OPEN SET RECOGNITION BY REGULARISING CLASSIFIER WITH FAKE DATA GENERATED BY GENERATIVE ADVERSARIAL NETWORKS

Inhyuk Jo[⋆] *Jungtaek Kim*[⋆] *Hyohyeong Kang*[†] *Yong-Deok Kim*[†] *Seungjin Choi*[⋆]

[⋆] Department of Computer Science and Engineering, POSTECH, Korea
{ihcho, jtkim, seungjin}@postech.ac.kr
[†]Software R&D Center, Device Solutions, Samsung Electronics, Korea
{hh1208.kang, yd.mlg.kim}@samsung.com

ABSTRACT

We present a new method to generate fake data in unknown classes in generative adversarial networks (GANs) framework. The generator in GANs is trained to generate somewhat similar to data in known classes but the different one by modelling noisy distribution on feature space of a classifier using proposed marginal denoising autoencoder. The generated data are treated as fake instances in unknown classes and given to the classifier to make it be robust to the real unknown classes. Our results show that synthetic data can act as fake unknown classes and keep down the certainty of the classifier on real unknown classes meanwhile the classification capability of known classes is not degenerated, even improved.

Index Terms— Generative adversarial networks, Denoising autoencoder, Open set recognition, Feature matching

1. INTRODUCTION

Deep learning has been received much attention in various fields, and growing faster recently. Among varied machine learning tasks, image classification task is one of the well-known problems in computer vision and addressed in numerous ways by many researchers [1, 2, 3]. Some networks were even much deeper or wider than previous ones and achieved human-level performance [4, 5]. All those models are trained to classify given data in known classes (positive data) as a pre-defined known class. Gathering all categories of data, however, is not possible in practice. The only data we have are limited number of categories and tremendous other categories are out there. The difficulty arises when data in unknown classes (negative data) that is not seen during training is given to the model because it just assigns given negative data to a particular known class even if the data does not belong to any of the known classes.

The task of identifying unknown classes can be addressed by novelty detection or anomaly detection. Although many researches have been presented to resolve novelty detection and well summarised in [6], it tends to differentiate unknown classes from known classes while ignoring class labels.

Open set recognition problem was formalised as open set risk minimisation [7] but its nature is rather straightforward. In open set recognition, the model should have abilities to classify known classes and to distinguish unknown classes from known classes at the same time. To address it, additional novelty detection model could be employed or one could introduce uncertainty measure (score) that how much the model is confident on its prediction. Several choices are possible for uncertainty measure such as entropy of the model's prediction [8], maximum of logits (the values before the softmax function in the last layer), and sophisticated score analysis [9, 10]. Whatever uncertainty measure was employed, even the simplest one, we would be able to detect unknown classes by explicitly maximising uncertainty of generated data so-called regularisation, provided that generating fake data in unknown classes using only known classes is feasible. If we could design such fake negative data generator, additional models or complicated score analysis to detect unknown classes would be no need anymore. The only thing we need is a classifier that simultaneously classify known classes and detect unknown classes.

Generative adversarial networks (GANs) [11] seems the most notable deep generative models these days. In GANs, the generator and the discriminator are trained adversarially. It generates sharp and convincing realistic data, although the networks have some problems including difficulty balancing between the generator and the discriminator, lack of monitoring convergence measure, and ignoring few modes. If we trained the generator in GANs to generate fake negative data, the generated data could be used as augmented data for regularisation of the classifier.

In this paper, we proposed a new objective function for the generator in GANs to generate synthetic negative data using marginal denoising autoencoder that models the noisy distributions of positive data on the feature space of the classifier. The generator trained to match the noisy distribution could generate fake negative data which were served effective augmented data for the classifier.

