Controlling the quality of GAN-based generated images for predictions tasks

Hajar Hammouch, Mounim El-Yacoubi, Huafeng Qin, Hassan Berbia, Mohamed Chikhaoui

Controlling the Quality of GAN-Based Generated Images for Predictions Tasks

Hajar Hammouch[1,2(✉)], Mounim El-Yacoubi[1(✉)], Huafeng Qin[3(✉)], Hassan Berbia[2(✉)], and Mohamed Chikhaoui[4(✉)]

[1] Institut Polytechnique de Paris, Palaiseau, France
{hajar.hammouch,mounim.el_yacoubi}@telecom-sudparis.eu
[2] Mohammed V University, Rabat, Morocco
h_berbia@yahoo.com
[3] Chongqing Technology and Business University, Chongqing, China
qin_huafeng@163.com
[4] Institute of Agronomy and Veterinary Medicine, Rabat, Morocco
mchikhaoui@gmail.com

Abstract. Recently, Generative Adversarial Networks (GANs) have been widely applied for data augmentation given limited datasets. The state of the art is dominated by measures evaluating the quality of the generated images, that are typically all added to the training dataset. There is however no control of the generated data, in terms of the compromise between diversity and closeness to the original data, and this is our work's focus. Our study concerns the prediction of soil moisture dissipation rates from synthetic aerial images using a CNN regressor. CNNs, however, require large datasets to successfully train them. To this end, we apply and compare two Generative Adversarial Networks (GANs) models: (1) Deep Convolutional Neural Network (DCGAN) and (2) Bidirectional Generative Adversarial Network (BiGAN), to generate fake images. We propose a novel approach that consists of studying which generated images to include into the augmented dataset. We consider a various number of images, selected for training according to their realistic character, based on the discriminator loss. The results show that, using our approach, the CNN trained on the augmented dataset generated by BiGAN and DCGAN allows a significant relative decrease of the Mean Absolute Error w.r.t the CNN trained on the original dataset. We believe that our approach can be generalized to any Generative Adversarial Network model.

Keywords: Deep neural networks · Generative Adversarial Networks · Control of GAN output quality · Regression task

The authors thank Campus France and Morocco CNRST for the financial support of this research, and Telecom SudParis, and Univ. Mohammed V during the different phases of this study. This work was also supported in part by the National Natural Science Foundation of China under Grant 61976030 and the funds for creative research groups of Chongqing Municipal Education Commission under Grant CXQT21034.

M. El Yacoubi et al. (Eds.): ICPRAI 2022, LNCS 13363, pp. 121–133, 2022.
https://doi.org/10.1007/978-3-031-09037-0_11

1 Introduction and Related Work

Over recent decades, the decrease in water available for irrigation around the world has become a major issue especially in arid and semi-arid regions [1]. A useful solution involves estimating the amount of water required based on soil moisture [2]. Deep Learning has shown great potential in many agriculture tasks such as fruit counting, plant diseases recognition and soil moisture prediction [3]. Convolutional Neural Networks (CNNs) are ones of the most common deep learning models, but they require large amounts of training data. To overcome this problem, Generative adversarial networks (GANs) are a well-known effective technique for augmenting the training data sets [4,5]. Although GANs provide excellent results, not all their generated samples are realistic [6]. Evaluating GANs, or more specifically, the samples generated by GANs, is a challenging process [7]. From the state of the art, structural similarity (SSIM) and Peak Signal to Noise Ratio (PNSR) are the most two used image quality assessment metrics. They are based on the realism and diversity of the generated images [7]. SSIM compares the corresponding pixels and their neighborhoods in two images based on the luminance, the contrast and the structure, while PNSR computes the peak signal-to-noise ratio between two samples [7]. Although SSIM and PNSR are currently the most popular metrics, they are increasingly criticized [8,9]. They may fail capturing fine features in the images and provide high scores to images with bad quality [7]. Furthermore, the scores are not always well correlated with human perception [10]. In other words, the generated sample with high metrics values does not always appear better to humans than the sample with lower metrics values [10]. In addition to the image quality metrics mentioned above, some other measures such as Inception Score (IS) and Fréchet Inception Distance (FID) [11], have been widely adopted for evaluating GANs [12]. These two measures use a pre-trained Inception neural network trained on the ImageNet dataset to extract the features and capture the required properties of generated images [7]. However, the two Inception-based measures do not consider real images at all, and so cannot measure how well the generator approximates the real distribution [7]. In other words, these scores are limited to measuring only the diversity of the generated images [11]. Furthermore, IS and FID focus on the evaluation of the generative models, which cannot be utilized to evaluate the quality of each single fake image [13]. In addition, better FID (i.e., lower) does not always imply better image quality [14]. The authors in [15], aimed to increase their original limited agriculture data set using GAN for a segmentation task, by proposing a GAN that automatically generates realistic agriculture scenes. They only generate instances of the objects of interest in the scene, not all the image. Furthermore, they proposed an empirical evaluation where they extract features from the original and generated samples to calculate performance using various metrics such as: IS and FID. These measures showed that the fake samples' quality was similar to original samples' quality. Then, without selecting the best generated images, all the augmented synthesized images were added to the original data set to train a CNN model for a crop segmentation purpose. The results showed that the GAN enhanced dataset improved the performance

