# PML&DL Project Deliverable 3

## 🤖Neural Face Editor (NFE)🎨


### Team

*Andrey Palaev, a.palaev@innopolis.university, B19-DS*

*Mikhail Rudakov, m.rudakov@innopolis.university, B19-DS*

*Anna Startseva, a.startseva@innopolis.university, B19-DS*


---

https://github.com/Palandr1234/NFE

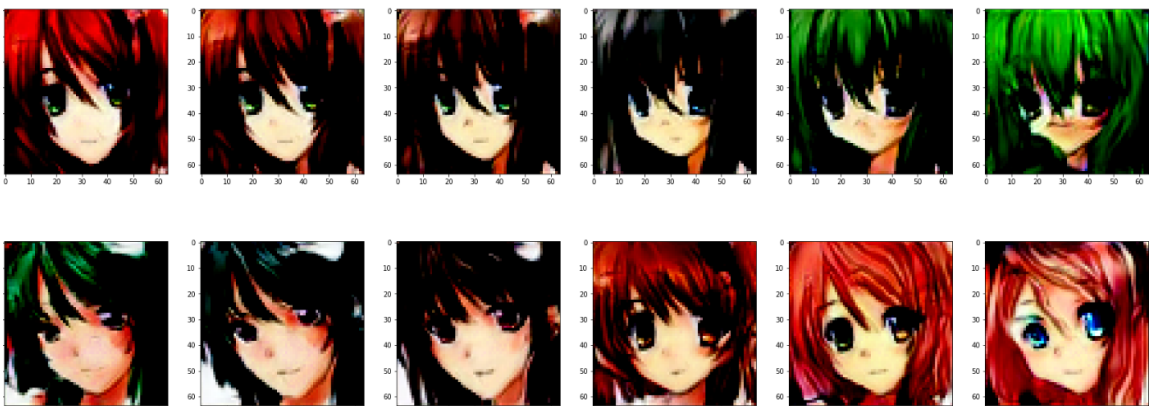
## Phase 3 Progress Overview

- Tried Image manipulation of several faces attributes
- Implemented first GAN inversion hybrid approach
- Created telegram bot with random StyleGAN faces generation

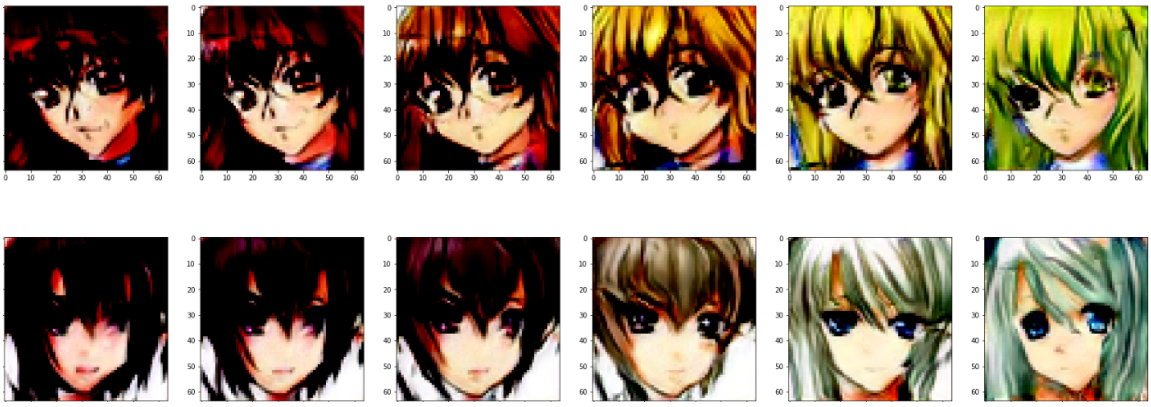
## Image manipulation

After getting separating hyperplanes, we implemented the way of attribute manipulation from [1]. Specifically, given the separating hyperplane n and the latent vector z, we generate several images using $z + an$, where `a=np.linspace(-3, 3, 6)`. Examples of image manipulation:

