
Teachable Machine을 활용한 수업 진단



제출일	2020년 04월 03일	전공	컴퓨터 시스템과
과목명	머신러닝 기초	학번	201645081
담당교수	이세훈	이름	김지태

목차

1. 머신러닝서비스개요

2. 데이터준비

3. 머신러닝모델개발과정

4. 서비스개발과정(코드설명포함)

5. 실행화면캡처

6. 후기

-참고문헌(인터넷사이트, 유튜브URL, 책 등)

1. 머신러닝 서비스개요

요즘 코로나 때문에 학교들이 인터넷을 통한 수업을 진행하고 있습니다. 인터넷으로 수업을 한다면 시간과 공간 그리고 바이러스부터 자유롭고 이동시간을 단축함으로써 시간 활용도가 높아지는 장점이 있습니다. 하지만 반대로 실제 사람 대 사람으로 하는 수업이 아님으로 집중도 이해도 학습도가 떨어진다는 단점이 생깁니다. 그런 문제를 해결하기 위해서 학생이 수업을 집중하는지 자세로 판단을 하는 간단한 서비스를 생각해 보았고 학생들이 수업에서 사용하는 교과서를 잘 가지고 있는지 확인하는 것도 생각했다.

1. 포즈를 이용한 집중도 확인
2. 이미지 확인으로 교과서 확인

2. 데이터준비

데이터는 노트북 웹캠과 핸드폰을 활용해서 사진을 옮기는 방법으로 진행했다.

3. 머신러닝 모델개발과정

머신러닝 모델 개발과정은 Teachable Machine을 활용하여 진행을 할 것입니다.

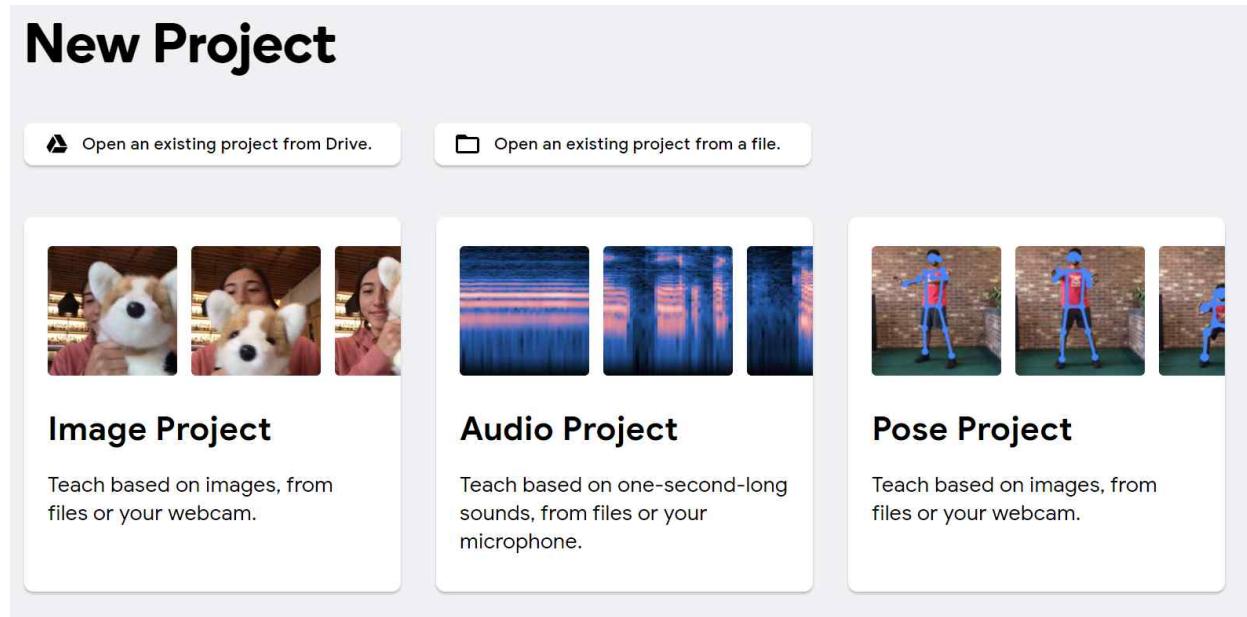


그림1.Teachable Machine 메인 화면.

