Capstone Design Project

Midterm Presentation

[Team: Jjangdol]

2016314474 윤성빈 2016310703 남득윤
2016314668 위성은 2018310656 이다솔

index

01 Objective

1. Efficiency
Reduce cost and simplify the process of making font in Korean by applying font generation model.

2. Accesibility
Develop web-service and provide functions such as creating own font design and providing it in .ttf file.

3. Maintainability
Complement and improve the output of font design to increase user satisfaction by applying Ensemble.

4. Entertainment
Increase user interest by providing a function that can combine inputs of several users on font production.

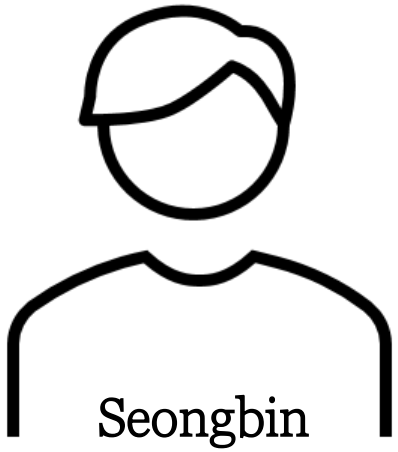
02 Schedule milestone

: Project plan on proposal

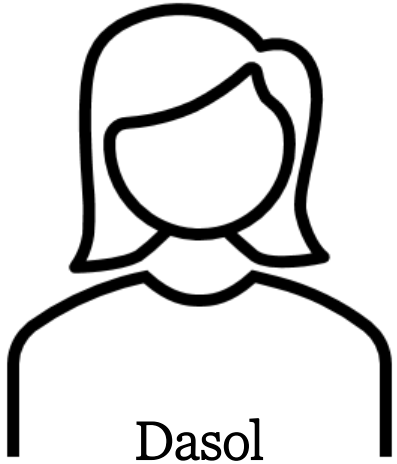
: Actual Progress

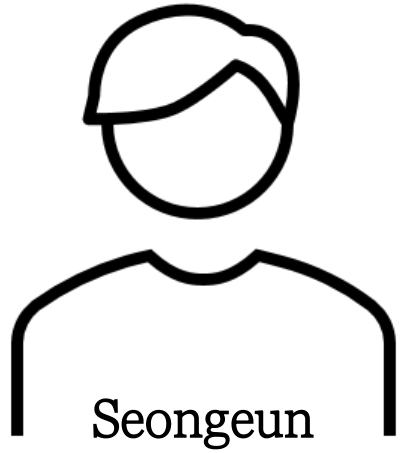
: Research/ Study for Progress

03 Role of each member


Seongbin


Deukyun


Dasol


Seongeun

<Font Generation>

Implement on font generation by applying DM-font

Automate file format transformation process

<FrontEnd and Backend>

Develop Web application using React and Nodejs

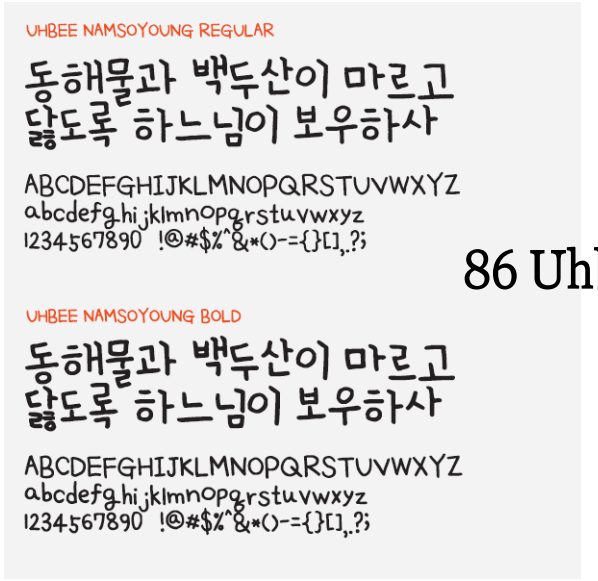
Connect Servers and Construct Backend

<Integration>

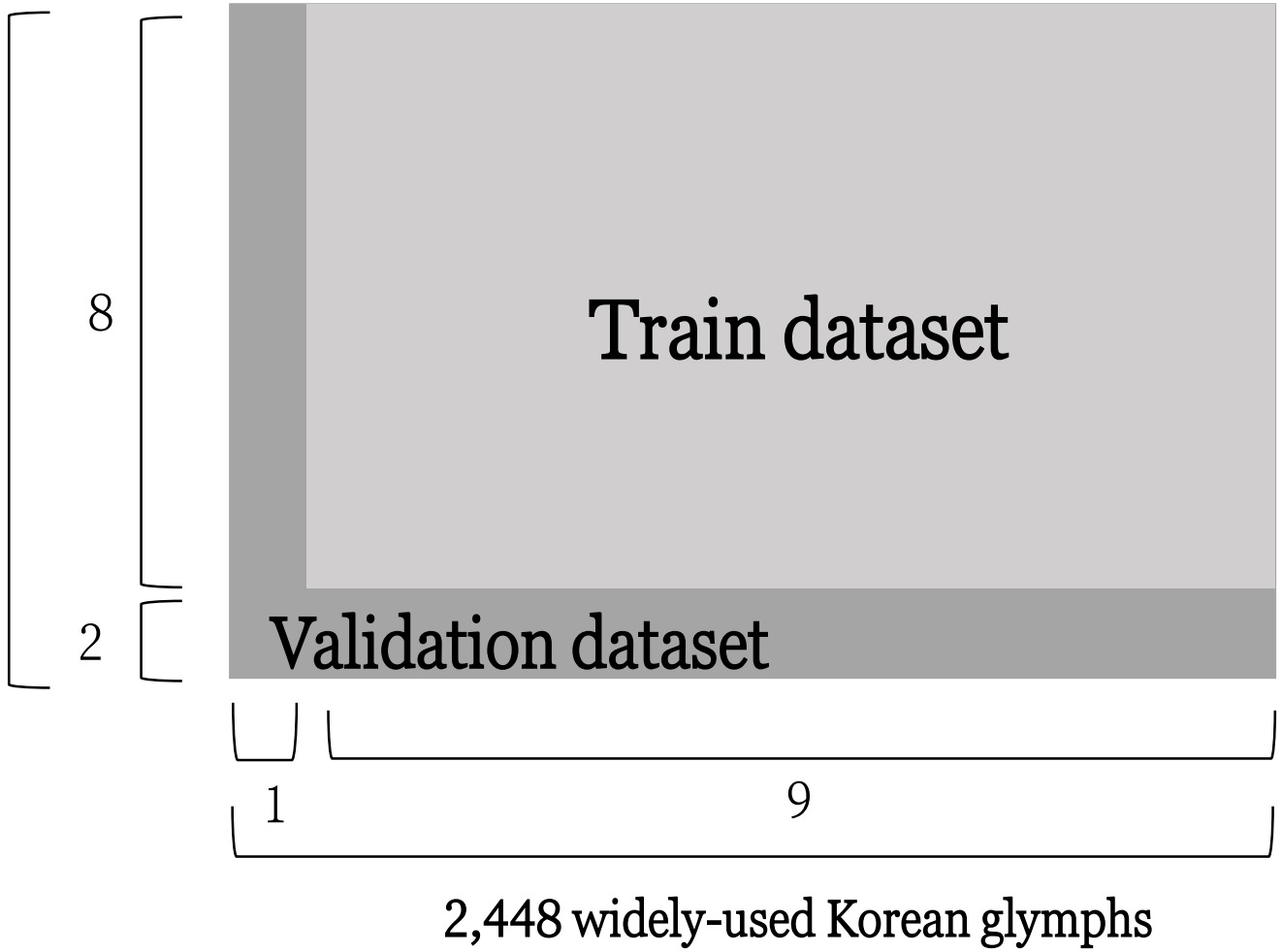
Integrate the codes of each functions and build an executable environment

04 ML Model: DM-Font

[Data Set]

