Handwriting Profiling using Generative Adversarial Networks

Arna Ghosh*, Biswarup Bhattacharya* & Somnath Basu Roy Chowdhury*
Department of Electrical Engineering, IIT Kharagpur, India

Abstract

Handwriting is a skill learned by humans from a very early age. The ability to develop one's own unique handwriting as well as **mimic another person's handwriting** is a task learned by the brain with practice. This paper deals with this very problem where an intelligent system tries to learn the handwriting of an entity using Generative Adversarial Networks (GANs). We propose a modified architecture of **DCGAN** [3] to achieve this. We also discuss about applying reinforcement learning techniques to achieve faster learning. Our algorithm hopes to give new insights in this area and its uses include identification of forged documents, signature verification, computer generated art, digitization of documents among others. Our early implementation of the algorithm illustrates a good performance with MNIST datasets.

Introduction

The generator learns to generate words looking similar to the reference word provided. The discriminator provides feedback to the generator while developing words from individual alphabets regarding the desired style aided by reinforcement learning policies.

Our goal: To develop an intelligent system which learns the handwriting style of a specific entity and is able to write any text in that particular style.

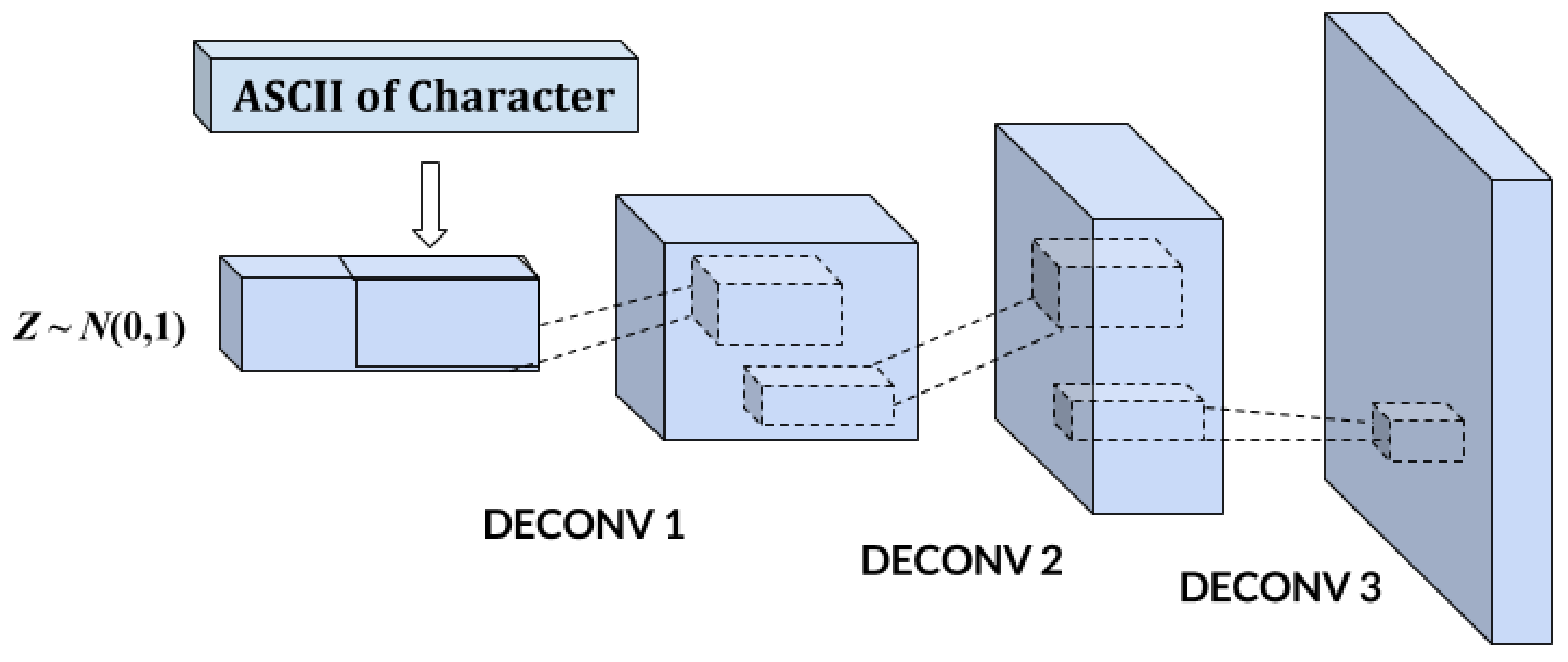
Key Idea: Integration of **DCGAN with reinforcement learning** to mimic a specific style of handwriting. With every word we write, we tend to adjust our writing style to correct any difference from the reference - our desired style of writing. Thus, we expect the algorithm to provide human-like results.

