UNSUPERVISED ADVERSARIAL IMAGE RECONSTRUCTION REPRODUCIBILITY REPORT

Lei Guo, Yao Zhou, Nathaniel Fernandes
lg1y19@soton.ac.uk, yz1y19@soton.ac.uk, nf1g16@soton.ac.uk

ABSTRACT

To address a large problem surrounding the lack of reproducibility of published results in AI papers, and with inspiration from the International Conference on Learning Representation challenge, the Unsupervised Adversarial Image Reconstruction has been reproduced. The method of solving the problem of recovering an underlying signal from lossy, inaccurate observations in an unsupervised setting has been re-implemented. A GCGAN model with a combination of Prior and Likelihood methods has been used, and reconstructions on the CelebA dataset have been evaluated. The proposed approach is the reproducibility of this method.

1 INTRODUCTION

Unsupervised Adversarial Image Reconstruction is an effective technique used to obtain accurate and correct results from limited information and a lossy signal. Typically, good results are usually accompanied by sufficient prior information, a clear dataset and a good structure. However, in some cases, such as identifying a criminal from CCTV footage, important information, like the person's face, is too hard to obtain due to fuzzy images or videos. Recovering the image from sources with less information, using an unsupervised setting, can be considered as a solution. In this setting, only the observed result, the unclear images and prior information on the measurement process are available, which are not enough for other supervised learning techniques, such as a neural network, to generate satisfactory results. In section II, the principles and experiments on this problem will be illustrated, followed by an analysis and discussion in section III and IV respectively.

2 EXPERIMENTS

2.1 METHODOLOGY

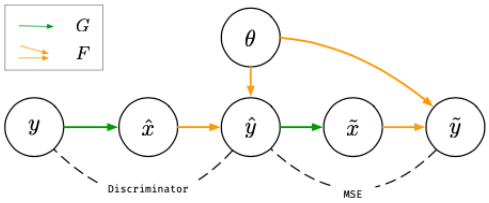

Figure 1: General Approach

MODEL:
Our general approach used a Generative Adversarial Network (GAN) as the basic model, combined with a Prior and Likelihood to train the model. Our aim was to train a good-performing Generator to recover and reconstruct the lossy image. As shown in Figure 1, this required the construction $\hat{x} := G(y)$ to have a high probability under the Likelihood and the Prior.
Prior: A measurement y was sampled from the data, a reconstruction $\hat{x}$ produced, and a perturbation parameter sampled. A Discriminator and Generator were used to produce the simulated measurement $\hat{y} := F(\hat{x};)$ to be similar to measurements in the data.

Likelihood: A Generator was used to produce reconstructions with a high likelihood, and compute $||\hat{y} - y||^2$, the mean squared error (MSE) between $\hat{y}$ and y. The MSE was constrained to be small.

TRAIN:

Training algorithm and train step, described below, was obtained along with the dependency structure illustrated in Figure 1.

Algorithm 1 Training Procedure.

Require: Initialize parameters of the generator G and the discriminator D.

while (G, D) not converged **do**

Sample $\{y_i\}_{1 \le i \le n}$ from data distribution p_Y

Sample $\{\theta_i\}_{1 \le i \le n}$ from P_Θ

Sample $\{\varepsilon_i\}_{1 \le i \le n}$ from $P_\mathcal{E}$

Set $\hat{y}_i$ to $F(G(y_i), \theta_i) + \varepsilon_i$ for $1 \le i \le n$

Update D by ascending:

$$\frac{1}{n}\sum_{i=1}^{n} \log D(y_i) + \log(1 - D(\hat{y}_i))$$

Update G by descending:

$$\frac{1}{n}\sum_{i=1}^{n} \lambda \cdot \|\hat{y}_i - F(G(\hat{y}_i); \theta_i)\|_2^2 + \log(1 - D(\hat{y}_i))^3$$

