2021 43rd Annual International Conference of the
IEEE Engineering in Medicine & Biology Society (EMBC)
Oct 31 - Nov 4, 2021. Virtual Conference

Multi-class Generative Adversarial Networks: Improving One-class Classification of Pneumonia Using Limited Labeled Data

Saman Motamed[1] and Farzad Khalvati[2]

Abstract— **From generating never-before-seen images to domain adaptation, applications of Generative Adversarial Networks (GANs) spread wide in the domain of vision and graphics problems. With the remarkable ability of GANs in learning the distribution and generating images of a particular class, they have been used for semi-supervised disease detection in medical images such as COVID-19 and Pneumonia in X-rays. However, the challenge is that if two classes of images share similar characteristics, the GAN might learn to generalize and hinder the classification of the two classes. In this paper, first we use MNIST and Fashion-MNIST datasets that are easy to visually inspect, to illustrate how similar images cause the GAN to generalize, leading to the poor classification of images. We then show how this generalization can misclassify pneumonia X-rays as healthy cases when using GANs for semi-supervised detection of pneumonia. We propose a modification to the traditional training of GANs that, using small sets of labeled data, allows for improved classification in similar classes of images in a semi-supervised learning framework.**

I. Introduction

Generative Adversarial Networks [1] is one of the most exciting inventions in machine learning in the past decade. While applications of GANs spread wide in the field of computer vision, image classification using GANs is relatively unexplored. One of the early uses of GANs in image classification was detecting anomalies in images, first introduced by Schlegl *et al.* [2] to detect and identify anomalies in the form of retinal fluid or hyper-reflective foci in optical coherence tomography (OCT) images of the retina. By defining a variation score $V(x)$ (eq. 2), their proposed Anomaly Detection GAN (AnoGAN) captured the characteristic and visual differences of two images; one generated by the GAN and one real image. The idea was to, for instance, train the GAN on only healthy images. When GAN is trained, the generator can generate images similar to those in the healthy image class. During the test phase, the variation score $V(x)$ must be low if the test image is healthy and GAN's generator (G) can generate a similar image to that of the healthy image. If the test image is not healthy and contains anomalies, $V(x)$ would be larger, and the generated image would look visually different than the real image containing anomalies.

Recently, RANDGAN, a Generative Adversarial Network was proposed for binary classification of COVID-19 negative (healthy and viral pneumonia) and COVID-19 positive chest X-ray images, without the need to use any COVID-19 positive images for training the model [3]. By training two GANs, one on normal X-rays and one on pneumonia X-rays, the authors calculated a variation score for COVID-19 negative (normal and pneumonia) images [3] where a higher variation score for image x increased the probability of the image belonging to COVID-19 positive category (unknown class) while a lower variation score increased the probability of image x belonging to one of the known classes (normal and pneumonia). In this work, using a similar approach to RANDGAN [3] and AnoGAN [2], we used DCGAN [4] for one-class classification. By training the GAN on the class of known (C1) images with labels, we aimed to detect the images of the unknown class (C2).

We observed that, in some instances, training a GAN on images of class C1 generated not only low variation scores for test images of the same class (expected behaviour), but also low scores for test images of class C2 (unexpected behaviour), hindering the ability to classify C1 from C2. We hypothesized the reason to be the ability of the GAN's generator G, being trained on C1 images, generalizing to learn and generate images that visually look similar to C2 images. In this work, we carried out multiple experiments using different datasets to understand how visually similar images affect GAN-based image classification's performance. We propose *MCGAN*, a GAN-based multi-class classifier, to overcome the challenge of classifying visually similar images using GANs. By using available labeled images from both classes in training the MCGAN, we force G to not generalize in a way that can generate similar images to images of other classes. We used images from the MNIST and Fashion-MNIST datasets, which are visually easy to inspect, in order to understand the generalization problem of GANs for classifying similar classes of images. We then showed how the generalization problem can hinder classification of more challenging problem, such as detection on pneumonia in X-rays, where inspection of images is more challenging non-experts.

