

Capstone Design Project

Midterm Presentation

[Team: Jjangdol]

2016314474 윤성빈 2016310703 남득윤
2016314668 위성은 2018310656 이다솔

index

01. Objective
02. Schedule mliestone
03. Role of each member
04. DM-font
05. Overall Structure
06. Implementation
07. Challenges
08. Limitation
09. Demo
10. Future Plans

01 Objective

1. Efficiency

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

2. Accessibility

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

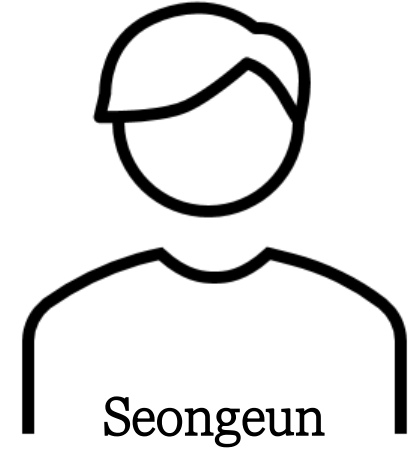
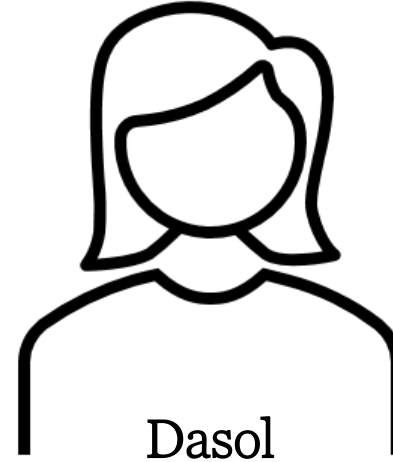
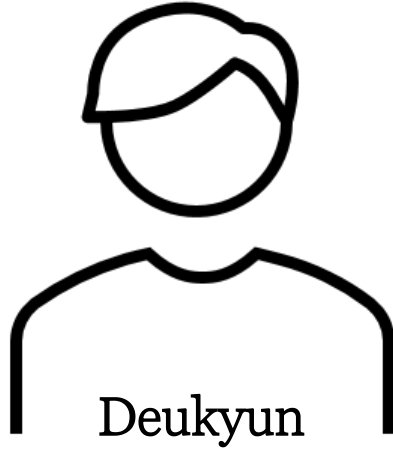
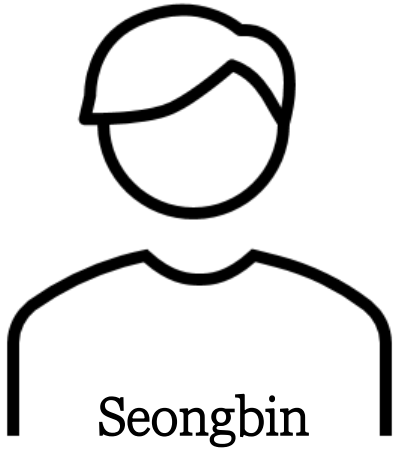
 : Actual Progress

 : Research/ Study for Progress



03

Role of each member



Implement on font generation by applying DM-font
Automate file format transformation process

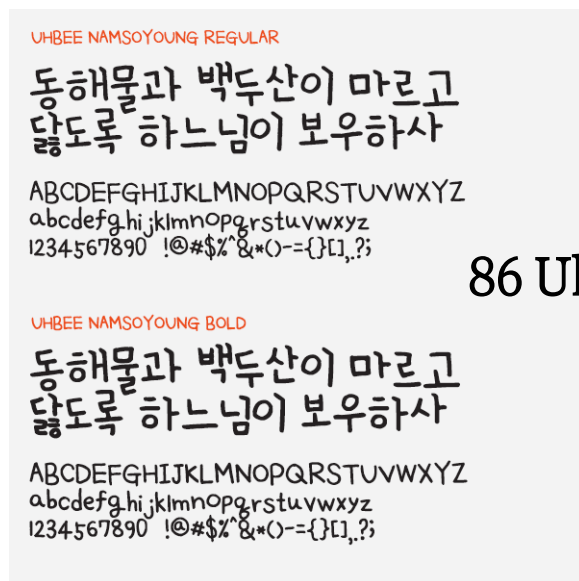
<FrontEnd and Backend>

Develop Web application using React and Nodejs
Connect Servers and Construct Backend

