



자신이 그린 그림을 암호로,

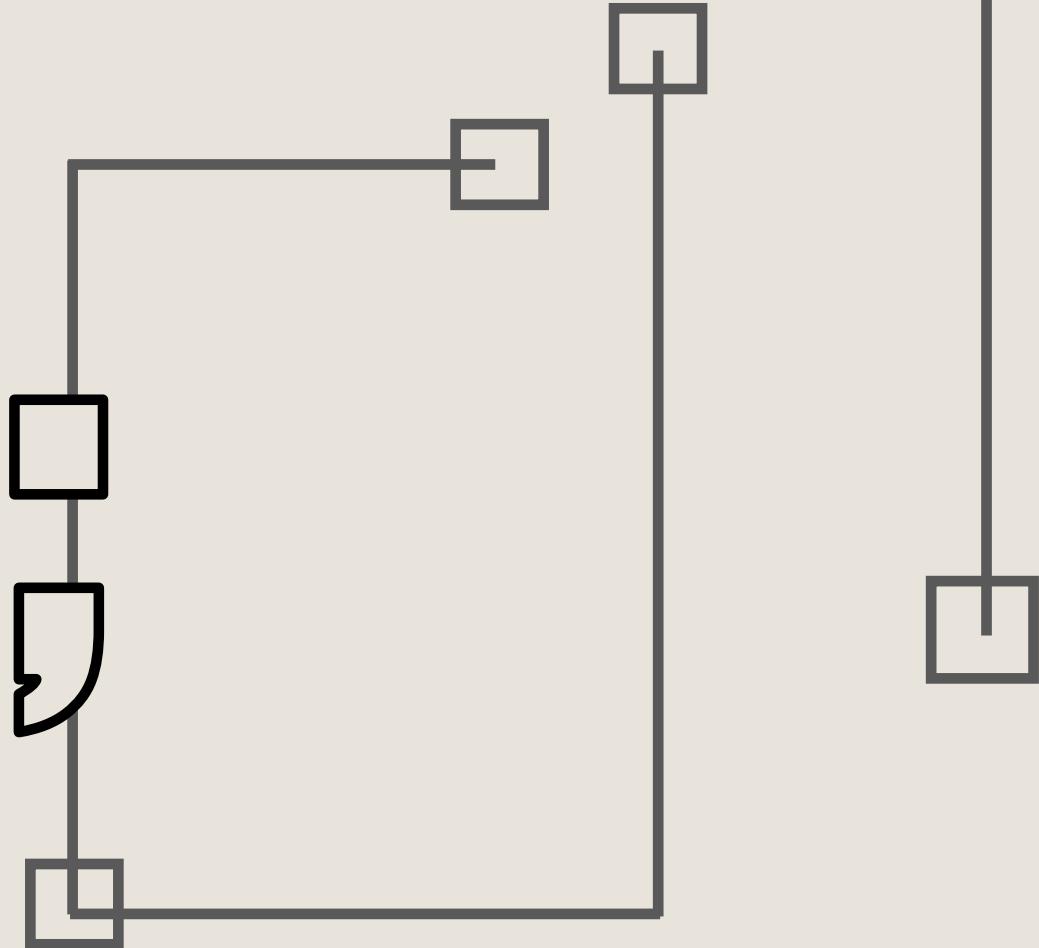
**Pass-**  
**D**raw

김나나

---

김은아(하얼빈공업대학교, 컴퓨터공학과)

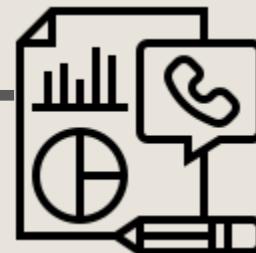
나윤서(연세대학교, 영어영문학과)



