Leveraging Semi-Supervised Generative Adversarial Networks to Address Data Scarcity Using Decision Boundary Analysis

Mohamed Ouriha[1], Omar El Mansouri[2], Younes Wadiai[3], Boujamaa Nassiri[4],
Youssef El Mourabit[5], Youssef El Habouz[6]
TIAD Laboratory, Sciences and Technology Faculty, Sultan Moulay Slimane University, Beni Mellal, Morocco[1,2,5]
Laboratory of Innovative Systems Engineering-National School of Applied Sciences of Tetouan,
Abdelmalek Essaâdi University, Tetouan, Morocco[3]
InterDisciplinaire Applied Research Laboratory, LIDRA International, University of Agadir Unversiapolis, Agadir, Morocco[4]
IGDR Umr 6290 Cnrs Rennes University, Rennes, France[6]

***Abstract*—Convolutional Neural Networks (CNNs) are widely regarded as one of the most effective solutions for image classification. However, developing high-performing systems with these models typically requires a substantial number of labeled images, which can be difficult to acquire. In image classification tasks, insufficient data often leads to overfitting, a critical issue for deep learning models like CNNs. In this study, we introduce a novel approach to addressing data scarcity by leveraging semi-supervised classification models based on Generative Adversarial Networks (SGAN). Our approach demonstrates significant improvements in both efficiency and performance, as shown by variations in the evolution of decision boundaries and overall accuracy. The analysis of decision boundaries is crucial, as it provides insights into the model's ability to generalize and effectively classify new data points. Using the MNIST dataset, we show that our approach (SGAN) outperform CNN methods, even with fewer labeled images. Specifically, we observe that the distance between the images and the decision boundary in our approach is larger than in CNN-based methods, which contributes to greater model stability. Our approach achieves an accuracy of 84%, while the CNN model struggles to exceed 72%.**

***Keywords*—Decision boundary; convolutional neural network; Generative Adversarial Networks; MNIST; classification; semi-supervised classification**

I. Introduction

Deep learning (DL), a branch of machine learning (ML), is characterized by its significant flexibility and learning power. It represents the world through concepts organized in nested hierarchies, where each concept is defined in simpler terms and more abstract representations [1]–[6].

One of the essential skills in computer vision is the accurate classification of images. The development of image collection equipment, combined with the widespread use of digital platforms, has led to an exponential growth in the volume of digital data, necessitating the creation of robust and advanced models to analyze this vast influx of visual information. Deep learning has been proposed for image classification due to its capability to provide more detailed insights into a subject's response to specific visual stimuli. Recent research indicates that strategies based on deep learning have yielded impressive results [7].

Image classification is one of the most common challenges in computer vision. The success of a classification system is highly dependent on the quality of the attributes derived from an image, with the accuracy of the results improving in proportion to the quality of these features. These attributes are often utilized in supervised learning, where a set of features X (usually extracted from an image) is employed to predict a certain outcome Y. Before the widespread adoption of deep learning in 2012, commonly used machine learning models included support vector machines, artificial neural networks, and random forests; these traditional methods were the primary techniques for processing computer vision tasks [8].

Convolutional neural networks (CNNs) are now the most popular method for image analysis and classification due to the growing interest in deep learning. CNNs have achieved significant results across a wide range of classification problems. Despite their tremendous potential, they continue to face several challenges. These difficulties are largely due to the vast scale of the networks, which may contain millions of parameters, a lack of sufficient training datasets, overfitting issues, and poor generalization capabilities. Additionally, a growing concern among researchers is the need to prevent adversarial attacks that could mislead deep neural networks (DNNs) [9].

To address these issues and enhance performance, researchers are modifying network architectures, developing new learning algorithms, and acquiring more data. A typical challenge is the scarcity of high-quality data or an unequal distribution of classes within datasets. Currently, the most efficient DNNs are quite large and require massive amounts of data, which can be difficult to obtain. For example, the popular CNN architecture VGG16 has 16 layers of neurons and 138 million parameters [10].

Generally, deep learning algorithms are considered data-hungry, necessitating many labeled images to produce the desired fits. This requirement may render these technologies inaccessible for smaller projects, which often have limited datasets. Data augmentation [11] and transfer learning [12] are two approaches to addressing this challenge. We continue along this path to find a relevant solution by presenting a semi-supervised approach (SGAN) with a novel learning technique: utilizing both labeled and unlabeled images based on Gener-

ative Adversarial Networks. We compare our approach with CNNs using the same dataset.

To evaluate these approaches, we focus on a crucial aspect that plays an indispensable role in understanding deep learning: the decision boundary. For each approach, we will examine how the decision boundary evolves during training.Our approaches demonstrate excellent performance in image classification. A brief description of GANs, CNNs, and decision boundaries is provided below.

The rest of the paper is organized as follows: Section II delineates the methodology and the proposed approach. Section III is dedicated to the presentation of results and their subsequent discussion. Finally, Section IV offers the concluding remarks.

II. Methodology

A. Convolutional Neural Networks (CNN)

A convolutional neural network [13]–[16] comprises an input layer, an output layer, and several hidden layers. Each layer converts a set of activations into another using a differentiable function. Generally, there are three main types of hidden layers: convolutional layers, pooling layers, and fully connected layers (Fig. 1).

