

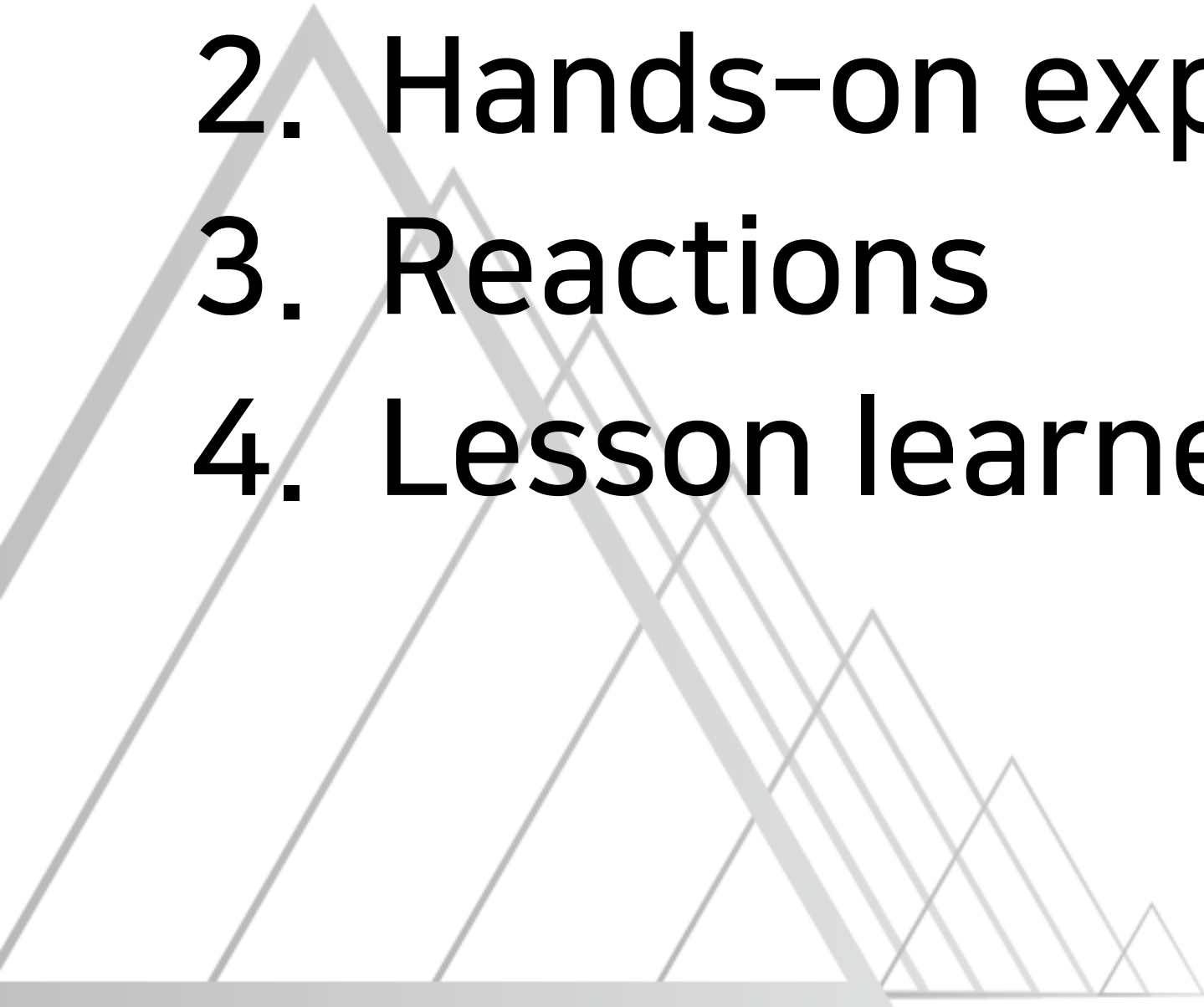
# TensorFlow로 디자인시스템 만들기

(Using TensorFlow JS to build an AI-powered design system)



# CONTENTS

1. Motivation
2. Hands-on experience
3. Reactions
4. Lesson learned

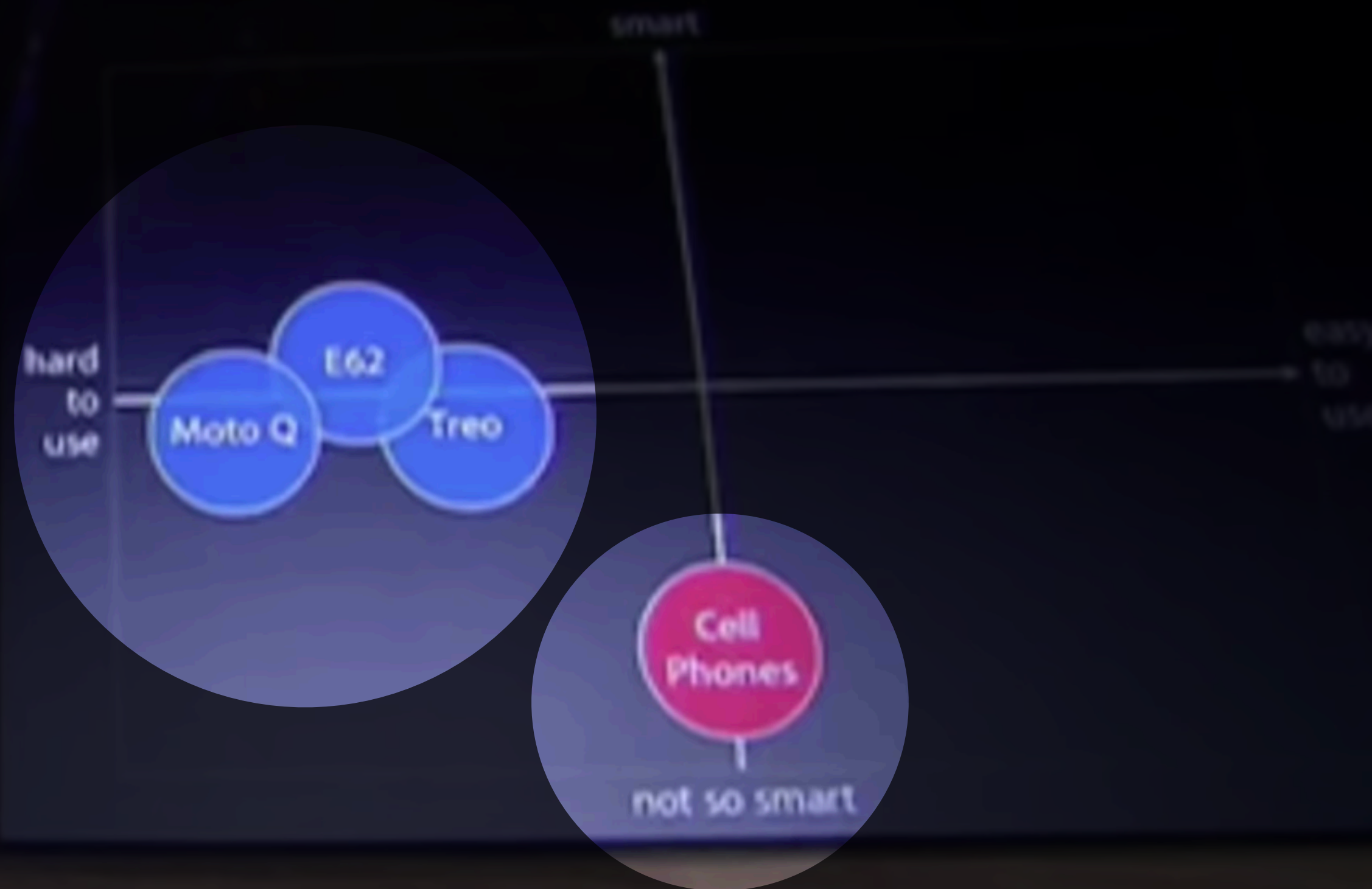


# Motivation

# Phone experiences (2007)

Not so smart

A little smarter  
but harder to use



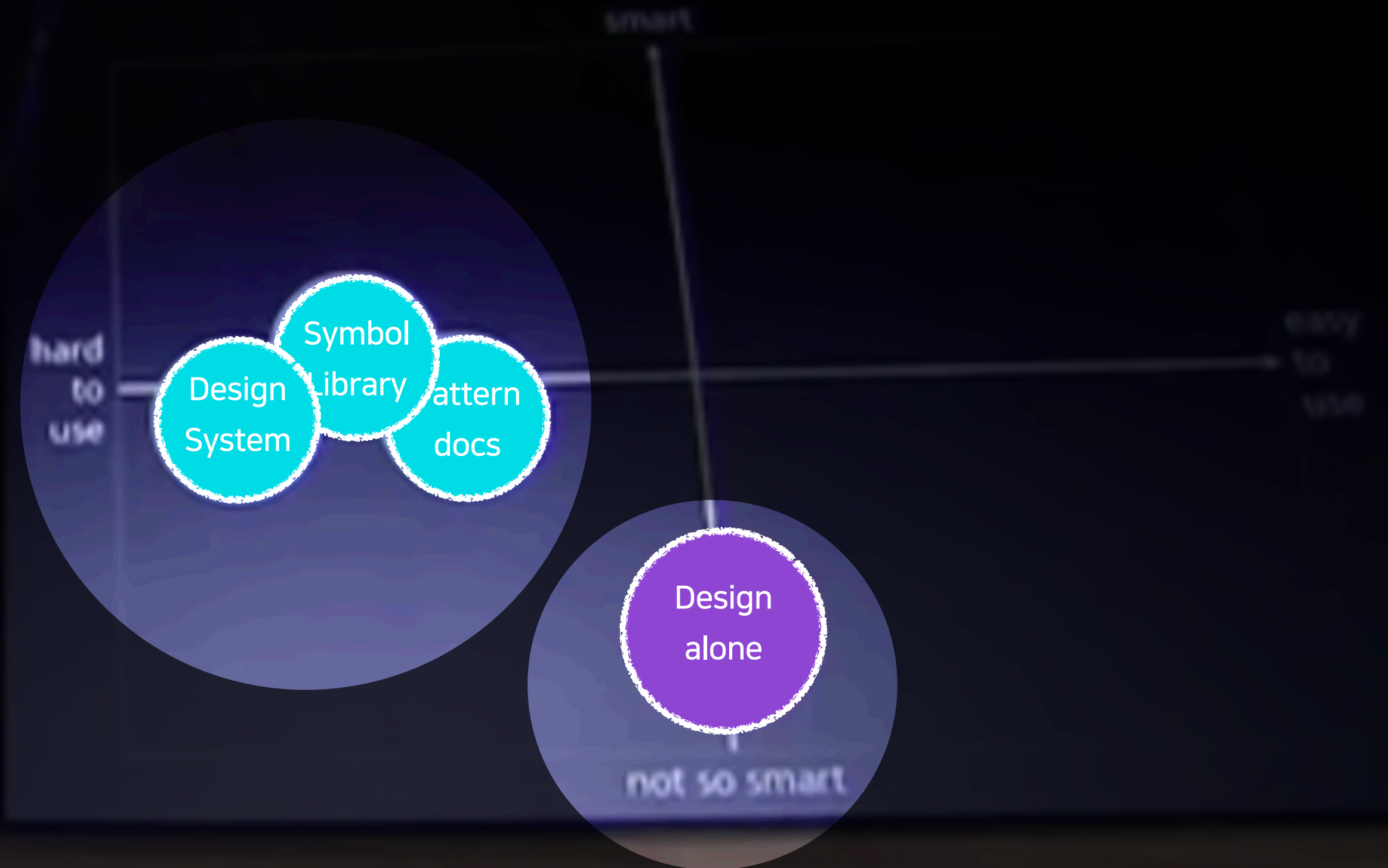
# Designing experiences (2021)

디자인 파일로 작업하는 것은?

Not so smart

라이브러리를 사용하는 것은?

A little smarter  
but harder to use



## Badges

Documentation and examples for badges, our small count and labeling component.

### Examples

Badges scale to match the size of the immediate parent element by using relative font sizing and `em` units.

Example heading **New**

Example heading **New**

Example heading **New**

Example heading **New**

Example heading **New**

## Breadcrumb

Indicate the current page's location within a navigational hierarchy that automatically adds separators via CSS.

### Examples

Home

Home / Library

Home / Library / Data

## Button group

Group a series of buttons together on a single line with the button group, and super-power them with JavaScript.

### Basic example

Wrap a series of buttons with `btn` in `btn-group`. Add on optional JavaScript radio and checkbox style behavior with our [buttons plugin](#).

Left Middle Right

### Button toolbar

Combine sets of button groups into button toolbars for more complex components. Use utility classes as needed to space out groups, buttons, and more.

1 2 3 4 5 6 7 8

## Buttons

Use Bootstrap's custom button styles for actions in forms, dialogs, and more with support for disabled, active, and more.

### Examples

Bootstrap includes several predefined button styles, each serving its own semantic purpose, with a few exceptions.

Primary Secondary Success Danger Warning Info Light Dark Link

### Outline buttons

Bootstrap includes several predefined button styles, each serving its own semantic purpose, with a few exceptions.

Primary Secondary Success Danger Warning Info Dark

라이브러리가 점점 커질수록..  
사람이 학습해야하는 것이 늘어난다.

We'll never share your email with anyone else.

Password

Placeholder

Check me out

Label

Select

Message

Label

Textarea

Message

Label  Label

### How it works

Progress components are built with two HTML elements, some CSS to set the width, and a few attributes. We don't use the `HTML5 <progress>` element, ensuring you can stack progress bars, animate them, and place text labels over them.

Progress bar with varying widths and colors.

### Labels

Add labels to your progress bars by placing text within the `progress-bar`.

Progress bar with a label.

### Height

We only set a `height` value on the `progress`, so if you change that value the inner `progress-bar` will automatically resize accordingly.

Takes the basic nav from above and adds the `nav-tabs` class to generate a tabbed interface. Use them to create tabbable regions with our [tab JavaScript plugin](#).

Active Link Link Disabled

### Pills

Take that same HTML, but use `nav-pills` instead.

Active Link Link Disabled

### Fill and justify

Force your `nav's` contents to extend the full available width with one of two modifier classes. To proportionally fill all available space with your `nav-items`, use `nav-fill`. Notice that all horizontal space is occupied, but not every `nav-item` has the same width.

Active Much longer nav link Link Disabled

## Navbar

Documentation and examples for Bootstrap's powerful, responsive navigation header, including support for our collapse plugin, branding, navigation, and more.

### Brand

`navbar-brand` can be applied to most elements, but an anchor works best as some elements might not be visible on all devices.

Navbar Home Link Dropdown Disabled Search

Bootstrap Home Documentation

### Color schemes

Using the navbar has never been easier thanks to the combination of theming classes and `background-color-light` for use with light background colors, or `navbar-dark` for dark background colors. Then, customize the text and links colors.

Navbar Home Features Pricing About Search

Navbar Home Features Pricing About Search

Navbar Home Features Pricing About Search

### Responsive behaviors

Navbars can utilize `navbar-toggleable`, `navbar-collapse`, and `navbar-expand` classes to choose when to show or hide particular parts of the navigation.

