

Retraction

Retracted: Feature Learning-Based Generative Adversarial Network Data Augmentation for Class-Based Few-Shot Learning

Mathematical Problems in Engineering

Received 18 July 2023; Accepted 18 July 2023; Published 19 July 2023

Copyright © 2023 Mathematical Problems in Engineering. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

- (1) Discrepancies in scope
- (2) Discrepancies in the description of the research reported
- (3) Discrepancies between the availability of data and the research described
- (4) Inappropriate citations
- (5) Incoherent, meaningless and/or irrelevant content included in the article
- (6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

- [1] B. Subedi, V. E. Sathishkumar, V. Maheshwari, M. S. Kumar, P. Jayagopal, and S. M. Allayear, "Feature Learning-Based Generative Adversarial Network Data Augmentation for Class-Based Few-Shot Learning," *Mathematical Problems in Engineering*, vol. 2022, Article ID 9710667, 20 pages, 2022.

Research Article

Feature Learning-Based Generative Adversarial Network Data Augmentation for Class-Based Few-Shot Learning

Bharat Subedi,¹ V. E. Sathishkumar², V. Maheshwari,³ M. Sandeep Kumar,³ Prabhju Jayagopal,³ and Shaikh Muhammad Allayear⁴

¹AgileSoDa, Seoul, Republic of Korea

²Department of Industrial Engineering, Hanyang University, 222 Wangsimni-ro, Seongdong-gu, Seoul, Republic of Korea

³School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, India

⁴Department of Multimedia and Creative Technology, Daffodil International University, Daffodil Smart City, Khagan, Ashulia, Dhaka, Bangladesh

Correspondence should be addressed to V. E. Sathishkumar; sathishkumar@kakao.com

Received 10 May 2022; Revised 21 June 2022; Accepted 23 June 2022; Published 21 July 2022

Academic Editor: Shimin Wang

Copyright © 2022 Bharat Subedi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As training deep neural networks enough requires a large amount of data, there have been a lot of studies to deal with this problem. Data augmentation techniques are basic solutions to increase training data using existing data. Geometric transformations and color space augmentations are well-known augmentation techniques, but they still require some manual work and can generate limited types of data only. Therefore, there are many interests in generative-model-based augmentation lately, which can learn the distribution of data. This study proposes a set of GAN-based data augmentation methods that can generate good quality training data. The proposed networks, f-DAGAN (data augmentation generative adversarial networks), have been motivated by the DAGAN that learns data distribution from two real data. The basic f-DAGAN uses dual discriminators handling both generated data and generated feature spaces for better learning the given data. The other versions of f-DAGANs have been proposed for generating hard or easy data that have additional dual classifiers for both generated data and feature spaces to control the generator. Hard data is useful for optimized training to increase the target performance such as classification accuracy. Easy data generation can be used especially in few-shot learning. The quality of generated data has been validated in two ways: using t-SNE visualization of generated data and classification accuracy by training with generated data using the MNIST data set. The t-SNE representations show that data generated by f-DAGAN are evenly distributed for every class better than the exiting generative model-based augmentation methods. The f-DAGAN also shows the best classification accuracy by training with generated data. The f-DAGAN version for easy and hard data generation generates data well from five-shot learning and performs well in sample data generation experiments.

1. Introduction

Machine learning (ML) is a subset of artificial intelligence (AI), which imparts the framework and the advantages to naturally gain from the ideas and information without being unequivocally customized. Deep learning depends on the assortment of ML techniques that models significant level deliberations in the data with numerous nonlinear changes. Deep learning is otherwise called deep structure learning and various leveled discoveries that comprise different layers that incorporate nonlinear preparing units with the end goal of change and

highlight extraction. A deep learning innovation takes a shot at the artificial neural system (ANNs). These ANNs continually take learning techniques, and by constantly expanding the measure of data, the proficiency in preparing procedures can be improved. The learning procedure can be the supervised or semisupervised path by utilizing unmistakable phases of reflection and complex degrees of portrayals [1].

The general focal point of deep learning is the portrayal of the independent data and speculation of the educated examples for use on data unseen. The decency of the data portrayal largely affects the presentation of data by machine

learners; an unfortunate data portrayal is probably going to decrease the exhibition of even a propelled machine learner, while a decent data portrayal can prompt superior for a moderately easier machine learner. These features include designing, which centers on building highlights and data portrayals from actual data [2], which is a significant component of deep learning. Key idea basic deep learning (DL) strategies are dispersed portrayals of the data, in which countless potential arrangements of the theoretical highlights of the information are achievable, taking into consideration a conservative portrayal for each example and prompting a more extravagant generalization. The quantity of potential designs is exponentially identified with the quantity of removed conceptual highlights. Taking note of that they watched information was created through connections of a few known/obscure variables, and along these lines, when an information design is gotten through certain setups of educated components, inconspicuous information examples can probably be depicted through new arrangements of the scholarly factors and examples [3, 4]. Compared to learning dependent on nearby generalizations, the number of examples that can be obtained utilizing an appropriated portrayal scales rapidly with the number of learned elements.

In the following sections, a brief introduction of the few-shot learning approach, data augmentation approach, and the importance of the data augmentation approach will be presented.

Current DL methods cannot quickly sum up from a couple of models. The previously mentioned effective DL applications depend on gaining from huge scope of information. Conversely, people are fit for learning new assignments quickly by using what they realized previously. Overcoming this issue among DL and people is a significant heading. It very well may be handled by DL, which is worried about the subject of how to build PC programs that naturally improve with experience [5, 6]. To gain from a pre-determined number of models with directed data, another AI worldview called few-shot learning (FSL) [7, 8] is proposed. FSL can help calm the weight of gathering huge scope-directed information. Driven by the scholastic objective for DL to move toward people and the modern interest for inexpensive learning, FSL has drawn a lot of ongoing consideration and is currently a hot research territory.

Existing information growth strategies can be separated into two general classes: conventional, white box strategy, or discovery techniques dependent on deep neural networks. The most well-known customary methodology is to perform a mix of relative picture transformation and shading alteration. Geometric mutilations are generally used to expand the number of tests for preparing the deep neural networks, to adjust the size of data sets also for their productivity improvement. The most mainstream strategies are histogram evening out, upgrading complexity or splendor, white-adjust, sharpen, and blur.

Generative adversarial network (GAN) is a generally amazing asset to perform unsupervised actual data utilizing the min-max technique [9]. GANs are seen as very helpful in a wide range of information age and control issues such as text-to-image translation, the image in a painting, and so on.

One of the significant tests to utilize DL models is any way to accumulate and clarify enough data training. Fluctuates heuristics are normally used to prevent overfitting, for example, dropout, penalizing the standard of the system parameters, or early halting of the improvement technique. Aside from the regularization strategies identified with the optimized strategy, diminishing overfitting can be accomplished with data argumentation. Another significant purpose of data argumentation is to build the size of data and sum up a model for better forecast outcomes for unseen data.

The main contributions of this work include

- (1) We examine and compare several data augmentation techniques for a few-shot image classifications using metric learning, in order to compare with different generative-based data augmentation techniques.
- (2) We designed two different types of feature learning generative models for data augmentation.
- (3) We successfully build a model that can generate realistic images even a few samples available in the training set.
- (4) This work shows the feasibility of generating synthesized training data generation using adversarial training with few training data required to achieve the performance of the analysis.

The remainder of the paper is organized as follows. Section 2 describes the related works published in the field of data augmentation and few-shot learning. Section 3 gives an overview of data augmentation. Hallucination-based few-shot learning is explained in Section 4. Proposed work is elaborated in Section 5, and augmentation considering a class is described in Section 6. The evaluation process design and the result of the one-class-based augmentation are described in Sections 7 and 8, respectively. Section 9 concludes the paper.

2. Related Works

Few-shot learning (FSL) is the method of taking care of a learning model with a limited quantity of preparing information, rather than utilizing an enormous measure of preparing information to the generalized model for inconspicuous information. This strategy is for the most part used in the field of computer vision, where utilizing an item arrangement model despite everything gives proper outcomes even without having a few preparing tests. The basic FSL situation is the place models with supervised data are hard or difficult to obtain because of security, well-being, or ethical issue. A run of the model is drug revelation, which attempts to find properties of new atoms to distinguish valuable ones as new medications [10]. FSL can lessen the information gathering exertion for information serious applications, for example, picture arrangement, picture recovery, object following, video occasion discovery, language demonstrating, and neural engineering search.

FSL is called one-shot learning. One set of one-shot learning algorithms achieves an information transfer focused on the similarities between previous and recent classes

by reuse model parameters. Classes of objects are learned first by numerous training examples, and then new classes of objects were also learned by transforming model parameters from previously learned classes or selecting appropriate classifier parameters. A further class of algorithms ensures the transfer of knowledge through the sharing of object categories or functions. In patches of already-learned classes, the machine-learning algorithm extracts “diagnostic knowledge” from shared information patches and then applies it to new class learning. One shot of prior experience in horse and cow classes, for example, may be acquired in a dog class, as dog items can have identical distinctive patches. The one-shot research focused on the similarities of new classes of objects and their previously studied ones passes qualitative awareness to a worldwide experience of the object’s environment.

Zero-shot learning expects to perceive objects whose occasions might not have been seen during preparation [11, 12]. It involves the grouping of pictures where there is no named preparing information, and a few methodologies have been proposed; each year has been expanding quickly with no specific benchmark. For a solid model, envision recognizing a class of items in photographs without ever having seen a photograph of that sort of article previously. Most zero-shot learning techniques utilize some association between accessible data and inconspicuous classes. Early works of zero-shot learning [13, 14] utilize the traits inside a two-phase way to deal with construing the name of a picture that has a place with one of the concealed classes. In the broadest sense, the features of an information picture are anticipated in the principal stage; at that point, its class mark is surmised via looking through the class that accomplishes the most comparative arrangement of features.

While a large portion of zero-shot learning strategies gets familiar with the cross-model mapping between the picture and class installing space with discriminative misfortunes, there are a couple of generative models [15, 16], which address each class as a likelihood appropriation [17].

Few-shot learning (FSL) techniques can be generally sorted into three classes: hallucination-based data argumentation, meta-learning, and metric-learning. Data augmentation is a great method to expand the measure of accessible information and accordingly valuable for few-shot learning [18–20]. A few strategies propose to gain proficiency with an information generator for example adopted on Gaussian noisy [21, 22]. Be that as it may, the age models regularly fail to meet expectations when trained on not many shot information. An option is to combine information from numerous tasks that, be that as it may, is not successful because of fluctuations of the information across undertakings [23].

Meta-Learning few-shot is based on aggregate understanding from learning numerous assignments [24, 25], while base-learning centers display the information appropriation of a solitary understanding. A best-in-class illustrative of this, in particular model-agnostic meta-learning (MAML), figures out how to scan for the ideal introduction state to quick-adjust a base-student to another assignment. Its task-agnostic property makes it conceivable to sum up the

few-shot supervised learning just as unsupervised reinforcement learning [26, 27]. In any case, in our view, there are two primary constraints of this sort of approach restricting their adequacy: (i) these techniques, as a rule, require countless comparable errands for meta-training, which is costly, and (ii) each assignment is commonly displayed by a low-complexity base learner (for example, a shallow neural system) to keep away from model overfitting, hence being not able to utilize further and all the more impressive structures [28].

