Using GANs for DeepFake Image Detection (Group ID: 12)

Aditya Upadhyayula supadhy6@jhu.edu
Amrit Nidhi anidhi3@jhu.edu
Samarth Marudheri smarudh1@jhu.edu
Shreyash Kumar skumar77@jhu.edu

1 Introduction

Recent advances in synthesizing artificial images using Generative Adversarial Networks (GANs) have had their fair share of advantages and disadvantages. GANs have proven useful in learning the underlying representation of data which is then used in solving a multitude of problems faced by the deep learning community. Few of them include artificial data synthesis, and to defend adversarial attacks. The same solutions can also be used in affecting confirmation bias thus affecting public opinion.

The goal of the study is to detect fake images. An autoencoder is trained adversarially to obtain a latent rich representation of real images. The latent representation is then used to reconstruct an image based on the given input image. We propose a comparative study to test if reconstructed features differ strongly for real and fake images leading to enhanced classification performance. We further argue that adversarially training a classifier (DCGAN) generalizes better on unseen data as compared to CNN based architectures.

Related Work: The detection of deep fakes using GANs has been shown to be effective by [Zhang et al., 2019] and [Yarlagadda et al., 2018].

2 Methodology

2.1 Datasets:

Real face image dataset - CelebA-HQ (Celebrity face images dataset) - There were a total of 202,599 images. Due to limited computational power, we used 16000, 4000 and 10000 of the images to train, validate and test respectively.

Fake image dataset - Nvidia's StyleGAN generated images [1] - There are a total of 1 million images. But we used only 10000 of them for testing purposes as the unseen data passed to the models.

2.2 Data Preprocessing:

Each of the original images (218 x 178) are resized to (64x64) and converted to Numpy arrays for efficient training using Google Colab resources. Finally, the images are normalized and given to the PyTorch Dataset class as an input.

2.3 Setup, Training and Evaluation:

2.3.1 General Setup

An AutoGAN network is trained to learn the underlying distribution of real faces from the CelebA-HQ dataset. This trained network is then used to generate a fake image counterpart for a given input image. AutoGANs are known for their convergence stability while training in comparison to regular GANs [Gong et al., 2019],[Lindqvist et al., 2018]. The AutoGAN architecture consists of decoder, encoder and a discriminator model as shown in Figure 3. The encoder consisted of a single residual block [2] that outputs a latent representation (256 x 16 x 16). These latent representations are then given to the decoding layer to generate the reconstructed image. The discriminator consists of 4 convolution layers with leaky ReLU as the activation function. This network was trained for 25 epochs on 16000 real images from the CelebA dataset, and was validated on 4000 images from the CelebA dataset. We used Adam optimizer (learning rate = 0.00001) for training the AutoGAN (See figure 6)

2.3.2 Experiment 1

This experiment works via two approaches as summarized in Figure 1. For Approach 1, we prepare an aggregation of images from Nvidia and CelebA datasets and then perform a train-val split on the complete dataset (train:val - 80:20). An additional 20000(10k from CelebA + 10k from Nvidia) images were used to test the models. Then, we perform the following steps for each of training, validating and testing phases. The Nvidia fake images, and the Celeb-A real images are passed to the simple CNN for classification (Approach 1). The accuracy of the simple CNN (2 layers of convolutions, and 1 fully connected layer with max pooling, dropout (p = 0.2), and SGD optimization) from Approach 1 is used as a baseline. Additionally, the Nvidia fake images and Celeb-A real images are passed through the trained AutoGAN model to obtain the reconstructed images for every image that is passed. The test data in this case consisted of 20000 of these reconstructed images. The obtained outputs are then passed to the simple CNN classifier for classification (Approach 2) to see if performance in this approach is better than Approach 1. For Approach 2, we also varied the complexity of AutoGAN's generator to have 1, and 4 residual layers.

