

AI-Generated Self-Portraits

Time: 2019

Material: programmed digital imaging with Python

Size: 256px by 256px (each of 5)

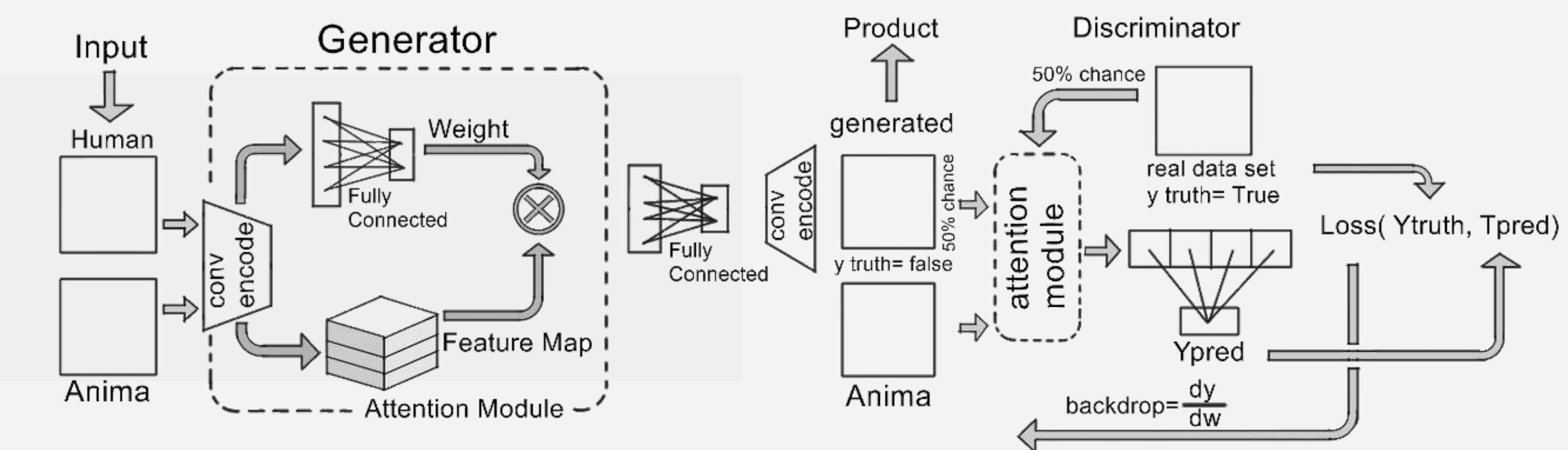
Reference Research Paper: “UGATIT: Unsupervised Generative Attentional Networks with Adaptive Layer-Instance Normalization for Image-to-Image Translation” (2019)

Dataset Used:

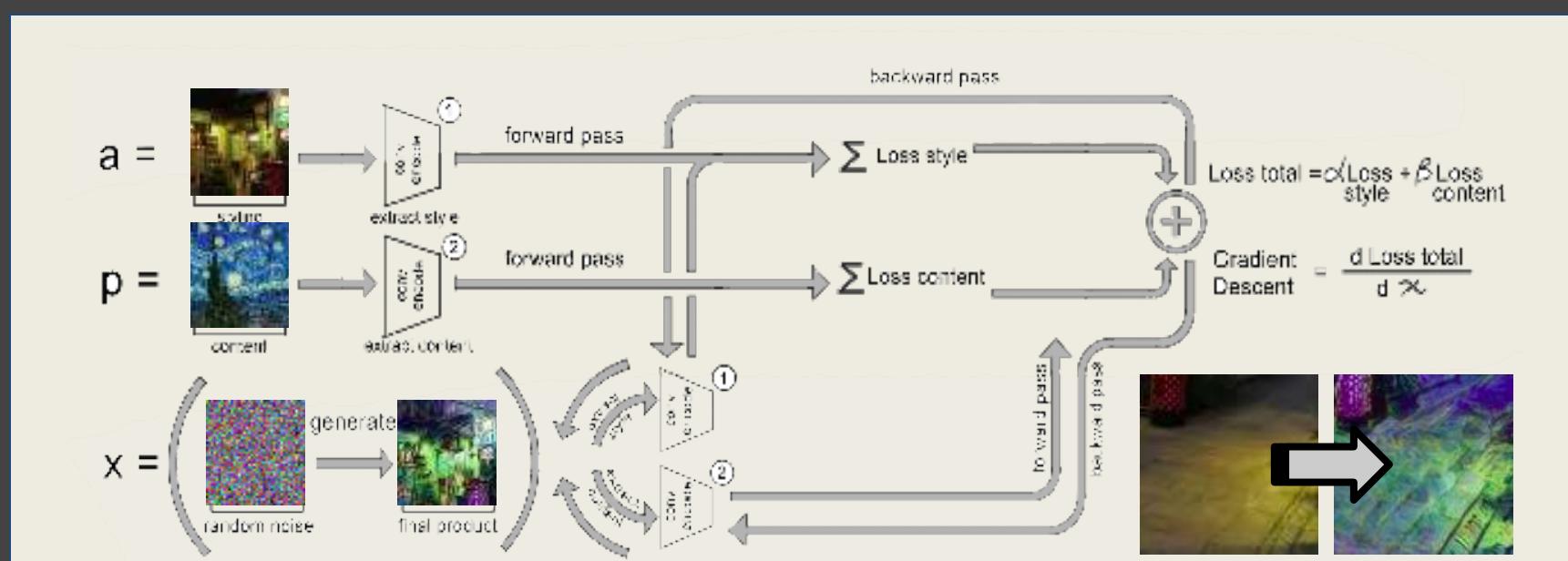
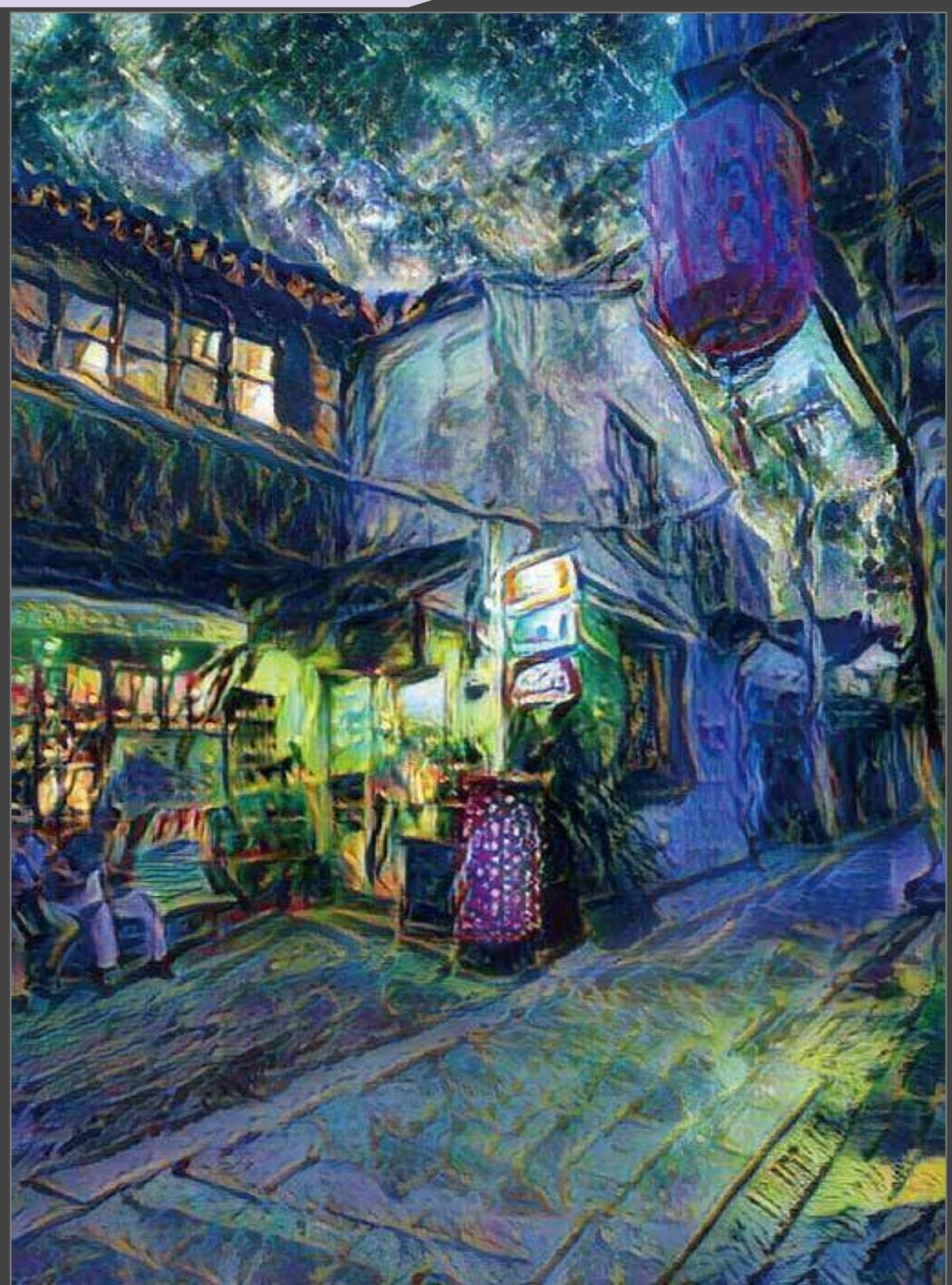
- animeface-character-dataset
- selfie-dataset

