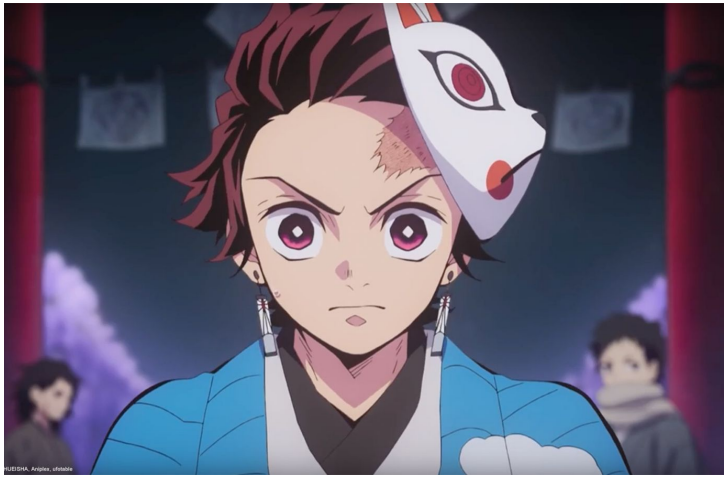
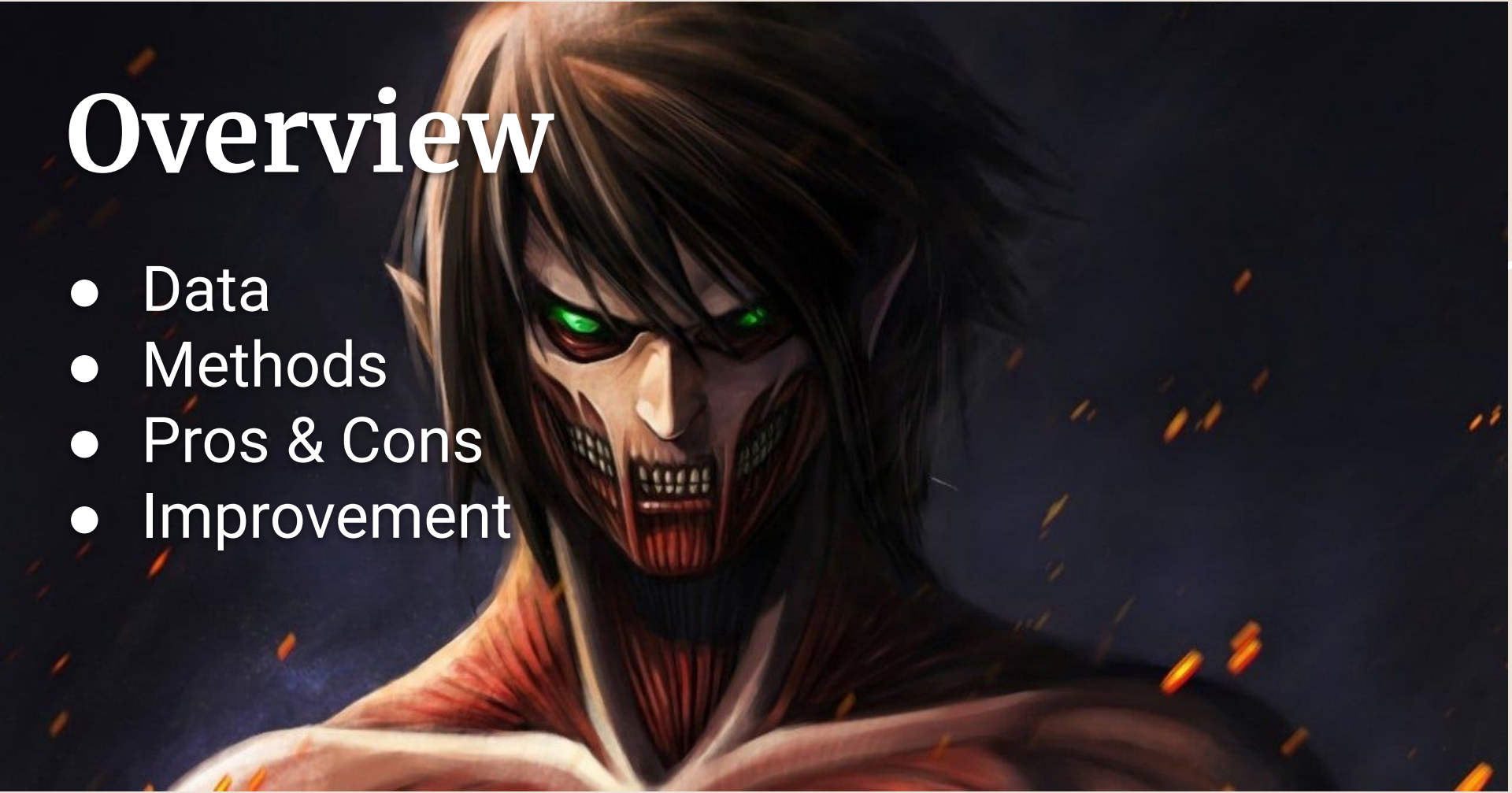
Fake Anime Faces with Pytorch