1) Convolution layer: Convolution is performed by translating the convolution kernel through the input image matrix. To establish a network of local connections, each neuron in the local window is linked to a corresponding neuron in the convolution layer. This configuration enables weights and a global bias to be learned for each connection [31]. The convolution operation is mathematically defined as follows:

$$a_{ij} = \varphi\left(b_i + \sum_{k=1}^{3} w_{ik} x_{j+k-1}\right) = \varphi\left(b_i + \mathbf{w}_i^T \mathbf{x}_j\right) \qquad (1)$$

Here, a_{ij} represents the activation or output of the j-th neuron of the i-th filter in the hidden layer, φ denotes the neural activation function, b_i signifies the shared overall bias of filter i, $\mathbf{w}_i = [w_{i1} \quad w_{i2} \quad w_{i3}]^T$ is the vector comprising shared weights, and $\mathbf{x}_i = [x_j \quad x_{j+1} \quad x_{j+2}]^T$

The output produced by this layer is known as a feature map, which contains information concerning the input by filtering and learning the weighted inputs. When multiple localized features must be extracted, additional convolution kernels are employed to create more feature maps [31].

2) Pooling layer: This layer carries subsampling to simplify and summarise the attribute map. Max-pooling selects the maximum value for each kernel, reducing the size of the feature map and the computational cost while preserving the essential characteristics of the images. There are many types of pooling: Average, Max, Sum, etc.

3) Fully connected layers: Fully connected layers: After several layers of convolution, ReLU and pooling, fully connected layers link each neuron in one layer to each neuron in the next layer [11]. This structure works in the same way as the classical multilayer perceptron (MLP) neural network [17], [18]. With a softmax activation function, the latter is generally used to predict posteriori probabilities of each class.

One common issue with CNN is that it is perceived to be data-hungry [12]. Because of the vast number of learnable parameters, CNNs may require a substantial amount of data (particularly labeled images) to provide accurate predictions. Limited training data in little applications can lead to overfitting. We present techniques to tackle this problem. We will compare our results to those of the CNN model. The architecture of our CNN model is presented in the following sections.

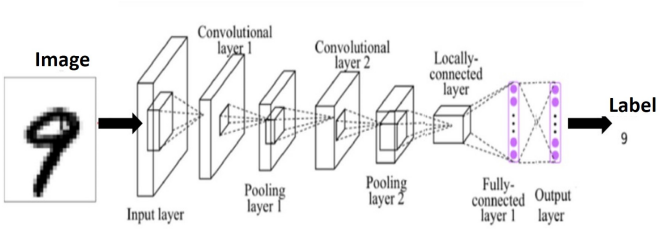

Fig. 1. CNN architecture.

B. Generative Adversarial Network (GAN)

Presented by Goodfellow et al. [19], GANs are a novel technique that work by alternating the training of two distinct neural networks: the discriminator D, which is responsible for learning the characteristics of real images in order to differentiate between "fake" and "real" images; and the generator G, which creates samples from a predetermined distribution to fool D (Fig. 2).

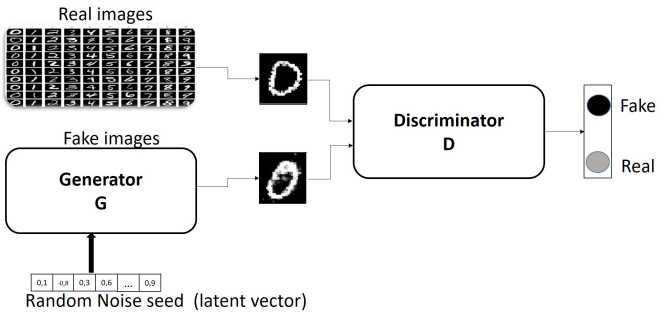

Fig. 2. GAN architecture.

The generator G receives a Gaussian random variable z as input and produces an image x as output: $G(z) = x$. The discriminator and generator typically employ CNNs, renowned for their efficiency in image identification. Throughout training, the generator and discriminator are trained in opposing directions: D's parameters are updated while G's remain unchanged, and vice versa, as outlined in algorithm 1 [20].

The discriminator's task is to distinguish between real images $x^{(1)}, \ldots, x^{(n)}$ and generated images $G(z^{(1)}), \ldots, G(z^{(n)})$ whereas the generator aims to deceive the discriminator. Let $D(x)$ be the probability that image x is real. Training the discriminator involves minimizing the binary cross-entropy loss [Eq. (2)].

$$L(D,G) = -\sum_{n=1}^{N} \log\left[D(x^{(n)})\right] + \log\left(1 - D(G(z^{(n)}))\right) \tag{2}$$

The ideal discriminator D given a fixed generator G is shown in Eq. (3).

$$D_{\text{opt}} = \text{argmin}\, L(D,G) \tag{3}$$

The ideal generator G given a fixed discriminator D is shown in Eq. (4).

$$G_{\text{opt}} = \text{argmax}\, L(D,G) = \text{argmax}\left(-\sum_{n=1}^{N} \log(1 - D(G(z^{(n)})))\right) \tag{4}$$

In practice, the loss function for G is frequently expressed by the subsequent Eq. (5):

$$G_{\text{opt}} = \text{argmax} \sum_{n=1}^{N} \log(D(G(z^{(n)}))) \tag{5}$$

Several researchers have explored developing a supervisory classification model using features from the GAN discriminator [21]. The Auxiliary-Condition GAN [22] has been the most effective method proposed for addressing the challenge of controlling generated images. In our approach, we demonstrate the way this model adapted into a supervised classification model.

