Journal of Theoretical and Applied Information Technology
30th June 2023. Vol.101. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

CLASS IMBALANCE LEARNING WITH COSTSENSITIVE-ACGAN

ATHIRAH HAZWANI BINTI ABDUL HALIM[1], NORIDAYU BINTI MANSHOR[2], NOR AZURA BINTI HUSIN[3]

[1,2,3]Department of Computer Science, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400, Serdang, Selangor

E-mail: [1]athirahazwani@gmail.com, [2]ayu@upm.edu.my, [3]n_azura@upm.edu.my

ABSTRACT

The class imbalance problem has been recognized in many real-world applications and negatively affects machine learning performance. Generative Adversarial Networks or GANs have been known to be the next best thing in image generation. However, most GANs do not consider classes and when they do, cannot perform well under the imbalance problem. Based on related works, the modification of the loss function and various resampling methods have been commonly applied to counter the problem of class imbalance. In this study, CostSensitive-ACGAN is introduced which is a variation of Auxiliary Classifier GAN (ACGAN) that can work better under the class imbalance condition. This method incorporated the idea of applying cost-sensitive learning in the loss function to further improve the classification of minority classes. Cost-sensitive parameters are determined adaptively according to the classification error of the class to improve minority classes presence. By applying higher misclassification costs for minority classes, these instances can be magnified and recognized by the discriminator thus improving image generation altogether. This method has shown comparatively competitive results with existing benchmark models.

Keywords: *Computer Vision, Generative Adversarial Networks, Imbalance Learning, Cost-sensitive, Convolutional neural network, Loss function, Image generation.*

1. INTRODUCTION

The emergence of technology in past years calls for rapid development in the field of computer vision. Computer vision is an area of artificial intelligence (AI) that enables computers and systems to formulate, and derive meaningful information from digital images, videos, and other visual inputs. In AI, computers are programmed to be the brain and computer vision is the eyesight for the system to be able to see, observe and understand. Computer vision has been employed in many areas such as industry, healthcare, e-commerce, and many more. There are several major areas in computer vision that are being actively researched, for example, image reconstruction, image segmentation and classification, motion tracking, object detection, and more.

While the concept of computer vision is inspired by human vision and expected to give the same outcome, human vision already has a collection of lifetime context to train how to tell objects apart, regenerate images from imagination and memories and detect anomalies in an image. Regardless of the problem, computer vision needs a lot of data for it to be able to discern distinctions and ultimately recognize images. However, that is not always possible with real-life datasets. For instance, in the task of cancer detection, there is an imbalance between healthy and cancerous cells. The pixels of the target class which is the cancerous cells are sparse compared to healthy cells. This can affect the model in a way that it will not have enough examples of cancerous cells to learn from. The key component in any machine learning model is to have enough data for the model to make the right decision without bias. However, the class imbalance will deviate the bias towards the majority class and somehow will ignore the minority class altogether.

In recent years, Generative Adversarial Networks (GANs) have gained mass attention in computer vision. It is a machine learning framework designed by [1] that consists of two models that are trained simultaneously – a generative model G and a discriminative model D. These two neural networks are being pitted against each other. The generative model produces ‘fake’ output from random noise, trying to fool the discriminator about the output’s legitimacy, and the discriminator tries to determine whether the synthetic output is fake or not. This competition will drive both models to improve their methods until the fake sample is indistinguishable from the genuine ones. GAN can be used for semi-supervised learning where it can improve the performance of a supervised task such as classification with the existence of unlabelled data. However, the most common task for GAN is to be able to produce images that humans can find visually realistic. GAN models can learn from data distribution through its discriminator and allows the generator to generate images that have the same features but with different variations. The likeness of the features shows how much the generator has fool the discriminator because the aim of the model is to produce images with different variations with the same features.

Some GAN applications include image generation for anomaly detection as seen in [2] where they introduced Anomaly GAN, to solve the problem of anomaly detection from aerial images. The model is trained by feeding the model with an urban space dataset and a rural space dataset with annotated anomalous regions. Anomalous regions are spaces that contain human-made structures in this problem. This way, more variation of different possible landscapes with the anomalous region can be generated instead of manual construction of the region. Next, text-to-photo image synthesis can be done by using Stacked Generative Adversarial Networks [3]. StackGAN consisted of a two-stage process where the first stage sketches the promotive shape and colors of the object based on the text description and the second stage takes the output of the first stage along with the description to further refine the output image. This model is able to generate images not only from image input but also from textual input to further enhance the generated images.

However, it is particularly important that GAN can perform under imbalanced conditions. According to recent research, as mentioned in [4], GAN has a big impact in the healthcare domain. Traditional data augmentation methods have become irrelevant as they only learn from original distribution without introducing new variations into the training. Not only GAN is needed to generate more possible incidences among different patients to further enhance detection ability but for it to be able to perform under data scarcity is also very important. This is due to healthcare data has been known to be one of the famous datasets that is highly imbalanced. Imbalanced datasets can hinder the model's ability to learn minority instances thus affecting not only the image formation but also will affect deep learning classifier negatively. Since formation of outputs in GAN depends heavily in the discriminator and generator working together, having a discriminator that can distinguish not only fake vs. true sample but discriminate classes successfully can contribute to the better generation of images. Since the discriminator is a classification model, it is possible to manage training under imbalanced conditions by applying methods like sampling methods, ensemble, and cost-sensitive learning that usually is done to counter imbalances in image classification problems. In related works, we explored these methods and decisions on what is suitable to be implemented in GANs architecture.