of four different state-of-the art segmentation architectures. The authors in [4], have proposed a model to develop a deep learning based framework to recognize the tomato plant diseases by investigating tomato leaf images. They generated synthetic images from the PlantVillage dataset using conditional GAN model for a classification purpose. The generated samples were first evaluated visually and then using two metrics: PNSR and IS. The comparable values of PNSR and IS showed that the quality of images generated by C-GAN were very close to the quality of real images. Then, in the second experiment, the models were trained first on the original data set and then on the augmented data set. The results show that the augmented data set gives better accuracy as compared to the original dataset. In [13], the authors used a GAN-based data augmentation technique to improve the training performance. In this work, the proposed GAN model has been assessed by various metrics such as: Inception Score, Fréchet Inception Distance, Precision and Recall. The results showed that, by adding the generated samples to the original set, the proposed model results in significant performance accuracy. Overall, GANs technology is quickly evolving and is being used in more and more agriculture applications for classification or segmentation. State of the art methods, however, add all the generated images into the augmented data for a classification or a segmentation purpose [4,13,15]. Sometimes, they just evaluate the generated data, but do not assess which ones to select for the addition. However not all generated images are realistic and relevant [6]. Our study is devoted to handle this issue by proposing a framework that studies which generated images to include in the augmented dataset. This approach allows to train a deep learning model using only the most realistic generated images. In this paper, and as presented in our previous work [16], we are interested in the prediction of soil moisture dissipation rates from synthetic aerial images using a CNN model. This entails a regression task where the CNN, given an input image, has to predict a vector of regression values associated with the dissipation rates at different locations of the agricultural field. CNNs, however, require large data sets to successfully train the deep network. To this end, we propose and compare between two Generative adversarial networks models: (1) Deep Convolutional Generative Adversarial Network (DCGAN) and (2) Bidirectionnal Generative Adversarial Network (BIGAN) using the same architectures for generating fake agriculture images. Then, we conduct several experiments to study which generated images to include in the augmented dataset. We consider different numbers of generated images, namely from 250 to 1200 images, with four scenarios regarding the discriminator loss ranges: (1) 0.20–0.80, (2) 0.30–0.70, (3) 0.40–0.60 and (4) 0.45–0.55. Additional augmented data did not improve the results. Following that, we use the baseline CNN model trained on the original data set to predict the regression vector serving as its ground truth in the augmented dataset. The results show that the CNN trained on the augmented data set generated by BiGAN and DCGAN models allowed respectively a relative decrease of the Mean Absolute Error by 19.34% and 14.23% instead of 16.05% and 12.40% (using all the generated images) w.r.t to the CNN trained on the original data set. This shows that the proposed GANs-based generated

images quality assessment approach can add high-quality agriculture images to the training set efficiently, leading to performance improvement of the regression model. To the best of our knowledge, this is the first paper using this approach, that can be applied to any Generative Adversarial Network model.

In summary, the main contributions of this article are outlined as follows:

- By adding conv layers and an encoder on the standard GAN respectively, we compare two known models using the same architectures: (1) Deep Convolutional Generative Adversarial Network (DCGAN) and (2) Bidirectional Generative Adversarial Network (BiGAN) for generating fake agriculture images.
- We propose a framework where we study which images to include into the augmented dataset by considering different sizes of images that are generated from 250 to 1200 images with four scenarios regarding the discriminator loss ranges: (1) 0.20–0.80, (2) 0.30–0.70, (3) 0.40–0.60 and (4) 0.45–0.55.
- We use the basic CNN model to generate the regression ground truth corresponding to the generated images.
- To evaluate the performance of our proposed approach, a publicly available agriculture RAPID data set has been used in the experiments. The presented approach is applicable using any other Generative adversarial networks model.