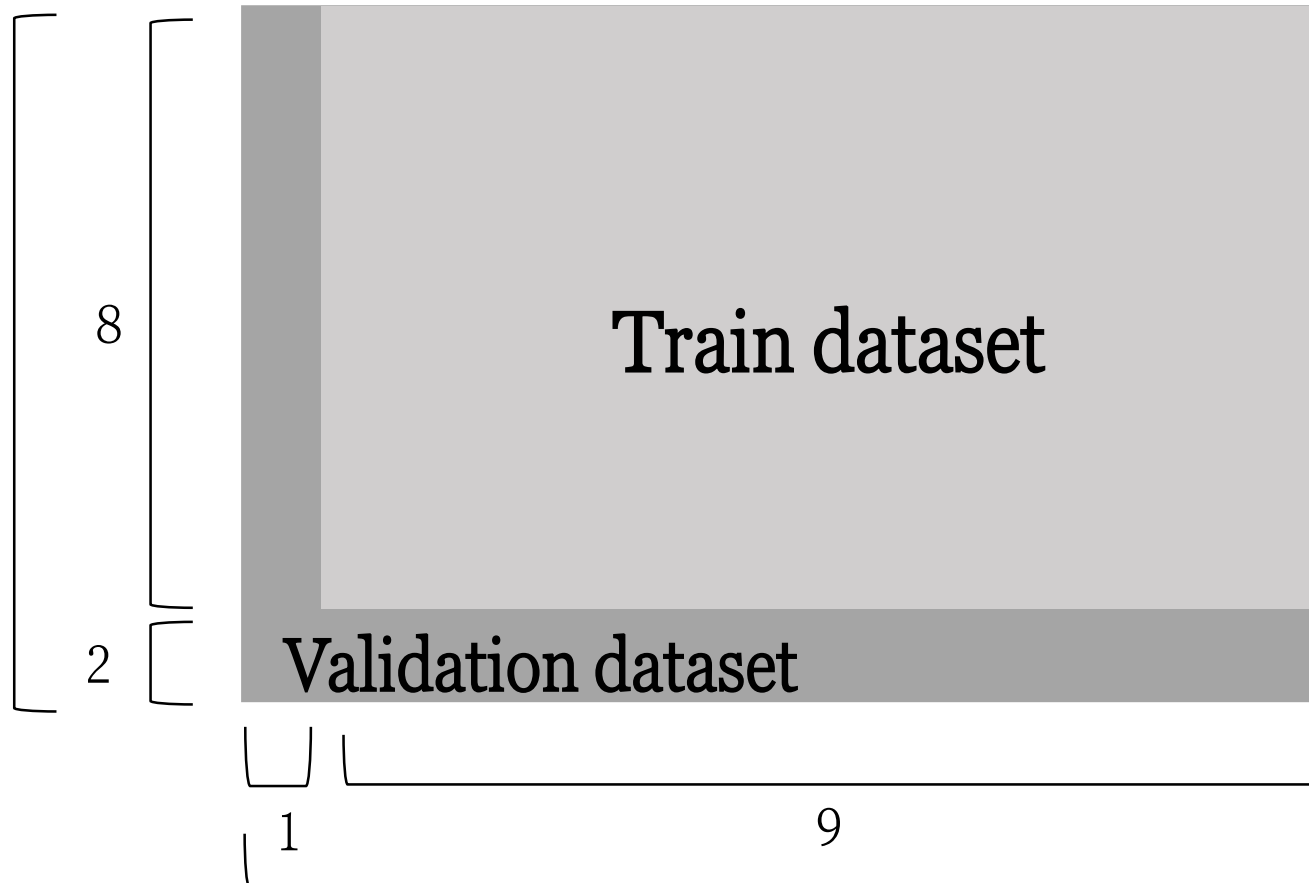
<Integration>

Integrate the codes of each functions and build an executable environment

[Data Set]