2. RELATED WORKS

Open set recognition and novelty detection both seek to find a way that differentiates unknown classes from known classes. [9, 10] analysed a score based on the extreme value theorem (EVT) by fitting Weibull distribution per instances and classes respectively but it requires lots of classes [9] or lots of instances of each class [10] to apply EVT. [12] trained null projection matrix to project data in the same class to a sole point by overcoming a drawback that is the number of training data should be smaller than the dimension of feature vector using kernel trick. [13] suggested that regularisation technique called dropout is a Bayesian approximation and showed that model uncertainty is obtained easily compared to Bayesian neural networks. [8] proposed that ensemble of models with adversarial training leads to lower classification error and a way of measuring uncertainty. [14] showed that temperature scaling which is simply scaling logits after training can improve distinguishing performance of unknown classes. [15] have utilised unlabelled data that is easily collected to find boundary that minimise empirical, structure and augment risk but we employed synthetic data and minimise cross

entropy of positive data and minus entropy of synthetic data.

Many variants of GANs have been introduced. [16] proposed deep convolutional generative adversarial network (DCGAN) which became guideline for other convolution based GANs. [17, 18, 19, 20, 21] addressed the instability of GANs in various ways. Recently, AnoGAN was proposed to detect anomalies on medical imaging data in unsupervised fashion [22]. They proposed two types of scores, residual score and discrimination score, and combine them to detect anomalies, yet they rely on vanilla GANs. In contrast, our work is based on supervised GANs and the generator is trained to generate fake negative data which regularises the classifier.

3. BACKGROUND

3.1. GANs and semi-supervised GANs

GANs are relatively new frameworks consisting of two networks: the discriminator D and the generator G. D tries to distinguish between real data and generated data, and G tries to generate convincing data to fool D. The minimax objective function for GANs is formulated as

$$\min_G \max_D \quad \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}\left[\log D\left(\mathbf{x}\right)\right] + \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}\left[\log\left(1 - D\left(G\left(\mathbf{z}\right)\right)\right)\right] \tag{1}$$

where p_{data} is the data distribution and $p_{\mathbf{z}}$ is rather simple prior distribution such as uniform or normal distribution.

[23] showed that using class labels improves the generated data qualitatively and quantitatively. They proposed substituting the discriminator D which merely distinguishes real data from generated data with the classifier C which classify real data as one of known classes. For labelled data, they trained classifier with cross entropy. For unlabelled data, they assigned generated class as $K+1$ where K is the number of classes in labelled data and train C and G adversarially as vanilla GANs.

$$\min_C \max_G \quad -\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}\left[\log\left(1 - p_C\left(y = K+1|\mathbf{x}\right)\right)\right] - \mathbb{E}_{\mathbf{x}\sim p_G(\mathbf{x})}\left[\log p_C\left(y = K+1|\mathbf{x}\right)\right] \tag{2}$$

where $p_C\left(y|\mathbf{x}\right)$ is membership probability of C and $p_G\left(\mathbf{x}\right)$ is generated data distribution of G. Note that $D\left(\mathbf{x}\right)$ in vanilla GANs is identical to $1 - p_C\left(y = K+1|\mathbf{x}\right)$ in semi-supervised GANs.

3.2. Denoising feature matching

Feature matching technique was first introduced in [23] and its purpose is to train G to match the statistics of the real data with the generated data. The objective function for G in feature matching is

$$\min_G \left\|\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}\left[\Phi_D\left(\mathbf{x}\right)\right] - \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}\left[\Phi_D\left(G\left(\mathbf{z}\right)\right)\right]\right\|^2 \tag{3}$$

where $\Phi_D\left(\cdot\right)$ is feature extractor of D. Feature matching, however, is less effective because it misses higher-order statistics of feature distribution of D. [24] presented a new objective function for G called denoising feature matching. It models the distribution on feature space of D when evaluated on positive data by denoising autoencoder (DAE) and leads significant improvements in objectiveness of images and inception score [23]. The objective function for G in denoising feature matching is to minimise the following loss

$$\mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}\left[\left\|\Phi_D\left(G\left(\mathbf{z}\right)\right) - r\left(\Phi_D\left(G\left(\mathbf{z}\right)\right)\right)\right\|^2 - \log D\left(G\left(\mathbf{z}\right)\right)\right] \tag{4}$$

where $r(\cdot)$ is DAE. D is trained with conventional adversarial loss in vanilla GANs.

