

Implementation of Virtual Touch Service Using Hand Gesture Recognition

A-Ra Cho[†] · Seung-Bae Yoo[†] · Byeong-Hun Yun[†] · Hyung-Ju Cho^{††}

ABSTRACT

As the need for hygiene management increases due to COVID-19, the importance of non-contact services is gaining attention. Hands, a tool for expressing intentions and conveying information, are emerging as an alternative to computer input devices such as the keyboard and mouse. In this study, we propose a method to address public health problems that arise when using unmanned ordering machines by controlling a computer using hand gestures detected through a camera. The focus is on identifying frequently used hand gestures, especially the bending of the index finger. To this end, we develop a non-contact input device using the MediaPipe framework and the long short-term memory (LSTM) model. This approach can identify hand gestures in three-dimensional space and provides scenarios that can be applied to the fields of virtual reality (VR) and augmented reality (AR). It offers improved public health and user experience by presenting methods that can be applied to various situations such as navigation systems and unmanned ordering machines.

Keywords : Camera-based Recognition, LSTM, MediaPipe, Contactless, Hand Gesture Recognition

손동작 인식을 이용한 가상 터치 서비스 구현

조 아 라[†] · 유 승 배[†] · 윤 병 훈[†] · 조 형 주^{††}

요 약

코로나19로 인해 위생 관리의 필요성이 높아지면서 비대면 서비스의 중요성이 주목받고 있다. 의사를 표현하고 정보를 전달하는 도구인 손은 키보드, 마우스 등 컴퓨터 입력 장치를 대체할 수 있는 도구로 떠오르고 있다. 본 연구에서는 카메라를 통해 감지된 손동작을 이용하여 컴퓨터를 제어함으로써 무인 주문기를 사용할 때 발생하는 공중보건 문제를 해결하는 방법을 제안한다. 자주 사용되는 손 제스처, 특히 검지를 구부리는 동작을 식별하는 데 중점을 둔다. 이를 위해 미디어파이프(MediaPipe) 프레임워크와 장단기 기억(long short-term memory, LSTM) 모델을 사용하여 비접촉 입력 장치를 개발한다. 이러한 접근 방법은 3차원 공간에서 손동작을 식별할 수 있으며, 가상현실(virtual reality, VR)과 증강현실(augmented reality, AR) 분야에 적용할 수 있는 시나리오를 제공한다. 내비게이션 시스템, 무인 주문기 등 다양한 상황에 적용할 수 있는 방법을 제시하여 향상된 국민건강과 사용자 경험을 제공한다.

키워드 : 카메라 기반 인식, LSTM 알고리즘, 미디어파이프(MediaPipe), 비접촉, 손동작 인식

1. 서 론

농림축산식품부가 실시한 음식점 경영 실태 조사에서 'POS(point of sale) 단말기 및 무인 주문기의 사용'에 대한 통계 자료에 따르면, 2018년 국내 음식점 산업에서 POS 단말기 및 무인 주문기의 사용이 0.9%이다. 최근에 무인 주문기 사용은 매년 약 1.4배부터 2배까지 증가하고 있다. 특히청 보도자료에 따르면 2018년 이후 무인 단말기 관련 특허 신청이 급증

했다는 것을 보여준다[1]. Fig. 1은 이러한 무인 단말기 관련 특허 신청이 빠르게 증가하는 상황을 보여준다. 2010년부터 2021년까지 특허 신청 건수를 살펴보면 2018년부터 신청률이 2배 이상 증가했다. 현대 사회에서 개인위생과 편리함을 높이기 위한 비접촉 서비스의 필요성이 증가했다. 2019년 말 코로나19의 발생으로 개인 방역의 중요성이 대두되면서, 비접촉과 비대면 서비스의 중요성이 증가했다. 이러한 변화에 부응하여 무인 단말기를 만지지 않고도 조작하고 정보에 접근할 수 있는 기술들이 필요하다.

인간에게 손은 의사 표현이나 개인의 위생에서 중요한 수단이고, 의사 표현에서 손을 효과적으로 사용해야 한다. 그러나, 공공장소에 설치된 입력 장치를 사용할 때 접촉 관련한 위생 문제와 오염 우려가 자주 발생한다. 이러한 문제를 해결하기 위해 비접촉 입력 방식이 대안으로 제시된다. 손을 장치에

※ 이 논문은 2024년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2020R111A3052713).

† 비 회 원 : 경북대학교 소프트웨어학과 학사과정

†† 정 회 원 : 경북대학교 소프트웨어학과 부교수

Manuscript Received : January 29, 2024

First Revision : July 15, 2024

Accepted : September 6, 2024

* Corresponding Author : Hyung-Ju Cho(hyungju@knu.ac.kr)

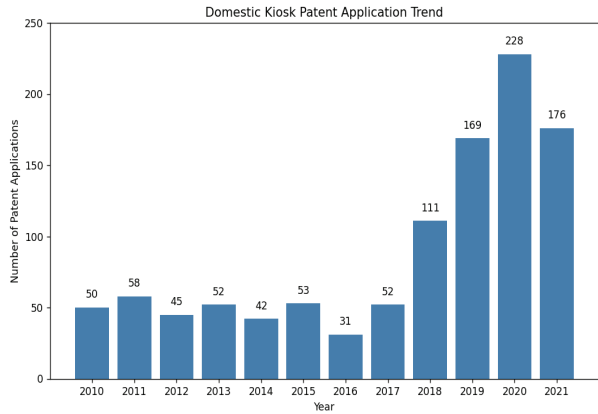


Fig. 1. Trends Trends in Patent Applications for Kiosks in Korea from 2010 to 2021.

