

MFC-GAN: class-imbalanced dataset classification using multiple fake class generative adversarial network.

ALI-GOMBE, A., ELYAN, E.

2019



Accepted Manuscript

MFC-GAN: Class-imbalanced Dataset Classification using Multiple
Fake Class Generative Adversarial Network

Adamu Ali-Gombe, Elyan Eyad

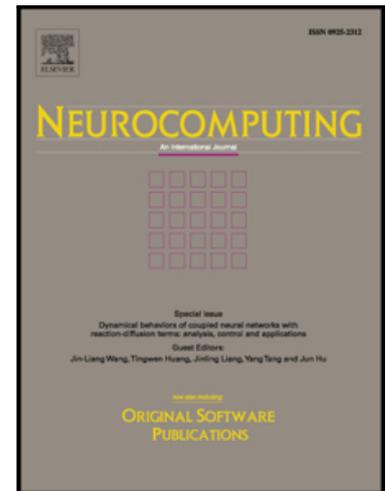
PII: S0925-2312(19)30925-7
DOI: <https://doi.org/10.1016/j.neucom.2019.06.043>
Reference: NEUCOM 20981

To appear in: *Neurocomputing*

Received date: 17 October 2018
Revised date: 8 April 2019
Accepted date: 18 June 2019

Please cite this article as: Adamu Ali-Gombe, Elyan Eyad, MFC-GAN: Class-imbalanced Dataset Classification using Multiple Fake Class Generative Adversarial Network, *Neurocomputing* (2019), doi: <https://doi.org/10.1016/j.neucom.2019.06.043>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.



MFC-GAN: Class-imbalanced Dataset Classification using Multiple Fake Class Generative Adversarial Network

Adamu Ali-Gombe, Elyan Eyad

Robert Gordon University Aberdeen

Abstract

Class-imbalanced datasets are common across different domains such as health, banking, security and others. With such datasets, the learning algorithms are often biased toward the majority class-instances. Data Augmentation is a common approach that aims at rebalancing a dataset by injecting more data samples of the minority class instances. In this paper, a new data augmentation approach is proposed using a Generative Adversarial Networks (GAN) to handle the class imbalance problem. Unlike common GAN models, which use a single fake class, the proposed method uses multiple fake classes to ensure a fine-grained generation and classification of the minority class instances. Moreover, the proposed GAN model is conditioned to generate minority class instances aiming at rebalancing the dataset. Extensive experiments were carried out using public datasets, where synthetic samples generated using our model were added to the imbalanced dataset, followed by performing classification using Convolutional Neural Network. Experiment results show that our model can generate diverse minority class instances, even in extreme cases where the number of minority class instances is relatively low. Additionally, superior performance of our model over other common augmentation and oversampling methods was achieved in terms of classification accuracy and quality of the generated samples.

Keywords: Image Classification, Imbalanced Data, Deep Learning.

1. Introduction

The class-imbalanced problem arises when the samples in a dataset are dominated by one class usually the negative class. It is common across different domains such as security, banking and medicine. This could occur in a binary classification or a multi-classification task [1]. Models trained on a class-imbalanced dataset tend to be biased towards the majority class. Existing approaches address this problem either at the data level or the algorithm level [2]. Data re-sampling techniques such as undersampling and oversampling are applied at data level to ensure equal representation of instances amongst

classes. Algorithmic solutions include modifying the learning objective to ensure equal participation of all classes during training.

Data augmentation is a common technique employed to synthesize more training data. Artificial variations are useful in minimizing any bias in data collection and class-imbalanced problem. For instance, in image domain, augmentation techniques used could range from simple image flips [3], random crops [4], noise [3] distortions to more advanced techniques like PCA colour augmentation [4] and image-pairing [5]. Data augmentation technique can be a source of more training data [6] or a regularizer [5] thereby improving generalization. These techniques have proved to be effective in learning from class-imbalanced datasets. However, in extreme class-imbalanced cases, applying augmentation to few samples may not provide the required variations to produce distinct samples to re-balance the dataset. Furthermore, the problem becomes compounded in a multi-class problem as the performance of a class may be affected while trying to improve another [7]. Besides, existing techniques may not necessarily be useful in deep learning [8].

More recently, Generative Adversarial Networks (GAN) have been used to generate images with high visual fidelity [9]. Researchers have shown that these images can be used as extra training data to support other processes such as classification [6, 10]. A GAN model produces quality samples with the required variations similar to the training data. Different GAN models have been proposed for data augmentation in previous works [1, 11, 12, 6, 13]. Also, GAN was used to tackle imbalanced data in a binary classification problem using none image data in [1] and used by Antoniou *et al.* [11] as an augmentation approach to improve image recognition accuracy. Our approach shares some similarities with these researches but differs in the sense that we use a different GAN model in image classification domain. Moreover, we are interested in performing multiple classification with an imbalance training data. With scarce minority classes, image generation can be challenging because a useful augmentation sample needs to be plausible, diverse and from the required minority class [12, 11].

In this paper, Multiple Fake Class Generative Adversarial Network (MFC-GAN) is proposed. MFC-GAN preserves the structure of the minority classes by learning the correct data distribution and produce unique images whenever it is sampled. We demonstrate the usefulness of MFC-GAN by addressing class-imbalanced problem in a multi-classification task. MFC-GAN differs from other GAN models that implement a classifier alongside the discriminator such as S-GAN [14], AC-GAN [15] and similar frameworks in the sense that we use a multi-fake class GAN model. Multiple fake class feature was implemented in Few-Shot Classifier GAN (FSC-GAN) [16] to generate samples and perform classification. Incorporating more fake classes in the FSC-GAN resulted in artefacts appearing in generated samples which may hinder using such samples as candidates for augmentation. This paper extends FSC-GAN idea and demonstrates that artefacts can be reduced significantly by conditioning image generation on real class labels only and modifying the classification objective. Thus, fake class labels are only employed when classifying generated images.

Incorporating more fake classes in this context stabilizes training early and

generates plausible samples with fewer epochs. Our argument is that since both minority and majority classes come from the same distribution, these classes share some common features. Hence, features learned from majority classes should aid in learning the minority classes. Consequently, class conditioned generation will focus the model into sampling minority classes. Our approach train MFC-GAN on the imbalanced dataset then generate and augment synthetic minority class instances to the original training data. A Convolutional Neural network (CNN) is then trained on the augmented dataset. We evaluated our approach using four imbalanced datasets namely; **E-MNIST**¹ and created artificial imbalance in **MNIST**², **SVHN**³ and **CIFAR-10**⁴ by reducing the number of samples in specific classes. Significant performance gain was obtained when MFC-GAN was used as an augmentation model when compared to the baseline (CNN classification without augmentation) and other common and state-of-the-art methods (SMOTE [17] & AC-GAN [15]).

The main contributions in this paper are as follows:

- MFC-GAN is proposed to learn data representation from low number of samples
- A method for handling class-imbalanced datasets by augmenting the original data with synthesized samples using MFC-GAN
- Experimental framework for evaluating MFC-GAN on four different multi-class imbalanced datasets

The remainder of this paper is organised as follows. In Section 2, we review related work. Section 3 presents the proposed method. Section 4 discusses in details experimental set-up and datasets used. Section 5 present the results obtained. Our findings are discussed in section 6. Finally, we draw conclusions and suggest future directions in Section 7.