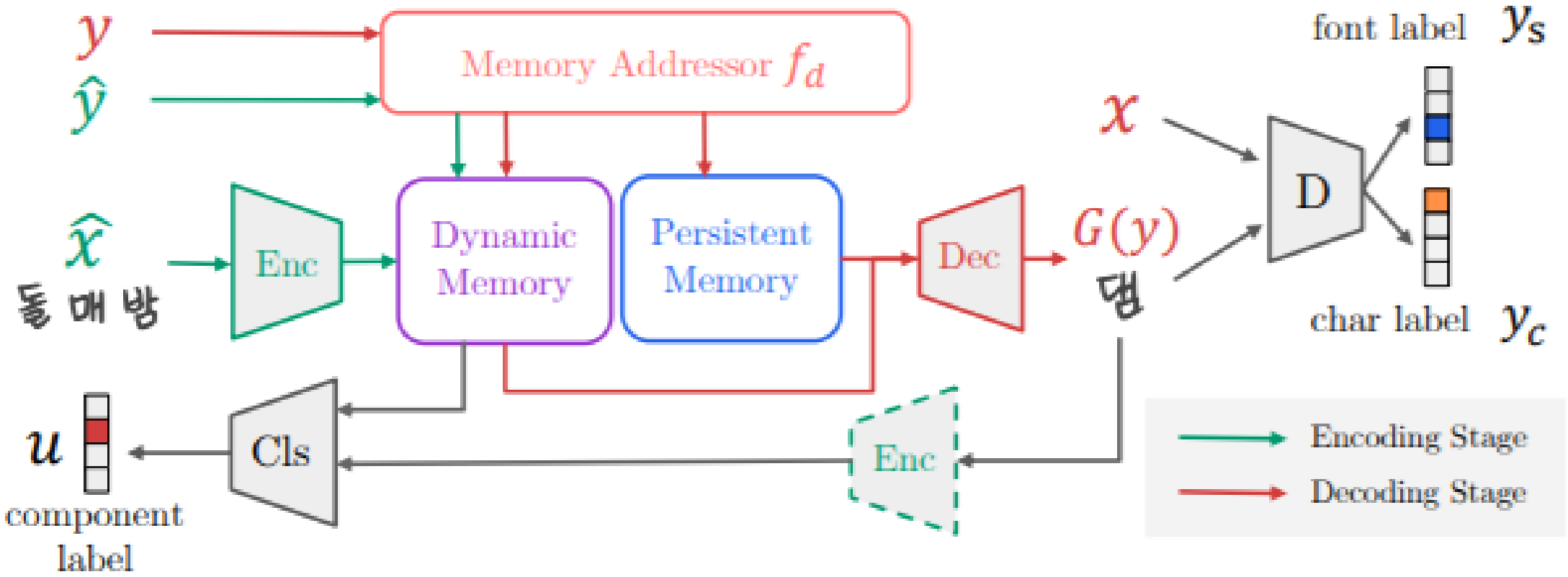


86 Uhbee Font

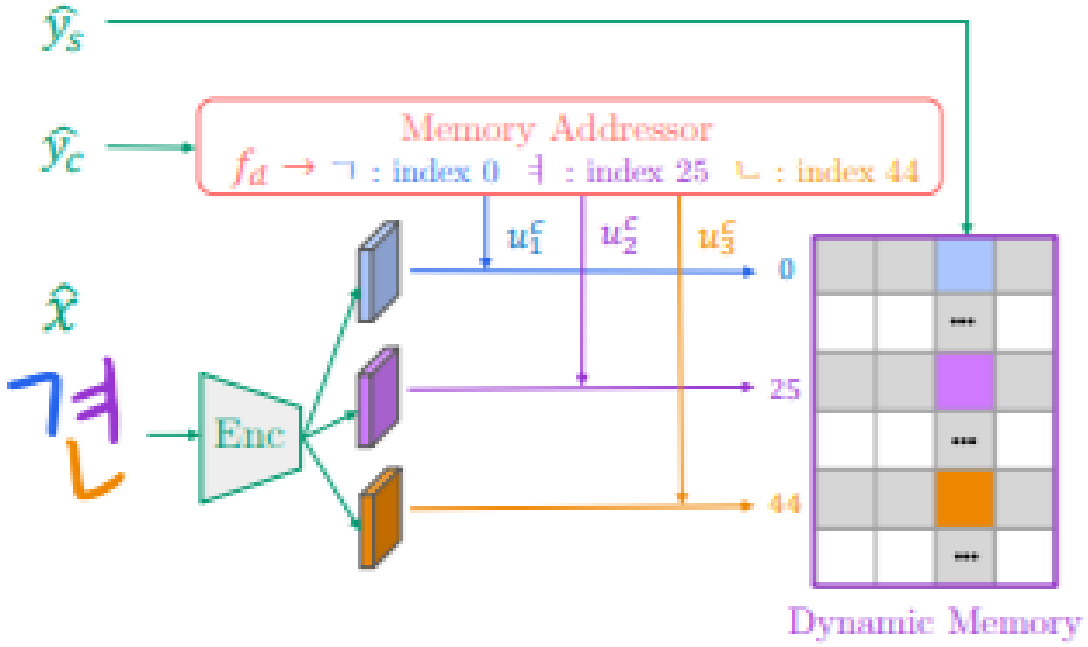


04 ML Model: DM-Font

