논문 17-42-09-12 The Journal of Korean Institute of Communications and Information Sciences '17-09 Vol.42 No.09
https://doi.org/10.7840/kics.2017.42.9.1796

초음파 환경에서의 사물인터넷 기기 제어를 위한 딥러닝 기반 동작 인식 시스템 구현

양 수 명•, 송 원 재*, 최 익 수*, 유 상 조°

Implementation of Deep Learning-Based Motion Classification System for IoT Device Control in Ultrasonic Sound Environments

Su-Myung Yang•, Won-Jae Song*, Ik-Soo Choi*, Sang-Jo Yoo°

요 약

음성 인식이나 제스처 인식 방법을 이용한 IoT 기기를 제어하는 방법은 많이 제시되었지만, 본 논문은 사물인터넷 기기를 제어하기 위해 딥 러닝을 적용한 동작 구별 방법을 제안한다. 초음파를 사용하는 스피커와 마이크로폰 사이에서 특정한 동작에 의한 도플러효과로 인해 생긴 미세한 주파수 변화를 이미지화하고 컨볼루션 뉴럴 네트워크(Convolutional Neural Network)를 통해 손동작을 구별한다. 구별된 각각의 동작 값을 MQTT(Message Queue Telemetry Transport) 프로토콜을 통해 원격으로 라즈베리 파이로 전송한다. 라즈베리 파이는 전송받은 동작 값을 통해 사물인터넷 기기들을 제어한다. 모의실험을 통해 사물인터넷 기기제어를 위한 새로운 제어방식 시스템의 성능을 확인하였다.

Key Words : Doppler effect, STFT, convolutional neural network, MQTT protocol, IoT

ABSTRACT

There are lots of IoT devices control methods using speech recognition or gesture recognition. In this paper, we propose a new gesture recognition method to control IoT devices using deep-learning technique. In the proposed method, we generate ultrasonic sound signals between speaker and microphone. During the generation of a sinusoidal wave signal, we can obtain a 2-dimension frequency-time domain data which captures doppler effect in accordance to specific gesture. In our system, we classify five different gestures using convolutional neural network (CNN). The classification index that is obtained by CNN is transmitted to the remote side raspberry pi device using MQTT protocol. The raspberry pi can control the IoT devices by the received data. Simulation results show that we can obtain the high classification accuracy for gestures and also can effectively control the IoT devices using gestures.

Ⅰ. 서 론

사람과 컴퓨터 간의 효과적인 상호작용을 하기 위해서는 두 개체 간의 의사를 잘 이해할 수 있는 편리하고 자연스러운 인터페이스가 요구된다. 원활한 상호작용을 위해 사용자의 직접적인 정보뿐만 아니라 경

※ 본 연구는 2017년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2017R1A2B4003512)
• First Author : School of Information and Communication Engineering, Inha University, syang0807@naver.com, 학생회원
° Corresponding Author : School of Information and Communication Engineering, Inha University, sjyoo@inha.ac.kr, 종신회원
* School of Information and Communication Engineering, Inha University
논문번호 : KICS2017-07-197, Received July 10, 2017; Revised August 9, 2017; Accepted August 11, 2017

험이나 지식에 기반을 둔 정보들에 대한 연구들이 활발히 진행되고 있다[1]. 사용자의 맥락정보를 활용한 인터페이스로는 음성[2], 촉감[3], 시각[4], 제스처[5]등이 있다.

제스처 인식은 다양한 손동작들을 인식하는 데 효과적이다. 제스처 인식에는 특수한 장갑을 이용한 기법과 마커를 이용하여 광학적으로 추적하는 기법이 있다[6]. 특수한 장갑을 이용한 제스처 인식은 높은 정확도를 나타내지만, 가격이 매우 비싸고 시스템과 유선으로 연결되어야 한다. 마커를 이용한 광학 추적기법은 적외선을 이용한 복잡한 시스템 구성이 필요하다. 따라서 기존의 특수한 장비나 고가의 카메라 및 센싱장비를 요구하는 제스처 인식시스템은 저가의 구현을 요구하는 생활환경에는 적용되기 어렵다. 그에 반해 음성 인식은 간단한 입출력 장치를 이용하여 쉽고 보편적으로 사용할 수 있는 상호작용 수단이다. 또한, 다른 인터페이스들과 함께 사용 가능한 장점도 있다[7]. 하지만 주변 소음에 취약하고 공공장소에서 사용하기 부적합하다는 단점이 있다.

사물인터넷(IoT: Internet of Things)에 대한 관심이 증가하고 사물인터넷 시장 가치가 부상함으로써 2020년에는 각종 센서와 네트워크 기능이 탑재된 사물인터넷 기기들이 260억 개에 이를 것으로 예상되고 있다[8]. 사물인터넷 기기들이 많아지면서 수많은 사물인터넷 기기들을 제어할 사용자들과 사물인터넷 기기와의 효과적인 상호작용방식 개발이 주목받고 있다. 인간의 가장 직관적인 상호작용방식은 제스처와 음성이고 이를 통해 사물인터넷 기기를 제어하려는 연구가 활발하게 이루어지고 있다[9].

본 논문에서는 사람의 귀로는 들리지 않는 비 가청 주파수 대역의 초음파 신호를 발생시키는 스피커와 간단한 마이크로폰을 이용하여 주변소음에 강하면서 공공장소에서 소리를 내지 않고 움직임만으로 다양한 IoT 장비를 제어할 수 있는 제스처 인식방법을 제안하고 이를 구현하였다. IoT 장비는 제스처 인식시스템과 직접 연결도 가능하고 클라우드 서버를 통해 제어정보를 원격의 IoT 단말에 전송할 수 도 있다. 본 논문에서는 원격 IoT 장비 제어를 위한 프로토콜을 구현하여 검증하였다.