2. Related Work

The class-imbalanced problem in binary classification is an active research area which has witnessed the development of well-established techniques. However, little attention is given to class-imbalanced problem in multi-classification [2]. Imbalanced classes in a multi-classification problem may require new sampling strategies and data pre-processing steps [2] other than those used in binary classification. Existing methods for handling such problem includes multi-class decomposition [7], Class Rectification Loss (CRL) [8] and mean squared false error [18]. Resampling methods such as oversampling and undersampling are

¹<https://www.nist.gov/itl/iad/image-group/emnist-dataset>

²<http://yann.lecun.com/exdb/mnist/>

³<http://ufdl.stanford.edu/housenumbers/>

⁴<https://www.cs.toronto.edu/~kriz/cifar.html>

widely used in this area. However, oversampling is prone to over-fitting and undersampling may discard essential data points [19].

Buda *et al.* [19] showed in an experimental study how the performance of CNN drops significantly when the data is imbalanced. Wang *et al.* [18] modified the learning algorithm to account for class-imbalance by penalising the misclassification of minority class instances (i.e., cost-sensitive methods). However, applying such methods require careful consideration of the cost matrix settings, which can be tricky in a real-life problem [2].

Common methods such as Synthetic Minority Over-Sampling Technique (SMOTE) proved to be ineffective in handling class-imbalance in extreme cases (hugely imbalanced datasets) and results in performance deterioration of the learning algorithm in such scenarios [2]. SMOTE can also lead to over-generalization with high variance [18].

In deep models such as CNN for example, Class Rectification Loss (CRL) [8] was used to handle class-imbalance. CRL algorithm performs hard mining of the minority class in each batch forcing the model to create a boundary for each minority class with a hard positive and negative threshold. Other approaches such as Large Margin Local Embedding (LMLE) [20] employs clustering among classes to maintain the structure of the minority data. However, these techniques can be computationally expensive in large data domain [8].

Data augmentation techniques are increasingly becoming an integral part of deep model approaches for classification. Dosovitskiy, et al. CNN [21] proposed a method (Exemplar) based on systematic augmentation of data and achieved state-of-the-art results on CIFAR-10 dataset. Data augmentation is a widely used technique to handle class-imbalanced datasets. Ali *et al.* [3] used affine transformation and noise distortion across classes to generate more samples and reduce the impact of class-imbalance. However, trivial augmentation may not suffice for extreme class-imbalanced data or when sufficient data is not available. Besides, orientation-related features in some domain may limit the application of simple augmentation approaches [12]. Thus, more sophisticated augmentation techniques such as image pairing [5] and mixup [22] have been proposed.

In recent years, generative models were successfully used to generate samples. GANs proved to be state-of-the-art in generating and capturing data [9]. In an imbalanced dataset, the aim is to generate class-specific samples, therefore supervised GAN models such as Conditional GAN (C-GAN) [23] is a potential solution for such a problem. However, these models and other established GAN frameworks such as vanilla GAN [24] and AC-GAN [15] have performed poorly on class-imbalanced datasets by failing to generate the required minority samples [12, 25]. Recently, good performance was reported by [6] using a Deep Convolutional GAN (DCGAN) [26] to synthesise artificial liver lesion images. This was achieved by using traditional augmentation techniques to oversample the training set. Similarly, Baur *et al.* [13] generated high-resolution skin lesion images using MelanoGAN (a variant of DCGAN + Laplacian GAN [27]) from a small dataset of 2k samples. The model was used to synthesise more skin lesion samples to reduce the effect of class-imbalanced data in training a ResNet-50 [28] for classification. These examples show that trivial data augmentation

techniques can be successful in handling class imbalance related problems ([6], [13]). However, it should be noted that these examples were applied to binary datasets with no orientation dependent features or fuzzy class boundaries.

Other approaches combine GAN model with other generative processes such as an auto-encoder training. Features learned by the auto-encoder are then used to initialize the generator and discriminator of the GAN model. This may require a second training step [12] or joint training [25] to perform conditional adversarial training. Data Augmentation Generative Adversarial Networks (DA-GAN) [11], Balancing GAN (BAGAN) [12] and Fine-grained Multi-attribute GAN (FM-GAN) [25] used a similar strategy to synthesize more samples for augmentation. Image refinement is another technique used which preserves the image class while producing diverse synthetic samples. Zhu *et al.* [10] applied image translation to generate minority samples using a reference sample in an emotion recognition task. However, this approach was evaluated using two closely-related classes (i.e. translate a face to another face image). Other approaches re-parametrise the adversarial training by adding extra losses or stricter conditions during the generation. This enforces learning and generation of minority samples such as in DeliGAN [29]. The latent space in DeliGAN is parametrized by a Gaussian Mixture Model (GMM) whose parameters are learned alongside the GAN parameters.

In summary, resampling methods don't perform well in hugely imbalanced datasets. Traditional data augmentation are still widely used. However, these are limited and often don't generate enough data variance, especially in extreme cases. GAN-based methods provide a more realistic solution to generate data samples and handle class-imbalance (i.e., a multi-modal [11, 12], image-translation [10]). Unlike these methods, MFC-GAN is simpler to train and generates specific-class samples even in extreme cases.

3. Method

Our approach uses MFC-GAN to generate plausible samples which were used to augment training data. GAN models are trained using two sets of training data; the original data from the training set (or real images) and generated samples (or fake images) obtained from the generator. Similarly, we consider real labels as the corresponding labels of the original training data and the associated fake labels for generated images. Class labels were prepared by converting each label into an n bit one-hot encoding vector, where n is the number of classes. To accommodate fake classes, we pad n zeros to the right of the label encoding to obtain a new representation for real labels. Hence, for each real label c , a corresponding fake class label c' is generated by padding n zeros to the left of the original label encoding. For example, if the real label for class 0 is encoded as 1000000000, we now represent this class label by 10000000000000000000 and its associated fake label as 00000000001000000000. To generate class specific samples, we conditioned MFC-GAN generator using real labels only. Label conditioning encourages the generator to work towards producing realistic samples and controls the generation of class-specific samples [14]. When training

MFC-GAN, we classify real images into real classes and generated images into different fake classes. MFC-GAN is trained with a modified AC-GAN objective. The objective maximises the log-likelihood of classifying real samples into real classes C and fake samples into fake classes C' as shown in equations 1, 2 and 3.

$$\mathcal{L}_s = \mathbb{E}[\log P(S = real|X_{real})] + \mathbb{E}[\log P(S = fake|X_{fake})] \quad (1)$$

$$\mathcal{L}_{cd} = \mathbb{E}[\log P(C = c|X_{real})] + \mathbb{E}[\log P(C' = c'|X_{fake})] \quad (2)$$

$$\mathcal{L}_{cg} = \mathbb{E}[\log P(C = c|X_{real})] + \mathbb{E}[\log P(C = c|X_{fake})] \quad (3)$$

Where \mathcal{L}_s is used to estimate the sampling loss, which represents the probability of the sample being real or fake. \mathcal{L}_{cd} and \mathcal{L}_{cg} are used to estimate the classification losses over the generator and the discriminator. X_{real} represents the training data and X_{fake} is the set of generated images.