The class imbalance problem remains one of the prominent problems in machine learning, especially in the computer vision domain. The inability of models to appropriately learn from minority instances is continuously being researched. In addition, it is found that GAN can be used to combat this issue and is being hyped as "the most interesting thing that has happened to the field of machine learning in the last 10 years" by Yann LeCun, the vice president and chief AI scientist of Facebook [5]. However, the discrepancy of minority class data still affected GANs ability to produce high-quality images and it is important to solve this issue as important domains rely heavily on GAN. Therefore, it is important to continue research to suit evolving needs in the world of AI.

2. RELATED WORKS

2.1 Imbalanced Problems in Computer Vision

Class imbalance has been a major problem in the data mining area. Machine learning works by using a training dataset to train a model until a good enough score is achieved e.g., accuracy, and precision. After some fine-tuning, the model is imposed on a test dataset. Class imbalance can affect machine learning by forcing the model to favor the majority class over the minority class, or in the worst-case scenario, just ignore it altogether. This can be highly detrimental to the learning process as the minority class is a class of interest in most cases. The data can be considered imbalanced by looking at the frequency of the classes. This information can be transformed into an imbalance ratio (IR) for better visualization of how vast the gap between the classes is.

Journal of Theoretical and Applied Information Technology
30th June 2023. Vol.101. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A lot of research has been conducted in the past two decades on this issue. The acquisition of data is the first step in developing a computer vision application, followed by pre-processing and pattern recognition steps to perform a certain task. There are subdivisions in computer vision such as object recognition, image reconstruction, and motion analysis. All of these require ample data for the models to be able to make predictions or generate images with high accuracy and quality. As mentioned in earlier sections, highly imbalanced and inadequate images might cause the desired task to be inapproachable. This is unfortunate because highly imbalanced datasets usually are from critical domains such as anomaly detection, medical image analysis, metallic surface defect detection, and many more.

2.2 Imbalanced Data Classification Approaches

There are various optimized solutions and algorithms that can be used to solve the problem of class imbalance. For dealing with imbalanced data, common approaches are divided into two categories which consisted of the resampling method and modification in the algorithm. On the data level, resampling techniques can be used to alter the distribution of the data and provide variation to the dataset. Modification of the algorithm is usually done on the loss function to alter the parameters or to combine different models in hope that it will drive the model to favor minority instances rather than majority instances. This modification can be a good approach as it does not require resampling the dataset which can introduce noise for oversampling and loss of information from undersampling.

Paper by Park et al. [6] mentioned that training an image generative model can be computationally high in terms of class imbalance as tackling this with oversampling, data augmentation of the training data will require a lot of power and memory. Therefore, they introduced a feature dictionary by using a training feature extractor and a classifier. This dictionary supposedly improves synthesizing artificial samples for data augmentation. However, training a dictionary can be tricky because it requires canvassing the whole dataset to be able to have a comprehensive library that works for the minority class. That itself will require more computational power. Another sampling approach by [7] where they utilized ensemble of both undersampling and oversampling methods to counter this issue in image classification. They proposed a combination of both methods by undersampling majority samples randomly and oversample minority class by using kernel based adaptive synthetic (Kernel-ADASYN). This approach can be seen as simple enough however prone to popular problems of sampling method which are loss of data and more noise introduction.

Next, algorithm-level modification has been known to be one of the methods to counter the imbalance problem. This modification is implemented by adjusting the cost or weight schema that will bend the bias and enable the learner to distinguish classes better [8]. One of methods is modification in the loss function. The loss function or otherwise also known as cost function is used to train a neural network and other machine learning models. Most deep learning models have an architecture where the neurons of the last layer are usually activated by a sigmoid or softmax function [9]. Therefore, usage of different loss functions such as cross entropy and focal loss are more appropriate.

In [10], they implemented a new loss function by combining cross entropy and focal loss functions. This implementation shows that a simple tweak in the objective function and by analyzing statistical information of the dataset can improve class imbalance learning. Another approach by [11] shows another variation of the modified cross-entropy loss function. They proposed a complement cross-entropy that forced the model to learn better from minority classes during training by neutralizing the higher softmax scores of wrong classes. [12] has proposed a Class Rectification Loss (CRL) that aimed to rectify learning bias in cross entropy loss in imbalanced training by incrementally reinforcing the minority class decision boundary margins to discover latent class boundaries whilst maximizing their discriminate margins either directly in the decision score space or indirectly in the feature space.

In class imbalance learning, there is a popular approach called cost-sensitive learning. The relevance behind cost-sensitive learning is that many conventional approaches presume equal misclassification error, and this presumption does not hold in practice. As mentioned in [13], for example, in fault diagnosis of tool condition, if a person is trying to find out if a machine is in a healthy state or a failure state, it is known that missing the detection of a failure state could lead to a major accident that costs much more than the others. Cost-sensitive learning approaches are often used to solve imbalanced classification issues with uncertain and non-identical costs at the algorithmic level. This paper proposed using evolutionary computing alongside deep belief network where adaptive differential evolution is used to optimize the misclassification cost and they also mentioned that this method can be adapted to other deep learning models as well. However, this paper only considers binary classification of the dataset, and future work is needed for the use of this method on high-dimensional data.