The objective of metric learning is to limit intra-class varieties and maximize between-class varieties. Early works utilized Siamese engineering [29, 30] to catch the likeness between pictures. The ongoing works [31] received the deep systems as feature embedded method and utilized triplet misfortunes rather than pairwise limitations to get familiar with the measurement. These measurement-learning techniques have been generally utilized in picture recovery [32], face acknowledgment, and individual redistinguishing proof [33]. Duan et al. [34] introduced a deep adversarial metric learning (DAML) to create manufactured hard negatives from the watched negative examples, where the potential hard negatives are produced for scholarly measurement as supplements. All the more as of late, Wu et al. [35] introduced a feature embedding method dependent on the neighborhood part examination. These works show that joining a deep model with appropriate targets is successful in learning the likenesses. In contrast to these techniques, we consider utilizing triplet-like systems to improve the component separation on the concealed class pictures for few-shot learning issues [36].

Metric learning-based strategies gain proficiency with a lot of project functions (embedding functions) and measurements to quantify the similitude between the question and test pictures and group them in a feed-forward way. Snell et al. [36] broadened the coordinating system by utilizing the Euclidean separation rather than the cosine separation and building a model portrayal of each class for a couple of shot learning situations, to be a specific prototypical system. Sung et al. [37] contended that the inserting space ought to be characterized by a nonlinear classifier and planned the connection module to get familiar with the separation between the feature embedding of support images and query images shown in Figure 1. The key distinction among metric-learning-based techniques lies in the way they get familiar with the measurement. Vinyals et al. [19] planned to start to finish trainable k-nearest neighbors utilizing the cosine separation on the picked-up inserting highlight, to be a specific coordinating system. Of late, Mehrotra and Dukkipati [21] prepared a deep leftover system along with a generative model to rough the expressive pairwise closeness between tests. This network is trained to learn relations between features of support and query images. The connection categories broadens the coordinating system and prototypical system by including a learnable nonlinear comparator. Jin et al. utilized several deep learning models for predicting the crack width of Longyangxia Dam, and the importance of influencing factors in cracks is analyzed [38]. Cen et al. [39] utilized recent

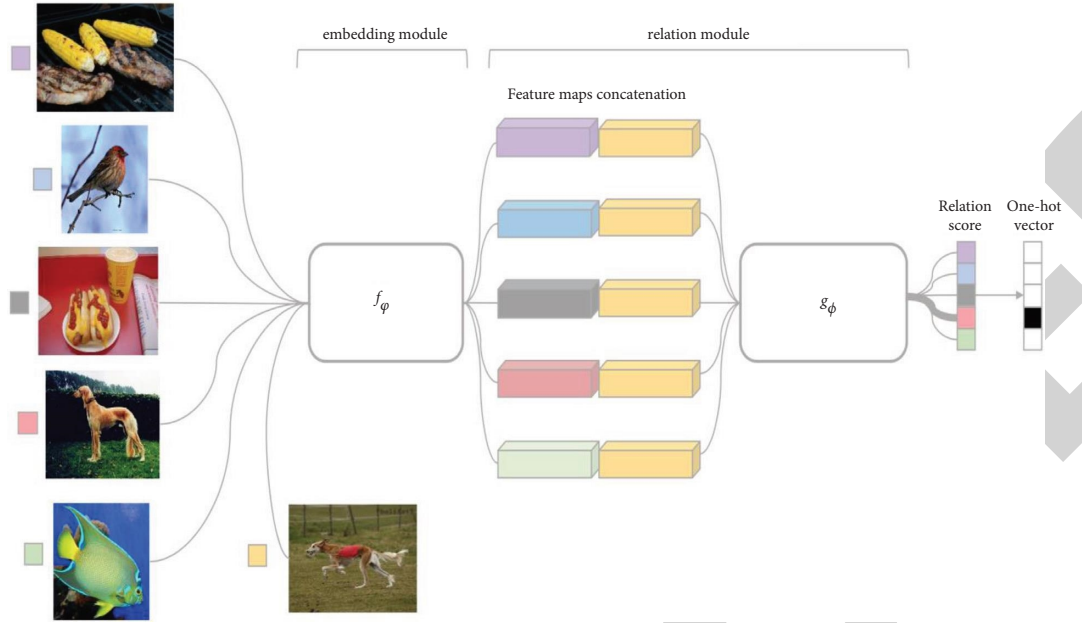


FIGURE 1: Learn to compare relation networks for few-shot learning.

graph neural networks for few-shot learning to represent fully connected graph samples of interest. Cao et al. proposed a BERT-based deep spatial-temporal network for taxi demand prediction by modeling complex spatial-temporal relations using global and local features that are heterogeneous [40]. Pu et al. used a convolutional neural network and recurrent neural network to fit the motion representation and spatial sequence in a video stream to improve the accuracy of fetal ultrasound standard plane recognition [41]. Several big data service architectures were discussed by Wang et al. [42]. Li et al. proposed a data-driven adversarial capsule network for regional traffic flow prediction with highly challenging data sets [43]. Blockchain-based technologies also play a vital role in several applications [44].

3. Data Augmentation

Argumentation of data involves a number of techniques that increase the size and consistency of training data sets in order to create stronger deep learning models. The augmentation algorithms addressed in this research includes mathematical augmentation technique and generative adversarial networks (GAN) based technique.

3.1. Mathematical Augmentation. An exceptionally conventional and acknowledged current practice for augmenting picture data is to perform geometric and color augmentations, for example, mirroring the picture, flipping, revolution, translation, noise injection, cropping, translating the picture, and changing the color palette of the picture [45]. The entirety of the transformation is the relative transformation of the first picture that takes the structure as follows:

$$y = wx + b. \quad (1)$$

The well-being of rotate argumentation is vigorously dictated by the rotate degree parameter. Moving pictures left, right, up, or down can be an extremely valuable chance to keep away from positional predisposition in the data. Another significant numerical argumentation technique is a noisy infusion, which comprises infusing a network of irregular qualities normally drawn from Gaussian dissemination. A noisy infusion of pictures can help CNNs learn progressively strong highlights. Picture data is encoded into pixel esteems for individual RGB color esteem. Lighting bias is among most of the time happening difficulties to picture recognition issues. A handy solution to excessively splendid or dim pictures is to circle through the pictures and decrease or increment the pixel esteems by a constant value can assist with learning the high dimensional feature of pictures. The mathematical augmentation technique is not suitable for all types of data set illustrated in Figure 2 in numeric data if rotation degree increases the label of data that is no longer preserved.

3.2. Generative Model-Based Augmentation. Later and additionally, an exciting technique for data augmentation is generative demonstrating. In Figure 3, two neural networks are trained opposite one another in a generative adversarial network (GAN). The generator G takes as input a noise vector z and outputs an image $X_{fake} = G(z)$. The discriminator D receives a training image or synthesized image as an input from the generator and outputs a distribution of probabilities $P(S|X) = D(X)$ over potential sources of image data. The discriminator is trained to optimize the log-likelihood of the source as follows:

$$L = E[\log P(S = real|X_{real})] + E[\log P(S = fake|X_{fake})]. \quad (2)$$



FIGURE 2: Rotation and flipping.

The generator has the task of generating convincing fake data from random noise. The discriminator gets as input either fake or real data and has to determine whether its input is real or fake.

Generative displaying refers to the act of making artificial cases from a data set with the end goal that they hold comparative qualities to the first set. The standards of adversarial training prompted an extremely intriguing and hugely famous generative system known as GANs. GAN is an approach to open extra data from a data set. GANs are by all accounts not the only generative displaying procedure that exists; anyway, they are drastically driving the path in calculation speed and nature of results. The impressive presentation of GANs has brought about expanded consideration on how they can be applied to the undertaking of data argumentation. These systems can create new training data for those outcomes in better performing order models.

Another valuable system for generative displaying worth referencing is variational autoencoder (VAE), which is described in Figure 4. The GAN system can be stretched out to improve the nature of tests delivered with variational autoencoders. Variational autoencoder gains proficiency with a low-dimensional portrayal of data focuses. An auto encoder organized is a couple of two associated systems, an encoder and a decoder. An encoder arranged takes in an independent variable and changes over it into a littler, thick portrayal, which the decoder system can use to change over it back to the actual independent variable. Variational autoencoders (VAEs) have a very simple property that isolates them from vanilla autoencoders, and that property makes them so useful for generative demos.

The model contains an encoder function $g(\cdot)$ parameterized by Φ and a decoder function $f(\cdot)$ parameterized by θ . The encoding for input x on the bottleneck layer is z , and the data restored is given in

$$x' = f_{\theta}(g_{\Phi}(x)). \quad (3)$$

4. Hallucination-Based Few-Shot Learning

Hallucination-based learning is to straightforwardly manage data inadequacy by figuring out how to enlarge like humans imagination illustrated in Figure 5. This class of technique takes in a generator from data in the base classes and utilizes the educated generator to hallucinate new novel class data for data argumentation. These generators either move fluctuation in base class data to novel classes since hallucination-based techniques frequently work with other few-shot strategies together (e.g., use hallucination-based and metric learning-based techniques together) and lead to entangled

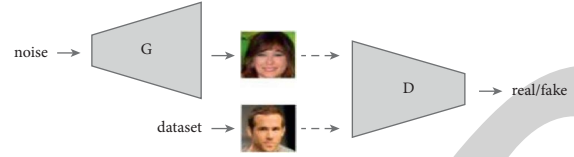


FIGURE 3: A simple graphical representation of the GAN setting.

correlation. Numerous conventional meta-learning strategies treat pictures as black boxes, disregarding the structure of the visual world. As humans, our insight into the class's diverse variety of articles may permit us to imagine what a novel item may resemble in other posture or environmental factors. On the off chance that machine vision could do such hallucinated samples, at that point, the fantasized models could be utilized as extra training data to manufacture better classifiers. Building models that can perform hallucination is hard. For general pictures, while extensive advancement has been made as of late in creating sensible examples, most current generative demonstrating approaches experience the ill effects of the issue of mode breakdown; they are just ready to catch a few methods of the data.

The key insight hallucination technique is the hallucination model that is valuable for learning classifiers. For expanding the classification, the accuracy utilizing daydream models a model that needs to map genuine guides to hallucination models. In the hallucination approach, training is first taken care of by the hallucinator; it delivers an extended preparation set, which is then utilized by the student. Utilizing meta-learning out how to train the hallucinator and the classification has two advantages. To start with, the hallucinator is legitimately prepared to deliver the sorts of fantasies that are valuable for class differentiation, evacuating the need to accurately tune authenticity or assorted variety or the correct methods of variety to hallucinate. Second, the classification technique is trained mutually with the hallucinator, which empowers it to consider any mistakes in the hallucination. On the other hand, the hallucinator can spend its ability to smother the blunders, which perplex the classification technique.

The contingent generative model incorporates highlights of unseen classes F-CLSWGAN by optimizing the Wasserstein separation regularized by a classification misfortune demonstration in Figures 6 and 7. F-CLSWGAN that produces features includes rather than pictures and is trained with a novel misfortune improving over option GAN-models. The principle key of feature-based classification is the capacity to create semantically rich CNN feature disseminations molded on class explicit semantic vector, for example, properties, without access to any pictures of that class. This reduces the irregularity among seen and unseen classes, as there is no restriction on the quantity of engineered CNN features that the model can produce.

5. Proposed Method

We proposed a new generative adversarial network for image data augmentation by conducting various feature vectors-based approaches. Furthermore, we interpreted data augmentation for class-based few-shot learning. Finally, we inform the application areas of the data augmentation.