# Index



Outro -  SWOT  
 예상 활용 분야



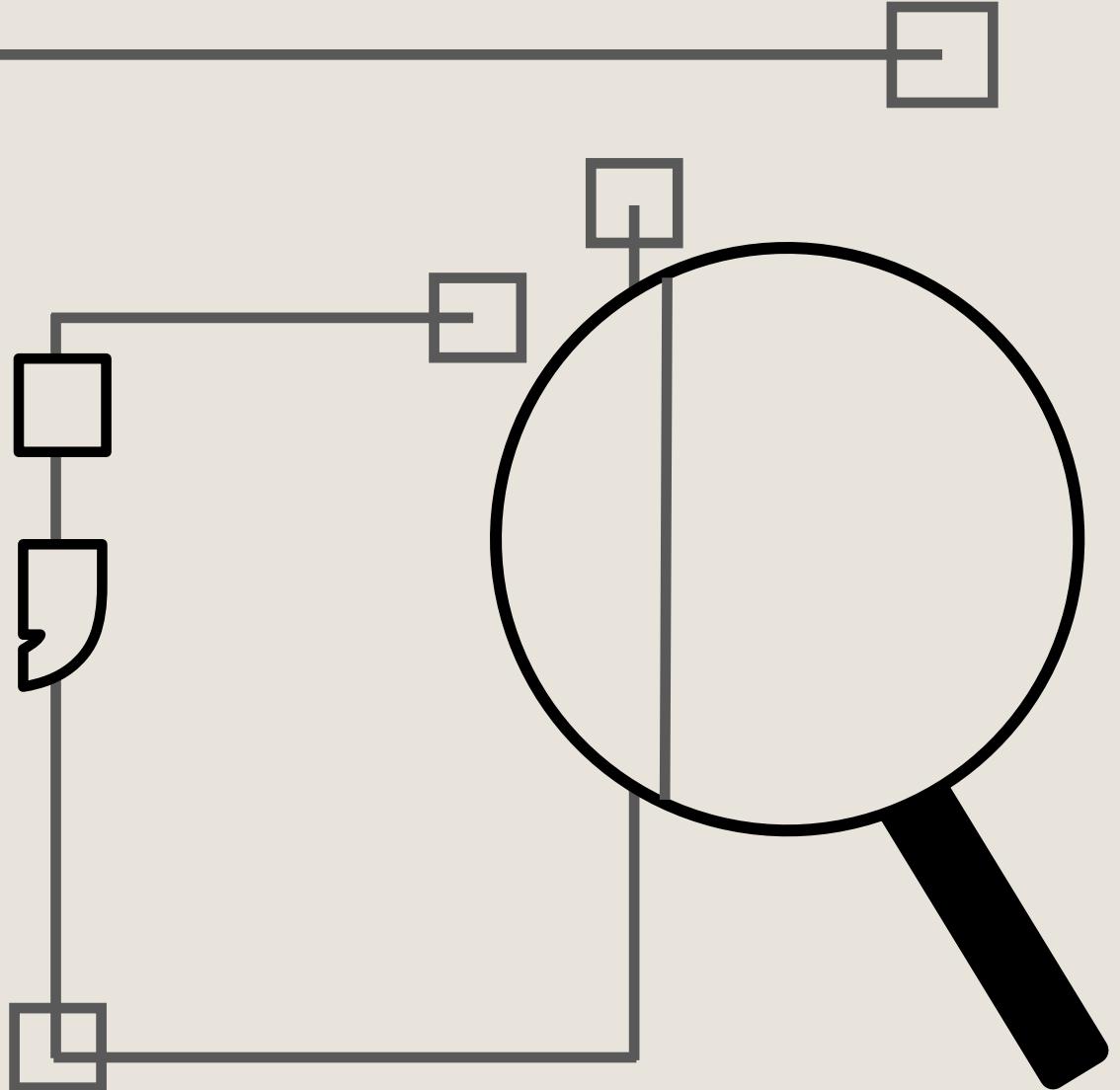
Service -  데이터 소개  
 데이터 구조



Intro -  기획배경  
 기술적 배경

# Intro

- 기획배경
- 기술적 배경



# Intro-

기획배경

## 어디에서나 언제나 필요한 아이디와 패스워드! 비밀 번호를 대하는 당신의 태도는?



1

썼던 암호를  
다른 사이트에서도 계속  
사용하는 **돌려 막기형**



2

모바일 디바이스 자동 암호 설정으로  
**나도 모르는 암호를 설정해 놓고**  
디바이스만 믿고 있는 헛똑똑이형



3

아이디와 비밀번호를  
서로 바꿔가며  
사용하는 **연쇄작용형**

### Key problem

다음의 유형 중에 당신이 포함된다면?  
**당신의 보안 상태는 '위험'입니다!**

마이데이터 등 서비스의 시작으로  
한 곳으로 모이기 시작한 정보, 더욱더 중요해진 정보 보안!  
그럼에도 숫자와 문자의 조합에는 한계가 있고  
그마저도 쓰는 패스워드는 정해져 있는데...  
점점 약해져 가는 정보 보안, 해결 방법은 없을까?

# Intro-

기획배경

## Key point

문제 해결을 위한 인사이트

## 어떤 보안 방법이 우리에게 필요한가?

취약한 보안 상태를 타개함과 동시에 사용자의 지속적인 활용을 위해  
다음의 내용이 필수적으로 선결되어야 한다

### 다양성



숫자와 문자열의 조합으로  
한정되어 있던  
패스워드의 다양성(가지 수)