Modern psychology tells us that human vision is not mere copying of reality, rather, our mental images are heavily processed by our brain. Therefore, I was interested in how Artificial Intelligence perceives things. This artwork reflects literally how an AI sees me.

I trained and deployed my Neural Network according to sample code and methods in an unpublished paper in Computer Vision. The resulting images are generated by my code. (the training process is shown on the network architecture diagrams)



Network Architecture: Generative Adversarial Network (GAN) used to produce art



AI as My Brush

Material: programmed digital imaging with Python

Reference Research Paper: Image Style Transfer Using Convolutional Neural Networks (CVPR 2016)

I deployed the Deep Learning algorithm presented in the paper to generate images with my code.

1. Starry Town

Time: 2019

Size: 768px by 512px

Allusion To: Vincent van Gogh: "Cafe Terrace at Night"

*Walking in an old town under the sunset,
cafe shops lit up into the bustle,
only with the starry sky still.*

2. Chinese Garden

Chinese gardens have unique relationships with nature. The placements of rocks, ponds, and artificially planted trees immerse the viewers fully into nature. Here, I tried to show this aspect of Chinese gardens by blending the plants, rocks, and architecture together with an abstract painting using AI technology.

3. Water Township

This is an old town: People who live outside want to get in, while people who live inside want to get out.