end while

```
def _train_step(self, data):
    netG = self.train_model["netG"]
    optimizerG = self.train_model["optimizerG"]
    netD = self.train_model["netD"]
    optimizerD = self.train_model["optimizerD"]
    criterion = self.train_model["criterion"]
    device = self.config["device"]
    lan = self.config["lan"]

    real_data = data[0].to(device)

    noise = model.get_noise(real_data, self.config)
    fake_data = netG(noise)
    label = model.get_label(real_data, self.config)

    fake_sample = netG(noise)
    errMSE = nn.functional.mse_loss(fake_data, fake_sample)

    errD, D_x, D_G_z1 = model.get_Discriminator_loss(netD, optimizerD,
                        real_data, fake_data.detach(), label, criterion, self.config)
    errG, D_G_z2 = model.get_Generator_loss(netG, netD, optimizerG,
                        fake_data, label, criterion, self.config)
    errG = errG + errMSE*lam

    return errD, errG, D_x, D_G_z1, D_G_z2
```

Figure 2: Training Algorithms and Train Step

2.2 IMPLEMENTATION

Architectures:

Our network architectures are inspired by the Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks (GCGAN) architecture [1]. They used convolutional neural networks in the Generator and Discriminator feature extraction layers to replace the Multilayer Perceptron in the original GAN [2]. The Generator is used to map the eigenvector Z to the data space. Since our data is an image, converting Z to data space required creating an RGB image of the same size as the training image, through a series of transposed convolutions, each of which is paired with a BN layer and a ReLU activation function.

The output of the generator activates the activation function through tanh, so that the output data range is the same as the input picture, between [-1, 1]. The discriminator is a binary classification network that takes images as input, and the probability that the input and output images are real images. The discriminator uses a 3x64x64 input image, through a series of convolution, BN layer and LeakyReLU activation functions, and finally outputs the final probability through the Sigmoid activation function.

Hyperparameters:

We use same learning rate 0.0002 of the Generators and the Discriminator, using the Adam optimizer, using = 0.5. The weights are initialized from a normal distribution. We set = 2. We set loss rate = 0.5.

Datasets:

We train and tested our approach using the CelebA image dataset. It is a dataset consisting of images of celebrities, containing approximately 200 000 samples. In addition, all the images have been resized to 64x64 and normalised.

Corruptions:

In order to observe the most realistic performance of our model, we corrupted all the images once. We randomly removed pixels in every image, and set a loss rate parameter to control the degree of corruption and difficulty of the reconstruction.

2.3 RESULT

The model was run for 5 epochs in the CelebA dataset, Figure 3 shows reconstructions obtained from our mode and the Generator and Discriminator Loss curves. The construction performance is not good as those of the original paper due to the simplification of the model, layers and hyperparameters.