### 편리함



장소와 시간에 구애 받지 않고  
단시간에 사용할 수 있는 편리함

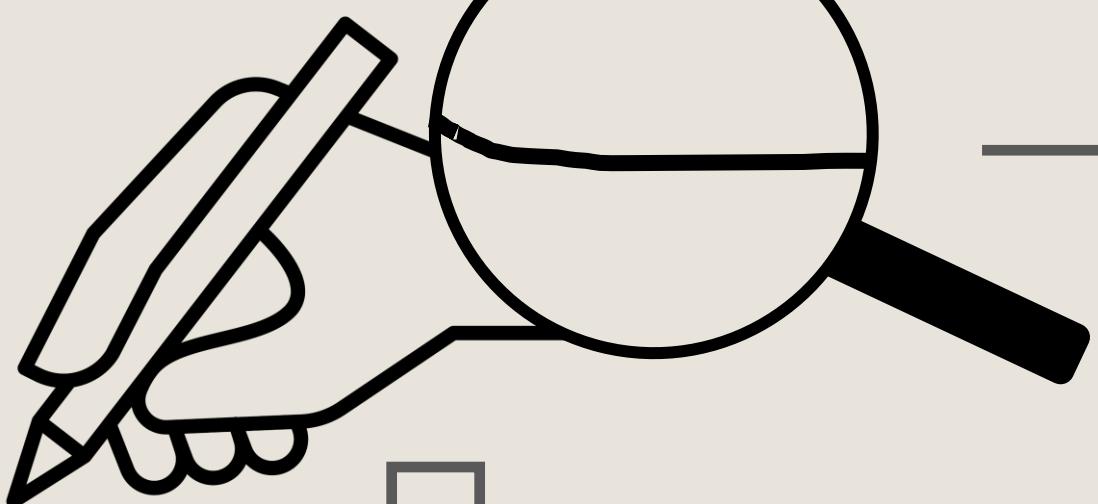
### 보안 성능



한 패스워드를 오래 사용해  
낮은 변동이 이루어짐에도  
높은 보안 성능

# Intro-

기획배경



**Think point** 문제 해결을 위한 인사이트  
숫자·문자 조합의 패스워드를  
2D or 3D로 구현한다면?

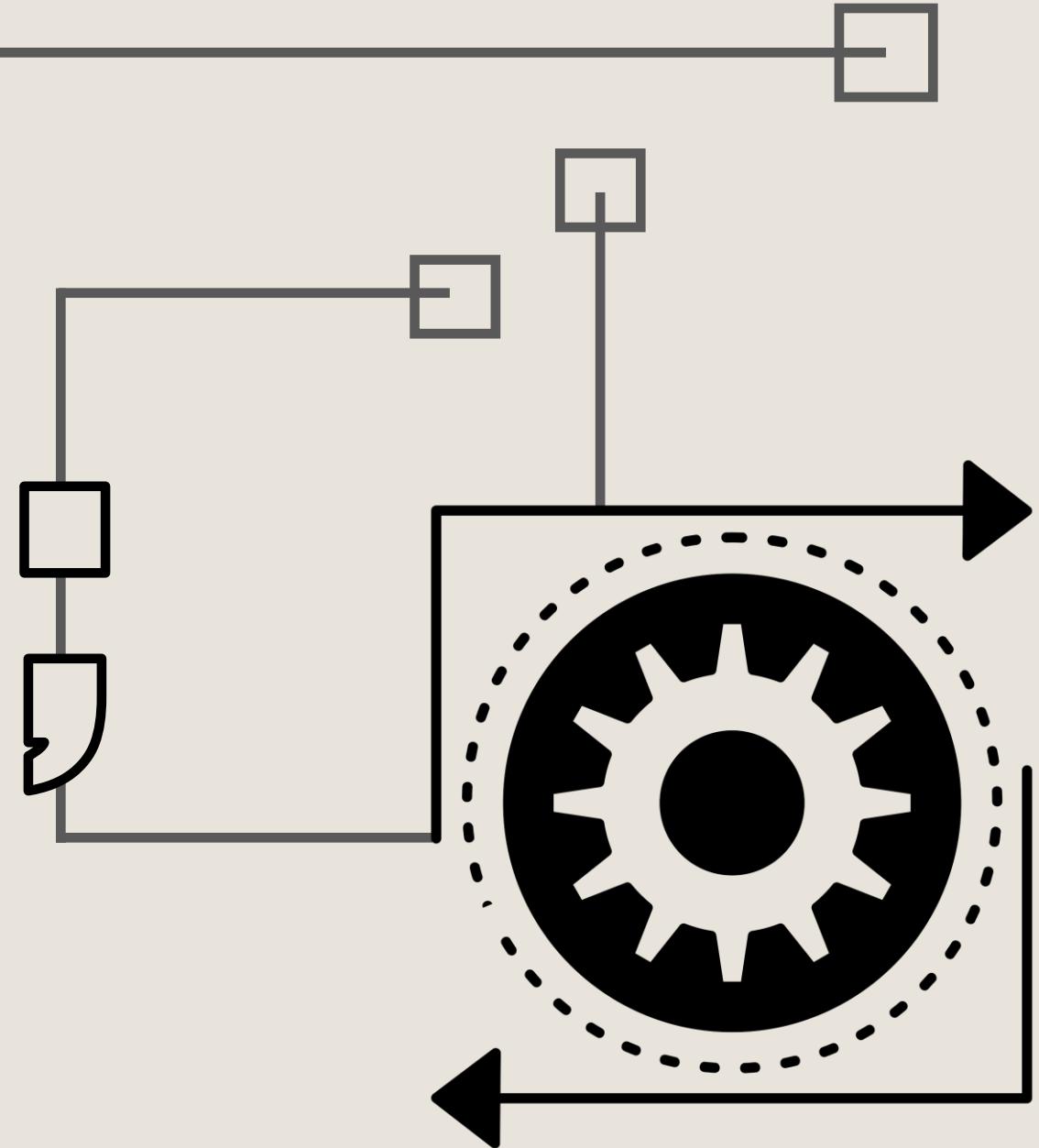
실행품(사진)을 기반으로 하여 기존의 패스워드 방식보다  
복잡성과 다양성을 갖추고  
패스워드를 푸는 키를 간단한 이미지 데이터(그림)로 풀어낸다면  
편리함까지 이끌어 낼 수 있지 않을까?



# Serv- ice

데이터 소개

Passdraw 구조

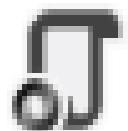
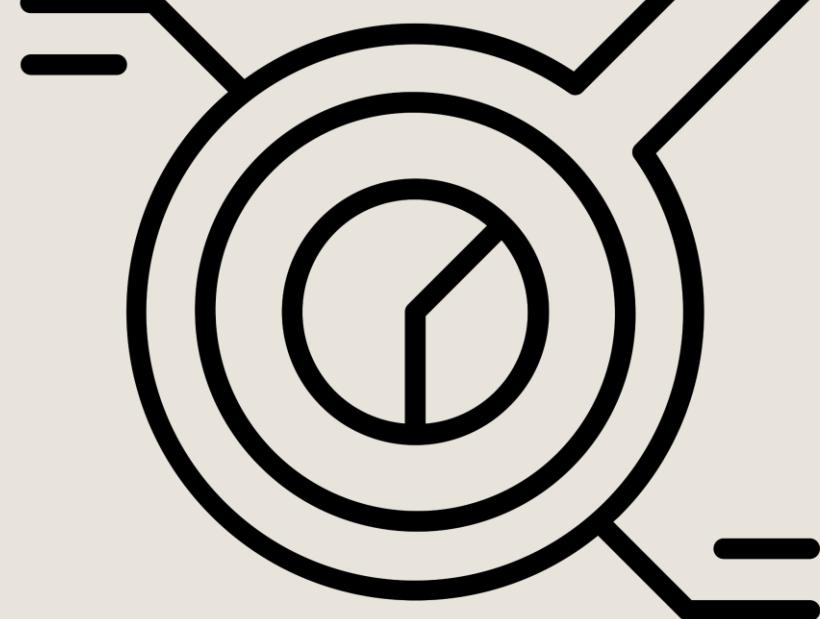