- A Nostalgic Feeling of My Hometown from Visiting Suzhou Watertown



2. Chinese Garden

Time: 2018

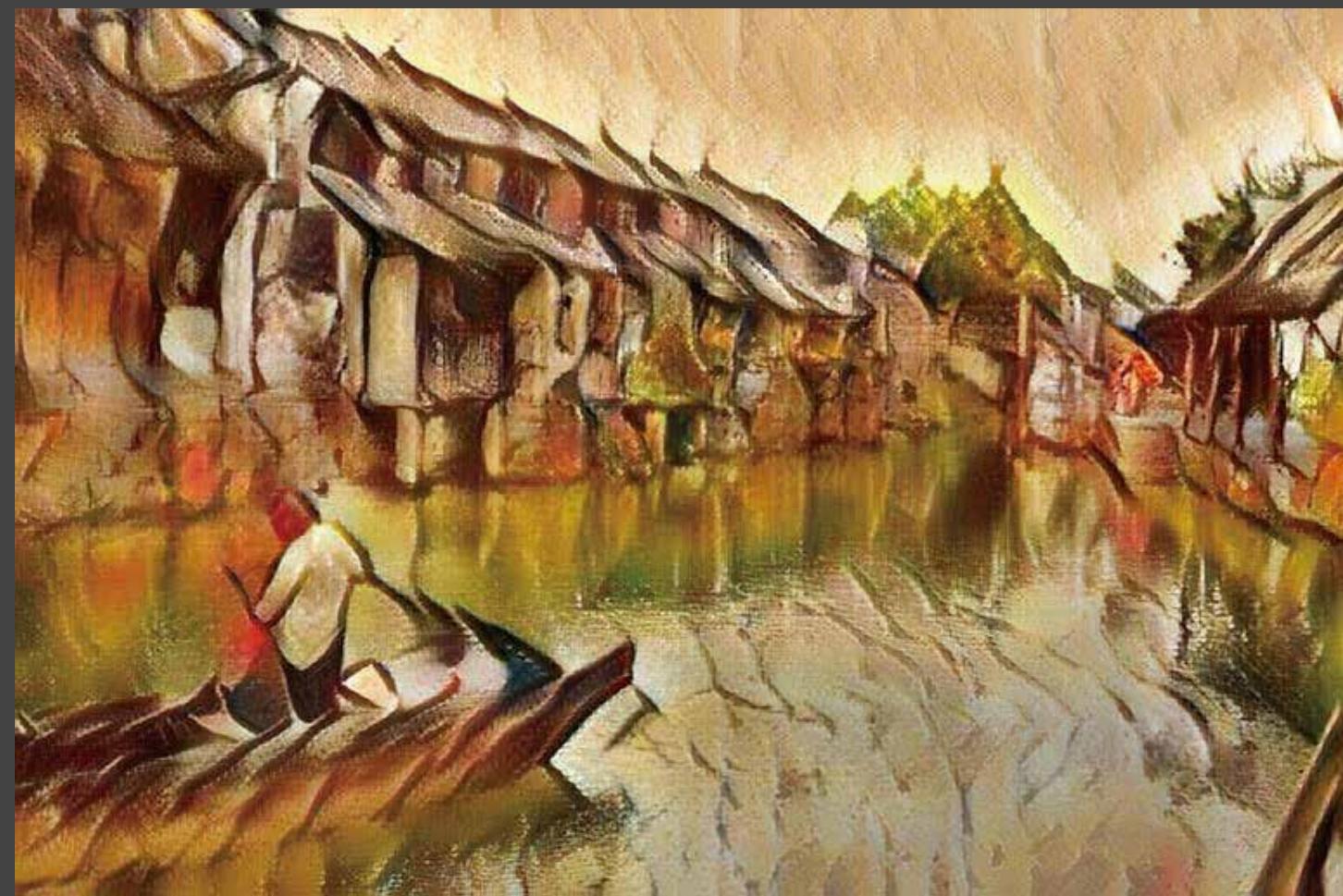
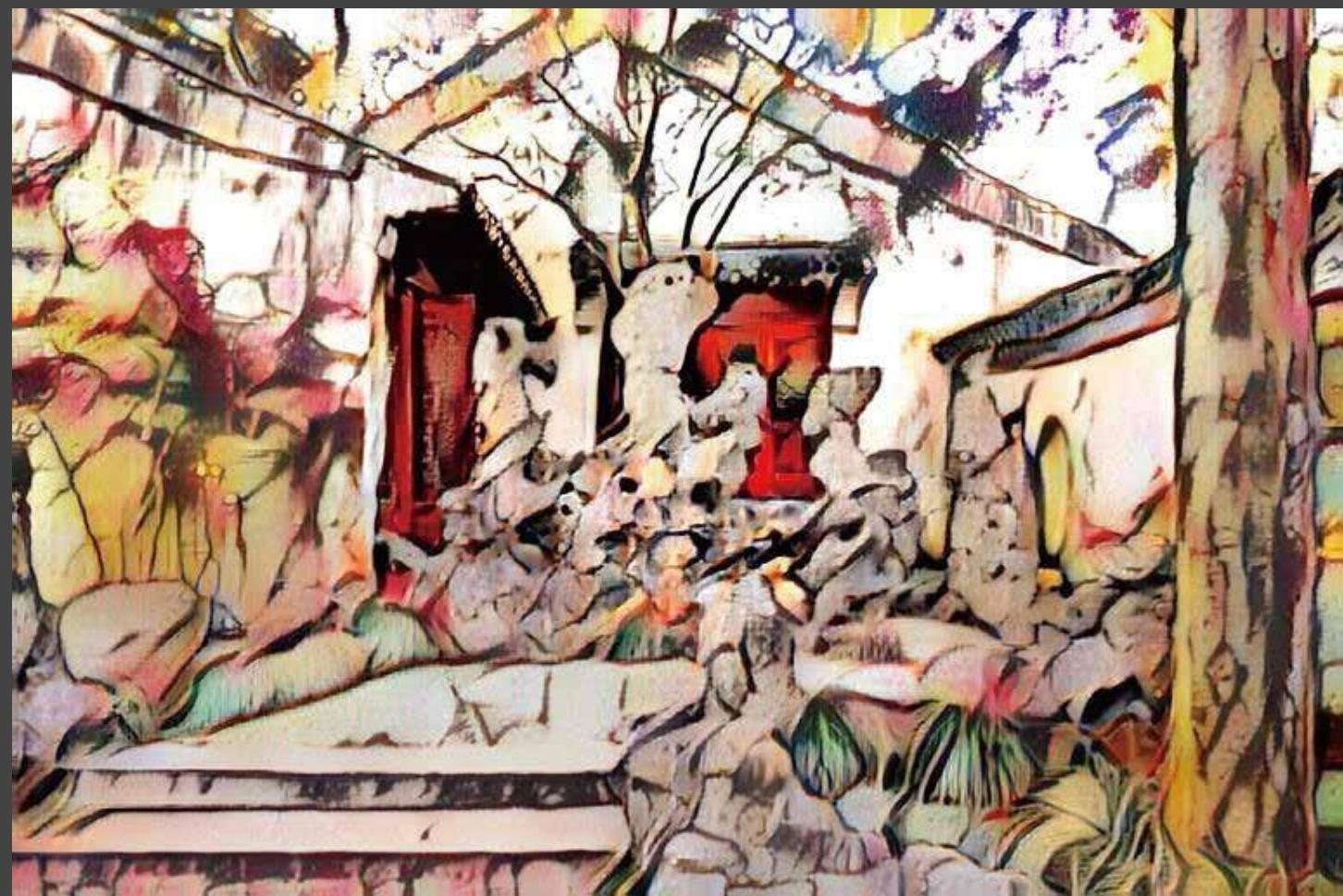
Size: 512px by 768px