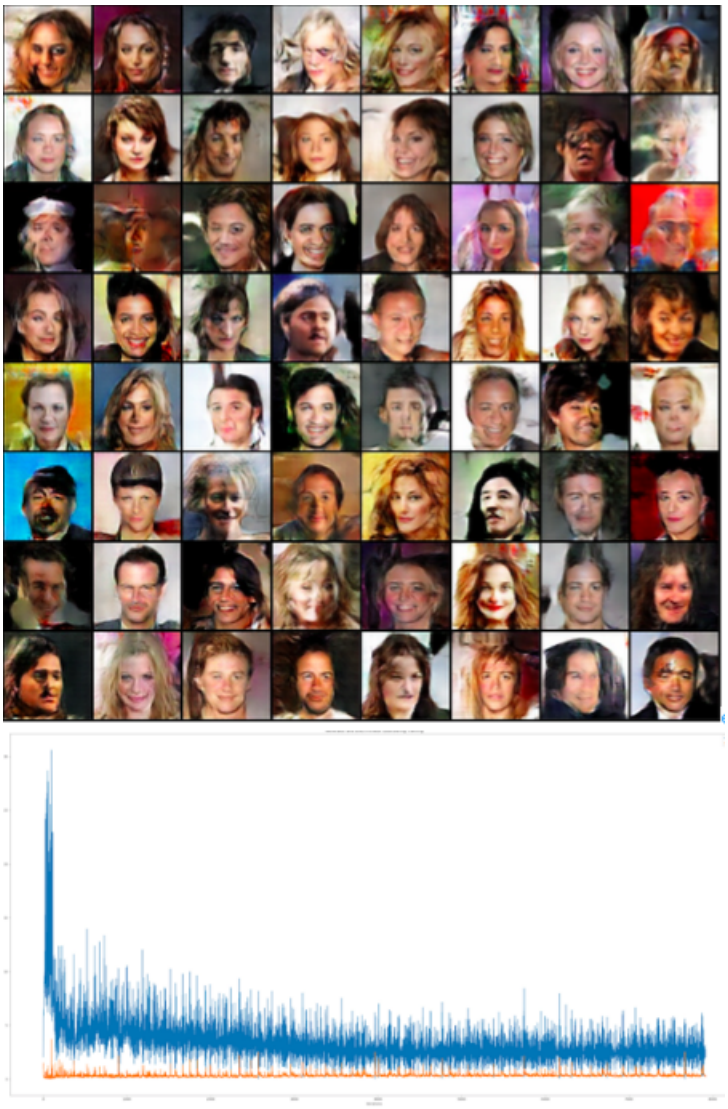

Figure 3: Reconstruction Images and Generator and Discriminator Loss curves

3 ANALYSIS

The original paper uses a 'maximum a posteriori' (MAP) estimate to get the result with a lossy observed signal. However, the likelihood term of it is too difficult to calculate because the sampling term is unknown(equation 4 on the original paper). Due to limited information including observed signal and prior on the measurement process, an AmbientGAN model[3] is introduced in the original paper to cope with this situation. Figure 4 shows the training of an AmbientGAN network, where generator G's output passes through f, a random measurement function. Discriminator D determines whether the signal is from the real dataset or the generator. The generator can produce distorted data to allow the model to train for usage with corrupted images.

AmbientGANs can be evaluated qualitatively by inputting highly noisy images and assessing the clarity of the generated output, and quantitatively using an 'inception score', a metric for determining the quality of images produced by generative models.

However, because of finite devices such as a GPU, the simplified method is implemented and in the next part, the results of experimental reproduction will be discussed.

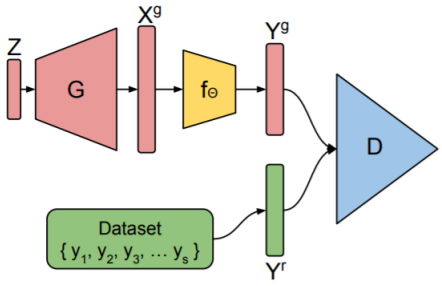

Figure 4: Training of an AmbientGAN network

4 DISCUSSION

We reproduced the method in the paper and implemented the baseline model. Through a large number of experiments, we have concluded that the method proposed in this paper is reproducible. Firstly, the model training method proposed in the paper is a combination of Prior and Likelihood. This method appears to be complicated, but the actual operation is straightforward. Secondly, the experimental results show that this method does improve the reconstruction performance of the lossy images. Additionally, this method can be used on several variant GAN models or other models such as VAE to improve the reconstruction performance.
However, we analysed the code that was submitted by the author of the paper and found that the author's code has some errors, poor readability and reusability, and the amount of resources used is too large to be reproduced on the same scale. Therefore, our implementation method has been simplified, using GCGAN instead of SAGAN, and the hyperparameters have been simplified.
In summary, the method proposed in the paper is reproducible and usable, but it takes a lot of resources and time to fully reproduce the results of the paper.

5 CONCLUSION

We have re-implemented a general approach to recover a signal from lossy measurements using GC-GAN, without access to uncorrupted signal data. This method is composed of a linear combination of an adversarial loss for recovering realistic signals, and a reconstruction loss to tie the reconstruction. It is concluded that this method is reproducible and the reconstruction performance is good. However, it is difficult to completely reproduce the author's code, we need more resources and times to reproduce in the future.

REFERENCES

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

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2014). Generative adversarial nets. *In Advances in neural information processing systems*, (pp. 2672-2680).

Ashish Bora, Eric Price, and Alexandros G. Dimakis. AmbientGAN: Generative models from lossy measurements.In *International Conference on Learning Representations*, 2018. URL https: //openreview.net/forum?id=Hy7fDog0b

A APPENDIX

Code: https://github.com/LeiGuo0417/COMP6248-Reproducability-Challenge-Unsupervised-Adversarial-Image-Reconstruction-reproducibility.git