# PinPlace: CNN based location image search
# And its adaptation to social network

TEAM H Week 8

## CNN Build

**Data processing**

HONG SEONGJUN

**Modify CNN model & improve accuracy**

CHE SEUNG YUN

## Front end

**Cover pages & listup pages**

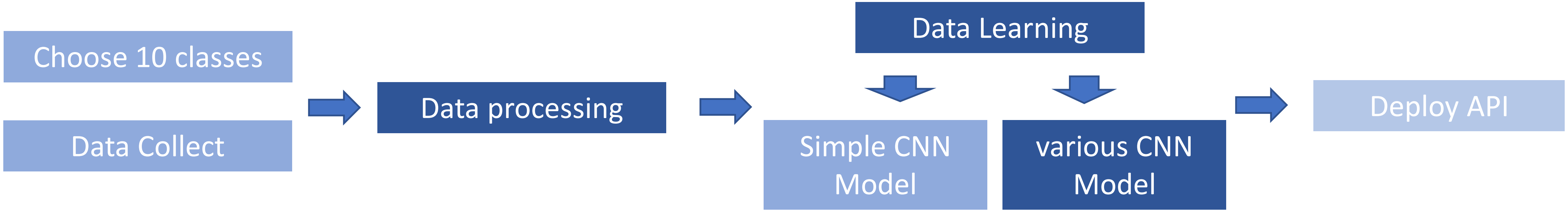
JEONG CHAEWON, LEE JI SEOP

# CNN Model build > Week 8

CNN Build

**Our  CNN model development process**

1. Collect and process data necessary for learning.
2. Train an appropriate artificial intelligence model using the processed learning data.
3. Deploy the trained model to utilize it in application.

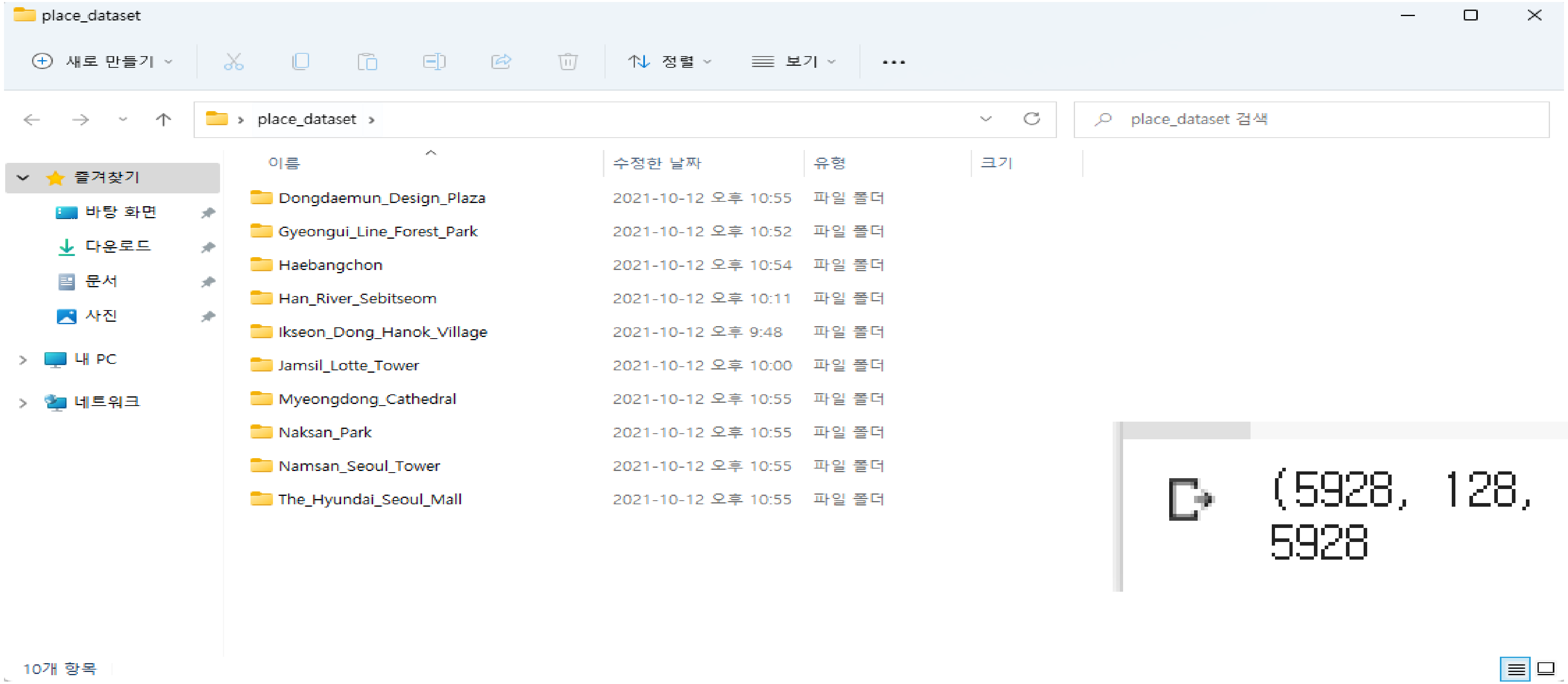


~ 10/14

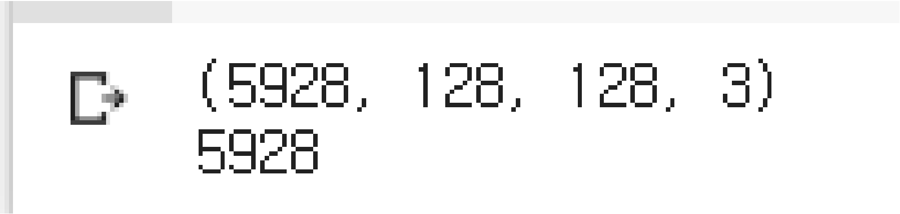
~ 10/28

# CNN Model build > Week 8

## Data processing



| 이름 | 수정한 날짜 | 유형 | 크기 |
|---|---|---|---|
| Dongdaemun_Design_Plaza | 2021-10-12 오후 10:55 | 파일 폴더 | |
| Gyeongui_Line_Forest_Park | 2021-10-12 오후 10:52 | 파일 폴더 | |
| Haebangchon | 2021-10-12 오후 10:54 | 파일 폴더 | |
| Han_River_Sebitseom | 2021-10-12 오후 10:11 | 파일 폴더 | |
| Ikseon_Dong_Hanok_Village | 2021-10-12 오후 9:48 | 파일 폴더 | |
| Jamsil_Lotte_Tower | 2021-10-12 오후 10:00 | 파일 폴더 | |
| Myeongdong_Cathedral | 2021-10-12 오후 10:55 | 파일 폴더 | |
| Naksan_Park | 2021-10-12 오후 10:55 | 파일 폴더 | |
| Namsan_Seoul_Tower | 2021-10-12 오후 10:55 | 파일 폴더 | |
| The_Hyundai_Seoul_Mall | 2021-10-12 오후 10:55 | 파일 폴더 | |



```
(5928, 128, 128, 3)
5928
```