[1] https://www.thispersondoesnotexist.com/

[2] We varied the complexity of the encoder by changing it to 4 residual blocks. See figure 6

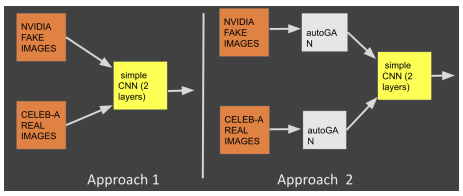

Figure 1: Evaluation workflow for testing the efficacy of generated fake images using AutoGAN

2.3.3 Experiment 2

Experiment 1 had the classifier trained on both Nvidia and Celeb-A datasets. We wanted to see how a classifier trained only on one dataset generalizes to an unseen dataset (Nvidia). In Experiment 2, we contrasted the performance of the CNN-classifier (a simple CNN with 2 convolution layers) against a DCGAN network by training on the real and reconstructed images for the Celeb-A dataset. The CNN was trained on real Celeb-A images and the corresponding images generated by AutoGAN (from 2.3.1). The DCGAN network was also trained on the same data as the CNN. Each of these models was then evaluated for classification performance on the unseen Nvidia fake image dataset. The flowchart is illustrated in Figure 2.

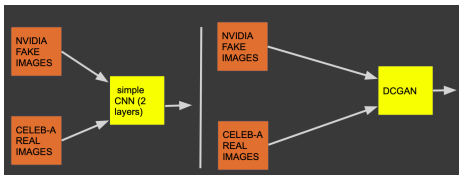

Figure 2: Evaluation workflow for Detecting Fake Images using CNN and DCGAN

3 Results

The AutoGAN model was trained for 25 epochs to reconstruct an image based on an input image for pairwise generation (for more details on AutoGAN training, see Appendix figures 9, 10).

3.1 Experiment 1 : Comparing simple CNN performance on approaches 1 & 2

The performance runs shows that the simple CNN based architecture, Approach 1 was able to beat Approach 2. In other words, the simple CNN performed better in classifying real-fake pairs when compared to the reconstructed real-fake pairs generated from AutoGAN. The accuracy of the classification task is as follows:

- CNN classifier - Training: 99.7%, Val: 99.2%, Test:98.9%
- Single Residual Layer GAN - Training: 91%, Val: 90.35%, Test: 64.24 %
- Four Residual Layer GAN - Training: 95 %, Val: 94.30%, Test: 64.80 %

This is shown in figures 7. The training and validation accuracy plots of the above task is shown in 5 & 6 However, an important result is the improvement in performance of a 4-layer AutoGAN over a 1-layer AutoGAN 6b. The result shows the potential of GAN architectures to successfully learn the underlying representation of real images to achieve comparable performance to CNN based architectures.

3.2 Experiment 2 : Comparing simple CNN and DCGAN's performance on test data

The simple CNN architecture on test data(Nvidia) yielded an accuracy of 45.3%. The DCGAN was able to generalize better on the same test data with an accuracy of 57.2% i.e an accuracy increase of 11.9%. This result implies the robustness of DCGAN. The comparison is shown in Figure 8.

4 Discussion

We wanted to check the efficacy of GANs in classifying an image as real or synthesized(fake). To achieve this, an AutoGAN model was trained to reconstruct images based on an input image. We found that a 4-layer AutoGAN yielded better classification than a 1-layer AutoGAN, albeit underperforming compared to simple CNN classification model. With more exploration into the AutoGAN architecture, we might be able to achieve comparable performance with the simple CNN. We also wanted to check how these models generalize on unseen data.So,we trained a DCGAN to detect fakes and the model's accuracy beat that of simple CNN. This ability of DCGAN to classify unseen Nvidia data better shows its robustness.This data was generated using Style-GAN method as compared to the data in the validation set(Celeb-A) and hence, better classification on Nvidia data shows better generalization.
Thus, DCGAN was able to generalize better compared to the simple CNN making it more robust to adversarial attacks in real world scenarios.
Our future work includes testing our model on a dataset with higher number of classes, and images with higher degrees of synthetic manipulation. We also plan to work towards improving the reconstruction performance of our generator and incorporate a different discriminator for our AutoGAN network to drive classification accuracy.