원하는 목록을 선택. 자세는 Pose Project를 들어가 줍니다.

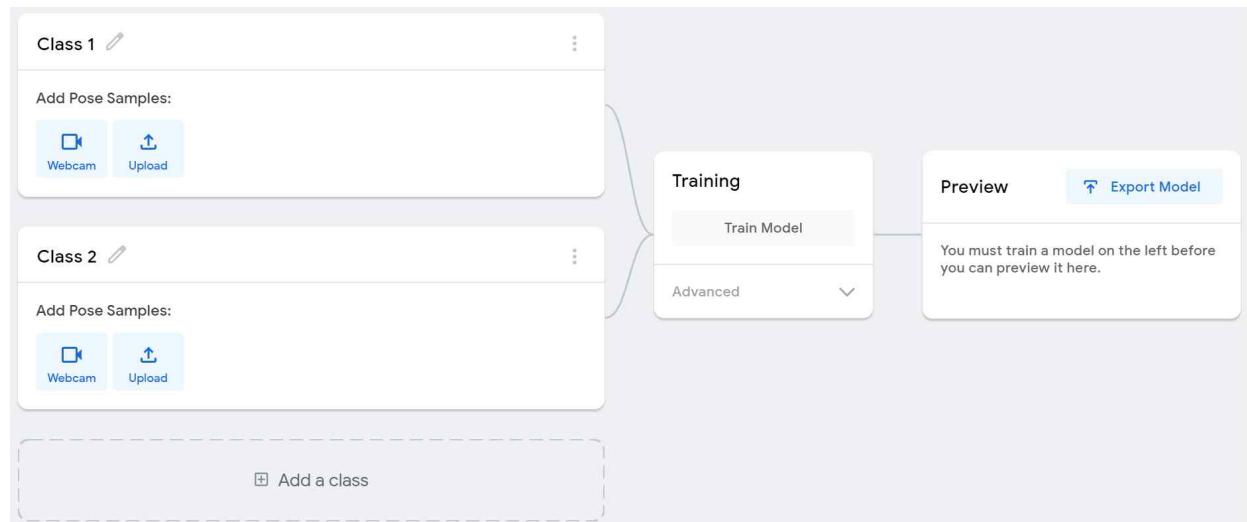


그림2. Teachable Machine 프로젝트 생성 화면.

모델을 등록하기 위해서는 두 가지 이상 경우의 수가 필수입니다. 그렇기에 처음 세팅은 Class2까지 있습니다. 추가적인 상황을 등록을 원하시면 Add a class를 눌러주시면 됩니다.

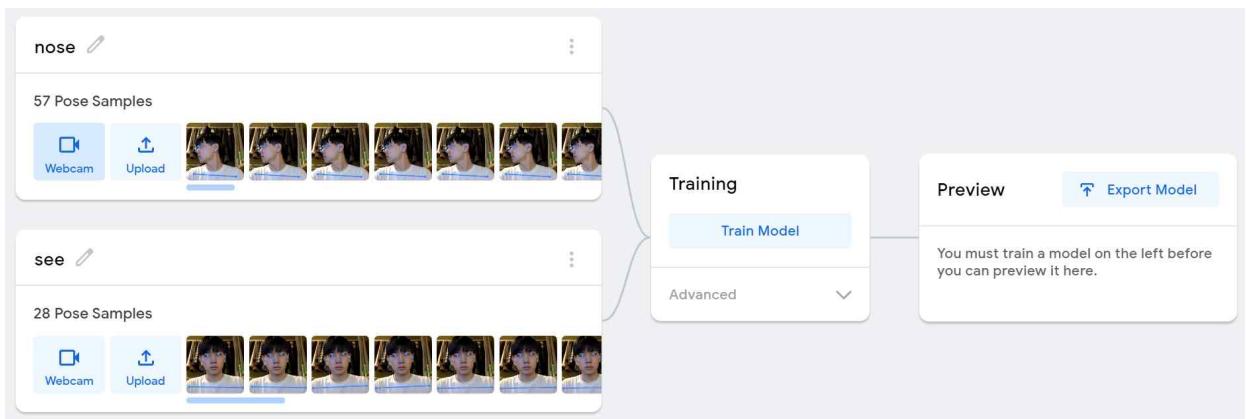


그림3. Teachable Machine 프로젝트 모델 생성 과정.