II. Generative Adversarial Networks

A GAN is a deep learning model comprised of two main parts; Generator (G) and Discriminator (D). G can be seen as an art forger that tries to reproduce artwork and pass it as the original. D, on the other hand, acts as an art authentication expert that tries to tell apart real from forged art. Successful training of a GAN is a battle between G and D where if successful, G generates realistic images and D is not able to

[1]S. Motamed is a student at the Institute of Medical Science, University of Toronto, the Hospital for Sick Children, Toronto, Canada sam.motamed@mail.utoronto.ca

[2]F. Khalvati is with the Department of Medical Imaging and Mechanical and Industrial Engineering and Institute of Medical Science, University of Toronto, The Hospital for Sick Children, Toronto, Canada farzad.khalvati@utoronto.ca

tell the difference between G's generated images compared to real images. G takes as input a random Gaussian noise vector and generates images through transposed convolution operations. D is trained to distinguish the real images (x) from generated fake images $(G(z))$. Optimization of D and G can be thought of as the following game of minimax [1] with the value function $V(G, D)$:

$$\min_G \max_D V(D, G) = E_{x \sim P_{data(x)}}[\log D(x)] + E_{z \sim P_{z(z)}}[\log(1 - D(G(z)))] \quad (1)$$

During training, G is trained to minimize D's ability to distinguish between real and generated images, while D is trying to maximize the probability of assigning "real" label to real training images and "fake" label to the generated images from G. The Generator improves at generating more realistic images while Discriminator gets better at correctly identifying between real and generated images. Today, when the term GAN is used, the Deep Convolution GAN (DCGAN) [4] is the architecture that it refers to.

A. Multi-class GAN

The goal of the proposed GAN-based multi-class (MCGAN) classifier is to distinguish two classes of data (C1, C2) from one another, while there are labels available for one class (C1) and limited labeled data for the other class (C2). A traditional GAN's (DCGAN, AnoGAN, etc.) discriminator takes as input the generator's output (labeled *Fake*) and a real image (labeled *Real*). This forces the generator to learn the distribution of the images from the real class. If the images of C1 and C2 shares similar characteristics, training the GAN on the images of C1 could cause G to learn and generalize well enough, leading to generating similar images to C2 and hence, hindering the classification of the two classes (C1 vs. C2). To overcome this challenge, we feed a third input to the discriminator; available labeled images of C2.

While these are real images from C2, we label them as *Fake*. This forces the generator not to learn to generalize to this similar class (C2) while learning the characteristics of C1. When G generates an image that could pass as belonging to C2, the discriminator flags it as a fake image, and G re-evaluates its learning at those stances. Figure 1 shows the architecture of Multi-class GAN (MCGAN).

B. Variation Score

The Variation score $V(x)$ for the query image x, proposed by Schlegl *et al.* [2], is defined as;

$$V(x) = (1 - \lambda) \times \mathcal{L}_R(z) + \lambda \times \mathcal{L}_D(z) \quad (2)$$

where $\mathcal{L}_R(z)$ (eq. 3) and $\mathcal{L}_D(z)$ (eq. 4) are the residual and discriminator loss respectively that enforce visual and image characteristic similarity between real image x and generated image $G(z)$. The discriminator loss captures image characteristics using the output of an intermediate layer of the discriminator, $f(.)$, making the discriminator act as an image encoder. Residual loss is the pixel-wise difference between image x and $G(z)$.

$$\mathcal{L}_R(z) = \sum |x - G(z)| \quad (3)$$

$$\mathcal{L}_D(z) = \sum |f(x) - f(G(z)| \quad (4)$$

Before calculating V(x) during test, a point z_i has to be found through back-propagation that tries to generate an image as similar as possible to image x. The loss function used to find z_i is based on residual and discriminator loss defined below.

$$\mathcal{L}(z_i) = (1 - \lambda) \times \mathcal{L}_R(z_i) + \lambda \times \mathcal{L}_D(z_i) \quad (5)$$

λ adjusts the weighted sum of the overall loss and variation score. We used $\lambda = 0.2$ to train our proposed MCGAN and AnoGAN [2]. Both architectures were trained with the same initial conditions for performance comparison.