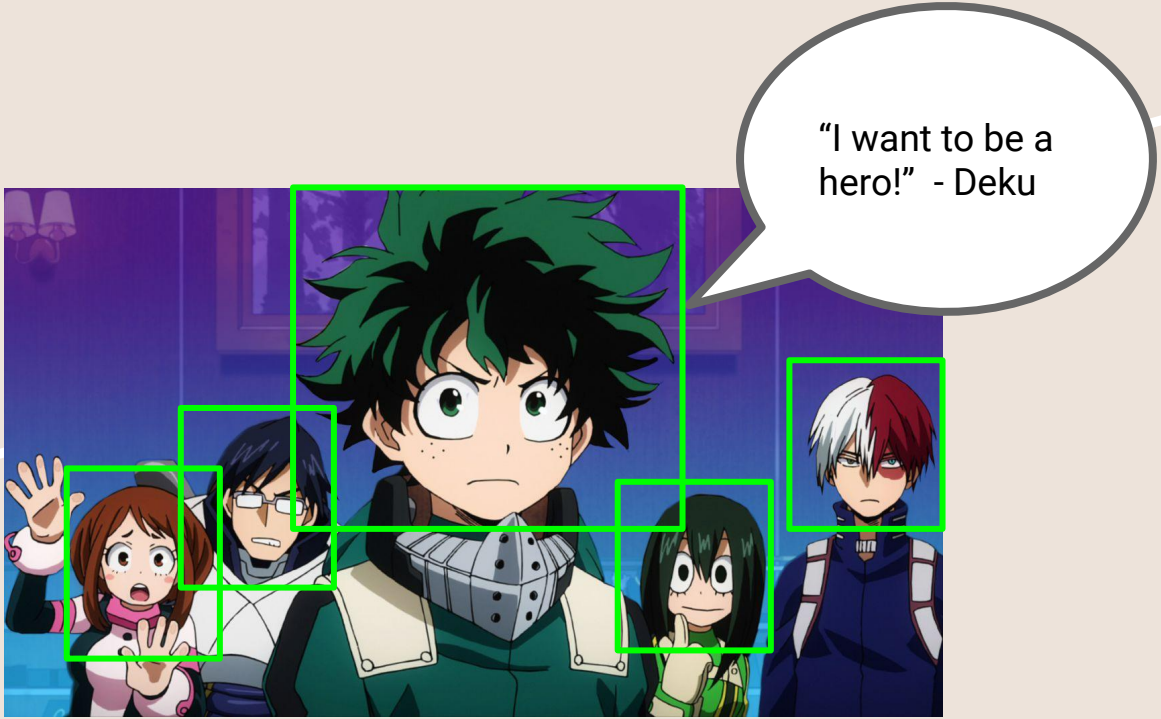
Danh Nguyen
Jared Mlekush



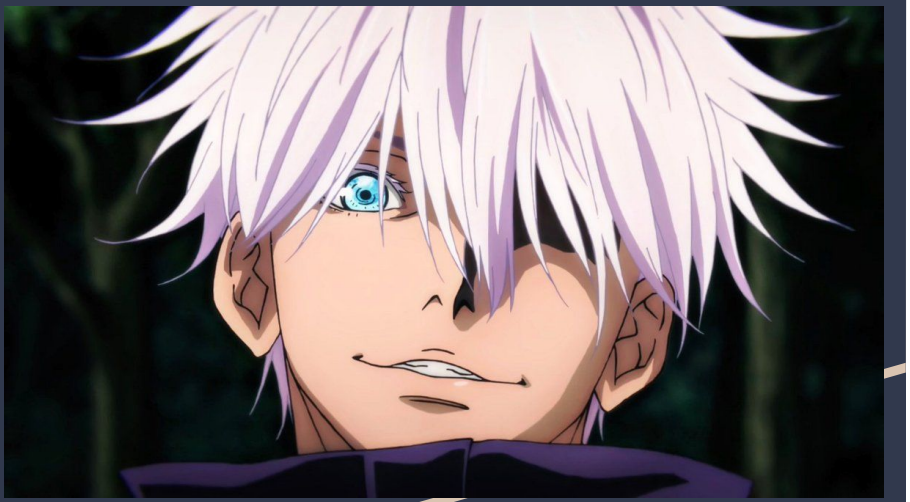
Overview

- Data
- Methods
- Pros & Cons
- Improvement

Recap
Why fake anime faces?!



Data



- Our clean data is just under 400mb, consisting of over 63,000 faces
- Data can be found here on Kaggle: Lots and lots of anime faces
- Images' size ~ 64 x 64

Methods

Summarizing the discriminator

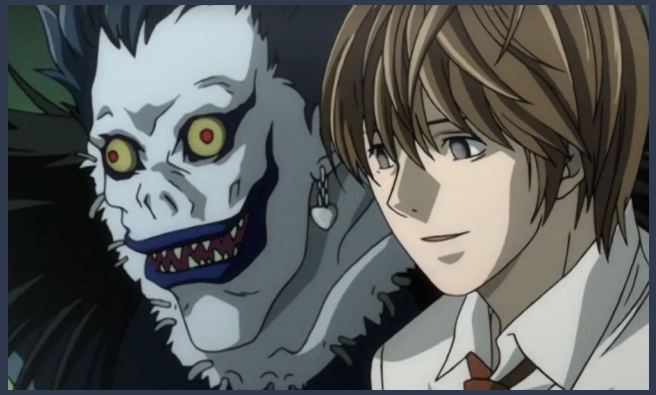
```
discriminator = Discriminator(maps = 64).to(device)
summary(discriminator, (3, 128, 128))
```