제안하는 인식 방법은 도플러 효과와 컨볼루션 뉴럴 네트워크(CNN: Convolutional Neural Network)를 이용하여 제스처를 인식하고 구별하고 구별된 제스처를 이용하여 IoT 단말기기와 연결된 사물을 원격으로 제어한다. 일반적인 스피커를 사용해 공공장소 또는 소리를 낼 수 없는 공간에서도 사용할 수 있는 22kHz 대역의 비 가청 주파수 신호를 발생시키고 마이크로폰으로 신호를 녹음한다. 녹음되는 동안의 제스처에 의한 도플러 효과에 따른 주파수 변화를 이미지화하고 딥 러닝 사용하여 제스처를 구별한다. 구별된 제스처를 IoT 프로토콜을 이용해 서버로 Publish하고 IoT 기기가 Subscribe 하여 연결된 전자기기를 작동시킨다.

본 논문의 구성은 다음과 같다. Ⅱ장에서는 제안하는 시스템의 배경 이론을 소개한다. Ⅲ장에서는 스피커와 마이크로폰을 이용하여 제스처를 인식하는 방법을 소개하고 인식한 제스처를 이용해 원격으로 사물기기를 제어하는 방법을 제시한다. Ⅳ장에서는 제안한 상호작용 방식 분석결과와 구현한 사물인터넷 기기제어 모형을 제시하고 Ⅴ장에서는 논문의 결론을 맺는다.

Ⅱ. 구현을 위한 기반 기술 적용

2.1 도플러 이론

본 시스템의 이론은 1849년 “CHRISTIAN DOPPLER”에 의해 발견된 이론을 적용한 것으로서 접근해 오는 기차의 기적 또는 벨(Bell) 소리는 그 음조(Pitch)가 변화되는 것을 듣게 됨으로써 증명된다. 이 원리는 음원과 관측자와의 상대운동의 영향으로 주파수가 변하는 것으로 일반적으로 도플러 변화라고 말한다. 즉, 도플러 효과란 음파 또는 전파의 발생지/수신지가 다가오거나 멀어짐에 따라 겉보기 수신 주파수가 높아지거나 낮아지는 현상을 의미한다[10].

2.2 Short-Time Fourier Transform (STFT)

STFT는 분석할 신호에 윈도우 함수를 이용하여 일정 구간에 대해 DFT(Discrete Fourier Transform)를 수행한다. DFT 처리공식은 다음과 같다[11].

$$f_j = \sum_{k=0}^{n-1} x(k) e^{-\frac{2\pi i}{n} jk} \quad j = 0, ..., n-1 \tag{1}$$

DFT의 연산 속도를 높이기 위해 FFT(Fast Fourier Transform)를 수행한다. 이 데이터들을 누적시켜 시간에 따른 주파수 변화를 확인할 수 있다. STFT에 대한 식은 다음과 같다[12].

$$STFT_x(t,f) = \int_{-\infty}^{\infty} x(u) h^*(u-t) e^{-j2\pi fu} du \tag{2}$$

$h(u-t)$는 윈도우 함수를 의미한다.

STFT를 이용해 수신한 데이터를 주파수-시간 축에 대하여 2-D 이미지화한다. 그리고 STFT에서 일시적으로 나타나는 주파수 변화를 제거하고 평균적인 주파수 변화량을 보기 위해 잡음 제거(Noise Cancellation)를 수행한다. 이를 위해 윈도우 중첩 비율을 늘린다.

2.3 서포트 벡터 머신(Support Vector Machine; SVM)

서포트 벡터 머신[13]은 기계학습 알고리즘 중 지도학습(supervised learning)에 속하는 알고리즘으로, 각 클래스 간 거리를 최대로 하는 경계선 또는 경계면(hyperplane)을 찾는다. 그리하여 새로운 데이터가 들어 왔을 때 일반화 오류를 최소화하는 모델이다. 이때 각 클래스에서 데이터까지의 최소 거리를 마진(Margin), 그리고 경계선으로부터의 최소 거리인 데이터 벡터를 서포트 벡터(Support Vector)라고 한다. 두 클래스에 포함된 샘플들이 선형적으로 완전 분리가 어려운 경우에는 식(3)과 같이 오류에 대한 허용 변수를 적용한 비용함수를 통해 최적화가 가능하다. 이때 훈련 샘플들을 이용하여 마진의 최대화와 분류 허용에 대한 조절 인자인 C의 적절한 값을 결정해야 한다[14].

$$\Phi(w) = \frac{1}{2} \| w \|^2 + C\sum_{i=1}^{N}\zeta_i \tag{3}$$

식 (3)에 대한 최적화 문제는 라그랑지승수(Lagrange multiplier) 방법을 도입하여 해를 찾을 수 있는데, 분류하고자 하는 두 클래스가 비선형 분리면을 갖는 경우에는 커널함수 $K(x_i, x)$를 도입하여 경계면을 얻음으로써 선형적으로 분리할 수 있다. 커널함수를 이용할 경우 최종적으로 얻어지는 SVM분류기는 식(4)와 같이 표현되며, 이를 이용하여 비선형 분리면을 갖는 클래스를 분류할 수 있다[15].

$$f(x) = sign(\sum_{i=1}^{N}\alpha_{o,i}d_iK(x_i,x) + b_o) \tag{4}$$

여기서 $\alpha_{o,i}$는 식(5-1)으로 주어지는 목적함수를 최대화하는 라그랑지승수를 나타낸다.

$$Q(\alpha) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_jd_id_jK(x_i,x_j) \tag{5-1}$$

$$\sum_{i=1}^{N}\alpha_id_i = 0,\ 0 \le \alpha_i \le C \text{ for } i = 1,2,...,N \tag{5-2}$$