접촉하지 않고 장치를 조작하고 상호 작용할 수 있게 함으로써, 편리함을 증가시키고 전염병 확산을 줄일 수 있다. 비접촉 입력 장치들은 사용자에게 유연성과 개인화된 경험을 제공한다. 비접촉 입력 장치에서 손을 자연스러운 입력 도구로 활용함으로써 공중보건 위생 문제를 개선할 수 있다.

본 연구에서 기존 가상 터치 기술 연구와 비접촉 입력 서비스를 조사했다. 예를 들면, 손의 위치를 파악하기 위해 깊이 카메라를 사용하여 하이터치(high-touch) 기술의 정확도를 높이는 연구가 수행되었다[2]. 깊이 카메라와 적외선 센서를 사용한 눈동자 추적 기술로 가상 터치를 구현한 연구가 있다 [3]. 이들은 별도의 맞춤형 장비를 요구하므로, 설치 비용이 비싸고, 범용성은 낮다. 반대로, 일반 카메라를 사용하여 가상 터치 기술을 구현한 연구들은 명확한 한계점이 있다. 첫째, 합성곱 신경망(Convolutional neural network, CNN)을 사용하여 손동작을 학습하고 검증한 연구 결과는 화면상의 아이콘을 터치하는 것과 같은 연속적인 움직임을 인식하지 못한다. 합성곱 신경망은 정지된 사진을 분석하는 모델로 가상 터치와 같은 시계열 이미지 처리에 적합하지 않다[7]. 둘째, 장단기 기억(long short-term memory, LSTM) 모델을 사용하는 연구들은 주먹을 쥐거나 엄지와 검지를 맞닿게 하는 인식, 완전히 열린 손이나 손을 펼친 상태 등의 손동작을 학습 데이터로 사용했다. 이것은 사용자들이 일반적으로 사용하지 않는 손동작을 이용하기 때문에 가상 터치 서비스에 적합하지 않다[3-6].

본 연구의 목표는 깊이 카메라나 적외선 센서, 프로젝터와 같은 맞춤형 장비 없이, 스마트폰이나 웹캠과 같은 일반 카메라와 손가락 인식 기술을 활용하여 가상 터치 서비스를 구현하는 것이다. 본 연구는 사용자들이 직관적으로 사용하는 손가락의 '포인터'와 '클릭' 등의 손동작을 학습해서, 범용성, 경제성과 접근성을 갖춘 가상 터치 서비스를 설계하고 구현하는 것이다. 논문의 구성은 다음과 같다. 2장은 배경지식을 소개한다. 3장은 본 연구에서 개발한 손동작 인식 및 가상 터치 서비스 구현 방법을 설명한다. 4장은 속도 개선, 기능 개선 방법과 실험 결과를 제공한다. 5장은 연구의 결론을 제시한다.

2. 배경지식

2.1 미디어파이프(MediaPipe)

미디어파이프는 추론 파이프라인을 통해 오디오나 비디오 같은 시계열 데이터 분석을 위한 딥러닝 구조이다. 손의 관절 점 추출과 함께 객체 인식, 이미지 분류, 얼굴, 손, 자세 감지 등 다양한 인식 기능을 제공한다[8]. 미디어파이프는 기계 학습을 지원하고 iOS, C++ 및 Python과 같은 다양한 프로그래밍 언어에서 사용할 수 있다. 미디어파이프가 제공하는 응용 프로그래밍 인터페이스(application programming interface, API)를 사용하여 동작 인식 애플리케이션 개발이 가능하다. 본 연구에서 손에 있는 관절 점과 손동작을 인식하기 위해서 미디어파이프 라이브러리를 사용한다.

2.2 장단기 기억(long short-term memory, LSTM) 모델

RNN(recurrent neural network)은 비디오, 오디오 및 텍스트와 같은 시계열 데이터를 처리한다. RNN은 인공 신경세포인 퍼셉트론(perceptron)의 순환 및 선택적 자체 피드백 연결을 활용하여 과거 이벤트에서 현재 처리로 데이터를 전파한다. 그러나 RNN은 기울기 소실 문제를 가지고 있다. 학습 중 과거 정보가 후속 출력에 과도하게 영향을 주는 문제를 해결하기 위해 장단기 기억 모델이 개발되었다. 장단기 기억 모델은 다중 게이트를 사용하여 1000개 이상의 시간 단계를 가지는 시퀀스에 대한 종속성을 효과적으로 학습한다[9]. 본 연구에서 장단기 기억 모델과 미디어파이프를 사용하여 연속적으로 손 랜드마크 좌표를 추출하고, 손가락으로 마우스 포인터(pointer), 마우스 클릭(click), 마우스 드래그(drag)를 모방하는 동작을 학습하고 처리한다.