Algorithm 1 MM-GAN training using minibatch stochastic gradient descent

for numberof training iteration **do**
  for k steps **do**
    - Sample a minibatch of m noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_z(z)$.
    - Sample a minibatch of m samples $\{x^{(1)}, \ldots, x^{(m)}\}$ from real data distribution p_r.
    - Update the discriminator by ascending its stochastic gradient:
    $\nabla_{OD} \frac{1}{m} \sum_{i=1}^{m} \log\left[D(x^{(i)}) + \log(1 - D(G(z^{(i)})))\right]$
  end for
  - Sample a minibatch of m noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_z(z)$.
  - Update the generator by descending its stochastic gradient:
  $\nabla_{OG} \frac{1}{m} \sum_{i=1}^{m} \log(1 - D(G(z^{(i)})))$
end for

C. Decision Boundary

A decision boundary is a fundamental concept in machine learning that delineates the input space into distinct labels. Recent research has focused on understanding neural networks through the lens of decision boundaries [23]–[25]. A decision boundary is a surface that separates data points into distinct classes. According to [25], [26], a decision boundary is defined as a region in the space where the output label of a classifier is ambiguous. Furthermore, [27], [28] note that the decision boundary can take various forms (Fig. 3), such as a hyperplane, a sphere, or a paraboloid. In higher dimensions, it can consist of multiple nonlinear hypersurfaces.

An intriguing and longstanding challenge in this field is identifying a decision boundary that elucidates the generalization capabilities of deep neural networks. Significant efforts are being made to address this issue. One popular approach involves adversarial attacks, which modify input images to influence label predictions, thereby characterizing the decision boundary of deep neural networks. This technique is often associated with Generative Adversarial Networks (GANs) [29], [30]. Several studies have leveraged this approach to investigate the decision boundaries of deep classifiers [23], [25], [26], [32].

Moreover, numerous works [33]–[37] have utilized decision boundaries to gain insights into the generalization of deep neural networks. Guan et al. [36] empirically demonstrate a negative relationship between decision boundary complexity and neural network generalization ability. This finding is further elucidated by Lei [37], who explains the inverse relationship between generalizability and decision boundary variability.

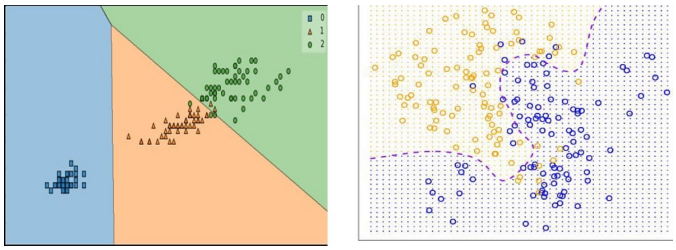

Fig. 3. Decision boundary.

Formally, the decision boundary is defined by Mickisch et al. [26] as follows:

Consider a neural network classifier $f : \mathbb{R}^n \to \mathbb{R}^c$, where n and c represent the dimensions of the input and output, respectively. For an input image $x \in \mathbb{R}^n$, the output $f(x)$ is determined by a classification decision defined as:

$$K(x) = \text{argmax}_{k=1,\ldots,c} f_k(x) \tag{6}$$

The decision boundary $D \in \mathbb{R}^n$ is defined by the formula below Eq. (7):

$$D = \left\{ x \in \mathbb{R}^n \mid \begin{array}{l} \exists k_1, k_2 = 1, \ldots, c, \\ k_1 \neq k_2,\ f_{k_1}(x) = f_{k_2}(x) = \max\limits_k f_k(x) \end{array} \right\} \tag{7}$$

D. Proposed Approach

Numerous works in the literature have identified the computational challenges associated with extracting useful features for image classification, particularly when dealing with limited labeled data. Two main approaches for training a classifier using a small number of labeled instances alongside a much larger collection of unlabeled data are semi-supervised learning and transfer learning. In this section, we introduce a semi-supervised approach (SGAN) that utilizes Generative Adversarial Networks (GANs) [38] through decision boundary analysis. This method effectively leverages both unlabeled and labeled data to enhance classifier training.

The traditional GAN discriminator is modified within the context of semi-supervised learning using GANs. This adapted discriminator is specifically designed to produce an output equal to the number of actual classes k [40]. An additional output is included, known as the $(k+1)th$ output. This extra output is utilized to identify fraudulent images generated by the GAN's generator component [39]–[41]. The $(k+1)th$ output primarily presents additional information in the form

of fake images, enabling the discriminator to classify them under the $(k+1)th$ label.

Our proposed semi-supervised learning architecture, based on Augustus Odena's model [41], employs a dual-mode training technique for the discriminator. This strategy combines supervised and unsupervised learning methods. In the first mode, the discriminator learns to predict the class labels for real images. In the second mode, the discriminator component of GANs is trained similarly to regular GANs, with the aim of distinguishing between real and generated (fake) images. Our proposed model offers a distinct advantage by merging unsupervised and supervised learning, facilitating effective control over the generated images and the extraction of key attributes for the classifier.

It is crucial to understand that the primary objective of our approach (SGAN) is to learn the supervised classifier. The architecture is shown in Fig. 4.

