EDITH Midterm Presentation

Team D
Chingis
Eunmin Kim
Gong He
Soohun Park

Objective

- Provide an AI-Powered end to end Anime Styled Face Generation Web App
- Create faces for both Males and Females
- Provide users options to pick tags (facial expressions, male/female, etc) that they want to use
- Provide users BW images so that they can color it themselves or by AI
- Allows users to draw freely before and after image generation

Initial Schedule

- Initial Schedule as shown in our proposal

Month	Subtask	Person
Oct.	Image Data Collection	Chingis, Soohun
	Processing	Chingis, Soohun
	Creation and Design of Splash Screen	Eunmin
	GenerateSketch API	Gone He
Nov.	Train Generation Model	Chingis
	Colorization Model	Chingis
	Creation and Design of Screens	Eunmin, Gong He
	Adding Editing Tools	Eunmin, Gong He
	Colorize API	Soohun
	History API	Soohun
Dec.	Testing	All
	Bug Fixing	All

Remaining Schedule

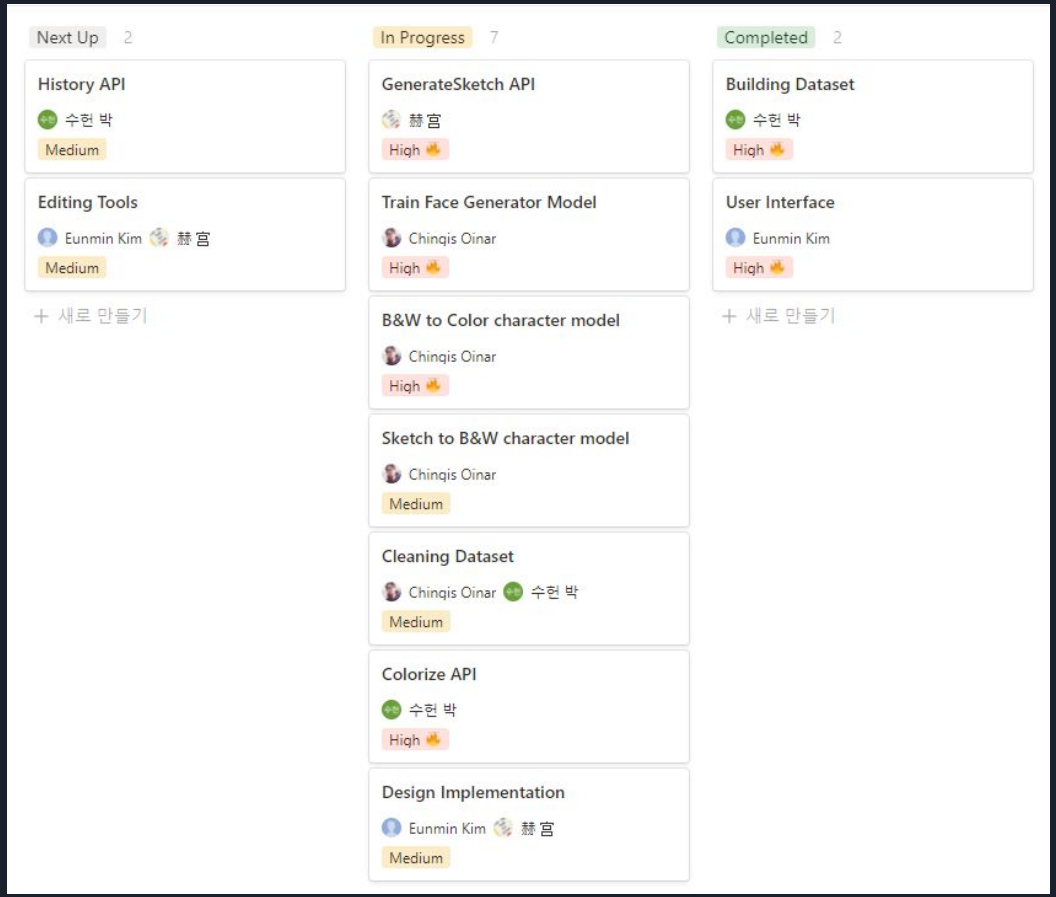
- Finished Image Data Collection, Processing, Design of screen
- Almost complete: Training of generation and colorization models, GenerateSketch API
- Doing: Frontend and backend coding (Jquery + Django/FastAPI)
- To be started soon: Colorize and history API, Adding Editing Tools

Month	Subtask	Person
Oct.	Image Data Collection	Chingis, Soohun
	Processing	Chingis, Soohun
	Creation and Design of Splash Screen	Eunmin
	GenerateSketch API	Gone He
Nov.	Train Generation Model	Chingis
	Colorization Model	Chingis
	Creation and Design of Screens	Eunmin, Gong He
	Adding Editing Tools	Eunmin, Gong He
	Colorize API	Soohun
	History API	Soohun
Dec.	Testing	All
	Bug Fixing	All

