola and Alexei A. Efros. "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks." 2017 IEEE International Conference on Computer Vision (ICCV) (2017): 2242-2251.

[12] Jaderberg, Max, Karen Simonyan, Andrew Zisserman and Koray Kavukcuoglu. "Spatial Transformer Networks." ArXiv abs/1506.02025 (2015): n. pag.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2017. ImageNet classification with deep convolutional neural networks. Commun. ACM 60, 6 (June 2017), 84–90.

[14] Neelakantan, Arvind, Luke Vilnis, Quoc V. Le, Ilya Sutskever, Lukasz Kaiser, Karol Kurach and James Martens. "Adding Gradient Noise Improves Learning for Very Deep Networks." ArXiv abs/1511.06807 (2015): n. pag.