4. METHODS

We consider supervised case where all the data and corresponding labels are given. We have positive dataset $\mathcal{D}_{\text{positive}} = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$ where $y_n \in Y = \{1, 2, \ldots, K\}$ and negative dataset $\mathcal{D}_{\text{negative}} = \{(\mathbf{x}_l, y_l)\}_{l=1}^L$ where $y_l \in \overline{Y} = \{K+1, \ldots\}$, sampled from data distribution $p_{\text{data}}\left(\mathbf{x}, y\right)$. $\mathcal{D}_{\text{negative}}$ is only available at test time. We would like to train a classifier C with $\mathcal{D}_{\text{positive}}$ that can predict a correct label given positive data and detect negative data by measuring uncertainty of prediction. This could be achieved by regularising classifier C with generated data, when G is capable of generating fake negative data.

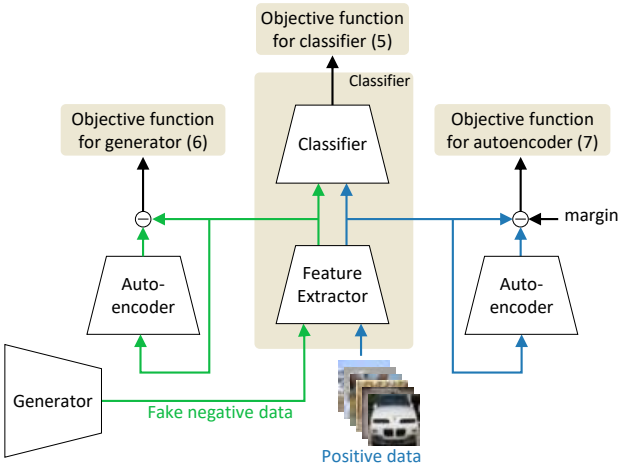


Fig. 1. Overview of our model. Note that two autoencoders share their parameters.

4.1. Classifier with uncertainty regularisation

Discriminating between real data and generated data is not our concern now. What we want is that C should have low uncertainty about positive data and high uncertainty about generated data. To enforce high uncertainty about generated data on C, we could add entropy regularisation term to regularise C with generated data. The objective function for C is

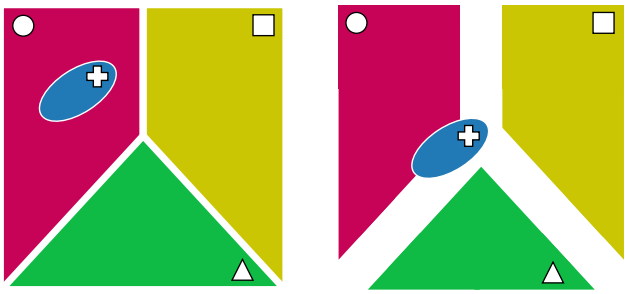
$$\min_C \quad -\mathbb{E}_{\mathbf{x},y\sim p_{\text{data}}(\mathbf{x},y)}\left[\log p_C\left(y|\mathbf{x}\right)\right] - \mathbb{E}_{\mathbf{x}\sim p_G(\mathbf{x})}\left[H\left(p_C\left(y|\mathbf{x}\right)\right)\right] \tag{5}$$

where $H\left(p_C\left(y|\mathbf{x}\right)\right)$ is the entropy of membership probability. Note that we do not explicitly discriminate generated data from real data.

This objective function indicates that C is trained to predict a label of given positive data and restrain high certainty of generated data at the same time. If generated data considerably act as negative data, C would be able to distinguish between known classes and

unknown classes by measuring uncertainty. Note that the choice of uncertainty measure is entirely up to the practitioners. Instead of the entropy, one could use negative log likelihood for uncertainty.

4.2. Marginal denoising feature matching



(a) Without regularisation (b) With regularisation

Fig. 2. Decision boundary on feature space of classifier. known classes: circle, square, triangle. Unknown class: cross. (a) Decision boundaries for known classes are close to each other and cross class resides on the same feature space of circle class. (b) Decision boundaries are tightened resulting in vague space between known classes. Cross class is more likely on the vague space which implies high uncertainty.