Role of Each Member

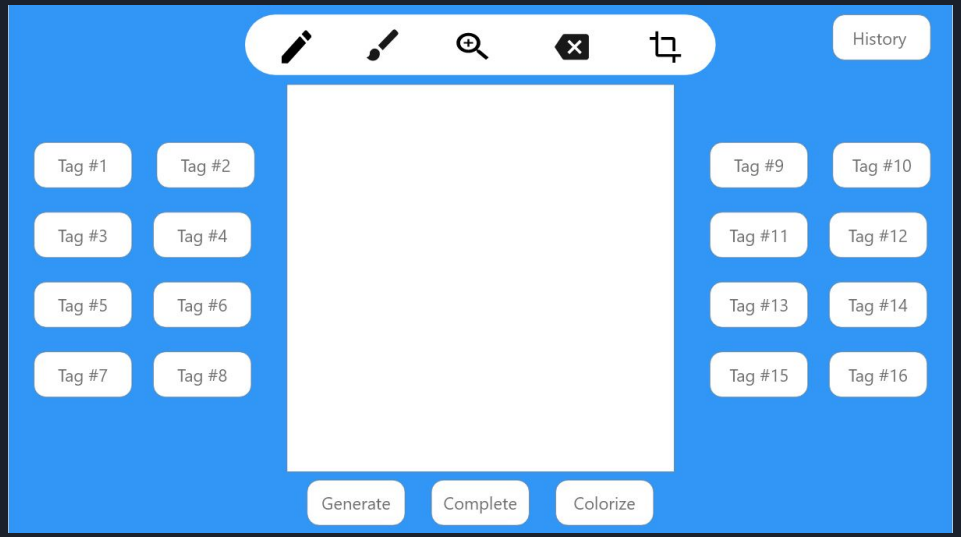
- Chingis (Team Leader): Processing, training GANs
- Soohun: Image Data Collection, Preprocessing & Cleaning, Backend
- Gong He: GenerateSketch API, Frontend + Backend
- Eunmin: Design of Screens, Frontend

How is it going?