The remainder of this paper is as follows. The background is briefly presented in Sect. 2. The used methods and materials are described in Sect. 3. Then, in Sect. 4, we evaluate the experimental results, followed by a discussion. The last section, Sect. 5, concludes the paper.

2 Background

Generative Adversarial Network: GAN is a deep learning model consisting of two modules: a discriminator D and a generator G [17]. D determines whether the data is real or fake, and G transforms a random vector into a realistic data, by fooling the discriminator into accepting its generated data as original data [17]. During training, G improves on generating fake samples, while D learns to distinguish between the original and generated examples [15]. The equilibrium is reached when G generates realistic images similar to real samples [17]. The discriminator will be totally confused in the ideal case, when guessing with 50% probability that all samples, whether original or generated, are fake [17]. GAN aims to reduce the probability distribution distance between real and generated samples. GAN training involves optimizing the following loss function:

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

where $E_{x\sim p_{\text{data}}(x)}$ is the expectation over the original samples, $E_{z\sim p_{\text{generated}}(z)}$ is the expectation over the generated samples, $p_{\text{generated}}(z)$ is the noise vector, $D(x)$ is the discriminator loss given the probability that x comes from the training data distribution, $G(z)$ is the generator loss and x represents the original dataset.

During training, the generator and discriminator networks are trained in competition with each other, and both play a two players minimax game, i.e., the generator aims to minimize (1), while the discriminator aims to maximize it. The characteristics of (1) are that $\log D(x)$ is maximized by the discriminator, so that the original and generated samples are properly classified. Besides, $\log (1 - D(G(z))$ is minimized by training the generator. Theoretically, achieving $p_{\text{generated}} = p_{\text{data}}$ is the solution to this minimax game when D starts randomly guessing whether the inputs are real or generated.

Bidirectionnal Generative Adversarial Network: The encoder is the difference between BiGAN and GAN models. The original GAN can generate an image for any random noise vector but cannot generate the corresponding random noise for any given generated image. Donahue et al. [18] proposed the bidirectional GAN which contains an additional encoder component E [18]. In addition to the encoder E which maps the data space to the latent space, BiGAN allows for the extraction of additional data characteristics and to perform the learning process [19]. With three components, encoder E, generator G and discriminator D, true images are fed into the encoder E to learn the feature representation E(x) while z, considered as a noise vector in the latent space, is input into the generator G to generate synthetic representation data G(z), and finally the obtained pairs (z, G(z)) and (E(x), x) are used to train the discriminator D. As a result, $\hat{z}$ = E(x) can be used as low-dimensional characteristics extracted by the encoder from data images. The structure of discriminator D is altered. Instead of using x or $\hat{x}$ = G(z) as the input of D, the discriminator discriminates joint pair (x,$\hat{z}$) or ($\hat{x}$, z) from data space and latent space, respectively. As GAN model, the purpose of BiGAN can be defined as a minimax objective:

$$\min_{G,E} \max_{D} \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x, E(x))] + \mathbb{E}_{z \sim p_{\text{generated}}(z)}[\log(1 - D(G(z), z)] \quad (2)$$

3 Materials and Methods

Dataset: We use RAPID [20], the only public available dataset. The examples are pairs of synthetic aerial images and corresponding soil moisture dissipation rate values. It contains 1400 aerial images of a vineyard with known soil moisture levels. The dataset is originally split into a train and test sets, of 1200 and 200 images respectively [20]. Each image contains 200 equidistant plants distributed in 10 columns and 20 rows, associated with a vector of 200 values of soil moisture values. Figure 1 shows an example of three images resized to 128×128. Yellow refers to higher dissipation rates, whereas green refers to lower rates.

Convolution Neural Network: We use a CNN regression model with Euclidean loss. To evaluate the accuracy, we use the Mean Absolute Error (MAE) metric (3), where y_{ij} is the actual output value (ground truth), i.e., the dissipation rate at plant j in image i, $\hat{y}_{ij}$ represents the predicted value, n the number of image plants,p the number of dataset elements, $1 \leq i \leq p$, $0 \leq j \leq n$, and the values of MAE are positive. The closer MAE to zero, the better the prediction.