3. Water Township

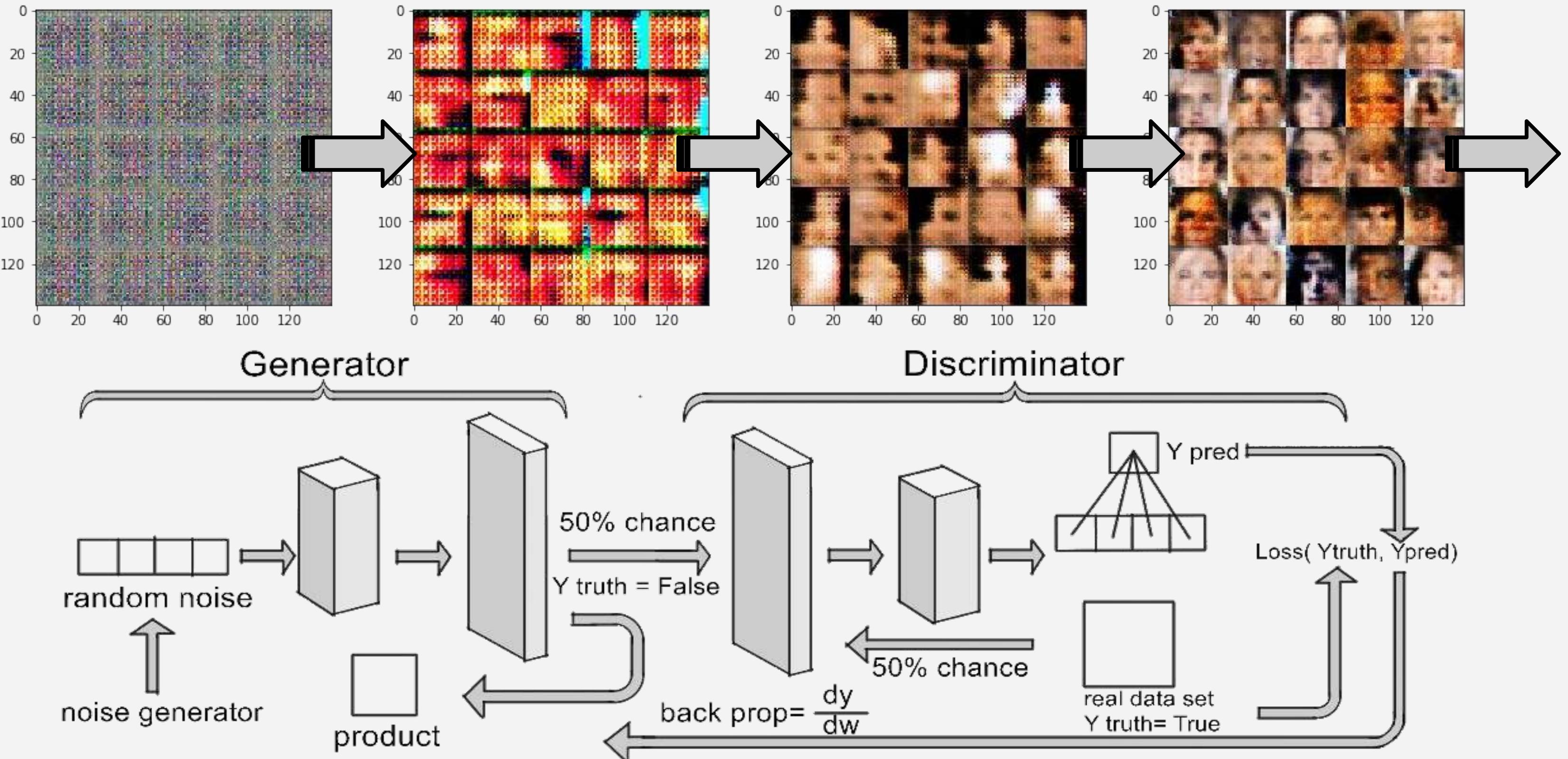
Time: 2018

Size: 512px by 768px



AI-assisted Art

Training Process: Images trained from random noise gradually become recognizable.



Network Architecture: Generative Adversarial Network (GAN) used to produce art

Fake News

Time: 2019

Material: programmed digital imaging with Python

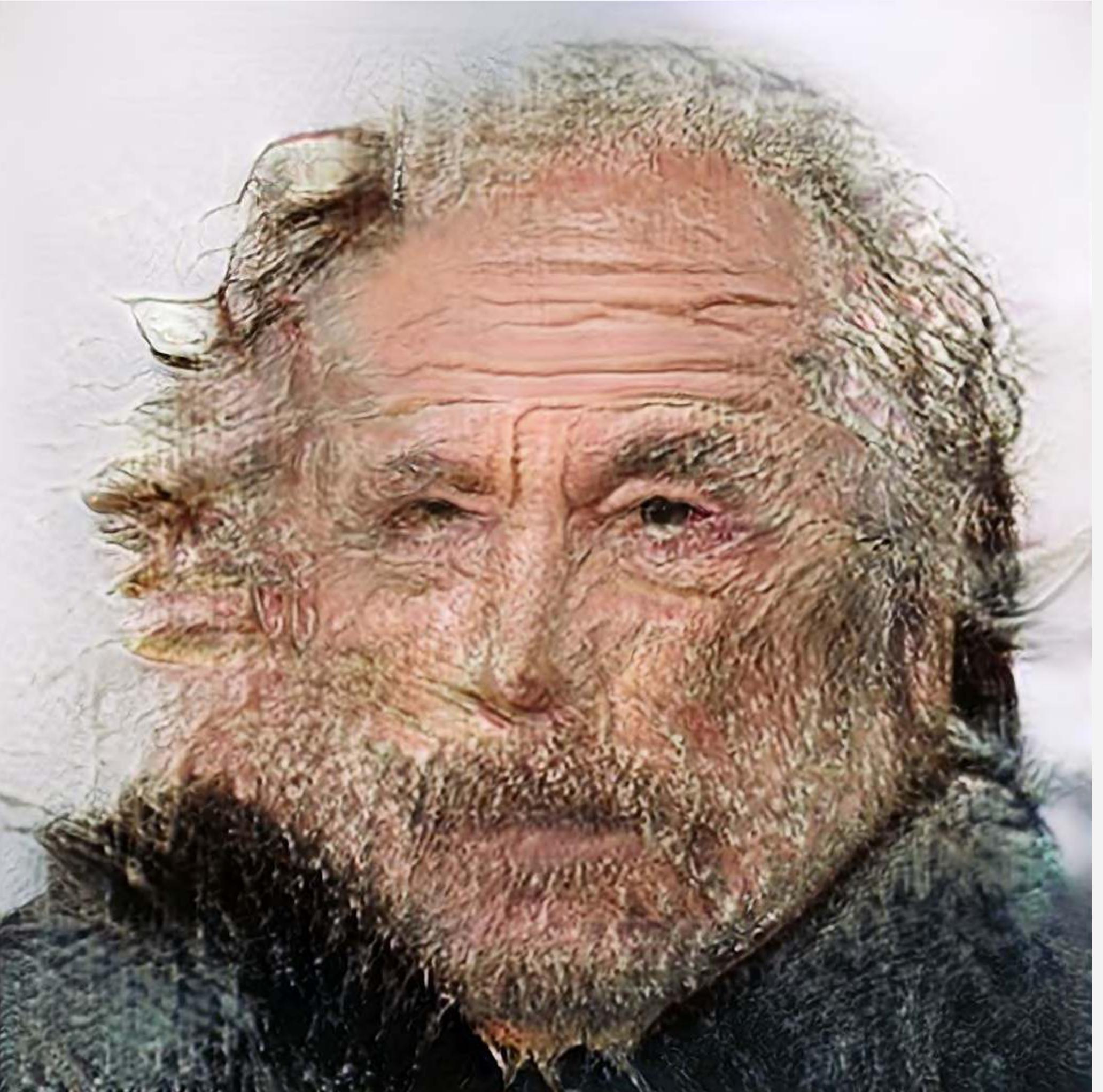
Size: 1024px by 1024px (each of many)

Reference Research Paper: "Generative Adversarial Networks" (2016)

Dataset Used: celebA

These paintings drawn with my AI algorithms was generated completely from random-noise inputs, which means that these people do not actually exist on earth.

I trained my algorithm and generated this piece to demonstrates the power of AI and how the growing technology can create problems like the "DeepFake crisis," generating distrust in our society.



Oracles

Time: 2019

Size: 40,000 characters with 2 types of font and various sizes, creating 3 pages of academic paper

Material: programmed digital imaging with Python, printing paper

Dataset Used: Meiling Han Oracle dataset created from Meiling Han's "Words from Heaven" by myself