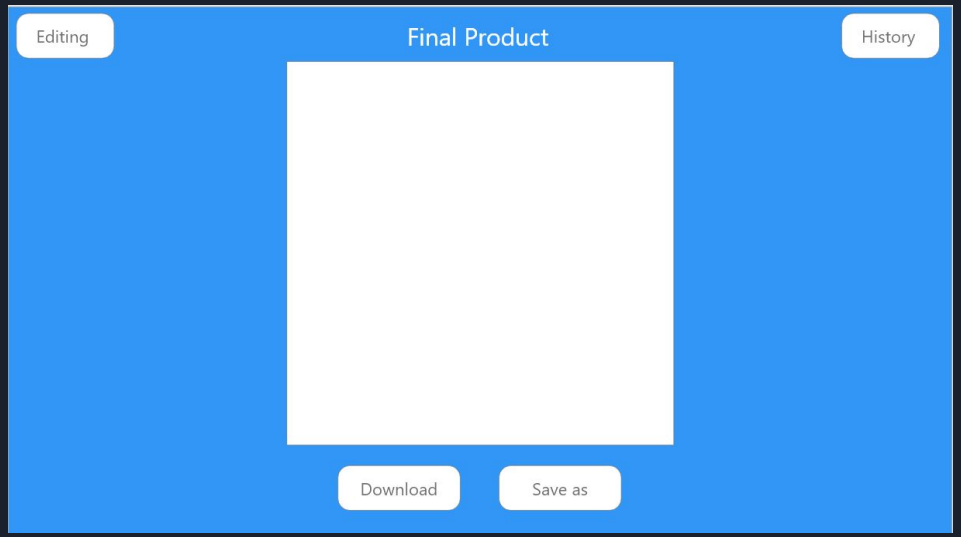
Design

Screen 1



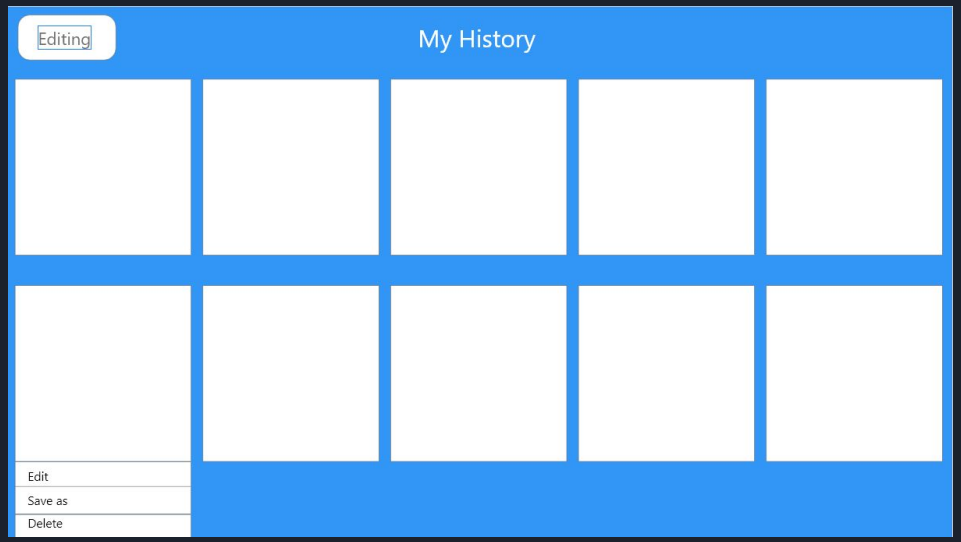
Design

Screen 2



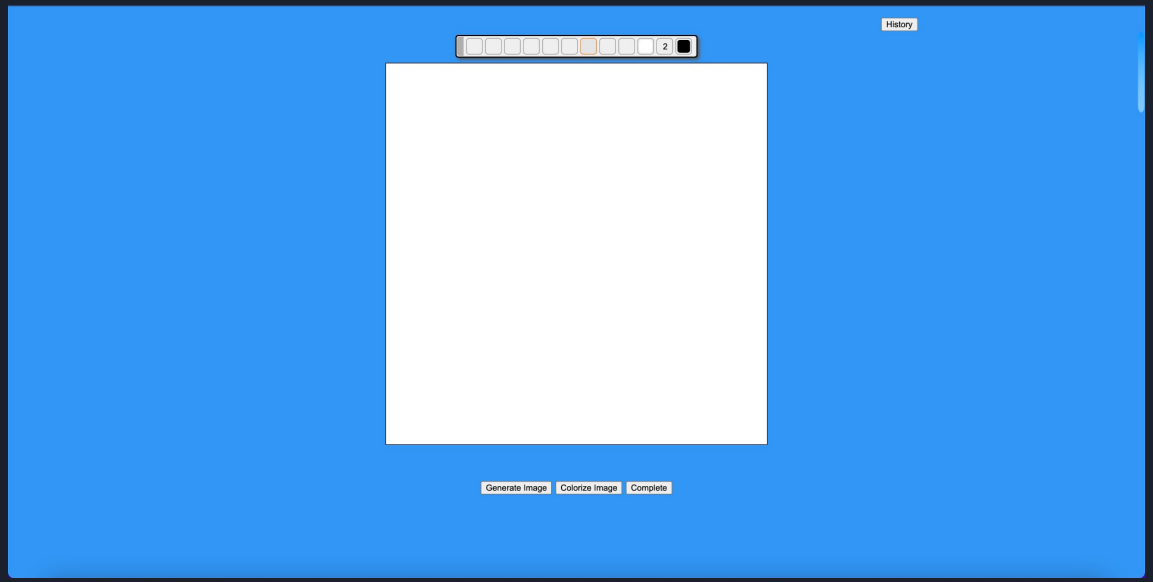
Design

Screen 3



Frontend Start

Frontend: Screen 1 (JQuery)

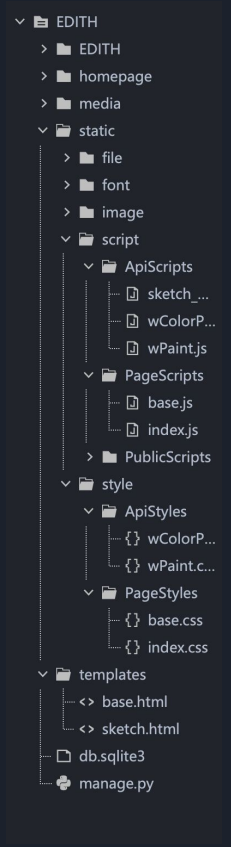


Implementation

- Frontend: JQuery
- Backend: Django / FastAPI
- ML/AI: Pytorch, AWS
- APIs: Fast API (with Python), AWS or Heroku
- Open Source Borrowed:
  Female dataset : Kaggle
  (https://www.kaggle.com/scribbless/another-anime-face-dataset)
  Face crop AI Model:
  (https://github.com/nagadomi/lbpcascade_animeface)
- Programming Languages: Python, Javascript

Backend

- Start Django project in Django 3.0.6 version.
- Start “homepage” app at Django project to process all page requests.
- Start “api” app at Django project to process all api requests.
- Add session energy to Django project setting.