### Toggler

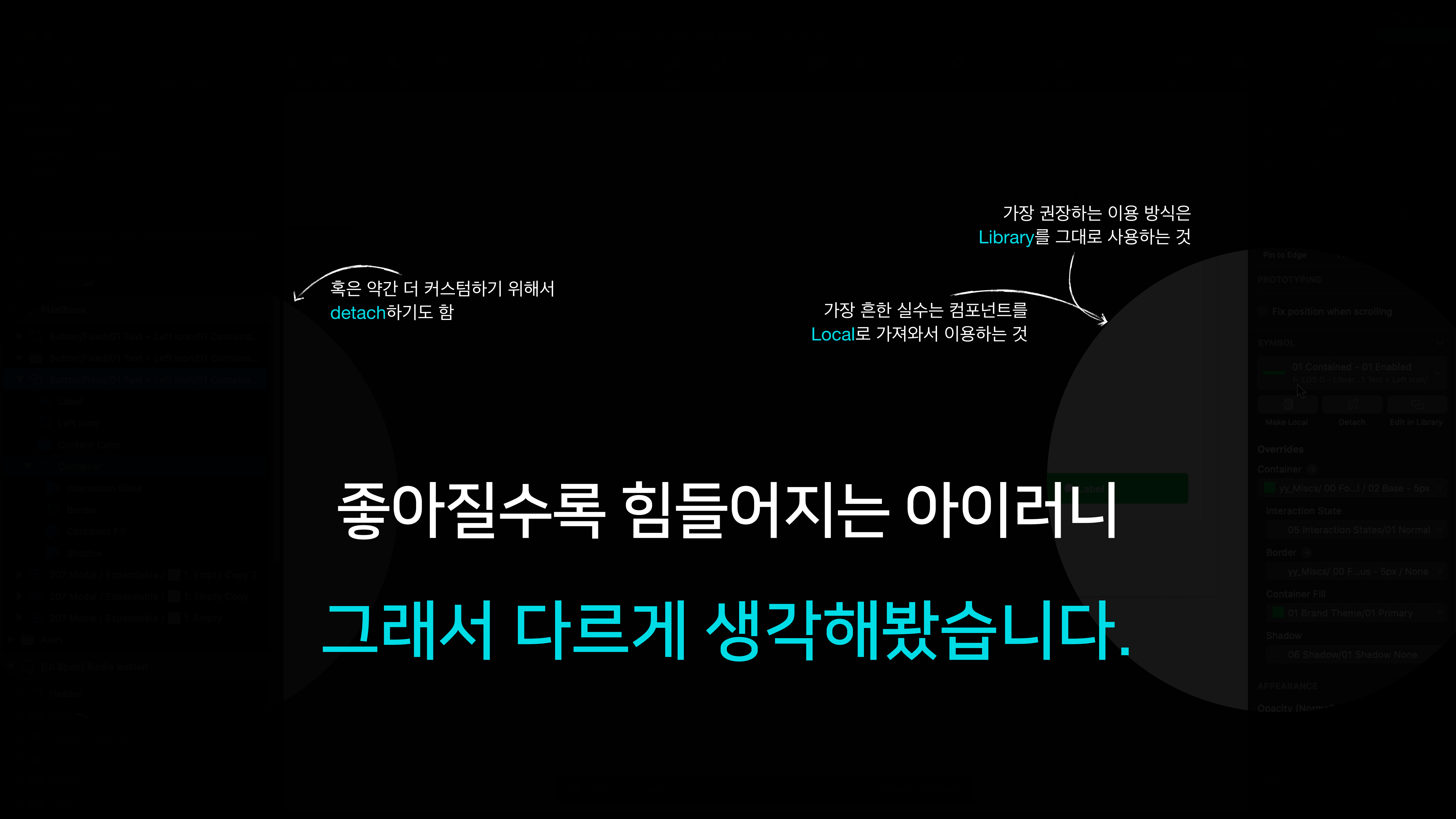
Navbar togglers are left-aligned by default, but should they follow a sibling element like a `navbar-brand`, they can be right-aligned. Reversing your markup will reverse the placement of the toggler. Below are examples of different t

Navbar

Source: Figma Community



그래서 신기능이 점점 추가되지만..  
역시 사람이 눈으로 맞춰야 한다.



혹은 약간 더 커스텀하기 위해서 detach하기도 함

가장 권장하는 이용 방식은 Library를 그대로 사용하는 것

가장 흔한 실수는 컴포넌트를 Local로 가져와서 이용하는 것

# 좋아질수록 힘들어지는 아이러니

# 그래서 다르게 생각해봤습니다.

Pin to Edge

PROTOTYPING

Fix position when scrolling

SYMBOL

01 Contained - 01 Enabled

LDS G - Librar... 1 Text + Left Icon

Make Local Detach Edit in Library

Overrides

Container

yy\_Miscs/ 00 Fo...l / 02 Base - 5px

Interaction State

05 Interaction States/01 Normal

Border

yy\_Miscs/ 00 F...us - 5px / None

Container Fill

01 Brand Theme/01 Primary

Shadow

06 Shadow/01 Shadow None

APPEARANCE

Opacity (Norma...



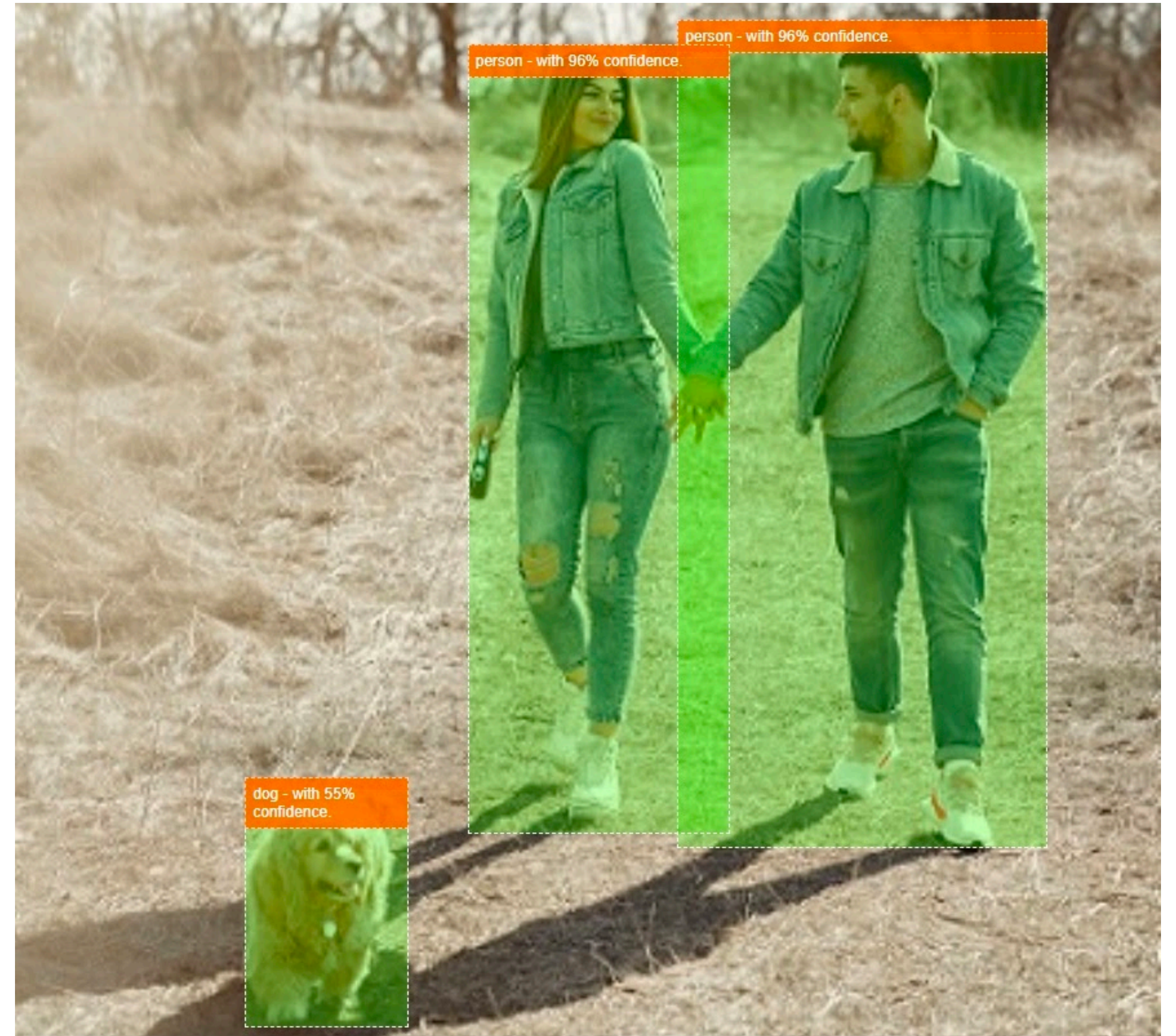
iPhone, a leapfrog product  
that is way smarter  
and super easy to use



# Motivation

## 이게 자동으로 안될까?

- 컴퓨터비전 기술을 사용해서
- 화면을 자동으로 분석할 수 있다면,
- 디자인 맞춤법 검사도 가능하잖을까?



② 디자인 라이브러리의  
맞춤법을 확인해서,

① 컴퓨터비전으로  
화면을 분석하고,

③ 오류를 자동으로  
찾아준다면?

GET TFJS GET ZEPLIN COMPARE

SELECT DATE

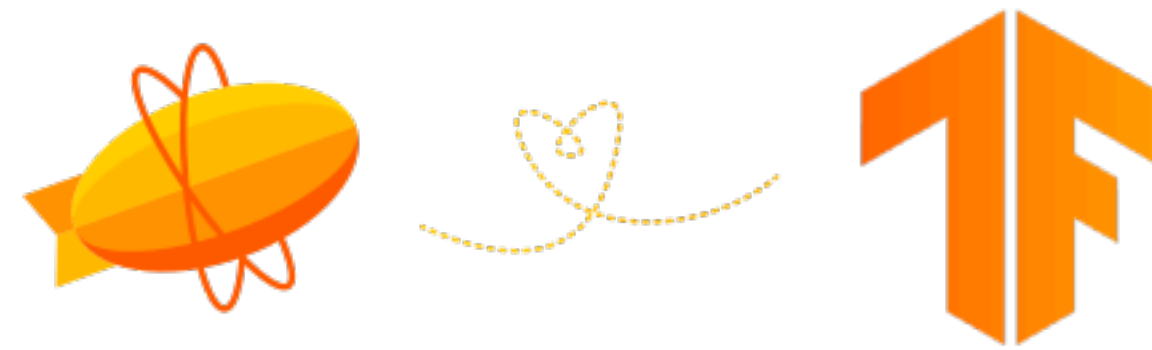
Mon, Nov 17

November 2021

2011	2012	2013
2014	2015	2016
2017	2018	2019
2020	2021	2022
2023	2024	2025
2026	2027	2028

CANCEL OK





# Zeplin ML

TensorFlow JS와 Zeplin API를 활용한  
디자인 맞춤법 검사 프로젝트

# Hands-on experience





## Please note!

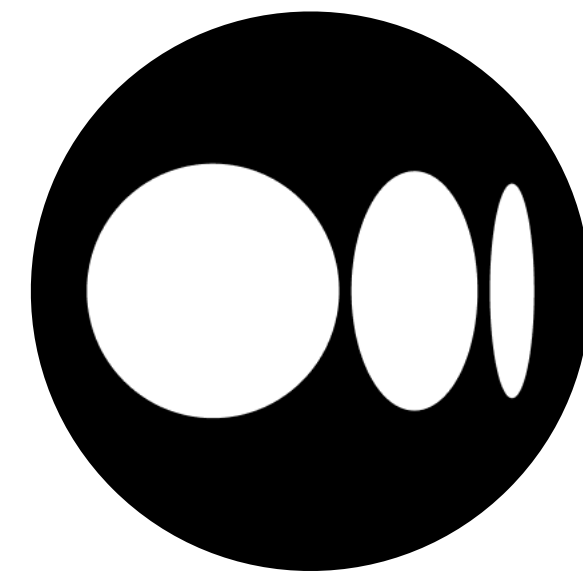
- 개인적인 경험을 바탕으로 데이터 준비부터 및 모델 훈련까지의 과정을 설명합니다.
- 보다 깊은 내용은 DEVIEW의 다른 새션을 참고해주세요!
- 상세한 노트북, 스크립트, 설명 등은 아래의 링크를 같이 참고해주세요.



[Colab 노트북](#)



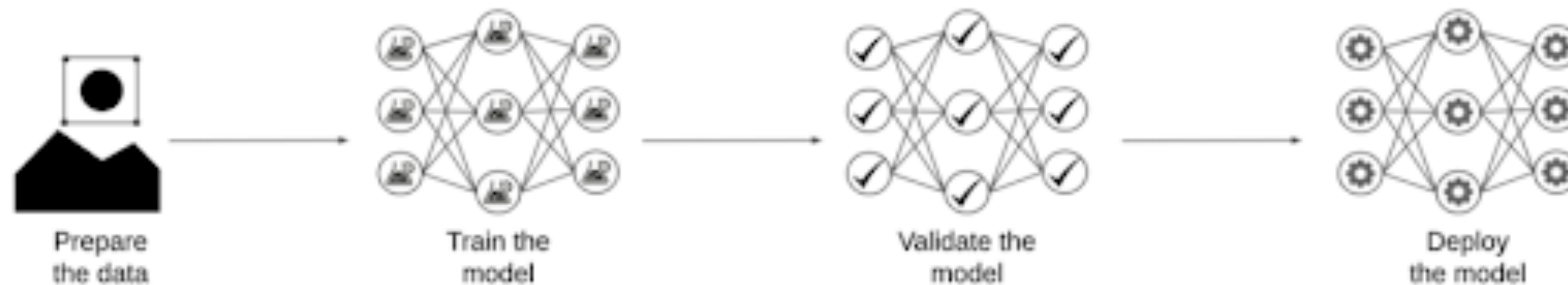
[Gist](#)



[Medium post](#)

# Hands-on experience

## Object Detection 파이프라인





# Hands-on experience

## Prepare the data

- 충분한 양의 고품질 UI 이미지 확보
- UI컴포넌트 정의 및 데이터 라벨링
- TensorFlow용 TFrecord로 변환





Browse

MY LIBRARY

Illustrations

Signups

Tooltips

+ New Library

Mobbin

Latest Mobile Design Patterns

About

Contact

# Browse

APPS PATTERNS

Filter by Categories



**Truebill**

Subscription & Expense Manager



<https://mobbin.design/>

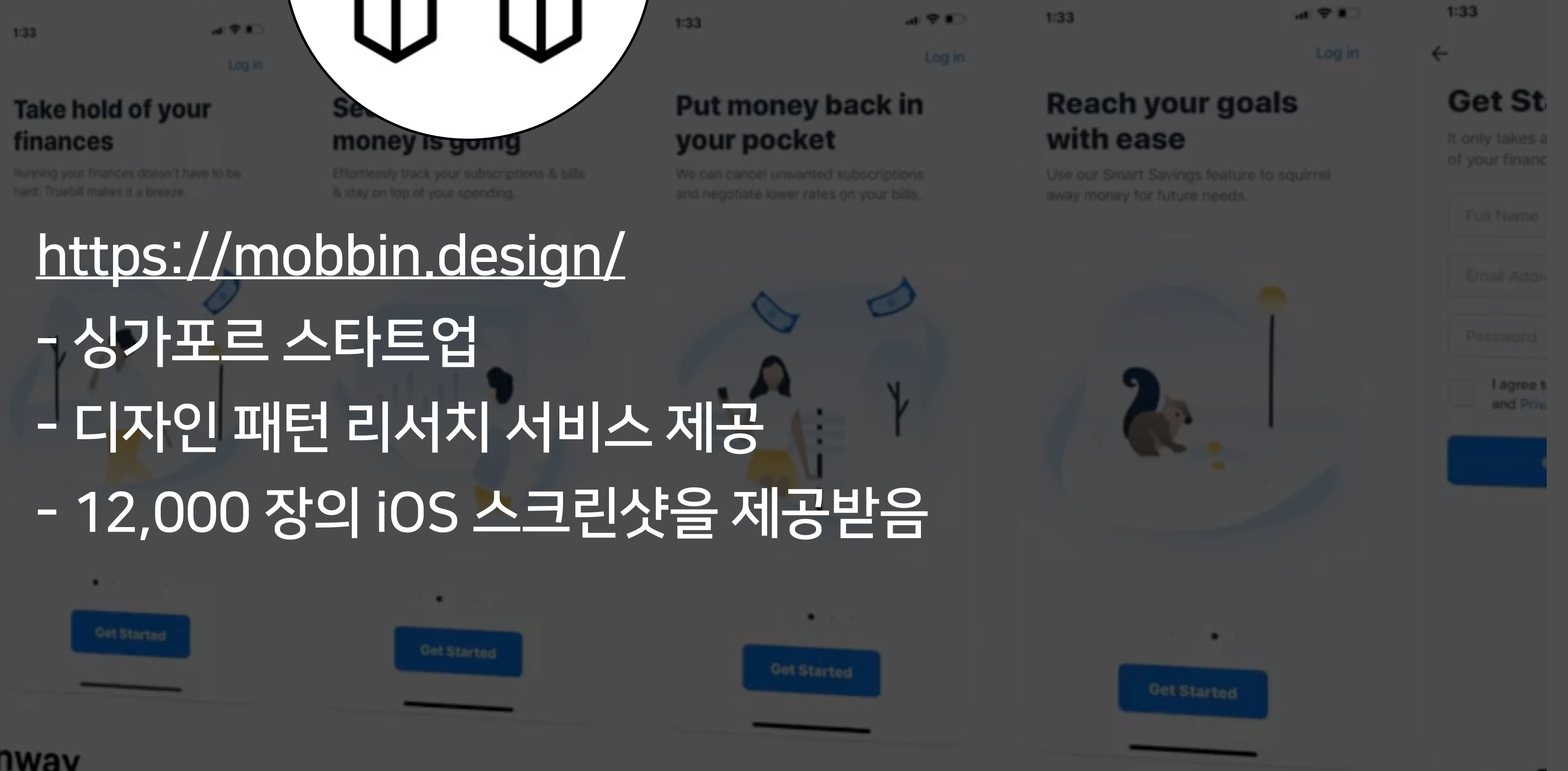
- 싱가포르 스타트업
- 디자인 패턴 리서치 서비스 제공
- 12,000 장의 iOS 스크린샷을 제공받음



**Rent the Runway**

Dress & Clothing Rentals

Try "Lyft"