Key Insight: Provide some insight into the role of GANs in attaining human-like behavior in complex activities and the combination of GANs with existing techniques like reinforcement learning to improve performance.

Architecture

The model architecture is built around a Deep Convolutional Generative Adversarial Network (DCGAN) [3] wherein:

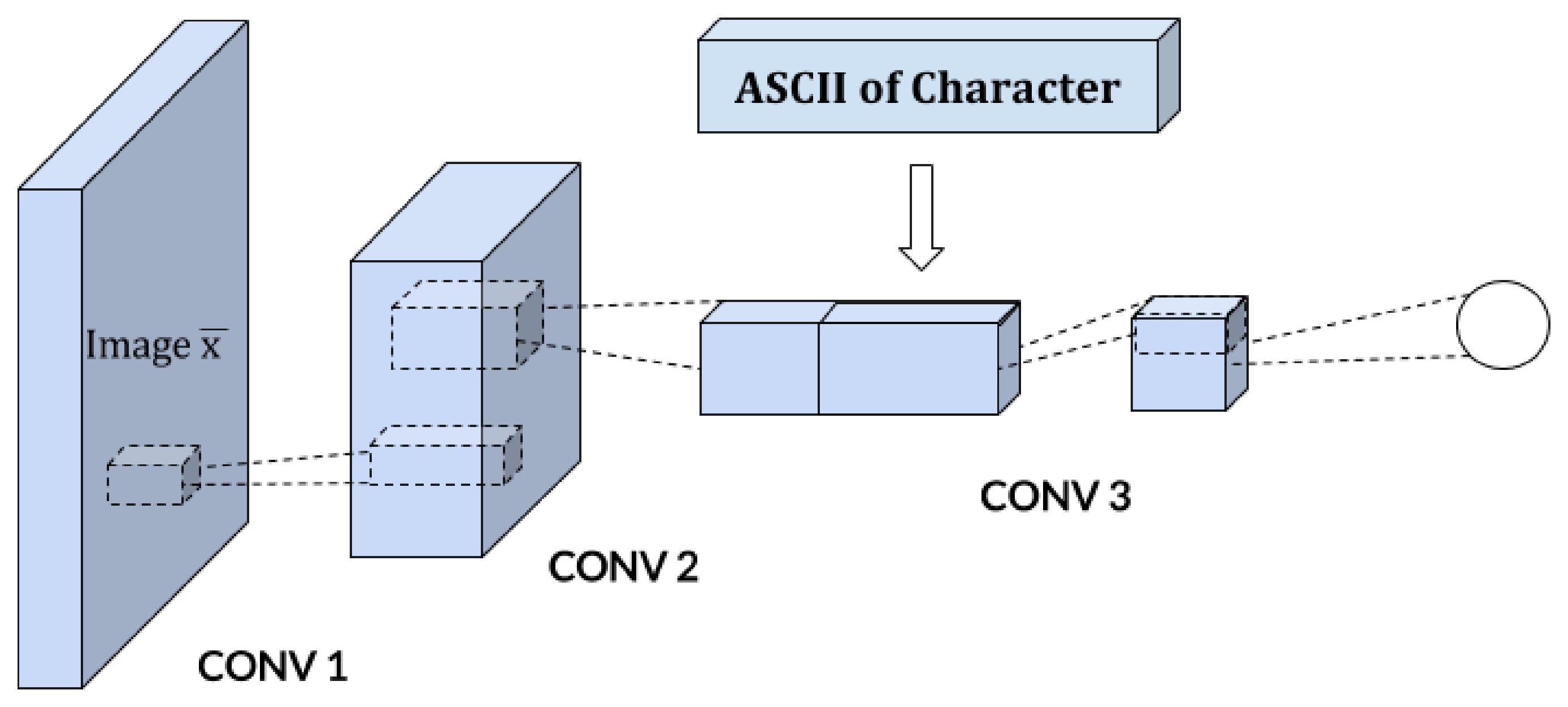
- **Generator network**: The base image x is encoded into a feature representation φ(x) using a convolutional network giving $\hat{x}$: G(z, φ(x), e) where z ∼ N(0, 1). The ASCII value of the character to be generated is concatenated with a noise vector and provided as input to the generator as shown in the figure. Feedforward through a normal de-convolutional network in generator G results in a synthetic image $\hat{x}$.



Generator Network

- **Training**: The easiest way to train the GAN is to view {character, image} pairs as joint observations and train the discriminator to judge pairs as real or fake. Similar to [4], additionally, we add a third type of input consisting of real images with incorrect character ASCII value, which the discriminator must learn to score as fake thus allowing it to learn whether real training images match the character embedding.