Completed backend path

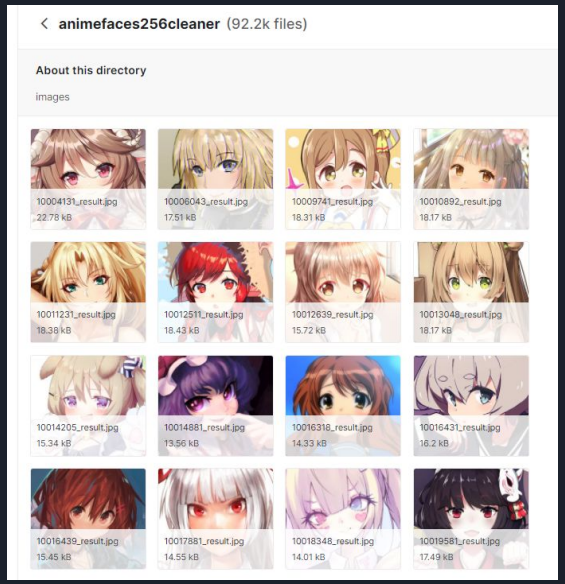
Editing Tools

- Free movement
- Undo
- Redo
- Clear
- Rectangle
- Ellipse
- Line
- Pencil
- Text
- Eraser
- Weight
- Color

Editing Tools

Dataset Preparation

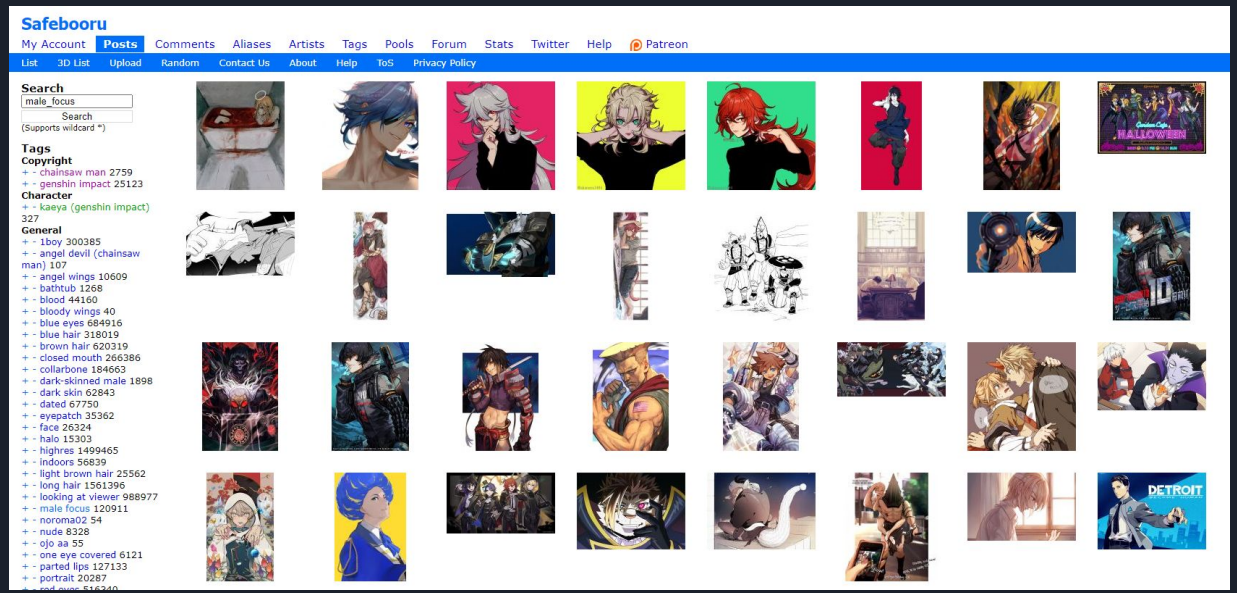
Plenty of female dataset(kaggle)