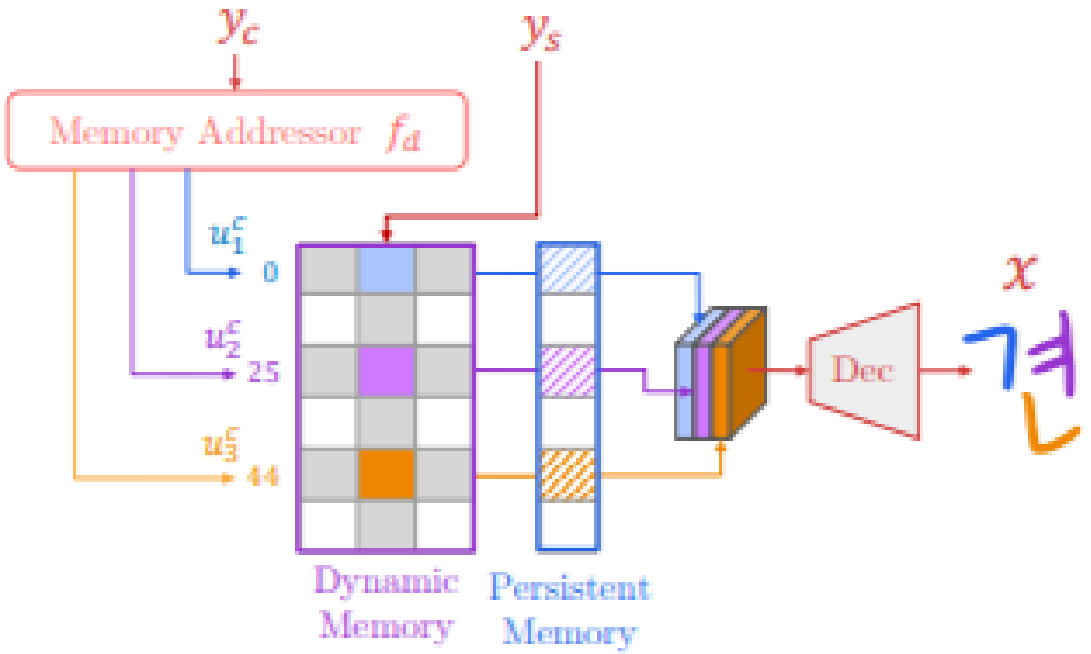
[Architecture overview]



04 ML Model: DM-Font



(b) Encoding phase detail.