# Service



## 데이터소개

- 분류체계 객체 1,002종별 실사이미지와 추상이미지 3종(스케치, 일러스트레이션, 픽토그램)을 데이터셋으로 한다
- 추상이미지 3종의 경우 제작자의 성별, 나이, 이미지 제작의 숙련도 등 기본 메타 정보를 정의하여 추가한다
- 인공지능 학습용 데이터 구축량은 실사이미지 약 20,000장, 스케치 25만장, 픽토그램 15만장, 일러스트레이션 10만장으로 구성한다



category\_all



category\_folder\_description\_euc\_kr



image\_data

다음의 데이터를 바탕으로 PassDraw 인공지능 모델을 구축!



# Service

데이터 구조

## Pass-Draw의 데이터구조

### Pass Draw

PassDraw의 인공지능 모델을  
통한 암호 사진 인식

PassDraw의 인공지능 모델을  
통한 암호 그림 인식

PassDraw 인식 결과 70%  
이하일 경우 잠금 해제 실패

PassDraw 인식 결과 70% 일  
경우 잠금 해제

암호를 등록하고자 하는  
대상 등록

암호를 동록한 대상의  
사진 혹은 그림 입력

사용자

# Service



## Teachable Machine 은 무엇일까?

구글이 2019년 발표한 누구나 활용하는 웹기반 머신러닝 툴

'티처블 머신(Teachable Machine) 2.0'을 통해 이미지의 데이터 패턴을 인식하고, 인식한 이미지를 해당하는 카테고리에 할당하는 방법으로 Pass Draw 머신러닝 학습을 진행했다.

Teachable Machine



# Service-

## PassDraw, javascript

```
● ● ●

<div>Teachable Machine Image Model</div>
<button type="button" onclick="init()">Start</button>
<div id="webcam-container"></div>
<div id="label-container"></div>
<script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@1.3.1/dist/tf.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/@teachablemachine/image@0.8/dist/teachablemachine-
image.min.js"></script>
<script type="text/javascript">
  // More API functions here:
  // https://github.com/googlecreativelab/teachablemachine-community/tree/master/libraries/image

  // the link to your model provided by Teachable Machine export panel
  const URL = "./my_model/";

  let model, webcam, labelContainer, maxPredictions;

  // Load the image model and setup the webcam
  async function init() {
    const modelURL = URL + "model.json";
    const metadataURL = URL + "metadata.json";

    // load the model and metadata
    // Refer to tmImage.loadFromFiles() in the API to support files from a file picker
    // or files from your local hard drive
    // Note: the pose library adds "tmImage" object to your window (window.tmImage)
    model = await tmImage.load(modelURL, metadataURL);
    maxPredictions = model.getTotalClasses();

    // Convenience function to setup a webcam
    const flip = true; // whether to flip the webcam
    webcam = new tmImage.Webcam(200, 200, flip); // width, height, flip
    await webcam.setup(); // request access to the webcam
    await webcam.play();
    window.requestAnimationFrame(loop);
  }
}
```

# Service-

## PassDraw.js

```
        labelContainer.appendChild(document.createElement("div"));
    }

async function loop() {
    webcam.update(); // update the webcam frame
    await predict();
    window.requestAnimationFrame(loop);
}

// run the webcam image through the image model
async function predict() {
    // predict can take in an image, video or canvas html element
    const prediction = await model.predict(webcam.canvas);
    for (let i = 0; i < maxPredictions; i++) {
        const classPrediction =
            prediction[i].className + ": " + prediction[i].probability.toFixed(2);
        labelContainer.childNodes[i].innerHTML = classPrediction;
    }
}
</script>
```

# Service-

## PassDraw, P5.js

```
● ● ●

<div>Teachable Machine Image Model - p5.js and ml5.js</div>
<script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.9.0/p5.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.9.0/addons/p5.dom.min.js"></script>
<script src="https://unpkg.com/ml5@latest/dist/ml5.min.js"></script>
<script type="text/javascript">
  // Classifier Variable
  let classifier;
  // Model URL
  let imageModelURL = './my_model/';

  // Video
  let video;
  let flippedVideo;
  // To store the classification
  let label = "";

  // Load the model first
  function preload() {
    classifier = ml5.imageClassifier(imageModelURL + 'model.json');
  }

  function setup() {
    createCanvas(320, 260);
    // Create the video
    video = createCapture(VIDEO);
    video.size(320, 240);
    video.hide();

    flippedVideo = ml5.flipImage(video);
    // Start classifying
    classifyVideo();
  }

  function draw() {
    background(255);
    image(video, 0, 0, 320, 240);
    image(flippedVideo, 0, 0, 320, 260);

    if (label != "") {
      fill(255, 0, 0);
      textSize(16);
      text(label, 10, 20, 200, 40);
    }
  }

  function classifyVideo() {
    classifier.classify(video, gotResult);
  }

  function gotResult(error, results) {
    if (error) {
      console.error(error);
    } else {
      let index = 0;
      let maxConfidence = 0;
      let resultLabel = '';
      for (let i = 0; i < results.length; i++) {
        if (results[i].confidence > maxConfidence) {
          maxConfidence = results[i].confidence;
          resultLabel = results[i].label;
        }
      }
      label = resultLabel;
    }
  }
</script>
```

# Service-

## PassDraw, P5.js

```
function draw() {
  background(0);
  // Draw the video
  image(flippedVideo, 0, 0);

  // Draw the label
  fill(255);
  textSize(16);
  textAlign(CENTER);
  text(label, width / 2, height - 4);
}

// Get a prediction for the current video frame
function classifyVideo() {
  flippedVideo = ml5.flipImage(video)
  classifier.classify(flippedVideo, gotResult);
  flippedVideo.remove();
}

// When we get a result
function gotResult(error, results) {
  // If there is an error
  if (error) {
    console.error(error);
    return;
  }
  // The results are in an array ordered by confidence.
  // console.log(results[0]);
  label = results[0].label;
  // Classifiy again!
  classifyVideo();
}
```

</script>

# Service

□ 레이터 이미셋을 기반으로 Pass Draw 머신 러닝 학습 과정을 알아볼까?