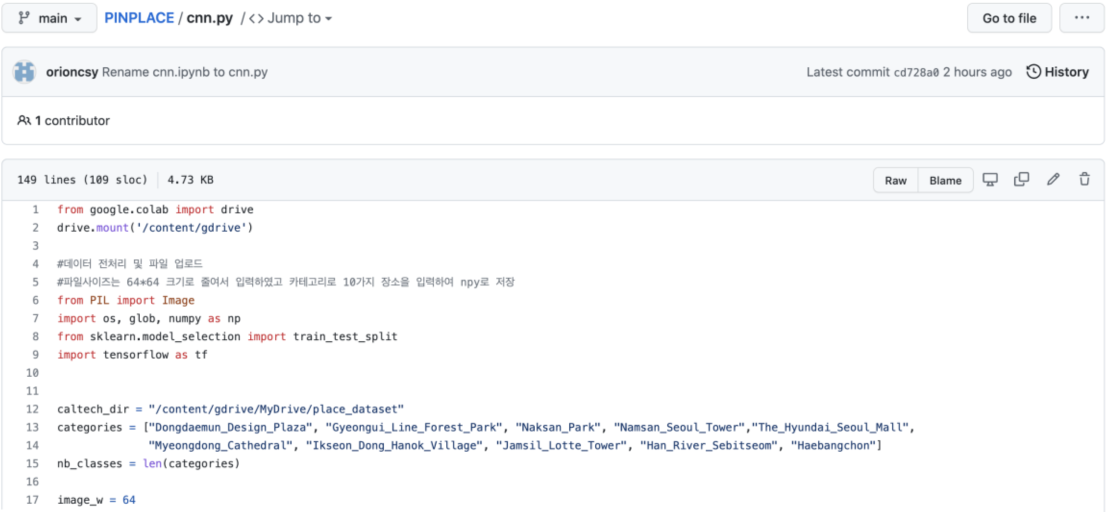
► Remove irrelevant images more carefully

After image processing, 600 images remain for each class.

► We tried to crawl Instagram images, but we failed.

# CNN Model build > Week 8

Modified CNN model & test it



PINPLACE / cnn.py / <> Jump to

orioncsy Rename cnn.ipynb to cnn.py

Latest commit cd728a0 2 hours ago

1 contributor

149 lines (109 sloc) | 4.73 KB

```
from google.colab import drive
drive.mount('/content/gdrive')

#데이터 전처리 및 파일 업로드
#파일사이즈는 64*64 크기로 줄여서 입력하였고 카테고리로 10가지 장소을 입력하여 npy로 저장
from PIL import Image
import os, glob, numpy as np
from sklearn.model_selection import train_test_split
import tensorflow as tf


caltech_dir = "/content/gdrive/MyDrive/place_dataset"
categories = ["Dongdaemun_Design_Plaza", "Gyeongui_Line_Forest_Park", "Naksan_Park", "Namsan_Seoul_Tower","The_Hyundai_Seoul_Mall",
              "Myeongdong_Cathedral", "Ikseon_Dong_Hanok_Village", "Jamsil_Lotte_Tower", "Han_River_Sebitseom", "Haebangchon"]
nb_classes = len(categories)

image_w = 64
```

Library

TensorFlow

Platform

colab

► Modify last CNN model using TenserFlow on google colab.

The collected and processed learning data is trained on the modified CNN model.

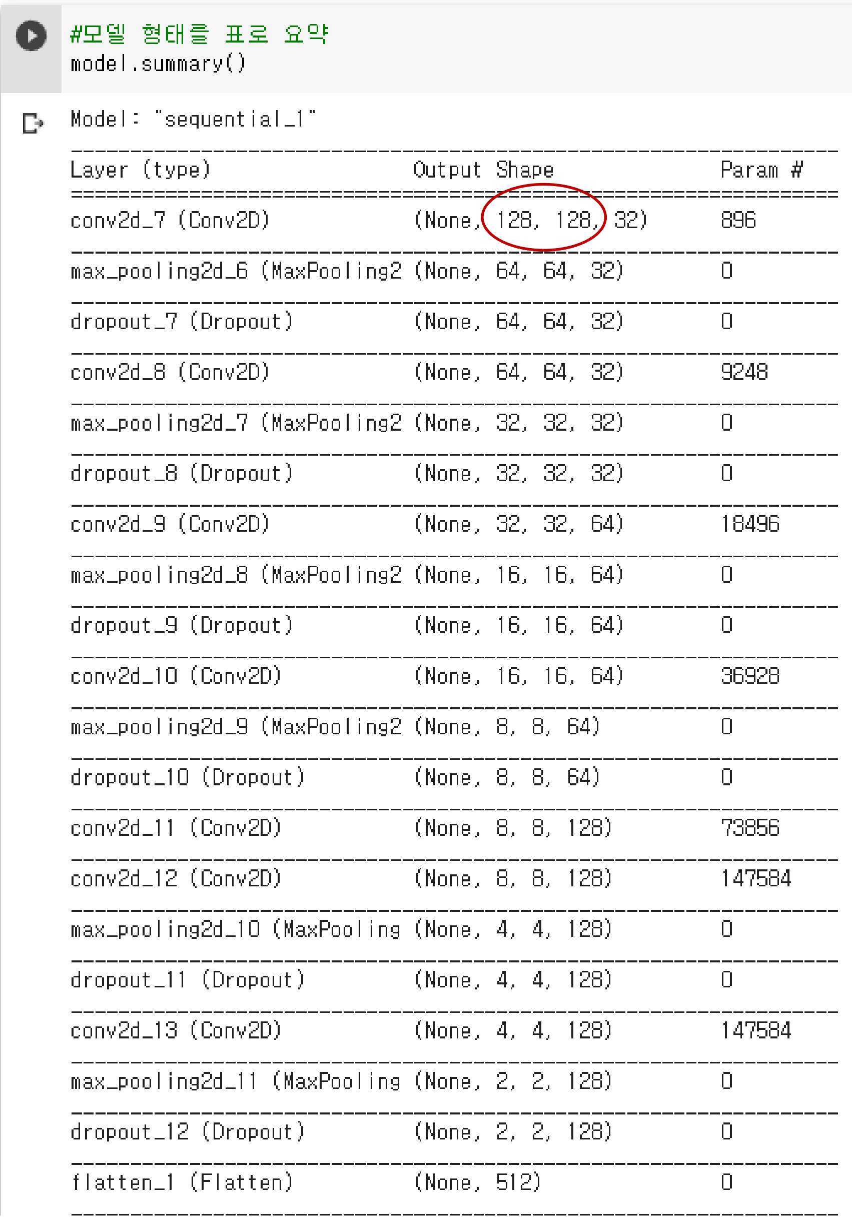
Check the accuracy.