No male dataset

Dataset Crawling

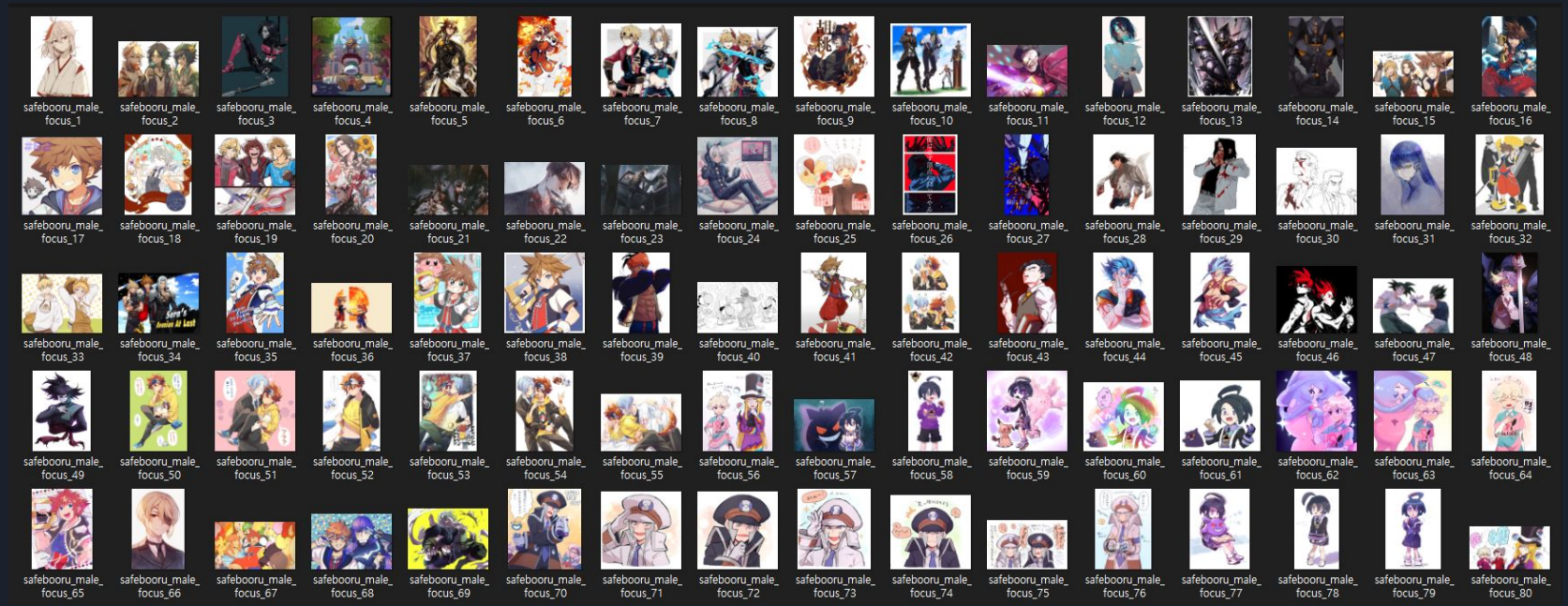
safebooru.org Keyword : **male_focus**



Dataset Crawling - First Try

26,636 images

BUT we need face dataset



Dataset Crawling - First Try

Anime-Face-Detector

A Faster-RCNN based anime face detector.

This detector is trained on 6000 training samples and 641 testing samples, randomly selected from the dataset which is crawled from top 100 pixiv daily ranking.

Thanks to OpenCV based Anime face detector written by nagadomi, which helps labelling the data.

The original implementation of Faster-RCNN using Tensorflow can be found here

30,112 images

Challenge #1: Dataset

Simply crawling images result in many noisy samples. Noisy samples include random crops or non-human images.

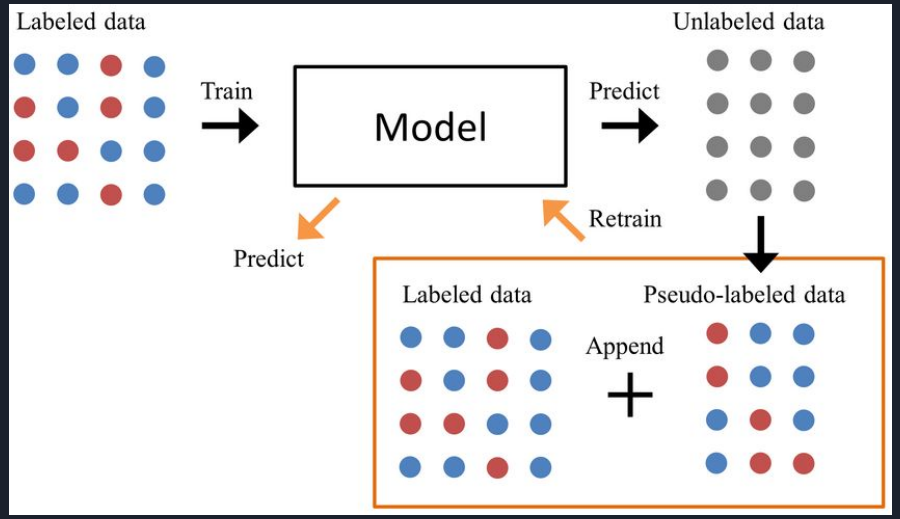
Violence

Non-human

Bad crops

Trial and error: Pseudo Labeling

It can distinguish male and females but it cannot distinguish good and bad samples.



Solution: Metric Learning

We can mine kNN of good samples by using embeddings of the images.

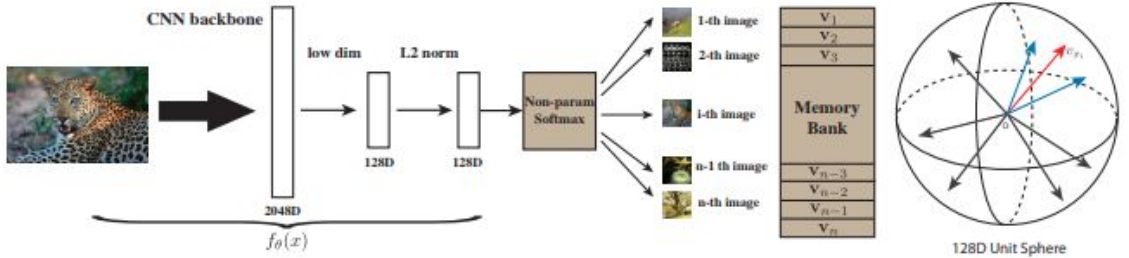


Figure 2: The pipeline of our unsupervised feature learning approach. We use a backbone CNN to encode each image as a feature vector, which is projected to a 128-dimensional space and L2 normalized. The optimal feature embedding is learned via instance-level discrimination, which tries to maximally scatter the features of training samples over the 128-dimensional unit sphere.