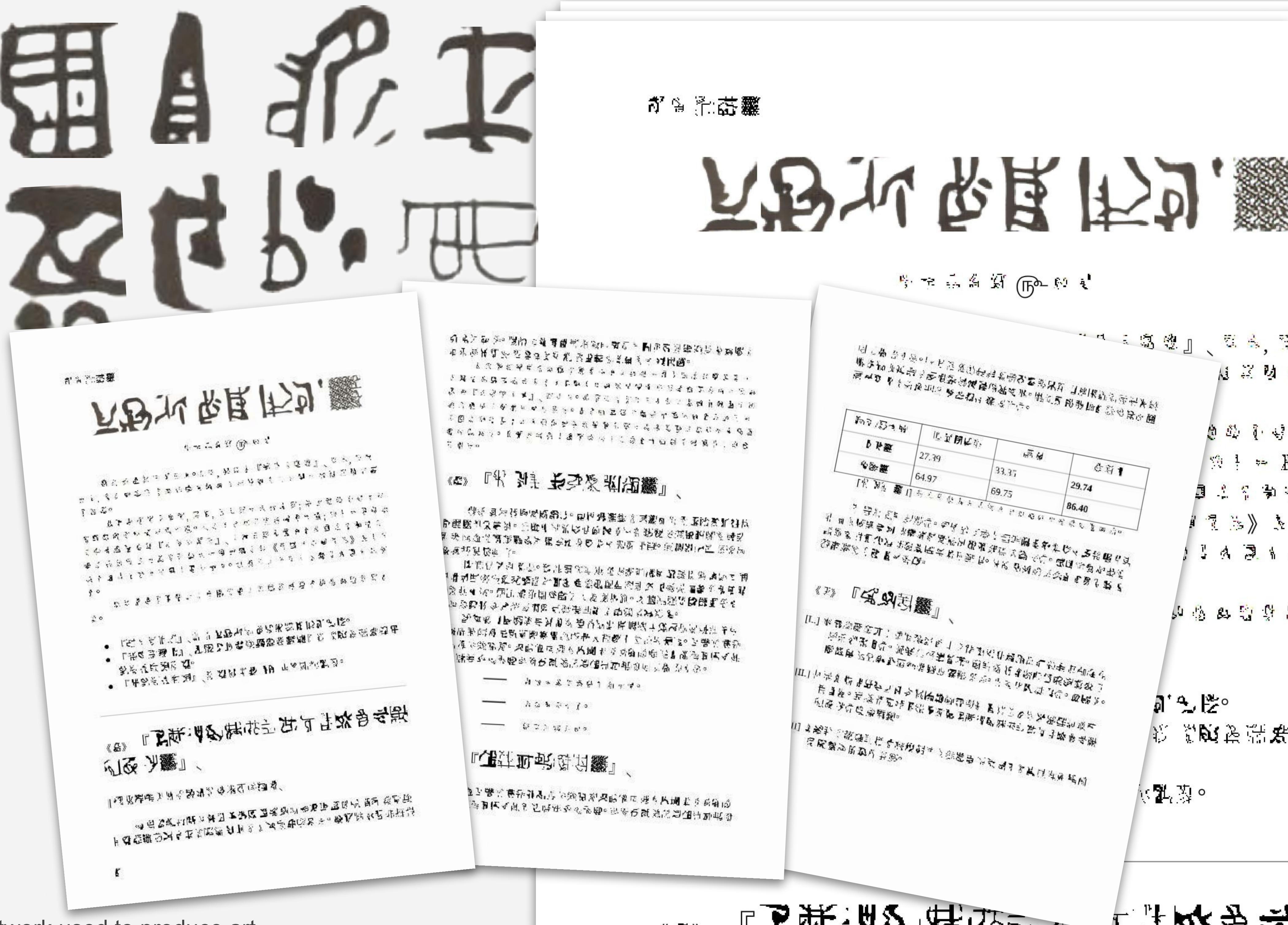
Allusion To: Book From the Sky by Bing Xu

I trained an AI model to generate 40,000 fake characters that don't exist in real life, and used them to make an academic paper.

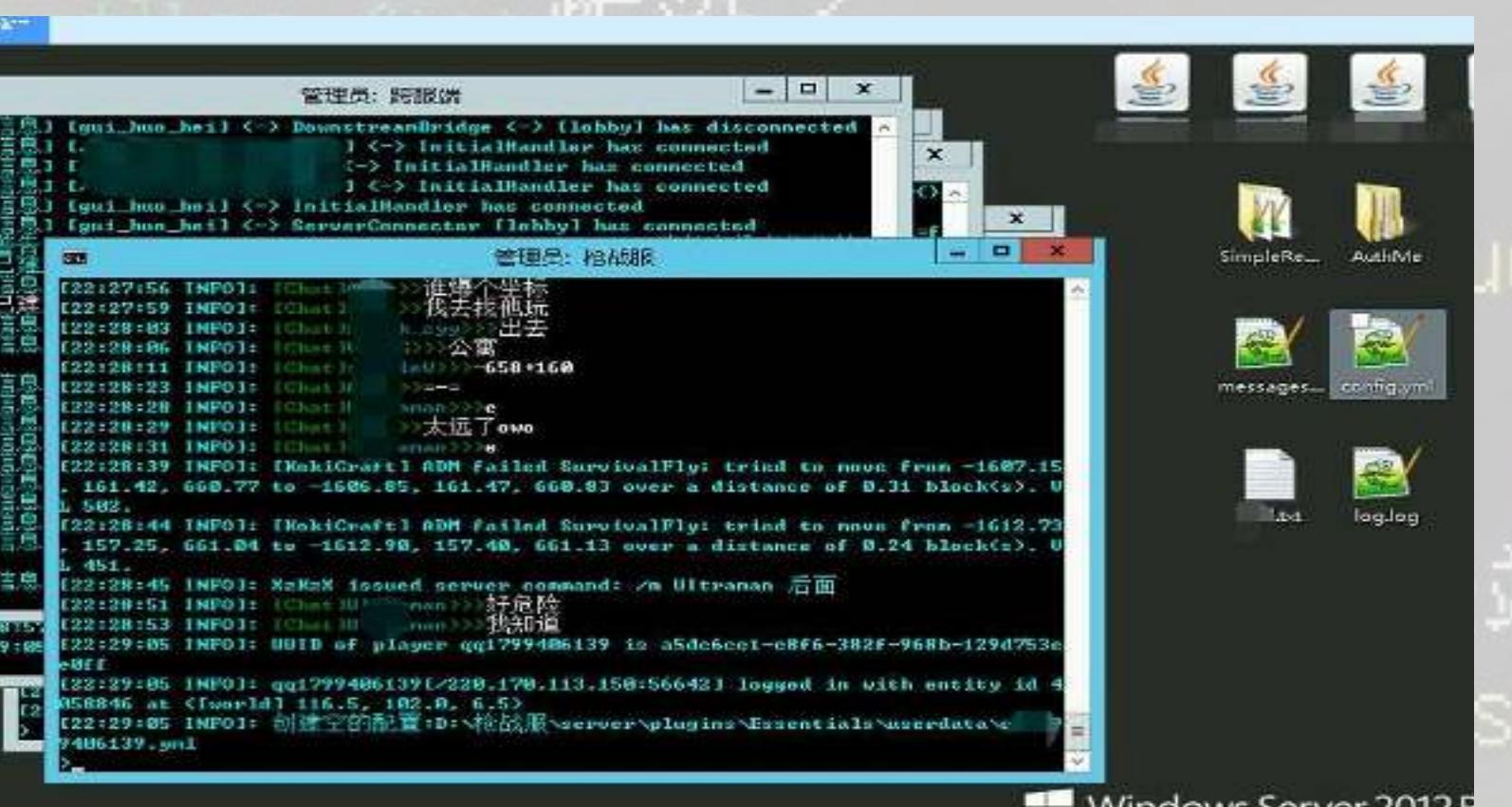
This work first makes the readers believe that they are reading a paper in an ancient language still in use in Asian countries. Then, after a closer look at the description, the readers start to find out that these words are computer-generated nonsense. They are tricked by the formality of the paper.