However, in [14], they implemented cost-sensitive learning for deep feature representation using convolutional neural networks. This approach gives us inspiration on how cost-sensitive learning can be useful in computer vision problems. Their method adaptively sets the class-dependent cost based on the data statistics and during training, the parameters for cost and network are updated alternatively to optimize the process. Another approach to cost-sensitive classification in convolutional neural networks by [15] utilized Global Mean Square Error Separation to further differentiate the cost between classes. This approach also adaptively set the cost for classes by looking at weighted sum errors. These cost-sensitive methods are all implemented on the algorithmic level and do not require modification on the data level. This might reduce the risk that most sampling methods can pose. All in all, it simplifies the whole process from end to end.

2.3 Generative Adversarial Networks

The goal of GAN is to generate output from random input. The components of the architecture are a discriminator, which acts like a police officer that determines the legitimacy of a sample i.e., whether it is fake or not, and a generator whose aim is to forge fake samples that can pass as a true sample by learning the discriminator behavior [1].

GAN on the surface can be seen as generative modeling where it learns the regularities and patterns of the input data in a way that the model can be used to generate or output new examples. However, as mentioned above, the architecture of GAN has enabled a cleverer way of training a generative model by having a built-in discriminator. This unique architecture has allowed the creation of more diverse output from the generative model. The architecture of GAN can be visualized Figure 1.

2.3.1 The discriminator model

The discriminator is simply a classifier where it takes an example from the domain as input which can be real or generated and outputs prediction on its class label whether it is real or fake. These two sources where the training data comes from:

- Real data instances, such as real pictures from the CIFAR-10 dataset. The discriminator uses these instances as positive examples as it is from the true distribution.
- Fake data instances are the fake images created by the generator. The discriminator uses these instances as negative examples during training.

From Figure 1, these instances are represented by 'sample' boxes which represent the two data sources that are being fed into the discriminator.

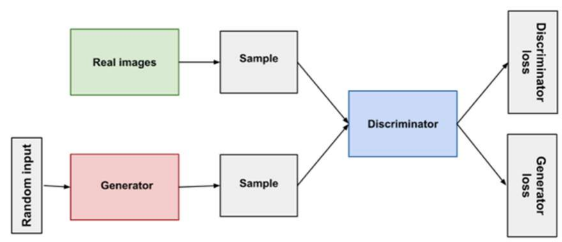

Figure 1: Architecture of GAN

2.3.2 The generator model

Training of the generator requires more integration between the generator and discriminator as opposed to discriminator training. This is because the discriminator simply must identify whether the sample belongs to fake or real distribution, but the generator must learn from the discriminator what a true sample looks like to produce samples that can fool the discriminator. In this process, these components are crucial:

- Random input
- Generator network
- Discriminator network
- Discriminator output
- Generator loss

The usage of neural networks requires some form of input for us to perform the task. Therefore, since our generator's task is to make images that conform to the true data, the introduction of noise is compulsory for the generator to be able to achieve that goal. The Generator takes noise as an input and transforms this into a wide variety of data, sampling from different places in the target distribution.

Both the discriminator and the generator are made of neural nets and to train them, the weights need to be altered to reduce the error or loss of their output. In GAN, the losses are not directly connected to each other. While the discriminator loss focuses on penalizing the discriminator for misclassification between fake and true samples, the generator loss penalizes the generator for making a sample that the discriminator network classifies as fake. By working together, ultimately, these loss functions reflect the distance between the distribution of the data generated by the GAN and the distribution of the real data which is the aim of GAN to replicate a probability distribution.

While GANs are said to be able to learn underlying true data distributions from the limited available images to generate synthetic ones, there is no option to consider specific classes for the task. Therefore, this is one of the challenges as there is not much control over what the output can be.

Conditional GAN or cGAN is the first variation of GAN that introduce control over GAN training. This conditional version of generative adversarial nets has another component which is the labels or classes [16]. This architecture was tested on the MNIST dataset to perform class conditional image generation where a label was determined while training and the model were expected to produce variations of images that obey the class condition. The work in [16] was the jumpstart for more GAN research because the class conditional element has been the pillar for most data augmentation problems that GAN can solve. However, while this research has been one of the most important GAN extensions, the output of the model does not include the generation of the label along with the images.

Therefore, [17] has introduced ACGAN where more structures are added, and all samples are labeled. This will allow class conditional generation due to the existence of an auxiliary decoder that is tasked to reconstruct class labels. This is a good feature that GAN architecture can have as it allows more control over conditional situations.

The performance of ACGAN however deteriorates under class imbalance circumstances as seen in [18]. Due to its inability to perform well under class imbalance, this paper has introduced BAGAN or balancing GAN, which consists of training an autoencoder in an initial stage to further jumpstart its initial value before undergoing GAN training. However, the autoencoder is prone to failure, and training the autoencoder itself is an additional process that is complex.

Next, [19] is a research paper that modifies ACGAN to have the classifier independent from the network rather than having the classifier augmented in the discriminator. This research also shows a modification of ACGAN and the usage of their method for data augmentation for a better classifying process, or in other words, smart oversampling. While this paper highlights its contribution to imbalanced conditions, their proposed model was trained entirely on a balanced dataset.

In this paper, we chose to modify ACGAN as it deals with GAN training with class condition which allows not only control over image generation but also production of image labels as an output. From literature review, cost-sensitive learning has shown to be most suitable to be implemented in our study as we want to incorporate it in ACGAN.

ACGAN model itself consists of two models train together simultaneously therefore usage of cost-sensitive learning is more feasible as opposed to ensemble learning and sampling methods. Usage of ensemble learning and sampling methods along with modification of ACGAN model to work under imbalanced condition will only make the process more complex.