(c) Decoding phase detail.

$$G(y_c, y_s) = Dec\left(\left[DM(u_i^c, y_s),\ PM(u_i^c) \mid u_i^c \in f_d(y_c)\right]\right)$$

$where, y_s = font\ label, y_c = char\ label \quad f_d(\text{“한”}) = \{\text{“ㅎ”}, \text{“ㅏ”}, \text{“ㄴ”}\}$

04 ML Model: DM-Font

[Objective Function]

$$\min_{G,C} \max_{D} \mathcal{L}_{adv(font)} + \mathcal{L}_{adv(char)} + \lambda_{l1}\mathcal{L}_{l1} + \lambda_{feat}\mathcal{L}_{feat} + \lambda_{cls}\mathcal{L}_{cls},$$

$$\lambda_{l1} = 0.1, \lambda_{feat} = 1.0, \lambda_{cls} = 0.1$$

- Adversarial loss on font and char
- Pixel level L1 loss
- Feature matching loss (~adv loss)
- Component-Classification loss

04 ML Model: DM-Font

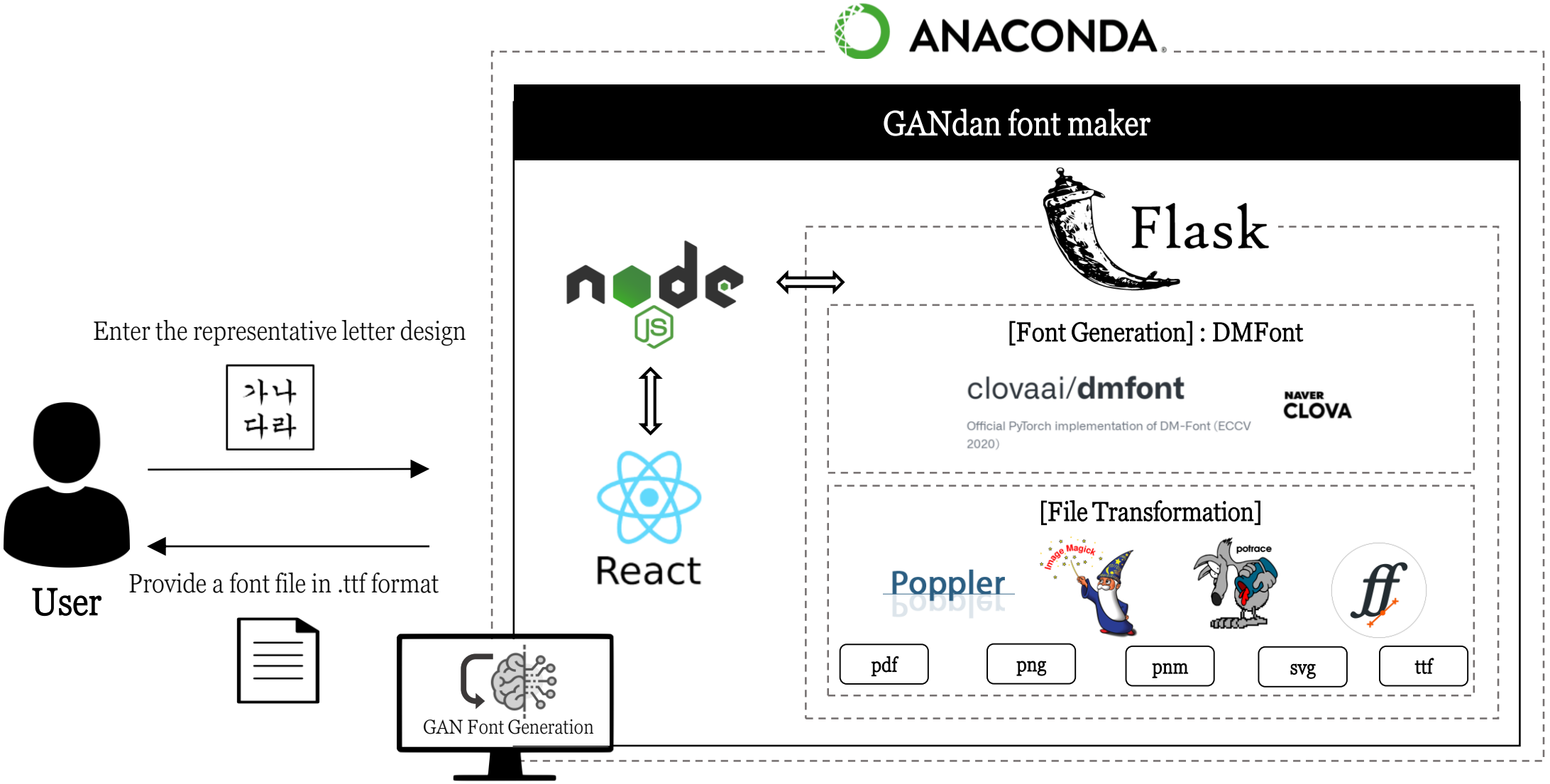
[Evaluation Metric]