III. Datasets

We used images from three different datasets. MNIST [5] dataset that contains 60,000 training images of handwritten digits and 10,000 test images. Fashion-MNIST [6] is a dataset of Zalando's article images—consisting of a training set of 60,000 examples and a test set of 10,000 examples. COVIDx dataset [7] that contains healthy (8,066), pneumonia (5,289) and COVID-19 (589) diagnosed X-ray images of the lung. We used the normal and pneumonia classes of images from the COVIDx dataset for our experiments and did not use the COVID-19 images. All gray-scale images were resized to 128×128 pixels, with pixel intensities scaled to -1 to 1.

IV. Experiments

A. MNIST and Fashion-MNIST

The purpose of using MNIST and Fashion MNIST datasets is to illustrate how similar images cause the GAN to generalize, leading to the poor classification of images. To pick a subset of similar classes from MNIST and Fashion-MNIST (F-MNIST) datasets that could cause generalization in GANs, we used metric learning [8]. The goal of metric learning is to train models that can embed inputs into a high-dimensional space such that "similar" inputs are located close to each other. To bring images from the same class closer to each other via the embedding, the training data was constructed as randomly selected pairs of images from each class matched to the label of that class, instead of traditional *(X,y)* pairs where y is the label for corresponding X as singular images of each class. By embedding the images using a shallow three-layer CNN, we computed the similarity between the image pairs by calculating the cosine similarity of the embeddings. We used these similarities as logits for a softmax. This moves the pairs of images from the same class closer together. After the training was complete, we sampled 10 examples from each of the 10 classes, and considered their near neighbours as a form of prediction; that is, the example and its near neighbours share the same class. This is visualized as a confusion matrix shown in Figure 2. The numbers that lie on the diagonal represent the correct classifications and the numbers off the diagonal represent

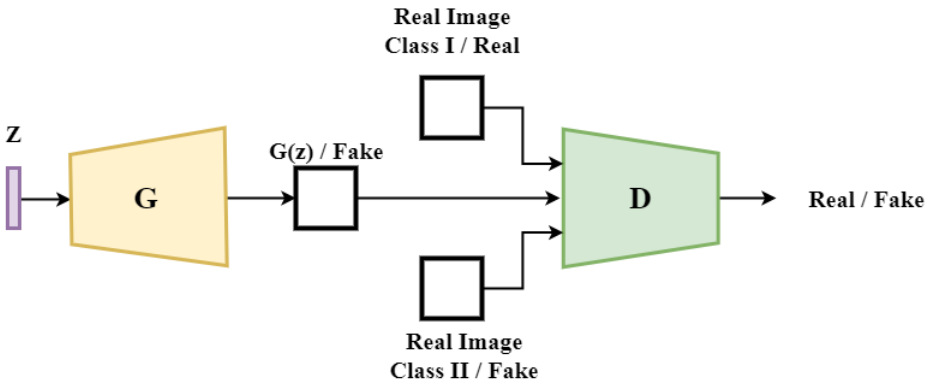

Fig. 1. Multi-class Generative Adversarial Network

the wrong labels that were misclassified as the true label. We intentionally used a shallow three-layer CNN to enforce some misclassification, as achieving near-perfect results in classifying datasets such as MNIST using CNNs is easy. Using the information from Figure 2, we picked the class pairs (9, 4) and (8, 3) from the MNIST dataset and (Coat, Shirt), (Coat, Pullover), and (Boot, Sandal) from F-MNIST dataset. For each pair of similar classes of images, we trained one AnoGAN and our proposed MCGAN. While AnoGAN uses only labels of one class(C1), MCGAN needs labels for both classes (C1 and C2). For the pairs of similar images, while training MCGAN, we used all the labels available for both classes. In turn, this makes the MCGAN a supervised model. Later on, for the COVIDx dataset, we explore a more limited set of labeled images from second class (C2).

