

보행 보조 로봇의 실시간 환경 인식을 위한 엣지 디바이스에서의 분류 네트워크에 관한 연구

신혜수¹, 이종원², 김강건²

¹이화여자대학교 휴먼기계바이오공학부

²한국과학기술연구원 지능로봇연구단

hyesoo030@gmail.com, jwlee@kist.re.kr, danny@kist.re.kr

A Study on Classification Network at Edge Device for Real-time Environment Recognition of Walking Assistant Robot

Hye-Soo Shin¹, Jongwon Lee², KangGeon Kim²

¹Dept. of Mechanical and Biomedical Engineering, Ewha Womans University

²Center for Intelligent and Interactive Robotics, KIST

요 약

보행 보조 로봇의 효과적인 보조를 위해서는 사용자의 보행 유형을 인식하는 것이 중요하다. 본 논문에서는 end-to-end 분류 네트워크 기반 보행 환경 인식 방법을 사용하여 사용자의 보행 유형을 강인하게 추정한다. 실외 보행 환경을 오르막길, 평지, 내리막길 3 가지로 분류하는 딥러닝 모델을 학습시켰으며, 엣지 디바이스에서 이를 사용하기 위해 네트워크 경량화를 진행하였다. 경량화 후 추론 속도는 약 47FPS 수준으로 실시간으로 보행 보조 로봇에 적용 가능한 것을 검증했으며, 정확도 측면에서도 97% 이상의 성능을 얻을 수 있었다.

1. 서론

보행 보조 로봇은 사람의 움직임을 보조하기 위한 기술로, 근력 보조 성능 극대화를 위해서는 사용자의 보행에 최적화된 보조가 필요하다. 이를 위해서는 실시간으로 사용자의 보행 유형을 인식하는 것이 중요하다.

지난 수년 동안 보행 보조 로봇을 위한 보행 유형 인식 방법들이 연구되어왔다. 대표적으로 보행 시 관절 각도에 따라 달라지는 센서 값을 이용해 보행 유형을 추정하는 방법이 있다[1]. 하지만, 이러한 방법은 실제 보행 상태가 변화하는 시각과 그 변화를 예측한 시간 사이에 차이가 발생할 수 있다는 단점을 가지고 있다. 그 외에도 근전도 센서를 사용하여 측정된 근육의 전기적 신호로 보행 유형 변화를 사전에 감지하는 방법이 존재한다[2]. 그러나, 보조 로봇 이외에 추가적인 센서를 부착해야 하며, 센서 부착 위치와 부착 위치의 상태에 따라 측정값이 달라지는 등의 문제점을 가지고 있다.

최근에는 보행 유형을 인식하기 위해 카메라를 사용하여 보행 환경을 인식하는 연구들이 수행되고 있다[3,4]. 기존의 보행 보조 로봇에 장착된 센서로부터 보행 유형을 추정하는 방법과는 달리, 카메라를 이용

하면 사용자의 보행 유형이 변할 수 있는 환경을 사전에 확인할 수 있어 보행 변화를 예상하고, 실시간으로 보조 방법을 전환할 수 있다. 이는 물리적으로 보행하기 이전에 환경을 인식하여 환경 변화에 따라 보행 방법을 변화시키는 사람의 보행 방식과 유사하다. 하지만, 대부분의 기존 연구는 엣지 디바이스가 아닌 GPU 와 같은 컴퓨팅 자원이 풍부한 컴퓨터에서 수행되고 검증되어왔다. 고성능의 컴퓨터에서 학습된 딥러닝 모델을 엣지 디바이스에서 사용하기 위해선 기존 모델의 성능을 유지하면서, 크기를 축소시키고, 연산을 간소화해야 한다.

이를 위해 본 논문에서는 보행 보조 로봇에 장착된 카메라를 이용하여 실외 보행 환경 데이터를 획득하고, 엣지 디바이스에서 추론 가능한 경량화된 네트워크를 제안한다. 또한, 엣지 디바이스와 컴퓨터 각각에서 학습된 네트워크 성능에 대한 비교 및 분석을 수행하여 실시간으로 보행 보조 로봇에 적용 가능한 것을 검증하였다.