I used this work to criticize the restriction of our writing format put on by our society. The overemphasis on structure and formality of academic papers often leaves out the main message. After all, they convey nothing more than Oracles.

Algorithm: Generative Adversarial Network used to produce art



Game Design



Server's backend: where I host the 10 sub servers & SQL database



Time: 2014-2016

Material: programmed digital imaging with Java, Photoshop

Game Based On: Minecraft

***Team Project:** my players contributed to making suggestions

***My Position:** business owner, game designer, software programmer

Video: <https://www.youtube.com/watch?v=PhJq5YnzfUo&t=1s>

At 14, I created “KokiCraft”, a game server that transformed Minecraft into a Grand Theft Auto-esque game with more player involvement and elaborate storyline through programming. It was 1st GTA Minecraft Server in China according to MCBBS, generating ~\$500/month in revenue while serving close to 350,000 players with 10 sub-servers of different gameplay. I maintained monthly updates to introduce new elements in the game for better user experience. I even recruited a customer support team from authorized players.



Interface Design: An animated lottery window



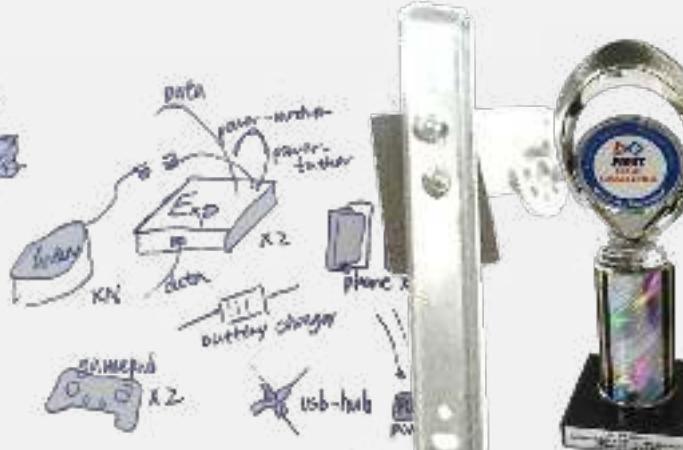
Inventory



Machinery / Installation



doogleing:
prototypes of the
robot designs



Judges' award:
From 2018 FTC
Robotics
Competition for
team's "unique
efforts,
performance or
dynamics merit
recognition."



Team Item: 3D printed

BEESTBot

Time: 2017-2019

*Team Project: working with 2~20 teammates in 3 years.

*My Position: team leader, main hardware, software and electrical design, design director

Link: <https://www.youtube.com/watch?v=BeDeAluq7HQ>

In my 10th grade, I built our school's first STEAM community from ground-up. It is fascinating to see our team spirit from all grades unite together on one project. I worked for 3 years on the robot's design, including implementing machine learning for object detection and adjusting motors' gear-ratio for hooking and lifting itself from the ground.

This autonomous robot is capable of picking and transporting "gold" and "silver" minerals; lifting itself onto a "rocket"; and landing to the "moon".

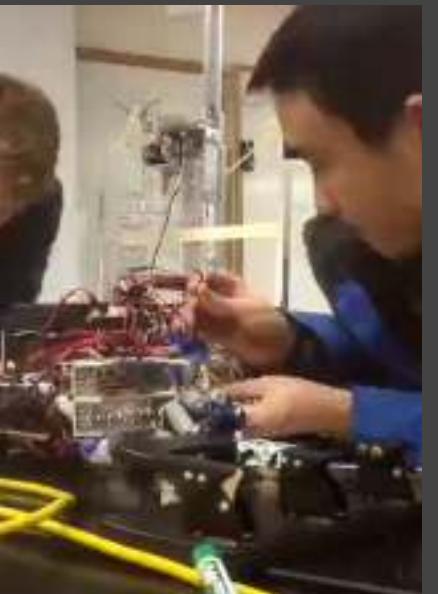
1. Robot Design

Material: Java programming language, steel, motors, servos, rubber bands, sensors, acrylonitrile butadiene styrene (for 3D printing), etc...

Size: 18 inches by 18 inches, height varies

2. Logo and Team Uniform Design

Material: digital imaging, pre-shrunk cotton, poly/cotton blend (for t-shirt & hoodie)



Programmed Art



Hanke Chen

How romantic it is to learn things together with my AI model on weekends —
[2019/02/02]