- Pixel-level
- Perceptual-level
  - Using resnet-50
  - Content-aware, Style-aware
- Human-level

[Default Config related to train]

```
batch_size: 12
max_iter: 200000
seed: 2
g_lr: 2e-4
d_lr: 8e-4
n_workers: 2
adam_betas: [0.0, 0.9]
init: kaiming
```

05 Overall Structure



06 Implementation(WEB)

[FrontEnd Part]

React

React: Javascript Library

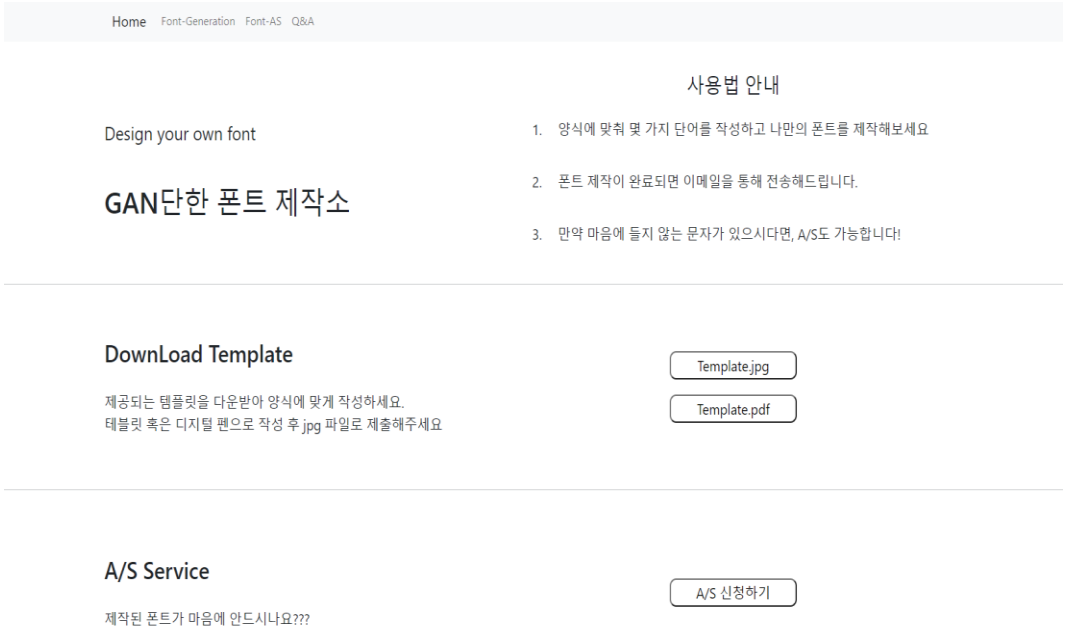
1. Create Webpages

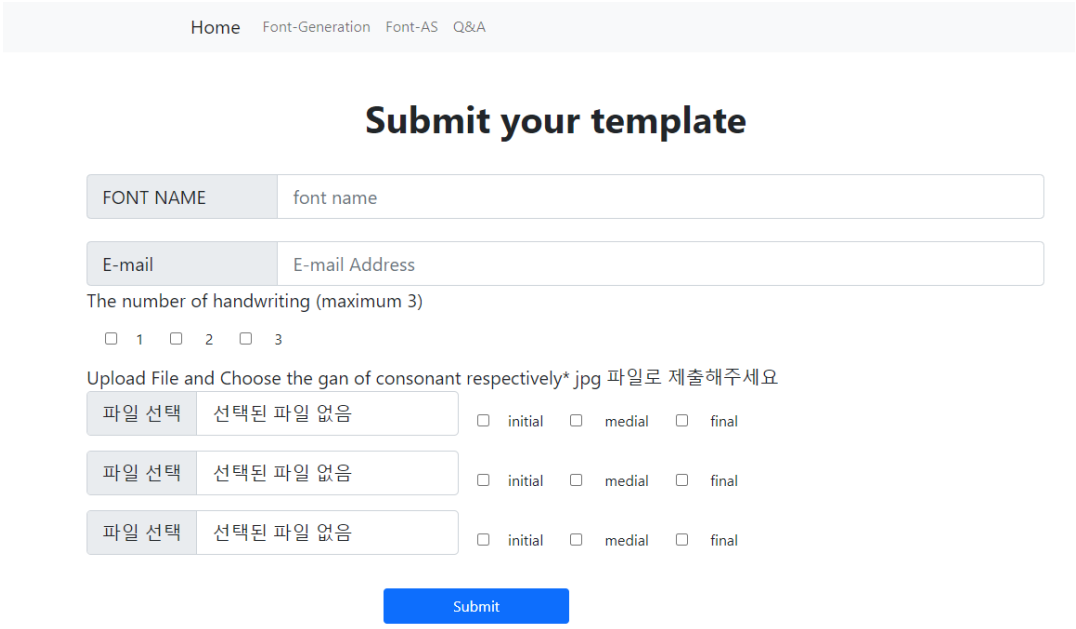
- Users send information and receive their result
- using Bootstrap(Navbar, Button ...)

2. Transferring to NodeJs

- Make the information in JSON format

-> Axios(Post Method to NodeJs)





06 Implementation(Server)

[BackEnd Part]

Nodejs

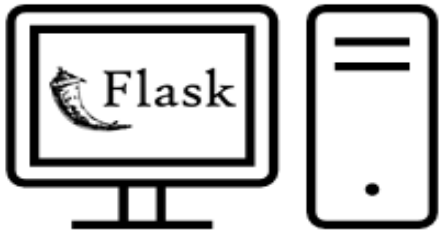
Nodejs: server based on Javascript

1. Receive template and user data

Template includes the handwritten words of user

2. Network of Nodejs and Flask

Axios(Nodejs) and RestAPI(Flask)



Flask

Flask: server based on Python

1. Execute AI model

[Font generation process] pdf -> png -> svg -> ttf -> Model -> png -> svg -> ttf

pdf -> png (opensource: pdf2images, poppeler)
png-> svg (opensource: ImageMagick(png->pnm), potrace (pnm->svg))
svg-> ttf (opensource: FontForge)
GAN Font Generation (opensource: DM-font)

2. Save font generation result and post to nodejs

07 Challenges

1. Font Generation Part

- DM font
- hard to understand the code -> ask to the author of the paper

2. File Transformation Part

- png2svg : lack of open sources and hard to expect quality of output
  -> conversion via pnm, not directly.