≡ Teachable Machine

사무용 책상

26 이미지 샘플

웹캠 업로드

일자형 책상

30 이미지 샘플

웹캠 업로드

학습

학습 중...

00:37 - 2 / 50

고급

에포크: 50

배치 크기: 16

학습률: 0.001

기본값 초기화

고급 설정

미리 보기

모델 내보내기

여기에서 모델을 미리 확인하려면 먼저 왼쪽에서 모델을 학습시켜야 합니다.

한국어

release-2-4-4 - 2.4.4#95c54c

# Service-

- 레이터 이미셋을 기반으로 Pass Draw 머신 러닝 학습 과정을 알아볼까?

### Teachable Machine

웹캠 업로드

독서실 책상

29 이미지 샘플

웹캠 업로드

사무용 책상

26 이미지 샘플

웹캠 업로드

학습

모델 학습 완료됨

고급

에포크: 50

배치 크기: 16

학습률:

0.001

기본값 초기화

고급 설정

# Service-

## 전자레인지 사진 식별

### Teachable Machine

장식장

52 이미지 샘플

웹캠 업로드



TV거실장

98 이미지 샘플

웹캠 업로드



거실테이블

학습

모델 학습 완료됨

고급

에포크: 50

배치 크기: 16

학습률:

0.001

기본값 초기화

고급 설정

미리  
보기

↑ 모델 내보내기

파일에서 이미지를 선택하거나 여  
기로 드래그 앤 드롭하세요.

Google Drive에서 이미지 가져오  
기



# Service-



전자레인지 사진 식별

## ≡ Teachable Machine

장식장



52 이미지 샘플

웹캠

업로드



학습

모델 학습 완료됨

고급

에포크:

50



배치 크기:

16



학습률:

0.001



기본값 초기화



고급 설정



미리  
보기

↑ 모델 내보내기

파일에서 이미지를 선택하거나 여  
기로 드래그 앤 드롭하세요.