2. 관련 연구

2.1 보행 환경 인식

보행 보조 로봇을 위한 보행 환경 인식 연구에서는

분류 네트워크를 이용한 보행 환경 분류 방법이 주로 연구되었다. Y. Qian et al.[3]은 RGB-D 카메라를 사용하여 보행 환경의 3차원 포인트 클라우드(point cloud)를 생성한 후 이를 2차원으로 투영하여 생성된 이진화 이미지로 네트워크를 학습시켜 보행 환경을 분류하였다. 이때, 허벅지에 위치한 두 개의 IMU 센서로부터 추정된 보행 위상 값을 함께 사용하여 보행 유형 인식의 정확도를 향상시켰다. B. Laschowski et al.[4]은 RGB 카메라를 사용하여 보행 환경을 분류하는 여러 네트워크에 대한 연구를 수행하였다. 보행 환경 이미지를 이용해 학습된 EfficientNet-B0[5], VGG16[6], MobileNetV2[7] 등 다양한 합성곱 신경망 구조(CNN architecture)의 성능을 비교, 분석하였다. 논문에서는 EfficientNet-B0 네트워크가 가장 좋은 성능을 보였고, VGG16 네트워크가 가장 빠른 추론 시간을, MobileNetV2가 최적의 정확도와 속도를 보였다.

2.2 경량 딥러닝

경량 딥러닝 기술은 경량 딥러닝 알고리즘과 알고리즘 경량화 기술로 나눌 수 있다[8].

경량 딥러닝 알고리즘은 알고리즘 자체의 크기와 연산량을 줄이면서 상대적으로 좋은 성능을 갖도록 설계된 알고리즘이다. 가장 많은 연산이 필요한 합성곱 연산을 채널 단위로 먼저 수행한 다음 한 픽셀 단위로 합성곱 연산을 수행하는 Depthwise Separable Convolution로 바꾸어 연산량을 줄이는 MobileNet[9], 직접 연결(skip connections)을 이용하여 모델이 출력과 입력의 차이를 학습할 수 있도록 하여 모델의 깊이가 깊더라도 쉽게 최적화할 수 있는 구조를 가진 ResNet[10] 등이 있다.

알고리즘 경량화 기술은 이미 생성된 알고리즘의 크기와 연산 수를 축소하는 것으로 모델 파라미터를 줄이는 모델 압축(Model Compression) 기법과 학습된 모델을 사용하여 새로운 모델을 학습하는 지식 증류(Knowledge distillation) 등의 기법이 있다.

3. 실외 보행 환경 인식을 위한 분류 네트워크

3.1 학습 데이터 수집 및 네트워크 학습

보행 환경 인식을 위한 분류 네트워크의 학습을 위해 도심에서 실외 보행 환경에 대한 RGB 이미지를 수집하였다<표 1>. 실외 보행 환경을 오르막길, 평지, 내리막길 3가지로 분류하였으며, 엡지 디바이스로 사용한 Luxonis의 OAK-D-IoT-40 모델의 카메라를 이용하여 이미지를 수집하였다. 카메라는 지면에서 약 1m 높이의 허리에 위치하며, 보행 방향을 향하게 하였다(그림 1).

분류 네트워크는 EfficientNet-B0[5]를 사용하였으며, 해당 네트워크의 완전 연결 층(Fully Connected Layer) 이후 sigmoid 함수를 추가하여 입력이 전체 유형 중 특정 클래스에 속할 확률이 아니라, 입력이 각각의 클래스일 확률을 예측하도록 하였다. 컴퓨터는 Intel Core i9-9900KF CPU 3.60GHz*16, 개발 환경은 Ubuntu 18.04.5 LTS, 그래픽 카드는 NVIDIA GeForce RTX 2080을 사용하여 개발하였다. 클래스 당 약 500 장의 이미지를 수집하였으며, 플립(flip)과 그레이스케일(grey scale)의 두 가지 증강 기법을 적용하여 각 클래스 별로 총 약 1500 장의 이미지를 획득하였고, 70%는 훈련용, 20%는 검증용, 그리고 10%는 시험용으로 사용하였다.

<표 1> 학습을 위해 수집한 RGB 이미지



(그림 1) 실험 디바이스 구성 및 배치

3.2 디바이스 및 분류 네트워크 경량화

이 연구에 사용된 엡지 디바이스는 Luxonis의 OAK-D-IoT-40 모델이다. 이 디바이스는 가로 60mm, 세로 45mm, 높이 25mm의 크기와 45.5g의 무게를 가진다. 컬러 카메라 모듈, 스테레오 카메라 모듈, Intel Movidius VPU를 탑재하여 최대 4TOPS(Tera Operations Per Second)의 처리 속도를 가진다. AI 모델의 처리 속도는 최대 1.4TOPS이다.

EfficientNet은 정확도를 높이기 위해 모델의 깊이, 너비, 입력 이미지의 크기를 조절하여 상대적으로 적은 파라미터로 좋은 성능을 보이는 경량 딥러닝 알고

리즘이다[5]. 깊이, 너비, 입력 이미지 크기에 따라 B0에서 B7까지 8개의 모델이 존재하며, 본 논문에서는 그중 가장 경량화 된 EfficientNet-B0를 사용하였다. 또한 알고리즘 경량화 기법을 사용하였으며, 경량화 기법 중 학습된 기존 신경망의 부동 소수점 수를 특정 비트 수로 줄여 파라미터의 불필요한 표현력을 줄이는 훈련 후 양자화(Post Training Quantization) 기법[11]을 적용하였다. 그 결과 32비트 부동 소수점으로 표현되던 가중치가 16비트 부동 소수점 표현 방식으로 변경되었다.

