

Transfer Learning 을 활용한 동작분류 모델의 구현과 활용

여대엽*, 류용환*, 유기쁨*, 조성배**

한동대학교 전산전자공학부¹

e-mail : id1423fo@gmail.com* snugbaejo@handong.edu**

Implementation and Application of Motion Classification Model using Transfer Learning

Tae-Yeob Yeo*, Yong-Hwan Ryu*, Gi-Ppeum Yoo*, Sung-Bae Jo**

Dept of Computer Science and Electronic Engineering, Han-Dong Global University

요 약

키보드와 마우스를 대체하기 위한 수단으로 기존에 사용되던 모션캡처 장비는 비용적 측면에서 일반 사용자들이 접근하기 어렵고 범용성이 낮다. 본 논문에서는 Transfer Learning 을 적용해 동작 분류 머신러닝 모델을 구현하고 그 활용 예시로 브라우저 동작을 제어하는 어플리케이션을 개발했다. 본 어플리케이션은 모션캡처 장비에 비해 상대적으로 저렴하며 보급률이 높은 웹캠과 머신러닝 모델을 사용해 브라우저를 제어한다. 해당 어플리케이션은 입력장치의 사용이 자유롭지 못한 환경에서 또는 입력장치의 조작이 어려운 장애인들이 PC 를 이용하는데 활용 가능하다. 또한 본 논문에서 사용한 동작인식 모델대신 손인식, 얼굴 인식 등의 모델로 대체함으로써 손 동작을 통한 컴퓨터 제어 등의 기술을 구현할 수 있게 될 것으로 기대한다.

1. 서론

키보드와 마우스 없이 동작만으로 컴퓨터를 조작하기 위해서는 그림 1 과 같이 레이저 센서나 적외선 센서가 탑재된 모션캡처 장비를 사용해야 한다. 하지만 대부분의 모션캡처 장비는 고가의 가격에 판매되고, 범용성과 보급률이 떨어진다는 단점이 있다. 이에 반해 웹캠은 대부분의 노트북에 내장되어 있어 보급률이 높고 휴대도 용이하다.



그림 1. 모션캡처 장비 Kinect XBOX 360



그림 2. 웹캠 EYE-2000

본 논문에서는 모션캡처 장치의 단점을 개선하기 위해 인공지능과 웹 기술을 융합하여 동작 분류 머신러닝 어플리케이션을 구현했다. 이 과정에서

Transfer Learning 기술을 활용해 적은 양의 데이터로 학습해야 하는 어려움을 극복했다.

또한 모델 활용의 예시로 웹캠으로 촬영한 동작을 통해 브라우저를 제어하는 어플리케이션을 개발했다.

2. 설계 및 개발환경

2.1 Image Classifier

(1) PoseNet

PoseNet 은 구글의 머신러닝 라이브러리인 Tensorflow.js 에서 제공하는 모델이다. Javascript 를 지원하기 때문에 웹 상에서 머신러닝 모델을 동작시키는 것이 가능하다.

PoseNet 은 입력된 영상, 또는 이미지로부터 눈, 코, 손목, 팔꿈치를 포함한 17 개 신체 부위의 좌표를 추출한다.

이 모델의 의의는 이미지에서 배경을 제거함으로써 촬영환경에 따른 동작분류의 오류를 줄이기 위함이다.

(2) MobileNet

MobileNet 은 이미지 분류 인공지능 모델로 모바일 환경에 적합하도록 개발되었다. MobileNet 은 크게 이미지로부터 특징을 추출하는 Feature Extractor(추출부), 특징에 따라 종류별로 분류하는 Feature Classifier(분류부) 로 구성된다.

¹ 이 논문은 과학기술정보통신부와 정보통신기술진흥센터의 소프트웨어중심대학 지원사업(2017-0-00130)의 지원을 받아 수행하였음

(3) Transfer Learning

Transfer Learning 은 특정 문제를 해결하도록 학습되어진 모델을 이와 비슷한 문제 해결에 재활용할 수 있게 학습하는 기술이다. 이 기술은 처음부터 모델을 학습하는 일반적인 학습방법과 달리 소량의 데이터와 적은 시간이 소요된다는 장점이 있다.

본 어플리케이션에서 사용한 MobileNet 은 ImageNet 에서 제공하는 이미지 데이터를 통해 학습되었으며 입력 받은 이미지를 1000 개의 라벨 중 하나로 분류한다.