3. 손동작 인식 및 가상 터치 서비스

3.1 손 객체 인식 시스템 구현

본 연구는 손 인식을 위해 광각 카메라를 사용한다. 광각 카메라는 넓은 시야를 제공하여 손동작 인식에 유리하다. 본 연구에서 개발한 손 객체 인식 시스템은 동영상 프레임에서 사용자의 한쪽 손만 추적하고 인식하도록 구현했다. 동영상에 여러 개의 손이 등장해도, 미디어파이프는 처음 인식된 손만 추적한다. 본 연구에서는 Fig. 2(a), 2(b), 2(c)와 2(d)에 설명된 '마우스 포인터', '마우스 좌클릭', '마우스 우클릭', '마우스 드래그' 네 가지 동작을 학습하고 인식한다. 인식된 마우스 좌클릭, 마우스 우클릭 동작은 마우스 클릭 이벤트를 실행시켜 관련 기능을 실행한다. 포인터 동작은 손 움직임과 동기화되는 마우스 포인터로 사용된다. 카메라와 화면 비율은 1:1로 설정한다. Fig. 3에서 보이는 검지의 끝부분(랜드마크 8번)과 마우스 이동을 동기화시킨다. 드래그 동작이 인식되면 시스템은 드래그 동작에 대한 이벤트 처리와, 이를 활용한 필기 기능을

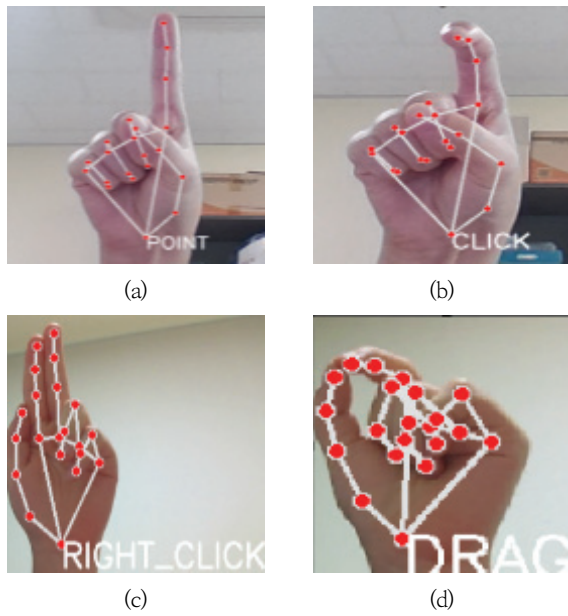


Fig. 2. Illustration of Gesture (a) Mouse Pointer (b) Mouse Left Click (c) Mouse Right Click (d) Mouse Drag

제공한다. Fig. 12는 손가락으로 공중에서 글씨를 쓰는 예제를 보여준다.

3.2 미디어파이프를 이용한 손 관절점 추출

본 연구는 손 관절점 추출을 위해서 미디어파이프 프레임워크를 이용한다. Fig. 3에서 보는 것처럼, 손 모델은 21개의 관절 지점(랜드마크)으로 구성된다. 미디어파이프가 손 랜드마크를 인식하면, 21개 손 관절의 x, y, z 좌표를 각각 제공한다. 미디어파이프로 추출한 21개 랜드마크를 이용하여 15개의 관절 사이 각도를 계산하여 손동작을 판별하는 학습 데이터로 사용한다. Fig. 4는 비디오 프레임에 대한 21개의 랜드마크 번호와 해당 좌표 추출 과정을 나타낸다. 미디어 파이프가 손 객체를 인식하고 손가락 관절에 해당하는 랜드마크 번호와 해당 좌표를 출력하는 것을 보여준다.

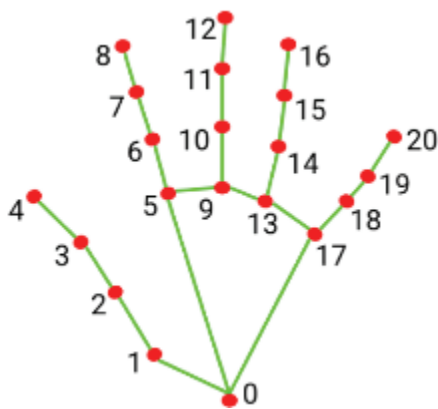


Fig. 3. Hand Landmarks Detection of MediaPipe

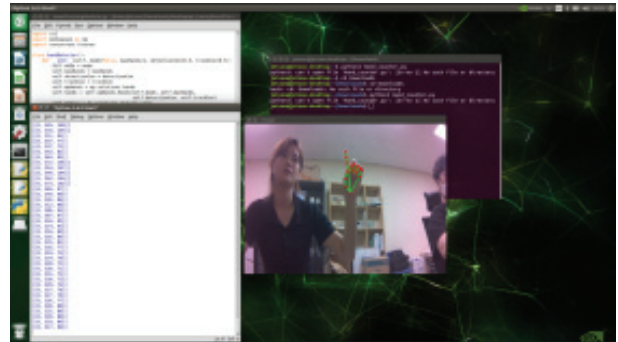


Fig. 4. Hand Object Recognition and Coordinate Value Output

3.3 입력 특징값 계산

3.2절에서 설명한 손 랜드마크에서 파생된 관절 각도는 손동작을 구별하는 학습 모델의 입력 특성이 된다. 손 움직임을 인식할 때, 관절 각도는 관절 위치 및 방향에 무관하게 손동작의 분류 정확도를 높인다. 관절 각도를 구하기 위해, 각 랜드마크 사이의 거리를 계산한다. 배열 v_1 에는 Fig. 3의 0번 인덱스부터 19번 인덱스의 랜드마크 좌표를, 배열 v_2 에는 Fig. 3의 1번 인덱스부터 20번 인덱스의 랜드마크 좌표를 넣는다. $v_2 - v_1$ 계산을 통해 각 랜드마크 사이의 거리 벡터가 계산되고, 이를 바탕으로 두 좌표 사이의 직선거리를 NumPy 라이브러리를 사용하여 계산한다. Fig. 5는 직선거리 공식을 사용하여 랜드마크 사이의 거리를 계산한 결과이다. 20개 거리(d_0 부터 d_{19} 까지)를 계산한다.