3.3 컴퓨터와 엣지 디바이스에서 네트워크 성능 비교

표 2는 각 디바이스의 정확도와 초당 추론 프레임 수(FPS: Frames per Second)를 나타낸다. 컴퓨터에서의 정확도는 약 97.8%, 엣지 디바이스에서의 정확도는 약 97.6%로, 분류 네트워크의 경량화 전후 정확도는 큰 차이가 없음을 확인할 수 있다. 이는 경량화 전 32비트의 부동 소수점으로 표현된 가중치와 경량화 후 16비트의 부동 소수점으로 표현된 가중치 값 사이에 큰 차이가 없기 때문으로 추정된다.

분류 네트워크가 보행 보조 로봇에서 실시간으로 작동할 수 있는지 확인하기 위해 초당 추론 프레임 수를 사용하여 비교를 진행하였다. 실외 보행 환경 데이터에 대해 인식을 수행하였을 때 컴퓨터와 엣지 디바이스 모두 실시간 성능(30FPS) 이상의 추론 속도를 확인할 수 있었다.

엣지 디바이스의 제한된 연산 성능에도 불구하고, 경량 딥러닝 알고리즘 사용 및 알고리즘 경량화 기술 적용을 통해 컴퓨터 연산 대비 큰 성능 저하 없이 실시간 추론이라는 목표를 달성할 수 있었다.

<표 2> 디바이스 별 분류 네트워크 성능 비교

디바이스	정확도(%)	FPS
컴퓨터		
Intel Core i9-9900KF CPU 3.60GHz*16	97.8	94.8
NVIDIA GeForce RTX 2080		
엣지 디바이스		
Luxonis OAK-D-IoT-40	97.6	46.8

4. 결론 및 향후 연구 계획

이 연구에서는 보행 보조 로봇의 효과적인 보조를 위해 장착된 카메라를 이용하여 실외 보행 환경을 인식하는 경량화된 분류 네트워크를 제안하였다. 또한, 보행 보조 로봇에서 사용할 수 있는 엣지 디바이스와 고성능의 컴퓨팅 자원을 가진 컴퓨터 각각에서 네트워크 성능을 비교 분석하여 실시간으로 보행 보조 로봇에서 환경 인식을 수행할 수 있음을 검증하였다.

그러나, 사용자가 특정 보행 환경에서 수행할 수 있는 보행 유형이 정해져 있지 않기 때문에 보행 환경을 인식하는 것만으로는 사용자의 보행 변화를 정확하게 예측하지 못할 수 있다. 따라서 향후 보행 환경 인식 방법과 함께 보행 보조 로봇에 장착된 센서로부터 보행 유형을 추정하는 방법 등을 서로 결합하여 보행 변화를 인식하는 연구를 진행할 예정이다.

참고문헌

- [1] Wang, J., Wu, D., Gao, Y. et al. "Integral Real-time Locomotion Mode Recognition Based on GA-CNN for Lower Limb Exoskeleton." *Journal of Bionic Engineering*, vol. 19, pp. 1359-1373, 2022.
- [2] Nasr, Ali, et al. "Myoelectric Control of Robotic Leg Prostheses and Exoskeletons: A Review." *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. ASME, Virtual, 2021, pp. V08AT08A043.
- [3] Qian, Yuepeng, et al. "Predictive locomotion mode recognition and accurate gait phase estimation for hip exoskeleton on various terrains." *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6439-6446, 2022.
- [4] Laschowski, Brokoslaw, et al. "Environment classification for robotic leg prostheses and exoskeletons using deep convolutional neural networks." *Frontiers in Neurobotics*, vol. 15, pp. 730965, 2022.
- [5] Tan, Mingxing, and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." *International conference on machine learning*. PMLR, Long Beach, California, USA, 2019, pp. 6105-6114.
- [6] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556*, 2014.
- [7] Sandler, Mark, et al. "Mobilenetv2: Inverted residuals and linear bottlenecks." *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, 2018, pp. 4510-4520.
- [8] Lee, YongJu, et al. "Recent R&D Trends for Lightweight Deep Learning." *Electronics and Telecommunications Trends.*, vol. 34, no. 2, pp. 40-50, 2019.
- [9] Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." *arXiv preprint arXiv:1704.04861*, 2017.
- [10] He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*, Las Vegas, NV, USA, 2016, pp. 770-778.
- [11] Krishnamoorthi, Raghuraman. "Quantizing deep convolutional networks for efficient inference: A whitepaper." *arXiv preprint arXiv:1806.08342*, 2018.