3. Web, Server Part

- Image transmission(React -> NodeJs)
  -> using multuer in NodeJs

08 Limitation

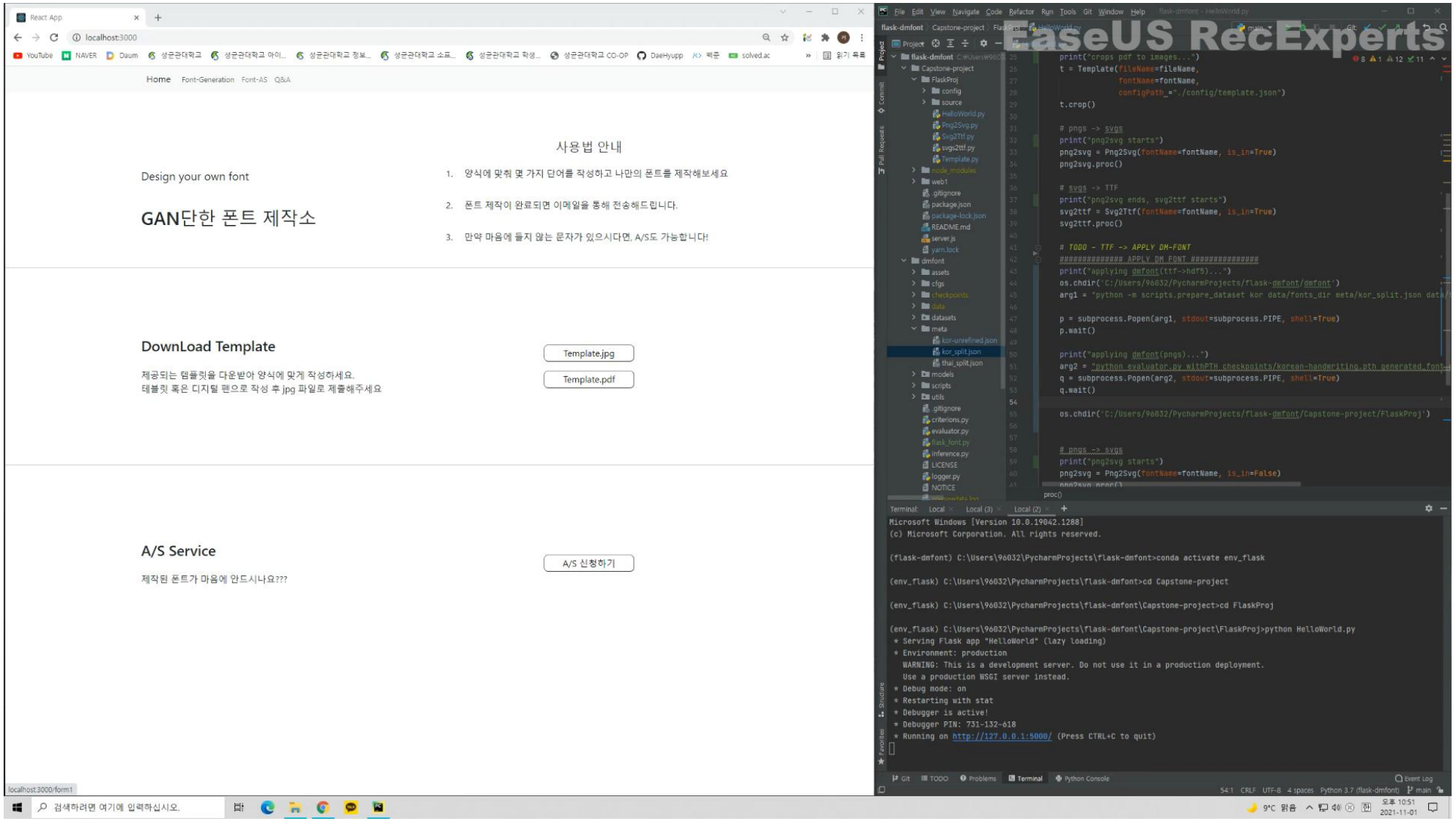
1. So far, we use the pre-trained model
Does not using Dynamic Memory dynamically
-> We will try the validation process to get better performance results

2. Single API call between Nodejs and Flask
Difficult to manage errors during ML work in Flask server
-> Design that client(Nodejs) can receive the internal progress of server(flask) in real time.

3. Lack of validation on user input
Example: user upload "not pdf file"?
-> Improve web-service by add validation steps on user input

4. In the case of multiple user request concurrently
-> the number of concurrent users will be limited to one by session management policy.

09 Demo video



10 Future Plans

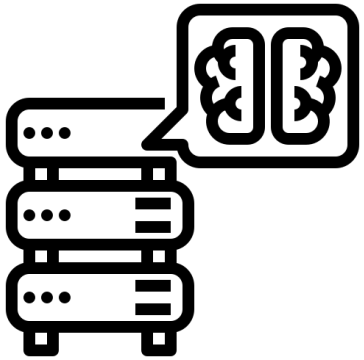
In Web application part,

1. Connect NodeJs with database

- Store user information and font generation result
- User authentication

2. User-friendly

- Change in page shape according to the user's window size
- provide a link to download the requested font



In Machine learning part,

1. Combining user's fonts

- Create font result that includes all styles of multiple users

2. User feedback

- Increase user satisfaction using Adaboost

Thank you