<https://mobbin.design/>

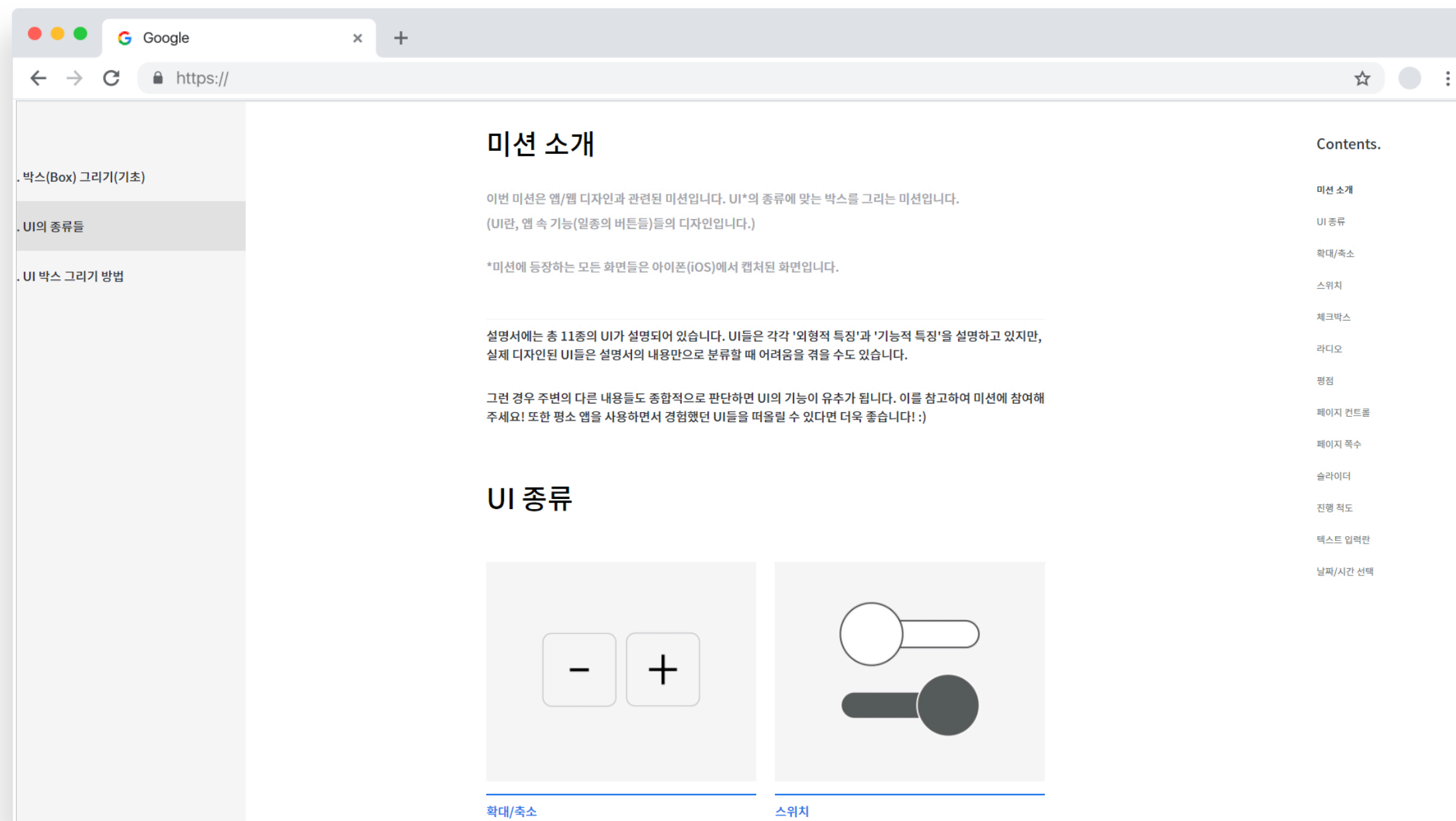
- 싱가포르 스타트업
- 앱 디자인 패턴 리서치 서비스 제공
- 12,000 장의 iOS 스크린샷을 제공받음

2020년 8월 Grida의 ui-crawler를 이용해서 크롤링 중..

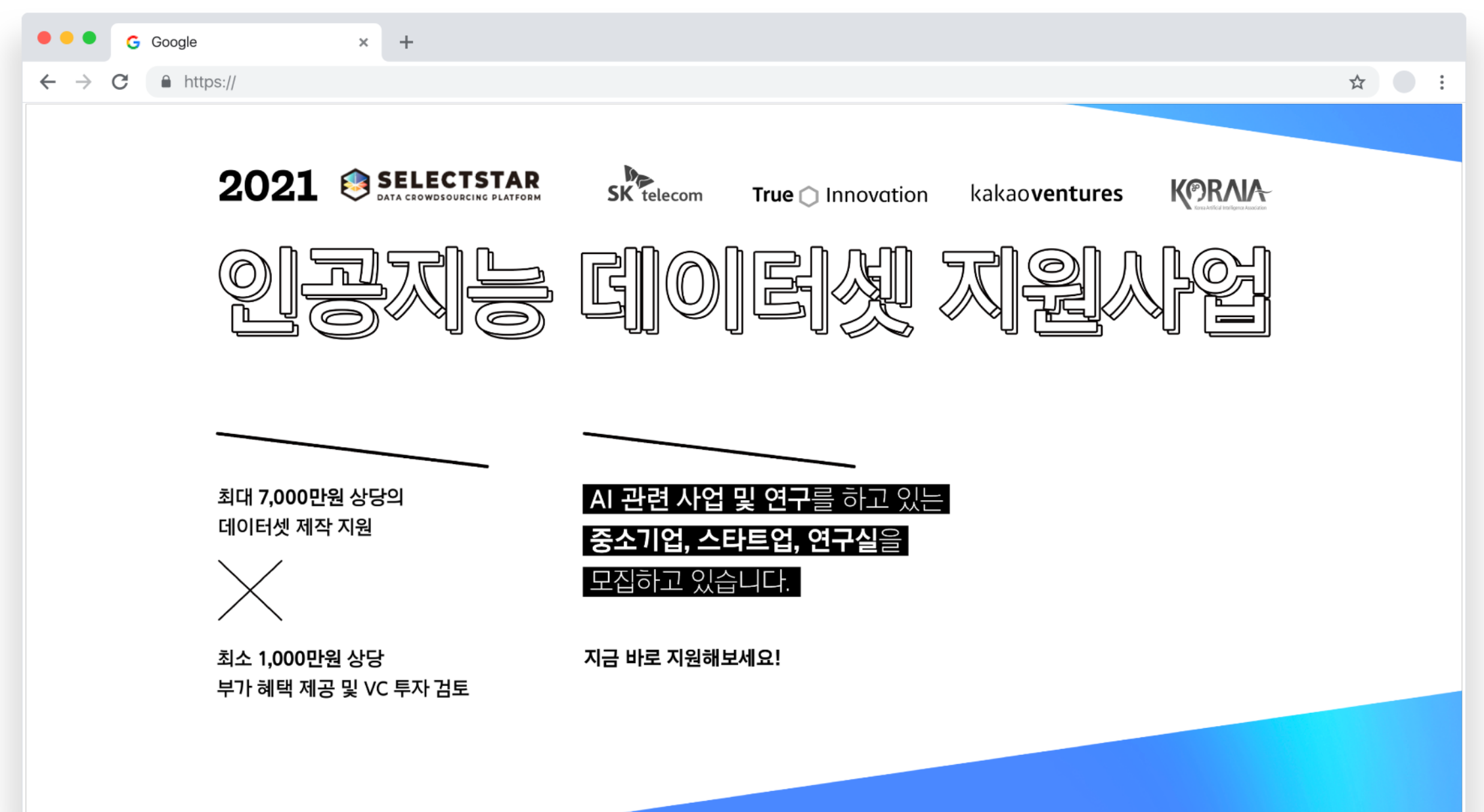
# UI컴포넌트 정의 및 데이터 라벨링

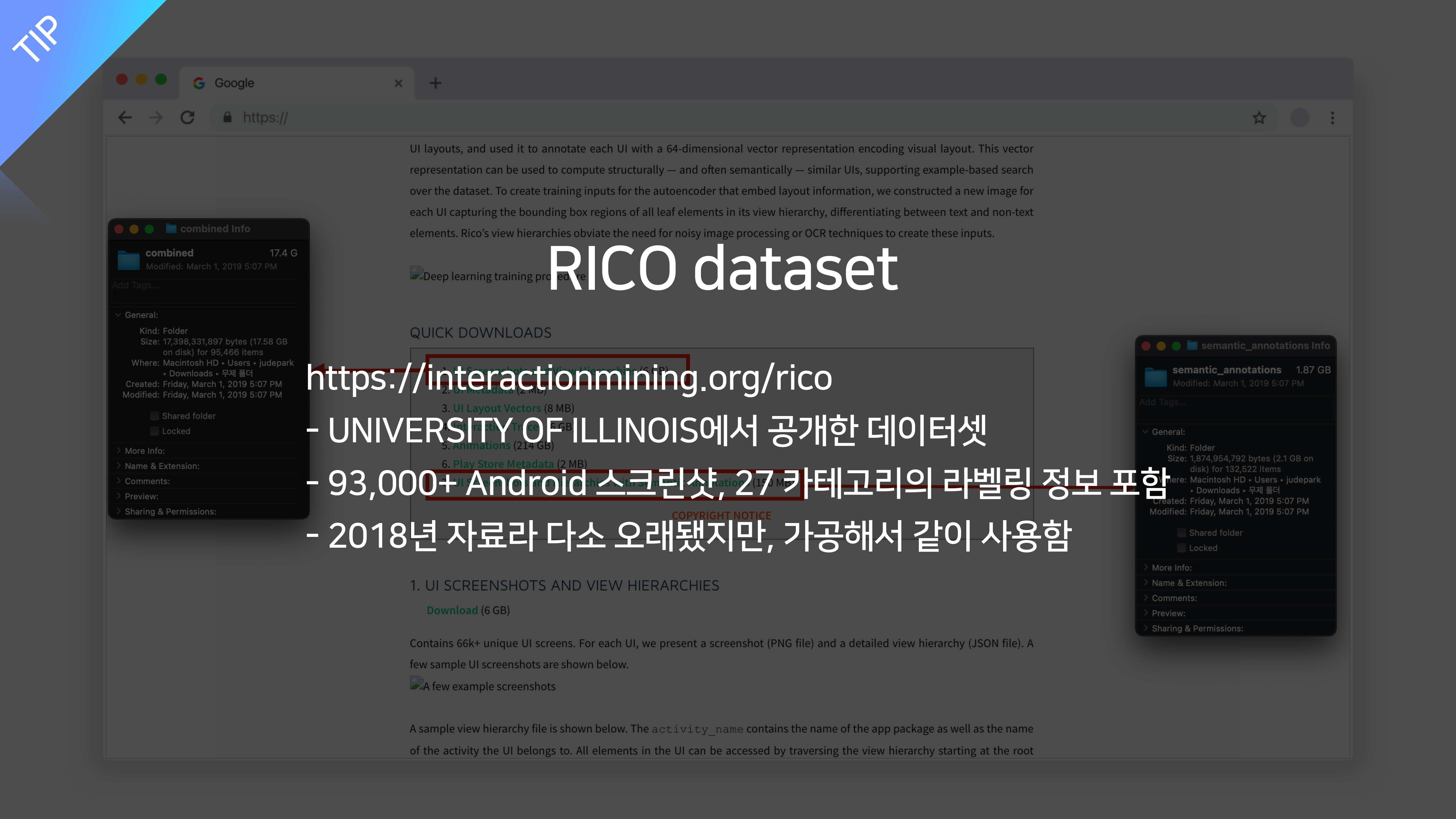
- 데이터 라벨링 기업,  **SELECTSTAR**의 지원사업에 선정됨
- 12,000장의 스크린샷을 14개 카테고리의 UI컴포넌트를 라벨링
- 라벨링 가이드 문서 및 관련 노하우 축적

## 라벨링 가이드 (캐시미션)



## 데이터셋 지원사업 ([dataset.or.kr](https://dataset.or.kr))





TIP

# RICO dataset

- <https://interactionmining.org/rico>
- UNIVERSITY OF ILLINOIS에서 공개한 데이터셋
- 93,000+ Android 스크린샷, 27 카테고리의 라벨링 정보 포함
- 2018년 자료라 다소 오래됐지만, 가공해서 같이 사용함

UI layouts, and used it to annotate each UI with a 64-dimensional vector representation encoding visual layout. This vector representation can be used to compute structurally — and often semantically — similar UIs, supporting example-based search over the dataset. To create training inputs for the autoencoder that embed layout information, we constructed a new image for each UI capturing the bounding box regions of all leaf elements in its view hierarchy, differentiating between text and non-text elements. Rico’s view hierarchies obviate the need for noisy image processing or OCR techniques to create these inputs.

## QUICK DOWNLOADS

1. UI Screenshots and View Hierarchies (6 GB)
2. UI Metadata (2 MB)
3. UI Layout Vectors (8 MB)
4. Interaction Traces (5 GB)
5. Animations (214 GB)
6. Play Store Metadata (2 MB)

### COPYRIGHT NOTICE

#### 1. UI SCREENSHOTS AND VIEW HIERARCHIES

[Download](#) (6 GB)

Contains 66k+ unique UI screens. For each UI, we present a screenshot (PNG file) and a detailed view hierarchy (JSON file). A few sample UI screenshots are shown below.

A few example screenshots

A sample view hierarchy file is shown below. The `activity_name` contains the name of the app package as well as the name of the activity the UI belongs to. All elements in the UI can be accessed by traversing the view hierarchy starting at the root

**combined Info**

**combined** 17.4 G  
Modified: March 1, 2019 5:07 PM

Add Tags...

General:

Kind: Folder  
Size: 17,398,331,897 bytes (17.58 GB on disk) for 95,466 items  
Where: Macintosh HD > Users > judepark > Downloads > 무제 폴더  
Created: Friday, March 1, 2019 5:07 PM  
Modified: Friday, March 1, 2019 5:07 PM

Shared folder  
 Locked

More Info:  
Name & Extension:  
Comments:  
Preview:  
Sharing & Permissions:

**semantic\_annotations Info**

**semantic\_annotations** 1.87 GB  
Modified: March 1, 2019 5:07 PM

Add Tags...

General:

Kind: Folder  
Size: 1,874,954,792 bytes (2.1 GB on disk) for 132,522 items  
Where: Macintosh HD > Users > judepark > Downloads > 무제 폴더  
Created: Friday, March 1, 2019 5:07 PM  
Modified: Friday, March 1, 2019 5:07 PM

Shared folder  
 Locked

More Info:  
Name & Extension:  
Comments:  
Preview:  
Sharing & Permissions:

TIP

UI layouts, and used it to annotate each UI with a 64-dimensional vector representation encoding visual layout. This vector representation can be used to compute structurally – and often semantically – similar UIs, supporting example-based search over the dataset. To create training inputs for the autoencoder that embed layout information, we constructed a new image for each UI capturing the bounding box regions of all leaf elements in its view hierarchy, differentiating between text and non-text elements. Rico’s view hierarchies obviate the need for noisy image processing or OCR techniques to create these inputs.

Deep learning training procedure

### QUICK DOWNLOADS

- 1. **UI Screenshots and View Hierarchies** (6 GB)
- 2. **UI Metadata** (2 MB)
- 3. **UI Layout Vectors** (8 MB)
- 4. **Interaction Traces** (6 GB)
- 5. **Animations** (214 GB)
- 6. **Play Store Metadata** (2 MB)
- 7. **UI Screenshots and Hierarchies with Semantic Annotations** (150 MB)

COPYRIGHT NOTICE

### 1. UI SCREENSHOTS AND VIEW HIERARCHIES

[Download](#) (6 GB)

Contains 66k+ unique UI screens. For each UI, we present a screenshot (PNG file) and a detailed view hierarchy (JSON file). A few sample UI screenshots are shown below.

A few example screenshots

A sample view hierarchy file is shown below. The `activity_name` contains the name of the app package as well as the name of the activity the UI belongs to. All elements in the UI can be accessed by traversing the view hierarchy starting at the root

**combined Info**

**combined** 17.4 G  
Modified: March 1, 2019 5:07 PM

Add Tags...

General:

- Kind: Folder
- Size: 17,398,331,897 bytes (17.58 GB on disk) for 95,466 items
- Where: Macintosh HD › Users › judepark › Downloads › 무제 폴더
- Created: Friday, March 1, 2019 5:07 PM
- Modified: Friday, March 1, 2019 5:07 PM

Shared folder  
Locked

More Info:  
Name & Extension:  
Comments:  
Preview:  
Sharing & Permissions:

**semantic\_annotations Info**

**semantic\_annotations** 1.87 GB  
Modified: March 1, 2019 5:07 PM

Add Tags...