커널함수는 입력벡터를 고차원의 특징벡터로 변환한 후 내적을 구하는 과정을 수행하는 함수로 일반적으로 RBF(Radial Basis Function)가 많이 사용된다. RBF의 식은 다음과 같다[16].

$$k(x_i, x) = \exp(\frac{\| x - x_i \|^2}{\sigma^2}) \tag{6}$$

2.4 딥 러닝

딥 러닝은 여러 비선형 변환기법의 조합을 통해 높은 수준의 추상화를 시도하는 기계학습 알고리즘의 집합으로 정의되며, 큰 틀에서 사람의 사고방식을 컴퓨터에 가르치는 기계학습의 한 분야라고 이야기할 수 있다. 어떠한 데이터가 있을 때 이를 컴퓨터가 알아들을 수 있는 형태로 표현하고 이를 학습에 적용하기 위해 많은 연구가 진행되고 있으며, 이러한 노력의 결과로 CNN(Convolutional Neural Network), DBN(Deep Belief Network)와 같은 다양한 딥 러닝 기법들이 있다.

CNN은 컨볼루션 계층과 풀링 계층을 번갈아가며 수행함으로써 학습 데이터의 특징을 추출한다. 컨볼루션 계층은 이전 계층으로부터 필터를 일정 칸만큼 움직이면서 컨볼루션 연산을 수행한다. 여러 이미지에 대하여 특징이 추출되고 이것을 학습하는 원리이다. 데이터의 크기가 커지면 학습하는데 많은 시간이 소요되므로 풀링(pooling)을 사용한다. 풀링은 평균 풀링(average-pooling)과 맥스 풀링(max-pooling)이 있다[17]. 컨볼루션 계층과 풀링 계층을 통과하면 이미지를 벡터 형태로 변환하고 완전 연결 계층(Fully Connected Layer)을 통과한다.

완전 연결 계층은 가중치 행렬(weight)과 바이어스(bias)들로 이루어져 있다. 데이터는 가중치 연산과 활성화 함수(activation function)를 통해 다음 계층으로 이동한다. 활성화 함수로는 Sigmoid, Tanh, ReLu 등이 있다[18]. 그리고 학습 시 시스템이 과적합(overfitting) 되는 것을 방지하기 위해 드롭아웃(dropout)을 사용한다. 드롭아웃은 임의의 확률로 선택한 은닉 뉴런의 출력을 0으로 만들어 안정적으로 모델을 학습시킨다. 여러 계층을 통과한 데이터는 최종적으로 소프트 맥스 계층을 통과하여 클래스 별로 분류될 수 있다. 따라서 충분한 데이터를 이용하여 학습하면, 데이터에 대한 분류를 성공적으로 할 수 있다.

2.5 IoT 통신 프로토콜

본 논문에서는 사람과 사물인터넷과의 통신 프로토콜로 MQTT 프로토콜을 사용할 것을 제안한다. MQTT는 M2M(Machine-to-machine)과 IoT 환경 에서 사용하려고 만든 경량의 Publish/ Subscribe 메시징 프로토콜이다. 낮은 전력을 사용하거나 제한적인 낮은 통신 대역폭에서의 모바일 기기나 소형 디바이스들에 최적화되어 있다[19]. MQTT는 1999년 IBM이 주도하여 개발하였고 2013년에는 OASIS(Organization for the Advancement of Structured Information Standards)에서 표준화되었다.

MQTT는 메시지를 Publish하고, 관심 있는 주제를 Subscribe 하는 것을 원칙으로 한다. Publisher와 Subscriber는 Broker에 대한 클라이언트로 작동한다. Publisher는 관심 있는 주제에 메시지를 발행하기 위한 목적으로, Subscriber는 관심 있는 주제에 메시지를 구독하기 위한 목적으로 Broker 서버에 연결한다. Broker는 단일 프로토콜을 지원하는 경량의 Broker부터 다양한 표준 통신 프로토콜을 지원해주는 프로토콜까지 다양하다. 이 개체 간의 통신은 관심 주제, 즉 Topic이라는 문자열을 통해 송수신하게 된다. Topic은 슬래시(/)를 이용하여 계층적으로 구성할 수 있어 대량의 기기들을 효율적으로 관리할 수 있다.

MQTT는 3단계 QoS(Quality of Service)를 제공한다. QoS 0은 Fire and Forget과 같이 메시지를 한 번만 전달하고 전달 여부를 확인하지 않는다. QoS 1은 메시지는 반드시 한 번 이상 전달한다. 하지만 메시지의 핸드쉐이킹 과정을 세밀하게 추적하지 않으므로 중복으로 전송될 수 있다. QoS 2는 메시지를 한 번만 전달하고 핸드쉐이킹 과정을 추적한다. 따라서 높은 품질은 보장하지만, 성능의 희생이 따른다.

Ⅲ. 제안하는 동작 인식 시스템 및 IoT 단말 제어 방법

본 연구의 전체 시스템 구성은 그림 1과 같다. 스피커에서 초음파를 발생시키고 마이크로폰이 이를 수신한다. 스피커와 마이크로폰 사이에서 제스처를 취함으로써 주파수 변화가 나타난다. 이 특징들을 가진 수천 개의 데이터가 기계학습을 통해 컴퓨터에 인지시킨다. 이 데이터를 MQTT 프로토콜을 이용해 데이터를 Publish하고 라즈베리 파이가 이를 Subscribe 함으로써 제스처에 해당하는 IoT를 구동시킨다. 제안하는 시스템의 모듈별 데이터 및 신호정보 흐름도는 그림 2와 같다.