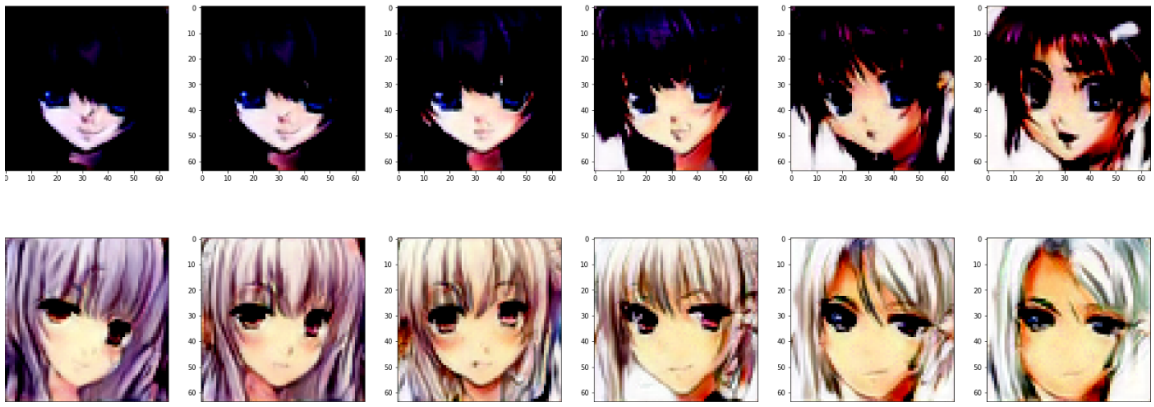
1. Making hair more (or less) green while keeping the hair length



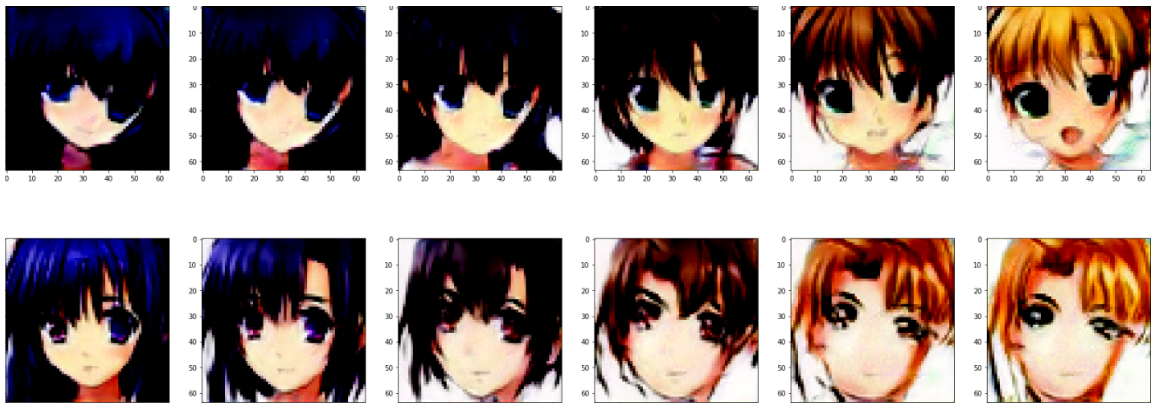
2. Making hair more (or less) red without controlling the hair length



3. Changing the hair length preserving the hair colour



4. Changing the hair length without controlling the hair colour



As we can see, both usual and conditional manipulation work well. However, it is not successful for eyes since the labelled annotator performed bad on eye colour. As a result, separating hyperplanes for eye colour are very far from the true ones.

## GAN Inversion

Hybrid approach for GAN inversion is implemented, as discussed in the previous report.