입력값을 정규화하기 위해, 각 랜드마크 벡터 v 는 단위벡터로 만들었다. 즉, $v = \frac{v_{i+1} - v_i}{\|v_{i+1} - v_i\|}$. 다음으로, 각 랜드마크 벡터의 사잇각을 계산한다. 2개 벡터의 사잇각은 두 벡터 v_i 와 v_{i+1} 의 내적을 이용하여 계산한다. 이를 위해 numpy 라이브러리인 아크코사인 함수(arccos)와 numpy.einsum 함수를 사용한다. 내적 값과 벡터의 크기를 이용하여 각도를 계산하는 식은 다음과 같다.

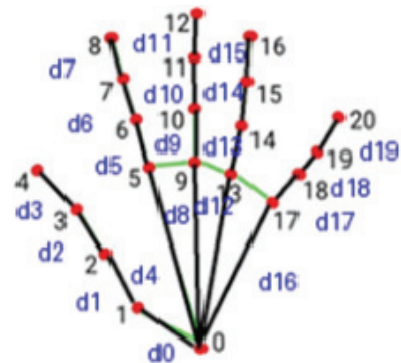


Fig. 5. Compute Distance Between Joints

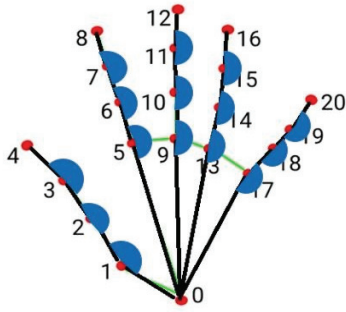


Fig. 6. Calculating Joint Angles

$$\theta = \arccos\left(\frac{v_i \cdot v_{i+1}}{\|v_i\| \times \|v_{i+1}\|}\right)$$

여기서 $v_i \cdot v_{i+1}$ 는 두 벡터의 내적이고, $\|v_i\|$ 와 $\|v_{i+1}\|$ 는 벡터의 길이이다. 따라서, 정규화된 벡터의 내적만으로 각도를 계산할 수 있다. 이렇게 구한 각도는 라디안 단위이므로, numpy 라이브러리의 degrees 함수를 이용하여 도(degree) 단위로 변환했다. 최종적으로 각 랜드마크 데이터와 계산된 사잇각을 결합하여 모델 학습에 사용하였다. Fig. 6은 관절 사이의 각도를 보여준다.

3.4 장단기 기억 모델을 이용한 학습

본 연구에서 손동작 인식을 위한 학습 모델로 Fig. 7에 제시된 LSTM 모델을 사용한다. 즉, 1개 장단기 기억(LSTM) 층과 3개 완전 연결(Dense) 층으로 구성된다. 각 프레임에 대한 입력 데이터는 3.3절에서 설명한 랜드마크의 사잇각을 포함하며, LSTM 층은 64개의 유닛으로 구성된다. 첫 번째 완전 연결 층은 32개의 출력을, 두 번째 완전 연결 층은 16개의 출력을, 세 번째 완전 연결 층은 구분해야 하는 손동작의 개수(N=4)를 출력으로 구성된다. 마지막 층에서 소프트맥스 활성화 함수를 사용하고, Adam 최적화기를 사용한다. 손실 함수로는 범주형 교차 엔트로피(categorical cross-entropy)를 사용한다.

```

model = Sequential([
    LSTM(64,
        activation='relu', input_shape=x_train.shape[1:3]),
    Dense(32, activation='adam'),
    Dense(16, activation='adam'),
    Dense(len(actions), activation='softmax')
])
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
model.summary()
    
```

Fig. 7. LSTM Algorithm Design

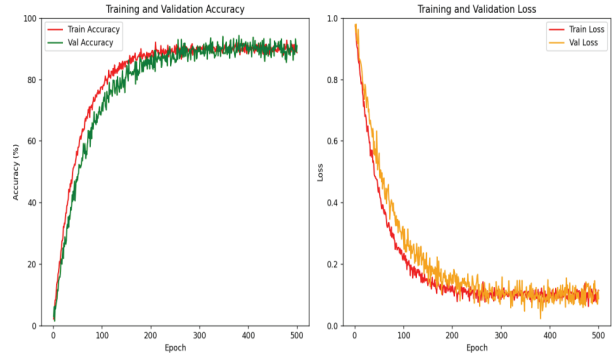


Fig. 8. Changes in the Accuracy and Loss of Training and Test Data

본 연구에서 입력 데이터는 5초 분량의 손동작을 녹화해서 초당 30개 프레임의 비디오 클립으로 구성된다. 입력 데이터의 특성은 3.3절에서 설명한 손 랜드마크의 좌표와 사잇각이다. 5명 학생들이 손동작 동영상을 생성했다. 4개 제스처(포인터, 왼쪽 마우스 클릭, 오른쪽 마우스 클릭, 마우스 드래그)에 대하여 각각 900개씩, 3,600개의 데이터 세트가 수집되었고, 데이터의 80%는 훈련 데이터로, 10%는 테스트 데이터로, 나머지 10%는 검증 데이터로 사용된다. 손동작 인식 학습은 구글 코랩에서 수행했고, 인텔(R) 제온(R) CPU @ 2.20GHz, GPU는 엔비디아 테슬라 T4를 사용했다.