# CNN Model build > Week 8

## Modified CNN model & test it

### 10 classses

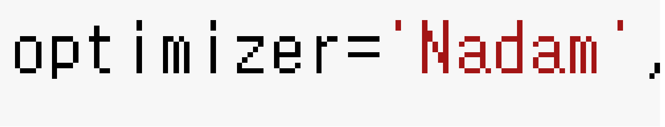
- Dongdaemun_Design_Plaza
- Gyeongui_Line_Forest_Park
- Haebangchon
- Han_River_Sebitseom
- Ikseon_Dong_Hanok_Village
- Jamsil_Lotte_Tower
- Myeongdong_Cathedral
- Naksan_Park
- Namsan_Seoul_Tower
- The_Hyundai_Seoul_Mall



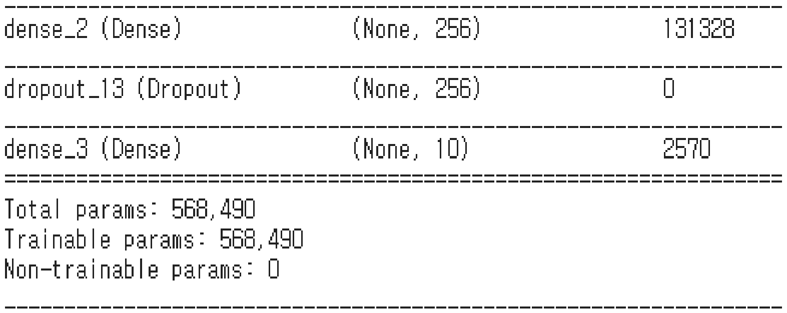
```
#모델 형태를 표로 요약
model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_7 (Conv2D)            (None, 128, 128, 32)      896
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 64, 64, 32)        0
_________________________________________________________________
dropout_7 (Dropout)          (None, 64, 64, 32)        0
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 64, 64, 32)        9248
_________________________________________________________________
max_pooling2d_7 (MaxPooling2 (None, 32, 32, 32)        0
_________________________________________________________________
dropout_8 (Dropout)          (None, 32, 32, 32)        0
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 32, 32, 64)        18496
_________________________________________________________________
max_pooling2d_8 (MaxPooling2 (None, 16, 16, 64)        0
_________________________________________________________________
dropout_9 (Dropout)          (None, 16, 16, 64)        0
_________________________________________________________________
conv2d_10 (Conv2D)           (None, 16, 16, 64)        36928
_________________________________________________________________
max_pooling2d_9 (MaxPooling2 (None, 8, 8, 64)          0
_________________________________________________________________
dropout_10 (Dropout)         (None, 8, 8, 64)          0
_________________________________________________________________
conv2d_11 (Conv2D)           (None, 8, 8, 128)         73856
_________________________________________________________________
conv2d_12 (Conv2D)           (None, 8, 8, 128)         147584
_________________________________________________________________
max_pooling2d_10 (MaxPooling (None, 4, 4, 128)         0
_________________________________________________________________
dropout_11 (Dropout)         (None, 4, 4, 128)         0
_________________________________________________________________
conv2d_13 (Conv2D)           (None, 4, 4, 128)         147584
_________________________________________________________________
max_pooling2d_11 (MaxPooling (None, 2, 2, 128)         0
_________________________________________________________________
dropout_12 (Dropout)         (None, 2, 2, 128)         0
_________________________________________________________________
flatten_1 (Flatten)          (None, 512)               0
_________________________________________________________________
```

### Simple CNN model



```
optimizer='Nadam',
```

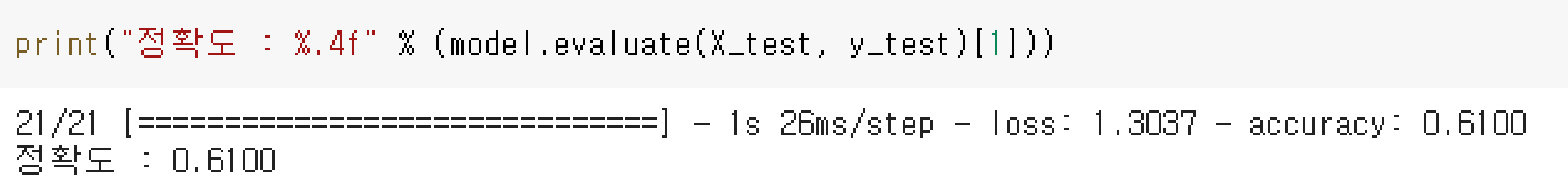


```
_________________________________________________________________
dense_2 (Dense)              (None, 256)               131328
_________________________________________________________________
dropout_13 (Dropout)         (None, 256)               0
_________________________________________________________________
dense_3 (Dense)              (None, 10)                2570
=================================================================
Total params: 568,490
Trainable params: 568,490
Non-trainable params: 0
_________________________________________________________________
```