Google Drive에서 이미지 가져오  
기



TV거실장



98 이미지 샘플

웹캠

업로드



거실테이블



52 이미지 샘플

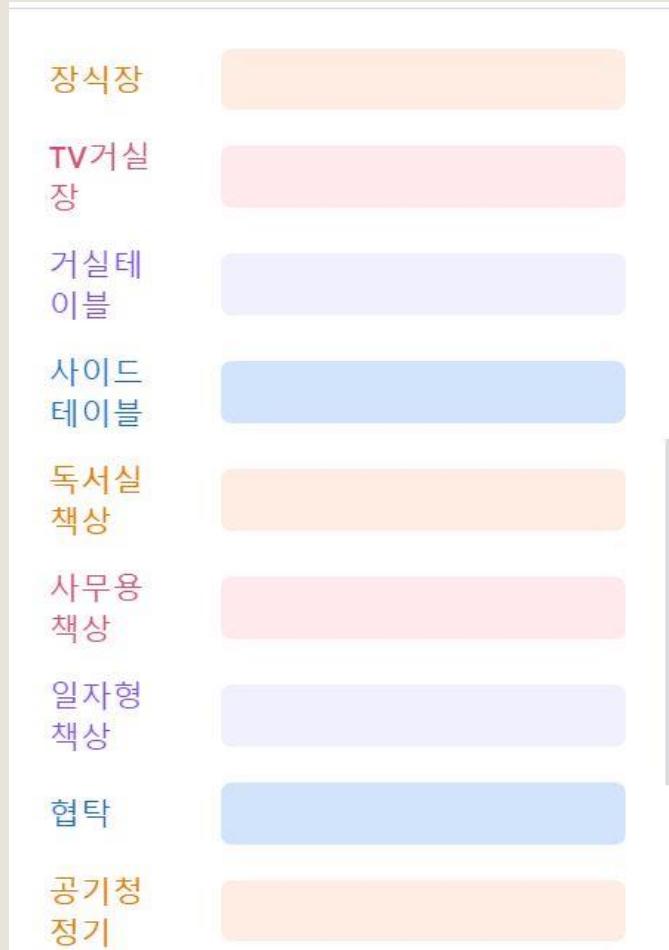
한국어

release-2-4-4 - 2.4.4#95c54c

# Service-



## 전자레인지 사진 식별

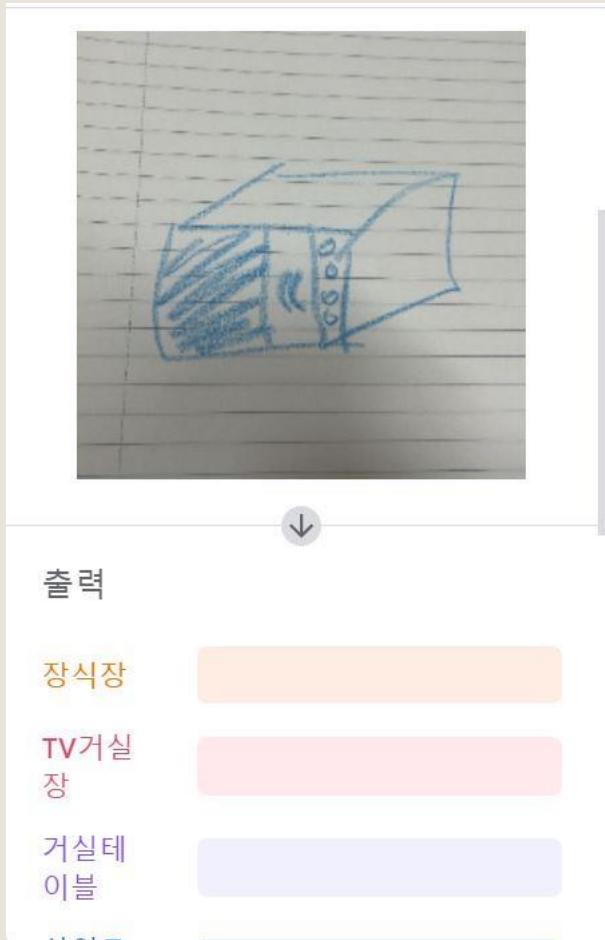


학습 데이터인 이미지 데어터 셋의  
이미지가 아닌 전자레인지의 사진을  
테스트 했을 때 전자레인지 100%  
결과가 나왔다.

# Service-



## 전자레인지 그림 식별



학습 데이터인 이미지 데어터 셋의  
이미지가 아닌 전자레인지의 그림을  
테스트 했을 때 전자레인지 87%  
결과가 나왔다.



# Service-



## 협탁 사진 식별



출력

장식장

TV거실  
장

거실테  
이블



학습 데이터인 이미지 데어터 셋의  
이미지가 아닌 협탁의 사진을 테스트  
했을 때 협탁 73% 결과가 나왔다.