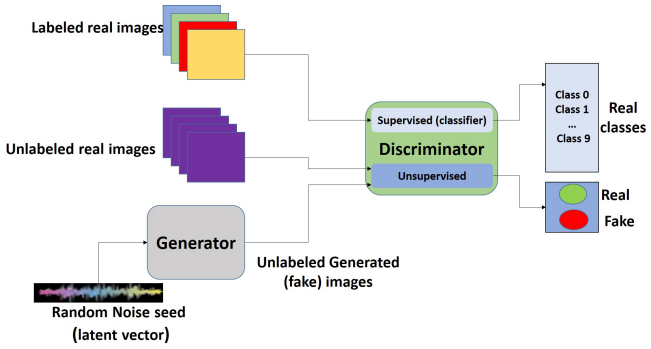

Fig. 4. Semi-supervised approach based GAN(SGAN).

The proposed architecture is composed of three primary components: a generator learned with a Gaussian distribution and examined by the discriminator; a discriminator learned with unlabeled data and influenced by a Gaussian distribution; and a supervised classifier learned with a small set of labeled data. Notably, the weights and architecture of the discriminator and supervised classifier are the same.

The generator architecture is detailed in Table I.

TABLE I. GENERATOR PARAMETERS

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 100)	0
dense_1 (Dense)	(None, 12544)	1266944
leaky_re_lu (LeakyReLU)	(None, 12544)	0
reshape (Reshape)	(None, 7, 7, 256)	0
conv2d_transpose	(None, 14, 14, 128)	295040
leaky_re_lu_1 (LeakyReLU)	(None, 14, 14, 128)	0
conv2d_transpose_1	(None, 14, 14, 64)	73792
leaky_re_lu_2 (LeakyReLU)	(None, 14, 14, 64)	0
conv2d_transpose_2	(None, 28, 28, 1)	577

We concentrate on the discriminator that is used for classifying MNIST images, comparing our approach (SGAN) to a CNN model with the same architecture as our discriminator (see Table II), using precision, loss, and decision boundary evolution using DeepFool: A function to calculate the distance to the decision boundary (Algorithm 2) [46]

TABLE II. DISCRIMINATOR GAN / CNN CLASSIFIER PARAMETERS

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	(None, 28, 28, 1)	0
conv2d_3 (Conv2D)	(None, 14, 14, 32)	320
leaky_re_lu_3 (LeakyReLU)	(None, 14, 14, 32)	0
conv2d_4 (Conv2D)	(None, 7, 7, 64)	18496
leaky_re_lu_4 (LeakyReLU)	(None, 7, 7, 64)	0
conv2d_5 (Conv2D)	(None, 4, 4, 128)	73856
leaky_re_lu_5 (LeakyReLU)	(None, 4, 4, 128)	0
flatten_1 (Flatten)	(None, 2048)	0
dropout_1 (Dropout)	(None, 2048)	0
dense_2 (Dense)	(None, 10)	20490

Algorithm 2 DeepFool: A function to calculate the distance to the decision boundary

Require: Image $image$, Model $model$, Number of classes $num\_classes = 10$, Maximum iterations $max\_iter = 50$, Small constant $\epsilon = 0.02$

Ensure: Perturbed image $perturbed\_image$, distance $distance$

Convert $image$ to tensor format.
Initialize $perturbed\_image \leftarrow image$
Initialize $w \leftarrow 0$, $r\_tot \leftarrow 0$
Get initial prediction $f\_image \leftarrow model.predict(image)$
Set $label \leftarrow \arg\max(f\_image)$
for $i = 1$ to max_iter **do**
    Convert $perturbed\_image$ to tensor format and add batch dimension.
    Compute the gradient of loss:
    $loss \leftarrow f\_perturbed[label] - \max(f\_perturbed[\text{other classes}])$
    Calculate gradient $\nabla loss$ with respect to $perturbed\_image$.
    Compute the norm of the gradient $gradients\_norm$.
    Initialize $perturbation \leftarrow \infty$, $adv\_label \leftarrow None$
    for each class k in $num\_classes$ **do**
        if $k = label$ **then**
            continue
        end if
        Compute $w_k \leftarrow \nabla loss[k] - \nabla loss[label]$
        Compute $f_k \leftarrow f\_image[k] - f\_image[label]$
        Calculate $pert\_k \leftarrow \frac{|f_k|}{gradients\_norm}$
        if $pert\_k < perturbation$ **then**
            $perturbation \leftarrow pert\_k$
            $w \leftarrow w_k$
            $adv\_label \leftarrow k$
        end if
    end for
    Compute the perturbation $r_i \leftarrow \frac{(perturbation+\epsilon)\times w}{gradients\_norm}$
    Update $r\_tot \leftarrow r\_tot + r_i$
    Update $perturbed\_image \leftarrow clip(image + r\_tot, 0, 1)$
    Get the new prediction $f\_perturbed \leftarrow model.predict(perturbed\_image)$
    $p\_label \leftarrow \arg\max(f\_perturbed)$
    if $p\_label \neq label$ **then**
        break
    end if
end for
Compute $distance \leftarrow \|r\_tot\|$
return $perturbed\_image$, $distance$

III. Results and Discussion

A. Database

The Modified National Institute of Standards and Technology (MNIST) dataset is widely considered a standard for digit recognition systems [42]. LeCun et al. [43] introduced it in 1998.MNIST contains 70,000 grayscale images at a resolution of 28 x 28 pixels. The dataset contains patterns drawn from two sources: NIST's Special Database-1 (high school student handwriting) and NIST's Special Database-3 (U.S. Census Bureau employee handwriting).The dataset is divided into two sets: a training set of 60,000 images and a test set of 10,000 images, which are properly separated so that no writer appears in both sets [42], [44]. Fig. 5 shows that handwriting styles vary significantly among writers.