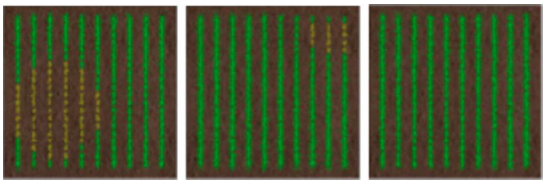

Fig. 1. Original images resized to 128×128. (Color figure online)

$$MAE = \frac{1}{p}(\frac{1}{n}\sum_{i=1}^{p}\sum_{j=0}^{n}|\hat{y}_{ij} - y_{ij}| \tag{3}$$

Our CNN contains six convolution layers, followed by a fully connected layer, with ReLU as the activation function. Inputs to the CNN are images of size $128 \times 128 \times 3$. The first layer contains 32 filters of size (3×3), followed by a dropout layer. The second layer has 32 filters of size (3×3), followed by a max pooling layer with size 2×2 and stride 0. The third layer is composed of 64 filters with size 3×3 and stride 0, followed by a dropout layer. The fourth layer has 64 filters with size 3×3. The following max pooling is of size 2×2 and stride 0. The fifth layer contains 128 filters of size (3×3), followed by a dropout layer. The last convolution layer has 128 filters of size (3×3), and is followed by dropout and max pooling layers of pooling size 2×2 with stride 0. This is followed by a fully connected layer of dimension 200 that predicts the vector of 200 soil moisture dissipation rates. The CNN architecture, is obtained by a Greedy optimization method on the validation set, of various hyperparameters such as: filter size, batch size, optimizer, learning rate, number of epochs, dropout.

Proposal and Experimental Design: In this work, we aim to develop an effective framework assessing the quality of the generated images, that allows selecting the generated images to be included into the augmented data. To this end, we use two GANs: DCGAN and BiGAN. In our scheme, not all generated images are added into the augmented data set, but only those who satisfy a quality criterion given by the discriminator loss, as the latter gives us a quantitative hint to how similar a generated image is to an original one. Concretely, we conduct several experiments where we consider different sizes of images that are generated from 250 to 1200 images with four scenarios regarding the discriminator loss. In one of the scenarios, for instance, we do not want to include very easy images for which the discriminator loss is close to 0, as these samples are too different from the real images, so inclusion may actually hinder the quality of the training data. We want rather realistic images that show variability and diversity with respect to the training data. We consider the first experiment where we include only images really close to the real ones by considering a loss between 0.45 and 0.55, so we are 5% on each size of the middle (of 50%), where the discriminator has total confusion on whether the image is real or fake. The second range is between 0.40 and 0.60, where we can afford more variability in

the training data. We can go further with a loss between 0.40 and 0.60 or even further with a discriminator loss between 0.20 and 0.80 that allows for more variability. Subsequently, the resulting samples along with their predicted soil moisture dissipation rates, used as their ground truths, will be used to augment the training set for a more robust CNN parameter estimation.

Experimental Details: For all the experiments, an Adam optimizer (LR = 0.0002) is used in our architectures to update the classifier parameters. The number of epochs and batch size were set respectively to 200 and 128, and the binary cross entropy is used as loss function.

Generator Network: The generator, shown in Fig. 2(b), takes, as input, a 100×1 random noise, and generates an image of size $128 \times 128 \times 3$. It is composed of four conv layers and three dense layers. The activation function LeakyReLU is applied for all layers except for the output layer that uses Tanh. An Upsampling layer is used after all conv layers, except after the input layer. Finally, batch normalization is used for the first two dense layers to stabilize the learning process by normalizing the input to have zero mean and unit variance.

Discriminator Network: The discriminator (see Fig. 2(c)) classifies whether the images are real or generated [17]. It takes as input images of size $128 \times 128 \times 3$. In this architecture, the input sample undergoes transformation embeddings through two conv layers and three Dense layers, followed by the Sigmoid activation function to predict whether the sample is real or generated. The LeakyReLU activation function follows each conv and dense layers except for the output layers. Finally, dropout layer is used for the first two dense layers.