3. METHODOLOGY

This section describes the methodology for this study which includes experimental settings, dataset and metrics to be measured.

3.1 Dataset

MNIST dataset is used in this experiment as many GAN works used this as a benchmark dataset [17],[18]. In the related works, more datasets have been found on dealing with the imbalanced problem, but we chose this dataset to align with our goal which is on image generation task, and it has been used in other GAN papers as well. Other than that, constraints like time and computational power in terms of hardware are also taken into consideration since the training of computer vision models requires a lot of resources.

The training procedure is performed on Google Colab platform using PyTorch framework. Colab allows anybody to perform machine learning and deep learning activities by providing hosted Jupyter

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Notebook as well as providing free access to computing resources including GPUs. PyTorch is an optimized tensor library that is used for deep learning using GPUs and CPUs. Since our study focus on the usage of deep learning for image generation, the usage of PyTorch makes the process more concise and flexible as they have classes and modules such as ‘nn’ classes that provided useful functions that can be used in the study. While this dataset is class balanced, a method from [20] was adopted to create a class imbalanced environment. There are ten classes corresponding to digits from 0 to 9. The number of examples per class in the original training dataset ranges from 5421 in class 5 to 6742 in class 1. In artificially imbalanced versions, data are at random and uniformly subsampled from each class to contain no more than 5 000 examples. The ground truth for the MNIST dataset is visualized in Figure 2 where it consisted of handwritten digits from ‘0’ to ‘9’.

Figure 2: MNIST ground truth

3.2 Training Of ACGAN

A variant of auxiliary classifier generative adversarial networks was proposed where cost-sensitive learning is implemented in its loss function. The architecture of ACGAN can be described in Figure 3.

The architecture of the model consists of a generator and a discriminator. As the name implies, the generator model is the part of our model that generates the image. The parameters in the convolutional networks of the generator are carefully chosen to make sure that the output tensor has the same dimension as the tensor coming from the training set. Since the generator inputs two parameters, random points from latent space and class labels, the generation of the image is expected to be more stable since it depends on the class label when attempting image generation. Discriminator, on the other hand, is provided with both an image and class label. Its task is to classify whether the image is real or fake and responsible for the prediction of the class label of the image.

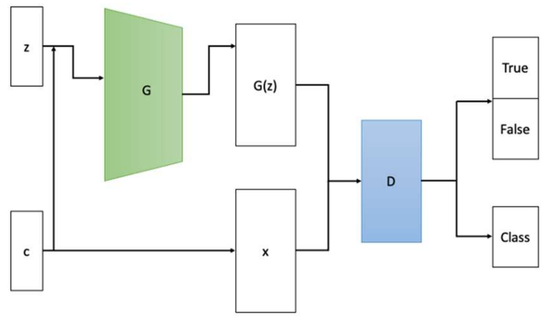

Figure 3: Architecture of ACGAN

Like any other machine learning model, it involves backpropagation where it is an algorithm used to train neural networks. The goal of backpropagation is to adjusts weights in the network to minimize loss function. Loss function is one of components in the model where it calculates the error between predicted and actual values. In ACGAN, cross-entropy loss has been used as a loss function as it deals with multiclass classification task in its discriminator. More elaboration regarding ACGAN loss function as well as modification in our CostSensitive-ACGAN is in Subsection 3.3.

In the ACGAN, every generated sample has a corresponding class label, $c \sim p_c$ in addition to the noise z. Generator, G, uses both to generate images $X_{\{fake\}} = G(c, z)$. The discriminator gives both a probability distribution over sources and a probability distribution over the class labels, $P(S \mid X), P(C \mid X) = D(X)$ [17] .

3.3 CostSensitive-ACGAN

Cost-sensitive learning is frequently used to tackle class imbalance problems, where it aids the model in recognizing difficult-to-learn situations. Cost-sensitive learning is a relatively versatile strategy since it may be applied straight into training without requiring the imbalance data to be altered in the preprocessing step, as approaches using resampling do. Data-level adjustments may result in overgeneralization (e.g., SMOTE) and excessive computational cost. Depending on the cost-sensitive learning approach, a parameter C that signals a misclassification cost during training is specified using a cost function $f(C)$ that depends on many dataset features such as class distribution, class separability, and overall classification error.

Journal of Theoretical and Applied Information Technology
30th June 2023. Vol.101. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Convolutional neural networks (CNNs) are used by GAN in both the discriminator and the generator, and their main goal is to generate an ideal model by reducing global error throughout the backpropagation process. To train a neural network, backpropagation is essential. It is the approach of fine-tuning the weights of a neural network depending on the error rate recorded in the previous epoch (i.e., iteration). By properly setting the weights, error rates can be lowered and improve the generalization of the model, which will make it more dependable.

In this study, cross-entropy is utilized as our loss function. Cross-entropy loss function is used as this research deals with multiclass classification problems and it can output the probabilities of belonging to the classes. Cost-sensitive learning is applied by adjusting the cost, w_c (instead of parameter C as not to confuse with C = classes) based on the error for each class. This is relevant because the minority class sample usually will have higher misclassification errors due to it being hard to recognize the model. To ensure that the minority samples are not overlooked, to punish these disparities, higher weights or costs is assigned for the high misclassification errors, which is the rationale of cost-sensitive learning.