# CNN Model build > Week 8

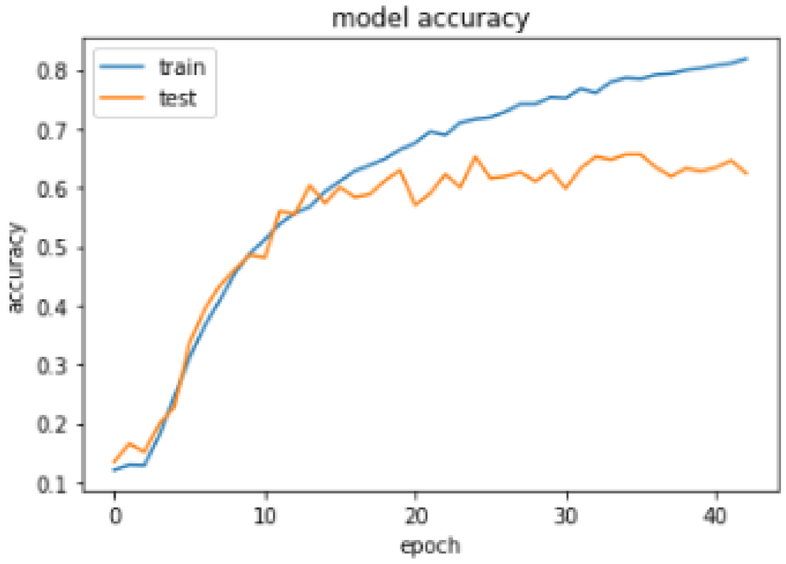
## Modified CNN model & test it

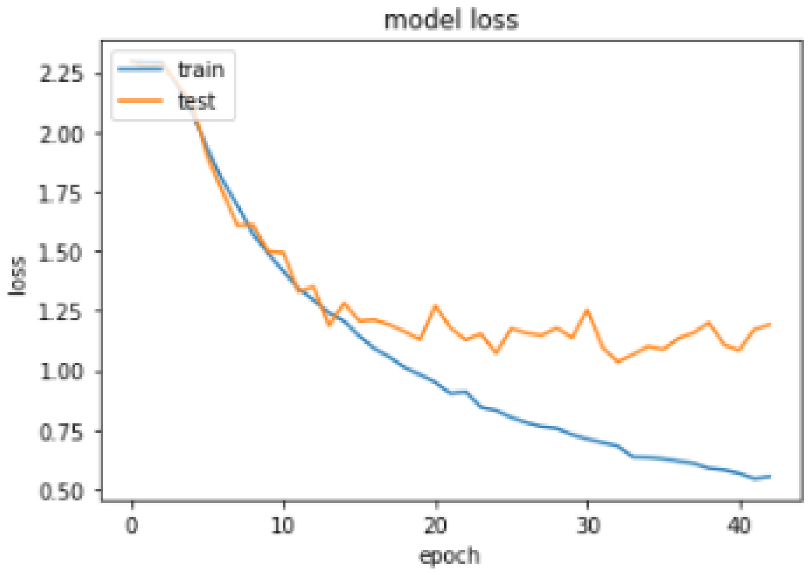
- Accuracy of model is 0.6100



```
print("정확도 : %.4f" % (model.evaluate(X_test, y_test)[1]))
```

```
21/21 [==============================] - 1s 26ms/step - loss: 1.3037 - accuracy: 0.6100
정확도 : 0.6100
```

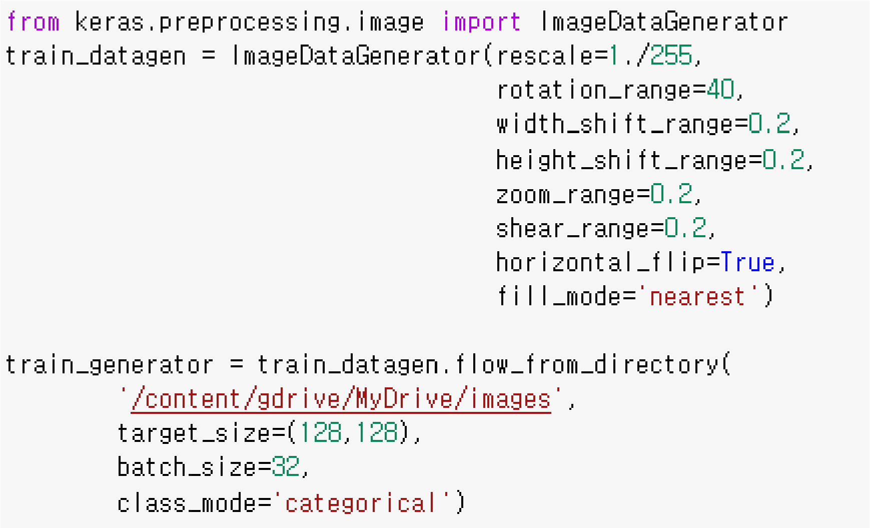




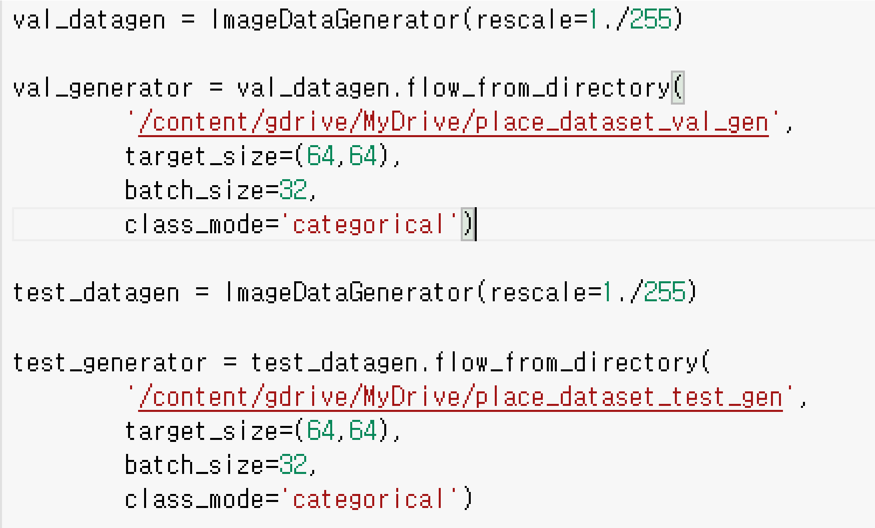
# CNN Model build > Week 8

## Modified CNN model & test it

### We tried data augmentation



```
from keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale=1./255,
                                   rotation_range=40,
                                   width_shift_range=0.2,
                                   height_shift_range=0.2,
                                   zoom_range=0.2,
                                   shear_range=0.2,
                                   horizontal_flip=True,
                                   fill_mode='nearest')

train_generator = train_datagen.flow_from_directory(
        '/content/gdrive/MyDrive/images',
        target_size=(128,128),
        batch_size=32,
        class_mode='categorical')
```