B. X-ray Data

We used the healthy and pneumonia X-rays from the Chest X-ray dataset [7] dataset in order to learn the classification of the two classes of images. While training an AnoGAN, we used images of one class with corresponding labels (normal or pneumonia). For training MCGAN, we used images of both labels; the class of images we want to learn to generate (C1) was used with all the labels while partial images from the class of similar images we do not want the GAN to (C2) generalize to was used. Through multiple instances of training the MCGAN, we studied the effects of available labeled data size from class C2 images on classification of C1 and C2.

C. Competing Methods

Ruff *et. al* proposed a Deep One-class classification model (Deep SVDD) [9] that outperformed shallow and deep semi-supervised anomaly detection models at the time, including AnoGAN (DCGAN and AnoGAN follow the same architecture and can be used interchangeably). We compare our Inception-GAN against these models (Isolation Forest, One-class SVM, DCGAN and Deep SVDD) as baselines.

D. Shallow Baselines

We followed the same implementation details of the shallow models as used in Ruff *et. al*'s Deep SVDD study. (i) **One-class SVM** (OC-SVM) [10] finds a maximum margin hyper-plane that best separates the mapped data from the origin. (ii) **Isolation Forest** [11] (IF) isolates observations by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of the selected feature. We set the number of trees to t = 100 and the sub-sampling size to 256, as recommended in the original work

E. Deep Baselines

Our Inception-GAN is compared with two deep approaches. (i) Ruff *et. al*'s Deep SVDD showed improved accuracy of one class classification in a framework where one class from MNIST [5] and CIFAR-10 [12] was kept as the known image, and the rest of the classes were treated as the anomaly. Deep SVDD learns a neural network transformation from inputs into a hypersphere characterized by center c and radius R of minimum volume. The idea is that this allows for the known (pneumonia / healthy) class of images to fall into the hypersphere and the unknown (healthy / pneumonia) class to fall outside of the hypersphere. (ii) **DCGAN / AnoGAN** is trained as the base GAN benchmark for the task of pneumonia detection

V. Results

A. MNIST and Fashion-MNIST

We calculated variation scores for both DCGAN and MCGAN for each pair of similar classes of images. Lower variation scores would translate to the test image having more probability of belonging to the class of images the GAN was trained to generate images of, while a larger variation score decreased this probability. We calculated the area under the ROC curve (AUC) of each model. Table I shows the AUC for one-class classification for each data pair and model. For each pair (C1, C2) for DCGAN, the first AUC is the result of training GAN on C1 images and the second AUC is the result of GAN training on C2 images. For MCGAN, the first AUC is the result of using all of C1 images with *Real* labels and all of C2 images with *Fake*, while the second AUC is for vice versa. In both scenarios of using MCGAN to generate images of class C1 or C2, MCGAN outperformed DCGAN in one-class classification. MCGAN in this scenario used all labels