5 Code

The code used to obtain our results can be found in our GitHub repository at https://github.com/Samarth2506/DL_Project

References

X. Gong, S. Chang, Y. Jiang, and Z. Wang. Autogan: Neural architecture search for generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3224–3234, 2019.

B. Lindqvist, S. Sugrim, and R. Izmailov. Autogan: Robust classifier against adversarial attacks. *arXiv preprint arXiv:1812.03405*, 2018.

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

S. K. Yarlagadda, D. Güera, P. Bestagini, F. Maggie Zhu, S. Tubaro, and E. J. Delp. Satellite image forgery detection and localization using gan and one-class classifier. *Electronic Imaging*, 2018(7):214–1, 2018.

X. Zhang, S. Karaman, and S.-F. Chang. Detecting and simulating artifacts in gan fake images. *arXiv preprint arXiv:1907.06515*, 2019.

Appendix

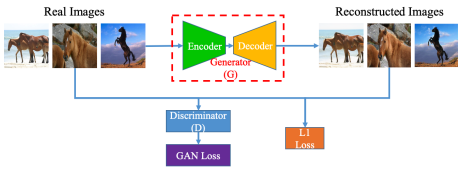

Figure 3: AutoGAN Architecture: Source [Zhang et al., 2019]

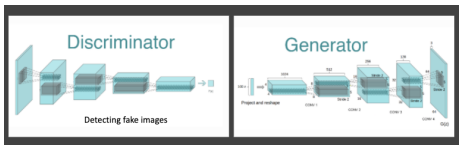

Figure 4: DCGAN Architecture: Source [Radford et al., 2015]

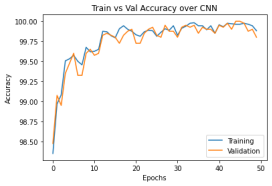

Figure 5: CNN classification accuracy as the baseline

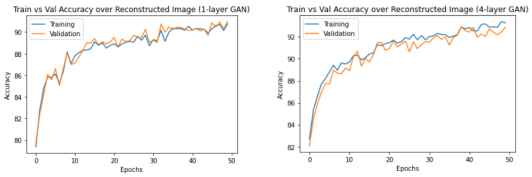

Figure 6: Classification accuracy on GAN reconstructed images

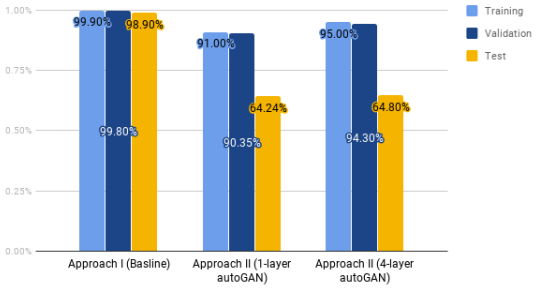

Figure 7: Experiment 1 results : Performance improvement of 4-Layer AutoGAN over 1-Layer AutoGAN

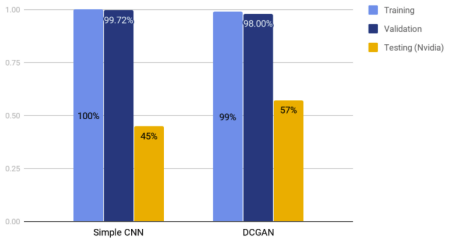

Figure 8: Experiment 2 Results : Performance improvement of DCGAN over simple CNN

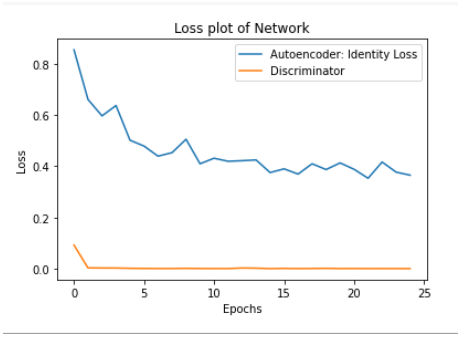

Figure 9: Loss profile of AutoGAN for over 25 epochs

Figure 10: Corresponding fake images generated by the AutoGAN