# Service

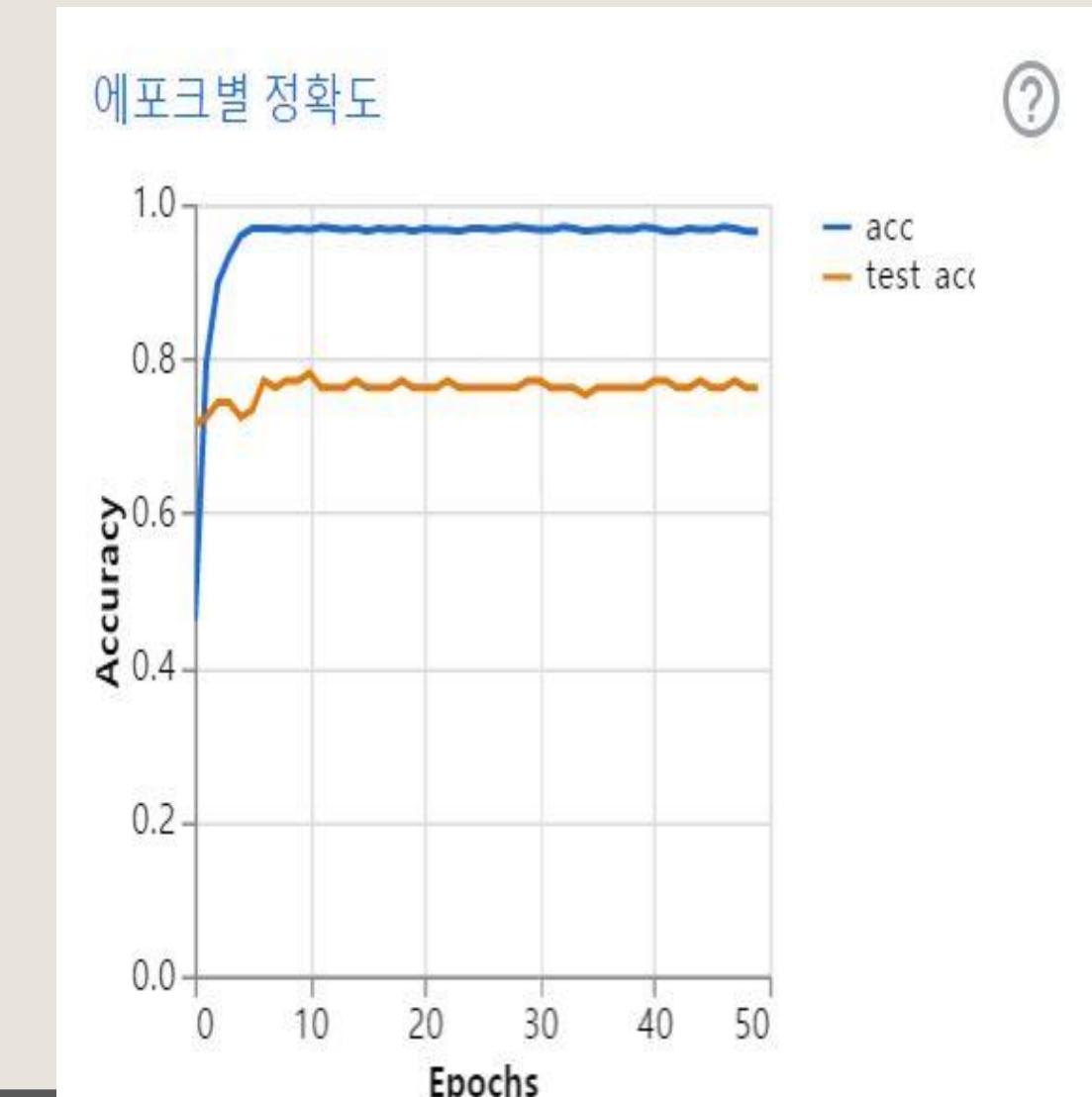
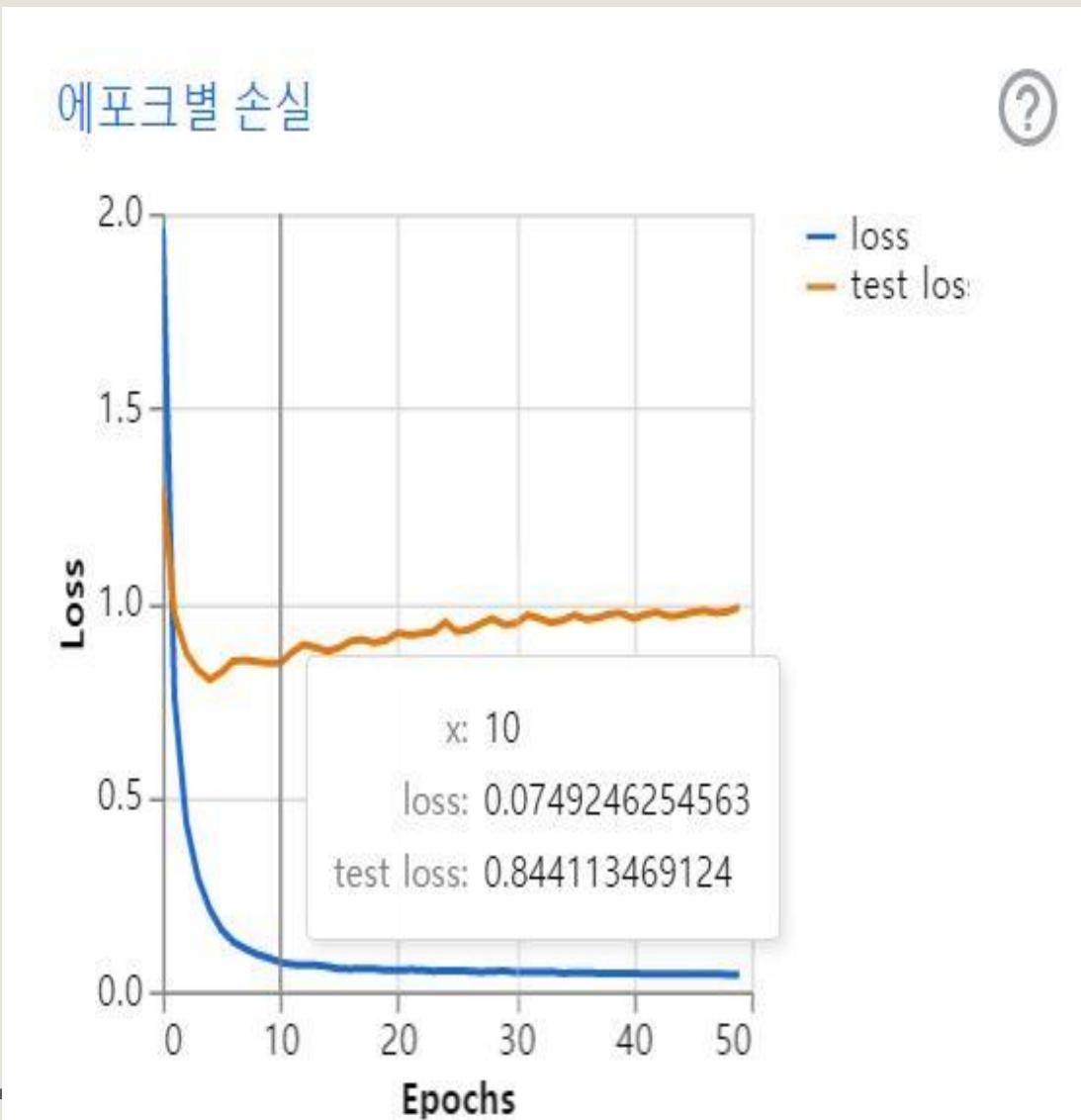
호흡을 돋우는 기계

	장식장	TV거실장	거실테이블	사이드테이블	독서실 책상	사무용 책상	일자형책상
Class	7	0	0	1	0	0	0
TV거실장	0	13	0	0	0	0	0
거실테이블	0	0	3	2	0	1	0
사이드테이블	1	0	1	0	0	1	0
독서실 책상	1	0	0	2	2	0	0
사무용 책상	0	0	0	1	1	0	1
일자형책상	0	0	0	1	0	0	3
협탁	0	0	0	0	1	0	1
공기청정기	0	0	0	0	0	0	0
스탠드에어컨	0	0	0	0	0	0	0
벽걸이에어컨	0	0	0	0	0	0	0
잔디깎이	0	0	0	0	0	0	0
환풍기	0	0	0	0	0	0	0
전자레인지	0	0	0	0	0	0	0
가스레인지	0	0	0	0	0	0	0
가스레인지후드	0	0	0	0	0	0	0

기록을 관리하는 기계

일자형책상	0.60	5
협탁	0.25	4
공기청정기	0.75	8
스탠드에어컨	1.00	6
벽걸이에어컨	1.00	8
잔디깎이	1.00	5
환풍기	1.00	4
전자레인지	0.75	8
가스레인지	1.00	8
가스레인지후드	1.00	8