Modelling whole unknown classes apart from known classes is not plausible indeed. We want to model relatively compact area based on observation of classifier. Modern C based on deep networks could be divided into two parts: feature extractor which extracts features from data, and classifier which classifies given features into specific classes. The behaviour of C with unknown classes is that if negative data are given to C, it would predict them as specific known classes with low uncertainty (low entropy). This indicates that feature space of negative data was close to feature space of positive data (Fig. 2 (a)). If we can generate fake feature nearby feature of positive data and use them to regularise C (tightening decision boundary by entropy maximisation), we would achieve better detection performance than C without regularisation. Naïve approach would be corrupting feature of positive data and using it to regularise C but feature extractor of C could not be trained end-to-end with this approach which leads negative data still reside adjacent to positive data on feature space. We were able to tighten decision boundary and separate negative data from positive data on feature space if we could generate fake negative data that reside around feature space of positive data (Fig. 2 (b)).

To make G generate such fake negative data, we proposed a similar approach used in [24]. DAE tries to restore original input from corrupted input. When we employ output of DAE as a target for G, it tries to mimic the statistics of positive data on the feature space and generates realistic positive data eventually. However, we want G to generate not the data in known classes but the data in unknown classes. We introduced marginal denoising autoencoder (MDAE) which tries to model the noisy distribution of known classes on the feature space of C. This implies that MDAE models m adjacent feature space of known classes where m is a hyper-parameter. If output of MDAE is set as a goal for G, it would generate data similar to the data in known classes but not the same one that we wanted to consider them as fake negative data.

Note that vanilla G modelled the distribution of known classes whereas our G modelled the distribution m away from the one of known classes. Here, the objective function for G in marginal denoising feature matching is

$$\min_G \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})} \left[\left\| \Phi_C \left(G \left(\mathbf{z} \right) \right) - M \left(\Phi_C \left(G \left(\mathbf{z} \right) \right) \right) \right\|^2 \right] \tag{6}$$

where $\Phi_C(\cdot)$ is feature extractor of C, $M(\cdot)$ is the MDAE and $M\left(\Phi_C\left(G\left(\mathbf{z}\right)\right)\right)$ is treated as constant, as [24], because it is the target for G. Note that we do not use adversarial loss. The objective function for MDAE is

$$\min_M \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})} \left[\left| \left\| \Phi_C \left(\mathbf{x} \right) - M \left(n \left(\Phi_C \left(\mathbf{x} \right) \right) \right) \right\|^2 - m \right| \right] \tag{7}$$

where $n(\cdot)$ is corruption function and m is a hyper-parameter to set the margin.

Algorithm 1 Training procedure of GAN-MDFM

Input: θ_C (parameter of C); θ_M (parameter of M); θ_G (parameter of G); $p_{\mathbf{z}}\left(\mathbf{z}\right)$; $\mathcal{D}_{\text{positive}}$; m

while until convergence **do**
  Sample positive data pair $(\mathbf{x}, y)$ from dataset $\mathcal{D}_{\text{positive}}$.
  Sample noise $\mathbf{z}$ from the given prior $p_{\mathbf{z}}\left(\mathbf{z}\right)$.
  Generate fake negative data $\mathbf{x}_G$ from $G\left(\mathbf{z}\right)$.
  $\theta_C \leftarrow \theta_C - \nabla_{\theta_C} \left[\log p_C \left(y | \mathbf{x} \right) + H \left(p_C \left(y | \mathbf{x}_G \right) \right) \right]$
  Obtain positive feature $\phi_{\mathbf{x}} = \Phi_C \left(\mathbf{x} \right)$.
  $\theta_M \leftarrow \theta_M - \nabla_{\theta_M} \left| \left\| \phi_{\mathbf{x}} - M \left(n \left(\phi_{\mathbf{x}} \right) \right) \right\|^2 - m \right|$
  Sample noise $\mathbf{z}$ from the given prior $p_{\mathbf{z}}\left(\mathbf{z}\right)$.
  Generate fake negative data $\mathbf{x}_G$ from $G\left(\mathbf{z}\right)$ and obtain corresponding fake negative feature $\phi_G = \Phi_C \left(\mathbf{x}_G \right)$.
  $\theta_G \leftarrow \theta_G - \nabla_{\theta_G} \left\| \phi_G - M \left(\phi_G \right) \right\|^2$
end while

4.3. Detection of unknown classes

G was only required to generate fake negative data and it is no more needed after a training step is completed. Detecting unknown classes was relatively easy because we simply calculated whatever uncertainty measure we employed to regularise our classifier C (entropy in our case). After C predicted membership probability given data $\mathbf{x}$, entropy of the membership probability was easily computed and treated as uncertainty. We could choose a threshold using validation set and evaluate whether the uncertainty exceeds the threshold or not on test data to identify unknown classes.