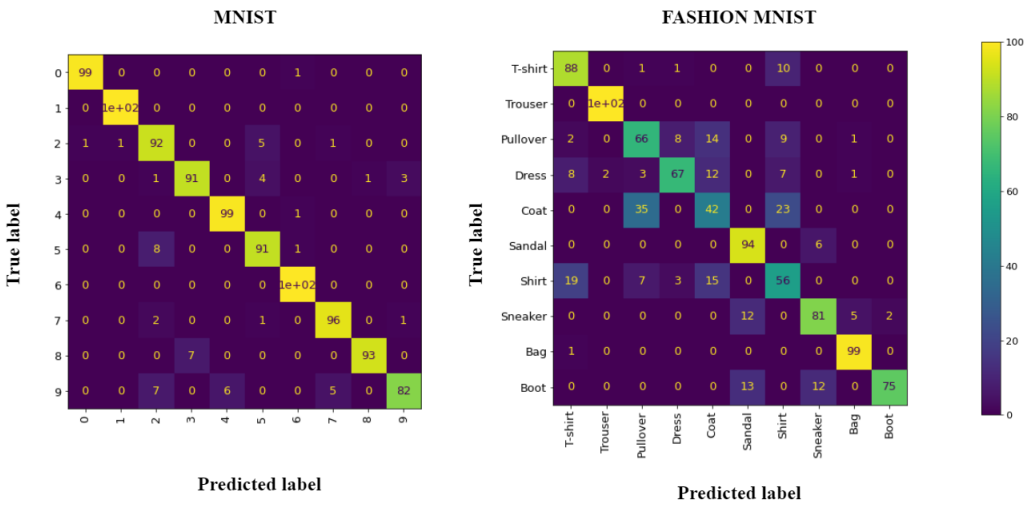

Fig. 2. Confusion matrix of MNIST and F-MNIST embeddings

for both classes. This is solely to demonstrate the problem of one-class classification of similar classes of images using GANs. When sufficient labeled data is available, supervised classifiers traditionally yield better performance. Figure 3 shows how a DCGAN, trained on images from C1, can generate images similar to C2 if both classes have similar characteristics in a way that while learning to generate C1 images, G learns to produce similar images to C2. MCGAN, on the other hand, forces the generator to avoid this generalization, which helps the classification task (table I). The first row of Figure 3 shows the test image which DCGAN and MCGAN's generators tried to generate a similar image of (4 and 3) while having been trained to generate images of class 9 and 8 respectively. The DCGAN's output looks closer to the test image compared to MCGAN. The second row result from the two GANs trained to generate images of class Boot and generate images similar to test images from class Sneaker and Sandal respectively. While DCGAN learns to generate images similar to Sneaker and Sandal while learning from images from class Boot, our MCGAN succeeds in not making this generalization.

B. Detection of Healthy and Pneumonia X-rays

To detect pneumonia from healthy X-rays using DCGAN, we trained one DCGAN on normal images and one on pneumonia images. Likewise, we trained IF, OC-SVM and Deep SVDD networks once using normal images and once using pneumonia images and tested the performance of the models on 1,000 randomly selected images (500 normal and 500 pneumonia) from the COVIDx dataset. Table II shows the AUC for classification of normal and pneumonia X-rays using different methods, once trained using normal images only and once trained on pneumonia images.

The DCGAN's poor performance in classifying the images, when trained on normal images, suggests this could be due to the generalization problem. The reason for success of the GAN trained on pneumonia images and its failure when trained on normal images can be deducted from Figure 4. When the GAN is trained on normal images, the generator learns to generate images that in the beginning look noisy and as training progresses, they look more healthy-like. Figure 4's green circle shows how the noisy lung image, when learning to generate normal images, could be classified as pneumonia. Whereas when the GAN is trained to generate pneumonia images (Figure 4 - yellow circle), the noisy image when training to generate pneumonia images, cannot be identified as a healthy lung.

