

<https://doi.org/10.7236/JIIBC.2022.22.6.121>

JIIBC 2022-6-18

# 스마트 미러간 화상 통화와 메시징 기능을 가진 CoMirror 시스템 구현

## Implementation of CoMirror System with Video Call and Messaging Function between Smart Mirrors

황기태\*, 김경미\*\*, 김유진\*\*, 박채원\*\*, 유송연\*\*, 정인환\*\*, 이재문\*\*

Kitae Hwang\*, Kyung-Mi Kim\*\*, Yu-Jin Kim\*\*, Chae-Won Park\*\*,  
Song-Yeon Yoo\*\*, Inhwan Jung\*\*, Jae-Moon Lee\*\*

**요약** 스마트 미러는 거울에 디스플레이와 임베디드 컴퓨터를 부착하여 거울 기능과 함께 사용자에게 다양한 정보를 제공해주는 IoT 장치이다. 본 논문은 스마트 미러가 사용자에게 정보를 제공하는 독립형 장치(stand alone device)라는 개념에서 벗어나 스마트 미러들이 연결되는 네트워크를 구성하고 사용자들이 다른 스마트 미러 사용자들과 대화하고 정보를 공유하는 CoMirror 시스템을 제안하고 구현하였다. CoMirror 시스템은 1개의 CoMirror 서버를 중심으로 여러 CoMirror 클라이언트들이 연결되는 구조이다. CoMirror 클라이언트는 라즈베리파이와 미러 필름, 터치 패드, 디스플레이 장치, 웹 카메라 등으로 구성되며, 서버에는 얼굴 학습과 인식, 사용자 관리, 클라이언트들 사이의 메시지 교환을 위한 중계 역할, 화상 통화 연결 설정 등의 기능이 구현되었다. 사용자들은 서버를 경유하여 다른 CoMirror 사용자들과 텍스트, 이미지, 오디오 등의 메시지를 주고받을 뿐 아니라, 1:1 화상 통화를 할 수 있도록 구현되었다.

**Abstract** Smart mirror is an IoT device that attaches a display and an embedded computer to the mirror and provides various information to the user along with the mirror function. This paper went beyond the form of dealing with smart mirrors only stand alone device the provide information to users, and constructed a network in which smart mirrors are connected, and proposed and implemented a CoMirror system that allows users to talk and share information with other smart mirror users. The CoMirror system has a structure in which several CoMirror clients are connected on one CoMirror server. The CoMirror client consists of Raspberry Pi, a mirror film, a touch pad, a display device, an web camera, etc. The server has functions such as face learning and recognition, user management, a relay role for exchanging messages between clients, and setting up for video call. Users can communicate with other CoMirror users via the server, such as text, image, and audio messages, as well as 1:1 video call.

**Key Words** : Smart Mirror, Video Call, Messaging, IoT

\*중신회원, 한성대학교 컴퓨터공학부(교신저자)

\*\*정회원, 한성대학교 컴퓨터공학부

접수일자 2022년 10월 22일, 수정완료 2022년 11월 22일

게재확정일자 2022년 12월 9일

Received: 22 October, 2022 / Revised: 22 November, 2022 /

Accepted: 9 December, 2022

\*Corresponding Author: calafk@hansung.ac.kr

Dept. of Computer Engineering, Hansung University, Korea

## I. 서 론

IoT 기술의 발전으로 TV, 냉장고, 세탁기, 에어컨 등 많은 전자제품들에 소형 컴퓨터와 통신 장치가 내장되어 정보를 주고받는 홈 네트워크 시스템이 활성화되고 있다 [1,2,3]. 한편, 2016년 구글 엔지니어인 Max Brown에 의해 스마트 미러가 개발되었다[4]. 스마트 미러는 거울의 안쪽에 디스플레이와 임베디드 컴퓨터를 부착하고 사용자에게 정보를 제공하는 간단한 IoT 장치이다. 그 후 스마트 미러는 AI 기술을 접목하여 얼굴 인식, 홈 트레이닝, 홈 헬스케어 등 다양한 기능으로 진화해왔다[5,6]. 거울 대신 냉장고에 디스플레이를 부착하여 스마트 미러와 유사한 삼성전자의 냉장고 패밀리 허브가 있다[7]. 이 제품은 인터넷과 연결하여 요리 방법을 보여주는 등 냉장고를 IoT 장치로 만든 사례이다. 지금까지 스마트 미러나 가전제품은 주로 인터넷을 통해 외부 세계의 정보를 제공하거나 홈 내 다른 장치와 연결하는 등 홈 네트워크를 구성하는데 주안점이 있어왔다.

본 논문은 스마트 미러가 사용자에게 정보를 제공하는 독립형 장치에서 벗어나 서버-클라이언트 모델로 스마트 미러들이 연결되는 네트워크를 구성하고, 다른 스마트 미러의 사용자와 대화하고 정보를 공유하는 CoMirror 시스템을 제안하고 구현하였다. 구체적으로 여러 CoMirror 클라이언트들이 1개의 CoMirror 서버로 연결되며, CoMirror 서버에는 모든 사용자들의 정보가 저장된다. 사용자들은 서버의 중계를 통해 텍스트, 이미지, 오디오 등의 메시지를 주고받으며, 화상 통화를 할 수 있도록 구현하였다.