3.1. MFC-GAN Vs FSC-GAN

As can be seen in equation 2 and 3, MFC-GAN classification objective differs from what was implemented in AC-GAN and FSC-GAN. Both FSC-GAN and MFC-GAN discriminators classify generated samples into different fake classes. This prevents classifying unrealistic samples into real classes by providing fine-grained training to the model. However, MFC-GAN differs from FSC-GAN in the way the loss function of the generator is defined as can be seen in Equation 3. In other words, in our model, the model's generator is penalised according to how far the generated sample is from the real class label. Notice, that in the FSC-GAN model, the generator model is penalised according to how far the generated sample is from fake class label. By having this key difference in our model, we ensure that poor generated samples guarantee higher loss, which is not necessarily the case in the FSC-GAN settings. This has also promoted early convergence of the model where MFC-GAN model proved to be able to generating plausible samples with far fewer epochs than both AC-GAN and FSC-GAN.

Furthermore, for every iteration, equation 2 means that the discriminator classifies samples as real or fake with the associated class (i.e., real class 1 or fake class 1) while equation 3 means that with every generator iteration, it tries to classify fake samples as real classes. As the generator performance improves, only subtle differences exist between the two set of images (fake, real) and this acts as a regularizer that penalizes the discriminator as the model approaches optimal performance. Similar to FSC-GAN, MFC-GAN is also capable of handling labelled and unlabeled data in training. Depending on the availability of labels, the network switcher feature [16] enables both models to alternate between two training modes. This switcher is a piece-wise function that oscillates between supervised and unsupervised training. Although, there is a slight difference in the way classification loss is evaluated (as shown in equation 2).

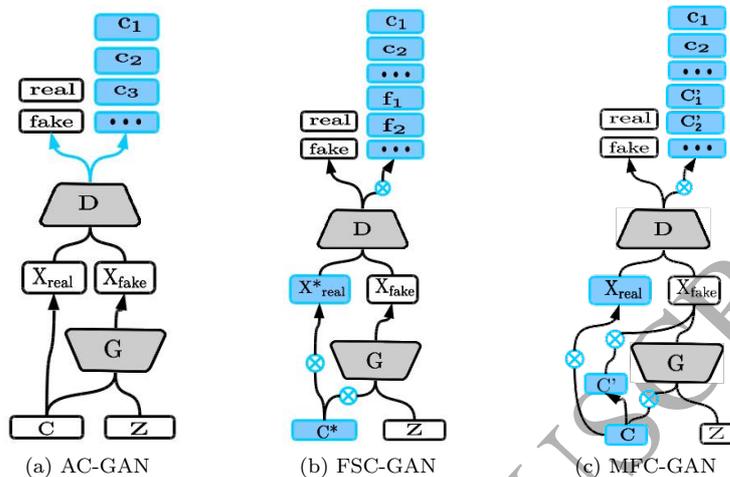


Figure 1: Comparing MFC-GAN architecture with AC-GAN and FSC-GAN models. C is a set of labels, z is a random noise vector, G is the generator, D is the discriminator, real & fake are GAN outputs representing the probability of an image being real or fake, c_1, \dots, c_n are the set of real classes, f_1, \dots, f_n and c'_1, \dots, c'_n are sets of fake classes, X_{real} is the original training images, X_{fake} is the set of generated images and \otimes is the network switcher feature that alternates between labelled and unlabeled training.

Figure 1 compares the structure of MFC-GAN to FSC-GAN and AC-GAN. With labelled data, the MFC-GAN discriminator is trained to maximise the sum of \mathcal{L}_s and \mathcal{L}_{cd} while the generator is trained to maximise the difference between \mathcal{L}_s and \mathcal{L}_{cg} . In this setup, the MFC-GAN generator is sampled using a noise vector conditioned on real class labels. In the absence of labels, MFC-GAN is trained using \mathcal{L}_s only and behaves like a vanilla GAN model as shown in equation 4. In the latter case, the generator is sampled using a noise vector only. Although, in these experiments, this feature was not exploited. Further comparisons and discussions around these differences can be found in section 5 and Figure 2.

$$V(D, G) = \begin{cases} C = \{\emptyset\} : \mathcal{L}_s \\ C \neq \{\emptyset\} : \mathcal{L}_s \pm \mathcal{L}_c \end{cases} \quad (4)$$

4. Experiments

The architecture of both the discriminator and generator used on MNIST & E-MNIST were adopted from FSC-GAN, details of this can be found in [16]. Regarding SVHN & CIFAR-10 experiments, we used the same architecture as in the original AC-GAN model [15], and added spectral weight normalization [30]

in both generator and discriminator for both AC-GAN, FSC-GAN & MFC-GAN. This is to ensure a fair comparison.

4.1. Experimental Set-up

In order to evaluate the performance of our method, we compared it with AC-GAN [15] which is one of the best supervised generative models. We also compared our method with SMOTE [17] which is one of the most common methods for generating data to handle class-imbalanced datasets. This was achieved by first training a classifier on the original dataset. This forms a baseline for comparing performances of the models. Then MFC-GAN, AC-GAN, and SMOTE were used to generate more samples from the minority classes. The resulting samples were then augmented into the original dataset and classification was performed again using CNN. The performance of the CNN on the three different augmented datasets are then compared and discussed. Algorithm 1 provides a schematic overview of this experiment.

Algorithm 1 Experimental procedure

procedure DATA AUGMENTATION

$d \leftarrow$ original imbalanced dataset

train:

$MFC\text{-}GAN(d)$

$AC\text{-}GAN(d)$

$FSC\text{-}GAN(d)$

augment:

$d_{mfc} \leftarrow d + MFC\text{-}GAN_{samples}$

$d_{smote} \leftarrow d + SMOTE_{samples}$

$d_{acgan} \leftarrow d + AC\text{-}GAN_{samples}$

$d_{fscgan} \leftarrow d + FSC\text{-}GAN_{samples}$

classify:

$r_1 \leftarrow CNN(d)$

$r_2 \leftarrow CNN(d_{mfc})$

$r_3 \leftarrow CNN(d_{smote})$

$r_4 \leftarrow CNN(d_{acgan})$

$r_5 \leftarrow CNN(d_{fscgan})$

compare(r_1, r_2, r_3, r_4, r_5)

end procedure

Furthermore, the fidelity of generated minority samples from MFC-GAN was compared to state-of-the-art AC-GAN.

All models were implemented using `tensorflow 1.0`⁵ and `Keras 2.0`⁶. SMOTE was implemented using ⁷. Models were evaluated subjectively based on the

⁵<https://www.tensorflow.org/>

⁶<https://keras.io/>

⁷<https://github.com/tgsmith61591/smrt>

plausibility of samples (i.e., visual inspection) and objectively by assessing the classification performance after augmentation.

4.2. Datasets

The models were tested using four publicly available datasets. These are, MNIST [31], E-MNIST [32], SVHN [33] and CIFAR-10 [34] datasets.

MNIST is a dataset of hand-written digits with ten classes (0 – 9) consisting of 28×28 grey-scale images. MNIST has a total of 50k images training set, 10k images for validation and 10k test images. Both the training and validation sets were merged to form a more significant training set, and the test set was used as a holdout sample in classification. MNIST is a balanced dataset, and so we induced imbalance among its classes by undersampling. Two classes were chosen arbitrarily and their instances were reduced significantly to mimic a multi-classification imbalance problem. We could have chosen more but given the size of the dataset, we do not want to inhibit learning due to the number of training examples. In our experiments, different experiments were run with adjacent classes chosen as minority classes in each run. The first run considers 0 and class 1 as minority, then classes 2 and 3 and so on. In each run only 50 samples in these classes were used (about 1% of the original). The rest of the classes remained unchanged and experiments were carried out on the new imbalanced MNIST dataset.