모델 파라미터의 학습을 위해서 500 이포크(epoch) 동안 진행하였으며, 손실 값이 가장 작은 모델 파라미터 값을 저장했다. 저장된 모델의 학습 데이터 세트의 모델 정확도는 98%이고, 손실 값은 0.082이다. 또한 검증 데이터 세트에서 모델 정확도는 97%이고 손실 값은 0.11이다. Fig. 8은 이포크의 변화에 따르는 훈련 데이터와 테스트 데이터에 대한 모델 정확도와 모델 손실 값의 변화를 보여준다.

4. 속도 개선 및 기능 개선

4.1 장단기 기억 모델 경량화

본 연구에서 실시간성과 사용자의 사용 경험을 개선하기 위해 연산량을 줄일 수 있는 모델 경량화 방법을 사용했다. 이를 위해 텐서플로로 학습된 장단기 기억 모델을 텐서플로 모델 최적화 툴킷(toolkit)과 텐서플로 라이트를 사용하여 최적화하였다. 모델 경량화 기술은 (1) 가지치기, (2) 양자화, (3) 지식 정제, (4) 콤팩트(compact) 네트워크 설계로 4가지로 요약된다. Table 1에서 모델 경량화 기술들을 요약했다[10].

본 연구는 구글 텐서플로와 케라스(Keras) 모듈을 사용하여 양자화와 가중치 가지치기 기술을 적용했다. 가중치 가지치기는 텐서플로 모델 최적화 툴킷 함수를 사용했고, tfmot.sparsity.keras.ConstantSparsity를 사용하여 고정된 희소도 비율을 유지한다. 반복 실험을 통해 희소도 값이 0.6일 때 모델 크기를 줄이면서 최상의 성능을 유지할 수 있음을 발견했

Table 1. Classification of Model Lightweight Technology [10]

Classification	Technical summary	Approach
Pruning	Repeated learning by weight or neuron removal below a specified threshold.	Use pre-trained models
Quantization	Replace weights, active values, and slope (gradient) values with values expressed in lower bit widths	Use pre-trained models, or apply to model learning
Knowledge Distillation	Use the teacher model to learn the lightweight student model using the pre-trained models	Use pre-trained models
Compact Network Design	Approach to Modifying Neural Network Architecture for Improved Computational Efficiency	From the Onset of Model Training

다. 모델의 메모리 효율성을 높이기 위해 양자화를 수행하였다. 양자화는 모델의 가중치와 활성화 값을 더 적은 비트 수로 표현하여 모델 크기를 최소화하는 기술로, 텐서플로 라이트를 사용하여 구현하였다. 양자화 설정은 텐서플로 라이트의 Optimize 클래스를 사용했고, tf.lite.Optimize.DEFAULT 설정을 통해 모델의 크기와 정확도 손실을 최소화하였다.

최종적으로, TFLiteConverter 클래스를 사용하여 가중치 가지치기와 양자화가 적용된 텐서플로 라이트 모델을 생성하였다. 양자화는 모델 내 가중치와 활성화 함수의 정밀도를 감소시키는 방식으로 모델 크기를 줄인다. 가중치 가지치기는 중요도가 떨어지는 가중치를 제거하여 모델 파라미터 개수를 줄여서 메모리 사용량을 줄인다[11, 12]. 경량화 장단기 기억 모델의 파라미터를 최적화하기 위해 기존 모델에서 사용되었던 데이터로 재학습(fine-tuning)을 진행하였다. 경량화 장단기 기억 모델의 성능을 개선하고 최적화된 상태를 유지하기 위한 과정으로, 추가적인 학습을 통해 모델의 정확도와 효율성을 개선한다. 재학습 과정에서 학습률을 조정하고 조기 종료(early stopping) 기법을 사용하여 최적의 성능을 도출하였다. 이를 통해 경량화 장단기 기억 모델의 성능을 기존 모델과 유사한 수준으로 유지하고 연산 자원과 메모리 사용량을 줄였다. 재학습 후, 경량화 장단기 기억 모델은 검증 데이터 세트에서 예측 정확도가 91%, 손실 값이 0.187로 나타났다.

4.2 장단기 기억 모델과 경량화 장단기 기억 모델 비교

3장에서 설명한 장단기 기억 모델과 경량화 장단기 기억 모델의 성능을 정량적으로 비교하였다. 비교 항목으로는 정확도, 손실 값, F1 점수, CPU 사용량, 메모리 사용량을 포함한다. 이를 통해 경량화 장단기 기억 모델이 실시간 응용 프로그램의 효과를 평가한다.