Fig. 5. A sample from MNIST dataset.

B. Classification Complexity

The t-SNE method aims to illustrate high-dimensional data by mapping every data instance to a specific location in two- or three-dimensional space [45]. Fig. 6 illustrates the dataset in two dimensions, where we can observe that some sample classes are intermixed.

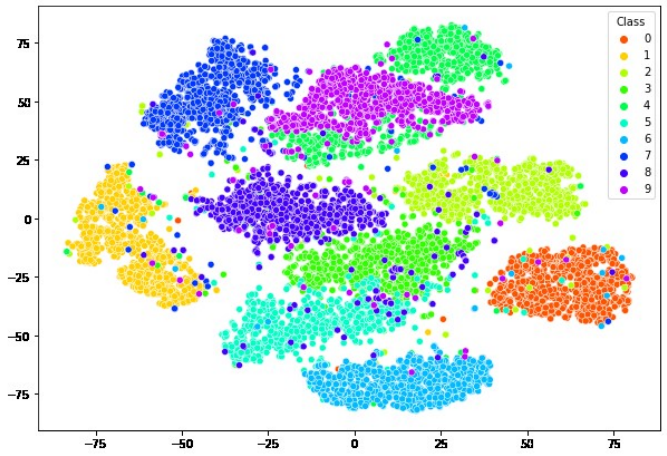

Fig. 6. TSNE visualization of MNIST dataset.

C. Results

In this section, we explore two deep learning techniques, CNN and SGAN classifiers, applied to the MNIST dataset. We specifically focus on configurations where only 100 labeled images are used for training, ensuring equal representation from each class. Notably, the CNN classifier employed in our work shares the same architecture as the SGAN model. The primary distinction lies in the training strategy: the CNN classifier is trained using supervised learning, whereas the SGAN model utilizes a semi-supervised learning. Our objective is to address the challenge of limited labeled data through the semi-supervised methodology of the SGAN.

For the testing phase, we used a dataset of 10,000 images (test set). We developed two classification models: an SGAN and a CNN classifier. The results are summarized in Table III, highlighting the accuracy and loss metrics for both models. Our experimental results indicate that the SGAN model outperforms the CNN classifier, achieving an accuracy of 84% compared to 72% for the CNN. Additionally, the SGAN model exhibited a lower loss of 0.54, while the CNN model recorded a higher loss of 0.93. These findings underscore the effectiveness of the SGAN approach compared to the traditional CNN. To further demonstrate the superiority of our approach, we analyzed a critical aspect of deep learning model evaluation: the decision boundary. Our analysis involved tracking the evolution of this boundary during training and quantifying the distance of images from it. As shown in Fig. 7 and Fig. 8, the distance from the decision boundary is significantly greater for the SGAN compared to the CNN, indicating better generalization. The incorporation of unlabeled images in training the SGAN notably enhances the performance of the MNIST image classification model, particularly when only a small proportion of labeled images are available.

TABLE III. Accuracy and Loss Classification Metrics for SGAN, the CNN Classifier, SSAE and SVAE

	CNN	SGAN
Loss	0.93	0.54
Accuracy	72%	84%

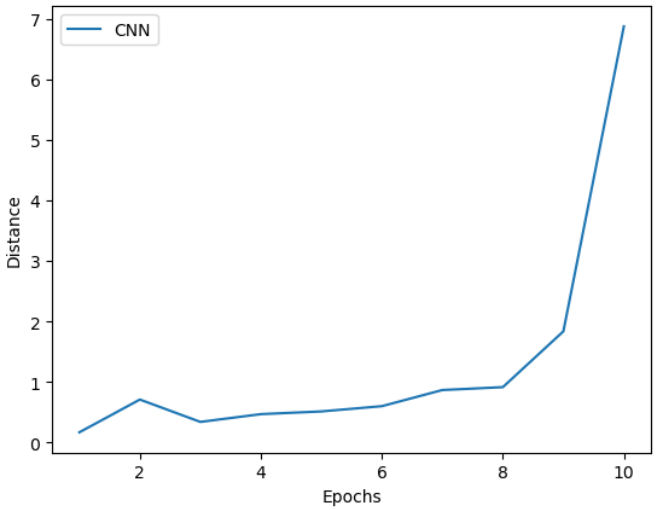

Fig. 7. Variability of the distance of CNN images to the decision boundary.

other results are presented in the appendix Section V

IV. Conclusion

We introduced a novel approach: SGAN applied to the MNIST dataset. Our experimental results highlight the superior efficiency of the SGAN models compared to the CNN model, with the SGAN achieving an accuracy of 84%. The scarcity of labeled data poses a significant challenge for image classification models; however, our proposed method effectively addresses this issue. By employing semi-supervised techniques and a novel training strategy that leverages both labeled and unlabeled images, we observed a substantial improvement in image classification performance. Notably, in terms of

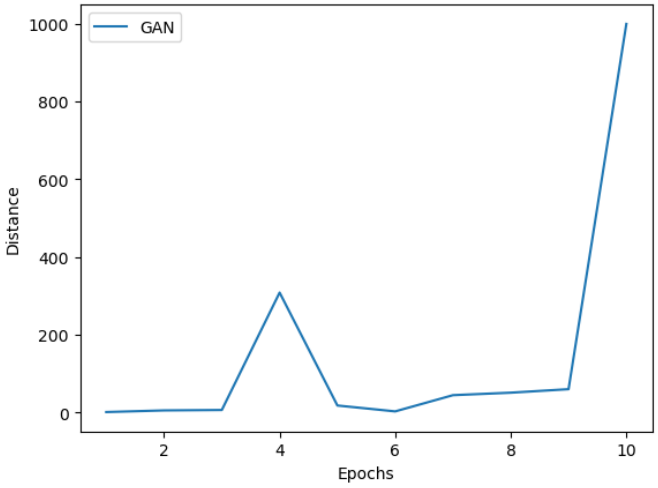

Fig. 8. Variability of the distance of SGAN images to the decision boundary.