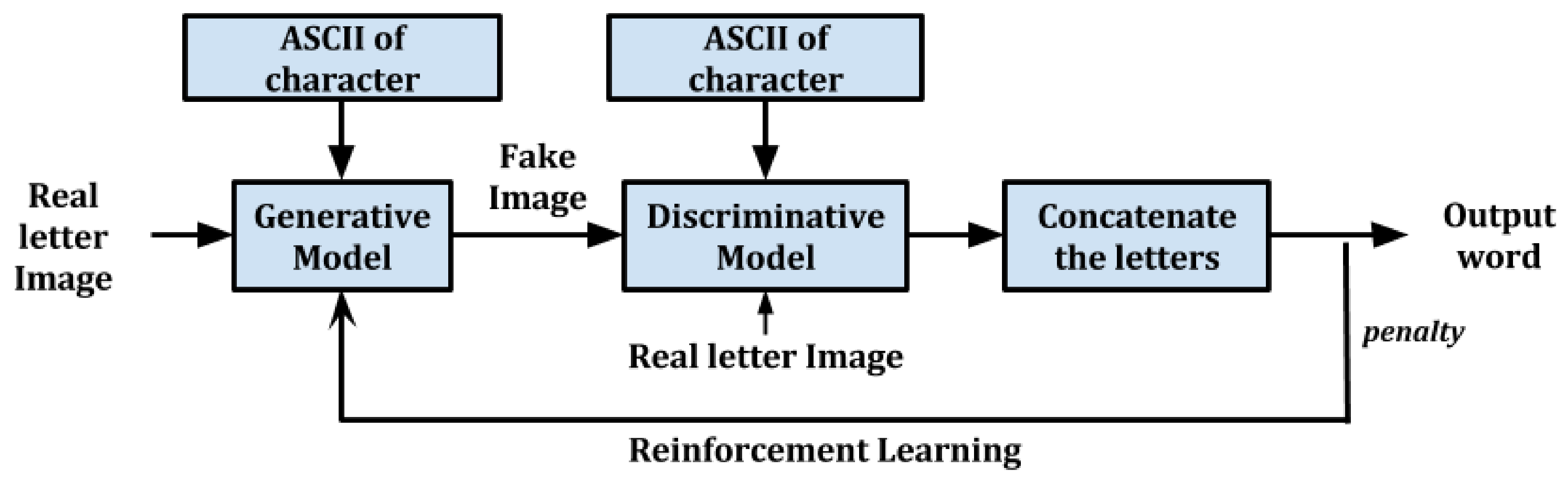
- The **discriminator network** is a convolution network followed by leaky ReLU. It is interpreted as performing feedforward inference on a given triplet (x, e, $\hat{x}$). We reduce the dimensionality of the image and finally concatenate the character embedding to the output to calculate the score from D. The ASCII of character is also fed into the discriminator network in order to produce the output as shown in the figure.



Discriminator Network

Reinforcement Learning

The idea of reinforcement learning is applied to the average distance between consecutive letters, angle of ascent and angle of descent of strokes. Our system exclusively uses a **penalty-based system**. Objective functions of the difference between the spacing between two letters, angles of ascent/descent need to be minimized.



The GAN system has a controller which keeps updating the thresholds according to the result of this objective function. The policy of the system is to achieve zero as the result of all the objective functions indicating perfectly following expected profile. The penalty system is expected to converge to this policy given enough training data.

Experiments

Preliminary experiments were done with the truncated **MNIST dataset** of a specific handwriting only. Instead of using the character dataset of one's handwriting, we use the handwritten digit database to test the scope of our architecture in generating digits.

We correspondingly use the ASCII value of digits 0 to 9 for the purpose. The wide variation of the style in writing leads to variations in generated images of the same digit. For a particular style of writing, we believe that such variations will be low and the generator network will be far more accurate in replicating a particular style of writing.

Further experiments would include training the architecture on a particular handwriting dataset. We plan to use a sentence like ”the quick brown fox jumped over the lazy dog” written multiple times to generate a dataset covering most of the variations. Further work would involve using trained generator and discriminator to generate words.

References

[1] Arica, N., and Yarman-Vural, F. T. 2002. Optical character recognition for cursive handwriting. IEEE Trans. Pattern Anal. Mach. Intell. 24(6):801–813.

[2] Graves, A.; Bunke, H.; Fernandez, S.; Liwicki, M.; and Schmidhuber, J. 2008. Unconstrained online handwriting recognition with recurrent neural networks. In in Advances in Neural Information Processing Systems 20. MIT Press.

[3] Radford, A.; Metz, L.; and Chintala, S. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. CoRR abs/1511.06434.

[4] Reed, S.; Akata, Z.; Yan, X.; Logeswaran, L.; Schiele, B.; and Lee, H. 2016. Generative adversarial text to image synthesis. arXiv preprint arXiv:1605.05396.

* All authors have equal contribution