저는 두 가지 경우로 생성하려고 합니다. 원하는 동작을 사진 또는 웹캠을 이용하여 등록이 가능합니다. 등록이 끝났다면 Train Model 버튼을 클릭하여 모델을 생성해 줍니다.

개인적인 생각으로 케이스는 필요한 것부터 차츰차츰 늘려 나가는 것이 좋다고 생각되고 모델을 생성하기 위한 이미지들은 많다고 좋은 것이 아닙니다.

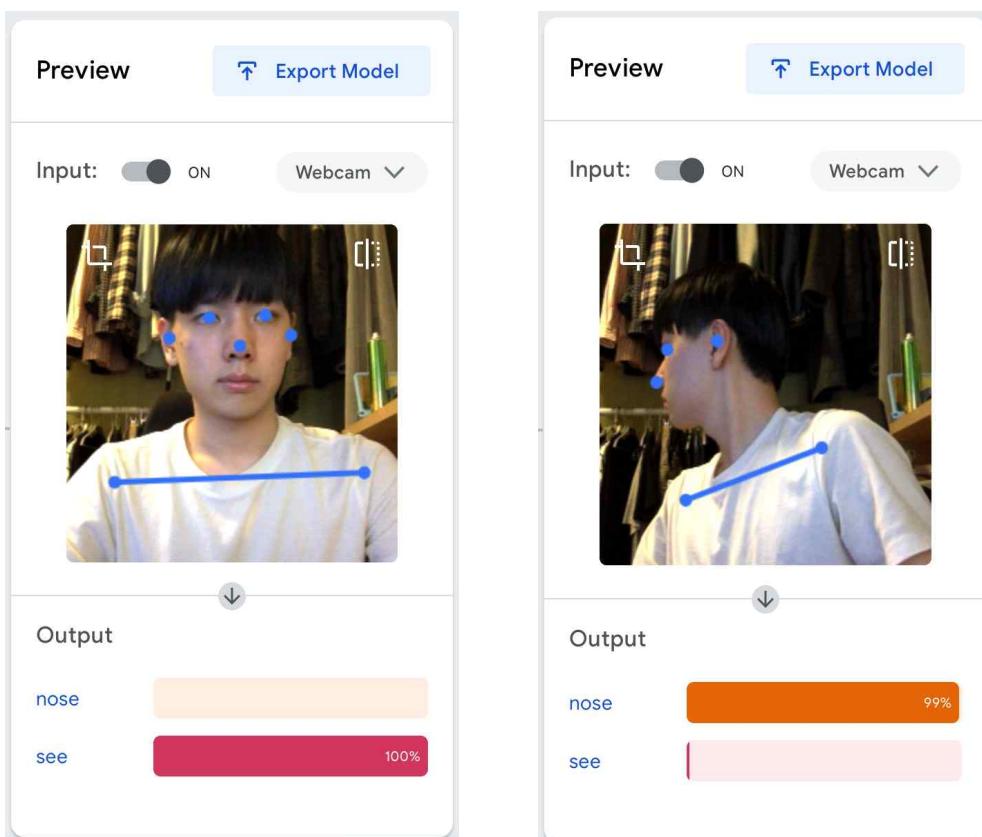
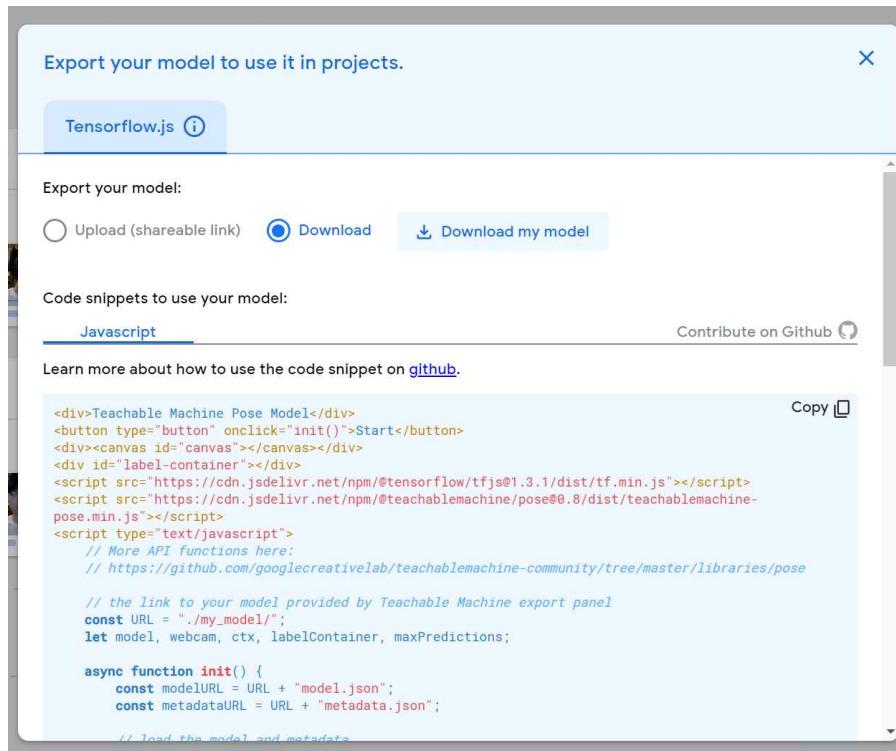