We trained different instances of MCGAN, using all labeled normal training images and randomly chose 50, 100, 200, 400, 800, 1600, 3200 and finally all (4,789) of labeled pneumonia training images as the secondary class (C2) to train MCGAN. first row of Table III shows the effect that different sizes of labeled images from the pneumonia class (C2) have on training the MCGAN, with a fixed number (7,566) of normal (C1) images. With 400 labeled pneumonia images, the model achieves the same accuracy as Deep SVDD trained only on normal images (AUC: 0.64). With 800 labeled images, MCGAN outperforms the one-class classifiers and keeps improving with increase in the number of labeled pneumonia images. In finding success when training the DCGAN on pneumonia images, we experimented with training instances of the DCGAN only on the same number of labeled pneumonia images as MCGAN used (50, 100, 200, 400, 800, 1,600, 3,200 and 4,789) to understand whether DCGAN with limited number of labels can outperform MCGAN which uses the same number of labeled pneumonia images and all normal labeled images. Second row of Table III shows the AUC achieved by DCGAN, trained only on pneumonia images, in detecting pneumonia and normal X-rays. DCGAN failed to converge and generate images of pneumonia when trained on 800 images and less. With 1,600 labeled pneumonia images, DCGAN and MCGAN achieve the same performance. With larger number of labeled pneumonia data, DCGAN outperforms MCGAN without the need for any normal labeled images. With limited pneumonia (800 images and less) and sufficient normal data however,

	MNIST (8 / 3)	MNIST (9 / 4)	F-MNIST (Boot / Sandal)	F-MNIST (Coat / Shirt)	F-MNIST (Coat / Pullover)
DCGAN	0.87 / 0.9	0.88 / 0.78	0.79 / 0.72	0.68 / 0.54	0.67 / 0.32
MCGAN	**0.92 / 0.95**	**0.91 / 0.84**	**0.87 / 0.82**	**0.79 / 0.74**	**0.77 / 0.71**

TABLE I

CLASSIFICATION AUC OF DCGAN USING ONE LABEL ONLY AND MCGAN USING BOTH LABELS

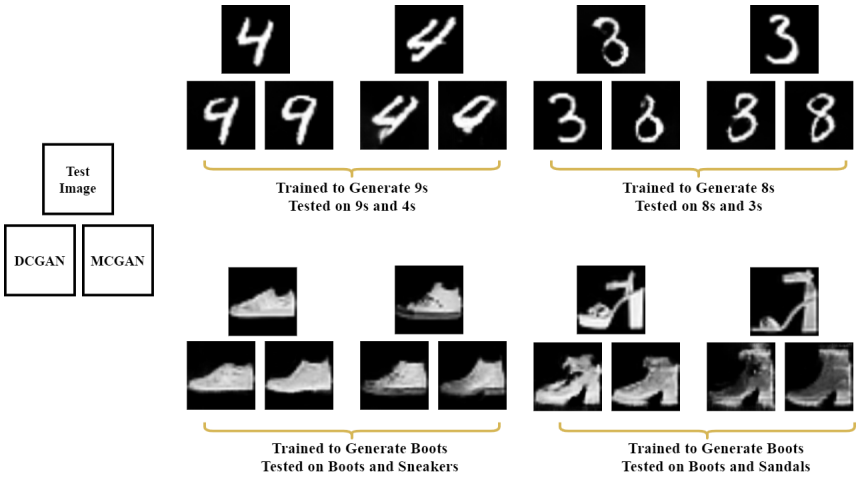

Fig. 3. DCGAN and MCGAN generated images

	IF	OC-SVM	Deep SVDD	DCGAN
Trained on normal	0.51	0.53	**0.64**	0.46
Trained on pneumonia	0.49	0.54	0.69	**0.76**

