

기계 학습을 활용한 자가 운동치료 웹 페이지

김혜리*, 김수빈*, 조민규**, 고희정**, 이형봉*

*강릉원주대학교 컴퓨터공학과

**강릉원주대학교 컴퓨터공학과 졸업

{khr6575, subinee21, 0215boc, tlahsh1240}@naver.com, hblee@gwnu.ac.kr

Self-exercise Therapy Web Page using Machine Learning

Hye-Ri Kim*, Su-Bin Kim*, Min-Kyu Cho**, Hee-Jung Kho**,
Hyung-Bong Lee*

*Dept. of Computer Science & Engineering, Gangneung-Wonju National
University, **Graduated

요 약

최근 코로나 19 상황으로 인해 많은 사람들이 모이는 병원 방문을 꺼리거나, 치료비에 부담을 느끼는 근골격계 재활 환자들이 많다. 이러한 환자들을 위해 이 프로젝트에서는 재활 치료 빈도가 높은 어깨와 손목 등 여섯 가지 근골격 부위의 자가 재활 치료를 돕는 기계 학습 기반 웹 페이지를 구현한다. 이 웹 페이지는 각 부위에 대한 재활 치료 자세를 구글 티처블 머신으로 학습 시킨 데이터를 기반으로 환자가 올바른 자세로 운동하는지를 판별해 준다. 이 때, 사용자의 재활 치료 자세는 웹 카메라로부터 캡처한다.

1. 서론

근골격계 재활 치료는 외부의 물리적인 힘이 필요한 경우도 있지만, 정확한 자세에 의한 근육 자극 운동으로 가능한 경우도 많다. 이 프로젝트에서는 어깨, 목, 허리, 발목, 손목, 무릎의 관절 재활 치료를 위한 자세를 구글 티처블 머신[1]으로 학습시킨 후, 이를 이용하여 환자의 재활 치료를 위한 정확한 운동 자세를 유도해주는 도우미 웹페이지를 개발한다. 그 외에 건강보험 심사평가원이 제공하는 공공 데이터를 이용하여 성별 연령대별로 자주 겪는 근골격계 질환에 대한 통계 정보도 제공한다.

2. 재활 치료 방법

환자(사용자)는 카메라가 부착되어 있는 노트북이나 태블릿 PC에서 웹 페이지에 접속하여 치료하고 싶은 부위를 선택한다. 그리고 나서 유튜브에 게시된 해당 부위의 재활 치료 자세에 대한 영상을 참조하여 웹 카메라 앞에서 치료 동작을 따라 하면 웹 페이지가 카메라 영상으로부터 환자의 자세를 인식하여 올바른 자세로 교정할 수 있도록 도움을 제공한다.

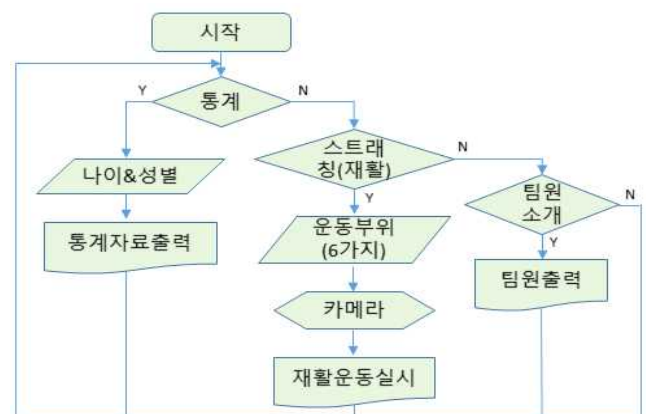
3. 웹 페이지 구현

3.1 기계 학습 데이터 획득

구글 티처블 머신[1]은 이미지, 오디오, 포즈 등 세 가지 모델에 대한 기계 학습 결과 데이터를 제공한다. 이 프로젝트에서는 세 가지 모델 중 포즈(Pose) 모델을 활용했고, 학습 데이터는 웹 캡으로 수집하였으며, 학습 결과는 Json 형식의 파일로 얻었다.

3.2 웹 페이지 전체 흐름도 및 구성

자바 스크립트를 이용한 재활 치료 웹 페이지의 전체 흐름도 및 구성은 그림 1~그림 5와 같다.