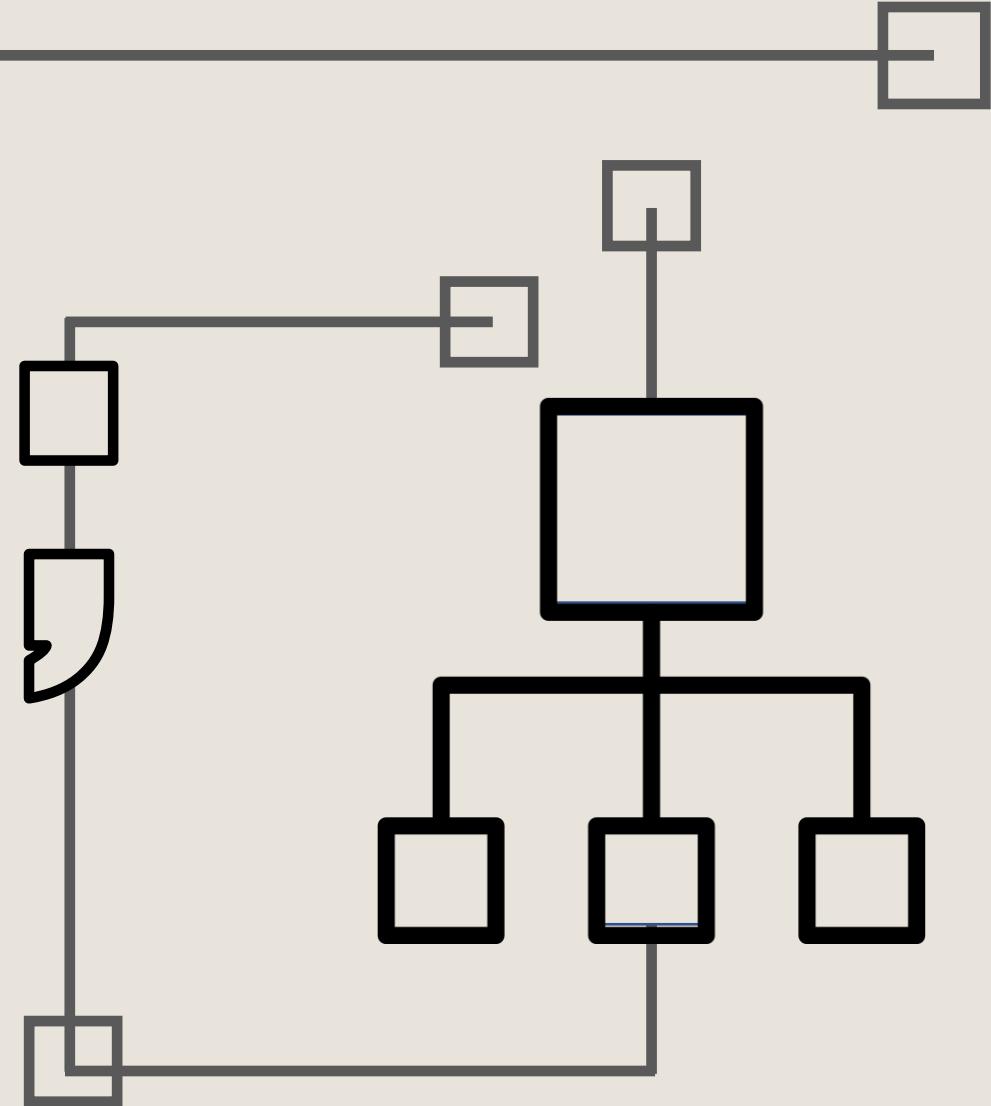
# Service-



# Outro

□ SWOT

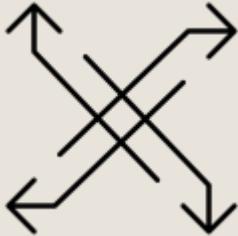
□ 예상 활용 분야



# Outro-



## Self-evaluation 현재 상황-에 대해서 Passdraw의 현지표는?



개개인 고유의 필압과  
사진의 다양성을 통해 갖는  
**패스워드의 복잡성**



보안 해제시  
**활용 요소 단순화** 설정으로  
요구되는 드로잉 요소 간편화



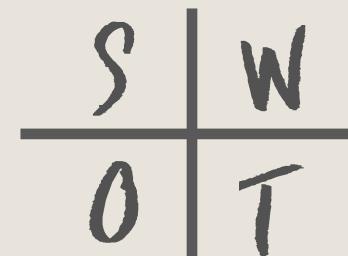
좁은 화면 면적의 경우  
보다 정확한 드로잉을 위한  
**보조도구 필요**



**핀테크**의 사업 영역  
확장으로 인한  
모바일 디바이스 보안



아이폰, 아이패드, 맥북 등  
**디바이스 페어링** 이용자 증가로 인한 개  
인 정보 보안 영역



숫자·문자 조합의 패스워드  
한붓그리기 방식의 패스워드에  
의숙한 **디바이스** 유저

# Outro-

기대효과 및 예상 활용 분야

Our expectation  
앞으로-에 대해서

## 어디서 어떻게 Passdraw가 사용될 수 있을까?

높은 수준의 보안이 요구되는  
인터넷 뱅킹 금융 서비스 및 어플리케이션에 활용될 것이라 사료된다

### 은행



카드 발급, 통장 개설 시  
사용되는 패스워드 설정

\*\* 은행사 아이디 가입에 사용되는  
숫자·영문 패스워드 대체

### 핀테크



카카오뱅크와 같은 핀테크에서의  
자산관리 시 사용되는 패스워드 설정

\*\* 기존의 9개의 점으로 하는 한붓 그리기 형식의  
모바일 디바이스 패스워드 대체

### 모바일 결제 서비스



삼성페이 등 모바일 결제 서비스 사용시  
필요한 패스워드 설정

\*\* 모바일 디바이스 자체의 패스워드와 결제 서비스가  
동시에 해제된다는 점을 이중 보완으로 해결



자신이 그린 그림을 암호로, Pass Draw

**Thank-**

**you**

---

감사합니다