$$P(i|\mathbf{v}) = \frac{\exp\left(\mathbf{v}_i^T \mathbf{v}/\tau\right)}{\sum_{j=1}^{n} \exp\left(\mathbf{v}_j^T \mathbf{v}/\tau\right)},$$

Dataset Crawling - Second Try

Crawled 9,386 images → Cropped face part → 10,669 images

Crawled also the meta tags in the html file

```
safebooru_male_focus_1.jpg 1boy,bangs,black_vest,brown_hair,collarbone,collared_shirt,commentary_request,dress_shirt,eyebrows_visi
safebooru_male_focus_2.jpg 1boy,black_blindfold,black_gloves,black_legwear,blindfold,boots,child,gloves,jumping,long_sleeves,male_
safebooru_male_focus_3.jpg 2boys,ahat_(ragnarok_online),bangs,bat,black_coat,black_hair,blonde_hair,blue_eyes,choker,coat,commenta
safebooru_male_focus_4.jpg 1boy,blurry,blurry_background,boku_no_hero_academia,ceiling,commentary,english_commentary,forehead,from
safebooru_male_focus_5.jpg 1boy,ahat_(ragnarok_online),ahoge,bangs,blonde_hair,blue_eyes,blush,bow,commentary_request,corset,cross
safebooru_male_focus_6.jpg 1boy,ahat_(ragnarok_online),ahoge,alternate_color,bangs,blonde_hair,blue_eyes,blush,bow,commentary_requ
safebooru_male_focus_7.jpg 1boy,bangs,blue_eyes,brown_cape,brown_gloves,brown_headwear,brown_pants,brown_shirt,cape,card,card_in_m
safebooru_male_focus_8.jpg 1boy,bangs,black_hair,en&#039;en_no_shouboutai,hair_between_eyes,highres,looking_at_viewer,male_focus,s
safebooru_male_focus_9.jpg 2boys,ahat_(ragnarok_online),ahoge,animal_collar,animal_ear_fluff,animal_ears,bangs,blonde_hair,blue_ey
safebooru_male_focus_10.jpg 1boy,absurdres,bangs,black_hair,en&#039;en_no_shouboutai,fire,glowing,glowing_eyes,hair_between_eyes,h
safebooru_male_focus_11.jpg 1boy,animal_ears,bangs,belt,blue_shirt,blush_stickers,brown_bag,brown_belt,brown_eyes,brown_hair,brown
```

Total 12K images
Male : 6k images
Female : 6k images
Ready to train

Drawing steps

Machine Learning

Sketch to B&W Character Generation

Conditional Generation

B&W to Color Character Generation

Main hyperparameters

Epochs: 500 - 1000

Learning rate: 0.0002

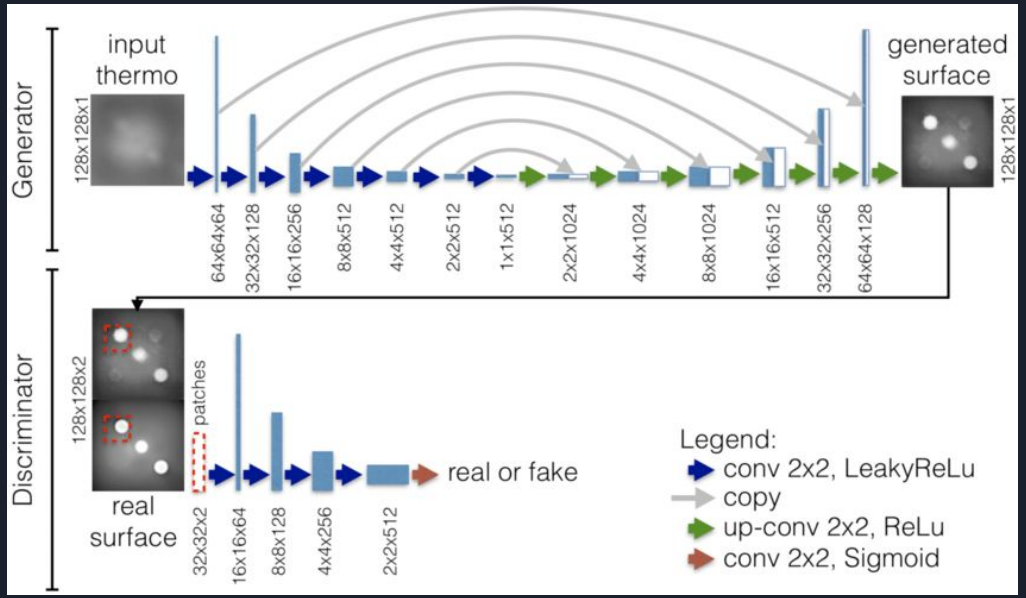
Optimizer: Adam with beta1 = 0.5 and beta2 = 0.999

Batch size = 128

Image size = 256

Pix2Pix GAN

“Image-to-Image Translation with Conditional Adversarial Networks” proposed by Isola et al. at CVPR 2017.