그림4, 5. 만들어진 결과 테스트.

4. 서비스 개발 과정(코드 설명 포함)



The screenshot shows the Teachable Machine web interface with a modal window titled "Export your model to use it in projects." The modal contains the following information:

- Model name: Tensorflow.js (with an info icon)
- Export options:
 - Upload (shareable link) (radio button)
 - Download (radio button, selected)
 - Download my model (button)
- Code snippets to use your model:
 - JavaScript (selected tab)
 - C++ (tab)
- Contribute on GitHub (link)
- Learn more about how to use the code snippet on [github](#).
- Code snippet content (JavaScript):

```
<div>Teachable Machine Pose Model</div>
<button type="button" onclick="init()">Start</button>
<div><canvas id="canvas"></canvas></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/pose@0.8/dist/teachablemachine-pose.min.js"></script>
<script type="text/javascript">
  // More API functions here:
  // https://github.com/googlecreativelab/teachablemachine-community/tree/master/libraries/pose

  // the link to your model provided by Teachable Machine export panel
  const URL = "./my_model/";
  let model, webcam, ctx, labelContainer, maxPredictions;

  async function init() {
    const modelURL = URL + "model.json";
    const metadataURL = URL + "metadata.json";
    // load the model and metadata
  }
</script>
```
- Copy (button)

만들어진 모델을 내보내기 기능을 이용하여 개발을 시작한다.

- 두 가지 기능을 구현하지만 방법은 동일하여 하나만 설명하겠습니다.
- 영상과 같이 HTML과 JavaScript를 활용했습니다.

```

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta name="Title" content="TM을 활용한 수업 진단">
  <meta name="keywords" content="TM, 온라인강의, 자택수업, 코로나, 교과서, 집중도">
  <meta name="description" content="간단하게 수업 집중도와 교과서를 확인해 보세요">
  <meta name="author" content="김지태">
  <meta property="og:title" content="TM을 활용한 수업 진단">
  <meta property="og:description" content="페이지 설명">
  <meta property="og:image" content="img.jpg" />
  <meta property="og:url" content="https://musing-joliot-d30067.netlify.com">
  <title>TM을 활용한 수업 진단</title>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>

```

head단에서는 만드는 서비스 설명 또는 정보를 기본적으로 입력을 해주었습니다.



그림6. head에 등록된 meta 정보가 적용 된 모습.

```

<div id = "header">
  <nav class="nav">
    <a href="Index.html" class="nav-item is-active" active-color="orange">집중도 확인하기</a>
    <a href="Second.html" class="nav-item" active-color="green">교과서 확인하기</a>
    <span class="nav-indicator"></span>
  </nav>
</div>

```

두 기능을 편하게 사용하기 위해서 nav로 메뉴 버튼을 만들어 전환을 쉽게 하도록 했습니다.