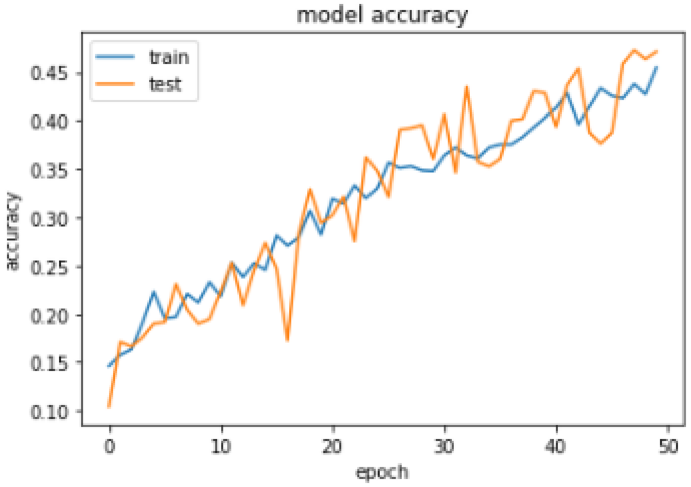


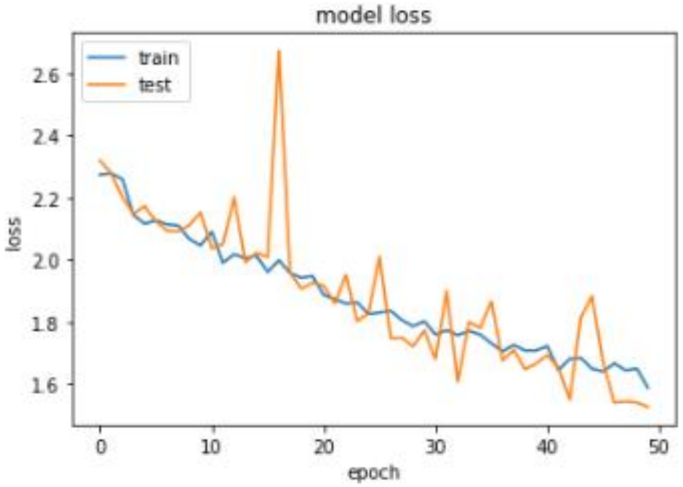
```
val_datagen = ImageDataGenerator(rescale=1./255)

val_generator = val_datagen.flow_from_directory(
        '/content/gdrive/MyDrive/place_dataset_val_gen',
        target_size=(64,64),
        batch_size=32,
        class_mode='categorical')

test_datagen = ImageDataGenerator(rescale=1./255)

test_generator = test_datagen.flow_from_directory(
        '/content/gdrive/MyDrive/place_dataset_test_gen',
        target_size=(64,64),
        batch_size=32,
        class_mode='categorical')
```

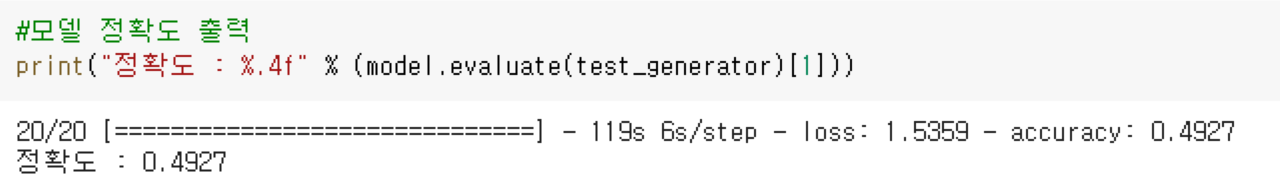
# CNN Model build > Week 8

## Modified CNN model & test it

► We tried data augmentation







```
#모델 정확도 출력
print("정확도 : %.4f" % (model.evaluate(test_generator)[1]))
```

```
20/20 [==============================] - 119s 6s/step - loss: 1.5359 - accuracy: 0.4927
정확도 : 0.4927
```

**We are in process in augmentation version**

1. Accuracy is not enough as about 54%.
2. But we can check decrease in difference between test accuracy and learning accuracy
3. We need to modify the code.
4. Deploy the trained model to utilize it in application.