E-MNIST is an extended version of MNIST. The dataset also consists of 28×28 grey-scale images with 62 classes (0 – 9, A – Z and a – z). For our experiments, the *byclass* grouping was used with 814,255 samples in total. The dataset consists of 697,932 training samples and 116,323 samples for testing. The distribution of samples across classes in the training data is not balanced; thus, experiments on this dataset did not require inducing artificial imbalance. E-MNIST contains many classes with a considerably small number of samples than others with 21 out of 62 classes having less than 3000 samples. These classes include class G, K, Q, X, Z, c, f, i, j, k, m, o, p, q, s, u, v, w, x, y & z, where the 10 least populated were used in our experiment.

SVHN dataset contains google street view of house numbers across ten categories (1, 2, 3, 4, 5, 6, 7, 8, 9, 0). This dataset consists of 32×32 pixels images with 73k and 26k train and test images set. These images appear noisy with other numbers in the background and the dataset is not balanced. Similar to MNIST, we induced artificial imbalance by considering 50 samples in classes 1&2 to form a multi-class imbalance scenario with the rest of the classes unaltered.

CIFAR-10 dataset is made up of 32×32 images of real objects. It has fifty thousand training images grouped into ten classes namely; Aeroplane, Automobile, Bird, cat, Deer, Dogs, Frog, Horse, Ship and Truck. Samples distribution across these classes is balanced with five thousand samples in each class. We induced artificial imbalance by considering 50 samples in *Aeroplane* and *Automobile* classes. The dataset has ten thousand tests set with one thousand samples from each category. In all the datasets, the test sets were used as a hold out in evaluating the classification model.

4.3. Image generation

We perform augmentation by synthesizing more samples. AC-GAN, FSC-GAN and MFC-GAN were first trained using the imbalanced datasets described in section 4.2. The three models are then used to generate minority samples, these samples were then used to augment the original datasets. Samples generated using SMOTE were produced by repeatedly applying SMOTE to over-sample the class of interest as the minority sample and the rest of classes as the majority sample.

Regarding SVHN and CIFAR-10, the four models MFC-GAN, FSC-GAN, AC-GAN, and SMOTE were used to generate the class of interest (the minority class). These are classes 1&2 in SVHN and *Aeroplane* and *Automobile* classes in CIFAR-10. As for E-MNIST, we chose classes $G, K, Q, f, j, k, m, p, s, y$ as the class of interest (minority classes). These were chosen because they have the least number of instance. Every class in the MNIST dataset was considered a minority class (by undersampling each of them at different runs).

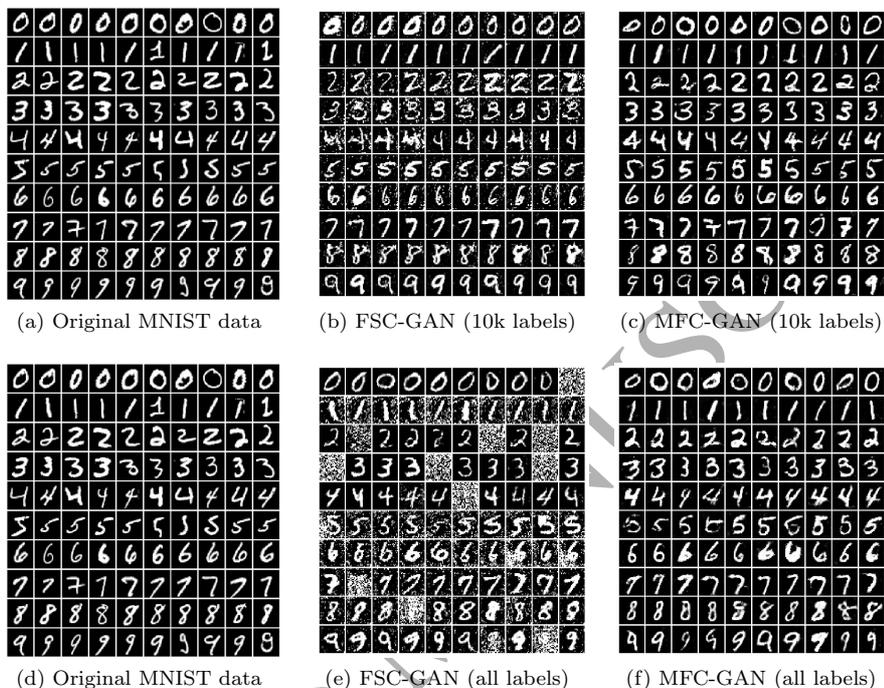
4.4. Image Classification

Our classification model is Convolutional Neural Network (CNN). The CNN used for MNIST & E-MNIST has three layers with a soft-max activation layer on top. The first two layers are convolution layers with 3×3 kernels which are followed by a 2×2 max-pooling layer. The two layers have a filter map of size 32 and 64 respectively. This is followed by a fully connected layer with 128 neurons that feeds into the final soft-max layer (with 10 and 62 output neurons for MNIST and E-MNIST respectively). All layers are ReLu activated, and a dropout ratio of 0.5 was used in the fully connected layer. Adadelata optimiser [35] (an extension of Adagrad) was used with default settings and weights were initialised using random uniform distribution. The same model was used in SVHN experiment but with a different input channel and input size to accommodate the images.

For CIFAR-10 experiment, we increase the number of convolution layers to three (with channel sizes 32,32 & 64) and reduced the dropout ratio to 0.2. The number of neurons in the fully connected layer was also increased to 512 and the CNN was trained with SGD optimizer using learning rate of 1e-3 and decay of $lr/epoch$. The initial experiment trains the CNN on the original dataset. Then the model is trained by augmenting the dataset using one of the approaches considered. Both CNNs were trained using a batch size of 64 for CIFAR-10 and 100 for the others over 25 epochs and we evaluate on the holdout test sets from each of the datasets described.

The choice of the CNN models above was made to evaluate the proposed method (MFC-GAN) on generating images of minority classes. This was achieved by first, classifying the original datasets using CNNs, then classifying the augmented datasets and comparing the results. In this way, we can have an objective measure for the quality of samples generated by our model and how it does compare with other methods. This is in addition to the subjective evaluation based on the visual inspection of the generated images.

5. Results

Figure 2: *FSC-GAN versus MFC-GAN on MNIST dataset*

A preliminary experiment comparing MFC-GAN against FSC-GAN [16] was carried out using the MNIST dataset. This was achieved by reducing the number of labelled instances in the dataset across all classes. Figure 2 shows that MFC-GAN generated better quality samples and considerably reduced the amount of artefacts. The results also show that MFC-GAN can effectively handle both labelled and unlabelled instance. It is worth noting that MFC-GAN generates good quality images even in the presence of a large number of unlabelled instances (50K unlabelled instances, Figure 2c). The training time was also reduced considerably (by a factor of 10) with MFC-GAN producing plausible samples at about 50 epochs while FSC-GAN reaches optimum at 500 epochs. The results suggest that MFC-GAN would be a suitable model for augmentation.