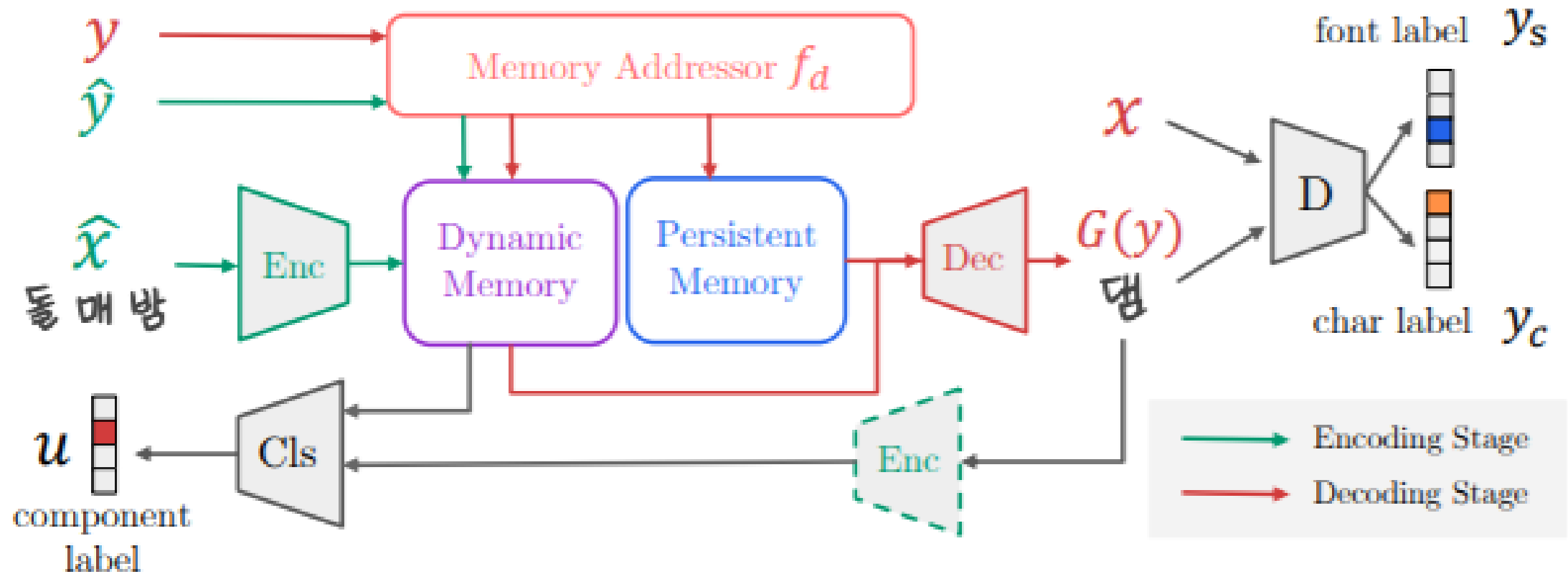


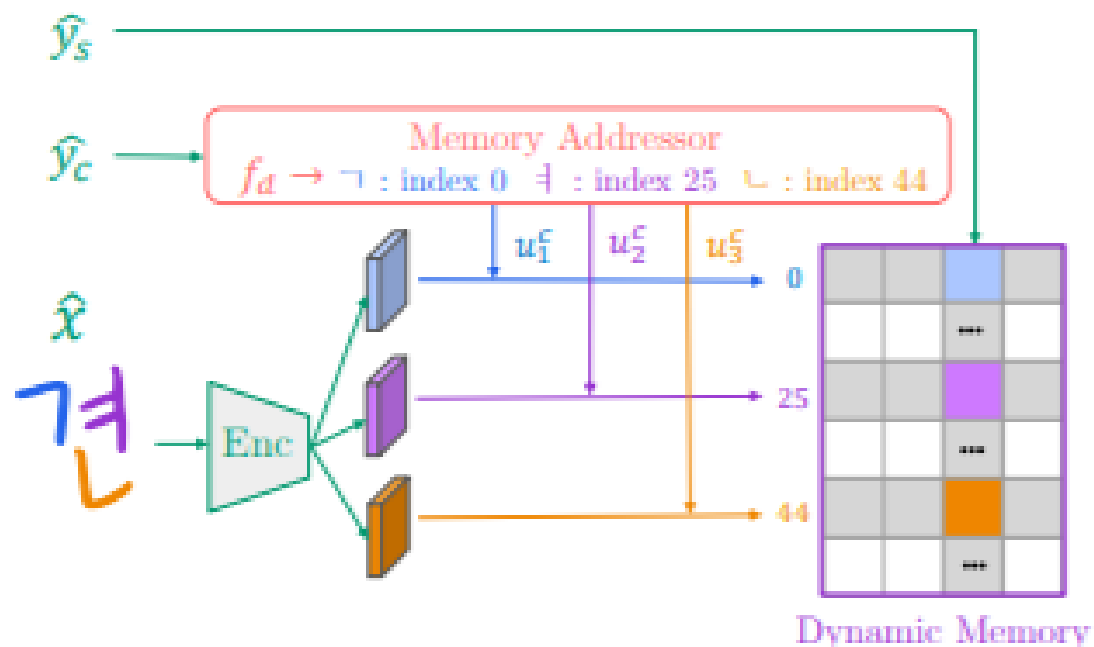
86 UhbEE Font



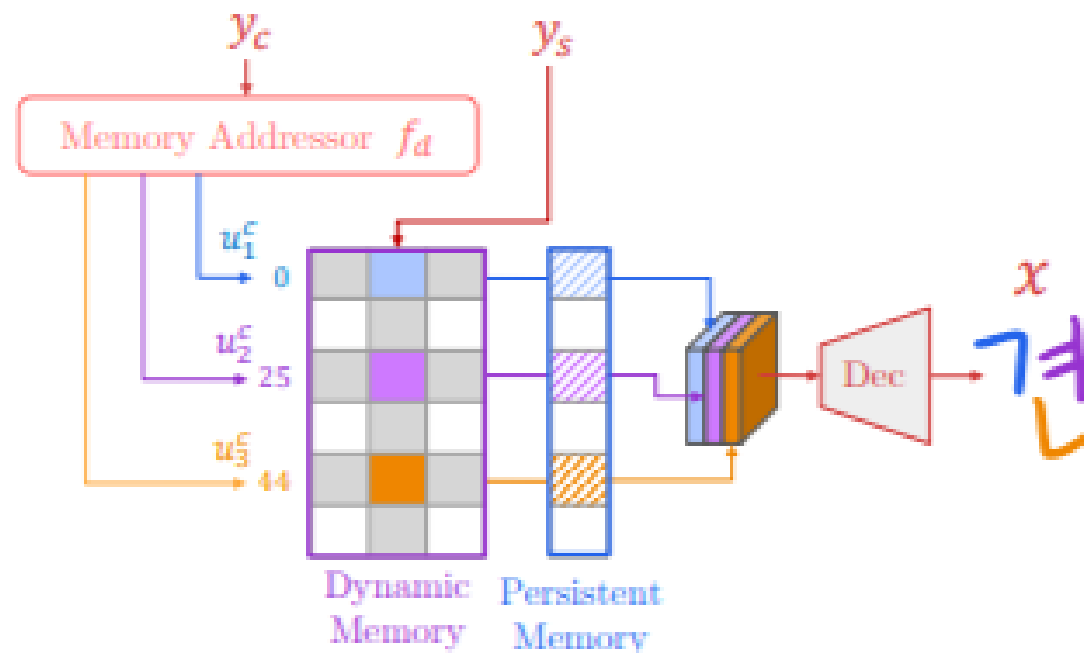
2,448 widely-used Korean glyphs

[Architecture overview]





(b) Encoding phase detail.



(c) Decoding phase detail.

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

where, $y_s = \text{font label}$, $y_c = \text{char label}$ $f_d(\text{"ㄣ"}) = \{\text{"ㄣ"}, \text{"ㄚ"}, \text{"ㄥ"}\}$

[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

[Evaluation Metric]