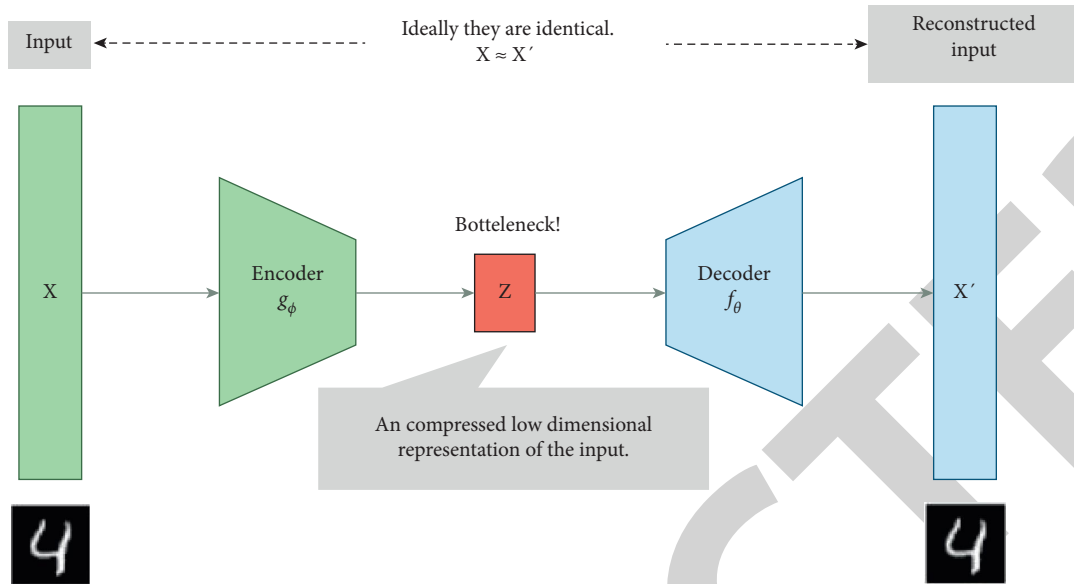


FIGURE 4: Illustration of autoencoder model architecture.



FIGURE 5: Humans are good at imagination.

5.1. The Proposed Approach. The proposed feature learning-based data augmentation generative adversarial network is given in Figures 8 and 9. In two different ways, we tried to design networks and present the output result of networks [1]. The overall flow of the proposed system is shown in Figure 10.

6. Data Augmentation for Class-Based Few-Shot Learning

The general idea of class-based data augmentation is to increase the number of data by changing data slightly to be different from the original data in a few-shot approach, but the data still can be recognized by humans. The generated

data involved the same training classes are identical to the original class. Class-based data augmentation randomly interchanged regions between various images of the same class for improving the generalization of feature distribution. To use GAN for class-based data augmentation, we design our generative network that can extract features from random Gaussian noise, which is an input of the generator network and concatenation of those features with real image features generated by CNN, which is the input for the discriminator.

6.1. Augmentation considering a Class. To achieve our goal, we purposed the f-DAGAN network for a single class

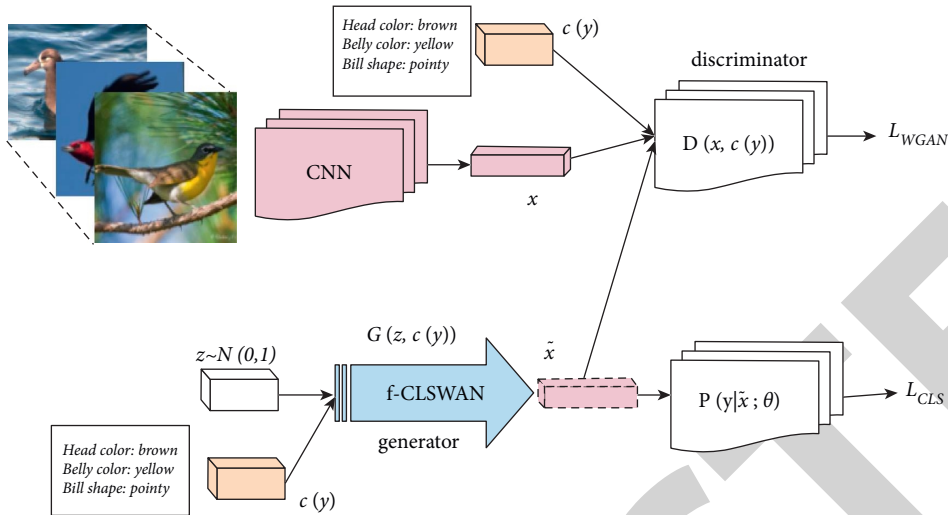


FIGURE 6: F-CLSWGAN.

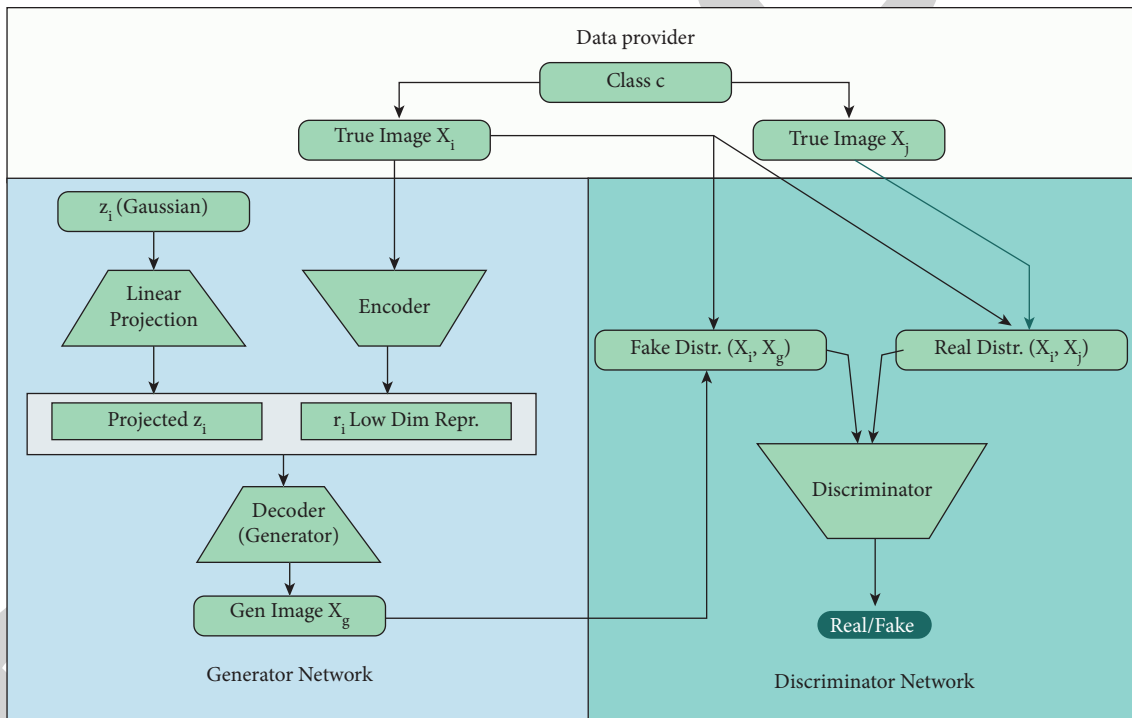


FIGURE 7: AGAN architecture [46].

augmentation and the h-DAGAN network for hard example generation. The data augmentation for both networks can be learned using an adversarial approach. Consider a source image class consisting of data $D = \{x_1, x_2, \dots, x_n\}$. Our networks take some input data point x_i and a second data point from the same class x_j .

The main idea of our first purposed architecture f-DAGAN for a single class is that we combined the generator feature and CNN feature of a real image to create realistic images. This can be done by concatenating both image features and random noise features along the channel axis. For example, a given image of dimensions $[W \times H \times$

$C]$ with its corresponding generator feature of dimensions $[W \times H \times C]$ results in a feature with dimensions of $[W \times H \times 2C]$. When training the GAN, the generator is now modified to generate a feature vector, instead of just an image. This change, in its most trivial form, can be achieved by simply modifying the convolutional layer in the generator, such that the number of channel outputs is equal to the number of channels of the required CNN feature of an input image.

For the discriminator network, we used two discriminators. One discriminator is used to discriminate between real and fake features and another discriminator to

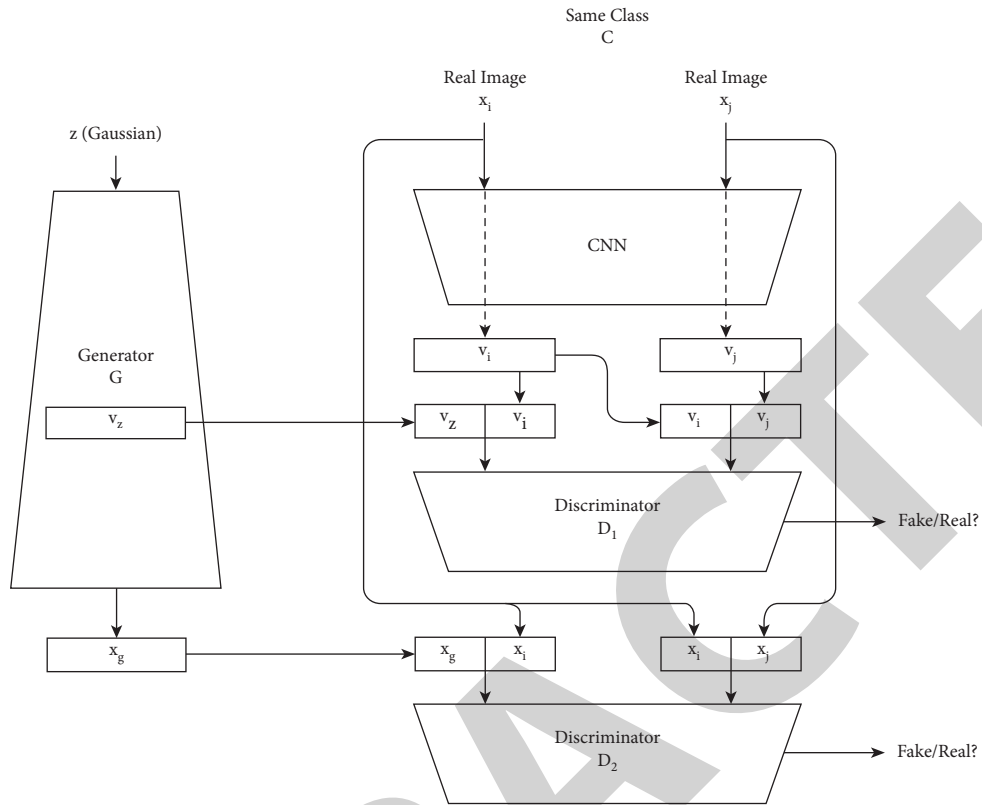


FIGURE 8: f-DAGAN structure.