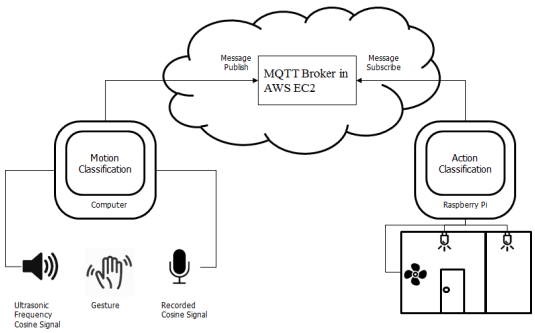

그림 1. 전체적 시스템 구성도
Fig. 1. System Model

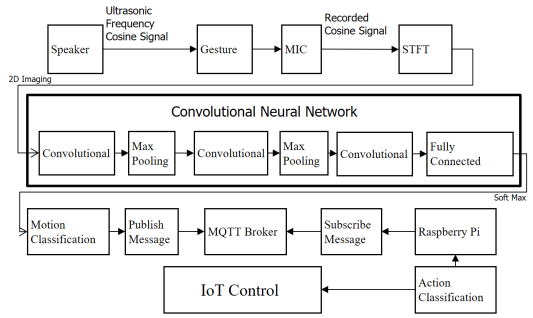

그림 2. 플로우 차트
Fig. 2. Flow Chart

3.1 도플러 효과를 이용한 학습데이터 수집

본 논문에서는 초음파를 발생시키고 연산을 위하여 MATLAB을 사용한다. 발생시키는 음파 신호의 제원은 표 1과 같다. 스피커에서 신호를 발생시키고 제스처를 통해 마이크로폰에서 수신된 주파수를 변화시킨다. 제스처를 수행하게 되면 도플러 효과가 나타나고 주파수의 변화는 다음과 같다.

$$f_r = f_t[\frac{c+v}{c-v}] \tag{7}$$

표 1. 시뮬레이션 파라미터
Table 1. Simulation Parameters

Simulation Parameters	Values
Generated Signal	cosine wave
Generated Frequency	22,000Hz
Sample Rate	45,056
Overlapped Ratio	80%
Period	2 s
Training Data	6,400
Validation Data	1,600
Test Data	1,600
Total Data	10,000

f_r은 마이크로폰에서 수신된 주파수이고, f_t는 스피커에서 발생한 주파수이다. c는 공기 중에서의 음속이고, v는 손의 속도이다. 손이 스피커 방향으로 이동하면 f_r은 증가하고, 반대로 이동하면 f_r은 감소한다.

제스처 인식에 사용되는 데이터를 수집할 때 간섭을 제어하기 위하여 두 가지 방법을 고려하였다. 첫째로, 여러 주파수가 혼재되어 있는 환경에서 효과적인 제스처 인식을 위해 간단한 센싱을 하여 지속적인 에너지가 검출되는 주파수는 회피하여 비 가청주파수를 송신한다. 본 연구에서는 22,000Hz의 주파수를 선택하여 송신하였다. 두 번째로, 잡음 및 주변의 간섭을 제거하기 위해 FFT에서 중첩기법을 사용하였고 이 방법으로 순시적인 간섭 및 잡음을 제거할 수 있다.

3.2 STFT를 이용한 시간-주파수 축 이미지화

도플러 효과에 의한 주파수 변화를 파악하기 위해 수신된 신호를 STFT를 하여 스펙트로그램을 그려서 시간에 따른 주파수 변화를 확인할 수 있다. 그림 3은

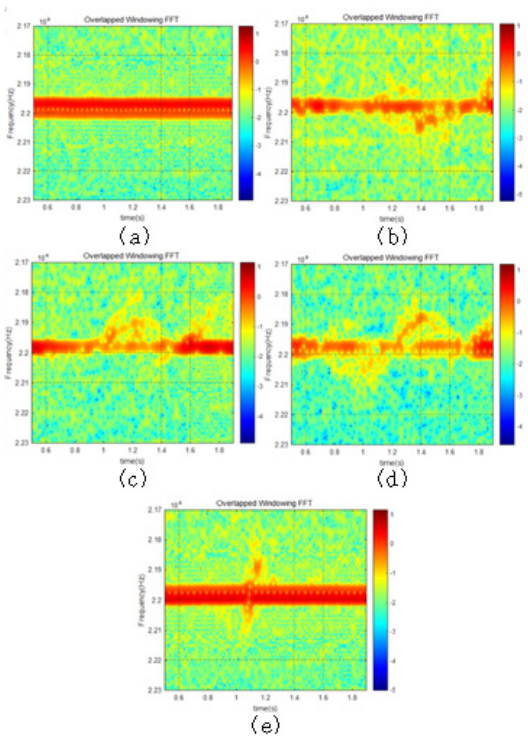

그림 3. 제스처에 따른 스펙트로그램 결과 (a) 제스처를 취하지 않았을 때 (b) 손을 스피커 쪽으로 움직였을 때 (c) 손을 마이크로폰 쪽으로 움직였을 때 (d) 손을 좌우로 흔들었을 때 (e) 순간적으로 위로 손을 움직였을 때

Fig. 3. Spectrogram Result Depending on Gesture (a) Do nothing (b) Move to speaker (c) Move to MIC (d) Swing (e) Up

손동작들에 따른 스펙트로그램의 결과를 보여준다. 아무런 제스처를 취하지 않았을 때 발생한 신호의 주파수인 22kHz 주변에서 선명하게 빨간색이 나타난다. 빨간색이 짙을수록 세기가 강함을 의미한다[20]. 하지만 스피커 방향으로 제스처를 취했을 때 마이크로폰에서 수신되는 주파수는 증가하게 되어 1초 부근에서 신호의 변화가 나타난 것을 알 수 있다. 마찬가지로 마이크로폰 방향으로 제스처를 취했을 때 주파수가 감소하게 되고, 스피커와 마이크로폰 사이를 흔드는 제스처를 취했을 때 주파수가 증가하였다 감소하는 것을 알 수 있다. 마지막으로 위로 빠르게 제스처를 취했을 때 짧게 주파수 변화가 나타난 것을 알 수 있다. 전반적으로 200Hz 증감이 나타났는데 이것은 제스처의 속도가 3m/s 정도이고 음속인 340m/s에 비해 약 1% 증감하기 때문이다. 이런 이유로 발생한 주파수 22kHz의 1%인 200Hz의 변화가 나타난 것을 알 수 있다.