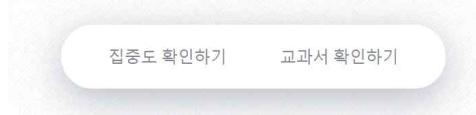


그림7. nav 생성된 모습

```
<button type="button" onclick="init()">Start</button>
```

버튼을 누르면 init 기능이 실행됩니다.

init 기능 속 내용

```
document.getElementById('name').innerHTML= "잠시만 기다려주세요. 시간이 걸립니다."
btndiv.style.display = "none";
```

init 기능 속에는 버튼을 누르면 알림 메시지와 버튼이 없어지는 기능을 구현하였습니다.

```
const modelURL = URL + "model.json";
const metadataURL = URL + "metadata.json";
모데들을 불러옵니다.
```

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

init 기능이 loop를 계속 실행하게 만들어 웹캠을 수시로 업데이트 합니다.

```
if(prediction[0].probability.toFixed(2) == 1.00){
    document.getElementById('name').innerHTML= "집중하세요!!"
} else if(prediction[1].probability.toFixed(2) == 1.00){
    document.getElementById('name').innerHTML= "잘하고 있군요"
}
```

업데이트를 하면서 모델과 웹캠을 비교를 실시합니다.

5. 실행화면캡처

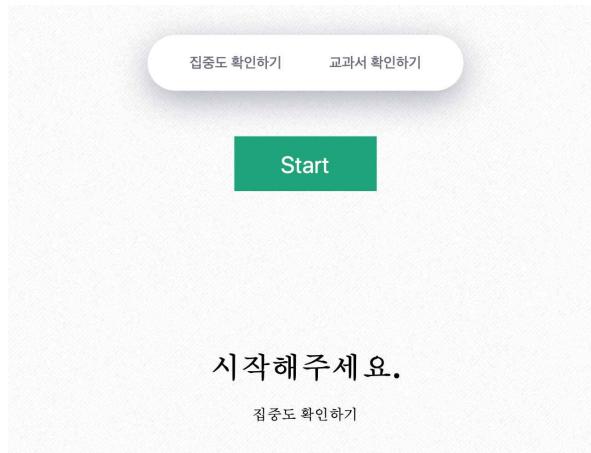


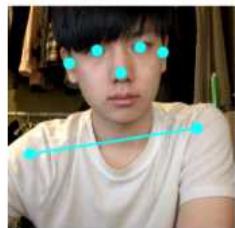
그림8. 서비스 메인 화면 모습입니다.



그림9. Start 버튼을 누른 후 모습.

집중도 확인하기

교과서 확인하기



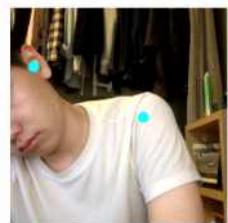
잘하고 있군요

집중도 확인하기

그림10. 집중도 테스트 모습 모범

집중도 확인하기

교과서 확인하기



집중하세요!!

집중도 확인하기

그림10. 집중도 테스트 모습 불량

집중도 확인하기

교과서 확인하기



교과서가 아닙니다.

교과서 확인하기

그림11. 교과서를 인증하는 모습.

집중도 확인하기

교과서 확인하기



교과서가 맞습니다.

교과서 확인하기

그림12. 등록된 교과서를 비교하는 모습.

6. 후기

Teachable Machine을 처음 봤을 때 아무 생각 없이 신기하고 재미있다고만 생각했다. 하지만 이렇게 무엇에 이용할 수 있을까 하면서 생각하고 만들어보고 꾸미니깐 남들이 보면 완벽한 서비스는 아니더라도 그럴싸한 서비스가 생긴 것 같아서 기분이 좋다. 어렵지도 않았고 아이디어만 있고 활용만 한다면 혼자서 재미있게 할 수 있을 것 같다.

-참고문헌(인터넷사이트, 유튜브URL, 책 등)

유튜브 조코딩 채널

<https://www.youtube.com/channel/UCQNE2JmbasNYbjGAcuBiRRg/about>

CSS 모음 사이트

<https://codepen.io/>

동적 웹페이지 운영 사이트

<https://www.netlify.com/>

html 메타 태그

<http://www.tcpschool.com/html-tags/meta>