TABLE II

CLASSIFICATION AUC OF DIFFERENT MODELS, TRAINED ONCE ON ONLY NORMAL AND ONCE ON ONLY PNEUMONIA IMAGES

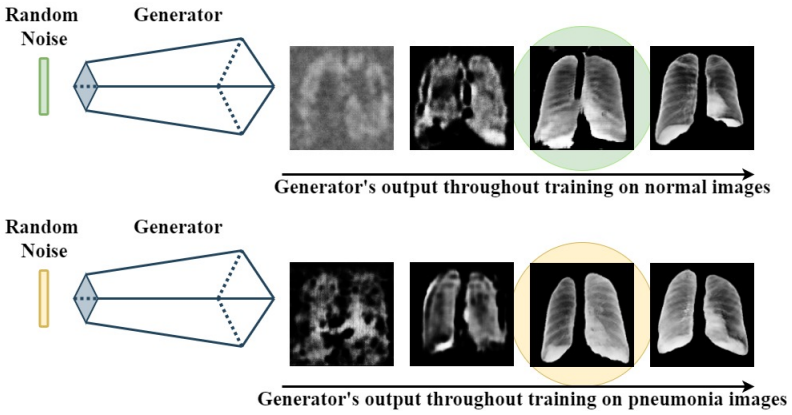

Fig. 4. DCGAN generator output throughout training

MCGAN achieves he best classification performance.

VI. DISCUSSION

In classification settings where we do not have enough labeled images for a class, semi-supervised modes of training that do not require images of that class to train are of value. While GANs can be used to classify images, we showed that in some settings where labeled images share similar characteristics, the generalization ability of GANs can hinder the performance of classification. Using images from MNIST, Fashion MNIST and Chest X-rays, we showed how, for instance, a GAN trained to generate images of healthy chest X-rays can also generate images that are similar to X-rays with pneumonia. To use GANs in classifying

#pneumonia	50	100	200	400	800	1,600	3,200	all (4,789)
MCGAN'S AUC	0.54	0.57	0.62	0.64	0.67	0.68	0.69	**0.71**
DCGAN'S AUC	Fail	Fail	Fail	Fail	Fail	0.68	0.72	**0.76**

TABLE III

CLASSIFICATION AUC FOR DCGAN AND MCGAN MCGAN TRAINED ON ALL NORMAL IMAGES AND DIFFERENT NUMBER OF PNEUMONIA IMAGES WHILE DCGAN TRAINED ONLY ON PNEUMONIA IMAGES

normal from pneumonia X-rays, this generalization would result in not only low variation scores for healthy X-rays, but also for pneumonia X-rays. We proposed MCGAN, which used both classes in training the GAN's discriminator. By labeling the No as *fake*, we guided the generator to not generate images that can identify as having pneumonia while learning to generate images of normal X-rays. The labeling of the similar class of images as *fake* improved classification of similar classes from one another.

The goal for this study was not to achieve the state-of-the-art classification results using semi-supervised methods on the the three datasets, rather identifying a potential problem when using GANs for classification tasks and using a simple GAN architecture and showing how the proposed modification in training the discriminator can improve classification in settings where over-generalization is possible. With development of more complicated GAN architectures, such as RANDGAN [3], this modification can further improve the accuracy of the models.

VII. CONCLUSION

In this work, we demonstrated how GANs could learn to generalize to different classes of images if they share similar characteristics with the class of training images. This generalization can hinder the ability of GANs for the task of image classification. We proposed using available labeled images in training the discriminator to penalize the generalization. The multi-class discriminator training showed improved accuracy of semi-supervised image classification.

VIII. ACKNOWLEDGEMENTS

This research was funded by Chair in Medical Imaging and Artificial Intelligence funding, a joint Hospital-University Chair between the University of Toronto, The Hospital for Sick Children, and the SickKids Foundation.

REFERENCES

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[2] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International conference on information processing in medical imaging*, pages 146–157. Springer, 2017.

[3] Saman Motamed, Patrik Rogalla, and Farzad Khalvati. Randgan: randomized generative adversarial network for detection of covid-19 in chest x-ray. *Scientific Reports*, 11(1):1–10, 2021.

[4] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[5] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[6] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[7] Linda Wang and Alexander Wong. Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. *arXiv preprint arXiv:2003.09871*, 2020.

[8] Brian Kulis et al. Metric learning: A survey. *Foundations and trends in machine learning*, 5(4):287–364, 2012.

[9] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International conference on machine learning*, pages 4393–4402. PMLR, 2018.

[10] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

[11] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008.

[12] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.