5. EXPERIMENTS

For all experiments we conducted, the following setup were used. Although we regularised classifier by only entropy, we measured two uncertainties: the entropy of membership probability and the maximum of logits. The corruption function is isotropic Gaussian noise with $\sigma = 1$. The hyper-parameters we used is following: Batch size was 128, Adam optimisation [25] was used with $\alpha = 1\text{e}{-5}, \beta_1 = 0.5$, and $\beta_2 = 0.999$. $\mathbf{z}$ is sampled from a uniform distribution [-1, 1] with dimension of 64 for MNIST, 128 for CIFAR10. We followed the rules suggested in DCGAN [16] and used similar architecture of theirs for the classifier and the generator. The classifier consists of convolutional layers and fully connected layers at the end of the network. The generator is composed of fully connected layers which

Table 1. Classification accuracy

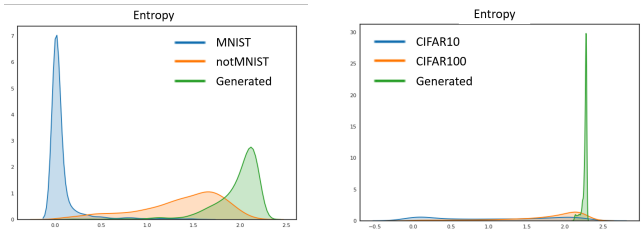
	Baseline	Convex	PCA	VAE	GAN	GAN-MDFM
MNIST	0.987	0.986	0.987	0.988	**0.991**	0.987
CIFAR10	0.707	0.654	0.700	0.702	0.616	**0.728**

Table 2. Area under the curve

		Baseline	Convex	PCA	VAE	GAN	T-scaling	GAN-MDFM
MNIST vs. notMNIST	entropy	0.930	0.976	0.907	0.926	**0.987**	0.938	**0.987**
	max logit	0.885	0.969	0.840	0.865	0.982	0.887	**0.991**
CIFAR10 vs. CIFAR100	entropy	0.666	0.671	0.656	0.666	0.641	0.707	**0.729**
	max logit	0.696	0.664	0.687	0.691	0.641	0.696	**0.721**

project and reshape of input at the beginning of the network and transposed convolutional layers. Marginal denoising autoencoder is simply stacked fully connected layers with hidden dimension of the same as input dimension. The margin m is set to the same value as input dimension.

The baseline is a classifier that have the same architecture of classifier in GAN-MDFM but it is trained with cross entropy only. We regularised C with various generating methods to compare with proposed model. Convex is a method of weighted sum of data. We performed PCA on each dataset and used principal directions multiplied by Gaussian noise as fake negative data. Variational autoencoder (VAE) [26] and GAN have similar architectures of GAN-MDFM and we generated fake negative data by sampling $\mathbf{z}$ from higher variance normal distribution or wider range of uniform distribution than trained prior $p_{\mathbf{z}}(\mathbf{z})$. For VAE, $\mathbf{z}$ was sampled from normal distribution with variance of 10 for MNIST and variance of 5 for CIFAR10. For GAN, $\mathbf{z}$ was sampled from uniform distribution [-10, 10] for MNIST and [-5, 5] for CIFAR10. We refer T-scaling as temperature scaling which is one of post-processing after training introduced in [14].



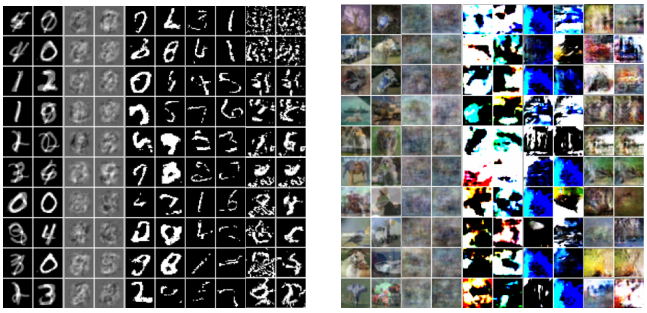
(a) MNIST vs. notMNIST (b) CIFAR10 vs. CIFAR100

Fig. 3. Entropy histogram of GAN-MDFM. Positive data have low entropy and generated data have high entropy as a result of our objective function for C.