주파수의 평균 변화량이 더욱 잘 나타나도록 만들기 위해 데이터의 80%를 누적하여 윈도우를 이동시켰다. 이후 딥러닝에 사용될 데이터를 더 잘 학습시키기 위하여 21,670Hz~22,330Hz에 해당하는 100x60 크기의 행렬을 추출하였고 10,000만큼 증폭시켰다. 5개의 제스처에 대해 2,000개씩 데이터를 수집하여 총 10,000개의 데이터로 학습하였다.

3.3 CNN을 이용한 데이터 구별

취득한 데이터를 학습시키기 위해 텐서플로우라는 파이썬 기반 오픈소스 라이브러리를 이용하였다. CNN 모듈은 컨볼루션과 맥스 풀링 연산이 번갈아가며 구성되어 있다. 필터는 총 24개를 사용하였다. 그림 4는 딥러닝의 전체 구성도를 나타내고 있다.

그림에서 첫 번째 층은 입력 층이고, 입력 데이터로 100x60의 2D 이미지를 사용한다. 두 번째 층은 컨볼루션 층이며, 3x3 크기의 6개 필터로 구성되어있다. 이미지와 필터를 컨볼루션 연산하여 98x58x6 크기의 데이터를 생성한다. 세 번째 층은 컨볼루션과 풀링 층으로, 상위 컨볼루션 층에서 획득한 데이터에 3x3 크기의 필터를 거친 후 2x2 맥스 풀링하여 48x28x6 크기의 데이터가 생성된다. 네 번째 층은 컨볼루션 층이고, 상위 풀링 층에서 출력되는 데이터와 3x3 크기의 12개 필터에 대해 컨볼루션 연산을 하여 46x26x12 크기의 데이터를 생성한다. 다섯 번째 층은 컨볼루션과 풀링 층으로, 상위 컨볼루션 층에서 획득한 데이터에 3x3 크기의 필터를 거친 후 2x2 맥스 풀링을 통해 22x12x12 크기의 데이터를 생성한다. 여섯 번째 층은

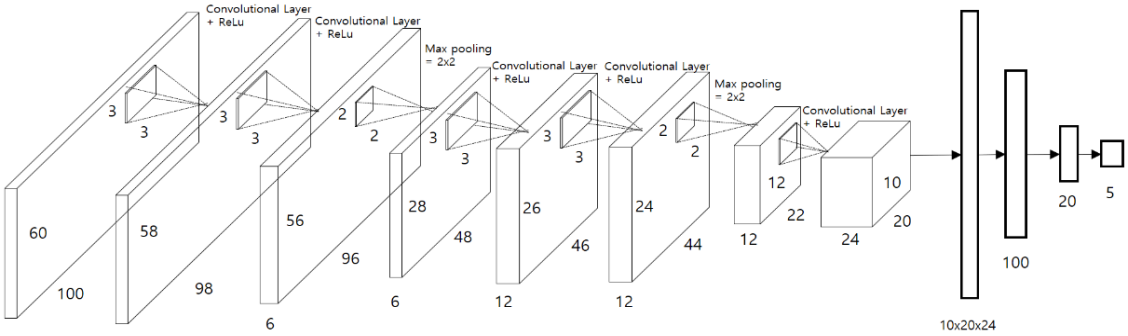

그림 4. 본 연구의 제스처 인식을 위한 컨볼루션 뉴럴 네트워크 구성도
Fig. 4. Convolutional Neural Network Architecture

컨볼루션 층이고, 상위 풀링층에서 출력되는 데이터와 3x3 크기의 24개 필터에 대하여 컨볼루션 연산을 하고 20x10x24 크기의 데이터를 생성한다.

여기까지가 CNN 모듈이고 출력된 4,800 크기의 데이터는 완전 연결 계층을 통과한다. 완전 연결 계층은 총 3개의 계층으로 구성되어있다. 첫 번째 층에서는 100개 노드로 구성되어 있으며, 두 번째 층은 20개, 세 번째 층은 총 5개의 제스처를 구별하기 위해 5개 노드로 구성되어있다. 첫 번째 계층에서는 입력받은 데이터와 100개 노드에 대해 가중치 연산을 수행한다. 두 번째 계층에서는 이전 계층에서 입력받은 데이터와 20개 노드에 대해 가중치 연산을 수행한다. 최종적으로 소프트맥스 계층을 통해 각각의 동작을 구별하게 된다.

각 Hidden Layer 층에서는 비선형 함수로써 ReLu를 사용하였다. 각 계층을 통과할 때 80% dropout을 통해 학습시켰고 learning rate는 0.001로 설정하였다.

과적합을 방지하기 위해 학습, 검증, 테스트 데이터를 3:1:1의 비율로 나누어 학습 비용과 검증 비용을 비교하였고 이는 그림 5와 같다. 그래프 상에서 step 250 이후로 검증 비용이 점차 증가하였다. 이것은 과적합을 의미하므로 이 지점의 weight, bias를 추출하여 MATLAB 상에서 동작을 구별하였다. 학습에 걸린 시간은 7,756초 걸렸다.