General:

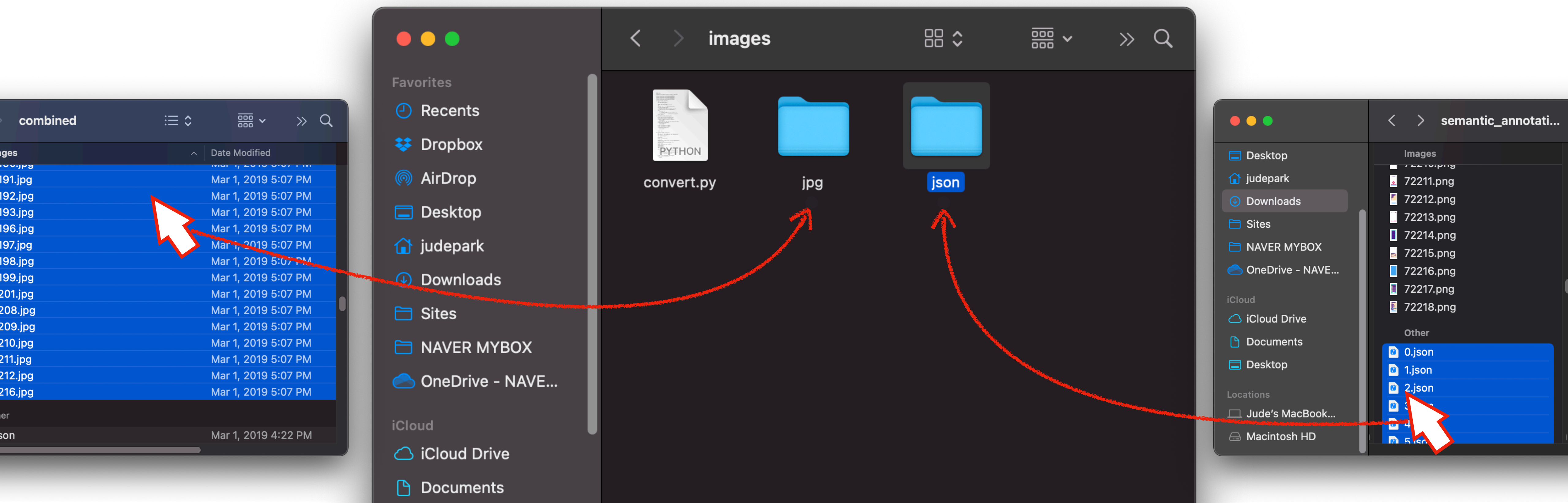
- Kind: Folder
- Size: 1,874,954,792 bytes (2.1 GB on disk) for 132,522 items
- Where: Macintosh HD › Users › judepark › Downloads › 무제 폴더
- Created: Friday, March 1, 2019 5:07 PM
- Modified: Friday, March 1, 2019 5:07 PM

Shared folder  
Locked

More Info:  
Name & Extension:  
Comments:  
Preview:  
Sharing & Permissions:

## RICO 데이터셋 가공하기

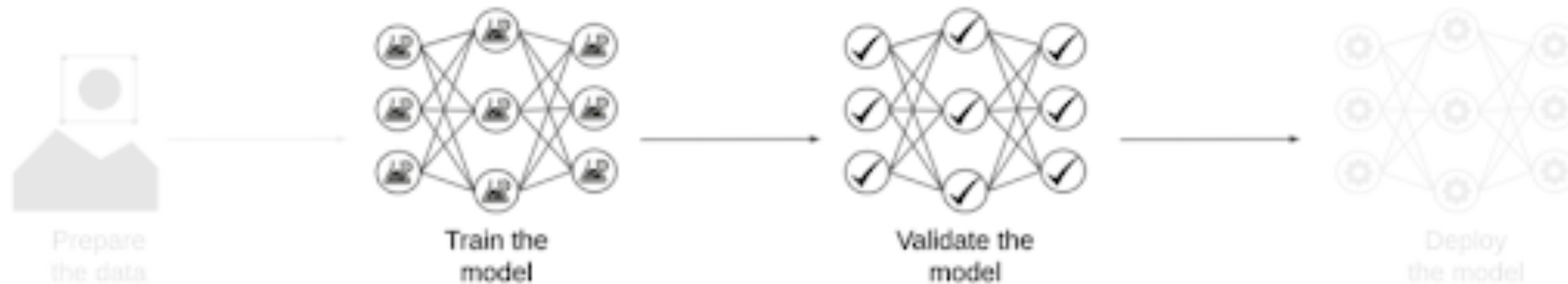
- 원본 이미지, Annotation json을 다운로드
- 이미지, annotation 파일만 각각의 폴더에 옮기기
- Python 스크립트(convert.py)를 만들어서 Pascal VOC 포맷의 XML로 변환



# Hands-on experience

## Training and validating the model

- 학습 방법 이슈
- backbone의 성능 및 선택 이슈
- 트레이닝 세팅 이슈





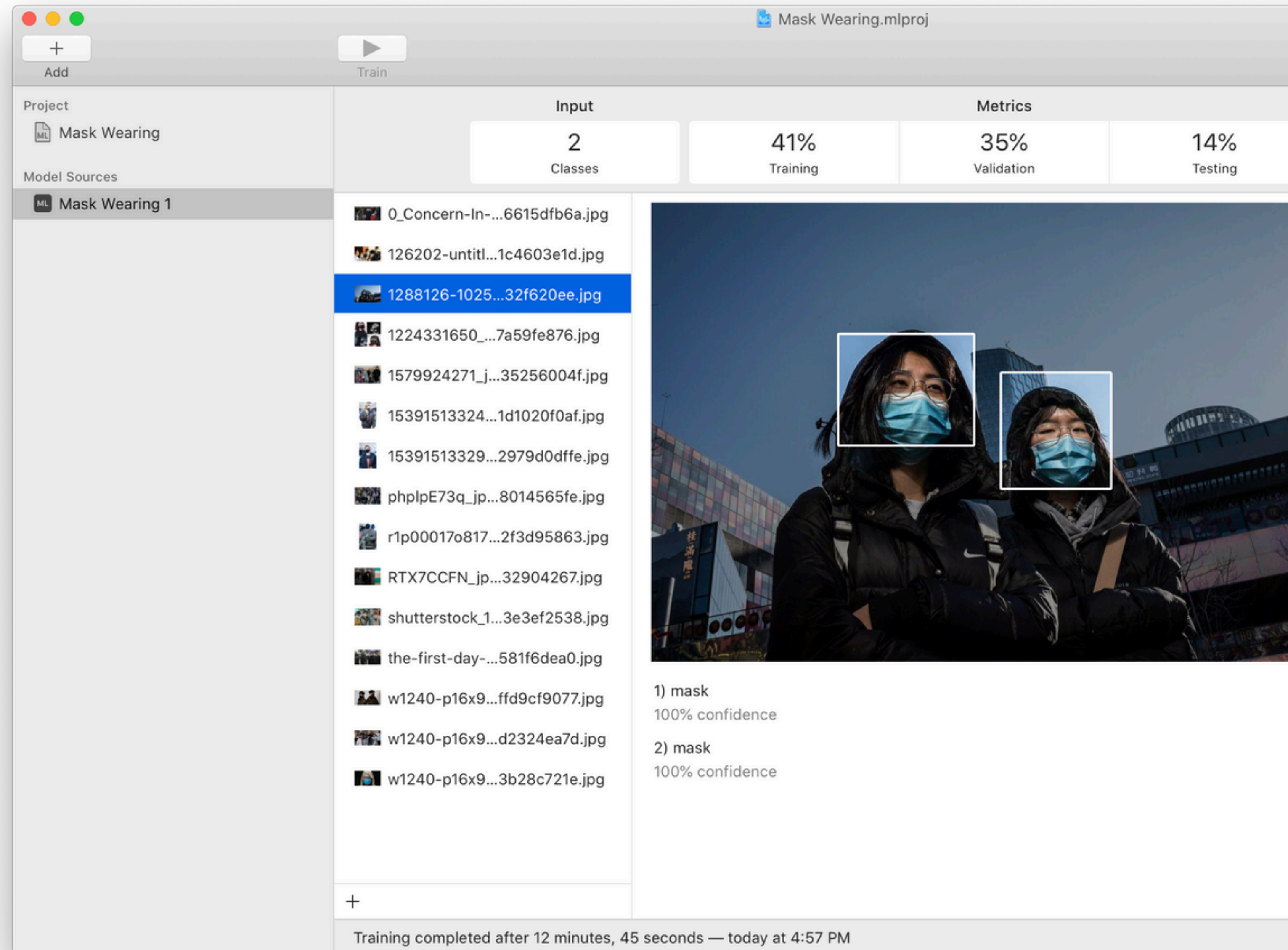
## 학습방법의 이슈

- □ TensorFlow
- □ CreateML
- □ AutoML (AWS, GCP..)
- □ Template match
- □ Transformer, YOLOX..

The screenshot shows the Roboflow ML interface for a project named "Mask Wearing.mproj". The interface includes a sidebar with "Project" and "Model Sources" sections, both containing "Mask Wearing" and "Mask Wearing 1". The main area displays training progress: 41% Training, 35% Validation, and 14% Testing. A list of image files is shown, with "1288126-1025...32f620ee.jpg" selected. To the right, a sample image shows two people wearing blue face masks, with white bounding boxes around their faces. Below the image, the model's predictions are listed: "1) mask 100% confidence" and "2) mask 100% confidence". At the bottom, a status bar indicates "Training completed after 12 minutes, 45 seconds — today at 4:57 PM".

## 학습방법의 이슈

- □ TensorFlow
- ~~☒ CreateML~~
  - ▶ YOLOv2 의존성
  - ▶ '기타용량' 문제
  - ▶ Apple 생태계 전용 CoreML
- □ AutoML (AWS, GCP..)
- □ Template match
- □ Transformer, YOLOX..
- 



## 학습방법의 이슈

- TensorFlow
- CreateML
- AutoML (AWS, GCP..)
  - ▶ 가격대 성능비 ...
  - ▶ 의존성 문제..
- Template match
- Transformer, YOLOX..

### Cloud AutoML Services Benchmarked on Pascal VOC 2012

	TRAINING TIME AND COST	MAX SIZE LIMIT	MODEL PERFORMANCE	INFERENCE TIME	INFERENCE COST
AWS Rekognition	26 hours → \$26	72 Training Hours	0.7212 mAP@0.5	2.0s	\$4 per hour
GCP AutoML Vision	20 hours → \$60	100MB Annotations	0.61 mAP@0.5	0.8s	\$1.82 per hour
Azure Custom Vision	1 hour → \$20	100,000 Training Images	0.64 mAP@0.5	0.5s	\$0.002 per image
YOLOv5x (Baseline)	12 hours → \$19 + Dev Time*	None	0.725 mAP@0.5	0.03s + Overhead†	\$1.58 per hour

\* US average computer vision engineer salary: \$147,000 (Indeed, 2020)

† Network latency

YOLOv5x Machine Type: Google Cloud n1-highmem-2 (P100 GPU)

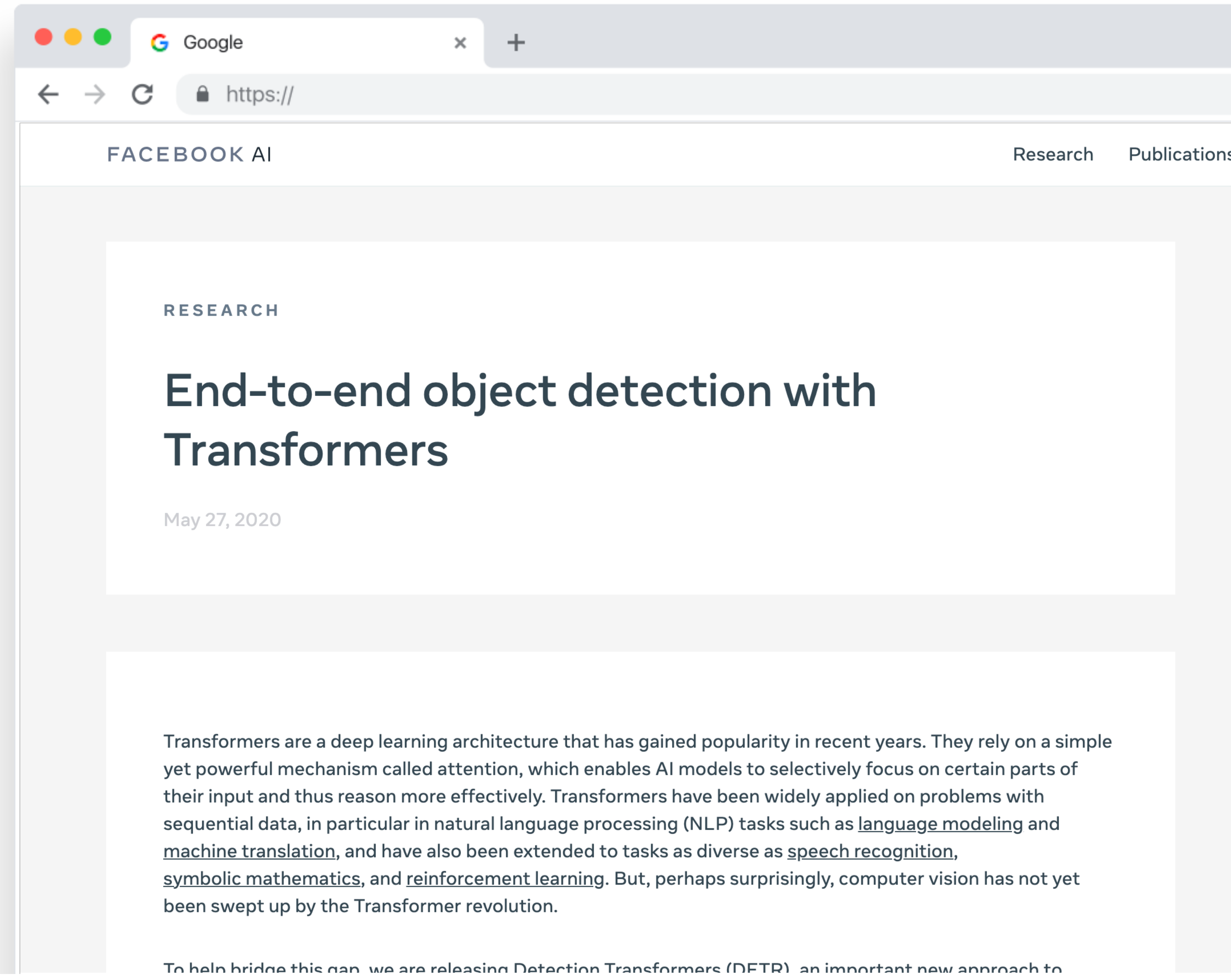
Details: [blog.roboflow.ai/automl-vs-rekognition-vs-custom-vision](https://blog.roboflow.ai/automl-vs-rekognition-vs-custom-vision)

**roboflow**

Source: Facebook research (2020)

## 학습방법의 이슈

- □ TensorFlow
- ~~⊗ CreateML~~
- ~~⊗ AutoML (AWS, GCP..)~~
- ~~⊗ Template match~~
- ~~⊗ Transformer, YOLOX..~~
  - ▶ 웹브라우저 환경
  - ▶ 디자인 시스템의 특성
  - ▶ 레퍼런스/샘플코드 용이성 등



## 학습방법의 이슈

### - TensorFlow JS

- ▶ Zero install
- ▶ Reach billions instantly
- ▶ No complex setup
- ▶ Highly shareable

-  CreateML

-  AutoML (AWS, GCP..)

-  Template match

-  Transformer, YOLOX..