Encoder Network: Our BiGAN generator and discriminator architectures are similar to our Deep Convolutional GAN (DCGAN). The architecture of the encoder E (see Fig. 2(a)), contains four convolutional layers and three Dense layers followed by batch normalization layers, with LeakyReLU as activation function. The BIGAN discriminator concatenates both the data point x and its latent z as input, which is then passed through two convolutional layers and three dense layers serving as fully connected layers before obtaining a scalar as output.

Ground truth Prediction: To get a full augmented dataset, we use the same CNN to generate the ground truth of the fake samples generated by GAN and BiGAN models (Fig. 3). The latter, therefore, consist of GAN-based and BiGAN-based generated images, along with their generated ground truth vector.

4 Results

Convolutional Neural Network: As a baseline, we consider the CNN of Sect. 3, applied to the original data only, without data augmentation. Table 1 shows the accuracy and loss for the training, validation, and test sets.

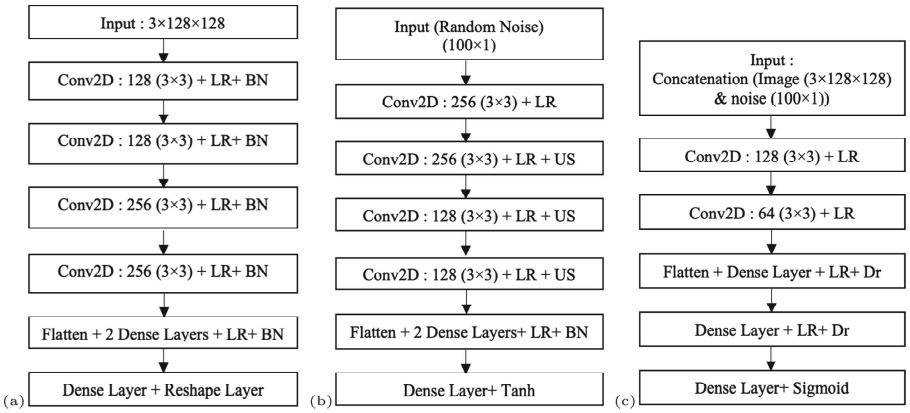

Fig. 2. (a) Encoder Architecture, (b) Generator Architecture, (c) Discriminator Architecture; LR: LeakyReLU, US: UpSampling2D, Dr: Dropout, BN: BatchNormalization.

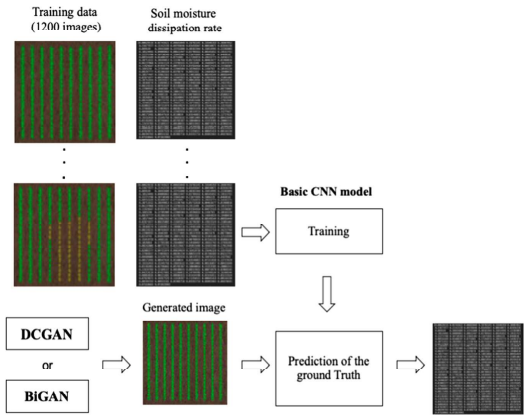

Fig. 3. Prediction of the ground truth using DCGAN or BiGAN model.

Table 1. CNN results using the original data.

	Loss: MSE	Metric: Mae
Train	0.0014	0.0288
Validation	0.0013	0.0267
Test	0.0013	0.0274

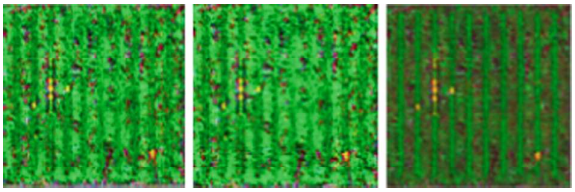

Fig. 4. Images with a discriminator loss close to 0.