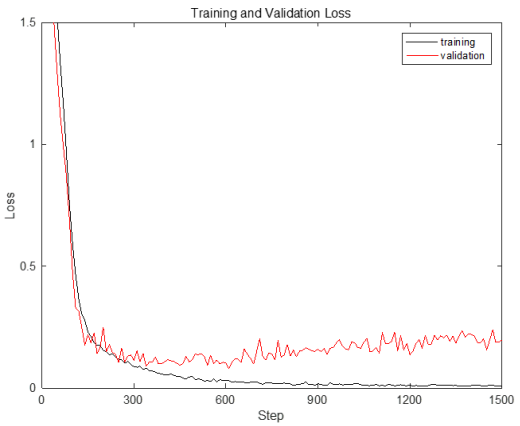

그림 5. 트레이닝과 검증 비용 비교
Fig. 5. Training Loss vs Validation Loss

3.4 MQTT 프로토콜 통신

컴퓨터에서 동작 인식을 완료하면 아마존 AWS의 EC2에 설치한 Mosquitto Broker에게 동작에 해당하는 Topic을 발행한다. 라즈베리 파이는 MQTT로부터 Topic을 구독하게 되고 각 Topic에 해당하는 표적기기를 동작시킨다.

3.5 IoT 단말 제어

본 절에서는 MQTT broker로부터 메시지를 구독하여 IoT 기기들을 제어하는 방법에 관해 설명한다. IoT 단말 제어 기기로는 라즈베리 파이3을 사용하였다. 그림 6은 사용하는 단말이 Broker에게 메시지를 구독하고, 구독한 메시지에 따라 메시지에 해당하는 기기를 작동시키는 흐름도이다. 라즈베리 파이에서 동작하는 수도코드는 그림7과 같고 해당 표적기기를 동작시키

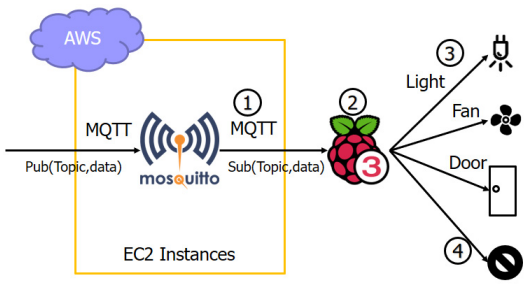

그림 6. IoT 단말기기 제어 흐름
Fig. 6. Flow of Control IoT Device

```
Repeat
    Subscribe Topic from Mosquitto Broker
    if message is affordable
        if value = '1'
            if White LED = 0
                White LED = 1
                Turn on White LED
            else
                White LED = 0
                Turn off White LED
        elif value = '2'
            if Yellow LED = 0
                Yellow LED = 1
                Turn on Yellow LED
            else
                Yellow LED = 0
                Turn off Yellow LED
        elif value = '3'
            if Fan = 0
                Fan= 1
                Turn on Fan
            else Fan= 1
                Fan = 0
                Turn off Fan
        elif value = '4'
            if Servo-Motor = 0
                Servo-Motor = 1
                Close Door
            else Servo-Motor = 1
                Servo-Motor = 0
                Open Door
        Default : Do Nothing

Until Raspberry Pi close
```

그림 7. 라즈베리 파이 수도 코드
Fig. 7. Raspberry Pi Pseudo Code

는 제스처 동작들은 그림 8과 같다. 스피커 방향으로 제스처를 취했을 때 White LED를 On/Off 시키고, 마이크로폰 방향으로 제스처를 취했을 때 Yellow LED를 On/Off 시킨다. 그리고 좌우로 흔들 때 Fan을 동작시키며 빠르게 위쪽으로 제스처를 취했을 때 서보 모터를 이용하여 문을 여닫도록 설계하였다.

그림 9는 IoT 기기 제어의 실제 모형이다. MATLAB의 GUI를 통하여 현재 어떤 제스처를 취하고 있는지 나타내고 해당하는 STFT를 그렸다. 스피커에서 초음파가 발생하고 스피커와 마이크로폰 사이에

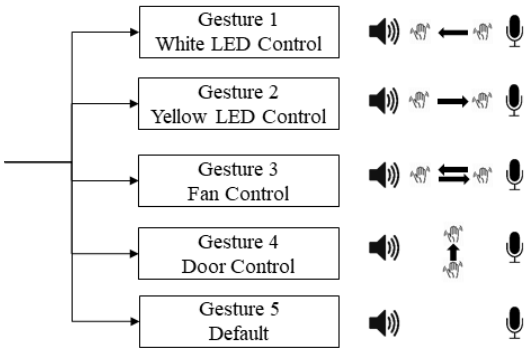

그림 8. 구독한 메시지의 동작 및 표적기기
Fig. 8. Gesture Classification & IoT Control using Raspberry Pi

그림 9. IoT 기기 제어 실제 모형
Fig. 9. IoT Control Model

서 제스처를 취함으로써 해당하는 IoT 기기를 동작시킨다.

Ⅳ. 실험 결과

본 절에서는 딥러닝을 이용한 초음파 환경에서 사물인터넷 기기를 제어를 위한 실험을 수행하였다. 스피커와 마이크로폰 사이에서 제스처를 취하고 각각의 제스처에 대해서 2,000개 데이터를 수집하였고 1,200개는 트레이닝, 400개는 검증, 400개는 테스트로 데이터를 나눴다. 트레이닝 데이터에서 600개는 음악 소음이 있는 환경에서, 600개 데이터는 소음이 없는 환경에서 데이터를 생성하였다. 소음이 있는 환경에서 해당하는 표적기기가 잘 동작함을 확인하였다. 최종적인 인식률은 표 2와 같다.