Training

Discriminator loss

$$\max_D \mathbb{E}_{\mathbf{x}\sim \mathbf{p}_{\text{Real}}}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim \mathbf{p}_{\text{Noise}}}[\log(1 - D(G(\mathbf{z})))] \quad (1)$$

Generator loss

$$\min_G -\mathbb{E}_{\mathbf{z}\sim \mathbf{p}_{\text{Noise}}}[\log D(G(\mathbf{z}))] \quad (2)$$

Reconstruction loss

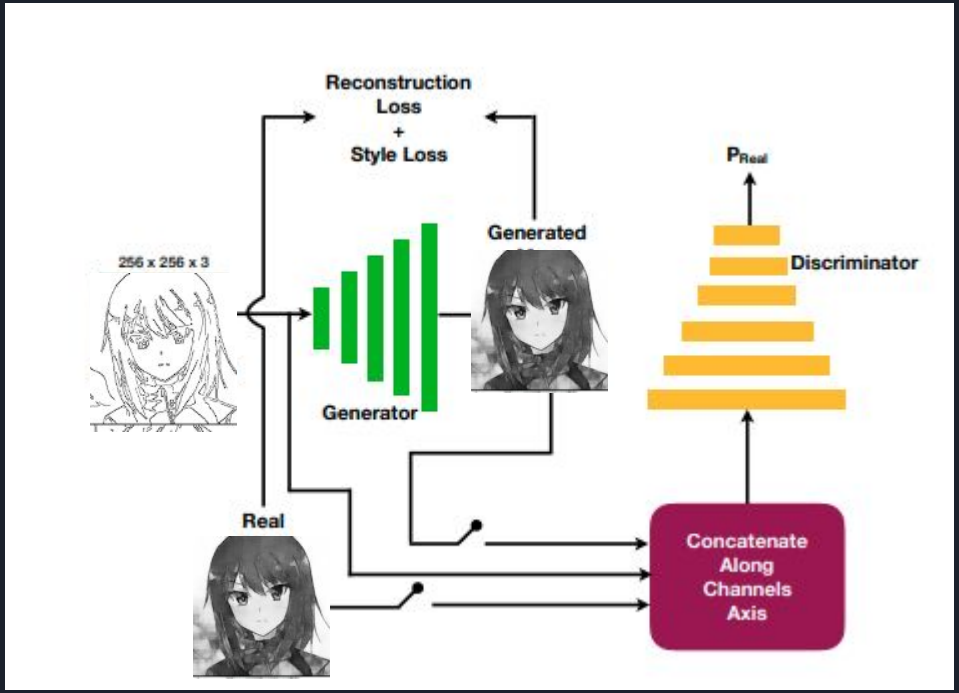
$$||M_{\text{fake}} - M_{\text{real}}||_2 \quad (3)$$

Style loss

$$G_{ij} = \sum_k F_{ik} F_{jk}$$

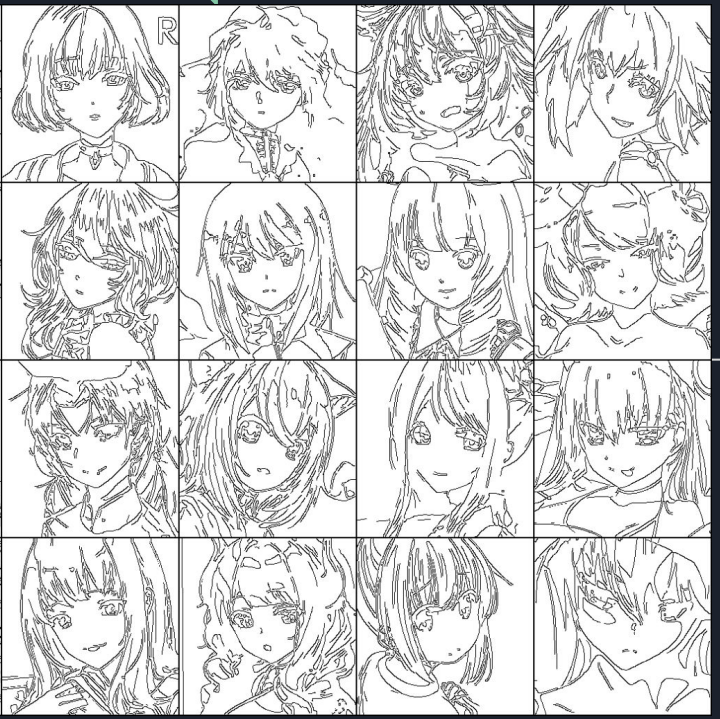
$$L_{style} = \frac{1}{4N^2M^2}(G_{ij} - A_{ij})^2 \quad (4)$$