Deep Convolutional GANs: DCGAN was able to produce samples that look like the real data at about 80 epochs. The quality of these fake samples continued to improve until roughly 200 epochs. However, and as mentioned above, not all generated samples are realistic. It makes sense, therefore, not to include very easy images, where the discriminator loss is close to 0, as shown in Fig. 4. Therefore, we considered different numbers of generated images, from 250 to 1200 images, with four scenarios regarding the discriminator loss ranges: (1) 0.20–0.80, (2) 0.30–0.70, (3) 0.40–0.60 and (4) 0.45–0.55. Figure 5(a), (b), (c) shows a grid of real and generated samples corresponding to (1) and (4). Visually, the number of columns is the same for all cases. Furthermore, the distribution of the colors is the same, except for (1) and (2), where the distribution of yellow color is quite different w.r.t the original samples. As shown in Fig. 5(c), the best generated images correspond to the last case, where the discriminator loss is between 0.45 and 0.55. Therefore, we conclude that the more the discriminator loss is closer to 0.5, the better the quality of the images is.

$$Absolute\,change = Mae_{final} - Mae_{initial} \tag{4}$$

$$Relative\,change = (\frac{Mae_{final} - Mae_{initial}}{Mae_{initial}}) \times 100 \tag{5}$$

For each case study, the selected best synthesized images were added to the original training dataset to show how much the regression performance can be improved by our quality assessment framework. The evaluation is performed based on the MAE metric. First, we set 1000 original images and 200 original images in the training validation sets respectively. 250, 500, 750, 1000, 1200 fake images were added gradually to the original training data to get respectively: 1250, 1500, 1750, 2000 and 2200 augmented training data sets. Additional augmented data did not improve the results. The results show that for all cases, the more generated images are added, the better the result is. For all cases, the best result was the last case when adding 1200 generated images reaching a validation MAE of 3.18%, 2%, 3.10% and 2.72% where the discriminator loss is respectively between: (1) 0.20–0.80, (2) 0.30–0.70, (3) 0.40–0.60 and (4) 0.45–0.55. We considered, therefore, this optimal configuration, and tested for each case the associated trained model on the held-out 200 test images. We obtained a MAE of 2.93%, 2.92%, 2.91% and 2.35%. The best result was the last case, when discriminator loss was set between 0.45 and 0.55, and when adding 1200

generated images reaching a validation MAE of 2.35% instead of 2.74%, which amounts to a relative decrease of the MAE of 14.23% instead of 12.40% without using our approach (i.e., adding all the generated images) (see Eqs. (4) and (5)).

Bidirectionnal Generative Adversarial Networks: BiGAN was trained using the same hyperparameters as DCGAN model. The architectures of the generator and discriminator were also the same. BiGAN, however, was able to produce samples that look like the real data at about just 20 epochs. The quality of these generated samples continued to improve until roughly 200 epochs. Figure 5(a), (b'), (c') shows a grid of real and generated samples corresponding to (1) and (4). Visually, the number of columns is the same in all cases. Furthermore, the distribution of the colors is the same except for the first two cases (1) and (2) where the distribution of yellow color is quite different w.r.t the original data. The best generated images correspond to the last two cases (3) and (4), where the discriminator loss is between 0.40–0.60 and 0.45–0.55. Visually, the quality of images generated by BiGAN model outperforms those generated by DCGAN. However, in both cases, the generated images continue to improve when the discriminator loss gets closer to 0.5. Using the same DCGAN experiments, we note again that, for all cases, the more generated images are added, the better the result is. Here again, for each case, the best result was the last when adding 1200 generated images, reaching respectively a validation MAE of 2.66%, 2.44%, 2.34% and 2.25% where the discriminator loss is between

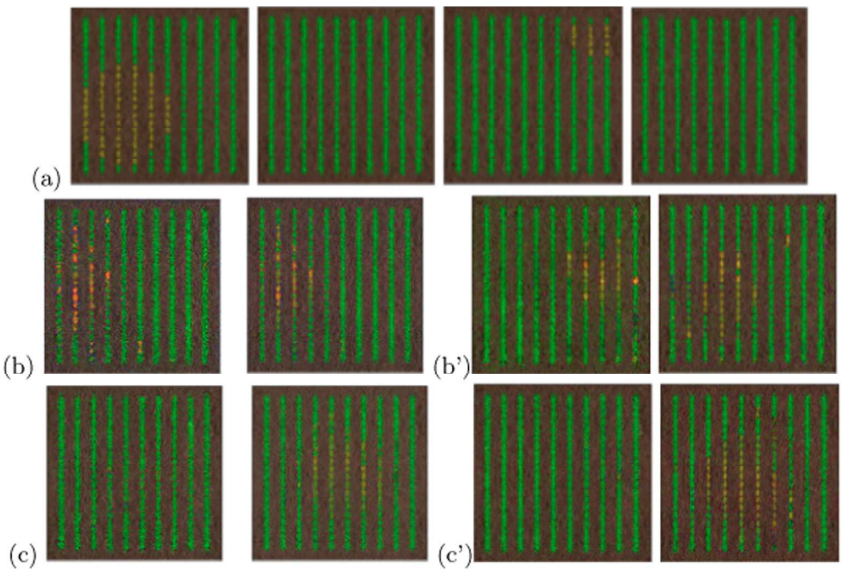

Fig. 5. (a) Original images; Generated images using DCGAN and BiGAN respectively and corresponding to a discriminator loss between: (b)–(b') 0.20 and 0.80, and (c)–(c') 0.45 and 0.55.