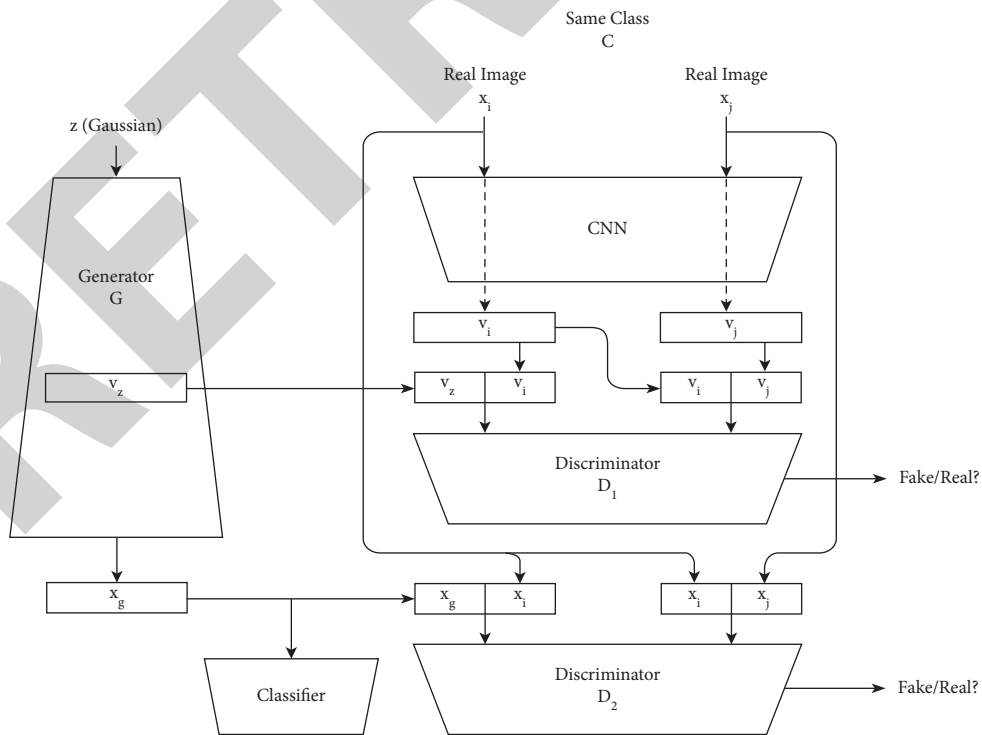


FIGURE 9: DAGAN few samples training.

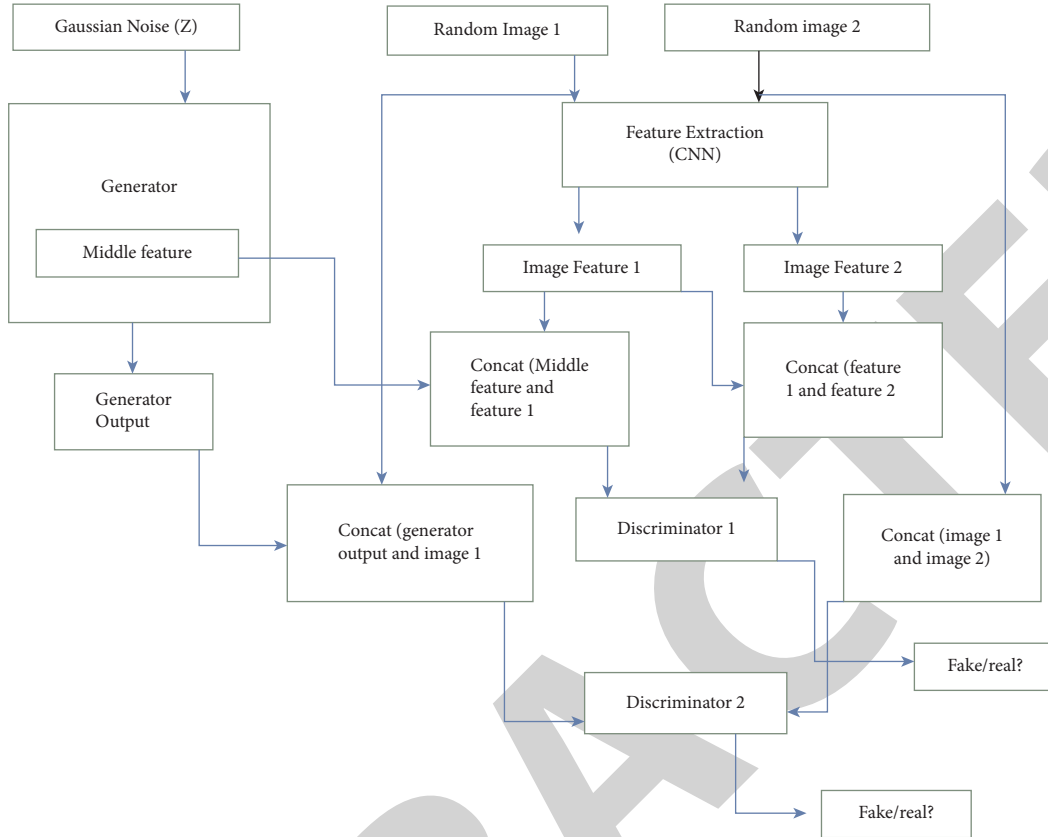


FIGURE 10: System workflow.

discriminate between real and generator images. At first, the feature discriminator network now takes a concatenated feature of the generator network and real image x_i feature as input, and its goal is to correctly decide if any given feature is real or synthetic. Second, the input of the feature discriminator network is the concatenated feature of pairs of real images x_i and x_j . At first, the image discriminator takes concatenated fake image generated by the generator and real image x_i as input, and its main goal is to correctly decide if any given image is real or synthetic. Second, the input of the discriminator network is the concatenation of pairs of real images x_i and x_j . To calculate the final loss of the discriminator, we used the sum of feature discriminator network loss and image discriminator loss.

Our second purposed architecture h-DAGAN is an example for augmentation. Classifiers are used for calculating feature and image-based easiness (hardness) of generating samples. This can be done by implementing classifiers for both fake image x_g generated by the generator and latent vector features generated by the generator network. We calculate classification loss in two ways, one for feature classification and another for fake image classification using binary cross-entropy loss. The final classification loss is the combination of feature and fake image loss. In the second step, we concatenate the concatenated feature from the generator network with a real image x_i feature from CNN, which is one of the inputs of the feature discriminator network. Another input of the feature

discriminator network is the concatenated pairs of real images x_i and x_j features from CNN. To calculate feature discriminator network loss, we utilized fake feature logits and real feature logits.

For image discriminator networks, we utilized pairs of real images x_i and x_j and images generated by the generator network. The first input of the image discriminator network is the concatenation of randomly selected one real image x_i and a fake image from the generator network. The second input of the image discriminator network is the concatenation of randomly selected real image pairs x_i and x_j . We calculate image discriminator loss by using combined fake and real image logits. The total discriminator loss is the addition of feature discriminator loss and image discriminator loss with the subtraction of total classification loss. Classifier C_1 is used for classifying fake image feature vectors, and classifier C_2 is used for classifying fake images. Probabilities of the target class in C_1 and C_2 are used for class loss CL_1 and CL_2 using binary cross-entropy loss. Total classifier loss Cl is calculated as follows:

$$CL = \alpha CL_1 + \beta CL_2. \quad (4)$$

The final generator loss is the sum of fake feature logits loss and fake image logits loss. Our second GAN architecture that we use to generate the image is illustrated in Figure 10. In both networks, every generated sample has a corresponding class pair of two images, $x_i \sim p_{x_j}$ in addition to the noise z . G uses both to generate images $X_{fake} =$

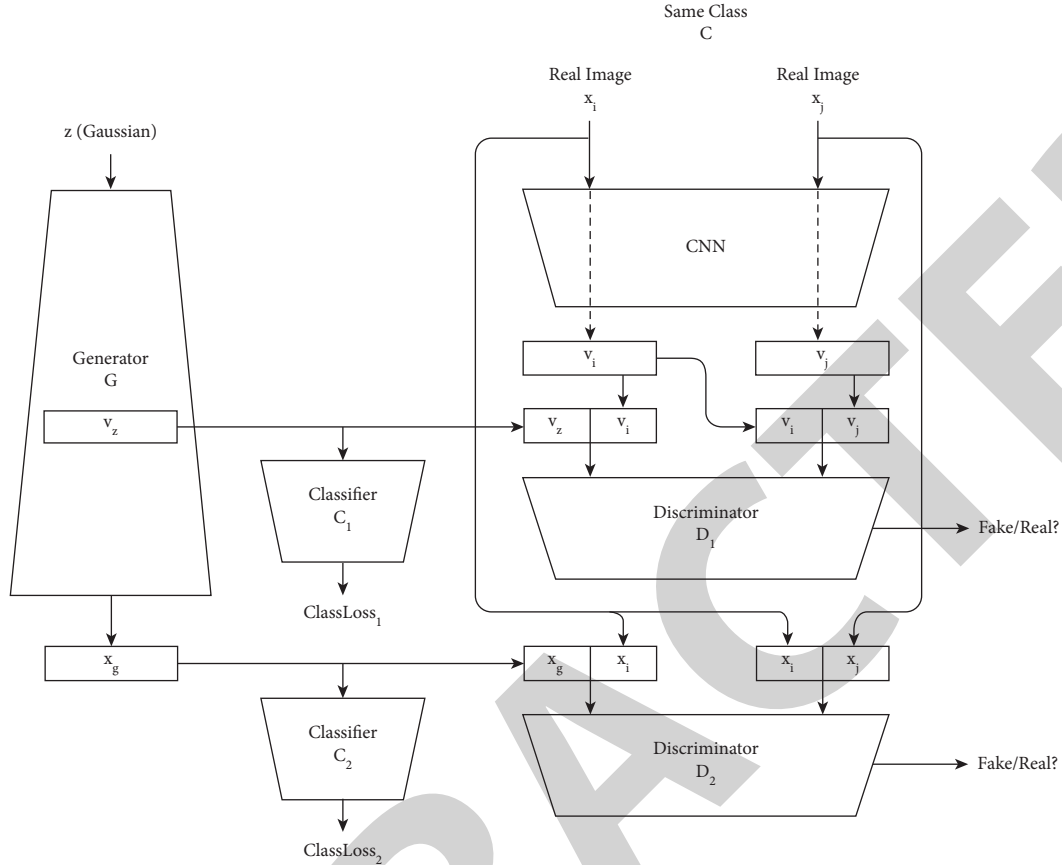


FIGURE 11: h-DAGAN hard example generation network.

$G((x_i, x_j), z)$. The discriminator gives both a probability distribution over sources x_i and x_j as follows:

$$P(x_i|X), P(x_j|X) = D(X). \quad (5)$$

GAN is studied as a minimax game and uses the alternating gradient descent on the cost function J to optimize the discriminator D and generator G . The objective function is defined as follows:

$$J^{(D)}(\Phi, \theta) = -E_{x \sim p_{real}} \log D_\theta(x) - E_z \log(1 - D_\theta(G_\Phi(z))), \quad (6)$$

$$J^{(G)}(\Phi, \theta) := -J^{(D)}(\Phi, \theta). \quad (7)$$

The complete game can be specified as follows:

$$\min_{\Phi} \max_{\theta} \{J(\Phi, \theta) = E_{x \sim p_{real}} \log D_\theta(x) + E_z \log(1 - D_\theta(G_\Phi(z)))\}. \quad (8)$$

In f-DAGAN, we want to find the equilibrium where the discriminator θ maximizes J and the generator Φ minimizes it. f-DAGAN learns a representation for z that is independent of different source images. Structurally, this model is not tremendously different from many existing few-shot models. The final discriminator loss of h-DAGAN is the difference between total discriminator loss and classifier loss.