Framework


Reconstruction
Loss
+
Style Loss
256 x 256 x 3
Generated
Generator
P_{Real}
Discriminator
Real
Concatenate
Along
Channels
Axis

Sketch to B&W Character Generation

Preprocessed images (Bilateral filter - > Canny edge detector)



Pix2Pix

Generated Images (Model is still training at the moment)

B&W to Color Character Generation

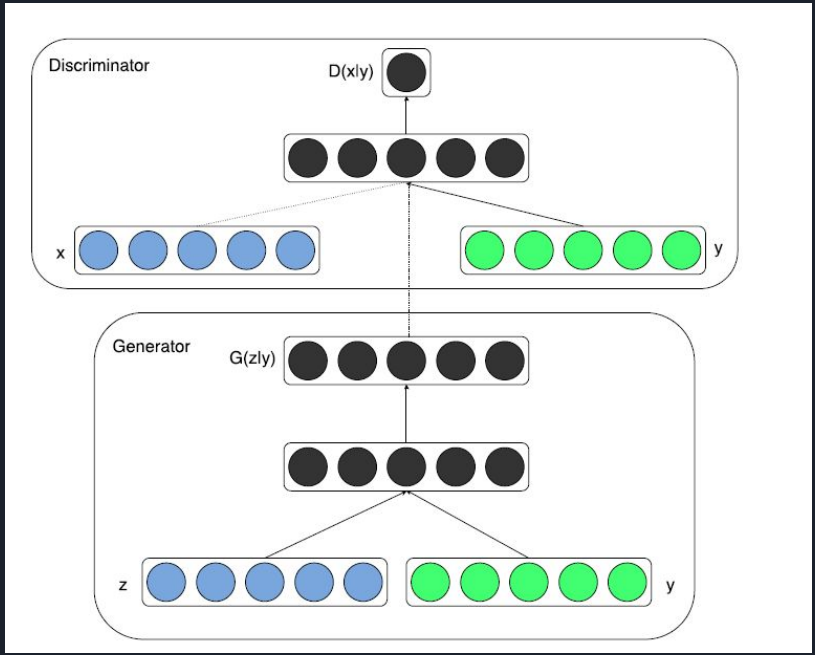
Preprocessed images (leave some color marks)

Pix2Pix

Generated Images

Conditional GAN

"Generative Adversarial Text to Image Synthesis" by **Reed et al.**



Algorithm 1 GAN-CLS training algorithm with step size α, using minibatch SGD for simplicity.

Input: minibatch images x, matching text t, mismatching $\hat{t}$, number of training batch steps S
for $n = 1$ **to** S **do**
  $h \leftarrow \varphi(t)$ {Encode matching text description}
  $\hat{h} \leftarrow \varphi(\hat{t})$ {Encode mis-matching text description}
  $z \sim \mathcal{N}(0,1)^Z$ {Draw sample of random noise}
  $\hat{x} \leftarrow G(z, h)$ {Forward through generator}
  $s_r \leftarrow D(x, h)$ {real image, right text}
  $s_w \leftarrow D(x, \hat{h})$ {real image, wrong text}
  $s_f \leftarrow D(\hat{x}, h)$ {fake image, right text}
  $\mathcal{L}_D \leftarrow \log(s_r) + (\log(1 - s_w) + \log(1 - s_f))/2$
  $D \leftarrow D - \alpha \partial \mathcal{L}_D / \partial D$ {Update discriminator}
  $\mathcal{L}_G \leftarrow \log(s_f)$
  $G \leftarrow G - \alpha \partial \mathcal{L}_G / \partial G$ {Update generator}
end for

Conditional Generation

Generated Images (still in the process…)

Challenge #2: Generation Model

- Discriminator becomes over confident, Generator struggles to fool the Discriminator.
- Mode Collapse.
- The training process is not stable.

How we are addressing it.

Over confidence:

- Label smoothing (0.9).
- Adding noise to inputs of Discriminator.

Generator struggles to fool the Discriminator:

- Double or triple steps for Generator per Discriminator update.
- Different learning rates (0.0004 and 0.0002 for Discriminator and Generator respectively).

Mode collapse:

- Gradient penalty as in “Wasserstein GAN” by Arjovsky et al.

To try:

- Different architecture since it might have a better capacity.

Evaluation

1. The **Fréchet inception distance (FID)** is a metric used to assess the quality of images created by a generative model, like a generative adversarial network (GAN). Unlike the earlier inception score (IS), which evaluates only the distribution of generated images, the FID compares the distribution of generated images with the distribution of real images that were used to train the generator.
2. Evaluate visually.

Limitations

- Constraints/Assumptions when user sketches:
  - Is a human (Not an animal or inanimate object)
  - Only face + neck, not full body
  - Includes desired key features (Nose, eyes, lips, etc)

Question Time