Like the working of the original GAN model, a minmax game takes place here where the discriminator is trying to maximize its reward and the generator is trying to minimize the discriminator's reward. Both models use useful information from each other to improve their own model quality and objective function of ACGAN are modified to improve the model's performance under imbalance conditions.

3.3.1 Loss Function of CostSensitive-ACGAN

There are two parts in the objective function of ACGAN, which consist of the log-likelihood of the correct source L_s, and the log-likelihood of the correct class L_c. The focus is on L_c as it is responsible for the classification of the sample. Its ability to classify minority class samples is amplified by incorporating a cost-sensitive component, parameter w_c.

$$L_s = E[\log P(S = real \mid X_{real})] + E[\log P(S = fake \mid X_{fake})] \quad (1)$$

$$\text{cost}-\text{sensitive } L_c = E[w_c \log P(C = c \mid X_{real})] + E[w_c \log P(C = c \mid X_{fake})] \quad (2)$$

During training, both the weights of the discriminator along with the parameter w are jointly optimized. The comparison between objective functions of the original ACGAN and our modification can be seen in the Table 1.

Table 1: Difference between ACGAN and CostSensitive-ACGAN

Normal ACGAN Objective Function	CostSensitive-ACGAN Objective Function
$L_s = E[\log P(S = real \mid X_{real})] + E[\log P(S = fake \mid X_{fake})]$	$L_s = E[\log P(S = real \mid X_{real})] + E[\log P(S = fake \mid X_{fake})]$
$L_c = E[\log P(C = c \mid X_{real})] + E[\log P(C = c \mid X_{fake})]$	$\text{cost}-\text{sensitive } L_c = E[w_c \log P(C = c \mid X_{real})] + E[w_c \log P(C = c \mid X_{fake})]$

Like mentioned in Subsection 3.3, our process of deriving the parameter w_c, is inspired by usage of cost-sensitive learning in CNN [15]. Error of classification of each class in every training iteration are used to determine suitable a misclassification cost for each class. The weights are updated alongside with the training of neural network. The training procedure can be seen on Algorithm 1.

Algorithm 1: CostSensitive-ACGAN training algorithm:

Initialize $w_c = 1/K$ where K is number of classes
For every epoch:
  For every batch:
  Compute $\text{cost}-\text{sensitive}; L_c + L_s$
  Update weights of D
  Compute $\text{cost}-\text{sensitive}; L_c - L_s$
  Update weights of G
Compute the error for every class using the updated models
Compute softmax of the errors and use that as new w_c

The expected outputs are to produce a trained model that can perform image generation and for the discriminator to classify correctly under imbalanced conditions.

3.4 Architecture of Discriminator and Generator

Generative Adversarial Network consisted of Discriminator and Generator. Since this study deals with image processing problems, convolutional and transposed convolutional layers are used in building the Generator and the Discriminator. The main reason for using convolutional neural networks is it allows convolution operations to be done on image data, which helps to identify important features in images such as edges, corners and other patterns.

In the Discriminator, only standard convolutional layers are being used along with the linear layer as the fully connected layer to produce prediction between true or false samples and classification of the images. As seen on Table 2, the discriminator consists of four convolutional layers as the discriminator is a classification model that classifies true and fake samples along with the classes.

As for the Generator, the goal is to produce bigger dimensions of images from basically nothing, transposed convolutions are used to produce images that match our dataset dimensions which are 1 x 28 x 28 from the initial input of 1 x 1 x 100 vector. Table 2 shows the architecture of the discriminator and generator of CostSensitive-ACGAN. Our generator consists of one linear layer and five transposed convolution layers as the goal of our study is to produce image output.

Kernel, strides, dropout, and nonlinearity are important component in building CNNs. Kernel is a matrix of weights that is applied to small regions of the input image that are used for feature extractions and known as filter. Strides refers to the number of pixels that the kernel moved across the input image to reduce image complexity therefore allows faster computation. Dropout is a regularization technique that is applied on the layers to prevent overfitting and lastly, nonlinearity is introduced through activation function to allows the network to model complex relationships of the data. These layers and hyperparameter settings as seen in Table 2 initially are set based on [17] and as the training progress, trial and error method is done in changing the hyperparameters as to suit the model training with cost-sensitive modification.

Table 2: Architecture of Discriminator and Generator of CostSensitive-ACGAN

Operation	Kernel	Strides	Dropout	Nonlinearity
$G_x(z) - 110 \times 1 \times 1$ input				
Linear	N/A	N/A	0.0	ReLu
Transposed Convolution	5×5	2×2	0.0	ReLu
Transposed Convolution	5×5	2×2	0.0	ReLu
Transposed Convolution	5×5	1×1	0.0	ReLu
Transposed Convolution	5×5	1×1	0.0	ReLu
Transposed Convolution	5×5	1×1	0.0	Tanh
$D_x(z) - 32 \times 32 \times 1$ input				
Convolution	5×5	1×1	0.3	Leaky ReLu
Convolution	3×3	1×1	0.3	Leaky ReLu
Convolution	5×5	1×1	0.3	Leaky ReLu
Convolution	3×3	1×1	0.3	Leaky ReLu

4. RESULT AND DISCUSSION

For the evaluation of the proposed method, as mentioned in Subsection 3.1, the methodology was conducted on MNIST dataset. The experiment consisted of training the modified model for image generation and testing the quality of images generated on a pre-trained classification model.