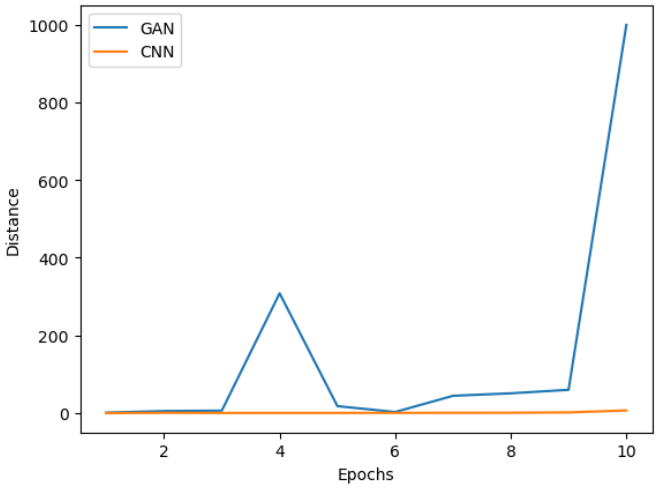

Fig. 9. Variability of the distance of CNN and SGAN images to the decision boundary.

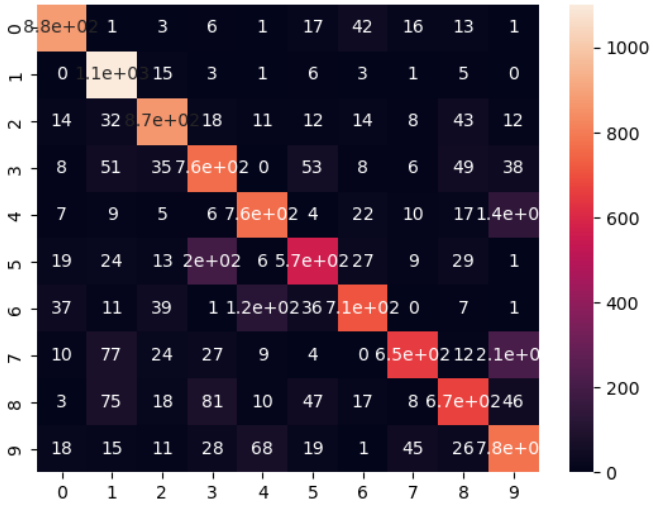

Fig. 10. The confusion matrices for the CNN modeL.

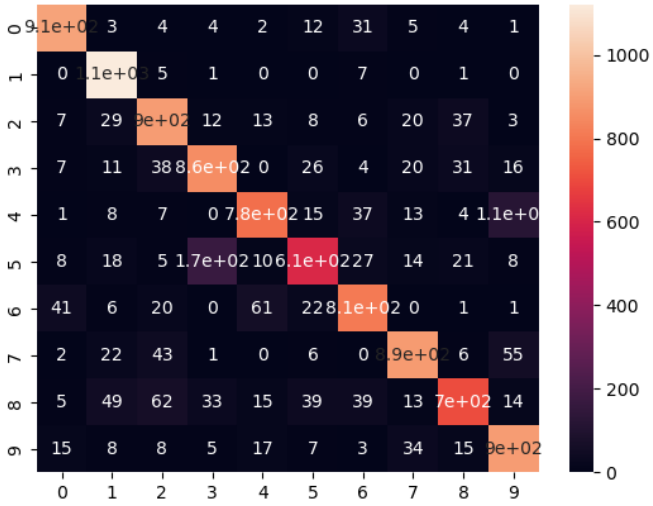

Fig. 11. The confusion matrices for the SGAN modeL.

decision boundary analysis, our models produced promising results that significantly outperform those of CNNs.

V. Appendix

In this section, we present additional results. Fig. 9 illustrates the variability of the decision boundary distance for both the CNN and SGAN models. The confusion matrix, which is a table that compares the model's predictions with the actual results, provides insight into the overall performance of the classification model. Fig. 10 and Fig. 11 show the confusion matrices for the CNN and SGAN models, respectively. Finally, we present the images generated by the generator in our SGAN approach (Fig. 12) .

Declarations

- Funding
No funding was received to assist with the preparation of this manuscript.

- Conflict of interest/Competing interests
The authors declare that they have no competing interests

- Availability of data
All data generated or analysed during this study are included in this published article

References

[1] ALZUBAIDI, Laith, ZHANG, Jinglan, HUMAIDI, Amjad J., et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Journal of big Data, 2021, vol. 8, p. 1-74.

[2] BHATTACHARYA, Sweta, SOMAYAJI, Siva Rama Krishnan, GADEKALLU, Thippa Reddy, et al. A review on deep learning for future smart cities. Internet Technology Letters, 2022, vol. 5, no 1, p. e187.