For training the Encoder, we used the following loss function:

$$E^* = \arg\min_E(||G(E(imgs)) - imgs||^2 + 0.01 * D(G(E(imgs))) + 0.002||VGG(G(E(imgs))) - VGG(imgs)||^2)$$

The Encoder was trained on the same dataset as GAN

Then, for the specific input image we initialize z as $z_{init} = E(img)$

Finally, the latent vector $z_{init}$ is optimized as follows:

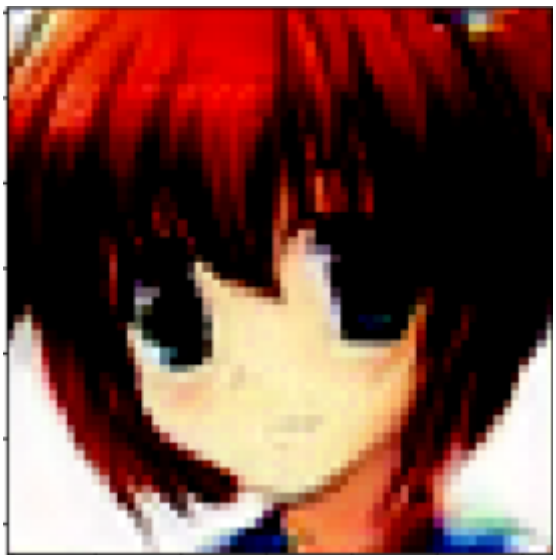
$$z^* = \arg\min_z ||\sigma(out, 3, 11.0) - \sigma(img, 3, 11.0)||^2 + 20||z||^2$$

$\sigma$ is the Gaussian kernel

Here we list some preliminary results to access quality of GAN inversion.





As we see, general faces features are reconstructed by GAN Inversion. However, some tiny parts (blush, cheekbones) are not exact. We plan to tune GAN inversion a little bit to achieve better results.

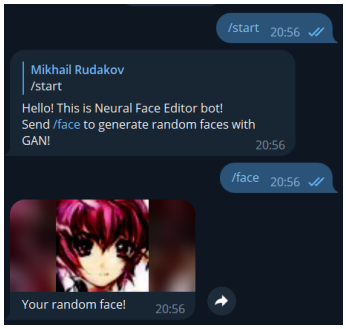
## Telegram Bot basic functionality

We also created telegram bot for our project, which is currently hosted on local computer and thus is not available 24/7. Alias: https://t.me/neural_face_editor_bot

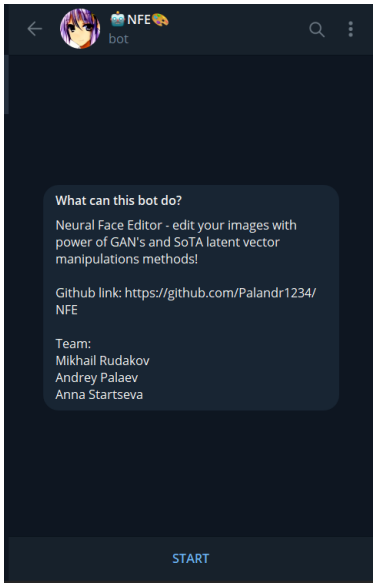
This bot is intended to download anime faces photos from the users (or generate random ones) and allow to change some image attributes in the convenient telegram interface.

At this step, telegram bot uses trained StyleGAN to generate random anime faces. Intro message is available by `/help`. Send `/face` to try generating random images! (when bot would be online..)

Here are some examples of bot interaction:




Help message




Bot welcome page


Faces generation

Next, we plan to extend bot functionality with uploading custom faces & image manipulation. We also aim to upload the bot to the could, so it could be publicly accessible. Stay tuned!

## Team Work Distribution

Andrey - Finetuned GAN architecture, GAN attribute manipulation

Anna - Added random anime face generation to telegram bot, github readme, GAN inversion

Mikhail - Wrote basic Telegram Bot functionality, helped with bot faces generation

## Sprint 4 Tasks:

- Implement full project pipeline: GAN inversion - GAN - attribute manipulation
- Add GAN inversion and attribute manipulation to Telegram Bot
- Optional: finetune GAN