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1          [-1, 64, 64, 64]           3,136
         LeakyReLU-2          [-1, 64, 64, 64]               0
            Conv2d-3         [-1, 128, 32, 32]         131,072
       BatchNorm2d-4         [-1, 128, 32, 32]             256
         LeakyReLU-5         [-1, 128, 32, 32]               0
            Conv2d-6         [-1, 256, 16, 16]         524,288
       BatchNorm2d-7         [-1, 256, 16, 16]             512
         LeakyReLU-8         [-1, 256, 16, 16]               0
            Conv2d-9           [-1, 512, 8, 8]       2,097,152
      BatchNorm2d-10           [-1, 512, 8, 8]           1,024
        LeakyReLU-11           [-1, 512, 8, 8]               0
           Conv2d-12          [-1, 1024, 4, 4]       8,388,608
      BatchNorm2d-13          [-1, 1024, 4, 4]           2,048
        LeakyReLU-14          [-1, 1024, 4, 4]               0
           Conv2d-15             [-1, 1, 1, 1]          16,385
          Flatten-16                   [-1, 1]               0
================================================================
Total params: 11,164,481
Trainable params: 11,164,481
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.19
Forward/backward pass size (MB): 9.63
Params size (MB): 42.59
Estimated Total Size (MB): 52.40
----------------------------------------------------------------
```

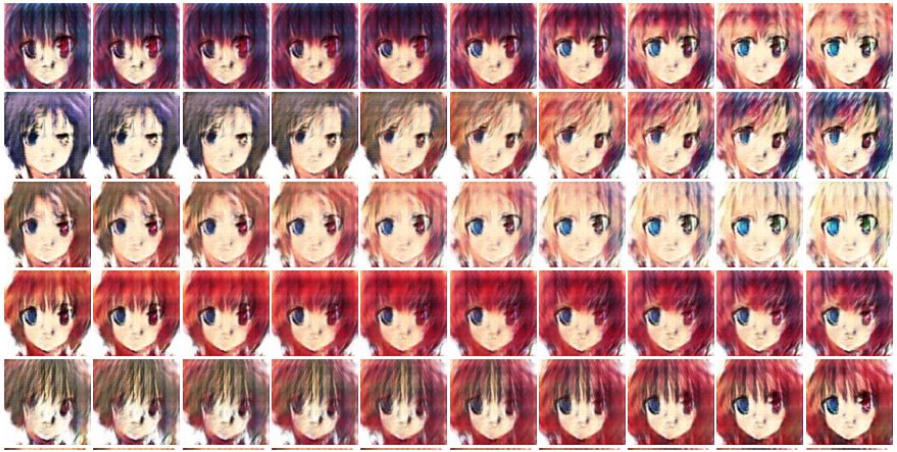
- We found an implementation very similar to how we had envisioned our project
- Recreated the project and learned
- Tried additional approaches to **improve model**
  - Lost Functions
  - Learning Rate
    - One uniform rate
    - Different rates for D(x) vs. G(x)
  - Different transformations

Source: Inspiration

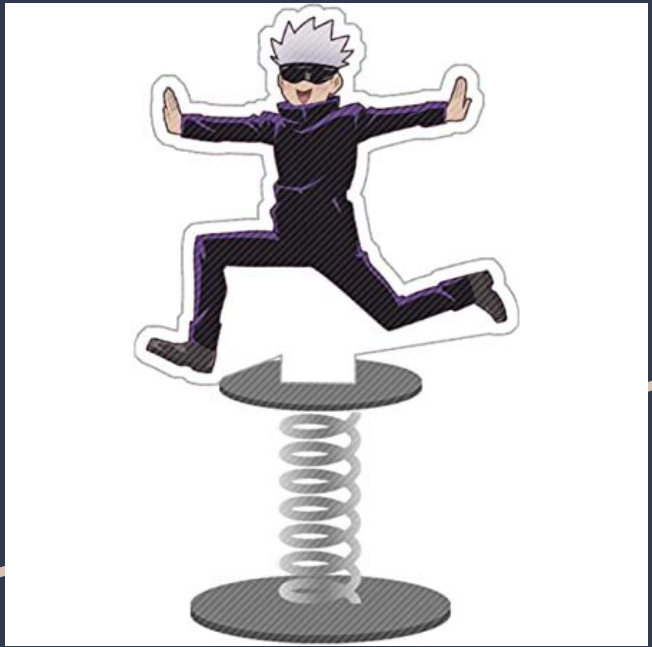
Pros & Cons



- Pros
  - Good out of the box
  - Generated distinctive anime faces
- Cons
  - Mode collapse
  - Unproportional facial features
  - Underestimated how long it would take to run



Improving



Try

- Adding additional layers
- Try different loss functions
- Trying additional combinations of image augmentations
- More images/higher quality images

Arigatou!