UNMET DEPENDENCIES

Project by:

Rishabh Bhonsle (17217) | Prithvi Poddar (17191) | Sourav Yadav (17285)

Department of Electrical Engineering and Computer Science

DSE601: Artificial Intelligence and its Scientific Applications

AN ADVISORY OF SORTS

Dear viewer,

We would firstly like to thank you for your time to go through this presentation and witness our efforts.

Being an end semester project, we could not do justice to the project due to the time constraints of submission. We had to limit our codes to the availability of data sets and priority of importance of associated topics. Specifics of the code can be found in the report. This would just be a brief overview.

However that did not stop us from giving you the complete picture through this presentation. Hope you enjoy learning from this project as much as we did from it!

Regards,

Team Unmet Dependencies

PART 1: HANDWRITING GENERATION

1. The Beginning
2. Individuality of Alphabets
3. Tiny Adjustments
4. Learning from Society
5. United we Stand

May it be through advertisements, pamphlets, brochures, building labels, sweet boxes, brand titles – incredible fonts surround us. However, their importance is often overlooked!

A company might charge $99 for permission to use a specific typeface on the web, or $1,000 to install a font on, say, 20 computers. But commissioning a brand new, custom typeface can cost upwards of $50,000 per face. It's also easy to see how piracy or plagiarism could be tempting. Making a font is hard.

Thus, using this project, an individual can be creative and build his/her own custom font for official or personal purposes just by using his/her handwriting!

Fact check:

https://www.wired.com/2015/10/you-wouldnt-think-it-but-typeface-piracy-is-a-big-problem/

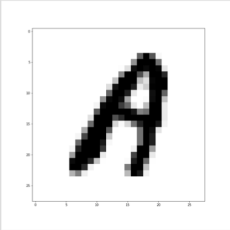
THE BEGINNING – DEFINING TRAINING INPUTS

- We consider multiple hand-written texts in the form of images of a selected user who wishes to generate his/her own font using his/her handwriting. We use this as our training data with it's corresponding digital text as labels. These texts could be in the form of words or sentences.
- We shall however skip this particular part in the code (why? - refer advisory). But here's and example of how the input image might look like:

Label: INPUT EXAMPLE

INDIVIDUALITY OF ALPHABETS – THE ACTUAL INPUTS

- Applying image segmentation to the data (skipped) stated in the previous slide and applying image augmentation (padding/cropping) will give us a result similar to the NIST dataset. We thus start here. (We already know that we can work with the concepts of image segmentation and augmentation from our performance is assessment 3 and hence can safely assume that we could have, if sufficient time was provided, done the previous step :P)
- We have considered a better (for achieving our goal) version of the NIST dataset from Kaggle for our code. The format of our data is as follows:



Label: A

TINY ADJUSTMENTS– PREPARING DATA FOR GANS

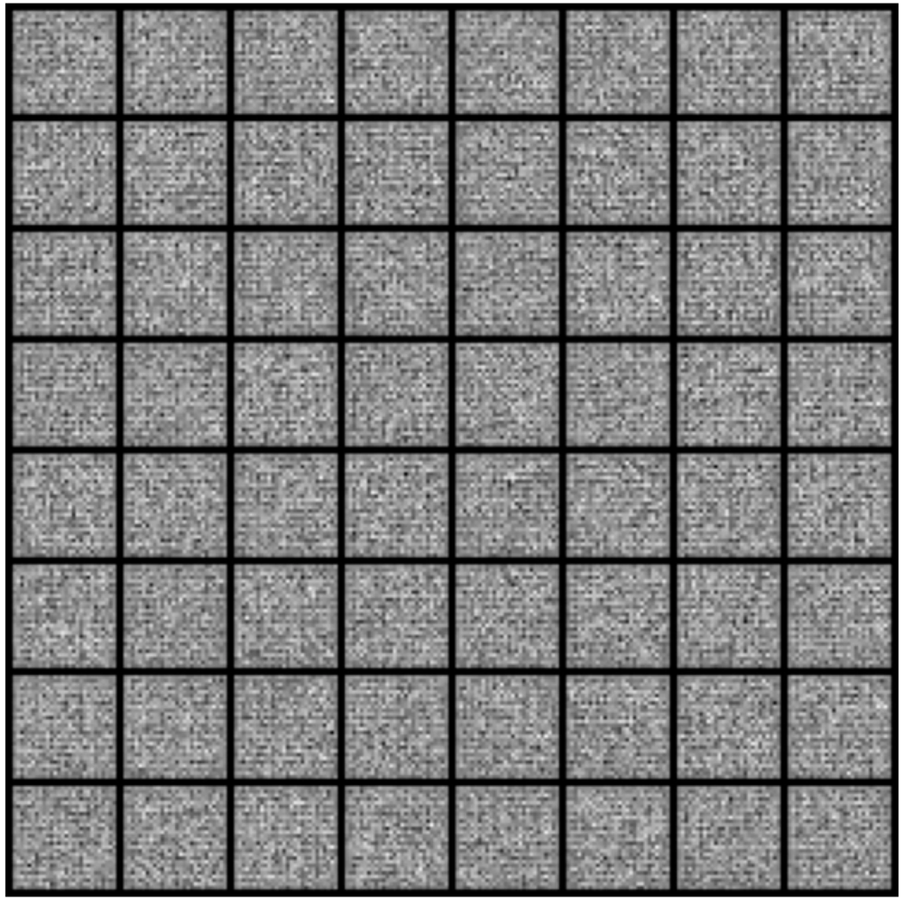
- We aim to apply GANs to individual alphabet images per user in order to ”learn to write” the corresponding user’s handwriting – alphabet by alphabet.
- We were however unable to find sufficient handwritten alphabet images corresponding to a single user and thus had to drop this attribute. We instead used the corresponding alphabet images data from the NIST dataset of all users combined.