(1) 0.20–0.80, (2) 0.30–0.70, (3) 0.40–0.60 and (4) 0.45–0.55. We considered, therefore, this optimal configuration, and tested the associated trained model on the held-out 200 test images. We obtained a MAE of 2.39%, 2.30%, 2.27% and 2.21%. The best result was the last case where the discriminator loss is between 0.45 and 0.55 and when adding 1200 generated images reaching a validation MAE of 2.21% instead of 2.74%, which amounts to a relative decrease of the MAE of 19.34% instead of 16.05% without using our approach (i.e., adding all the generated images). Additional augmented data did not improve the results. Table 2, summarizes the best results (i.e., generating 1200 images) obtained by both DCGAN and BiGAN, using and without using our approach. Thus, the performance of the CNN model has been further improved by a relative decrease of the MAE of 12.40% and 16.05% (including all the generated images), 14.23% and 19.34% (using our approach) through the DCGAN and BiGAN models respectively. Another interesting finding is that, for all cases, the time consumption for generating images using DCGAN model was much larger than that of generating images using BiGAN model. In addition, using our approach on four cases, we showed that our BiGAN model provides better performances in terms of the MAE metric and also in terms of images generation quality compared to the DCGAN model. This result is explained by the good feature extraction capability of the encoder. In other words, by adding the encoder, BiGAN is able to better extract the features from the data. Overall, our proposed selection framework improves significantly the CNN prediction performance using either DCGAN or BiGAN.

Table 2. Final results; W.A: Without using our approach, R.C: Relative Change, Case1: 0.20–0.80, Case2: 0.30–0.70, Case3: 0.40–0.60, Case4: 0.45–0.55

Baseline	Mse-Train		Mae-Train				Mse-Val		Mae-Val		Mae-Test	
	0.0014		0.0288				0.0013		0.0267		0.0274	
	BiGAN Model						DCGAN Model					
	Train set		Validation set		Test set	R.C	Train set		Validation set		Test set	R.C
	MAE	MSE	MAE	MSE	MAE		MAE	MSE	MAE	MSE	MAE	
W.A	7.07.10−4	0.0197	0.0011	0.0246	0.0230	**16.05%**	4.67.10−4	0.0150	0.0013	0.0261	0.0240	**12.40%**
Case 1	9.95.10−4	0.0222	0.0012	0.0266	0.0239	–	6.08.10−4	0.0172	0.0018	0.0318	0.0293	–
Case 2	0.0013	0.0254	0.0010	0.0244	0.0230	–	6.67.10−4	0.0181	6.90.10−4	0.02	0.0292	–
Case 3	3.01.10−4	0.0116	9.84.10−4	0.0234	0.0227	–	7.10−4	0.0187	0.0018	0.0310	0.0291	–
Case 4	4.93.10−4	0.0155	8.66.10−4	0.0225	0.0221	**19.34%**	7.11.10−4	0.0181	7.20.10−4	0.0272	0.0235	**14.23%**

5 Conclusion

In this work, we have proposed a quality GAN-based generated images framework, where we control the discriminator loss to augment the training data set with the most realistic generated agricultural images for a regression task. We used and compared two generative adversarial networks models (DCGAN and BiGAN) to generate fake images from 250 to 1200 with four scenarios, with a baseline CNN used to generate the regression ground truth vector for each fake

image. We have shown that BiGAN model outperforms DCGAN model in terms of the MAE metric and of the quality of the generated images. Augmenting the training data set, using our approach, is effective and allows significantly improving the regression performance with a relative decrease of the Mean Absolute Error (MAE) of 19.34% and 14.23% using BiGAN and DCGAN respectively instead of 16.05% and 12.40% where all the generated images were added into the training data. In our future work, we intend to build our own dataset from an agriculture field in Morocco. We will also apply our approach using the new collected dataset with various other GAN-based network architectures.