## TensorFlow용 TFrecord로 변환

- 앞선 노력의 결과 이미지 파일과 xml을 확보함
- 확보한 파일들을 묶어서 .tfrecord 파일로 변환해야함
- 이 과정에서 데이터셋에 문제가 없는지 여러번 검증 필요

## TensorFlow용 TFrecord로 변환

- 앞선 노력의 결과 이미지 파일과 xml을 확보함
- 확보한 파일들을 묶어서 .tfrecord 파일로 변환해야함
- 이 과정에서 데이터셋에 문제가 없는지 여러번 검증 필요
  - ▶ bounding box(bndbox)가 음수인 경우
  - ▶ xmax 또는 ymax 값이 전체 width, height 값보다 큰 경우
  - ▶ xmin 값이 xmax 보다 크거나 ymin 값은 ymax 보다 큰 경우

```
def isvalidbndbox(xmin,xmax,ymin,ymax):  
  
    ## 다중 if를 추가해서 bnd 박스를 검증해야 합니다.  
    # bounding box(bndbox)가 음수이면 안됩니다.  
    if(xmin<0 or xmax<0 or ymin<0 or ymax<0):  
        print("음수")  
        return False  
    # bndbox의 xmax, ymax 값은 xml의 width,  
    height를 넘으면 안됩니다.  
    if(xmax>width or ymax>height):  
        print("최대값 초과")  
        return False  
  
    # xmin 값은 xmax 보다 클 수 없습니다.  
    # ymin 값은 ymax 보다 클 수 없습니다.  
    if(xmin>xmax or ymin>ymax):  
        print("최소값 초과")  
        return False  
  
    return True
```

## TensorFlow용 TFrecord로 변환

- 앞선 노력의 결과 이미지 파일과 xml을 확보함
- 확보한 파일들을 묶어서 .tfrecord 파일로 변환해야함
- 이 과정에서 데이터셋에 문제가 없는지 여러번 검증 필요
  - ▶ bounding box(bndbox)가 음수인 경우
  - ▶ xmax 또는 ymax 값이 전체 width, height 값보다 큰 경우
  - ▶ xmin 값이 xmax 보다 크거나 ymin 값은 ymax 보다 큰 경우
  - ▶ (선택) 이미지 사이즈가 너무 큰 경우

```
// Combined 폴더의 이미지를 1/4로 줄여서 jpg 폴더에 저장
from PIL import Image
import os

raw_path = '/content/images/combined/' # 원본 이미지 경로
data_path = '/content/images/jpg/' # 저장할 이미지 경로

# resize 시작 -----
# 저장할 경로 없으면 생성
if not os.path.exists(data_path):
    os.mkdir(data_path)

#원본 이미지 경로의 모든 이미지 list 지정
data_list = os.listdir(raw_path)
print(len(data_list))

# 모든 이미지 resize 후 저장하기
for name in data_list:
    # 이미지 열기
    im = Image.open(raw_path + name)

    # 이미지 resize
    im = im.resize((360, 640))

    # 이미지 JPG로 저장
    im.save(data_path + name)
    print('end ::: ' + data_path + name)
```





앞서 만든 TFRecord 파일  
들(train, test) 직접 업로드



또는 구글 드라이브를 마운  
트해서 사용합니다.

The screenshot shows a Google Colab notebook interface. On the left, a 'Files' panel displays a directory structure with folders named 'models' and 'sample\_data'. Two circular callouts highlight the 'upload' and 'mount' icons in the top right of the file panel. On the right, a code cell contains Python code for mounting Google Drive:

```
[ ] #Downloading data by mounting Google drive

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

Below the code cell, the output shows 'Mounted at /content/drive'. The notebook also displays a large block of red text, which appears to be a warning or error message related to image annotations.

### Prepare Tensorflow 2 Object Detection Training Data

Though there were good services like Roboflow which creates TFRecord and label\_map files auto burden on my budget because.. I will use a large image of a large image of more than 78,000 this referring to the Tensorflow Guide and uploaded this Notebook directly.

If you wanna use create a dataset and generate TFRecords via Roboflow, follow [this step-by-step](#)

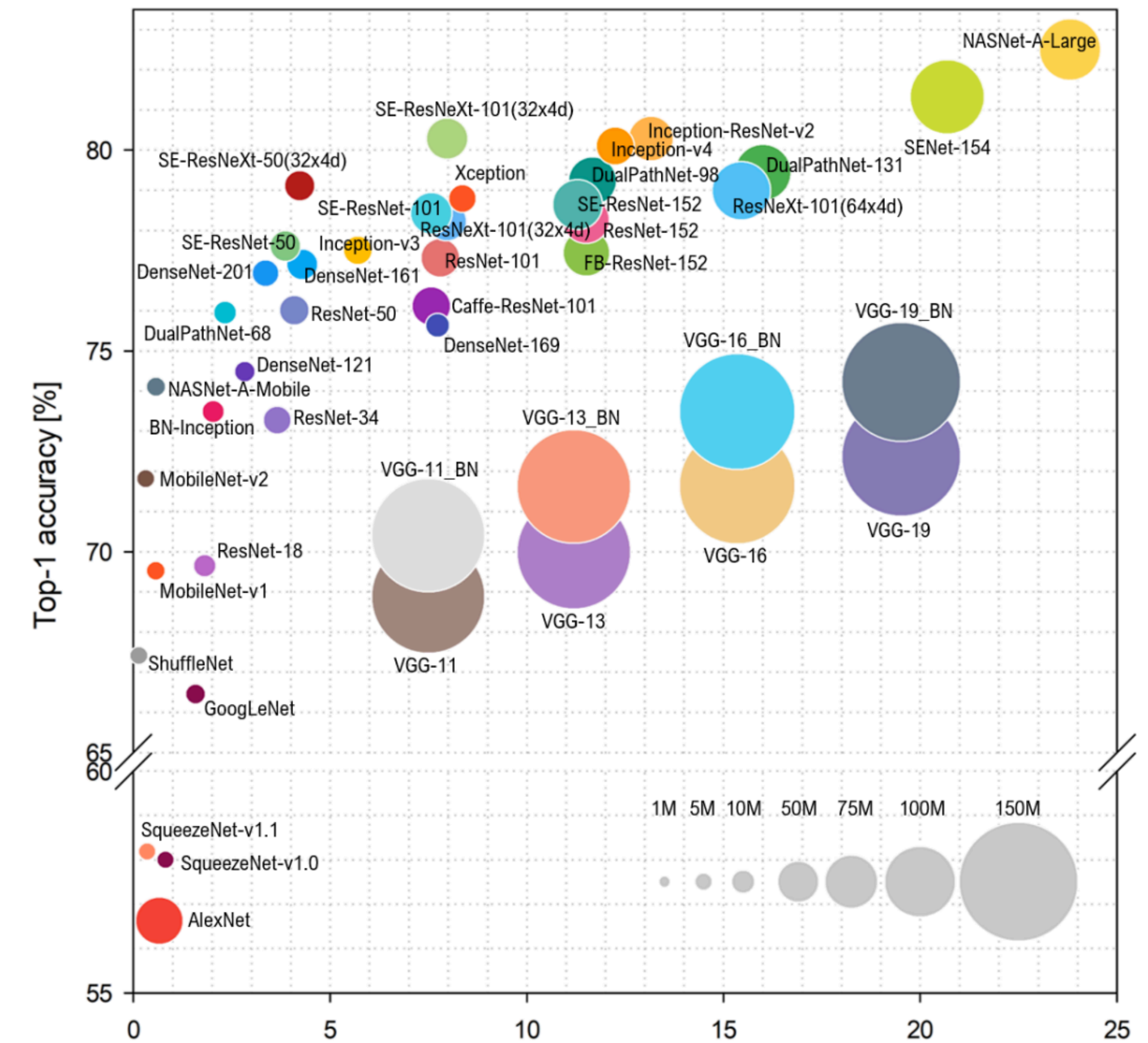
[ ] #Downloading data by mounting Google drive

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

Mounted at /content/drive

## backbone의 성능 및 선택 이슈

- ImageNet 성능 기준으로 좋은 딥러닝 모델을 선택
- 웹브라우저에서 동작해야 하므로 가벼운 모델을 선택
- Finetune 성능이 차이나는 경우도 있으므로 실험 필요
- ResNet50, efficientdet, MobileNetV2로 실험



ImageNet 성능 비교 그래프



(옵션)  
모델을 변경할 수 있습니다.



(옵션)  
트레이닝을 설정할 수 있습니다

Configure Custom TensorFlow2 Object Detection Training Configuration

In this section you can specify any model in the [TF2 OD model zoo](#) and set up your training configuration.

```

##change chosen model to deploy different models available in the TF2 object detection zoo
MODELS_CONFIG = {
  'efficientdet-d0': {
    'model_name': 'efficientdet_d0_coco17_tpu-32',
    'base_pipeline_file': 'ssd_efficientdet_d0_512x512_coco17_tpu-8.config',
    'pretrained_checkpoint': 'efficientdet_d0_coco17_tpu-32.tar.gz',
    'batch_size': 16
  },
  'efficientdet-d1': {
    'model_name': 'efficientdet_d1_coco17_tpu-32',
    'base_pipeline_file': 'ssd_efficientdet_d1_640x640_coco17_tpu-8.config',
    'pretrained_checkpoint': 'efficientdet_d1_coco17_tpu-32.tar.gz',
    'batch_size': 16
  },
  'efficientdet-d2': {
    'model_name': 'efficientdet_d2_coco17_tpu-32',
    'base_pipeline_file': 'ssd_efficientdet_d2_768x768_coco17_tpu-8.config',
    'pretrained_checkpoint': 'efficientdet_d2_coco17_tpu-32.tar.gz',
    'batch_size': 16
  },
  'efficientdet-d3': {
    'model_name': 'efficientdet_d3_coco17_tpu-32',
    'base_pipeline_file': 'ssd_efficientdet_d3_896x896_coco17_tpu-32.config',
    'pretrained_checkpoint': 'efficientdet_d3_coco17_tpu-32.tar.gz',
    'batch_size': 16
  },
}

#in this tutorial we implement the lightweight, smallest state of the
#if you want to scale up tot larger efficientdet models you will likel
chosen_model = 'ssd_mobilenet_v2'

num_steps = 7500 #The more steps, the longer the training. Increase if
num_eval_steps = 1000 #Perform evaluation after so many steps

```

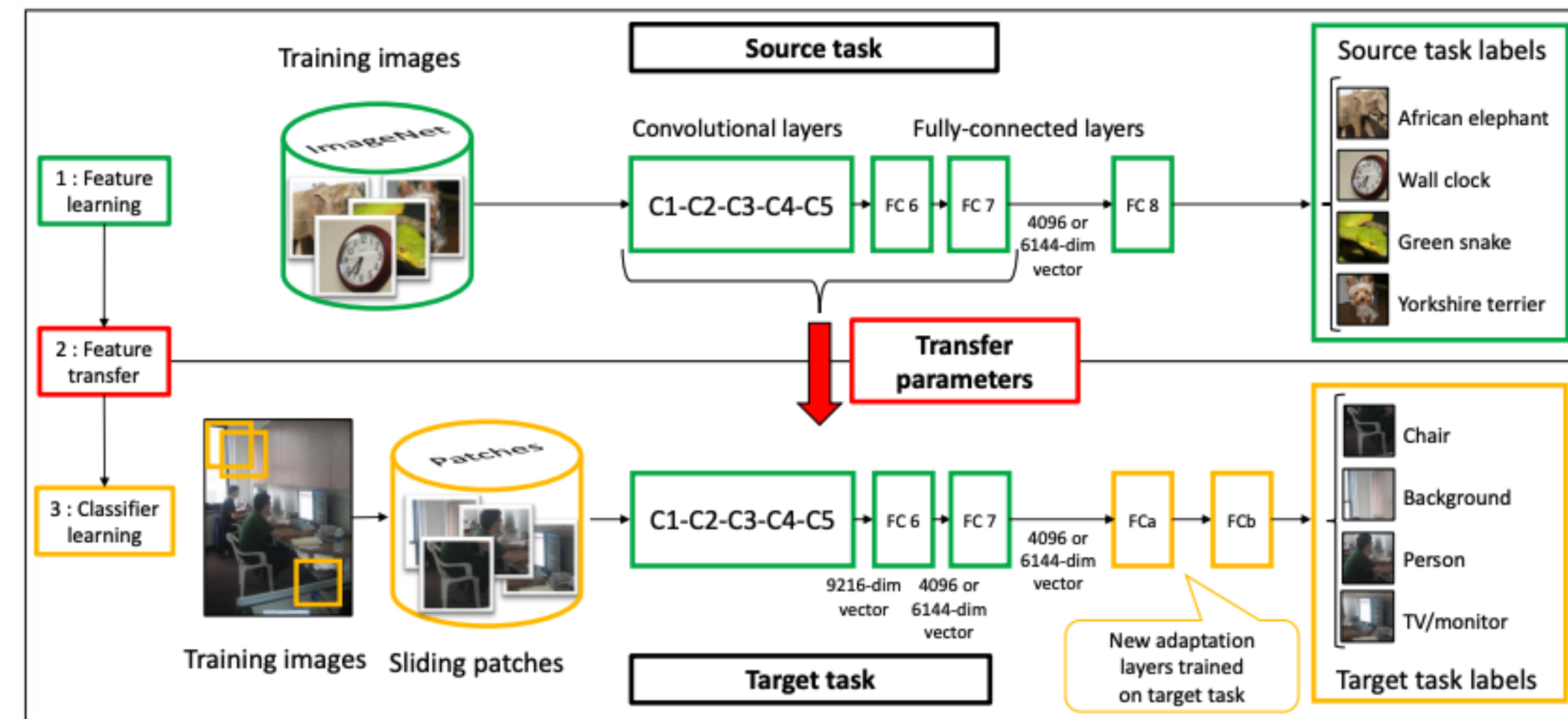
num\_steps = 7500 #The more steps, the longer the training. Increase if your loss function is still decreasing and validation metric  
num\_eval\_steps = 1000 #Perform evaluation after so many steps

## backbone의 성능 및 선택 이슈

- ImageNet 성능 기준으로 좋은 딥러닝 모델을 선택
- 웹브라우저에서 동작해야 하므로 **가벼운 모델**을 선택
- Finetune 성능이 차이나는 경우도 있으므로 실험 필요
- ResNet50, efficientdet, **MobileNetV2**로 실험

## 학습방법 이슈

- Transfer Learning
- Fine Tuning

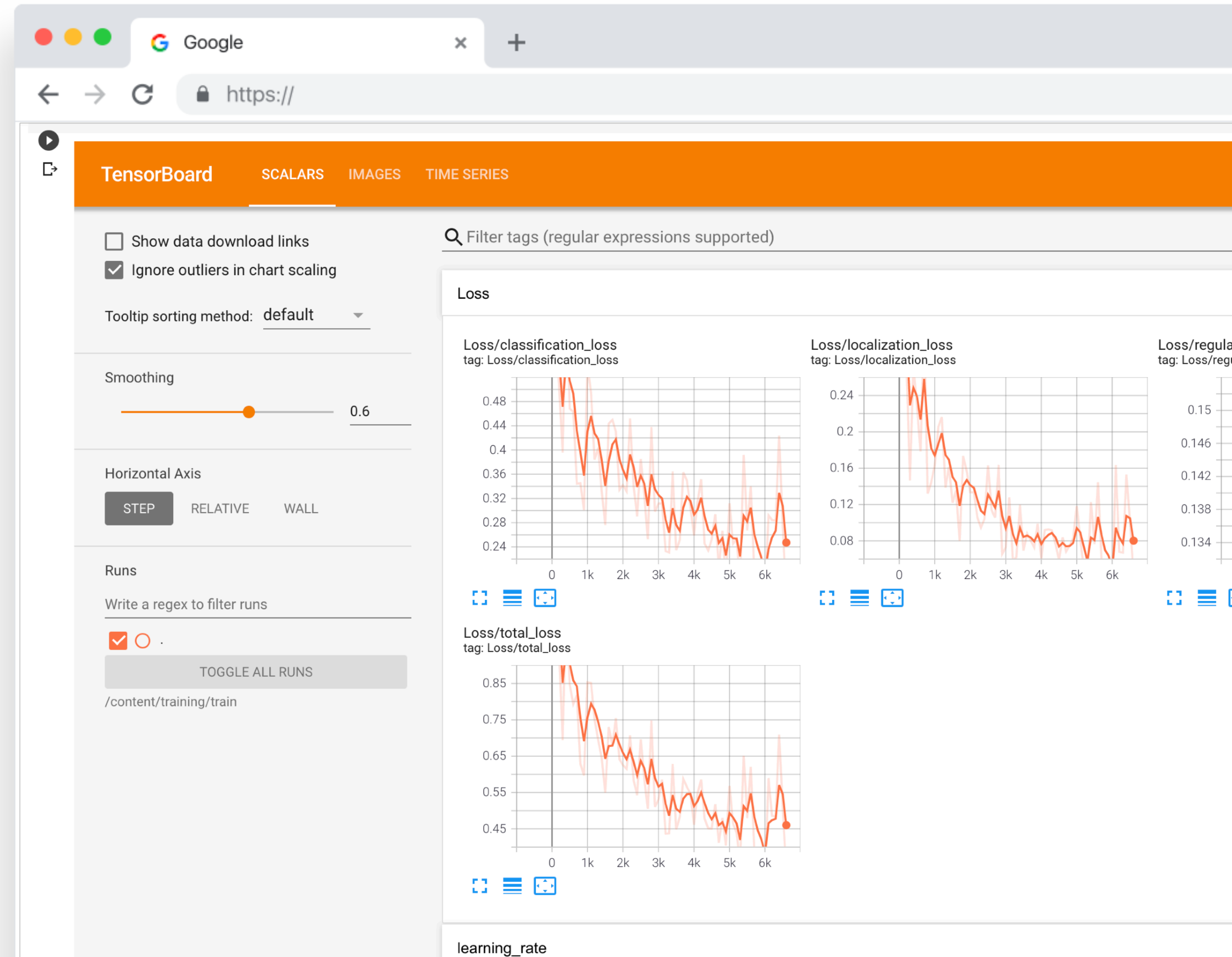


Transferring parameters of a CNN

## 트레이닝 이슈

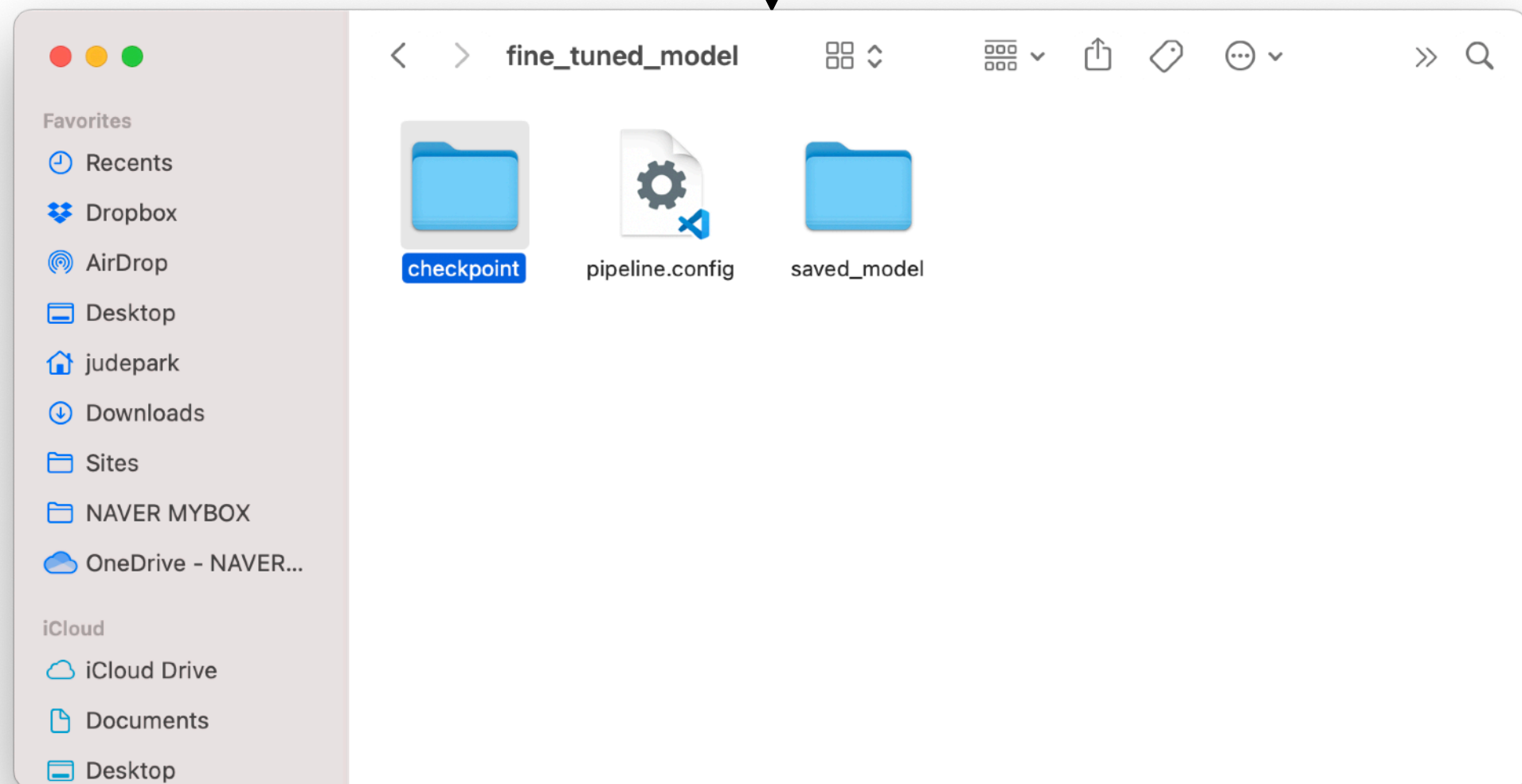
- 트레이닝은 평균 7시간씩 소요
- Colab의 Free GPU, 12시간 제한에 유의할 것
- Training에 실패하면 제일 먼저 데이터셋을 의심할 것
- Fine Tuning은 변경하며 실험

optimizer 종류, learning rate 값, weight decay 값, epoch 수, learning rate scheduling 방법, data augmentation 강도, 기타 regularizer 유/무 등





학습 완료 후, SavedModel을  
추출해서 다운로드합니다



Google

https://

Mobbin-TensorFlow2-Object-Detection.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
assets/ saved_model.pb variables/
```

```
[ ]
```

```
!zip -r /content/drive/MyDrive/2020-mobbin/mobbin_model.zip fine_tuned_model/
```

```
adding: fine_tuned_model/ (stored 0%)
adding: fine_tuned_model/pipeline.config (deflated 69%)
adding: fine_tuned_model/checkpoint/ (stored 0%)
adding: fine_tuned_model/checkpoint/ckpt-0.data-00000-of-00001 (deflated 8%)
adding: fine_tuned_model/checkpoint/ckpt-0.index (deflated 80%)
adding: fine_tuned_model/checkpoint/checkpoint (deflated 41%)
adding: fine_tuned_model/saved_model/ (stored 0%)
adding: fine_tuned_model/saved_model/assets/ (stored 0%)
adding: fine_tuned_model/saved_model/saved_model.pb (deflated 93%)
adding: fine_tuned_model/saved_model/variables/ (stored 0%)
adding: fine_tuned_model/saved_model/variables/variables.data-00000-of-00001 (deflated 8%)
adding: fine_tuned_model/saved_model/variables/variables.index (deflated 77%)
```

```
from google.colab import files
files.download("/content/drive/MyDrive/2020-mobbin/mobbin_model.zip")
```