MFC-GAN was also applied to imbalanced datasets to evaluate the quality of generated samples. The models were initially evaluated subjectively using visual inspection. Figures 4, 3, 5, 6 and 7 compare the original images and the generated samples. The minority classes in MNIST, SVHN, and CIFAR-10 dataset are highlighted using a red line for the different experiments conducted. For E-MNIST, we report the performance from the ten minority classes. Using

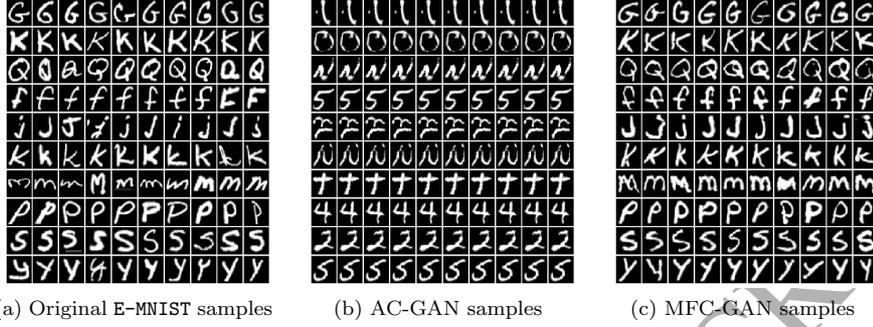


Figure 3: Original images (left) with AC-GAN and MFC-GAN generated samples (middle, right) from E-MNIST dataset with minority class instances highlighted in red.

MFC-GAN model, we were able to generate the minority classes without artefacts. Thus, the samples are good candidates for augmentation. As can be seen, poor minority class samples were generated by AC-GAN model and in some cases, it was biased toward the majority class. The classification performances are reported in tables 2, 1 and 3. Several common evaluation metrics were used in the experiments including balanced accuracy, sensitivity, specificity and Geometric Mean (G-Mean). These metrics were computed as follows:

$$Sensitivity = \frac{tp}{tp + fn} \quad (5)$$

$$Specificity = \frac{tn}{tn + fp} \quad (6)$$

$$G - Mean = \sqrt{Sensitivity \times Specificity} \quad (7)$$

$$F1 - score = \frac{2tp}{(2tp + fp + fn)} \quad (8)$$

$$BalancedAccuracy = \frac{tp + tn}{2} \quad (9)$$

$$Precision = \frac{tp}{tp + fp} \quad (10)$$

$$recall = \frac{tp}{tp + fn} \quad (11)$$

where tp stands for true positive, tn denotes true negative, fp and fn denotes false positive and false negative respectively.

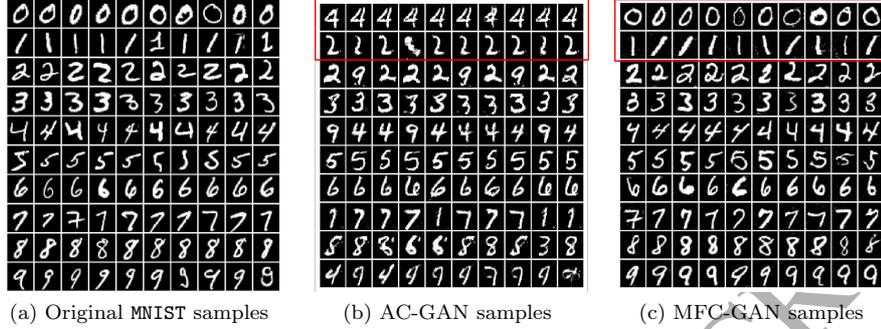


Figure 4: Original images (left) with AC-GAN and MFC-GAN generated samples (middle, right) from MNIST dataset with minority class instances highlighted in red.

Metric	Model	0	1	2	3	4	5	6	7	8	9
Sensitivity	Baseline	0.83	0.93	0.64	0.73	0.68	0.70	0.73	0.65	0.62	0.58
	SMOTE	0.92	0.94	0.76	0.89	0.81	0.87	0.87	0.79	0.79	0.76
	AC-GAN	0.77	0.89	0.55	0.71	0.58	0.88	0.85	0.66	0.68	0.70
	FSC-GAN	0.78	0.87	0.60	0.58	0.49	0.51	0.61	0.48	0.38	0.41
	MFC-GAN	0.98	0.98	0.83	0.85	0.76	0.71	0.88	0.90	0.89	0.83
Specificity	Baseline	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	SMOTE	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	AC-GAN	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	FSC-GAN	1.00	1.00	1.00	0.99	0.99	1.00	1.00	1.00	1.00	1.00
	MFC-GAN	1.00									
Accuracy	Baseline	0.91	0.97	0.82	0.87	0.84	0.85	0.86	0.83	0.81	0.79
	SMOTE	0.96	0.97	0.88	0.95	0.90	0.93	0.91	0.89	0.90	0.88
	AC-GAN	0.89	0.95	0.78	0.85	0.79	0.94	0.92	0.83	0.84	0.85
	FSC-GAN	0.89	0.94	0.80	0.79	0.74	0.75	0.80	0.74	0.69	0.63
	MFC-GAN	0.99	0.99	0.92	0.92	0.88	0.85	0.94	0.95	0.94	0.92
Precision	Baseline	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99
	SMOTE	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99
	AC-GAN	1.00	1.00	1.00	0.99	1.00	0.99	1.00	1.00	0.99	0.95
	FSC-GAN	1.00	1.00	0.99	0.99	1.00	1.00	1.00	1.00	1.00	0.97
	MFC-GAN	1.00	1.00	0.99	1.00	1.00	1.00	1.00	0.99	0.99	0.99
Recall	Baseline	0.83	0.93	0.64	0.73	0.68	0.70	0.73	0.65	0.62	0.58
	SMOTE	0.92	0.94	0.76	0.89	0.81	0.87	0.87	0.79	0.79	0.76
	AC-GAN	0.77	0.89	0.55	0.71	0.58	0.88	0.85	0.66	0.68	0.70
	FSC-GAN	0.78	0.87	0.60	0.58	0.49	0.51	0.61	0.48	0.38	0.41
	MFC-GAN	0.98	0.98	0.83	0.85	0.76	0.71	0.88	0.90	0.89	0.83
F1-score	Baseline	0.91	0.96	0.78	0.84	0.81	0.82	0.84	0.79	0.77	0.73
	SMOTE	0.96	0.97	0.87	0.94	0.89	0.93	0.93	0.88	0.89	0.86
	AC-GAN	0.87	0.94	0.71	0.83	0.73	0.94	0.92	0.80	0.81	0.80
	FSC-GAN	0.88	0.93	0.75	0.73	0.65	0.67	0.76	0.65	0.55	0.44
	MFC-GAN	0.99	0.99	0.91	0.91	0.87	0.83	0.93	0.94	0.94	0.90
G-Mean	Baseline	0.91	0.97	0.80	0.85	0.83	0.84	0.85	0.81	0.79	0.76
	SMOTE	0.96	0.97	0.87	0.94	0.90	0.93	0.94	0.89	0.89	0.87
	AC-GAN	0.88	0.95	0.74	0.84	0.76	0.94	0.92	0.82	0.83	0.83
	FSC-GAN	0.88	0.93	0.77	0.76	0.70	0.71	0.78	0.69	0.62	0.64
	MFC-GAN	0.99	0.99	0.91	0.92	0.87	0.84	0.94	0.95	0.94	0.91

Table 1: Results of SMOTE, AC-GAN, FSC-GAN and MFC-GAN classification performance on MNIST when each class is used as a minority.