Table 2와 3은 두 모델의 성능 비교 결과를 보여준다. 장단기 기억 모델의 정확도는 97%이며 F1 점수는 0.96이다. 경량화 장단기 기억 모델의 정확도는 91%로 감소했고, F1 점수는

Table 2. Comparison of the Two Models in Terms of Accuracy, F1 Score, and Loss

Model Classification	Accuracy(%)	F1 score	Loss
LSTM model	97	0.96	0.11
Lightweight LSTM model	91	0.89	0.187

Table 3. Comparison of the Two Models in Terms of CPU Usage, Memory Usage, and Processing Time

Model	CPU usage(%)	Memory usage(%)	Processing time(ms)
LSTM model	32	32	101
Lightweight LSTM model	25	24	53

0.89로 감소했다. 손실 값은 기존 모델이 0.11, 경량화 장단기 기억 모델이 0.187로 증가했다. 경량화 장단기 기억 모델은 연산량이 줄어들면서 예측 정확도가 떨어진다. 하지만 모델 계산량의 감소하면서 프로세스 처리 시간이 평균 101ms에서 53ms로 줄었다. CPU 사용량은 평균 32%에서 25%로, 메모리 사용량은 평균 32%에서 24%로 감소했다. 실시간 응용 프로그램은 사용자에게 빠른 응답 시간을 제공해야 한다. 모델 경량화는 사용자가 만족할 수 있는 빠르고 정확한 결과를 제공할 수 있다. 본 연구는 이러한 목표를 달성하기 위해 장단기 기억 모델에 경량화 기법을 적용하였다. 테스트는 AMD Ryzen 7600, 메모리 32GB, Windows 11 운영 시스템을 가지고 있는 컴퓨터에서 수행되었다. CPU 및 메모리 사용량과 처리 시간(Processing Time)은 같은 조건에서 15분 동안 총 6회 실행했고, Python의 time 함수를 사용하여 1분마다 평균을 계산했다.

4.3 필터링 방법의 성능 비교

실시간 손동작 인식 시스템에서 정확한 손 랜드마크 추출은 필수적이다. 그러나 실시간 동영상 스트리밍 환경에서 손 관절의 미세한 변동으로 인해 마우스 포인터의 위치가 고정되지 않는다. 즉, 손 관절의 미세한 움직임 때문에 손동작의 움직임을 정확하게 분류할 수 없다. 본 연구에서 다양한 필터링 방법을 비교하고 최적의 필터링 기법을 찾았다. 이를 통해 손 관절 점의 위치가 흔들리는 문제를 개선했다. 본 연구에서는 원유로 필터(one Euro filter), 평균이동 필터(moving average filter), 칼만 필터(Kalman Filter), 가우시안 필터(Gaussian Filter), 사비츠키-골레이 필터(Savitzky-Golay Filter)를 정량적으로 비교하였다. 필터의 성능 평가는 사용자가 특정 버튼을 클릭하거나 특정 좌표로 마우스를 이동시킬 때 사용자가 원하는 목표 좌표와 실제 마우스 포인터 좌표 사이의 차이를 측정해서 오차를 계산한다. 30회 반복 측정을 통해 오차를 기록하고, 오차들의 평균과 표준 편차를 계산한다. 필터 적용 전후의 손 랜드마크 위치 변동 크기를 조사했다. 또한, 특정 마

Table 4. Comparison of Filtering Methods for Landmark Positions

Filters	Standard deviation of Position (pixels)	Processing Time (ms)
None	5.2	53
One Euro filter	2.1	65
Moving average filter	2.8	58
Kalman filter	2.0	79
Gaussian filter	3.0	70
Savitzky-Golay filter	2.9	69

우스 이벤트를 처리하는 데 걸리는 평균 연산 처리 속도를 Python의 time 함수를 사용하여 측정했다.

Table 4는 필터들의 성능 비교 결과이다. AMD Ryzen 7600, 메모리 32GB, Windows 11 컴퓨터에서 수행하였다. 칼만 필터를 사용할 때, 2 픽셀 정도의 오차가 발생하면서 가장 좋은 성능을 보여주었다. 그러나, 칼만 필터를 적용했을 때, 계산이 26 ms 시간이 증가했다.

여러 필터링 방법을 비교했을 때, 원유로 필터와 칼만 필터가 마우스 포인터의 흔들림을 줄이는 데 가장 효과적이었다. 그러나 칼만 필터는 계산 복잡도가 커서 연산 처리 시간이 길어지는 단점이 있다. 따라서 본 연구에서는 계산 복잡도가 낮고, 성능이 만족스러운 원유로 필터와 평균 이동 필터를 사용하였다. 원유로 필터는 실시간 신호의 잡음을 줄이기 위해서 저역 통과 필터로, 두 개의 조정 가능한 파라미터를 사용한다. 원유로 필터는 각 시점에서의 신호 값을 지수 평활법을 사용하여 필터링하며, 평활 계수 α 는 신호의 변화율 정보를 사용해 동적으로 계산한다[13]. OneEuroFilter 라이브러리를 사용하였으며, min_cutoff와 beta 값을 각각 1.0과 0.5로 설정했다. 결과적으로 손의 미세한 움직임을 부드럽게 하여 흔들림을 감소시켰다. 손가락 관절 랜드마크 좌표 (x, y)에 대해 원유로 필터를 적용하여 잡음을 줄이고 손 랜드마크의 위치 떨림을 완화하였다.