### Run Inference on Test Images with Custom TensorFlow2 Object Detector

```
[ ] #downloading test images from Roboflow
#export dataset above with format COCO JSON
#or import your test images via other means.
```

```
%mkdir /content/samples/
%cd /content/samples/
!curl -L "https://github.com/dusskapark/data-labeling-helper/raw/master/samples.zip" > samples.zip; unzip samples.zip; rm samples.zip
```

```
mkdir: cannot create directory '/content/samples/': File exists
/content/samples
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 150 100 150 0 0 592 0 --:--:-- --:--:-- --:--:-- 592
100 243k 100 243k 0 0 514k 0 --:--:-- --:--:-- --:--:-- 9136k
Archive: samples.zip
```

# Hands-on experience

## Deploy the model

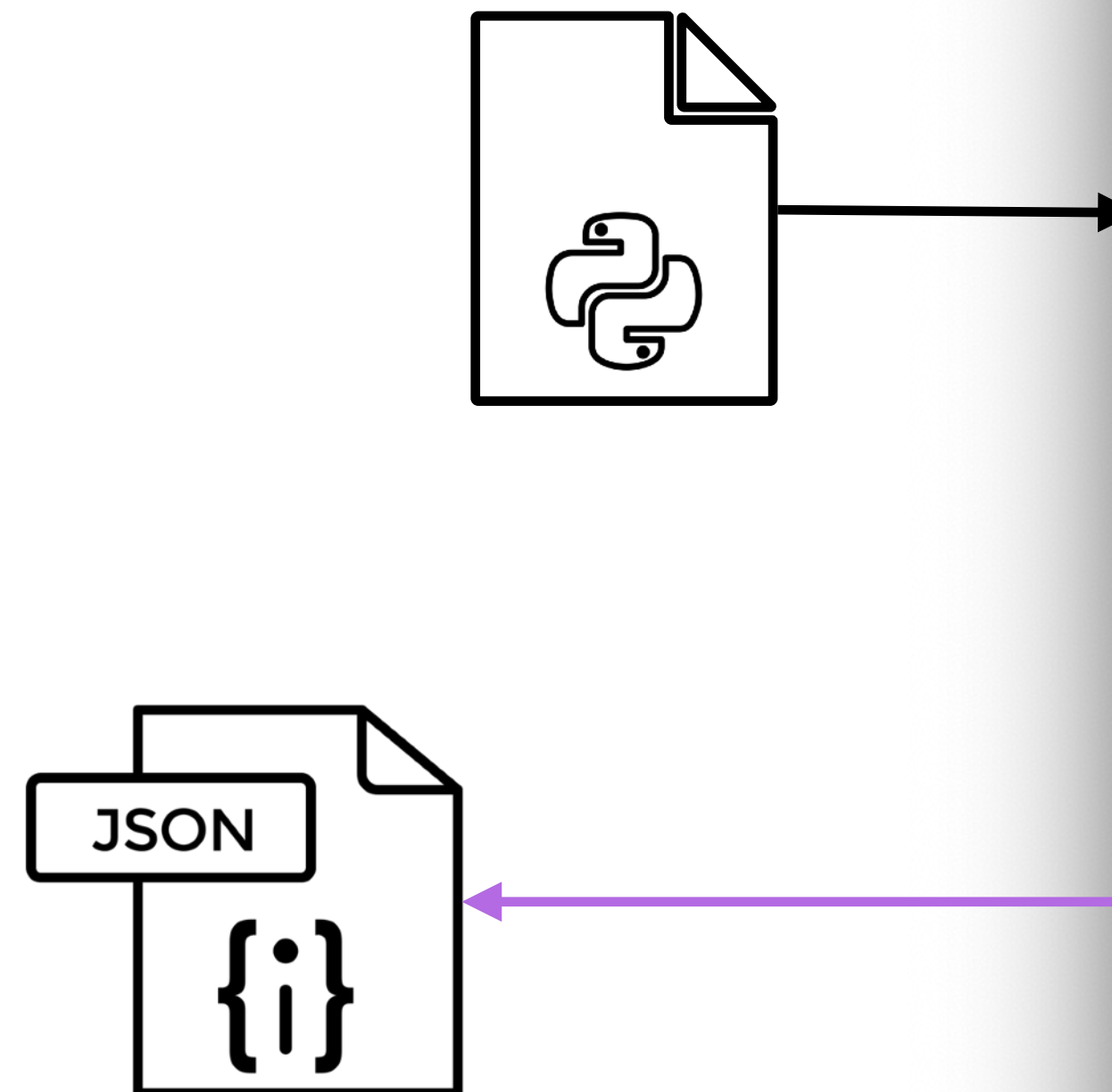
- TF2 모델 추출 및 TFJS 모델로 변환
- CodeSandbox에서 테스트
- React 앱에 통합 및 배포



## TensorFlow.js Converter CLI 화면

### TensorFlow JS 모델로 변환

- 앞서 만든 TF2 모델을 TFJS용으로 변환
- **model.json**과 weight 파일 여러 개가 만들어짐
- 파일들을 CDN에 업로드하면 준비 완료



```

fine_tuned_model — tensorflowjs_wizard — tensorflowjs_wizard — tens
[→ fine_tuned_model tensorflowjs_wizard
Welcome to TensorFlow.js Converter.
? Please provide the path of model file or the directory that contains model files. saved_model
If you are converting TFHub module please provide the URL.
? What is your input model format? (auto-detected format is marked with *) Tensorflow
? What is tags for the saved model? serve
? What is signature name of the model? signature name: serving_default
? Do you want to compress the model? (this will decrease the model precision.) No
[? Please enter shard size (in bytes) of the weight files? 4194304
? Do you want to skip op validation?
This will allow conversion of unsupported ops,
you can implement them as custom ops in tfjs-converter. No
? Do you want to strip debug ops?
This will improve model execution performance. Yes
? Do you want to enable Control Flow V2 ops?
This will improve branch and loop execution performance. Yes
? Do you want to provide metadata?
Provide your own metadata in the form:
metadata_key:path/metadata.json
Separate multiple metadata by comma.
[? Which directory do you want to save the converted model in? RICO
converter command generated:
tensorflowjs_converter --control_flow_v2=True --input_format=tf_saved_model --metada
ure_name=serving_default --strip_debug_ops=True --weight_shard_size_bytes=4194304 sa

2021-08-14 02:14:45.150985: I tensorflow/core/platform/cpu_feature_guard.cc:142] Thi
oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions i
VX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate compiler

```



## CodeSandbox에서 테스트 (1/3)

- 이미지 검출기 프로젝트를 Folk
- 모델을 직접 만든 모델로 변경
- **loadGraphModel**로 모델을 로드 (비동기)

The screenshot shows a CodeSandbox environment with the following components:

- Browser:** Displays a web application at `https://mos1p.csb.app/`. The page shows an image of a woman's face with several bounding boxes and labels. The labels include:
  - Web View 40.27%
  - Background Image 36.72%
  - Image 42.71%
  - Icon 32.53%
  - Icon 31.73%
  - Icon 31.73%
- Code Editor:** Shows JavaScript code for loading and using a TensorFlow.js model. A snippet is highlighted:
 

```
const model = await tf.loadGraphModel(
  "https://raw.githubusercontent.com/dussskapark/zeplin-ml/");
```
- File Explorer:** Lists files in the `src` directory: `28362.jpg`, `index.js`, `screenshot.png`, `styles.css`, `test.jpg`, `toby-office.jpg`, `.babelrc`, `index.html`, and `package.json`.
- Dependencies:** Lists installed packages: `@tensorflow/tfjs` (3.7.0), `@tensorflow/tfjs-backen...` (3.7.0), and `@tensorflow/tfjs-conver...` (3.7.0).
- Console:** Shows the output of the application, including the detected objects and their scores:
 

```
detected: {class: 2, label: "Image", score: "0.3644"}
detected: {class: 3, label: "Icon", score: "0.3321"}
detected: {class: 3, label: "Icon", score: "0.3253"}
detected: {class: 3, label: "Icon", score: "0.3211"}
detected: {class: 3, label: "Icon", score: "0.3199"}
detected: {class: 3, label: "Icon", score: "0.3173"}
```

## CodeSandbox에서 테스트 (2/3)

- 이미지를 텐서(Tensor)로 변환해서,
- `predict` 함수로 이미지 안의 객체를 탐색함
- 비동기 처리 필수

```

const expandedimg = loadImage(image);
const predictions = await predict(expandedimg, model);
const detections = renderPredictions(
  predictions,
  image.width,
  image.height
);
console.log("detected: ", detections);

```

```

162 c.width = image.width;
163 c.height = image.height;
164 const context = c.getContext("2d");
165 context.drawImage(image, 0, 0);
166
167 // Font options.
168 const font = "16px sans-serif";
169 context.font = font;
170 context.textBaseline = "top";
171
172 const model = await tf.loadGraphModel(
173   "https://raw.githubusercontent.com/dussskapark/zeplin-ml/

```

```

167 const y = item["bbox"][1];
168 const width = item["bbox"][2];
169 const height = item["bbox"][3];
170
171 // Draw the bounding box.
172 context.strokeStyle = "#00FFFF";
173 context.lineWidth = 4;
174 context.strokeRect(x, y, width, height);
175
176 // Draw the label background.
177 context.fillStyle = "#00FFFF";
178 const textWidth = context.measureText(
179   item["label"] + " " + (100 * item["score"]).toFixed(2)
180 ).width;

```

```

detected: {class: 2, label: "Image", score: "0.3644"}
detected: {class: 3, label: "Icon", score: "0.3321"}
detected: {class: 3, label: "Icon", score: "0.3253"}
detected: {class: 3, label: "Icon", score: "0.3211"}
detected: {class: 3, label: "Icon", score: "0.3199"}
detected: {class: 3, label: "Icon", score: "0.3173"}

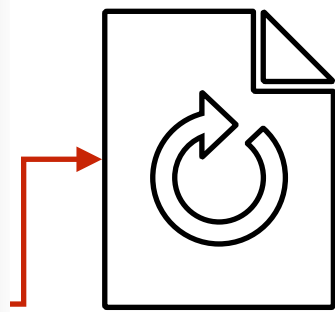
```

## CodeSandbox에서 테스트 (3/3)

- predict 응답 결과는 Tensor 형태
- [saved\\_model\\_cli](#) 로 해석 (링크 참고)
- dataSync, arraySync 로 변환

```

▼ (8) [Tensor, Tensor, Tensor, Tensor, Tensor, Tensor, Tensor, Tensor]
▶ 0: Tensor
▶ 1: Tensor
▶ 2: Tensor
▶ 3: Tensor
▶ 4: Tensor
▶ 5: Tensor
▶ 6: Tensor
▼ 7: Tensor
  kept: false
  isDisposedInternal: false
  ▶ shape: Array(1)
  dtype: "float32"
  size: 1
  strides: Array(0)
  ▶ dataId: Object
  id: 1969
  rankType: "1"
  scopeId: 1726
  ▶ abs: f () {}
  ▶ acos: f () {}
  ▶ acosh: f () {}
  ▶ add: f () {}
  ▶ all: f () {}
  ▶ any: f () {}
  ▶ argMax: f () {}
  