The results are divided into two categories which are in terms of structural similarity index measure or SSIM to ensure diversity in the image generated and classification accuracy on image generated by our model. The reasoning behind these measures is to ensure images generated to achieve the goals of generating minority class images that are representative of the desired class, not repetitive, and not identical to the ones from original dataset. Failure to fulfill these goals is possible due to model experiencing mode collapse, unable to learn to produce diverse images that is different from original dataset and simply learnt to redraw the available training images. To measure this, its structural similarity is compared to ensure diversity and classification accuracy to make sure images generated show belonging to their classes.

4.1 Experimental Result on MNIST

The experiments were conducted on MNIST dataset. While this dataset is class balanced, a method from [20] is adopted to create a class imbalanced environment. This is conducted due to where a class was selected, and significant amount

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

of its instances was dropped. There are ten classes corresponding to digits from 0 to 9. The number of examples per class in the original training dataset ranges from 5421 in class 5 to 6742 in class 1. The distribution of original dataset is illustrated in Figure 4.

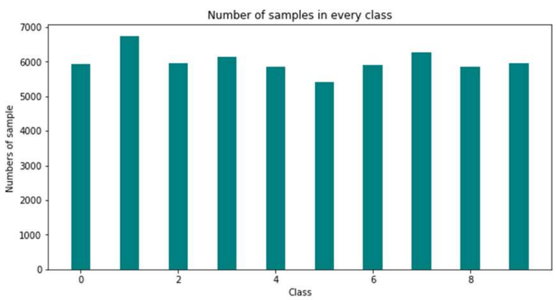

Figure 4: Distribution of samples against classes in MNIST dataset.

According to [20], there are two types of imbalances that is believed to represent real-world cases which are step imbalance and linear imbalance. Step imbalance is implemented in line with methodology in [18] where an imbalance environment was created by omitting 60% to 95% percentage of images with the class label '0' for each training. This condition can be represented by calculating the imbalance ratio.

Table 3: Imbalance Ratio and value of ρ

IR = (# of majority class data): (# of minority class data)

Percentage	Imbalance Ratio	ρ
Original	(1:1)	1
60	(20:1)	20
80	(40:1)	40
95	(190:1)	190

Step imbalance is when the number of examples is equal within the minority classes and equal within the majority classes but differs between the majority and minority classes. This type of imbalance is characterized by two parameters. One is the fraction of minority classes defined by:

$$\mu = \frac{|\{i \in \{1, \dots, N\}: C_i \text{ is minority}\}|}{N} \quad (3)$$

where C_i is a set of examples in class I and N is the total number of classes. The other parameter is a ratio between the number of examples in majority classes and the number of examples in minority classes defined as follows.

$$\rho = \frac{\max_i\{|C_i|\}}{\min_i\{|C_i|\}} \quad (4)$$

For this experiment, based on our imbalance ratio, the value of μ is 0.1 since only one minority class is considered with the variation of ρ as different degrees of imbalance in the minority class are introduced as seen in Table 3. The number of samples in the minority class are introduced against the majority class on different imbalance conditions in the Figures 5 – Figure 7 by using the step imbalance process. From the figures, the class '0' distribution significantly decreased as compared to other classes as it has been assigned as minority class.

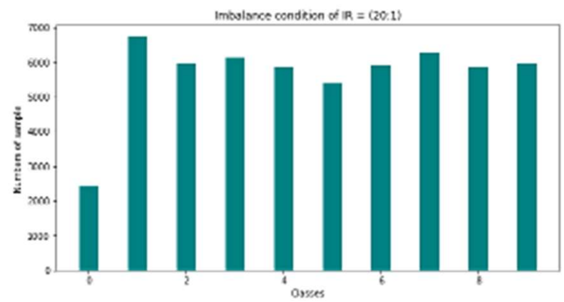

Figure 5: Imbalance condition of IR = (20:1)

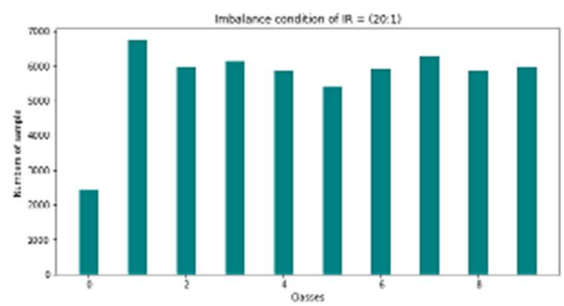

Figure 6: Imbalance condition of IR = (40:1)

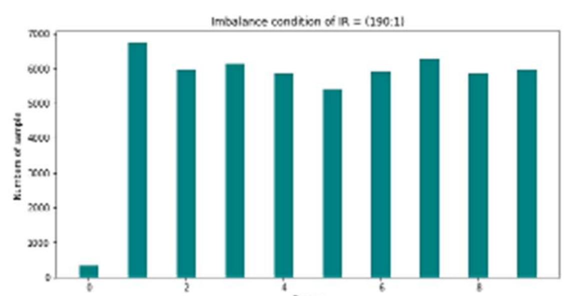

Figure 7: Imbalance condition of IR = (190:1)

CostSensitive-ACGAN were trained to generate the images under imbalanced conditions and the

Journal of Theoretical and Applied Information Technology
30th June 2023. Vol.101. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

example of image generated can be seen in Figure 8 and 9 for the minority class which is class '0'.

When Figure 8 and Figure 9 are compared, it is evident the images generated by proposed CostSensitive-ACGAN are more well-formed. This shows that CostSensitive-ACGAN is able to perform image generation task better than the original ACGAN model under highly imbalanced conditions.