[3] WANG, Ning, WANG, Yuanyuan, et ER, Meng Joo. Review on deep learning techniques for marine object recognition: Architectures and algorithms. Control Engineering Practice, 2022, vol. 118, p. 104458.

[4] SHORTEN, Connor, KHOSHGOFTAAR, Taghi M., et FURHT, Borko. Deep Learning applications for COVID-19. Journal of big Data, 2021, vol. 8, no 1, p. 1-54.

[5] TORRES, José F., HADJOUT, Dalil, SEBAA, Abderrazak, et al. Deep learning for time series forecasting: a survey. Big Data, 2021, vol. 9, no 1, p. 3-21.

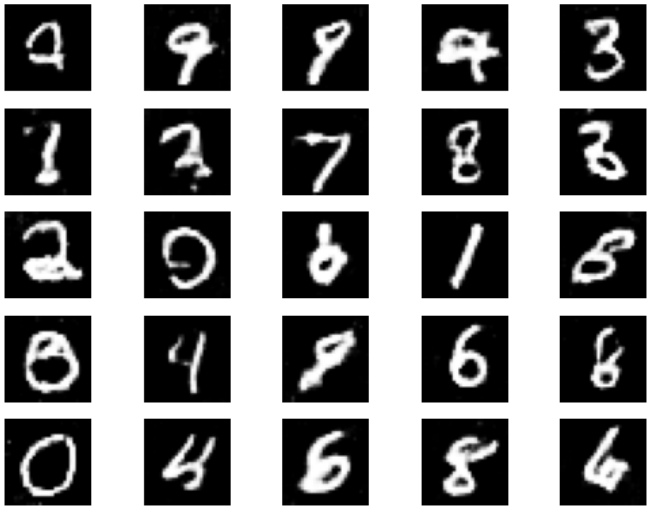

Fig. 12. Images generated by the SGAN generator.

[6] ABIDI, M. H., MOHAMMED, M. K., et ALKHALEFAH, H. Predictive Maintenance Planning for Industry 4.0 Using Machine Learning for Sustainable Manufacturing. Sustainability 2022, 14, 3387. 2022.

[7] BANSAL, Monika, KUMAR, Munish, SACHDEVA, Monika, et al. Transfer learning for image classification using VGG19: Caltech-101 image data set. Journal of ambient intelligence and humanized computing, 2023, p. 1-12.

[8] ELYAN, Eyad, VUTTIPITTAYAMONGKOL, Pattaramon, JOHNSTON, Pamela, et al. Computer vision and machine learning for medical image analysis: recent advances, challenges, and way forward. Artificial Intelligence Surgery, 2022, vol. 2, no 1, p. 24-45.

[9] ENGSTROM, Logan, TRAN, Brandon, TSIPRAS, Dimitris, et al. A rotation and a translation suffice: Fooling cnns with simple transformations. 2017.

[10] SIMONYAN, Karen et ZISSERMAN, Andrew. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[11] HABOUZ, Y. E., IGGANE, M., ES-SAADY, Y., et al. Deep neura l networks for otolith identification. Int. J. Imaging Robot, 2018, vol. 18, no 3, p. 1-10.

[12] STOCK, Michiel, NGUYEN, Bac, COURTENS, Wouter, et al. Otolith identification using a deep hierarchical classification model. Computers and Electronics in Agriculture, 2021, vol. 180, p. 105883.

[13] LECUN, Yann, BENGIO, Yoshua, et HINTON, Geoffrey. Deep learning. nature, 2015, vol. 521, no 7553, p. 436-444.

[14] GANESAN, Jothi, AZAR, Ahmad Taher, ALSENAN, Shrooq, et al. Deep learning reader for visually impaired. Electronics, 2022, vol. 11, no 20, p. 3335.

[15] ELKHOLY, Hassan Ashraf, AZAR, Ahmad Taher, MAGD, Ahmed, et al. Classifying upper limb activities using deep neural networks. In : Proceedings of the International Conference on Artificial Intelligence and Computer Vision (AICV2020). Springer International Publishing, 2020. p. 268-282.

[16] DUDEKULA, Khasim Vali, SYED, Hussain, BASHA, Mohamed Iqbal Mahaboob, et al. Convolutional neural network-based personalized program recommendation system for smart television users. Sustainability, 2023, vol. 15, no 3, p. 2206.

[17] HU, Xuefei et WENG, Qihao. Estimating impervious surfaces from medium spatial resolution imagery using the self-organizing map and multi-layer perceptron neural networks. Remote Sensing of Environment, 2009, vol. 113, no 10, p. 2089-2102.

[18] SRIVASTAVA, Nitish, HINTON, Geoffrey, KRIZHEVSKY, Alex, et al. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 2014, vol. 15, no 1, p. 1929-1958.

[19] GOODFELLOW, Ian, POUGET-ABADIE, Jean, MIRZA, Mehdi, et al. Generative adversarial nets. Advances in neural information processing systems, 2014, vol. 27.

[20] JÓNSDÓTTIR, Ingibjörg G., CAMPANA, Steven E., et MARTEINSDOTTIR, Gudrun. Otolith shape and temporal stability of spawning groups of Icelandic cod (Gadus morhua L.). ICES Journal of Marine Science, 2006, vol. 63, no 8, p. 1501-1512.

[21] RUBIN, Moran, STEIN, Omer, TURKO, Nir A., et al. TOP-GAN: Stain-free cancer cell classification using deep learning with a small training set. Medical image analysis, 2019, vol. 57, p. 176-185.