```



The screenshot shows a CodeSandbox environment with a browser window displaying object detection results. The browser shows a web view of an image with bounding boxes and labels. The code editor shows the following JavaScript code:

```

162 c.width = image.width;
163 c.height = image.height;
164 const context = c.getContext("2d");
165 context.drawImage(image, 0, 0);
166
167 // Font options.
168 const font = "16px sans-serif";
169 context.font = font;
170 context.textBaseline = "top";
171
172 const model = await tf.loadGraphModel(
173   "https://raw.githubusercontent.com/dussskapark/zeplin-ml/
174 );
175
176 const expandedimg = loadImage(image);
177 const predictions = await predict(expandedimg, model);
178 const detections = renderPredictions(
179   predictions,
180   image.width,
181   image.height
182 );
183 console.log("detected: ", detections);
184
185 detections.forEach((item) => {
186   const x = item["bbox"][0];
187   const y = item["bbox"][1];
188   const width = item["bbox"][2];
189   const height = item["bbox"][3];
190
191   // Draw the bounding box.
192   context.strokeStyle = "#00FFFF";
193   context.lineWidth = 4;
194   context.strokeRect(x, y, width, height);
195
196   // Draw the label background.
197   context.fillStyle = "#00FFFF";
198   const textWidth = context.measureText(
199     item["label"] + " " + (100 * item["sc
200   ).width;
  
```

The browser window shows the following object detection results:

- Web View 40.27%
- Background Image 36.72%
- Image 42.71%
- Icon 32.53%
- Icon 31.73%
- Icon 31.73%
- Icon 31.73%

The console shows the following output:

```

detected: ▶ {class: 2, label: "Image", score: "0.3644", bbox: ...}
detected: ▶ {class: 3, label: "Icon", score: "0.3321", bbox: ...}
detected: ▶ {class: 3, label: "Icon", score: "0.3253", bbox: ...}
detected: ▶ {class: 3, label: "Icon", score: "0.3211", bbox: ...}
detected: ▶ {class: 3, label: "Icon", score: "0.3199", bbox: ...}
detected: ▶ {class: 3, label: "Icon", score: "0.3173", bbox: ...}
  
```

## CodeSandbox에서 테스트 (3/3)

- predict 응답 결과는 Tensor 형태
- `saved_model_cli` 로 해석 (링크 참고)
- `dataSync`, `arraySync` 로 변환
- 캔버스에 박스 그리기

The screenshot shows a CodeSandbox environment for a project named 'mobbin-object-detection-with-tensorflow-js'. The browser window displays a video player with object detection results overlaid. The results include bounding boxes around the video frame with labels and scores:

- Background Image 36.72%
- Image 42.71%
- Icon 32.53%
- Icon 31.73%
- Icon 31.73%
- Icon 31.73%

The code editor shows the following JavaScript code:

```

162 c.width = image.width;
163 c.height = image.height;
164 const context = c.getContext("2d");
165 context.drawImage(image, 0, 0);
166
167 // Font options.
168 const font = "16px sans-serif";
169 context.font = font;
170 context.textBaseline = "top";
171
172 const model = await tf.loadGraphModel(
173   "https://raw.githubusercontent.com/dussskapark/zeplin-ml/
174 );
175
176 const expandedimg = loadImage(image);
177 const predictions = await predict(expandedimg, model);
178 const detections = renderPredictions(
179   predictions,
180   image.width,
181   image.height
182 );
183 console.log("detected: ", detections);
184
185 detections.forEach((item) => {
186   const x = item["bbox"][0];
187   const y = item["bbox"][1];
188   const width = item["bbox"][2];
189   const height = item["bbox"][3];
190
191   // Draw the bounding box.
192   context.strokeStyle = "#00FFFF";
193   context.lineWidth = 4;
194   context.strokeRect(x, y, width, height);
195
196   // Draw the label background.
197   context.fillStyle = "#00FFFF";
198   const textWidth = context.measureText(
199     item["label"] + " " + (100 * item["score"]).toFixed(2)
200   ).width;

```

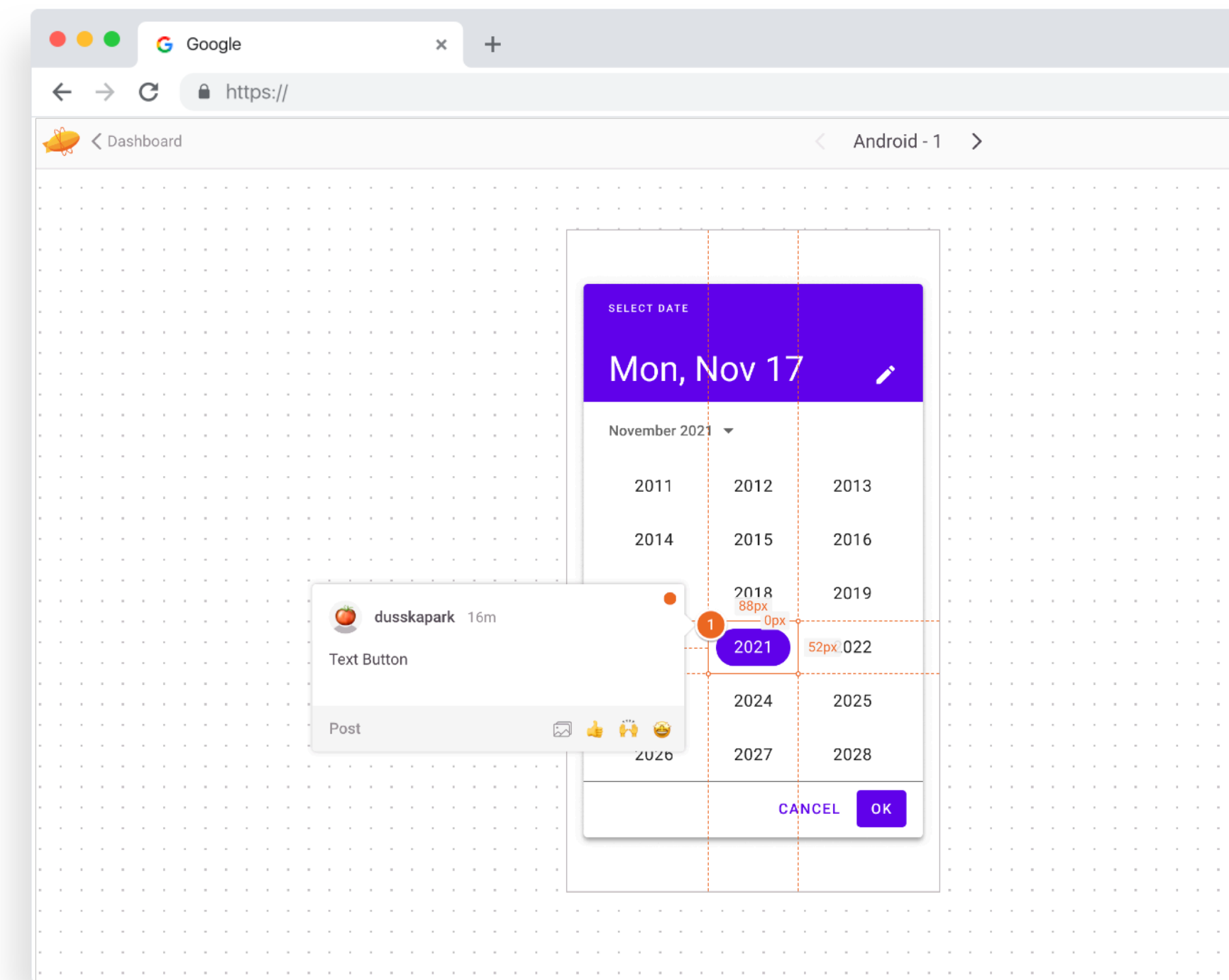
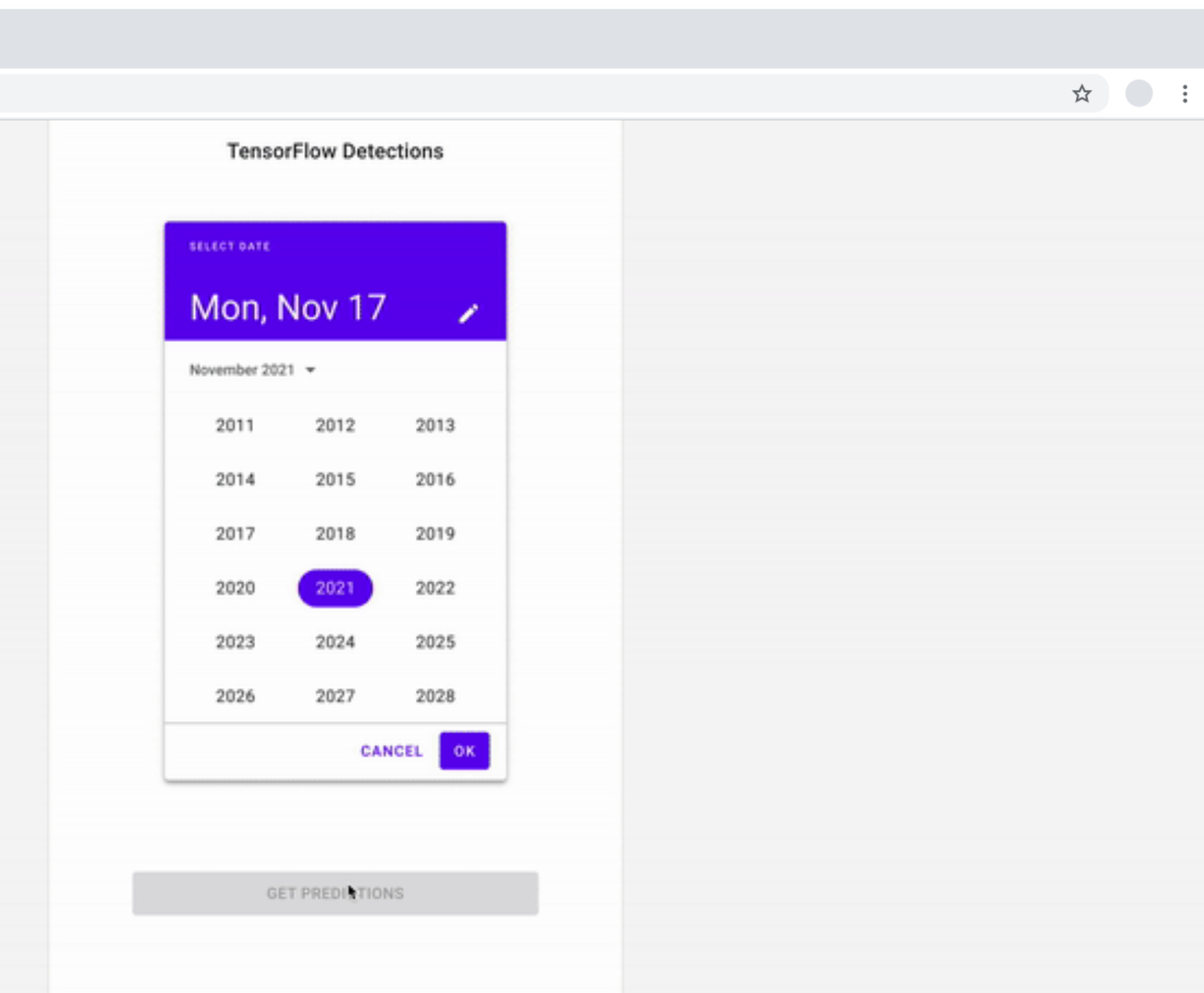
The console output shows the following JSON objects:

```

detected: {class: 2, label: "Image", score: "0.3644"}
detected: {class: 3, label: "Icon", score: "0.3321"}
detected: {class: 3, label: "Icon", score: "0.3253"}
detected: {class: 3, label: "Icon", score: "0.3211"}
detected: {class: 3, label: "Icon", score: "0.3199"}
detected: {class: 3, label: "Icon", score: "0.3173"}

```

# 모델을 Zeplin ML 프로젝트에 통합하기

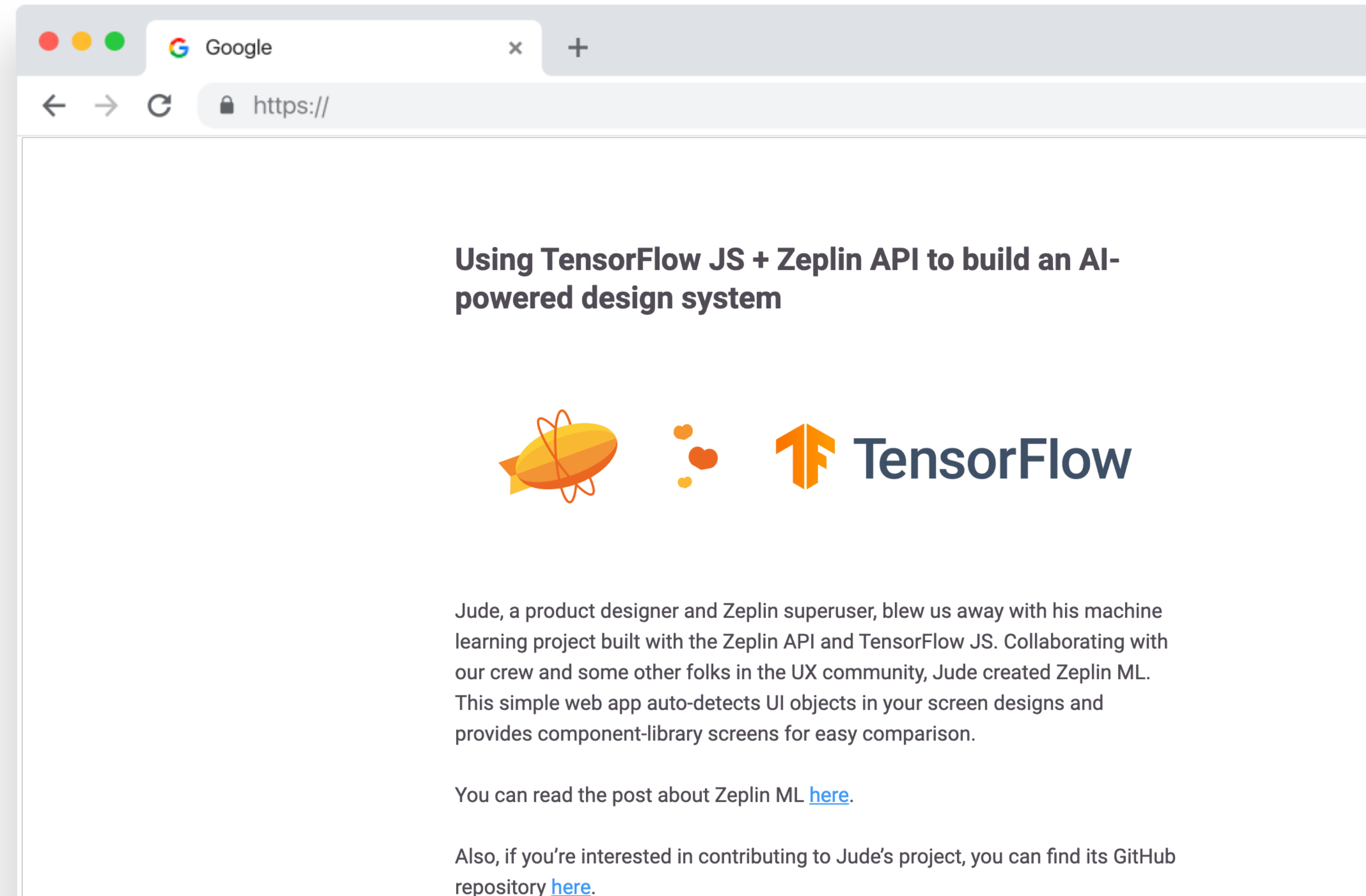


# Reactions & Next steps

# Reactions

“Jude, a product designer and Zeplin superuser, blew us away with his machine learning project built with the Zeplin API and TensorFlow JS.”