# CNN Model build > Week 8

## Next week

1. Improve the accuracy by changing CNN models and by modifying augmentation version

   **CHE SEUNG YUN**

2. Define classes more in detail

   **HONG SEONG JUN**

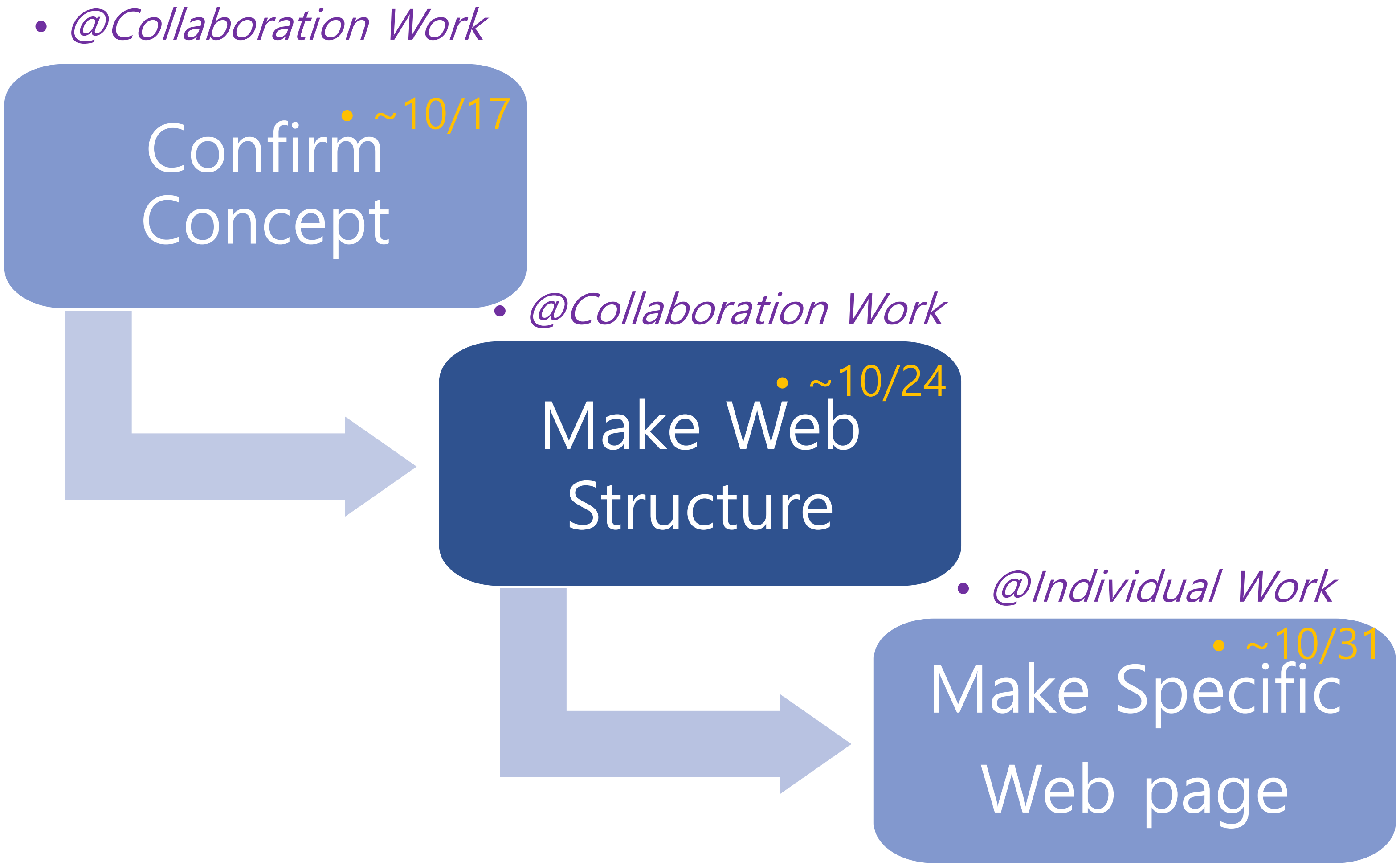
3. Save the model by file

   **CHE SEUNG YUN**

4. check the model works well

   **UHM JI YONG**

# Web Programming Part > October Plan



- *@Collaboration Work*

- ~10/17

Confirm Concept

- *@Collaboration Work*

- ~10/24

Make Web Structure

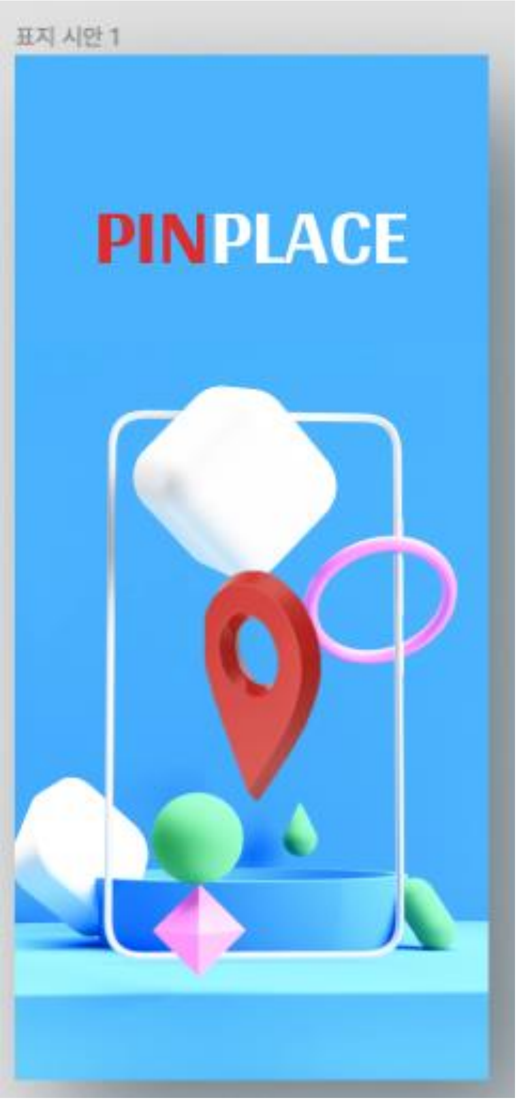
- *@Individual Work*

- ~10/31

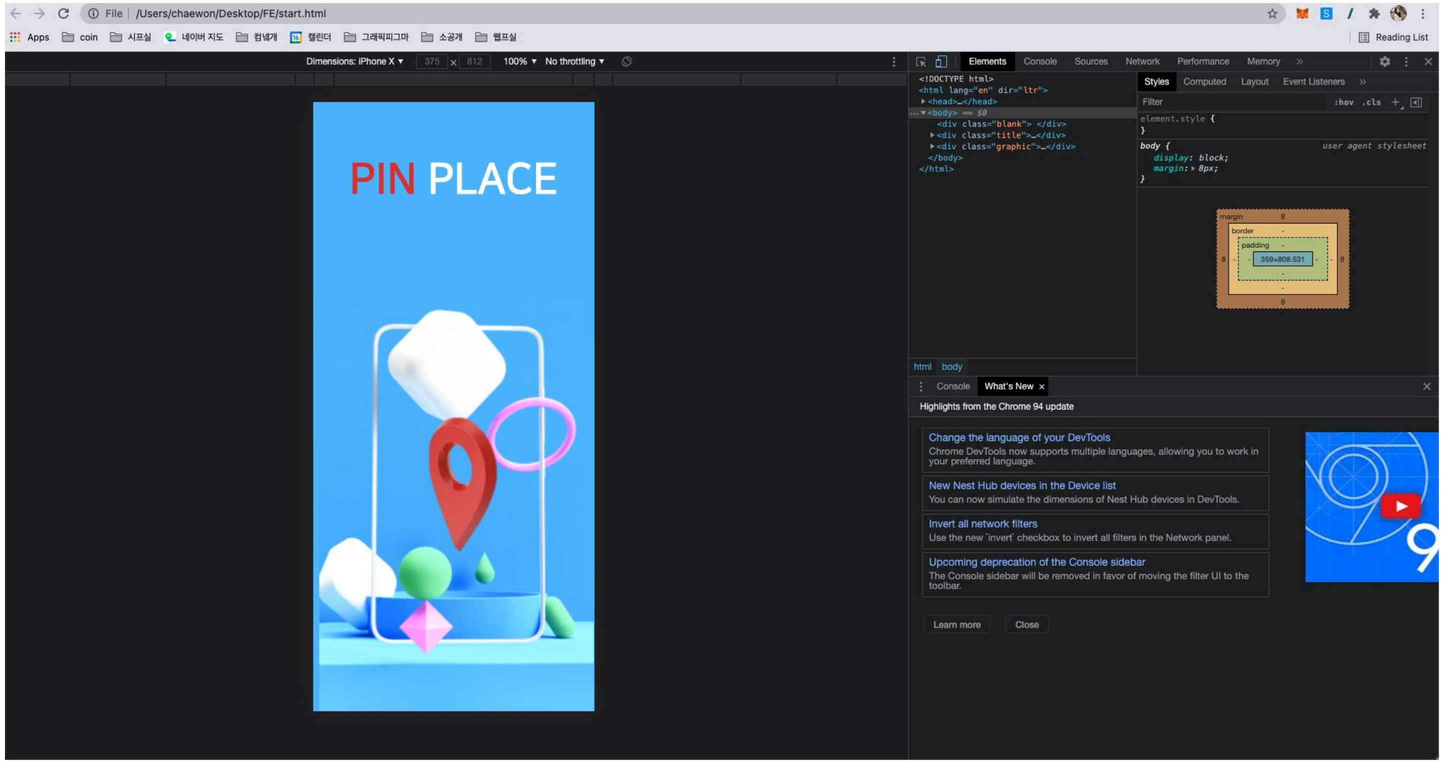
Make Specific Web page

Front end

# Web Programming Part > Structure Buildup



## # Cover Page



File | /Users/chaewon/Desktop/FE/start.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
 ▸<head>…</head>
…▾<body> == $0
    <div class="blank"> </div>
   ▸<div class="title">…</div>
   ▸<div class="graphic">…</div>
  </body>
</html>
```