[22] ODENA, Augustus, OLAH, Christopher, et SHLENS, Jonathon. Conditional image synthesis with auxiliary classifier gans. In : International conference on machine learning. PMLR, 2017. p. 2642-2651.

[23] HE, Warren, LI, Bo, et SONG, Dawn. Decision boundary analysis of adversarial examples. In : International Conference on Learning Representations. 2018.

[24] KARIMI, Hamid et TANG, Jiliang. Decision boundary of deep neural networks: Challenges and opportunities. In : Proceedings of the 13th International Conference on Web Search and Data Mining. 2020. p. 919-920.

[25] KARIMI, Hamid, DERR, Tyler, et TANG, Jiliang. Characterizing the decision boundary of deep neural networks. arXiv preprint arXiv:1912.11460, 2019.

[26] MICKISCH, David, ASSION, Felix, GREßNER, Florens, et al. Understanding the decision boundary of deep neural networks: An empirical study. arXiv preprint arXiv:2002.01810, 2020.

[27] LI, Yu, DING, Lizhong, et GAO, Xin. On the decision boundary of deep neural networks. arXiv preprint arXiv:1808.05385, 2018.

[28] FAWZI, Alhussein, MOOSAVI-DEZFOOLI, Seyed-Mohsen, FROSSARD, Pascal, et al. Empirical study of the topology and geometry of deep networks. In : Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018. p. 3762-3770.

[29] GOODFELLOW, Ian, POUGET-ABADIE, Jean, MIRZA, Mehdi, et al. Generative adversarial nets. Advances in neural information processing systems, 2014, vol. 27.

[30] GOODFELLOW, Ian J., SHLENS, Jonathon, et SZEGEDY, Christian. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572, 2014.

[31] KHALIF, Ku Muhammad Naim Ku, CHAW SENG, Woo, GEGOV, Alexander, et al. Integrated generative adversarial networks and deep convolutional neural networks for image data classification: A case study for covid-19. Information, 2024, vol. 15, no 1, p. 58.

[32] LI, Qian, QI, Yong, HU, Qingyuan, et al. Adversarial adaptive neighborhood with feature importance-aware convex interpolation. IEEE Transactions on Information Forensics and Security, 2020, vol. 16, p. 2447-2460.

[33] HEO, Byeongho, LEE, Minsik, YUN, Sangdoo, et al. Knowledge distillation with adversarial samples supporting decision boundary. In : Proceedings of the AAAI conference on artificial intelligence. 2019. p. 3771-3778.

[34] ALFARRA, Motasem, BIBI, Adel, HAMMOUD, Hasan, et al. On the decision boundaries of neural networks: A tropical geometry perspective. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, vol. 45, no 4, p. 5027-5037.

[35] CHOI, Kanghyun, HONG, Deokki, PARK, Noseong, et al. Qimera: Data-free quantization with synthetic boundary supporting samples. Advances in Neural Information Processing Systems, 2021, vol. 34, p. 14835-14847.

[36] GUAN, Shuyue et LOEW, Murray. Analysis of generalizability of deep neural networks based on the complexity of decision boundary. In : 2020 19th IEEE international conference on machine learning and applications (ICMLA). IEEE, 2020. p. 101-106.

[37] LEI, Shiye, HE, Fengxiang, YUAN, Yancheng, et al. Understanding deep learning via decision boundary. IEEE Transactions on Neural Networks and Learning Systems, 2023.

[38] EL HABOUZ, Youssef, EL MOURABIT, Yousef, IGGANE, Mbark, et al. Efficient semi-supervised learning model for limited otolith data using generative adversarial networks. Multimedia Tools and Applications, 2024, vol. 83, no 4, p. 11909-11922.

[39] DUMOULIN, Vincent, BELGHAZI, Ishmael, POOLE, Ben, et al. Adversarially learned inference. arXiv preprint arXiv:1606.00704, 2016.

[40] KUMAR, Abhishek, SATTIGERI, Prasanna, et FLETCHER, Tom. Semi-supervised learning with gans: Manifold invariance with improved inference. Advances in neural information processing systems, 2017, vol. 30.

[41] ODENA, Augustus. Semi-supervised learning with generative adversarial networks. arXiv preprint arXiv:1606.01583, 2016.

[42] BALDOMINOS, Alejandro, SAEZ, Yago, et ISASI, Pedro. A survey of handwritten character recognition with mnist and emnist. Applied Sciences, 2019, vol. 9, no 15, p. 3169.

[43] LECUN, Yann, BOTTOU, Léon, BENGIO, Yoshua, et al. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 1998, vol. 86, no 11, p. 2278-2324.

[44] MOHAPATRA, Ramesh Kumar, MAJHI, Banshidhar, et JENA, Sanjay Kumar. Classification performance analysis of mnist dataset utilizing a multi-resolution technique. In : 2015 International Conference on Computing, Communication and Security (ICCCS). IEEE, 2015. p. 1-5.

[45] VAN DER MAATEN, Laurens et HINTON, Geoffrey. Visualizing data using t-SNE. Journal of machine learning research, 2008, vol. 9, no 11.

[46] MOOSAVI-DEZFOOLI, Seyed-Mohsen, FAWZI, Alhussein, et FROSSARD, Pascal. Deepfool: a simple and accurate method to fool deep neural networks. In : Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. p. 2574-2582.