References

1. Papadavid, G., Hadjimitsis, D., Fedra, K., Michaelides, S.: Smart management and irrigation demand monitoring in Cyprus, using remote sensing and water resources simulation and optimization. Adv. Geosci. **30**, 31–37 (2011)
2. Hassan-Esfahani, L., Torres-Rua, A., Jensen, A., Mckee, M.: Spatial root zone soil water content estimation in agricultural lands using Bayesian-based artificial neural networks and high resolution visual, NIR, and thermal imagery. Irrig. Drain. **66**, 273–288 (2017)
3. Jha, K., Doshi, A., Patel, P.: Intelligent irrigation system using artificial intelligence and machine learning: a comprehensive review. Int. J. Adv. Res. **6**(10), 1493–1502 (2018)
4. Abbas, A., Jain, S., Gour, M., Vankudothu, S.: Tomato plant disease detection using transfer learning with C-GAN synthetic images. Comput. Electron. Agric. **187**, 106279 (2021)
5. Qin, H., El Yacoubi, M., Li, Y., Liu, C.: Multi-scale and multidirection GAN For CNN-based single palm-vein identification. IEEE Trans. Inf. Forensics Secur. **16**, 2652–2666 (2021)
6. Gu, S., Bao, J., Chen, D., Wen, F.: GIQA: generated image quality assessment. arXiv preprint arXiv:2003.08932 (2020)
7. Borji, A.: Pros and cons of GAN evaluation measures. Comput. Vis. Image Underst. J. **179**, 41–65 (2019)
8. Zhu, X., et al.: GAN-Based Image Super-Resolution with a Novel Quality Loss. Mathematical Problems in Engineering (2020)
9. Jean-François, P., Rhita, N.: Limitations of the SSIM quality metric in the context of diagnostic imaging. In: International Conference on Image Processing, ICIP 2015, pp. 2960–2963. IEEE, Canada (2015). https://doi.org/10.1109/ICIP.2015.7351345
10. Kovalenko, B.: Super resolution with Generative Adversarial Networks (n.d.)
11. Borji, A.: Pros and cons of GAN evaluation measures: new developments (2021). http://arxiv.org/abs/2103.09396
12. Shmelkov, K., Schmid, C., Alahari, K.: How good is my GAN? In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11206, pp. 218–234. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01216-8_14
13. Qin, Z., Liu, Z., Zhu, P., Xue, Y.: A GAN-based image synthesis method for skin lesion classification. Comput. Meth. Program. Biomed. **195**, 0169–2607 (2020)
14. Zhao, Z., Zhang, Z., Chen, T., Singh, S., Zhang, H.: Image augmentations for GAN training (2020). http://arxiv.org/abs/2006.02595

15. Fawakherji, M., Ptena, C., Prevedello, I., Pretto, A., Bloisi, D.D., Nardi, D.: Data augmentation using GANs for crop/weed segmentation in precision farming. In: CCTA 2020 Conference, Montréal, pp. 279–284. IEEE Xplore (2020). https://doi.org/10.1109/CCTA41146.2020.9206297
16. Hammouch, H., El Yacoubi, M., Qin, H., Berrahou, A., Berbia, H., Chikhaoui, M.: A two-stage deep convolutional generative adversarial network-based data augmentation scheme for agriculture image regression tasks. In: International Conference on Cyber-physical Social Intelligence, CSI 2021, Beijing. IEEE Xplore (2021)
17. Goodfellow, I., Pouget-Abadie, J., Mirza, M.: Generative adversarial networks. Commun. ACM **63**(11), 139–144 (2020)
18. Donahue, J., Krähenbühl, P., Darrell, T.: Adversarial feature learning (2016). http://arxiv.org/abs/1605.09782
19. Cui, L., Tian, X., Shi, X., Wang, X., Cui, Y.: A semi-supervised fault diagnosis method based on improved bidirectional generative adversarial network. Appl. Sci. **11**(20), 9401 (2021). https://doi.org/10.3390/app11209401
20. Tseng, D., et al.: Towards automating precision irrigation: deep learning to infer local soil moisture conditions from synthetic aerial agricultural images. In: CASE 2018 Conference, Munich, pp. 284–291. IEEE (2018). https://doi.org/10.1109/COASE.2018.8560431