그림 10은 각 제스처에 대한 1차 컨볼루션 필터들의 출력 값이다. 제스처로 인하여 발생한 도플러 효과로 주파수가 변하고 이 효과를 필터가 극대화하여 다음 계층으로 값을 전달하는 것을 알 수 있다.

딥러닝의 컨볼루션 뉴럴 네트워크와 기계학습의 서포트 벡터 머신(Support Vector Machine)의 성능을 비교하기 위해 SVM을 이용하여 제스처를 구별하였다. 구현 환경은 scikit-learn의 오픈소스 라이브러리를 사용하였고 비선형 입력에 대해 좋은 결과를 얻기 위해 RBF(Radial Basis Function) 커널 함수를 사용하였다. 총 2,000개 데이터로 학습을 시켰고 2000개 데

표 2. 제스처 인식 정확도, 재현률, F1, 정확률
Table 2. Gesture Classification Precision, Recall, F-1 measure, Accuracy

	Precision	Recall	F-1	Accuracy
CNN	0.98	0.98	98.2	0.98
SVM	0.75	0.71	0.69	0.70

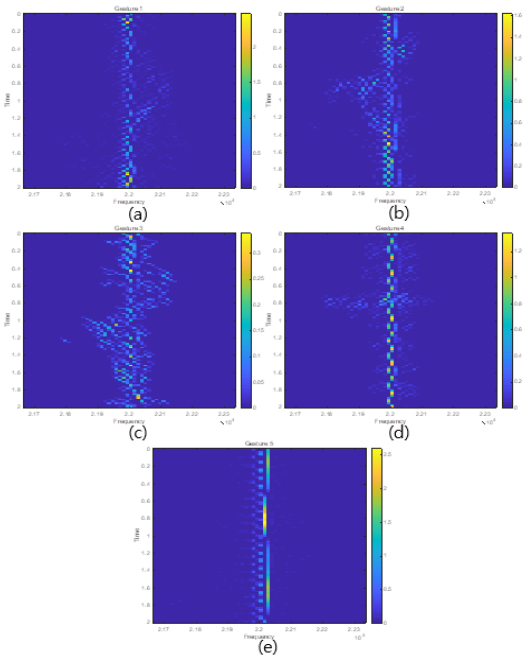

그림 10. 제스처에 대한 1차 컨볼루션 필터 출력값 (a) 제스처를 취하지 않았을 때 (b) 손을 스피커 쪽으로 움직였을 때 (c) 손을 마이크로폰 쪽으로 움직였을 때 (d) 손을 좌우로 흔들었을 때 (e) 순간적으로 위로 손을 움직였을 때
Fig. 10. First Convolution Filter Output for Gesture (a) Do nothing (b) Move to speaker (c) Move to MIC (d) Swing (e) Up

이터로 테스트하였다. 파라미터는 c=10, gamma=5e-05로 설정하였다. 트레이닝에 걸리는 시간은 40초로 CNN보다 빠른 것을 알 수 있다. 하지만 인식률은 CNN보다 약 27% 낮게 나타났다. 데이터가 많이 있을 때 딥러닝이 더 좋은 성능을 보인다는 것을 알 수 있다.

표 3, 4는 두 모델에 대한 제스처 타입 별 정확도를 나타낸다. 컨볼루션 뉴럴 네트워크의 경우 Gesture5의 재현율이 가장 높게 나타났다. 이에 반해 스피커로 이동 후 마이크로폰으로 이동하는 Gesture3의 경우 낮은 재현율을 나타낸다. 그 이유는 마이크로폰 방향으로 이동하는 제스처 Gesture2와 유사하므로 인식률이 가장 낮게 나타났다.

표 3. 컨볼루션 뉴럴 네트워크의 제스처 타입 별 정확률
Table 3. Gesture Type Accuracy of Convolutional Neural Network

	Gesture 1	Gesture 2	Gesture 3	Gesture 4	Gesture 5
Gesture 1	0.990	0.007	0.002	0	0
Gesture 2	0	0.98	0.002	0.012	0.005
Gesture 3	0	0.035	0.965	0	0
Gesture 4	0	0.005	0	0.987	0.007
Gesture 5	0	0	0	0.007	0.99

표 4. 서포트 벡터 머신의 제스처 타입 별 정확률
Table 4. Gesture Type Accuracy of Support Vector Machine

	Gesture 1	Gesture 2	Gesture 3	Gesture 4	Gesture 5
Gesture 1	0.592	0.292	0.027	0.032	0.055
Gesture 2	0	0.982	0	0	0.017
Gesture 3	0.135	0	0.840	0.005	0.02
Gesture 4	0.015	0.412	0	0.330	0.242
Gesture 5	0.01	0	0.18	0.005	0.802

서포트 벡터 머신의 경우 Gesture1의 일부분을 Gesture2로 인식하는 경우가 있고 Gesture4를 Gesture2와 Gesture5로 인식하는 경우가 있어서 낮은 정확률을 나타낸다.

V. 결 론

본 논문에서는 딥러닝을 적용한 초음파 환경에서 손동작을 이용한 IoT 기기 제어를 제안하였다. 소음이 있는 다양한 환경에서 취득한 음성 신호 데이터를 신호 처리를 통해 2D 이미지화 하였고 딥러닝을 통해 각각의 손동작을 높은 정확도로 구별하였다. 또한, 이 손동작으로 IoT 기기를 모방한 전구, 문, 선풍기들을 제어할 수 있음을 보였다. 이 기법을 응용하여 노트북 상에서 제스처를 이용해 앨범을 좌우로 넘기거나 독서실과 같은 소리를 낼 수 없는 공공장소에서 문을 여닫고 차 내부에서 운전을 하면서 기기를 제어하는데 사용할 수 있을 것으로 예상된다.