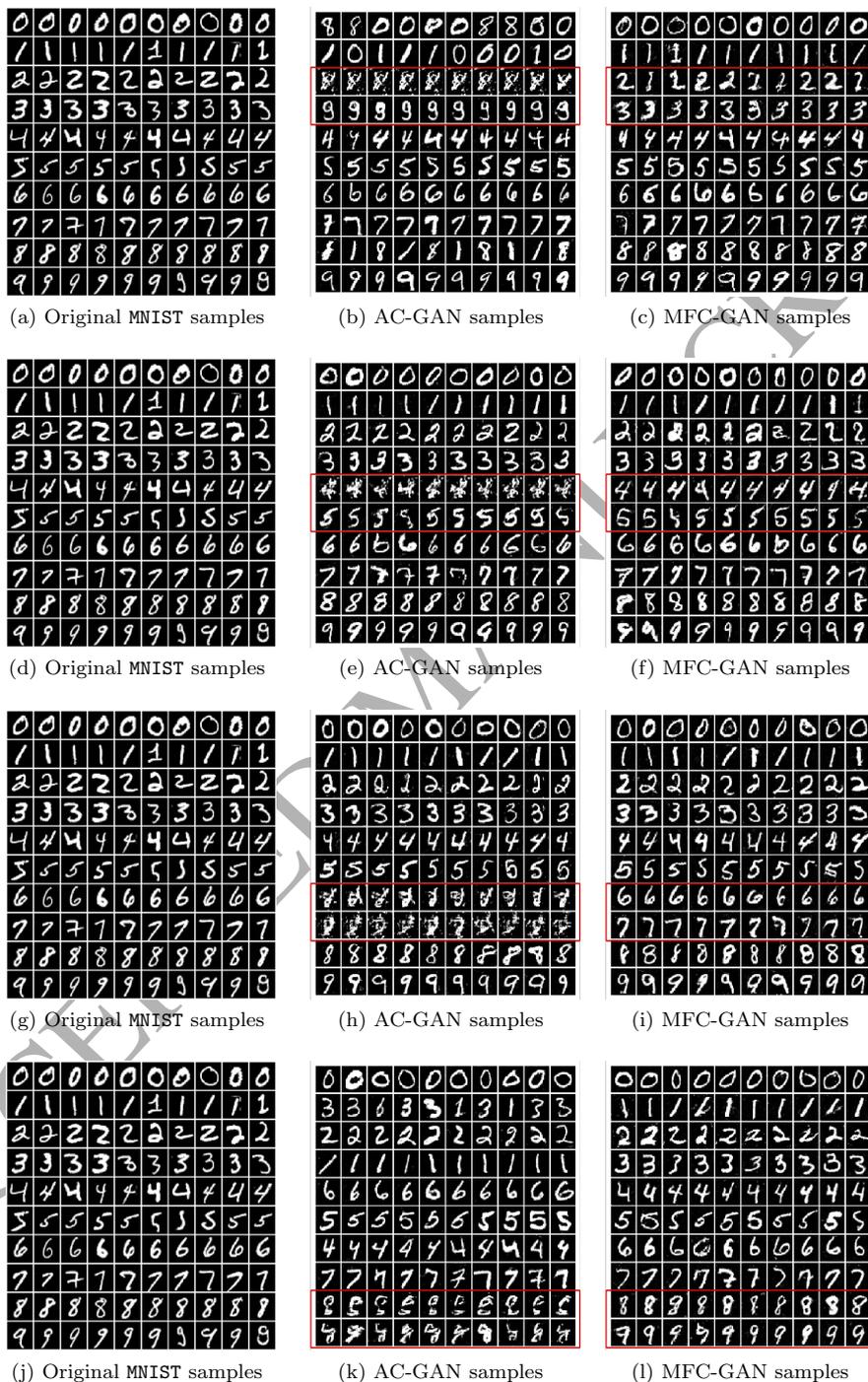


Figure 5: Minority class instances (highlighted in red) in different runs.

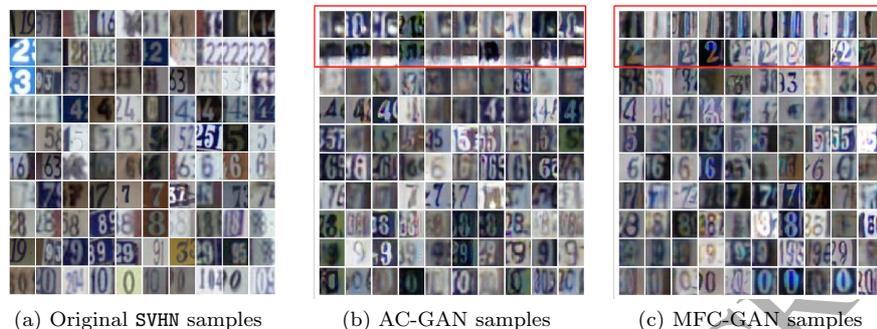


Figure 6: Original images (left) and generated images from AC-GAN and MFC-GAN, minority classes are highlighted in red rectangle

Metric	Model	G	K	Q	f	j	k	m	p	s	y
Sensitivity	Baseline	0.84	0.81	0.82	0.02	0.62	0.56	0.00	0.10	0.00	0.29
	SMOTE	0.82	0.73	0.80	0.25	0.84	0.58	0.23	0.38	0.01	0.48
	AC-GAN	0.77	0.76	0.87	0.14	0.57	0.57	0.00	0.21	0.00	0.18
	FSC-GAN	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	MFC-GAN	0.89	0.69	0.94	0.48	0.80	0.68	0.22	0.77	0.14	0.65
Specificity	Baseline	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	SMOTE	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	AC-GAN	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	FSC-GAN	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	MFC-GAN	1.00									
Accuracy	Baseline	0.92	0.90	0.91	0.51	0.81	0.78	0.50	0.55	0.50	0.65
	SMOTE	0.91	0.86	0.90	0.62	0.92	0.79	0.61	0.69	0.50	0.74
	AC-GAN	0.89	0.88	0.94	0.57	0.78	0.79	0.50	0.61	0.50	0.59
	FSC-GAN	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
	MFC-GAN	0.94	0.84	0.97	0.74	0.90	0.84	0.61	0.89	0.57	0.82
Precision	Baseline	0.91	0.64	0.91	0.43	0.72	0.79	0.00	0.55	0.00	0.53
	SMOTE	0.93	0.64	0.93	0.36	0.48	0.70	0.41	0.54	0.25	0.42
	AC-GAN	0.96	0.63	0.88	0.43	0.81	0.74	0.33	0.61	0.17	0.62
	FSC-GAN	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	MFC-GAN	0.80	0.63	0.61	0.36	0.50	0.61	0.40	0.36	0.13	0.33
Recall	Baseline	0.84	0.81	0.82	0.02	0.62	0.56	0.00	0.10	0.00	0.29
	SMOTE	0.82	0.73	0.80	0.25	0.84	0.58	0.23	0.38	0.01	0.48
	AC-GAN	0.77	0.76	0.87	0.14	0.57	0.57	0.00	0.21	0.00	0.18
	FSC-GAN	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	MFC-GAN	0.89	0.69	0.94	0.48	0.80	0.68	0.22	0.77	0.14	0.65
F1-score	Baseline	0.88	0.71	0.86	0.03	0.66	0.65	0.00	0.17	0.00	0.38
	SMOTE	0.87	0.68	0.86	0.29	0.62	0.64	0.29	0.45	0.01	0.45
	AC-GAN	0.86	0.69	0.88	0.21	0.67	0.65	0.00	0.32	0.00	0.28
	FSC-GAN	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	MFC-GAN	0.85	0.66	0.74	0.41	0.62	0.64	0.29	0.49	0.13	0.44
G-Mean	Baseline	0.92	0.90	0.90	0.12	0.78	0.75	0.00	0.32	0.00	0.54
	SMOTE	0.91	0.76	0.89	0.49	0.92	0.76	0.48	0.62	0.08	0.69
	AC-GAN	0.88	0.76	0.93	0.37	0.75	0.76	0.05	0.46	0.05	0.42
	FSC-GAN	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	MFC-GAN	0.94	0.83	0.97	0.69	0.90	0.83	0.47	0.88	0.37	0.80

Table 2: Sensitivity analysis of the classifier when using SMOTE, AC-GAN, FSC-GAN and MFC-GAN on ten E-MNIST minority classes.