6.2. Network Structure. During this research work, we designed and implemented feature generative networks. Our feature vector-based data augmentation generative adversarial network (f-DAGAN) as shown in Figure 11 and feature vector-based hard data augmentation generative adversarial network (h-DAGAN) are implemented based on feature space learning. We designed f-DAGAN for feature vector-based data augmentation and h-DAGAN for hard example generation. We already discussed our f-DAGAN network. The main objective of our h-DAGAN network is to create hard sample data that can help generalize the classifier and increase the accuracy of the network. Furthermore, the classifier becomes more robust when we trained on hard example data. For a training network with a few examples, we design another network. The main objective to add a classifier is to control the generator during the training network with a few examples.

6.3. Training Process and Implementation Details. Our single-class GAN was trained on the MNIST data set using ResNet50 architecture. During the training phase, there are three parts to the network. The CNN network takes in two images from the same class as the input image and returns a feature of input images. The concatenate CNN features of two images are then passed into the feature discriminator network. The generator network takes a random noise vector and generates a feature of a random vector and a fake image.

The x_i image feature concatenates with the generator generate feature also passed into the feature discriminator network. On the other side, the concatenated image of the generator image and real image x_i is passed into the image discriminator network, and concatenated image of real x_i and x_j images is passed into the image discriminator network. In each training cycle, a randomly selected sample from the source was provided for each real example.

For few-shot training, we divided the MNIST data set into small sets of data. To create 5 images sample set of the data set, we randomly selected 50 images from each class; for 10 images sample set of the data set, we randomly selected 100 images from each class, and similarly for 100 and 1,000 images sample set. We trained our f-DAGAN and h-DAGAN networks using a generator learning rate of 0.0005, and the discriminator learning rate is 0.002 with Adam optimizer parameters of $\beta_1 = 0.2$ and $\beta_2 = 0.9$. The generator has a total of 3 ResNet blocks, each block having 4 convolutional layers (ReLU activations and batch normalization) followed by one downscaling or upscaling layer. Downscaling layers were convolutions with stride 2, followed by ReLU and batch normalization. Upscaling layers were stride 1/2 replicators, followed by a convolution, ReLU, and batch normalization. Feature generated by the first ResNet block is followed by the attention block. During h-DAGAN, we used $\alpha = 0.1$ $\beta = 0.5$ for controlling loss of classifiers.

The feature generative network has a total of 2 ResNet blocks; each block consists of 4 convolutional layers with the ReLU activation and batch normalization that is followed by one downscaling layer. Feature discriminator network has a total of 3 ResNet blocks, having 4 convolution layers with ReLU activation function and batch normalization layers followed by 1 downscaling layer and dense layer. Downscaling layers were convolutions with stride 2. The image discriminator network consists of 4 ResNet blocks followed by a downscaling layer. Also, each block of ResNet had skip connections. For training and validation, we used an AMD server with 1920X CPU and NVIDIA RTX 1080ti GPU. As the deep learning framework, Python 3.7 and the GPU version of TensorFlow 2.3 were used. The configuration of the h-DAGAN network is the same as f-DAGAN; only the difference is the addition of feature and fake image classifier shown in Figure 10. Feature and image classifier network consist of two convolution blocks; each convolution block contains a 64-filter 3×3 convolution, a batch normalization, 2×2 max polling, ReLU nonlinearity layer, and fully connected layer with Sigmoid layer.

7. Evaluation

In this section, we report a series of experiments conducted on a different set of MNIST data sets, and the results of these experiments are followed by a discussion of the findings in this research work. A performance comparison of the different network architectures introduced in the previous sections (DAGAN, VAE, f-DAGAN, and C-GAN) is also presented. As a performance evaluation metric, classification accuracy is primarily used. For a detailed investigation of the



FIGURE 12: MNIST data set samples visualization.

classification accuracy of different network-generated data sets, we used the ResNet50 network.

7.1. Evaluation Process Design. It is difficult to assess the quality of data generated by GANs. This also causes it to be challenging to accurately compare the quality of data produced by different GAN architectures, algorithms, and hyperparameter settings. One way to measure the performance of generative models is an evaluation by humans. However, next to being time-consuming and expensive, this method also varies under evaluation conditions. Specifically, the evaluation setup and motivation of the annotators affect the scoring. Furthermore, when annotators are given feedback, they learn from their mistakes and make fewer errors. The (part of the) output of the discriminator that indicates whether the generated data is regarded as real could be used to monitor the convergence of GANs. However, for any specific discriminator, this output heavily depends on the generator that it is trained with. Therefore, the discriminator output cannot be used trivially to quantitatively evaluate the quality of the generated data. To overcome the problem of generated data evaluation, we used two ways that were employed to measure visual quality and data generation diversity. The classification accuracy measured how generated data performed classification on the original MNIST data set. The t-SNE visualization creates a probability distribution using the Gaussian distribution that defines the relationships between the data points in high-dimensional space [47].

7.1.1. Purposes and Performance Metrics. The use of these as feature extractors on labeled data sets is one common learning technique for evaluating the quality of unsupervised representative learning algorithms and for assessing the performance of linear model models on generated images. To evaluate for instance the consistency of the GAN model representations trained the GAN model on the MNIST data set and generate synthesized images, then CNN is used to classify them. If the CNN classifier performs well on the original data set, this indicates that the GAN synthesized images are accurate and sufficient to be informative about object class.

In order to test GANs, Ye et al. [48] suggested an analytical metric known as the GAN Consistency Index. Firstly, a generator G is trained on a labeled real data set with N classes. Secondly, a classifier C_{real} is trained on the real data set. A second classifier, called the GAN-induced

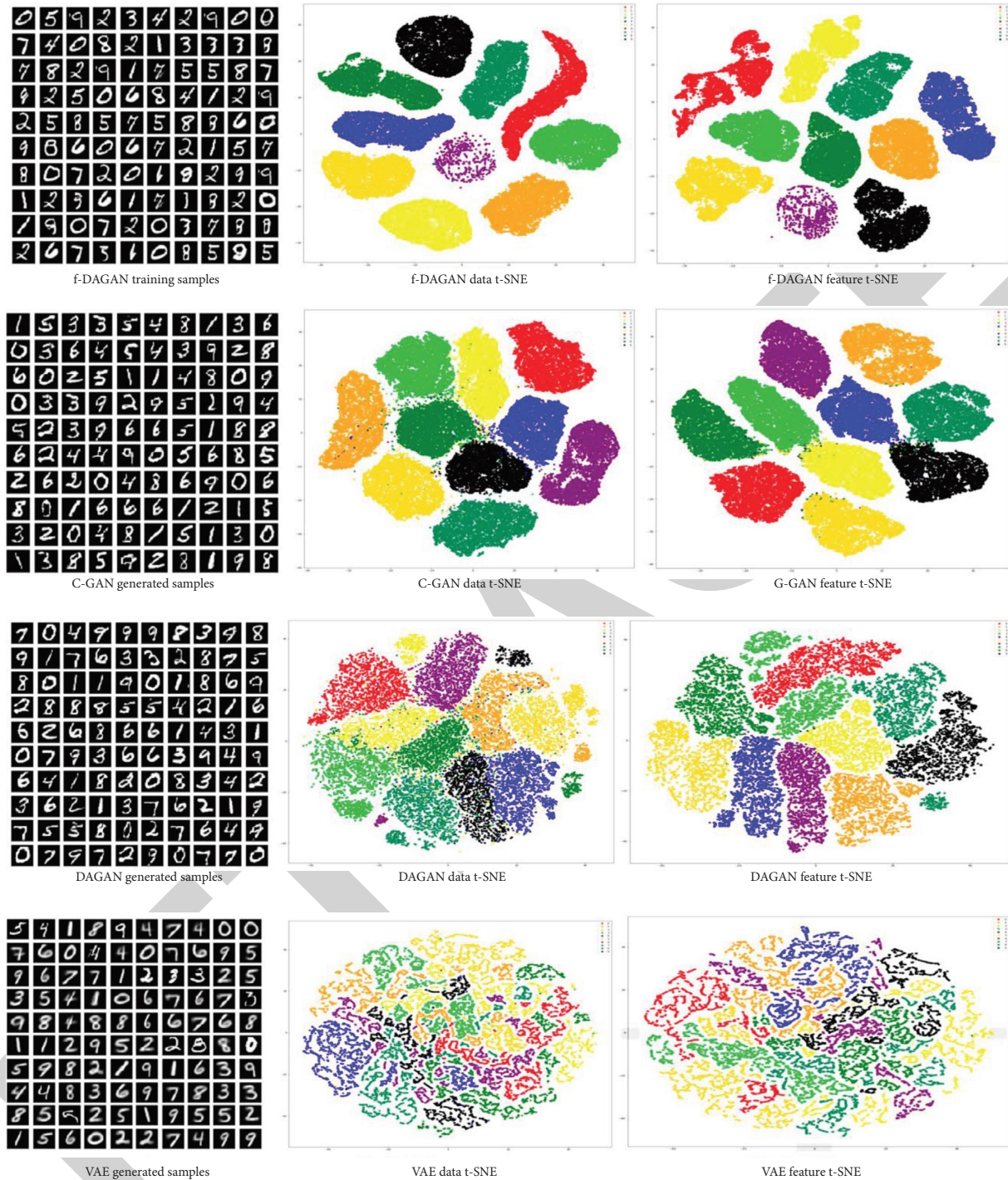


FIGURE 13: t-SNE for 5 samples data and feature spaces distribution.

classifier C_{GAN} , is trained on the generated data [49]. Finally, the GQI is defined as the ratio of the accuracies of the two classifiers. The formula for calculating GQI is as follows:

$$GQI = \frac{ACC(C_{GAN})}{ACC(C_{real})} \times 100. \quad (9)$$

GQI is an integer between 0 and 100. In higher GQI, the GAN distribution correlates best with the actual data distribution.