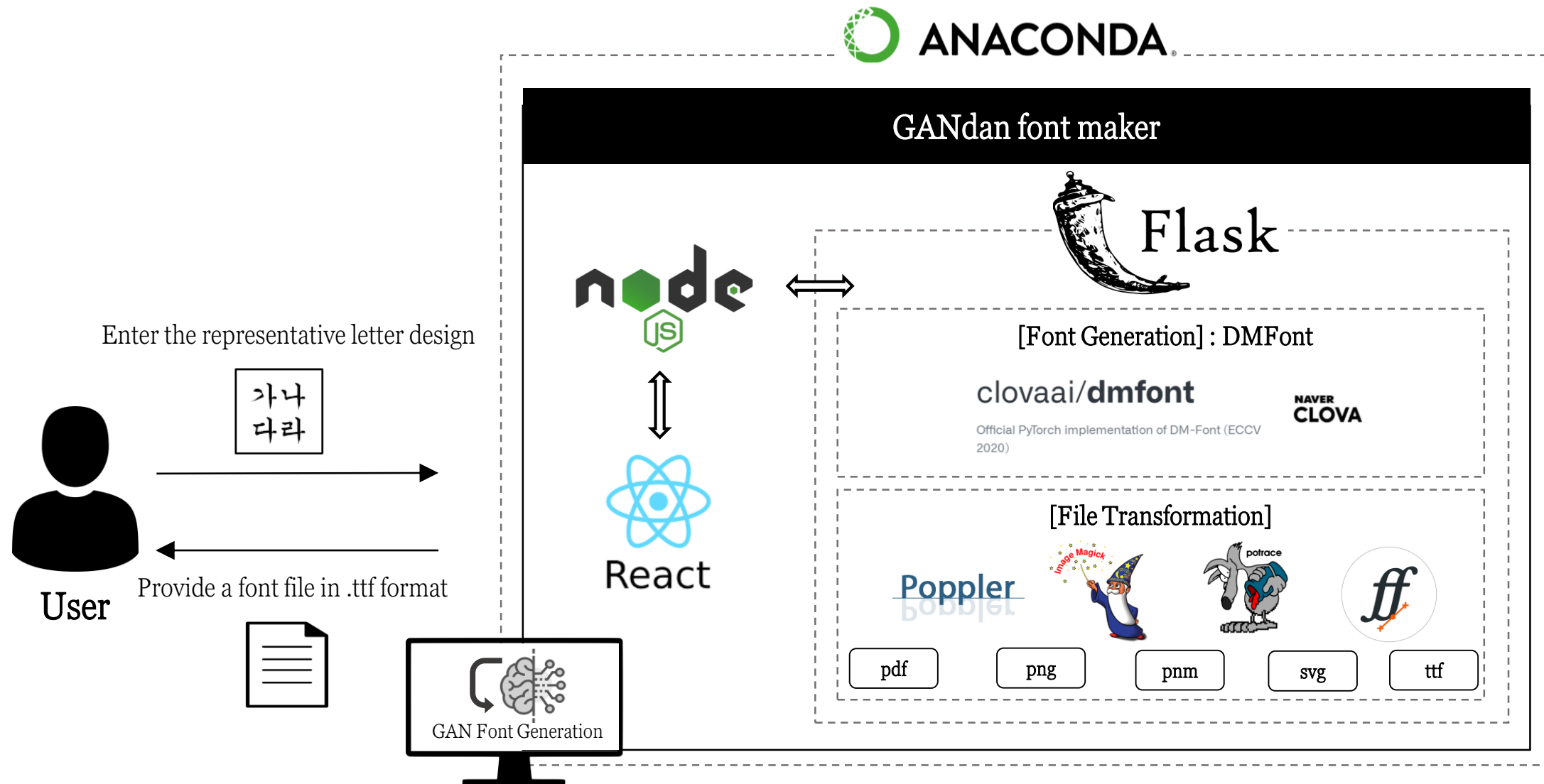
- Pixel-level
- Perceptual-level
 - Using resnet-50
 - Content-aware, Style-aware
- Human-level

[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



[FrontEnd Part]



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)

[Home](#)
[Font-Generation](#)
[Font-AS](#)
[Q&A](#)

사용법 안내

Design your own font

GAN단한 폰트 제작소

Download Template

제공되는 템플릿을 다운받아 양식에 맞게 작성하세요.
테블릿 혹은 디지털 펜으로 작성 후 .jpg 파일로 제출해주세요

Template.jpg

Template.pdf

A/S Service

제작된 폰트가 마음에 안드시나요???

A/S 신청하기

[Home](#)
[Font-Generation](#)
[Font-AS](#)
[Q&A](#)

Submit your template

FONT NAMEfont name

E-mailE-mail Address

The number of handwriting (maximum 3)

☐ 1
 ☐ 2
 ☐ 3

Upload File and Choose the gan of consonant respectively* .jpg 파일로 제출해주세요

파일 선택선택된 파일 없음

☐ initial
 ☐ medial
 ☐ final

파일 선택선택된 파일 없음

☐ initial
 ☐ medial
 ☐ final

파일 선택선택된 파일 없음

☐ initial
 ☐ medial
 ☐ final

Submit

[BackEnd Part]



Nodejs

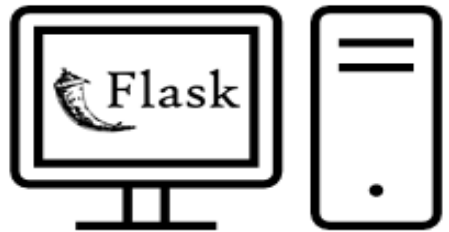
Nodejs: server based on Javascript

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, poppler)

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

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

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.