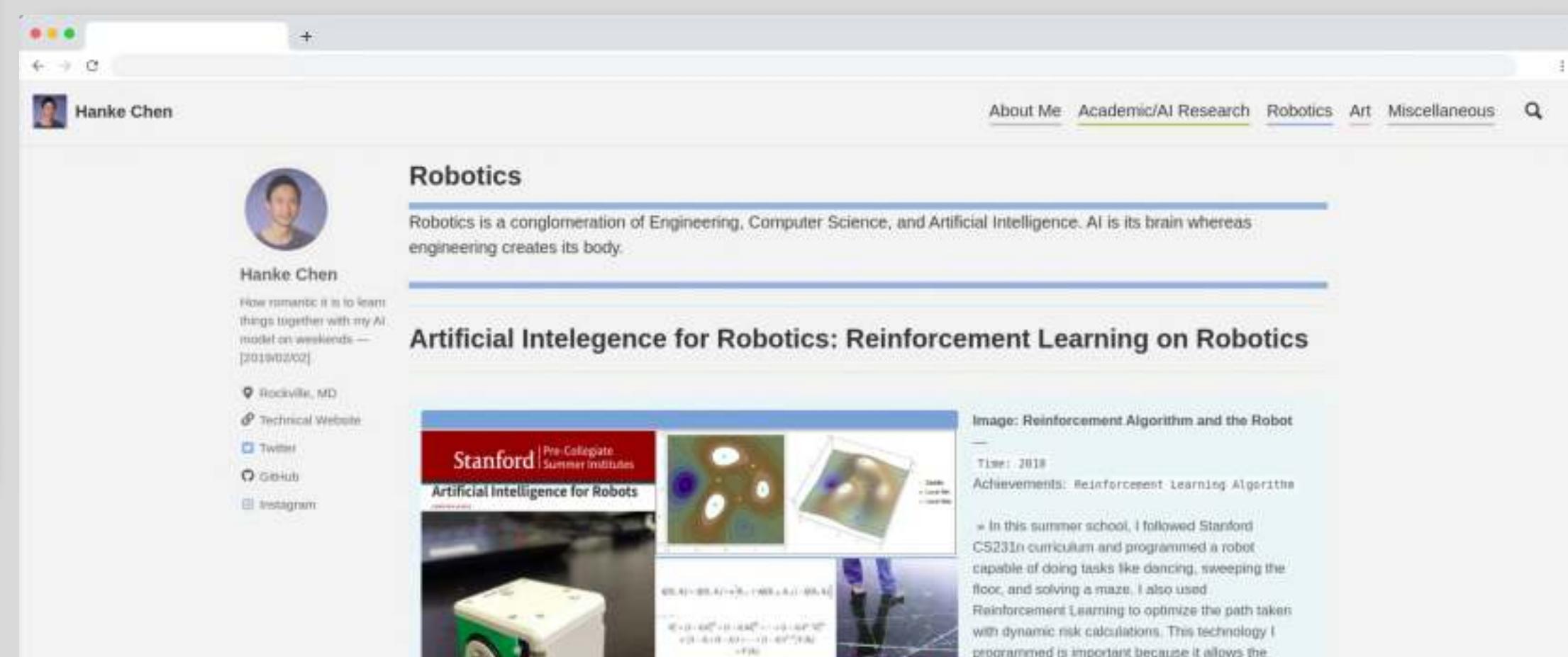
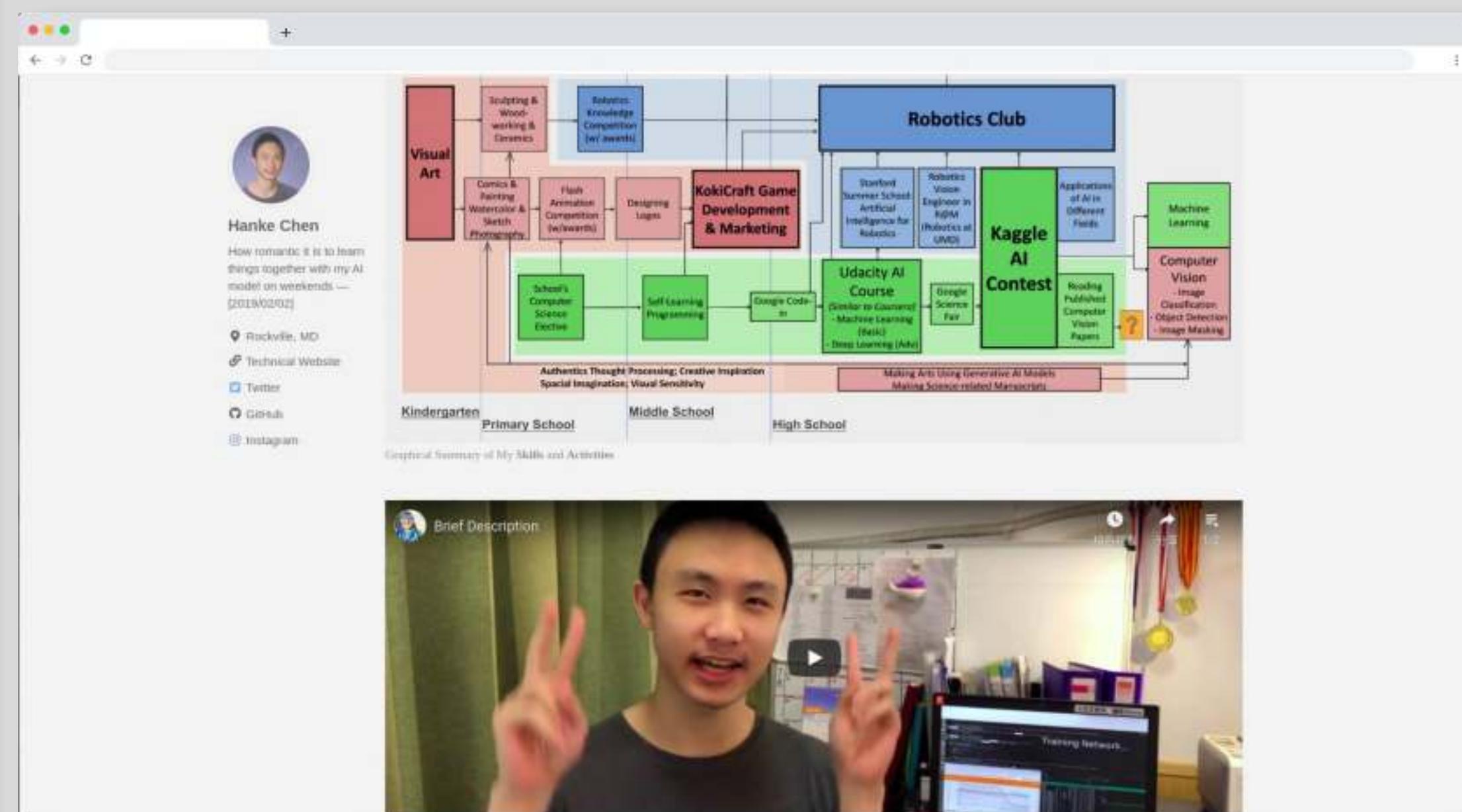
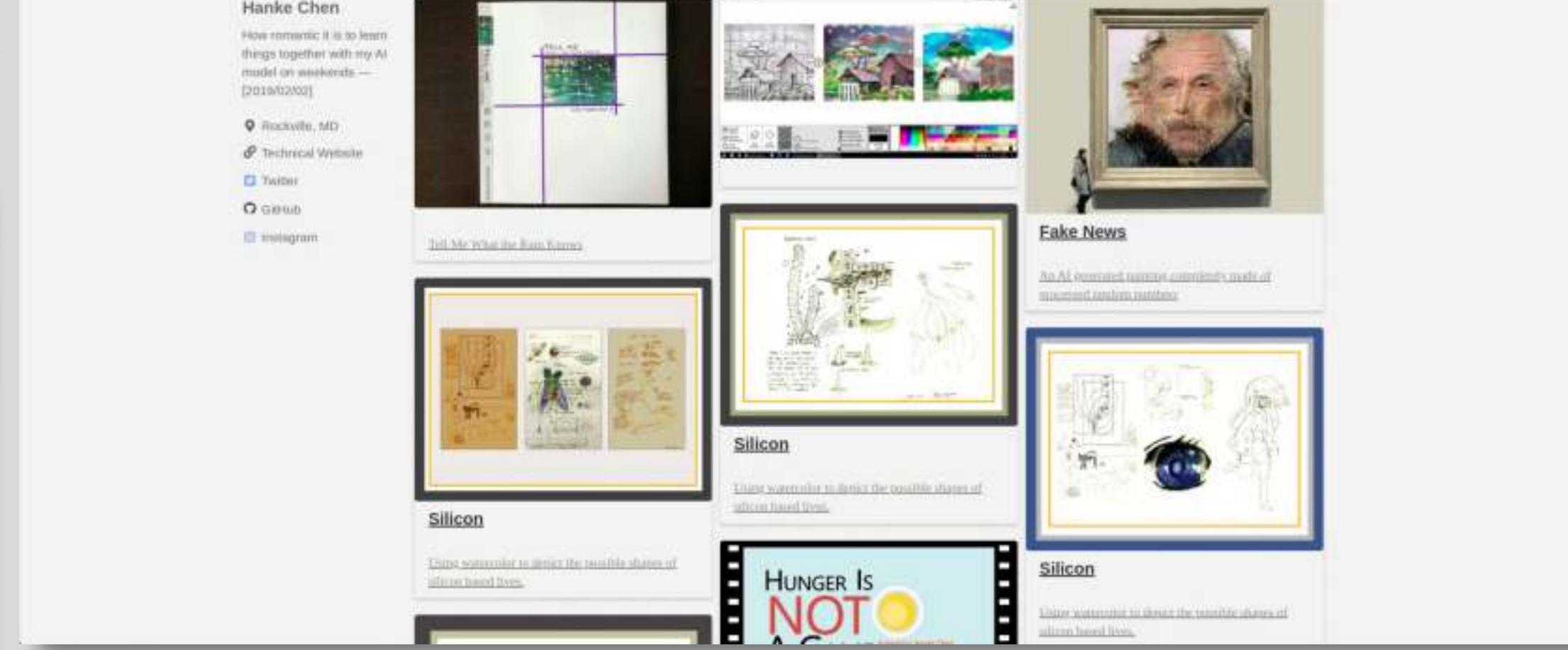
📍 Rockville, MD