7.1.2. Data Sets. The MNIST data set is the application used for the analysis studies. A sample of the MNIST data set is shown in Figure 12. This data set consists of black and white images of handwritten instances of the digits 0–9 having class labels of the corresponding integers. The data set comprises a training set of 60,000 images and a test set of 10,000 images. The digits in the training and test sets were written by disjoint sets of writers. The size of the MNIST images is 28×28 pixels. Figure 11 shows a sample of the

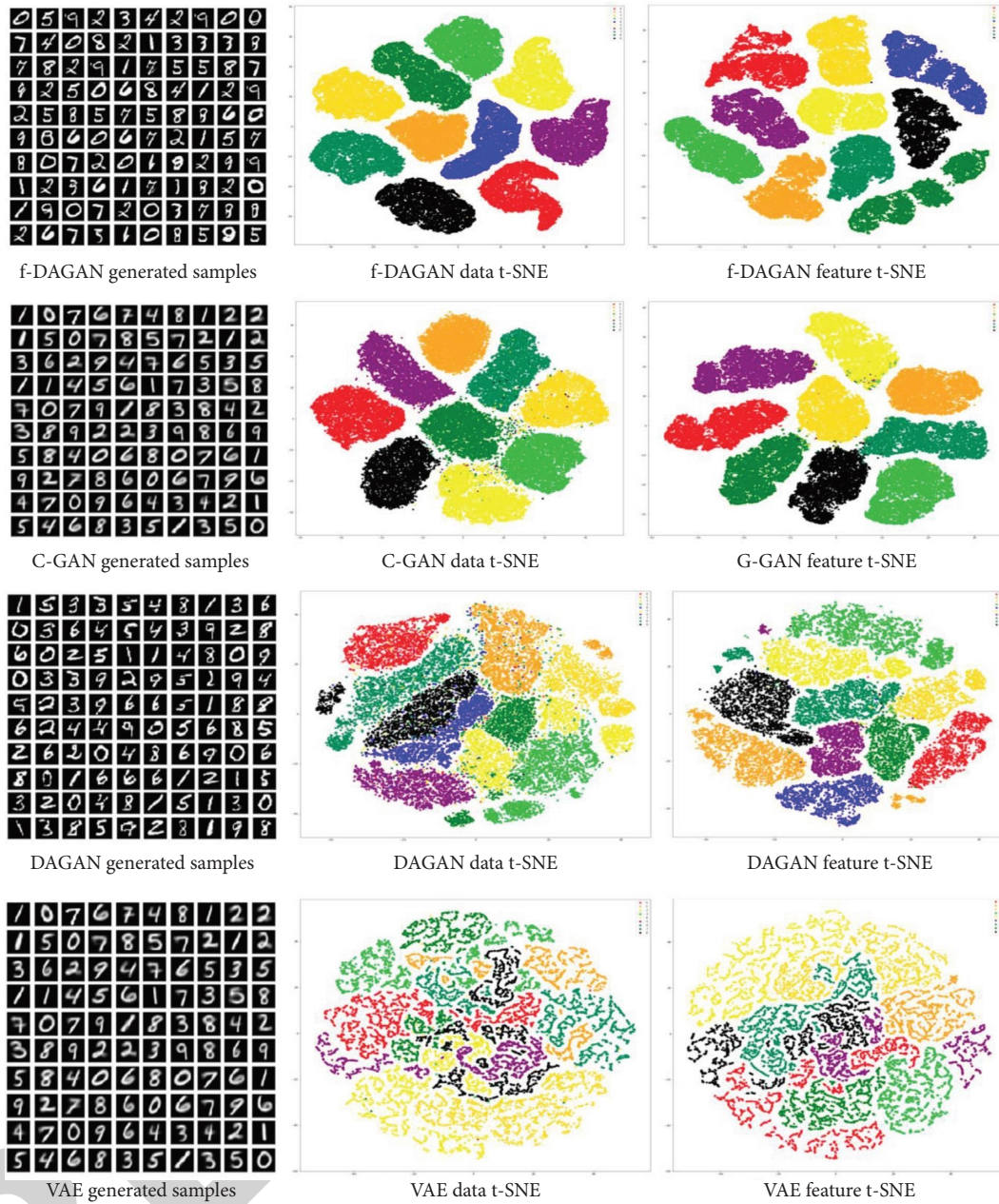


FIGURE 14: t-SNE for 10 samples data and feature spaces distribution.

MNIST data set. For the few-shot approach of training and generating synthesized images, we used a different version of the MNIST data set by splitting the training set of the MNIST data set. We split the MNIST data set into two categories:

- (a) Randomly selected 5 images from each class
- (b) Randomly selected 10 images from each class
- (c) Randomly selected 100 images from each class
- (d) Randomly selected 1,000 images from each class

The main purpose of splitting the original data set into a subset of a data set is to identify how our GAN can generalize data consisting of unseen classes.

7.1.3. References Network for Comparisons. We wanted to demonstrate that f-DAGAN could be used for data augmentation for few-shot learning, and hard example can help increase the performance of the network, as comparison of our network with other popular data augmentation architectures like DAGAN, C-CAN, and VAE. We trained each architecture until convergence as deemed their respective implementation on the MNIST data set. Then we sample 6,000 images each class uniformly at random from each generator to use as our generated set.

Here, we demonstrate that our GAN best approximates the true distribution, while DAGAN performs slightly worse. The worst performing model is the VAE, as expected.

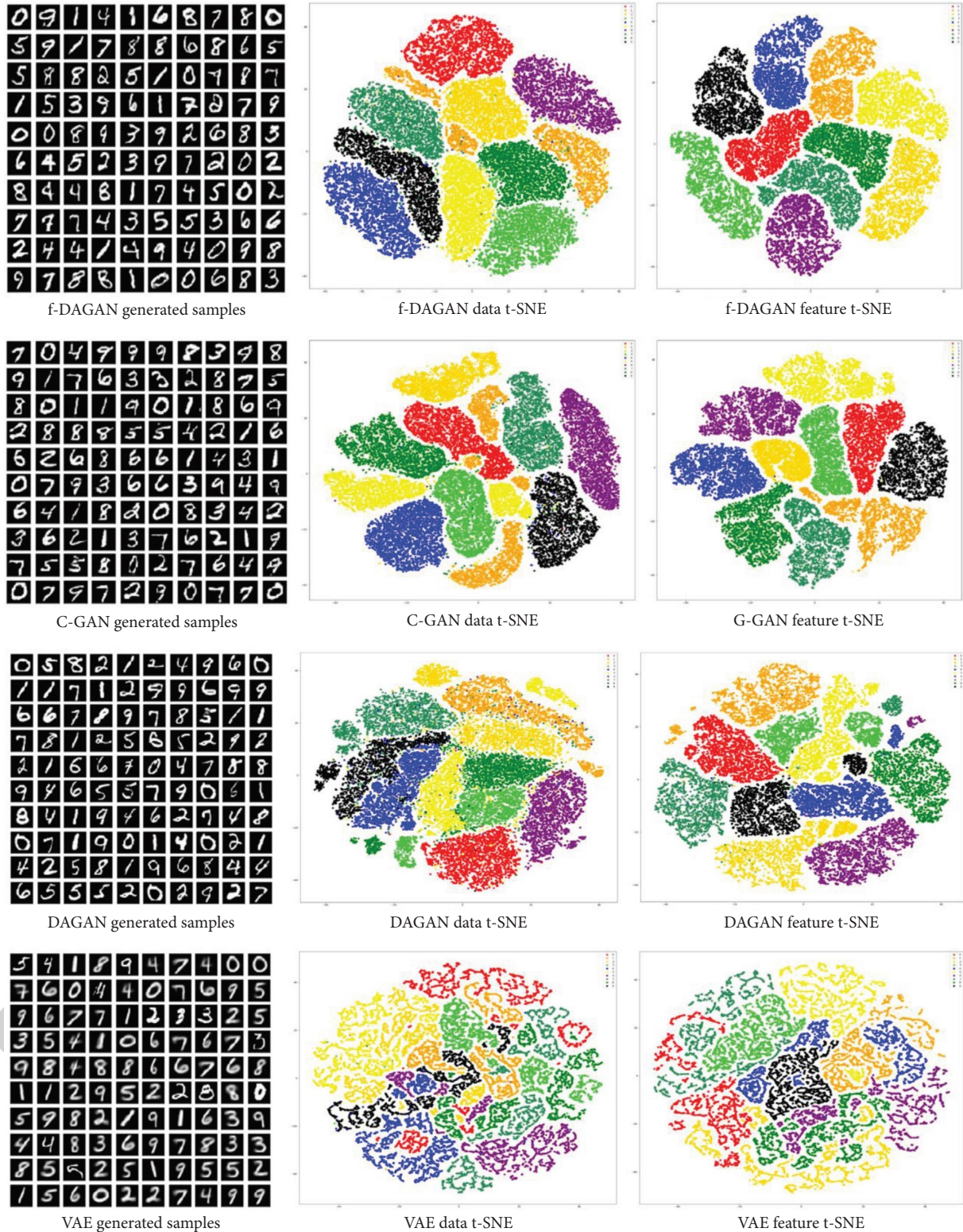


FIGURE 15: t-SNE for 100 samples data and feature space distribution.

8. Result of the One-Class-Based Augmentation

In this section, we compute the classification accuracy, our GAN, on different subsets of MNIST data sets and compare

the results of the architectures described. We demonstrate that our GAN achieves the correct ordinal rankings for each subset of the data set. Due to the architectures having a great dissimilarity in their outputs, we want to start with a baseline

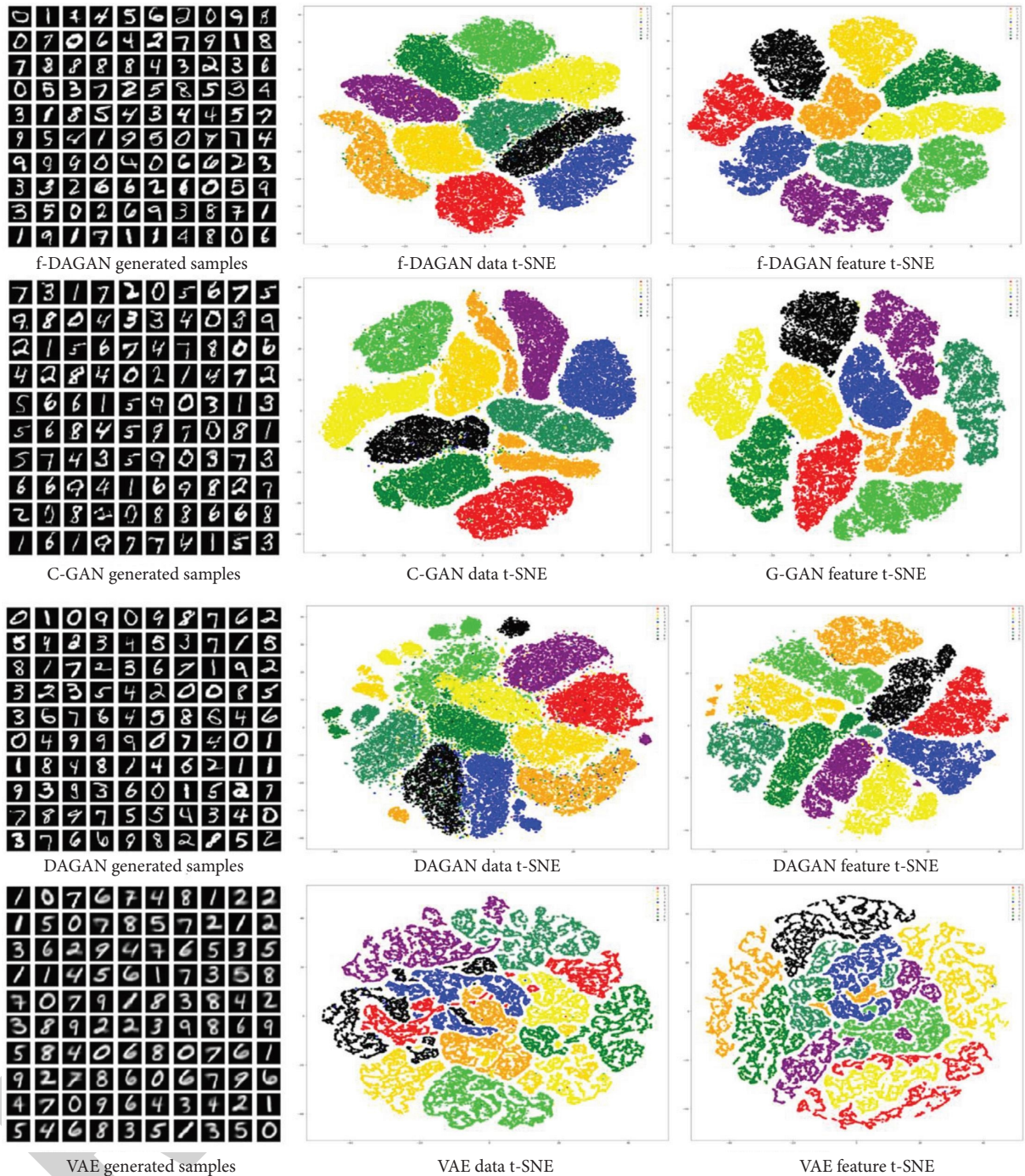


FIGURE 16: t-SNE for 1,000 samples data and feature spaces distribution.

task to ensure the model works under supervision before proceeding to more complicated comparisons that may look equivalent to human observers and vary subtly. For our MNIST experiment, we tested DAGAN, VAE, C-GAN, and f-DAGAN. We can observe that the correct ordinal ranking is achieved by the measure, highlighting that the measure detects the missing modes of the distribution. By ranking the small DCGAN better than weakened GAN, it highlights that it is not fooled by noise and by ranking C-GAN better than small GAN; it further highlights the importance of the full distribution for a better score.