평균이동 필터는 일정 시간 동안 손가락 관절 랜드마크 위치의 평균을 계산하여 랜드마크의 흔들림을 감소시키는 방법이다. 이 필터는 효과적으로 신호의 변동을 줄이는 데 유용하다. 본 연구에서는 슬라이딩 윈도우 크기 $N = 5$ 로 설정하였다. 평균이동 필터링은 각 랜드마크 좌표 (x, y)에 대해 최근 5개 프레임의 위치 데이터를 사용하여 랜드마크 위치의 평균을 계산한다. 최근 5개 위치 좌표의 평균은 랜드마크 위치의 변동을 부드럽게 만들고 안정적인 마우스 움직임을 제공한다. 원유로 필터와 평균이동 필터를 사용했을 때, 평균 연산 처리 속도가 67ms이고, 미세 떨림으로 인한 포인터 불안정 문제를 해결했다. 본 연구에서 마우스 포인터의 떨림 문제를 해결하기 위해서 원유로 필터와 평균이동 필터를 사용한다.

4.4 시연 결과

Fig. 9는 젯슨나노 시스템을 이용하여 개발한 손동작 인식 가상 터치 서비스 시연을 보여준다. 이 시스템은 사용자가 카



Fig. 9. Hand Gesture Recognition Using Jetson Nano



Fig. 10. The Technology Works in the Kiosk

메라 앞에 손을 대면 손동작 인식이 시작된다. 젯슨나노 시스템에 부착된 카메라가 손동작을 인식하여, 마우스 포인터를 움직이거나 클릭하는 이벤트를 발생시킨다. 이 기능을 사용하여 유튜브 동영상을 실행하는 시연을 보여준다. 가상 터치 서비스는 1개 손 객체를 인식하고, 사전 훈련된 4개 손동작(마우스 포인터 이동, 마우스 좌/우클릭, 마우스 드래그)을 인식하고 처리한다. 예를 들어, 사용자가 '포인터' 손동작을 모델이 인식하여 마우스 포인터를 위, 아래, 왼쪽, 오른쪽 어떤 방향으로든 이동한다. 원하는 위치에서 '좌클릭' 손동작을 실행하면 마우스 좌클릭 이벤트를 발생시킨다.

본 연구에서 개발한 손동작 인식과 가상 터치 서비스는 다양한 시나리오에서 사용할 수 있다. 리눅스, 윈도우, 스마트폰 운영체제에서 작동한다. Fig. 10과 같이 일반 모니터에 카메라를 장착한 젯슨나노 시스템을 연결하였다. 카메라가 없는 일반 모니터에 무인 주문기 메뉴를 실행하여 가상 터치 서비스를 통해 메뉴를 주문하는 서비스를 구현했다. 이것은 일반 모니터와 젯슨나노 시스템을 사용하여 무인 주문기 프로그램을 구현할 수 있음을 보여준다. Fig. 11에서 보는 것처럼, 카메라가 부착된 태블릿 컴퓨터에서 가상 터치 서비스를 구현했다. 태블릿 컴퓨터에서 기존 브라우저에서 이용하는 차량 길 안내 시스템에 접속하여 길 안내 프로그램을 실행한다. 태블릿 컴퓨터에서 가상 터치 서비스의 이용 가능성을 보여준다. 본 연구 결과를 활용하면 카메라가 부착된 컴퓨터에서 손 객체 인식 및 손동작 인식 서비스를 제공할 수 있다. Fig. 12에서 보는 것처럼, 드래그 이벤트를 통해 비접촉 방식으로 필기

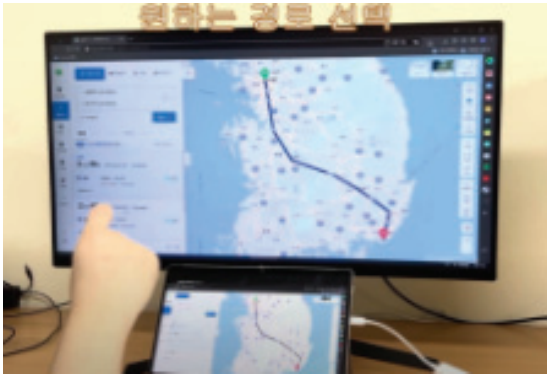


Fig. 11. Navigation Enabled with Applicable Technology

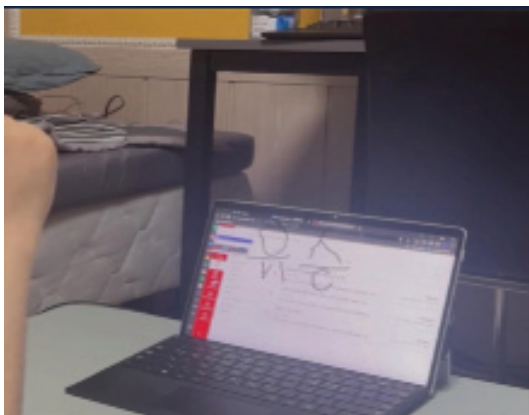


Fig. 12. Example of Writing in the Air with a Finger

할 수 있다. 이를 통해 사용자는 태블릿 컴퓨터나 스마트폰에서 공중에서 손동작으로 필기, 그림 그리기 작업을 수행할 수 있다. 손 글씨 작성은 교육, 엔터테인먼트, 업무 등의 다양한 시나리오에서 활용될 수 있다. 예를 들면, 태블릿 컴퓨터에 서명할 때, 비접촉으로 자신의 이름을 쓸 수 있다.

5. 결 론

본 연구는 가상 터치 서비스에서 일반적으로 사용되는 3차원 또는 깊이 카메라 없이 손동작을 인식하는 방식을 개발했다. 미디어파이프 프레임워크를 활용하여 손가락 관절점 위치를 추적하면서, 각 지점 사이의 각도를 계산한다. 딥러닝 학습과 예측을 위한 입력 벡터를 생성하기 위해 손동작은 5초마다 30개 프레임으로 캡처한다. 각 프레임의 입력 벡터는 손동작 학습을 위한 장단기 기억 모델에 제공된다. 본 연구에서 제한한 방법은 장치에 부착된 일반 카메라에서 작동하므로 특수 카메라가 필요하지 않다.