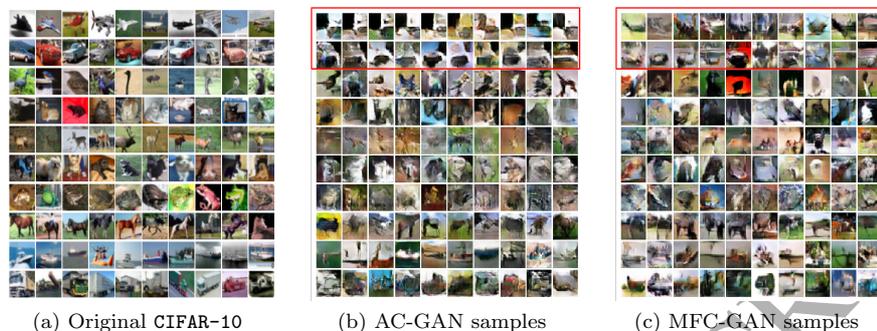


Figure 7: Original sample images(left) with AC-GAN and MFC-GAN generated samples (middle, right). Minority classes are highlighted in red

6. Discussion

Tables 1, 2 and 3 show that the CNN achieved better performances when it was trained on the MFC-GAN generated samples. Higher sensitivity, balanced accuracy and G-Mean demonstrate that the MFC-GAN model was able to generate samples from minority classes in a multi-classification problem. It has to be pointed out that all the figures in all tables have been rounded to the nearest two decimal points. Results also show that MFC-GAN out-performed SMOTE and AC-GAN on all SVHN & CIFAR-10 minority classes, and in 7 out of 10 E-MNIST & MNIST, minority classes. The fidelity and diversity of MFC-GAN minority samples made classification easier for the CNN. The diversity of generated samples indicates no sign of mode collapse in the model. Thus, with multiple fake classes, the GAN model was able to distinguish among classes better. A similar performance was recorded across all methods using the specificity, and this is reasonable as most classification models will accurately predict the majority class instances (tn).

FSC-GAN samples did not improve the classification in all experiments conducted as can be seen in Tables 1, 2 and 3. The results obtained showed that the classifier performed below the baseline when FSC-GAN samples were added to the training data. This is because FSC-GAN generated poor samples even when the number of classes is fairly balanced as shown in Figure 2. The other datasets are more challenging than MNIST and FSC-GAN goes into mode collapses when trained on the imbalanced datasets. The results indicate how negatively FSC-GAN is affected by the class-imbalanced problem.

AC-GAN model performed poorly on all the datasets in minority class image generation. This was evident by the below-average performance of the CNN when it was trained on AC-GAN samples. As can be seen in Figures 4, 5, 6 and 7, AC-GAN generated plausible majority class instances, however, the quality of generated minority class instances dropped significantly. In some cases, the model completely failed and became biased towards the majority class instances.

Metric	Model	Class 1	Class 2	Aeroplane	Automobile
Sensitivity	Baseline	0.01	0.00	0.07	0.04
	SMOTE	0.18	0.31	0.06	0.07
	ACGAN	0.00	0.02	0.07	0.05
	FSC-GAN	0.02	0.09	0.00	0.00
	MFC-GAN	0.51	0.68	0.07	0.08
specificity	Baseline	1.00	1.00	1.00	1.00
	SMOTE	1.00	1.00	1.00	1.00
	ACGAN	1.00	1.00	1.00	1.00
	FSC-GAN	1.00	1.00	1.00	1.00
	MFC-GAN	1.00	0.99	1.00	1.00
Accuracy	Baseline	0.50	0.52	0.53	0.52
	SMOTE	0.59	0.65	0.53	0.53
	ACGAN	0.50	0.51	0.53	0.52
	FSC-GAN	0.51	0.54	0.50	0.50
	MFC-GAN	0.75	0.83	0.54	0.54
Precision	Baseline	1.00	0.99	0.93	1.00
	SMOTE	0.99	1.00	0.97	0.98
	ACGAN	1.00	1.00	0.93	0.89
	FSC-GAN	0.99	0.99	1.00	1.00
	MFC-GAN	0.98	0.96	0.80	0.81
Recall	Baseline	0.01	0.05	0.07	0.04
	SMOTE	0.18	0.31	0.06	0.07
	ACGAN	0.00	0.02	0.07	0.05
	FSC-GAN	0.02	0.09	0.00	0.00
	MFC-GAN	0.51	0.68	0.07	0.08
F1-score	Baseline	0.02	0.09	0.12	0.08
	SMOTE	0.30	0.47	0.11	0.12
	ACGAN	0.00	0.03	0.12	0.09
	FSC-GAN	0.04	0.16	0.00	0.00
	MFC-GAN	0.67	0.79	0.14	0.14
G-Mean	Baseline	0.09	0.21	0.25	0.21
	SMOTE	0.42	0.56	0.24	0.25
	ACGAN	0.00	0.13	0.26	0.22
	FSC-GAN	0.14	0.30	0.00	0.00
	MFC-GAN	0.71	0.82	0.27	0.28

Table 3: SMOTE, AC-GAN, FSC-GAN and MFC-GAN performance on SVHN (Class 1 & Class 2) & CIFAR-10(Aeroplane & Automobile) minority classes.

This is consistent with the findings observed by [12]. For some specific classes a mode dropping in AC-GAN was observed, and the model generated the same image in all samples as can be seen in Figure 7b.

It was also observed from results that classification improvement was achieved when oversampling using SMOTE rather than augmenting with AC-GAN gen-

erated samples (Tables 1, 2 and 3). SMOTE achieved slightly better recall than MFC-GAN on two E-MNIST minority classes as seen in table 2. This is because E-MNIST has more samples in the minority class (with the smallest class having 1896 samples). However, on the other datasets, SMOTE didn't perform well when the number of minority class instances drops significantly. This also proves that MFC-GAN maintains good performance even with minimum number of samples in comparison with SMOTE and AC-GAN.

While good results have been obtained on MNIST, E-MNIST and SVHN, poor performances were recorded on CIFAR-10 by all models on minority class instances. AC-GAN model collapsed completely on CIFAR-10 while salient features required to distinguish samples effectively were not synthesized by MFC-GAN. These results might be attributed to the relatively small size of these images (i.e., 32×32 CIFAR-10 image patches) and the level of details within such tiny size. Although the samples generated by these models may look realistic, the characteristic features that will be vivid enough to train a classification model were missing. Increasing the number of minority samples from 50 to 100, 150, 200, 250 and 300 showed better but not significant improvement in performance. That said, as can be seen in Table 3, MFC-GAN produced slightly better performance amongst all these models.

Interestingly, poor results were obtained by all models for some specific minority classes. In particular, in the E-MNIST's minority classes m and s (Table 2). These minority classes were entirely missed by the baseline classifier, and very poor performance was reported using SMOTE, FSC-GAN and AC-GAN. MFC-GAN has also performed poorly in these classes. These results might be due to the similarity between some of these minority class instances and other majority class instance (i.e., class s is similar to classes 5, S , 2, z).