8.1. Quality of Augmented Data. We demonstrate visualization maps of the generated feature and data from different models (Figures 13–17). Figure 12 shows the MNIST data set and their corresponding data and feature visualization, and Figure 18 shows the hard example generated (Figure 19) from h-DAGAN and their respective data and feature distribution.

9. Classification Performance

We evaluated the classification accuracy of the output obtained by different network ResNet50 classifiers. The

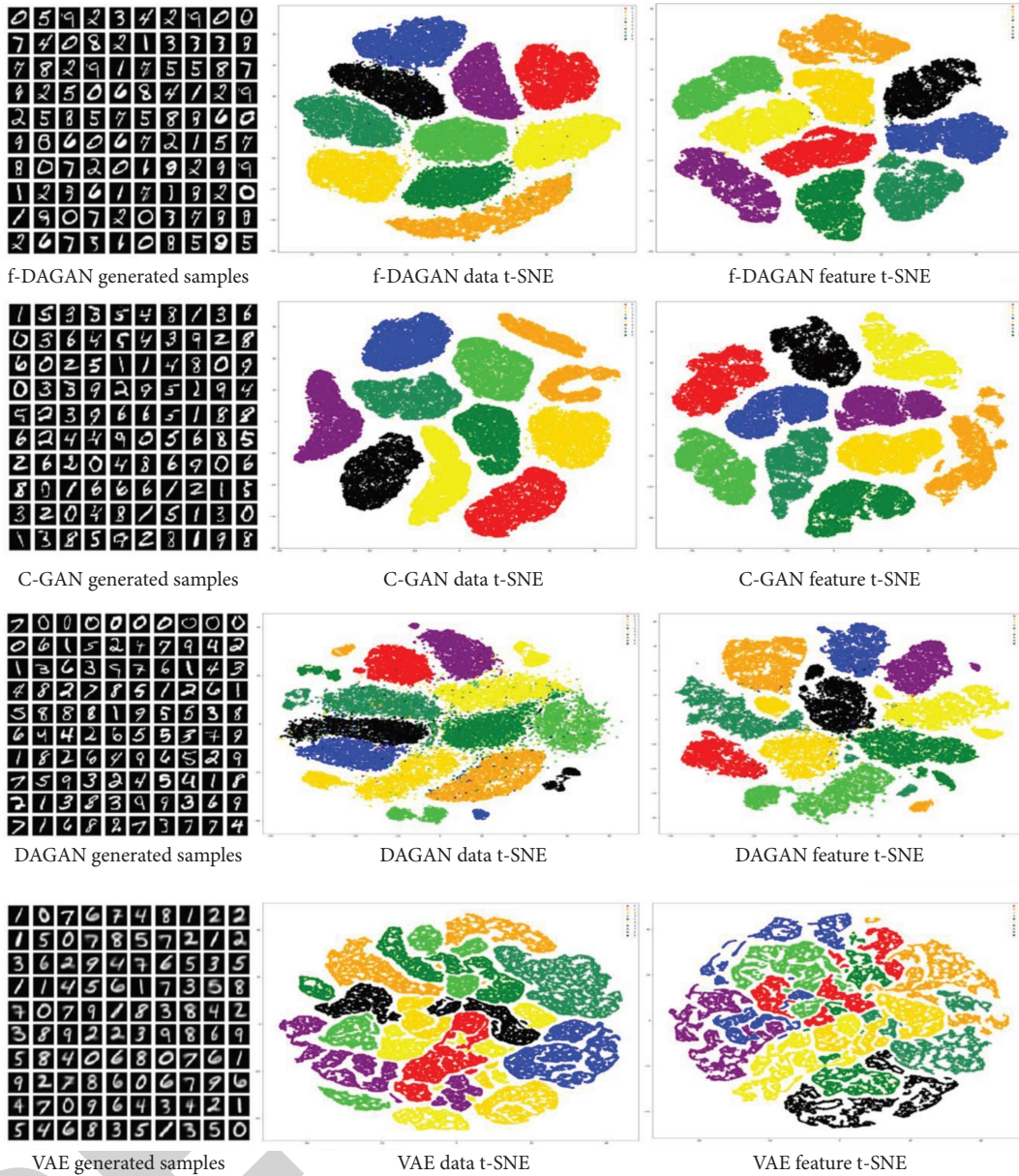


FIGURE 17: t-SNE for all MNIST data and feature spaces distribution.

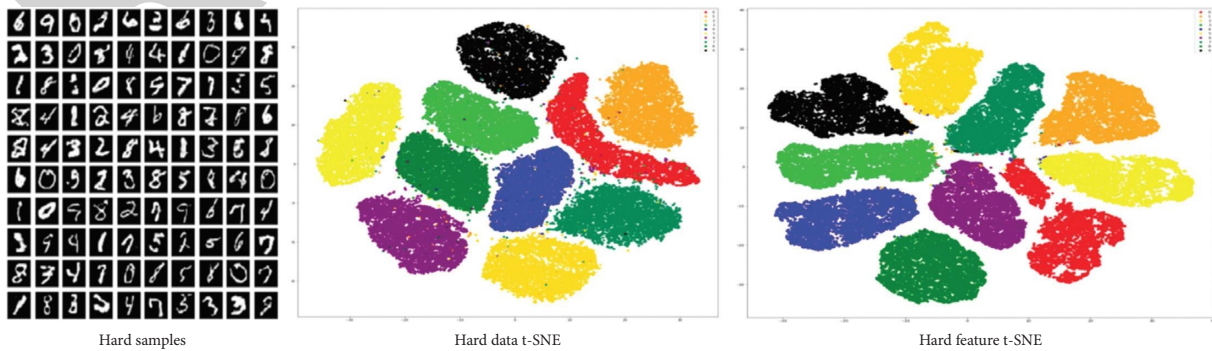


FIGURE 18: t-SNE for h-DAGAN data and feature distribution.

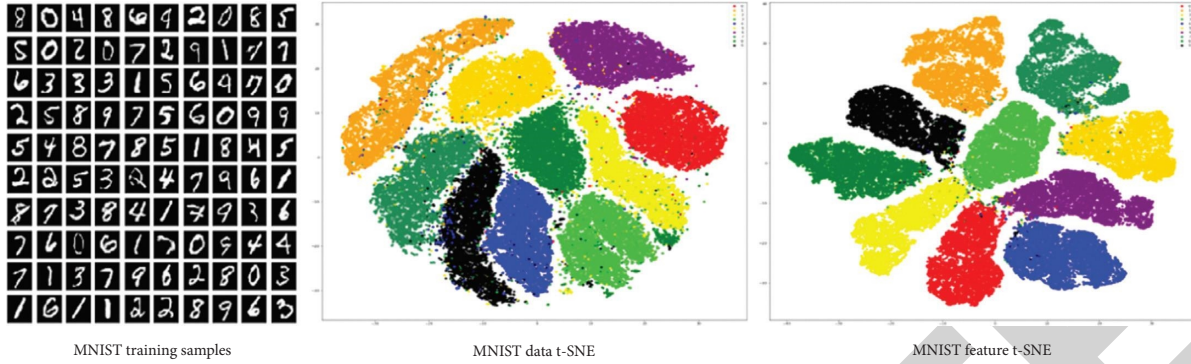


FIGURE 19: t-SNE for MNIST data and feature distribution.

evaluation metric we employ to measure the accuracy of the ResNet50 classifier is defined as the total number of correctly classified samples divided by the total number of test samples. To compute various networks, we train an image classification network created by different networks and then evaluate its output in a real-world image test set. To evaluate the creation of synthetic data the experiments were done using the following steps:

- (1) Trained the different networks using the randomly selected 50 samples (each class 5 samples), 100 samples (each class 10 samples), 1,000 (each class 100 samples), 10,000 (each class 1,000 samples), and a full training set of original data sets.
- (2) Used the trained different networks to generate a new synthetic data set with the exact size of the original.
- (3) The new data set is used to train classification accuracy using ResNet50. For training ResNet50, we used a total of 60,000 data sets with different combinations such as:
 - (a) 50 selected samples (5 from each class) and generated samples of 59,050
 - (b) 100 selected samples (10 from each class) and generated samples of 59,900
 - (c) 1,000 selected samples (100 from each class) and generated samples of 59,000
 - (d) 10,000 selected samples (1,000 from each class) and generated samples of 50,000
- (4) The ResNet50 network is tested using the test set of the original data set.

Intuitively, this measures the difference between the learned (i.e., generated image) and the target (i.e., real image) distributions. We can conclude that the image generated is similar to real images if the classification network can correctly classify real images, which learns features for discriminatory images generated for different classes. In other words, network training is akin to a recall measure, as a good network training performance shows that the generated samples are diverse enough. Network testing often needs adequate precision because the consistency of the sample may influence the classifier.

TABLE 1: Random five samples classification accuracy and GQI.

Data sets	Accuracy (%)	GQI	LS
MNIST (full)	99.13	1	0.873
VAE	91.18	0.9199	0.471
CGAN	92.43	0.9325	0.637
DAGAN	92.57	0.9339	0.732
f-DAGAN	93.14	0.9396	0.739

TABLE 2: Random 10 samples classification accuracy and GQI.

Data sets	Accuracy (%)	GQI	LS
MNIST (full)	99.13	1	0.861
VAE	92.07	0.9287	0.433
CGAN	92.93	0.9375	0.615
DAGAN	92.97	0.9379	0.717
f-DAGAN	94.42	0.9526	0.702

We reported the quantitative results of classification accuracy and GQI in Tables 1 and 2. Tables 3 and 4 is for experiments using the ResNet50 classifier. ResNet50 classifier provides the lower result in less number of training images. Because the ResNet50 network structure requires a large amount of data to train. We can test sample images that are collected from the MNIST test set. MNIST 50 consists of 5 samples from each class; MNIST 100 consists of 10 samples from each class; MNIST 1,000 consists of 100 samples from each class; MNIST 10,000 consists of 1,000 samples from each class; and MNIST 60,000 unlabeled data are used for adversarial training. In Table 1, our f-DAGAN performs better results than other networks. However, C-GAN classification accuracy is quite similar to DAGAN and higher than VAE. Then, for each trained GAN generated, we make random synthetic images samples, and we applied the LS and other GQI measures to the generated image sets and the original image subset. Results are shown in Tables 1–4. LS agrees with FID; F-DAGAN is the best; GAN is the worst model.

We choose different samples from a single class for a few-shot purpose to visualize in supporting and classify the pattern of data generate with various scenerios such as number of training samples is selected. Further, distribution of generated data with different class. It shows that our

TABLE 3: Random 100 samples classification accuracy and GQI.