```
element.style {
}
body {                                  user agent stylesheet
  display: block;
  margin: ▸ 8px;
}
```

Highlights from the Chrome 94 update

Change the language of your DevTools
Chrome DevTools now supports multiple languages, allowing you to work in your preferred language.

New Nest Hub devices in the Device list
You can now simulate the dimensions of Nest Hub devices in DevTools.

Invert all network filters
Use the new `invert` checkbox to invert all filters in the Network panel.

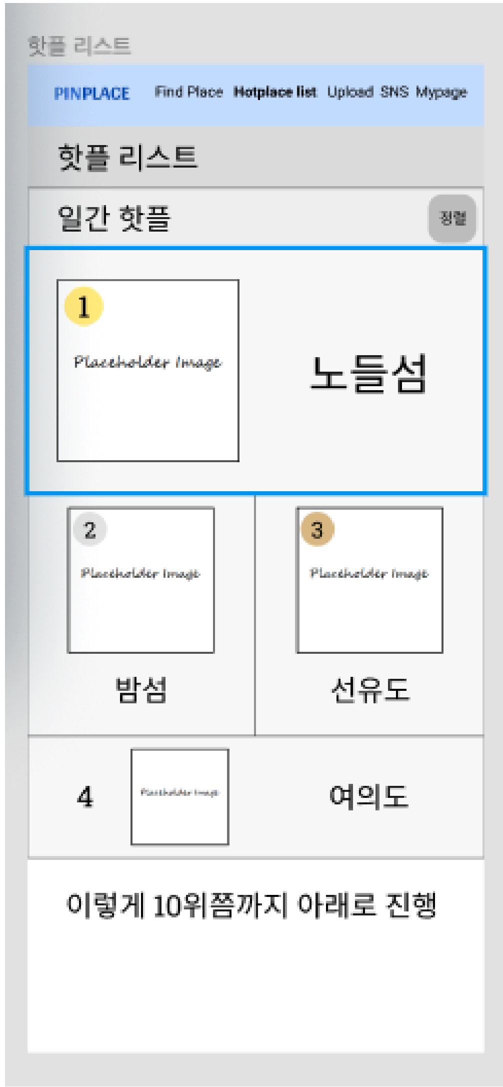
Upcoming deprecation of the Console sidebar
The Console sidebar will be removed in favor of moving the filter UI to the toolbar.

# # Cover Page

- Static splash image
- Problem Observed:
- The splash image is optimized for only one screen size (375x812 px)
- The image cannot be resized, since resizing may cause uncomfortable looking
- Possible Solutions:
- Make a few pre-resized images for various screen sizes
- Extend the image from its boundary
- Recreate as a vector image
- Future work:
- Make an introductory popup after the splash image

# Web Programming Part > Structure Buildup

# Common Layout for Subpages
# Hot Place List Subpage



핫플 리스트

PINPLACE Find Place **Hotplace list** Upload SNS Mypage

핫플 리스트

일간 핫플 정렬

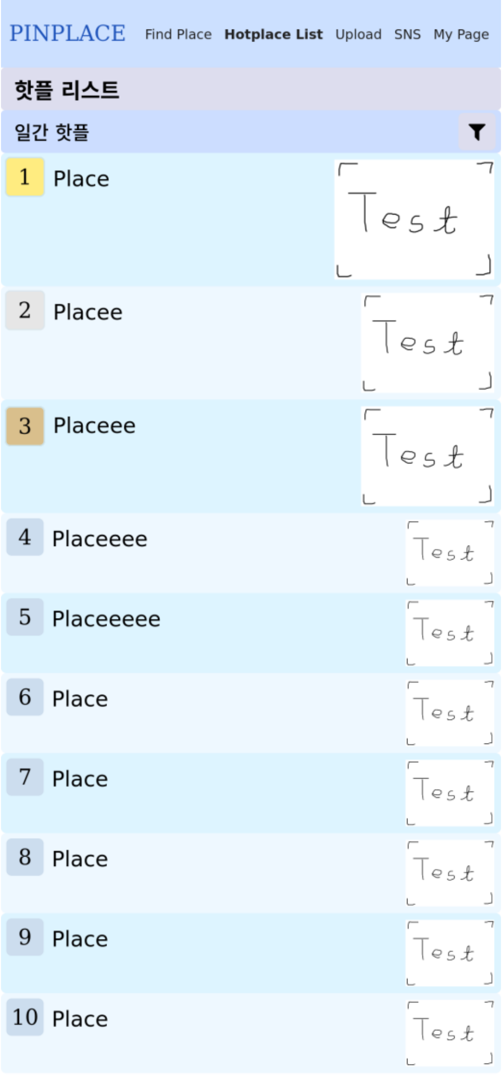