상기 모델에 Transfer Learning 기술을 적용하여, 추출부를 그대로 사용하고 분류부는 재학습하여 input 이미지를 6 개의 동작으로 분류하도록 재구성했다.

2.2 Chrome Extension

Chrome Extension 은 구글에서 만든 웹 브라우저 확장 프로그램이다. 이 확장 프로그램을 사용하면 접속하는 모든 웹사이트에서 부가적인 기능을 수행할 수 있다. 본 논문에서는 Chrome Extension 을 사용함으로 동작인식 서비스의 범용성을 넓혔다.

3. 구현

3.1 시스템 구성도

본 어플리케이션의 핵심 구조는 그림 1 과 같다. 첫번째로, 웹캠을 통해 입력 받은 영상에서 PoseNet 을 통해 사용자의 신체부위의 좌표를 받아와 뼈대 이미지를 생성한다. 두 번째로, MobileNet 에서 뼈대 이미지를 입력 받아 동작을 분류한다. 마지막으로, 분류된 동작을 Chrome Extension 에서 받아 연동된 브라우저 기능을 수행한다.

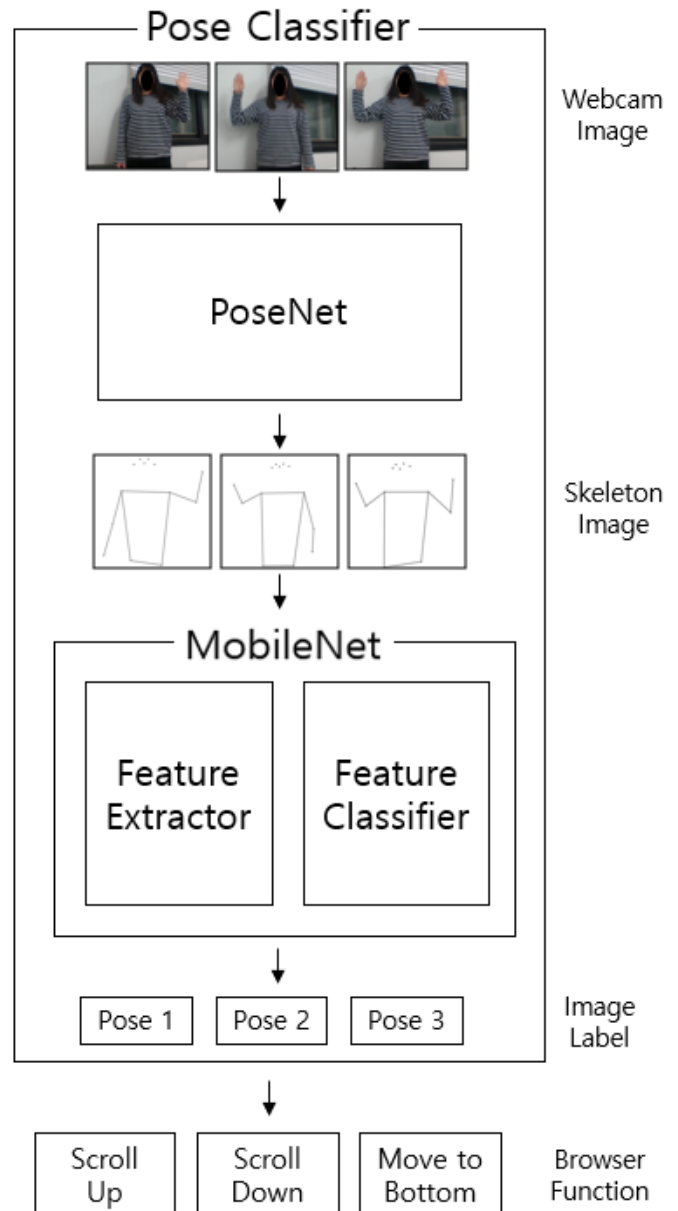


그림 1 - 시스템 구성도

3.2 User Interface

본 어플리케이션의 사용자 인터페이스는 크게 두가지가 있다. 첫번째는 사용자가 인식된 동작을 확인하게 해주는 거울 모드이다. 두번째는 어플리케이션의 세부 설정과 모델을 학습시킬 수 있는 설정 페이지다.

(1) 거울 모드

거울 모드는 그림 2 와 같이 입력된 영상과 뼈대 이미지를 동시에 보여준다. 거울 모드는 사용자가 자신을 보고 더 정교한 자세를 취할 수 있도록 도와준다.