One of the main focuses of GAN-based model is to have diversity in images generated. Mode collapse is known to be the main problem that researchers might encounter during GAN-based model development where the generator only outputs a single prototype that maximally had fooled the discriminator.

Figure 8: Image generated with 95% imbalance using ACGAN

Figure 9: Image generated with 95% imbalance using the CostSensitive-ACGAN

Therefore, in this study, the diversity of images generated were measured using Structural Similarity Index or SSIM. SSIM was first introduced by [21]. This metric focus on assessing perceptual image quality by assessing the degradation of structural information. It is inspired by the human visual perception system which it is highly capable of identifying structural information from pictures and able to identify differences between the information extracted between generated images and the reference.

While it is assumed that this metric focus on looking at differences to ensure that generated images look the same as the reference, the goal for this research is the opposite. This is due to the GAN model being made to ensure the image generated has variability or if that does not happen, this is known as mode collapse where the model is not able to produce enough variation of images. In this study, mode collapse happened during the experimentation phase as seen in Figure 10.

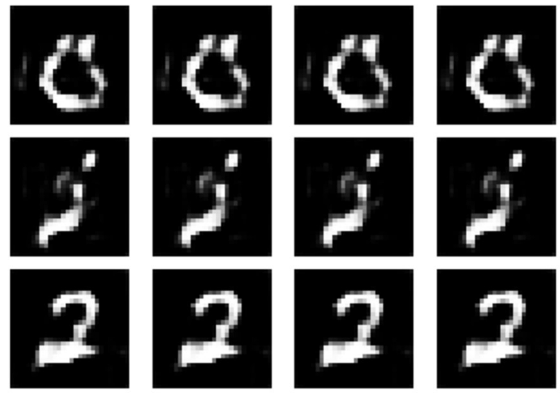

Figure 10: Examples of mode collapse

When training a GAN model, your goal is to have wide variety of outputs. Generator's goal is to always produce plausible output which the discriminator can pass as true image. However, when the discriminator is stuck in a local minimum and not able to reject plausible input from generator, the generator will only rotate through small set of output types as the discriminator is already overly optimized.

Values of SSIM are measured and samples that have higher diversity resulted in low SSIM scores and vice versa. The result is tabulated in Table 4.

Table 4: SSIM value across benchmarks vs CostSensitive-ACGAN

Data Imbalance	ACGAN	BAGAN	CostSensitive-ACGAN
60	0.4	0.3	0.12
80	0.4	0.3	0.12
95	0.42	0.35	0.17

CostSensitive-ACGAN has performed comparatively competitive as compared to other models. This shows that our approach can produce images that are structurally varied. However, the quality deteriorates as the imbalance increases. Using SSIM as a metric somehow can be a little bit tricky. This is because when a model produces lowly-formed images they can have high variation as well. Therefore, other metrics such as accuracy are needed to verify that synthesized images can be recognized and discriminable by the independent classifier.

Another important focus in developing class conditional GAN models is to have synthesized images appear to belong in the intended classes. It is important that the image generated has high

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

discriminability to show the proposed model able to produce images that belong to their classes accordingly. The performance of the proposed model is measured against other benchmarks in terms of accuracy by using the generated image and test them using ResNet-18 classifier.

Table 5: Accuracy for ResNet-18 classifier for generated images

Data Imbalance	ACGAN	BAGAN	CostSensitive-ACGAN
60	0.9925	0.9975	0.98
80	0.9885	0.9895	0.98
95	0.9775	0.98	0.97

From Table 5, image generated by CostSensitive-ACGAN produced comparable results with ACGAN and BAGAN. It is evident that CostSensitive-ACGAN not only can produce images with high variability but also produce images with high discriminability as well. However, like the previous metric, strong imbalance can affect the quality and discriminability of generated images as the accuracy decreases along with the increase of imbalance. In addition, possibly more training is required to achieve more discriminable images to produce stronger accuracy.

All in all, the proposed method, CostSensitive-ACGAN implied different degrees of data imbalance. The imbalance environment was created by using the step imbalance method which it consisted of one minority class against the majority classes. For image diversity, the experimental results showed CostSensitive-ACGAN to have the lowest SSIM score resulting in higher diversity in generated images. In terms of image discriminability, while not all imbalance conditions result in better accuracy than ACGAN and BAGAN, the difference in the tabulated results is very little thus making the generated images still very discriminable. It is evident that the usage of cost-sensitive learning in the discriminator has a significant impact on the process. Compared to [18], our process is much simpler as it does not require additional autoencoder training and is still able to produce promising results. While the results are relatively competitive, it is possible that more training is required so that the generator can learn to make better images from a very well-trained discriminator.

5. CONCLUSION

In this study, an image generation method for data imbalance named CostSensitive-ACGAN was proposed. The modified ACGAN structure contains a generator and a modified discriminator that applies cost-sensitive learning. This method is introduced to counter image generation problem under imbalance condition. The existing methods such as BAGAN, while also working on to combat imbalance problem in using GAN, requires more process beforehand. Therefore, this method is introduced in hope not only to enable image generation using GAN under imbalance condition, but also simplify its process altogether.

To verify the functionality of proposed method, two performance metrics which are SSIM, and Accuracy are used. SSIM is used to verify high variability in the image generated and this method surpasses benchmark methods in producing different variation of images. Accuracy, on the other hand, is used to verify generated images to have high discriminability by independent classifier. Based on our result, while not every degree of imbalance surpasses better accuracy than our benchmarks, the results are still significant and comparable to others.