🔗 Technical Website

🐦 Twitter

🐙 GitHub

📷 Instagram



About Me Academic/AI Research Robotics Art Miscellaneous

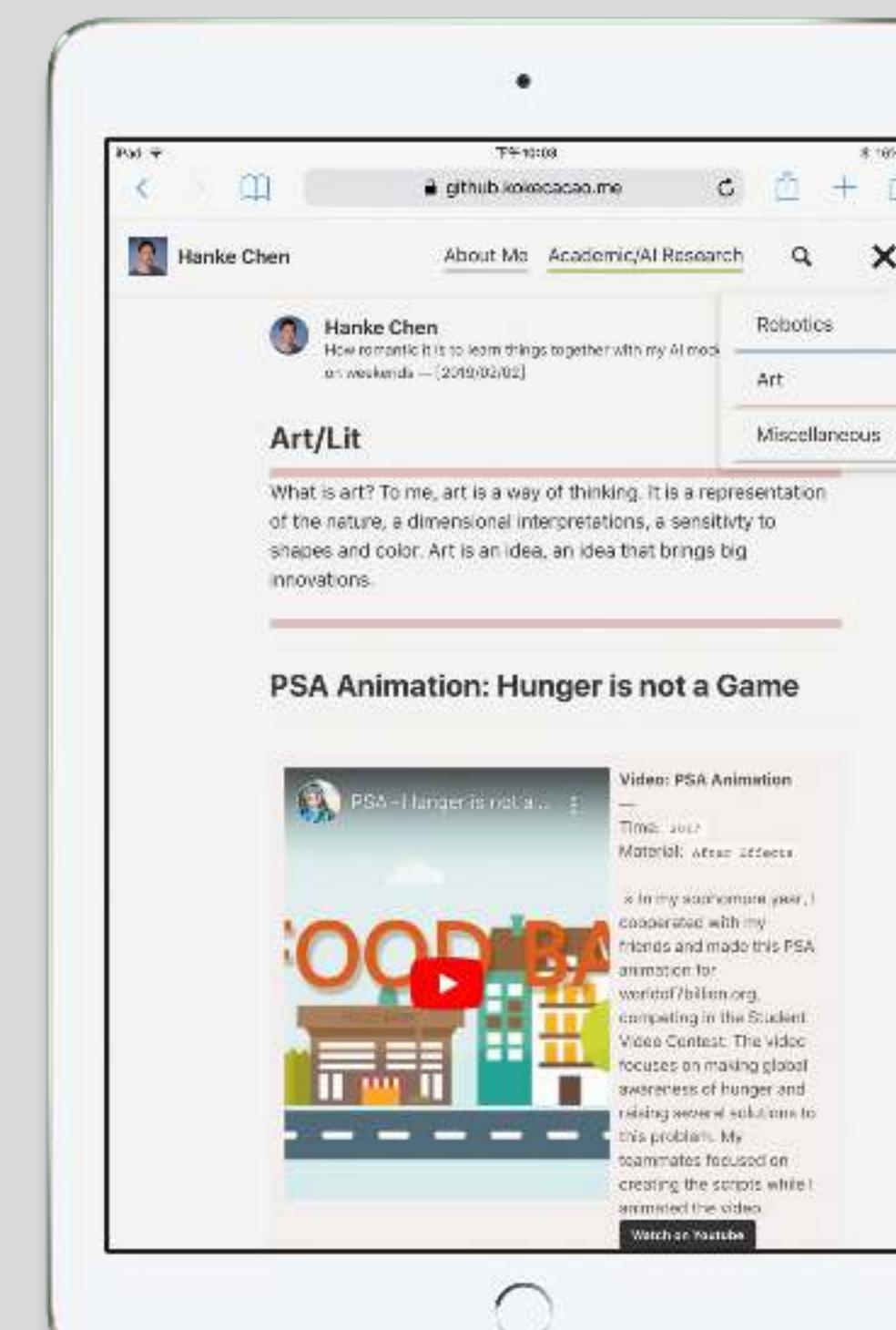
Website Design

Time: 2018

Material: programmed digital imaging with ruby, html, css, jekyll

Link: <https://chenhanke.me>

This general web page coded by me showcases every aspect of myself from AI Research, Robotics, to Art and Game Designs.



Responsive UI
Design: automatic resize website on different devices

Brownie UI: A Personal Tech-Webpage

Time: 2017-now

Material: programmed digital imaging with html, css

Link: <https://www.kokecacao.me>

This geek-style minimalistic design is an expression of my personal values and how I show them to my friends. The use of line, shape, hue, and saturation gives the viewer a sense of intimacy.

The screenshot shows a web browser window with the Koke Cacao website. The sidebar on the left contains links to various platforms: Github, Bilibili, Instagram, Kaggle, Twitter, Zhihu, WeChat, Youtube, and Email. The main content area features several sections:

- Game Design**: A section featuring a game controller icon and the text "KokiCraft is my first for game design. It".
- Computer Vision**: A section with an eye icon and text about taking Udacity's Deep Learning course and participating in Kaggle competitions.
- Hamster**: A section with a hamster icon and text about artificial intelligence being programmed on small devices.
- Codings**: A section listing programming languages and tools: Java, Python, Android, SQL, html, css, Lua, Github, Tensorflow, sklearn, OpenCV, Linux(Kali, Ubuntu), and Pytorch.
- Properties**: A section describing the author's interests and habits, mentioning Quantum delay experiment's strange bug, being extremely introvert but talkative, loving science and tech, watching anime, playing video games, using Sublime Text and Vim, learning Pytorch, using Ubuntu as desktop, and Occam's razor as the fundamental theory of science.
- MeaningOfLife.java**: A code snippet from a Java file.

Bitcoin Rating Visualization

Time: 2019

Material: programmed digital imaging with Gephi

Size: vector image (each of 2)

Dataset Used: Bitcoin OTC trust weighted signed network

Aren't we all connected in some ways?

Because the Bitcoin system is anomalous, there should be a way to track the reputation of each user in the system. Here, I present data visualization of trustworthiness by projecting high dimensional data to 2D graphs using position, length, area, and color cues.

Each color represents a distinct group of users that trust each other. The "authority" in the groups are labeled as bigger dots, and users who trust each other are closer together.

I learned the algorithm of creating graph visualization in the Data Visualization online course by UIUC Master in CS on Coursera. The algorithm simulates gravity and iteratively moves points closer to or away from each other based on "ratings" provided by users in the dataset.

Algorithm: ForceAtlas 2

Algorithm: YiFan Hu
Proportional