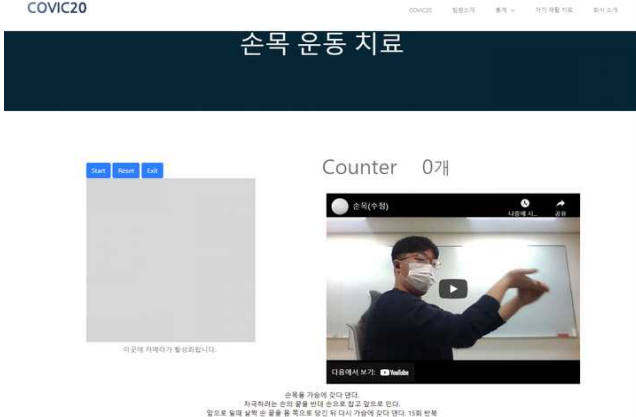
(그림 1) 웹 페이지 전체 흐름도



(그림 2) 전체 홈 페이지 구성



(그림 3) 재활 홈 페이지 구성



(그림 4) 재활 운동 페이지 구성



(그림 5) 통계 페이지 구성

3.4 참조하여 구현한 주요 코드

□ 개발 과정

웹 페이지는 Visual Studio Code 툴을 활용하여 제작하였고, 전반적인 틀은 템플릿 제공 사이트에서 적합한 디자인을 선택했으며, HTML5-CSS3-Java Script와 JQuery를 활용하여 Front-End를 개발하였다[2,3]. 여기에 간단한 PHP 코드를 도입하여 검색 기능에 활용했고, APM Monitor를 활용하여 웹 페이지 동작을 트레이스 하였다.

구글 티처블 머신에서 모션 데이터를 생성하고, 이들 모델 데이터를 구글 티처블 머신에서 제공하는 자바 스크립트 코드의 수정을 통해 동작 구분 시 참조하도록 하였다. 또한, 통계 자료는 SAS Studio를 활용하여 건강보험심사평가원의 15~18년도 근골격 질환 Big Data를 분석한 결과를 제공하도록 하였다.

□ 학습 모델 데이터 적재 및 웹 캠 초기화

그림 6의 init() 함수는 데이터와 모델을 저장해둔 경로에서 학습 데이터를 가져오며 판단 모델을 대기 상태로 선언하고 모델 클래스 배열의 최대 크기도 선언한다. 웹 캠의 사이즈를 정한 뒤 웹 캠을 활성화시키며, 여러 장의 사진을 연결하면 애니메이션 효과가 생기는 것처럼 window 객체의 애니메이션

```
const URL = "./my-pose-model(shoulder)/**";
let model, webcam, ctx, labelContainer, maxPredictions;

async function init() {
  const modelURL = URL + "model.json";
  const metadataURL = URL + "metadata.json";

  model = await tmPose.load(modelURL, metadataURL);
  maxPredictions = model.getTotalClasses(); //size of model Array

  const size = 500;
  const flip = true; // whether to flip the webcam
  webcam = new tmPose.Webcam(size, size, flip); // width, height,
  await webcam.setup(); // request access to the webcam
  await webcam.play();
  window.requestAnimationFrame(loop);

  // append/get elements to the DOM
  const canvas = document.getElementById("canvas");
  canvas.width = size; canvas.height = size;
  ctx = canvas.getContext("2d");
  labelContainer = document.getElementById("label-container");
  for (let i = 0; i < maxPredictions; i++) { // and class labels
    labelContainer.appendChild(document.createElement("div"));
  }
}
```

(그림 6) 학습 모델 데이터 적재 및 웹 캠 초기화

프레임 함수를 활용해 매 순간의 동작을 캡처하도록 한다. 이 부분에서 웹 캠의 활성화가 브라우저에 따라 실패하는 경우가 있어 고생했는데 이 프로젝트에서는 Chrome 브라우저를 사용했다.