References

[1] D. P. Hong, “HCI survey based on gesture recognition,” *Telecommun. Rev.*, vol. 18, no. 3, pp. 403-413, Jun. 2008.

[2] A. Fernandez-Lopez, O. Martinez, and F. M. Sukno, “Towards estimating the upper bound of visual-speech recognition : The visual lip-reading feasibility database,” *2017 12th IEEE Int. Conf. Automatic Face & Gesture Recognition*, pp. 208-215, Washington, DC, DC, USA, Apr. 2017.

[3] J. R. Parker and M. Baumback, "Finger recognition for hand pose determination," *2009 IEEE Int. Conf. Syst., Man and Cybernetics*, pp. 2492-2497, San Antonio, TX, Oct. 2009.

[4] A. Eigenstetter, M. Takami, and B. Ommer, "Randomized max-margin compositions for visual recognition," *2014 IEEE Conf. Comput. Vision and Pattern Recognition*, pp. 3590-3597, Columbus, OH, Sept. 2014.

[5] G. Marin, F. Dominio, and P. Zanuttigh, "Hand gesture recognition with leap motion and kinect devices," 2014 *IEEE ICIP*, pp. 1565-1569, Paris, Oct. 2014.

[6] M. Panwar and P. Singh Mehra, "Hand gesture recognition for human computer interaction," *2011 Int. Conf. Image Inf. Process.*, pp. 1-7, Himachal Pradesh, Dec. 2011.

[7] A. Sears, J. Feng, K. Oseitutu, and C.-M. Karat, "Hands-free, speech-based navigation during dictation: difficulties, consequences, and solutions," *HCI J.*, vol. 18, no. 3, pp. 229-257, Nov. 2003.

[8] C. S. Pyo, H. Y. Kang, and N. S. Kim, "IoT(M2M) Technology and Development," *KICS Inf. and Commun. Mag.*, vol. 30, no. 8, pp. 3-10, Jul. 2013.

[9] X. Han and M. A. Rashid, "Gesture and voice control of internet of things," 2016 *IEEE ICIEA*, pp. 1791-1795, Hefei, Oct. 2016.

[10] S. Garg, R. K. Singh, R. R. Saxena, and R. Kapoor, "Doppler effect: UI input method using gestures for the visually impaired," *2014 TIIEC*, pp. 86-92, Bangalore, Apr. 2014.

[11] J. K. Kim, "Algorithm about improving calculation time of DFT and noise cancellation of EKG," *Korean Inst. Inf. Technol.*, vol. 14, no. 1, pp. 93-98, Feb. 2016.

[12] H. S. Lee, M. K. Kang, and K. T. Moon, "Implementation of spectrum sensing module using STFT," *The Korea Contents Assoc.*, vol. 10, no. 1, pp. 78-86, Jan. 2010.

[13] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. 5h Annu. Workshop on Computational learning theory,* pp. 144-152, Jul. 1992.

[14] J. H. Park, C. S. Hwang, and G. S. Bae, "Analysis of target classification performances of active sonar returns depending on parameter values of SVM kernel functions," *J. KIICE*, vol. 17, no. 5, pp. 1083-1088, May 2013.

[15] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Trans. Neural Networks,* vol. 13, no. 2, pp. 415-425-194, Mar. 2002.

[16] K.-R. Müller, G. Rätsch, S. Sonnenburg, S. Mika, M. Grimm, and N. Heinrich "Classifying drug-likeness' with kernel-based learning methods," *J. Chemical Inf. and Modeling*, vol. 45, no. 2, pp. 249-253, Apr. 2005.

[17] M. Frigo and S. G. Johnson, "The design and implementation of FFTW3," in *Proc. IEEE*, vol. 93, no. 2, pp. 216-231, Feb. 2005.

[18] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML'13*, Atlanta, GA, USA, Jun. 2013.

[19] M. Schmidt and R. Obermaisser, "Middleware for the integration of Bluetooth LE devices based on MQTT and ISO/IEEE 11073," *2017 IEEE CCECE*, pp. 1-4, Windsor, ON, Canada, Jun. 2017.

[20] K. Zhang, S. Zhang, and Y. Liu, "Using technology of short time fourier transform and filtering to improve the performance of prony algorithm," *Inf. Technol. J.*, vol. 10, no. 8, pp. 1545-1553, Jun. 2011.

양 수 명 (Su-Myung Yang)

2011년 3월~현재 : 인하대학교 정보통신공학과(공학사)

<관심분야> 신호처리, 인공지능, 무선통신

송 원 재 (Won-Jae Song)

2011년 3월~현재 : 인하대학교 정보통신공학과(공학사)

<관심분야> IoT 응용프로그램, 무선통신

최 익 수 (Ik-Soo Choi)

2016년 2월 : 경상대학교 정보통신공학과(공학사)

2016년 3월~현재 : 인하대학교 정보통신공학과(공학석사)

<관심분야> 통신공학, 인공지능, 무선통신, Cognitive Radio Network

유 상 조 (Sang-Jo Yoo)

1988년 2월 : 한양대학교 전자통신학과(공학사)

1990년 2월 : 한국과학기술원 전기및전자공학과(공학석사)

2000년 8월 : 한국과학기술원 전자전산학과(공학박사)

1990년 3월~2001년 2월 : KT 연구 개발 본부

2001년 3월~현재 : 인하대학교 정보통신공학과 교수

<관심분야> 무선 네트워킹 프로토콜, Cross-layer 프로토콜 설계, Cognitive Radio Network, 무선 센서네트워크, 미래인터넷