Eg. Data for training on alphabet A corresponded to A’s images obtained from all users rather than a specific user.

LEARNING FROM SOCIETY – THE GAN

We then train on our data (the society) using GANs per alphabet. The video on the right shows the GAN "learning to write." We see that the initial output showed garbage values which gradually solidified into identifiable alphabet.

Note that we have used images of alphabet A obtained from all users rather than a specific user. Thus training upon alphabet images corresponding to a single user might yield better results due to it's similarities!



UNITED WE STAND – THE OUTPUT

- Upon training our GAN, the user then inputs the typed text who's font he/she desires…And Viola! The output is the handwritten font of the inputted text in the form of an image!
- Our model breaks down the sentence into individual alphabets and spaces, spits out the generated handwritten font per alphabet and finally concatenates (unites) the alphabets and spaces in order!

Input: INPUT EXAMPLE

Output: 

PART 2: HANDWRITING FRAUD DETECTION

1. Beginning Again
2. The Detector

Handwriting forgery is the process used by criminals to fraudulently make, alter, or write a person's signature—so that in most circumstances it appears identical with the genuine signature—with the intent of profiting from the innocent party. The consequences of being convicted of handwriting forgery are usually much less than many other major crimes, with the specific punishment generally being in the form of a misdemeanour and set by various state and federal statures. As a result of these minor consequences and because most people are uninformed of the various tactics used by such skilled criminals, the illegal activity of handwriting forgery is growing at a higher rate than most other crimes.

The act of handwriting forgery annually takes many millions of dollars in cash and property from victims, primarily in the form of fraudulent checks, credit card purchases, invoices, identification papers, and passports. Within the field of forensic science, investigative experts use scientific handwriting analysis to examine the legitimacy of signatures and legal identifications.

We thus intend to contribute to the field of forensics sciences by reducing human effort and optimizing fraud handwriting detection process with the help of our code.

Fact check:

https://www.encyclopedia.com/science/encyclopedias-almanacs-transcripts-and-maps/handwriting-forgery

BEGINNING AGAIN – DEFINING TRAINING INPUTS

- Here, again, owing to the lack of a proper dataset, we consider a slight variation yet keeping the concepts involved intact.
- Our inputs here are signatures of users instead of handwritten text, but we believe that it can be generalized to the later.
- The generalization can occur by detecting handwriting forgery per alphabet by performing image segmentation and then data augmentation to obtain a format similar to the NIST dataset used in Part 1.

Label: User 049 (unforged)

THE DETECTOR– SIAMESE NEURAL NETWORKS

- We now check whether a given signature (image) is forged or not by measuring the Euclidian distance between the vectorized original signature and the given signature.
- We perform this classification with the help of a Siamese Neural Network.
- Our final output looks something as follows:

Output: forged

PART 3: FUN EXPERIMENT

In this section we had planned to have a competition between the handwriting generator and the handwriting forgery detector to see if the generator can generate a handwriting that would be declared unforged by the detector.

Owing to this, the losing model could then be retrained to beat the other. We had planned to perform this experiment by building a GAN with our Part 1 model as the generator and the Part 2 model as our discriminator.

However due to the lack of sufficient time, we shall leave this under the heading of ”Future Prospectives” for this project.

THE END