□ 동작 예측(판별)

그림 7의 loop() 함수에서 timestamp를 따라 재귀 함수의 형태로 웹 캠을 업데이트하고 predict() 함수를 호출한다. predict() 함수에서 자세 변수와 횟수, 과정변수를 선언하고 prediction 변수에 데이터 모델 객체 배열을 선언하는데, 이는 곧 운동치료의 각 부분별 동작을 일컫는다. 위의 자료는 총 3개의 정확한 자세와 부정확한 자세 1개 등 총 4개의 모델 객체 원소로 이루어진 배열이다. 초기값 stand에서 자세를 취하면 해당 stand 변수의 값이 left, right로 변환한다. 다시 원래 자세를 취할 때 stand가 left, right일 경우 count를 1 증가시키고, count를 알리는 audio를 재생한다.

```

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

var progress = 327;
var status = "stand";
var count = 0;

async function predict() {
  const { pose, posenetOutput } = await model.estimatePose(webcam.canvas);
  const prediction = await model.predict(posenetOutput);
  if (prediction[0].probability.toFixed(2) > 0.90) {
    if (status == "left" || status == "right") {
      count++;
      var audio = new Audio(count % 10 + '.mp3');
      audio.play();
      progress = progress - 32.7;
      if (progress <= 0) {
        progress = 327 - 32.7;
      }
      $('.progress').css('stroke-dashoffset', progress);
      $('#counter').html(count);
    }
    status = "stand";
  } else if (prediction[1].probability.toFixed(2) == 1.00) {
    status = "left";
  }
  else if (prediction[2].probability.toFixed(2) == 1.00) {
    status = "right";
  }
  for (let i = 0; i < maxPredictions; i++) {
    const classPrediction =
      prediction[i].className + ": " + prediction[i].probability.toFixed(2);
  }
  // finally draw the poses
  drawPose(pose);
}

```

(그림 7) 웹 캠 영상 캡처 및 동작 판별

4. 시험

4.1 시험 환경

웹 캠이 고정으로 부착된 노트북에서 로컬 호스트 도메인으로 Chrome 브라우저를 이용해 시험을 실시하였다.

4.2 시험 결과

300회 학습시키고 100회 시험했을 때 어깨는 80% 전후의 정확도로 가장 높은 수치를 기록했으며 오류는 주로 긍정 오류가 발생되었다. 목과 허리는 동작 판별의 유연성을 보여주는 하나 어깨에 비해 정확도가 높지 않으며, 특히 일정 거리를 벗어나면 즉, 카메라와 너무 가깝거나 멀 경우 측정이 잘 되지 않는 현상이 발생되었다. 손목은 개발 과정에서는 가장 낮은 정확도를 보여주었으나, 동일한 치료 효과의 다른 동작으로 모델을 변경한 후에는 정확도가 대폭 증가했다. 무릎은 정면에서 수행할 때 60% 전후의 정확도를 보여주었으나, 어깨와 마찬가지로 긍정 오류가 높게 나왔다. 단순한 신체의 윤곽선으로 모델을 측정하다 보니 생긴 문제로 판단된다. 발목의 경우에는 가장 낮은 정확도를 보였으며, 모델을 업데이트해도 구분 동작의 차이가 불분명하다는 원인 즉, 재활운동이 왕복 동작으로 치료되는 것이 아닌 힘을 가중해서 치료하는 문제로 인해 이는 향후 보완사항으로 남겼다.

5. 결론

초반에 계획했던 대로 진행되지 않았던 점도 있었지만, 신속한 판단으로 다른 대안을 마련하여 문제를 해결함으로써 개발 전에 예상했던 것보다 더 높은 인식률과 정확성(60% 이상)을 가진 자가재활치료 웹 페이지를 개발하였다. 정확도를 향상시키기 위해 데이터 모델을 주기적으로 업로드하고, 각 부위별 1개의 동작에서 여러 개의 동작으로 증설시켜 개선해나간다면, 코로나 19 상황이나 경제적인 부담으로 재활치료를 받기 힘들었던 분들에게 이 웹 페이지가 많은 도움이 될 것으로 기대한다.

참고문헌

- [1] Google Teachable Machine, <https://teachablemachine.withgoogle.com/train>.
- [2] 고경희, Do it! 웹 프로그래밍을 위한 자바스크립트 기본 편, 이지스퍼블리싱, 2019.
- [3] 고경희, Do it! 자바스크립트 + 제이쿼리 입문, 이지스퍼블리싱, 2018.