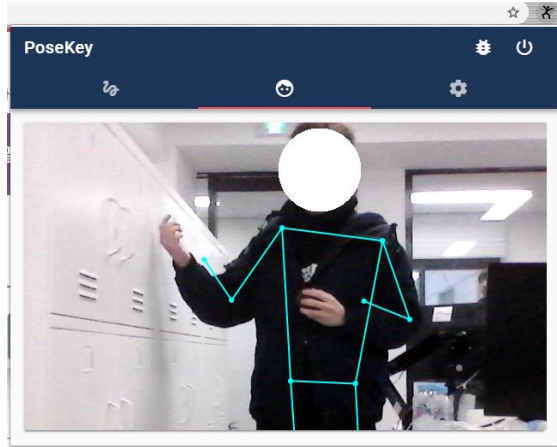


그림 2 - 거울 모드

(2) 설정 페이지

설정 페이지는 사용자가 어플리케이션에 대한 설정을 변경할 수 있는 화면이다. 그림 3 과 같이 사용자의 동작에 어느 브라우저 기능을 연동시킬 지 설정할 수 있다. 이러한 기능을 제공함으로써 사용자가 자신의 선호도에 맞는 동작을 브라우저 기능과 연동할 수 있다.

연동 가능한 브라우저 기능으로는 영상 제어, 웹 페이지 조회 관련 기능, 구글 슬라이드 제어 기능이 있다.

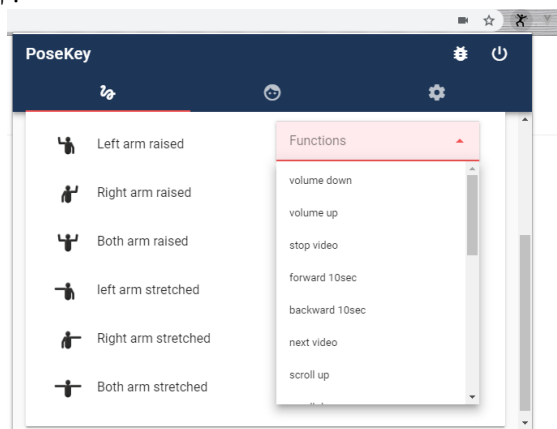


그림 3 - 설정 페이지 (기능 연동)

4. 결론

기존에는 키보드, 마우스와 같은 입력장치를 대신하기 위해 모션캡처 장비를 필요로 했다. 본 논문에서는 동작분류 기능을 대신하는 머신러닝 모델을 구현함으로써 웹캠만으로 컴퓨터 조작이 가능하게 했다. 또한 그 활용 예시로 동작 분류를 통해 브라우저를 제어하는 크롬 확장 프로그램을 구현했다.

본 어플리케이션은 물리적인 입력장치를 조작하기 어려운 상황이나 장소에서 활용 가능하다. 예를 들어, 대형 광고판과 같이 입력 장치를 배치하기 어려운 장소, 또는 발표, 운동, 요리 같이 손을 통해 입력 장치를 제어하기 어려운 상황에서 동작만으로 브라우저 조작이 가능하다. 마지막으로, 섬세한 마우스 조작에 어려움을 겪는 장애우분들이 PC 를 더

쉽게 사용할 수 있다. 위와 같이 본 어플리케이션은 사용자의 일상 생활에 많은 도움을 줄 수 있을 것으로 기대한다.

향후 결합 모델의 변경을 통해 신체 동작만이 아닌 얼굴 표정, 손 동작 등을 인식하여 다양한 프로그램을 구현하는 것이 가능할 것이다.

참고문헌

- [1] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, Trevor Darrell "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition" 2013.
- [2] Alex Kendall, Matthew Grimes, Roberto Cipolla "PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization" 2015.
- [3] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications" 2017.
- [4] Google Developers, "Chrome:What are extensions?", Retrieved from <https://developer.chrome.com/extensions>
- [5] Daniel Smilkov, Nikhil Thorat, Yannick Assogba, Ann Yuan, Nick Kreeger, Ping Yu, Kangyi Zhang, Shanqing Cai, Eric Nielsen, David Soergel, Stan Bileschi, Michael Terry, Charles Nicholson, Sandeep N. Gupta, Sarah Sirajuddin, D. Sculley, Rajat Monga, Greg Corrado, Fernanda B. Viégas, Martin Wattenberg "TensorFlow.js: Machine Learning for the Web and Beyond" 2019.