1 Placeholder Image 노들섬

2 Placeholder Image 밤섬

3 Placeholder Image 선유도

4 Placeholder Image 여의도

이렇게 10위쯤까지 아래로 진행



PINPLACE Find Place **Hotplace List** Upload SNS My Page

**핫플 리스트**

일간 핫플

1 Place Test

2 Placee Test

3 Placeee Test

4 Placeeee Test

5 Placeeeee Test

6 Place Test

7 Place Test

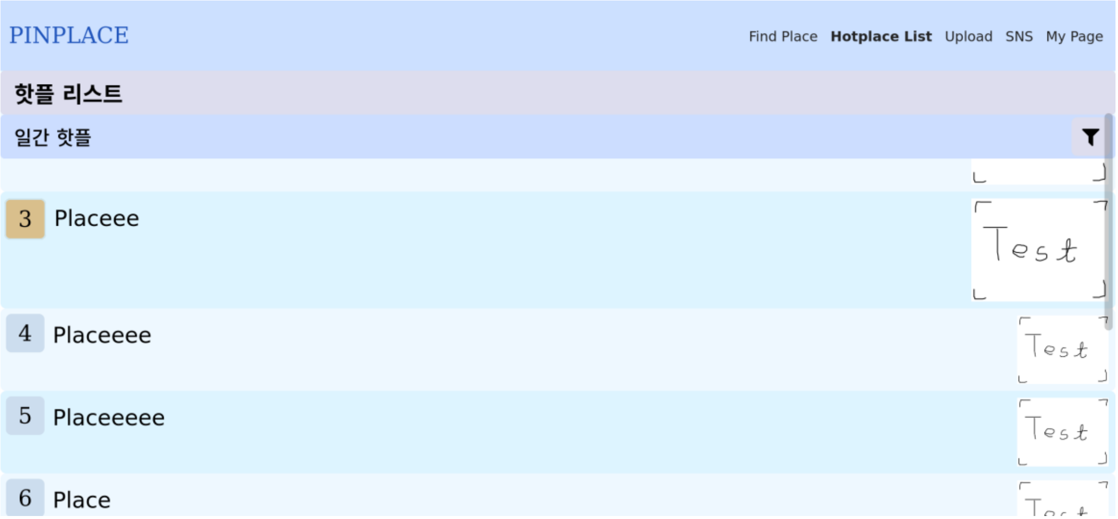
8 Place Test

9 Place Test

10 Place Test

# Web Programming Part > Structure Buildup

## # Common Layout for Subpages



핫플 리스트

일간 핫플

3 Placeee Test

4 Placeeee Test

5 Placeeeee Test

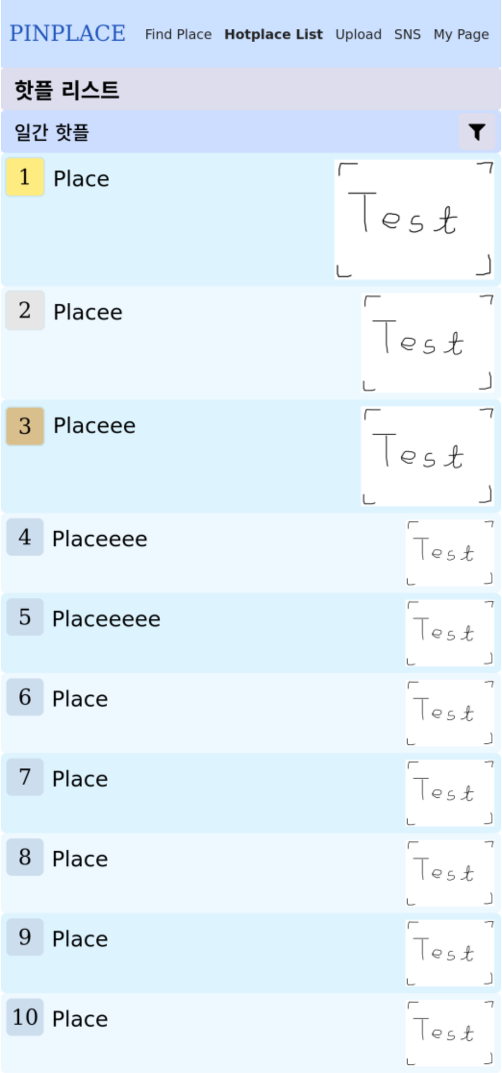
6 Place

- Prepared for various screen sizes
- Fixed the position of the header menu bar
- Color scheme for vertical listings
- The menu bar and the subpage are separate blocks
    - On each subpage’s activation, the new subpage block will replace the old one

# Web Programming Part > Structure Buildup

## # Hot Place List Subpage

- Designed for ten places and their representative images, one for each
- First, second, and third entries have larger images for emphasis
- General structure for each entry is identical, so the listing with actual places can be made automatically by JavaScript



핫플 리스트

일간 핫플

1 Place

2 Placee

3 Placeee

4 Placeeee

5 Placeeeee

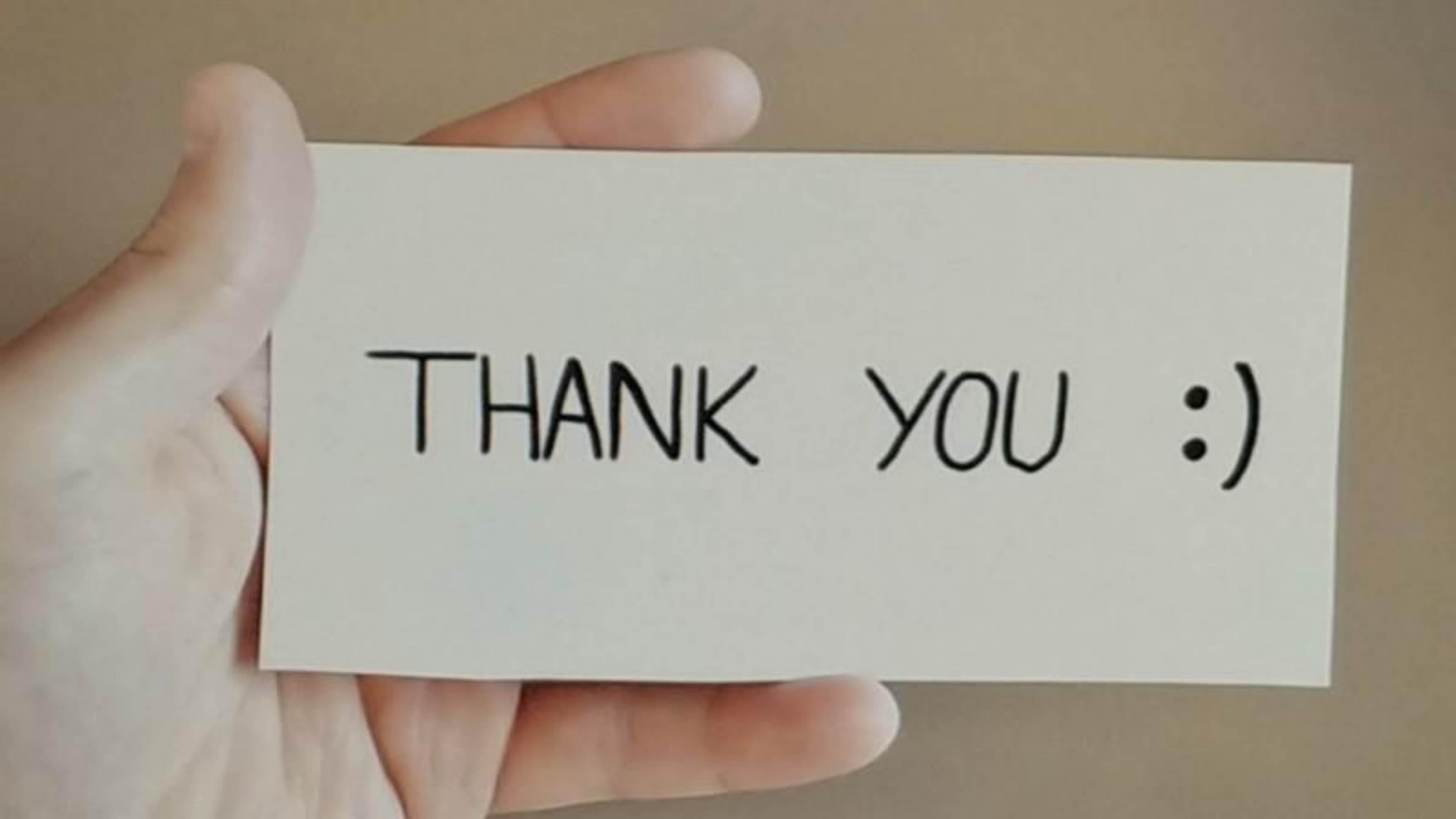
6 Place

7 Place

8 Place

9 Place

10 Place



THANK YOU :)