We trained GAN-MDFM with MNIST, CIFAR10 and evaluated the classification accuracy (Table 1). GAN-MDFM did not degenerate the accuracy at all but even improved on CIFAR10. We think that the accuracy improvement is because the data generated by GAN-MDFM could be thought of well-designed augmented data for regularising classifier that have effect on keeping away from each known classes resulting in tightening decision boundary on the feature space.

We evaluated how well GAN-MDFM can distinguish unknown classes from known classes by area under the curve (AUC). We trained GAM-MDFM with MNIST, CIFAR10 and evaluated on notMNIST, CIFAR100 respectively (Table 2). GAN-MDFM outperformed baseline and other generating methods on both datasets.

The generated data from GAN-MDFM seemed similar to positive data but not exactly the same as our purpose (Fig. 4). We showed random samples after training for other generating methods but for GAN-MDFM we displayed generated data every 20th epoch per row through training because GAN-MDFM suffered from mode collapsing problem. However, our concern was not to generate realistic and diverse data but fake negative data that are sufficient to regularise C.



(a) MNIST samples (b) CIFAR10 samples

Fig. 4. Generated data from various generating methods. Each 2 columns are corresponding to convex, PCA, VAE, GAN, GAN-MDFM, respectively.

6. CONCLUSIONS

We have proposed a unknown class generator that is able to generate fake negative data. This was achieved by marginal denoising autoencoder that provided a target distribution which is m away from distribution of positive data on feature space of the classifier to the generator. The generated data were treated as fake negative data and provided to the classifier for regularisation resulting in reliable membership probability as uncertainty measure. We have achieved that classification accuracy is comparable and even improved because of the effect of data augmentation. We have showed that the entropy of membership probability and max logit are fine uncertainty measures, and AUC was improved compared to other generating methods and the baseline only trained with cross entropy.

7. REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, V. Erhan, D.and Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[5] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proceedings of the British Machine Vision Conference*, 2016.

[6] M. A. F. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "Review: A review of novelty detection," *Signal Process.*, vol. 99, pp. 215–249, June 2014.

[7] W. J. Scheirer, A. de Rezende Roch, A. Sapkota, and T. E. Boult, "Towards open set recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 7, pp. 1757–1772, 2013.

[8] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," 2016, arXiv preprint arXiv:1612.01474.

[9] W. J. Scheirer, A. Rocha, R. J. Micheals, and T. E. Boult, "Meta-recognition: The theory and practice of recognition score analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1689–1695, 2011.

[10] A. Bendale and T. E. Boult, "Towards open set deep networks," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[11] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[12] P. Bodesheim, A. Freytag, E. Rodner, M. Kemmler, and J. Denzler, "Kernel null space methods for novelty detection," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[13] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2016.

[14] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2017.

[15] Q. Da, Y. Yu, and Z. Zhou, "Learning with augmented class by exploiting unlabeled data," in *Proceedings of the AAAI National Conference on Artificial Intelligence (AAAI)*, 2014.

[16] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, arXiv preprint arXiv:1511.06434.

[17] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, "Unrolled generative adversarial networks," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[18] T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li, "Mode regularized generative adversarial networks," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[19] D. Berthelot and L. Schumm, T.and Metz, "BEGAN: boundary equilibrium generative adversarial networks," 2017, arXiv preprint arXiv:1703.10717.

[20] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, arXiv preprint arXiv:1701.07875.

[21] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," 2017, arXiv preprint arXiv:1704.00028.

[22] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *Proceedings of International Conference on Information Processing in Medical Imaging (IPMI)*, 2017, pp. 146–157.

[23] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved Techniques for Training GANs," in *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[24] D. Warde-Farley and Y. Bengio, "Improving generative adversarial networks with denoising feature matching," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[26] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes.," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.