본 연구에서 경량화 장단기 기억 모델을 사용하여 계산량과 메모리 사용량을 줄이면서 실시간 응용 프로그램의 성능을 개선했다. 태블릿 컴퓨터 같은 성능이 낮은 하드웨어 환경에서 만족스러운 성능을 유지할 수 있도록 개선했다. 원유로 필

터와 평균이동 필터를 적용하여 손의 미세한 움직임으로 인한 손가락 관절 랜드마크 위치의 흔들림을 줄이면서, 마우스 포인터의 안정성을 개선했다. 손동작 인식의 정확도와 사용자의 비접촉 입력 장치 사용 경험을 개선했다.

본 연구의 활용 예시는 무인 주문기에 비접촉 방식으로 사용자가 주문할 수 있다. 카메라가 부착된 노트북 컴퓨터 또는 태블릿 컴퓨터에 애플리케이션을 설치하면 손동작 인식이 가능하고 가상 터치 기능을 사용할 수 있다. 본 논문은 기본 카메라와 장단기 기억 모델을 사용하여 가상 터치 기술의 구현 과정을 설명했다. 드래그 이벤트를 통한 실시간 필기 기능을 구현함으로써 가상 터치 기술의 실용성을 증대시켰다. 공중에서 글 쓰기 기능은 기존 연구와 차별화되는 부분으로, 추후 필기 인식을 활용하여 키보드의 역할을 대체할 수 있다. 추가 연구를 수행하여 실제 환경(예: 게임, TV, 발표)에서 사용할 수 있는 손 인식 및 가상 터치 서비스를 개발하는 것이 목표이다.

References

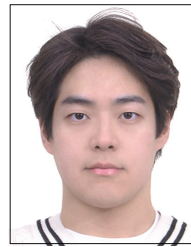
- [1] Unmanned terminals (kiosks) are spreading day by day, led by small and medium-sized businesses! [Internet], <https://rb.gy/74ce9d>
- [2] Digital worlds that feel human | Ultraleap [Internet], <https://www.ultraleap.com/>
- [3] VTOUCH [Internet], <https://vtouch.io/ko/products/virtual-touch-panel>
- [4] G.Y. Heo, M.J. Kim, B.D. Song, and B.J. Shin, "Hand expression recognition for virtual blackboard," *Journal of the Korea Institute of Information and Communication Engineering*, Vol.25, No.12, pp.1770-1776, 2021.
- [5] S. Y. Kim, S. J. Urm, S. Y. Yoo, S. J. Kim, and K. M. Lee, "Application of sign language gesture recognition using Mediapipe and LSTM," *Journal of Digital Contents Society*, Vol.24, No.1, pp.111-119, 2023.
- [6] Y. W. Park, W. J. Lee, M. S. Kim, M. J. Jeong, M. J. Kang, and S. H. Yeom, "Implementation of input device using motion and voice recognition," in *Proceedings of the Korean Society of Computer Information Conference, Daejeon*, pp.287-288, 2023.
- [7] Y. R. Kim, S. Y. Seo, and G. M. Park, "Touchless media art kiosks using gesture recognition," in *Proceedings of Korean Society of Broadcasting and Media Engineering Conference*, pp.232-235, 2022.
- [8] C. Lugaresi et al. "MediaPipe: a framework for building perception pipelines," *ArXiv abs/1906.08172*, 2019.
- [9] R. C. Staudemeyer and E. R. Morris, "Understanding LSTM - a tutorial into long short-term memory Recurrent Neural Networks," *ArXiv, abs/1909.09586*, 2019.

- [10] Deep learning model lightweight technology analysis [Internet], <https://rb.gy/xzis54>
- [11] X. Hu and H. Wen, "Research on model compression for embedded platform through quantization and pruning," *Journal of Physics: Conference Series*, Vol.2078, No.1, 2021.
- [12] Model optimization [Internet], <https://rb.gy/oio321>
- [13] G. Casiez, N. Roussel, and D. Vogel, "1€ filter: a simple speed-based low-pass filter for noisy input in interactive systems," In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp.2527-2530, 2012.



유 승 배

<https://orcid.org/0009-0005-3334-2220>
e-mail : e10013@naver.com
2024년 경북대학교 소프트웨어학과
학사과정
관심분야: Big Data Analysis, Machine Learning, Data Engineering



윤 병 훈

<https://orcid.org/0009-0006-5017-6399>
e-mail : gland45@naver.com
2024년 경북대학교 소프트웨어학과
학사과정
관심분야: Big Data Analysis, Machine Learning, Data Engineering



조 아 라

<https://orcid.org/0009-0007-9566-9549>
e-mail : zoanna5442@gmail.com
2024년 경북대학교 소프트웨어학과
학사과정
관심분야: Big Data Analysis, Machine Learning, Data Engineering



조 형 주

<https://orcid.org/0000-0001-7458-8888>
e-mail : hyungju@knu.ac.kr
2014년 ~ 현 재 경북대학교
소프트웨어학과 부교수
관심분야: Big Data Analysis, Machine Learning, Data Engineering