7. Conclusion

In this paper, a new augmentation method using Multiple Fake Class Generative Adversarial Networks (MFC-GAN) was presented and evaluated using four public datasets. We showed that MFC-GAN was capable of generating plausible samples of minority class instances. For evaluation, samples generated using our model were first added to the imbalanced datasets. Classification using Convolutional Neural Network was then carried out. Results showed that by augmenting the training set with MFC-GAN generated samples, performance improves across common metrics used for evaluating class-imbalanced datasets classification. Our method showed superior performance when compared with other common augmentation and oversampling techniques.

Future directions will include further evaluation and theoretical analysis of results on a higher resolution images. More specifically, it would be interesting to study the performance of the model under different settings where the number of minority class instances varies significantly. Other directions will include considering different models architectures such (i.e., ResNet).

References

References

- [1] G. Douzas, F. Bacao, Effective data generation for imbalanced learning using conditional generative adversarial networks, *Expert Systems with Applications* 91 (2018) 464–471.
- [2] B. Krawczyk, Learning from imbalanced data: open challenges and future directions, *Progress in Artificial Intelligence* 5 (4) (2016) 221–232.
- [3] A. Ali-Gombe, E. Elyan, C. Jayne, Fish classification in context of noisy images, in: *International Conference on Engineering Applications of Neural Networks*, 2017.
- [4] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: F. Pereira, C. J. C. Burges, L. Bottou, K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 25*, Curran Associates, Inc., 2012, pp. 1097–1105.
URL [http : / / papers . nips . cc / paper / 4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf](http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf)
- [5] H. Inoue, Data augmentation by pairing samples for images classification, *arXiv preprint arXiv:1801.02929*.
- [6] M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger, H. Greenspan, Synthetic data augmentation using gan for improved liver lesion classification, *arXiv preprint arXiv:1801.02385*.
- [7] A. FernáNdez, V. López, M. Galar, M. J. Del Jesus, F. Herrera, Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches, *Knowledge-based systems* 42 (2013) 97–110.
- [8] Q. Dong, S. Gong, X. Zhu, Class rectification hard mining for imbalanced deep learning.
- [9] T. Karras, T. Aila, S. Laine, J. Lehtinen, Progressive growing of gans for improved quality, stability, and variation, *arXiv preprint arXiv:1710.10196 ICLR2018*.
- [10] X. Zhu, Y. Liu, Z. Qin, Data augmentation in classification using gan, *arXiv preprint arXiv:1711.00648*.
- [11] A. Antoniou, A. Storkey, H. Edwards, Data augmentation generative adversarial networks, *arXiv preprint arXiv:1711.04340*.
- [12] G. Mariani, F. Scheidegger, R. Istrate, C. Bekas, C. Malossi, Bagan: Data augmentation with balancing gan, *arXiv preprint arXiv:1803.09655*.

- [13] C. Baur, S. Albarqouni, N. Navab, Melanogans: High resolution skin lesion synthesis with gans, arXiv preprint arXiv:1804.04338.
- [14] A. Odena, Semi-supervised learning with generative adversarial networks, arXiv preprint arXiv:1606.01583.
- [15] A. Odena, C. Olah, J. Shlens, Conditional image synthesis with auxiliary classifier gans, International conference on machine learning, page 2642-2651 70 (AUG 2017) 2642–2651.
- [16] A.-G. Adamu, E. Eyad, S. Yann, J. Chrisina, Few-shot classifier gan, in: Neural Networks (IJCNN), 2018 International Joint Conference on, IEEE, 2018.
- [17] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, Smote: Synthetic minority over-sampling technique, J. Artif. Int. Res. 16 (1) (2002) 321–357.
URL <http://dl.acm.org/citation.cfm?id=1622407.1622416>
- [18] S. Wang, W. Liu, J. Wu, L. Cao, Q. Meng, P. J. Kennedy, Training deep neural networks on imbalanced data sets, in: Neural Networks (IJCNN), 2016 International Joint Conference on, IEEE, 2016, pp. 4368–4374.
- [19] M. Buda, A. Maki, M. A. Mazurowski, A systematic study of the class imbalance problem in convolutional neural networks, arXiv preprint arXiv:1710.05381.
- [20] C. Huang, Y. Li, C. Change Loy, X. Tang, Learning deep representation for imbalanced classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5375–5384.
- [21] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, T. Brox, Discriminative unsupervised feature learning with convolutional neural networks, in: Advances in Neural Information Processing Systems, 2014, pp. 766–774.
- [22] H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, arXiv preprint arXiv:1710.09412.
- [23] M. Mirza, S. Osindero, Conditional generative adversarial nets, arXiv preprint arXiv:1411.1784.
- [24] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in neural information processing systems, 2014, pp. 2672–2680.
- [25] L. Wan, J. Wan, Y. Jin, Z. Tan, S. Z. Li, et al., Fine-grained multi-attribute adversarial learning for face generation of age, gender and ethnicity (2018).
- [26] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, arXiv preprint arXiv:1511.06434.

- [27] E. L. Denton, S. Chintala, R. Fergus, et al., Deep generative image models using a laplacian pyramid of adversarial networks, in: *Advances in neural information processing systems*, 2015, pp. 1486–1494.
- [28] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [29] S. Gurumurthy, R. K. Sarvadevabhatla, V. B. Radhakrishnan, Deligan: Generative adversarial networks for diverse and limited data, in: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 1, 2017.
- [30] T. Miyato, T. Kataoka, M. Koyama, Y. Yoshida, Spectral normalization for generative adversarial networks, arXiv preprint arXiv:1802.05957 and ICLR2018.
- [31] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, L. D. Jackel, Handwritten digit recognition with a back-propagation network, in: *Advances in neural information processing systems*, 1990, pp. 396–404.
- [32] G. Cohen, S. Afshar, J. Tapson, A. van Schaik, Emnist: Extending mnist to handwritten letters, in: *Neural Networks (IJCNN), 2017 International Joint Conference on*, IEEE, 2017, pp. 2921–2926.
- [33] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Y. Ng, Reading digits in natural images with unsupervised feature learning, in: *NIPS workshop on deep learning and unsupervised feature learning*, Vol. 2011, 2011, p. 5.
- [34] A. Krizhevsky, V. Nair, G. Hinton, Cifar-10 (canadian institute for advanced research).
URL <http://www.cs.toronto.edu/~kriz/cifar.html>
- [35] M. D. Zeiler, Adadelta: an adaptive learning rate method, arXiv preprint arXiv:1212.5701.



Adamu Ali-Gombe obtained his first degree in Computer Science from Abubakar Tafawa Balewa University Bauchi Nigeria in 2009. In 2013 Mr. Ali-Gombe received his Masters in Science from Africa University of Science and

Technology, Abuja Nigeria. Currently, his a PhD student at the School of Computing Science and Digital Media at Robert Gordon University. His main research interests are in Generative Adversarial Neural Networks, object detection and classification, and learning from imbalanced datasets.



Dr. Eyad Elyan obtained his first degree in Computer Science in 1999 from Al Quds University. He then received his MSc in Software Engineering in 2004 from the University of Bradford. In 2008, Dr. Elyan received his PhD from Bradford University for his work on modelling and representation of 3D Face Images Using Elliptic Partial Differential Equations. Eyad is a Fellow member of the Higher Education Academy and currently is a Reader at the School of Computing Science and Digital Media at Robert Gordon University. His research is primarily focused on learning from imbalanced datasets using advanced methods such as deep learning and ensemble learning.