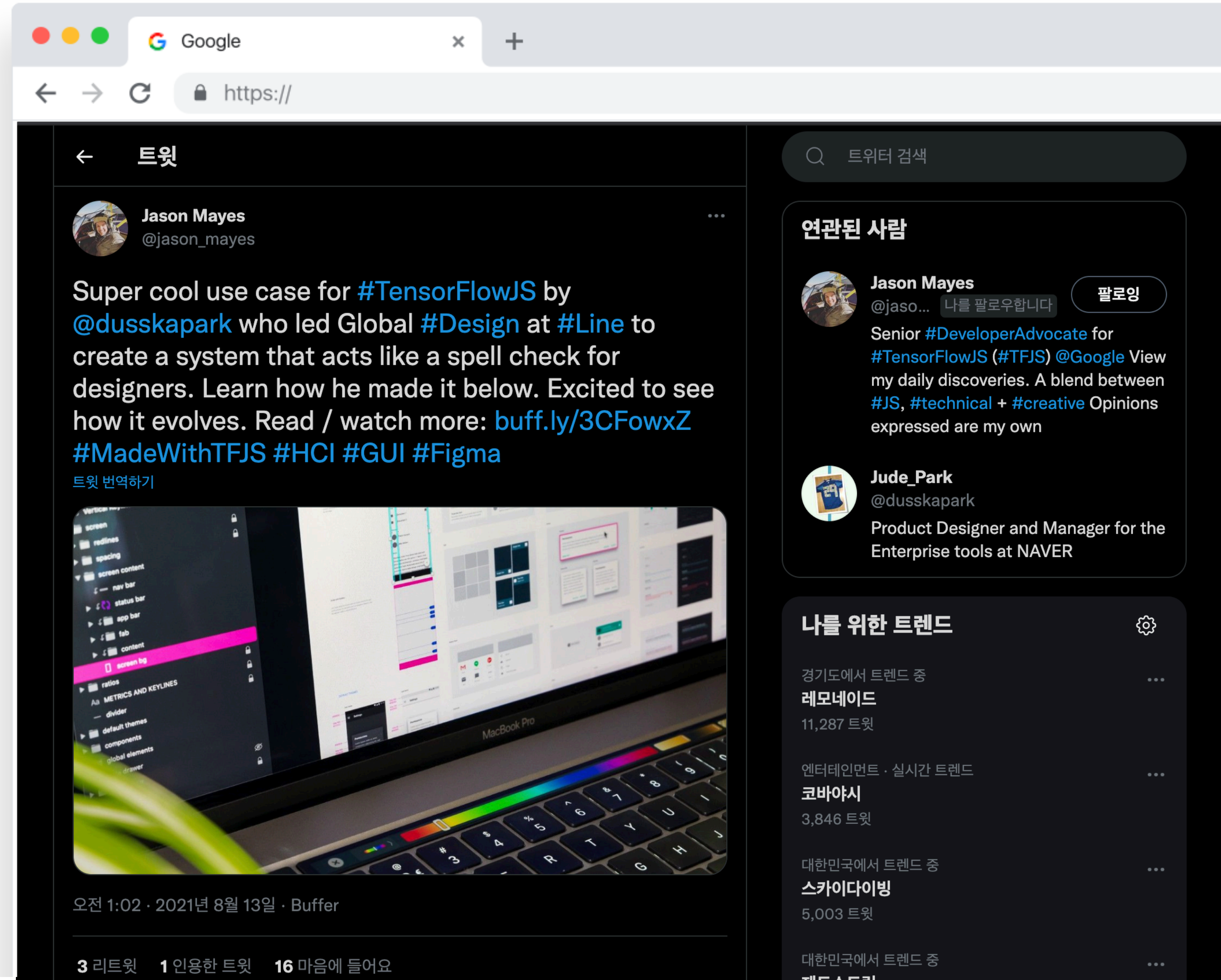
- Zeplin Newsletter Sep. 2021



# Reactions

“This is an amazing write up / journey to get all the data + produce a model like this. This may be a great candidate for #MadeWithTFJS show”

Jason Mayes  
Senior DevRel for TensorFlowJS



Google

https://

트윗

Jason Mayes  
@jason\_mayes

Super cool use case for [#TensorFlowJS](#) by [@dussskapark](#) who led Global [#Design](#) at [#Line](#) to create a system that acts like a spell check for designers. Learn how he made it below. Excited to see how it evolves. Read / watch more: [buff.ly/3CFowxZ](https://buff.ly/3CFowxZ) [#MadeWithTFJS](#) [#HCI](#) [#GUI](#) [#Figma](#)

트윗 번역하기

Jason Mayes  
@jason\_mayes  
Senior [#DeveloperAdvocate](#) for [#TensorFlowJS](#) ([#TFJS](#)) @Google View my daily discoveries. A blend between [#JS](#), [#technical](#) + [#creative](#) Opinions expressed are my own

Jude\_Park  
@dussskapark  
Product Designer and Manager for the Enterprise tools at NAVER

나를 위한 트렌드

경기도에서 트렌드 중  
**레모네이드**  
11,287 트윗

엔터테인먼트 · 실시간 트렌드  
**코바야시**  
3,846 트윗

대한민국에서 트렌드 중  
**스카이다이빙**  
5,003 트윗

대한민국에서 트렌드 중  
**레드스드릭**

오전 1:02 · 2021년 8월 13일 · Buffer

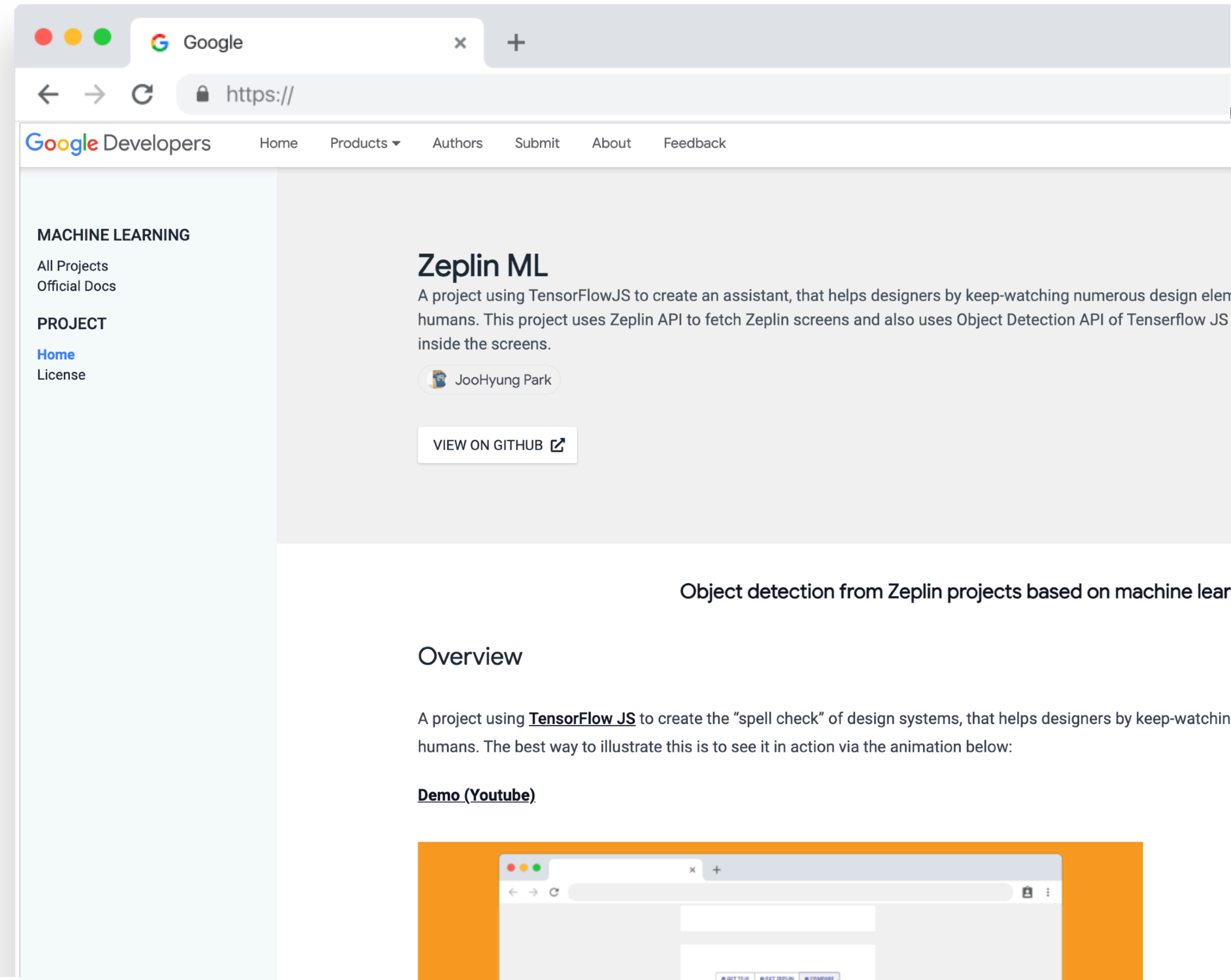
3 리트윗 1 인용한 트윗 16 마음에 들어요



# Reactions

- Google Dev Library
- TensorFlow Blog
- Youtube Live

- 
- 



와 진짜 짠이다. 필요했던 것이에요.

컬러, 타이포, 토크도 됐으면 좋겠어요.

Figma용 플러그인도 만들어주세요!

Object Detection 이렇게도 쓰는군요.

정확히 뭐가 된다는 것인지 모르겠지만

재미있네요. 약간 느린 것 같은데,

MobileNet V3도 한번 써보세요. 이제

와 진짜 찐이다. 필요했던 것이에요.

컬러, 타이포, 토크도 됐으면 좋겠어요.

Figma용 플러그인도 만들어주세요!

Object Detection 이렇게도 쓰는군요.

정확히 뭐가 된다는 것인지 모르겠지만

재미있네요. 약간 느린 것 같은데,

MobileNet V3도 한번 써보세요. 이제

와 진짜 짤이다. 필요했던 것이에요.

컬러, 타이포, 토크도 됐으면 좋겠어요.

Figma용 플러그인도 만들어주세요!

Object Detection 이렇게도 쓰는군요.

정확히 뭐가 된다는 것인지 모르겠지만

재미있네요. 약간 느린 것 같은데,

MobileNet V3도 한번 써보세요. 이제

# Next Steps



사이즈, 컬러 등  
다양한 디자인 체크



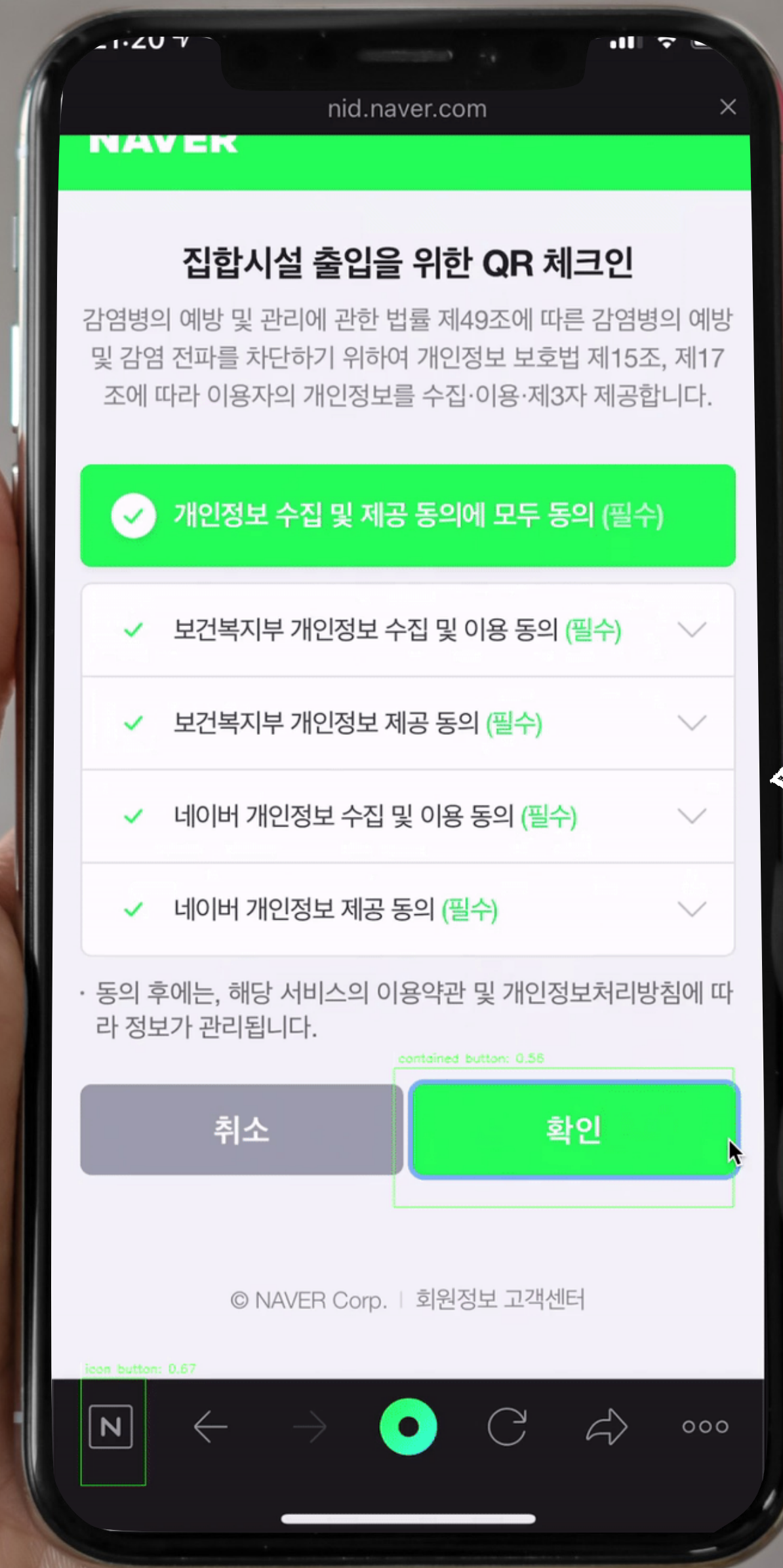
빠른 레이턴시,  
모델 퍼포먼스 향상

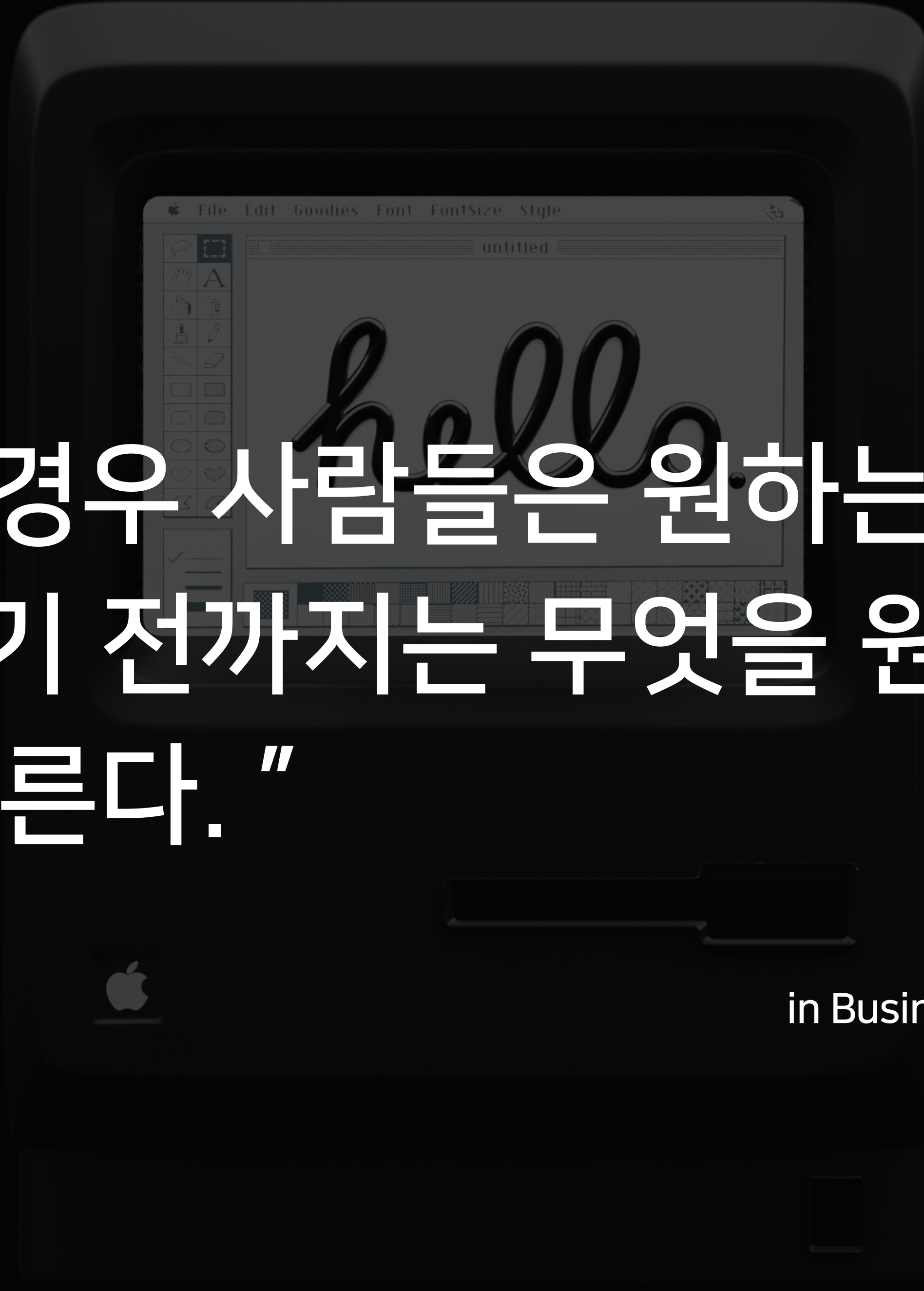


Figma 등 디자인툴  
용 플러그인 추가

# Lesson learned

작년 추석에 집에서  
무작정 만들어본  
버튼 UI 검출기





“ 많은 경우 사람들은 원하는 것을  
보여주기 전까지는 무엇을 원하는  
지도 모른다.”

- Steve Jobs

in Business Week 12 May 1998





Photo by Alexander Shatov on Unsplash