Data sets	Accuracy (%)	GQI	LS
MNIST (full)	99.13	1	0.851
VAE	93.76	0.9459	0.418
CGAN	94.31	0.9514	0.652
DAGAN	95.12	0.9597	0.783
f-DAGAN	96.51	0.9736	0.710

TABLE 4: Random 1,000 samples classification accuracy and GQI.

Data sets	Accuracy (%)	GQI	LS
MNIST (full)	99.13	1	0.886
VAE	95.92	0.9677	0.497
CGAN	96.63	0.9748	0.698
DAGAN	96.72	0.9756	0.738
f-DAGAN	98.93	0.9979	0.746

approach have capable to distribute with other methods. We used other GAN measures also for comparison with different methods even in every method our approach performed well.

10. Conclusion and Future Work

In the few shot context, lack of training data, classifying images, and labeling training data remain still a challenging problem. In this project, we concentrate on f-DAGAN design architecture by combining different feature vectors; we successfully build a model that can generate realistic images even a few samples available in the training set. In conclusion, this work shows the feasibility of generating synthesized training data generation using adversarial training with few training data required to achieve the performance of the analysis. We trust that our method provides valuable insights into the fine-grained data augmentation problem and opens a new horizon for deep learning with fewer amounts of data.

In the future, our f-DAGAN framework can be extended in various directions. For example, it is possible to utilize other different layers features or more proper architectures or training schemes that could further improve f-DAGAN performance. Specifically, the concatenation of the generator feature and image feature improves the visual quality and image diversity. The current study provides a basis for work employing various features or prior information to better design GAN generators and discriminators. In addition, we have planned to implement a hard example generation network to improve classification accuracy.

Abbreviations

D :	Discriminator
G :	Generator
X_{fake} :	Outputs on image
L :	Log-likelihood of source
X_{real} :	Original image
$g(\cdot)$:	Encoder function
$f(\cdot)$:	Decoder function

z :	Bottleneck layer
D :	Dimensions
$[W \times H \times C]$:	Feature dimensions
CL:	Classifier loss
x_i and x_j :	Sources
x_i and p_{x_j} :	Images.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

References

- [1] G. Yang, S. Yu, H. Dong et al., "DAGAN: deep de-aliasing generative adversarial networks for fast compressed sensing MRI reconstruction," *IEEE Transactions on Medical Imaging*, vol. 37, no. 6, pp. 1310–1321, 2017.
- [2] S. Rajendran, S. K. Mathivanan, P. Jayagopal et al., "Emphasizing privacy and security of edge intelligence with machine learning for healthcare," *International Journal of Intelligent Computing and Cybernetics*, vol. 15, no. 1, pp. 92–109, 2021.
- [3] S. Rajendran, S. K. Mathivanan, P. Jayagopal et al., "Language dialect based speech emotion recognition through deep learning techniques," *International Journal of Speech Technology*, vol. 24, no. 3, pp. 625–635, 2021.
- [4] K. Gao, F. Han, P. Dong, N. Xiong, and R. Du, "Connected vehicle as a mobile sensor for real time queue length at signalized intersections," *Sensors*, vol. 19, no. 9, p. 2059, 2019.
- [5] W. Guo, N. Xiong, H. C. Chao, S. Hussain, and G. Chen, "Design and analysis of self-adapted task scheduling strategies in wireless sensor networks," *Sensors*, vol. 11, no. 7, pp. 6533–6554, 2011.
- [6] X. Wang, Q. Li, N. Xiong, and Y. Pan, "Ant colony optimization-based location-aware routing for wireless sensor networks," in *Proceedings of the International Conference on Wireless Algorithms, Systems, and Applications*, pp. 109–120, Dallas, TX, USA, October 2008.
- [7] Y. Jiang, G. Tong, H. Yin, and N. Xiong, "A pedestrian detection method based on genetic algorithm for optimize XGBoost training parameters," *IEEE Access*, vol. 7, pp. 118310–118321, 2019.
- [8] R. Wan and N. Xiong, "An energy-efficient sleep scheduling mechanism with similarity measure for wireless sensor networks," *Human-centric Computing and Information Sciences*, vol. 8, no. 1, pp. 1–22, 2018.
- [9] E. Gayakwad, J. Prabhu, R. V. Anand, and M. S. Kumar, *Training time reduction in transfer learning for a similar dataset using deep learning. Intelligent Data Engineering and Analytics*, pp. 359–367, Springer, Singapore, 2021.
- [10] H. Altae-Tran, B. Ramsundar, A. S. Pappu, and V. Pande, "Low data drug discovery with one-shot learning," *ACS Central Science*, vol. 3, no. 4, pp. 283–293, 2017.
- [11] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, pp. 453–465, 2013.

- [12] H. Larochelle, D. Erhan, and Y. Bengio, "Zero-data learning of new tasks," in *Proceedings of the 23rd national conference on Artificial intelligence AAAI*, vol. 1, Chicago, Illinois, July 2008.
- [13] Z. Al-Halah, M. Tapaswi, and R. Stiefelhagen, "Recovering the missing link: predicting class-attribute associations for unsupervised zero-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5975–5984, Las Vegas, NV, USA, June 2016.
- [14] M. Norouzi, T. Mikolov, S. Bengio et al., "Zero-shot Learning by Convex Combination of Semantic Embeddings," 2013, <http://arxiv.org/abs/1312.5650>.
- [15] V. K. Verma and P. Rai, "A simple exponential family framework for zero-shot learning," in *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 792–808, September 2017.
- [16] Y. Li and D. Wang, "Zero-shot Learning with Generative Latent Prototype Model," 2017, <http://arxiv.org/pdf/1705.09474>.
- [17] W. J. Scheirer, A. Rocha, A. Sapkota, and T. E. Boult, "Towards open set recognition," *TPAMI*, vol. 36, 2013.
- [18] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," *ICML deep learning workshop*, vol. 2, no. 0, 2015, July.
- [19] O. Vinyals, C. Blundell, T. Lillicrap, and D. Wierstra, "Matching networks for one shot learning," *Advances in Neural Information Processing Systems*, vol. 29, 2016.
- [20] A. Khoreva, R. Benenson, E. Ilg, T. Brox, and B. Schiele, "Lucid data dreaming for video object segmentation," *International Journal of Computer Vision*, vol. 127, no. 9, pp. 1175–1197, 2019.
- [21] A. Mehrotra and A. Dukkipati, "Generative Adversarial Residual Pairwise Networks for One Shot Learning," 2017, <http://arxiv.org/abs/1703.08033>.
- [22] E. Schwartz, L. Karlinsky, J. Shtok et al., "Delta-encoder: an effective sample synthesis method for few-shot object recognition," *Advances in Neural Information Processing Systems*, vol. 31, 2018.
- [23] Y. X. Wang, R. Girshick, M. Hebert, and B. Hariharan, "Low-shot learning from imaginary data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7278–7286, Salt Lake City, UT, USA, June 2018.
- [24] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the International Conference on Machine Learning*, pp. 1126–1135, PMLR, July 2017.
- [25] S. Ravi and H. Larochelle, "Optimization as a Model for Few-Shot Learning," 2016.
- [26] E. Grant, C. Finn, S. Levine, T. Darrell, and T. Griffiths, "Recasting Gradient-Based Meta-Learning as Hierarchical Bayes," 2018, <http://arxiv.org/abs/1801.08930>.
- [27] C. Finn, K. Xu, and S. Levine, "Probabilistic model-agnostic meta-learning," *Advances in Neural Information Processing Systems*, vol. 31, 2018.
- [28] M. S. Kumar, M. Z. Khan, S. Rajendran, A. Noor, A. S. Dass, and J. Prabhu, "Imbalanced Classification in Diabetics Using Ensemble Machine Learning," *Computers, Materials and Continua*, vol. 72.
- [29] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," vol. Vol. 2, pp. 1735–1742, in *In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. Vol. 2, pp. 1735–1742, IEEE, New York, NY, June 2006.
- [30] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," vol. 1, pp. 539–546, in *In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 539–546, IEEE, San Diego, CA, USA, June 2005.
- [31] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: a unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 815–823, Boston, MA, USA, June 2015.
- [32] J. Wang, Y. Song, T. Leung et al., "Learning fine-grained image similarity with deep ranking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1386–1393, San Juan, PR, USA, June 2014.
- [33] T. Xiao, S. Li, B. Wang, L. Lin, and X. Wang, "Joint detection and identification feature learning for person search," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3415–3424, Honolulu, HI, USA, July 2017.
- [34] S. Bai, Y. Li, Y. Zhou, Q. Li, and P. H. Torr, "Adversarial metric attack and defense for person re-identification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 6, pp. 2119–2126, 2020.
- [35] Z. Wu, A. A. Efros, and S. X. Yu, "Improving generalization via scalable neighborhood component analysis," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 685–701, September 2018.
- [36] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [37] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208, Salt Lake City, UT, USA, June 2018.
- [38] J. Wang, Y. Zou, P. Lei, R. S. Sherratt, and L. Wang, "Research on recurrent neural network based crack opening prediction of concrete dam," *Journal of Internet Technology*, vol. 21, no. 4, pp. 1161–1169, 2020.
- [39] C. Chen, K. Li, W. Wei, J. T. Zhou, and Z. Zeng, "Hierarchical graph neural networks for few-shot learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 1, pp. 240–252, 2021.
- [40] D. Cao, K. Zeng, J. Wang et al., "Bert-based deep spatial-temporal network for taxi demand prediction," *IEEE Transactions on Intelligent Transportation Systems*, 2021.
- [41] B. Pu, K. Li, S. Li, and N. Zhu, "Automatic fetal ultrasound standard plane recognition based on deep learning and IIoT," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7771–7780, 2021.
- [42] J. Wang, Y. Yang, T. Wang, R. S. Sherratt, and J. Zhang, "Big data service architecture: a survey," *Journal of Internet Technology*, vol. 21, no. 2, pp. 393–405, 2020.
- [43] J. Li, H. Li, G. Cul, Y. Kang, Y. Hu, and Y. Zhou, "Gacnet: a generative adversarial capsule network for regional epitaxial traffic flow prediction," *CMC-COMPUTERS MATERIALS & CONTINUA*, vol. 64, no. 2, pp. 925–940, 2020.
- [44] R. Jothikumar, *Applying blockchain in agriculture: a study on blockchain technology, benefits, and challenges*. *Deep Learning and Edge Computing Solutions for High Performance Computing*, pp. 167–181, Springer, Heidelberg, Germany, 2021.
- [45] F. Zhang, H. Zhao, W. Ying, Q. Liu, A. N. J. Raj, and B. Fu, "Human face sketch to RGB image with edge optimization

- and generative adversarial networks,” *Intelligent Automation & Soft Computing*, 2020.
- [46] S. A. Vlahopoulos, O. Cen, N. Hengen et al., “Dynamic aberrant NF- κ B spurs tumorigenesis: a new model encompassing the microenvironment,” *Cytokine & Growth Factor Reviews*, vol. 26, no. 4, pp. 389–403, 2015.
- [47] S. Sabour, N. Frosst, and G. E. Hinton, “Dynamic routing between capsules,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [48] Y. Ye, L. Wang, Y. Wu et al., *GAN Quality index (GQI) by GAN-induced Classifier*, 2018.
- [49] X. Li, L. Yu, C. W. Fu, M. Fang, and P. A. Heng, “Revisiting metric learning for few-shot image classification,” *Neurocomputing*, vol. 406, pp. 49–58, 2020.

RETRACTED