6. LIMITATIONS AND FUTURE WORKS

GAN-based models have been known to have various issues when training them. Common issues like mode collapse and non-converging generator did happen during the research. Mode collapse happened when the generator is unable to present variety in its inference as in the example that has been mentioned in the earlier sections. Another problem encountered while developing the model is failure to converge where the discriminator and generator cannot reach equilibrium. While this problem has been handled by tweaking the parameters like batch size, random size etc., for future work, stable training can be achieved by applying Wasserstein GAN or WGAN. Since the performance of our model was only tested on MNIST dataset, the performance of this model can be tested on different datasets such as Flowers or CIFAR-10 where these datasets are consisted of three channel images or RGB images.

REFERENCES:

[1] I. J. Goodfellow *et al.*, “Generative Adversarial Nets,” 2014. [Online]. Available: http://www.github.com/goodfeli/adversarial

[2] M. A. Contreras-Cruz, F. E. Correa-Tome, R. Lopez-Padilla, and J. P. Ramirez-Paredes, “Generative Adversarial Networks

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

for anomaly detection in aerial images," *Computers and Electrical Engineering*, p. 108470, Nov. 2022, doi: 10.1016/J.COMPELECENG.2022.108470.

[3] H. Zhang *et al.*, "StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks," Dec. 2016, [Online]. Available: http://arxiv.org/abs/1612.03242

[4] L. Lan *et al.*, "Generative Adversarial Networks and Its Applications in Biomedical Informatics," *Frontiers in Public Health*, vol. 8. Frontiers Media S.A., May 12, 2020. doi: 10.3389/fpubh.2020.00164.

[5] Yann LeCun, "Quora Session with Yann LeCun," *Quora*, 2016. https://www.quora.com/What-are-some-recent-and-potentially-upcoming-breakthroughs-in-deep-learning (accessed Jun. 14, 2022).

[6] M. Park, H. J. Song, and D.-O. Kang, "Imbalanced Classification via Feature Dictionary-based Minority Oversampling," *IEEE Access*, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3161510.

[7] H. Ding *et al.*, "KA-Ensemble: towards imbalanced image classification ensembling under-sampling and over-sampling," *Multimed Tools Appl*, vol. 79, no. 21, pp. 14871–14888, 2020, doi: 10.1007/s11042-019-07856-y.

[8] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *J Big Data*, vol. 6, no. 1, Dec. 2019, doi: 10.1186/s40537-019-0192-5.

[9] Y. Tian, D. Su, S. Lauria, and X. Liu, "Recent advances on loss functions in deep learning for computer vision," *Neurocomputing*, vol. 497, pp. 129–158, Aug. 2022, doi: 10.1016/J.NEUCOM.2022.04.127.

[10] L. Wang, C. Wang, Z. Sun, S. Cheng, and L. Guo, "Class Balanced Loss for Image Classification," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.2991237.

[11] Y. Kim, Y. Lee, and M. Jeon, "Imbalanced image classification with complement cross entropy," *Pattern Recognit Lett*, vol. 151, pp. 33–40, Nov. 2021, doi: 10.1016/J.PATREC.2021.07.017.

[12] Q. Dong, S. Gong, and X. Zhu, "Imbalanced Deep Learning by Minority Class Incremental Rectification," *IEEE Trans Pattern Anal Mach Intell*, vol. 41, 2019, doi: 10.1109/TPAMI.2018.2832629.

[13] C. Zhang, K. Chen Tan, H. Li, and G. Soon Hong, "A Cost-Sensitive Deep Belief Network for Imbalanced Classification," *IEEE Trans Neural Netw Learn Syst*, vol. 30, no. 1, p. 109, 2019, doi: 10.1109/TNNLS.2018.2832648.

[14] S. H. Khan, M. Hayat, F. A. Bennamoun, and R. T. Sohel, "Cost-Sensitive Learning of Deep Feature Representations From Imbalanced Data," *IEEE Trans Neural Netw Learn Syst*, vol. 29, no. 8, p. 3573, 2018, doi: 10.1109/TNNLS.2017.2732482.

[15] Y. Hou, H. Fan, L. Li, and B. Li, "Adaptive learning cost-sensitive convolutional neural network," *IET Computer Vision*, vol. 15, 2021, doi: 10.1049/cvi2.12027.

[16] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," Nov. 2014, [Online]. Available: http://arxiv.org/abs/1411.1784

[17] A. Odena, C. Olah, and J. Shlens, "Conditional Image Synthesis with Auxiliary Classifier GANs," 2017. [Online]. Available: https://github.com/openai/improved-gan/.

[18] G. Mariani, F. Scheidegger, R. Istrate, C. Bekas, and C. Malossi, "BAGAN: Data Augmentation with Balancing GAN," 2018.

[19] S. Suh, H. Lee, P. Lukowicz, and Y. O. Lee, "CEGAN: Classification Enhancement Generative Adversarial Networks for unraveling data imbalance problems," *Neural Networks*, vol. 133, pp. 69–86, Jan. 2021, doi: 10.1016/J.NEUNET.2020.10.004.

[20] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249–259, Oct. 2018, doi: 10.1016/J.NEUNET.2018.07.011.

[21] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, 2003, pp. 1398-1402 Vol.2. doi: 10.1109/ACSSC.2003.1292216.