The image shows a web browser window on the left and a code editor on the right. The browser displays a website for 'GAN단한 폰트 제작소' (GAN-danhan Font Studio) with a '사용법 안내' (Usage Guide) section. The code editor shows the source code for a Flask application that generates fonts using GAN. The code includes file paths, imports, and logic for processing font data and generating SVG and TTF files. The terminal at the bottom shows the command to run the application, which outputs 'HelloWorld.py' and 'HelloWorld'.

React App
localhost:3000

Home Font-Generation Font-AS Q&A

사용법 안내

Design your own font

GAN단한 폰트 제작소

1. 양식에 맞춰 몇 가지 단어를 작성하고 나만의 폰트를 제작해보세요.

2. 폰트 제작이 완료되면 이메일을 통해 전송해드립니다.

3. 만약 마음에 들지 않는 문자가 있으시면, A/S도 가능합니다!

Download Template

제공되는 템플릿을 다운받아 양식에 맞게 작성하세요.
템플릿 혹은 디지털 펜으로 작성 후 jpg 파일로 제출해주세요

Template.jpg

Template.pdf

A/S Service

제작된 폰트가 마음에 안드시나요???

A/S 신청하기

flask-dmfont - HelloWorld.py

```

print("crops pdf to images...")
t = Template(fileName=fileName,
              fontName=fontName,
              configPath="./config/template.json")
t.crop()

# pngs -> svg
print("png2svg starts")
png2svg = Png2Svg(fontName=fontName, is_in=True)
png2svg.proc()

# svg -> TTF
print("png2svg ends, svg2ttf starts")
svg2ttf = Svg2Ttf(fontName=fontName, is_in=True)
svg2ttf.proc()

# TODO - TTF -> APPLY DM-FONT
##### APPLY DM-FONT #####
print("applying dmfont(ttf->hdfs)...")
os.chdir('c:/Users/96032/PycharmProjects/flask-dmfont')
arg1 = "python -m scripts.prepare_dataset kor_data/fonts_dir/meta/kor_split.json data/"

p = subprocess.Popen(arg1, stdout=subprocess.PIPE, shell=True)
p.wait()

print("applying dmfont(pngs)...")
arg2 = "python evaluator.py WITHPTH checkpoints/korean-handwriting.pth generated_font/"
q = subprocess.Popen(arg2, stdout=subprocess.PIPE, shell=True)
q.wait()

os.chdir('c:/Users/96032/PycharmProjects/flask-dmfont/Capstone-project/FlaskProj')

# pngs -> svg
print("png2svg starts")
png2svg = Png2Svg(fontName=fontName, is_in=False)
##### APPLY DM-FONT #####
proc()

```

Terminal: Local - Local (3) - Local (2) - +
Microsoft Windows [Version 10.0.19042.1288]
(c) Microsoft Corporation. All rights reserved.

```

(Flask-dmfont) C:\Users\96032\PycharmProjects\flask-dmfont>conda activate env_flask
(env_flask) C:\Users\96032\PycharmProjects\flask-dmfont>cd Capstone-project
(env_flask) C:\Users\96032\PycharmProjects\flask-dmfont\Capstone-project>cd FlaskProj
(env_flask) C:\Users\96032\PycharmProjects\flask-dmfont\Capstone-project\FlaskProj>python HelloWorld.py
* Serving Flask app "HelloWorld" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 731-132-618
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

```

localhost:3000/form1

검색하려면 여기에 입력하십시오.

9°C 맑음 10:51 2021-11-01



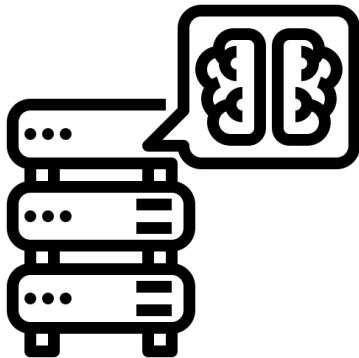
In Web application part,

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