본 논문은 터치가 가능한 디스플레이 장치를 직접 제작하고, 카메라와 Raspberry PI를 활용하여 CoMirror 클라이언트를 구현하였고, 모든 입력은 음성 인식과 터치로만 가능하도록 구현하였다. 음성 출력은 라즈베리파이 에 블루투스 스피커를 활용하였으며, 높은 처리 능력이 요구되는 얼굴 학습과 얼굴 인식[8]은 CoMirror 서버에 구현하였다.

## II. 연구 배경

### 1. 스마트 미러

스마트 미러는 필름 거울의 뒷면에 디스플레이를 달고 인터넷과 연결하여 날씨, 건강, 시사 정보 등을 보여주는 가전 장치이다[3]. 지금은 가전제품들을 모니터링하거나

제어하는 스마트 홈뿐 아니라, 쇼핑몰, 화장실, 피트니스 클럽의 전면 거울, 길거리 디스플레이 광고판 등 다양한 곳에 응용된다[9]. 최근 들어 AI 기술을 접목하여 AI 헬스케어, 운동, AI 비서 등 활용성은 더 넓어지고 있으며 스마트 미러 시장은 2027년까지 미화 59억 달러에 이를 것으로 전망된다[10].

### 2. Electron과 데스크톱 웹 애플리케이션

HTML5, CSS, 자바스크립트를 이용하여 웹 서버 없이 데스크톱에서 실행되는 애플리케이션을 작성할 수 있는 Electron 프레임워크가 최근 주목받고 있다[11].

Electron은 자바스크립트의 실행 엔진인 Node.js와 크롬 브라우저의 UI 부분에 해당하는 Chromium을 내장하고 있기 때문에, Electron을 활용하여 개발된 애플리케이션은 Windows나 리눅스, MacOS 등 운영체제에 독립적인 데스크톱 애플리케이션으로 실행된다. 이런 장점으로 인해 최근 Electron을 이용한 데스크톱 애플리케이션 개발이 확산되고 있으며, vscode, Discord, Slack, Atom 등이 대표적인 사례이다. 본 연구에서는 Electron 프레임워크를 이용하여 CoMirror 클라이언트에서 실행되는 애플리케이션을 개발하였다.

### 3. MQTT

오늘날 IoT 표준 프로토콜로 사용되는 MQTT는 여러 클라이언트들과 메시지를 중계하는 브로커로 구성된다. 클라이언트는 subscriber와 publisher로 구분된다. subscriber는 브로커에 토픽 문자열을 보내 받고자 하는 토픽을 등록하면, publisher가 토픽과 메시지를 브로커에 보낼 때, 브로커는 토픽을 등록한 모든 subscriber에게 메시지를 보내는 식으로 작동한다. 본 연구에서는 모듈 사이에 데이터를 전송하는 대부분의 경우에 MQTT를 활용하였으며, 이클립스 재단에서 공개 소프트웨어로 개발한 Mosquitto를 브로커로 사용하였다.

## III. CoMirror 시스템

### 1. 시스템 구성

본 논문에서 제안하는 시스템은 여러 CoMirror 클라이언트들이 하나의 CoMirror 서버에 연결되는 구조로 그림 1과 같다. 지금부터 CoMirror 클라이언트를 간단히 CoMirror, CoMirror 서버를 서버라고 부른다. 각 가정

에 CoMirror가 있고 서버는 외부에 있다. CoMirror들은 서버를 통해 데이터를 공유한다.

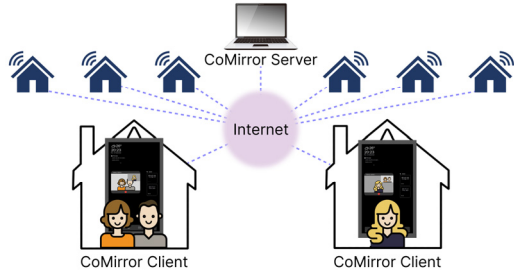


그림 1. 시스템 개요  
 Fig. 1. System View

본 연구에서 목표로 하는 CoMirror 시스템의 기능은 다음과 같다. 첫째, 얼굴 인식을 통해 로그인한다. 얼굴 학습과 얼굴 인식의 실질적인 처리는 서버에서 이루어진다. 둘째, 모든 입력은 음성인식과 터치로만 이루어진다. 셋째, 사용자들은 자신을 위한 메모를 남길 수 있다. 메모 형태는 텍스트, 사진 등의 이미지, 녹음된 음성 혹은 오디오 등이다. 넷째, 사용자들은 다른 사용자에게 메시지 형태로 메모를 보낼 수 있다. 동일한 CoMirror의 다른 사용자에게 메시지를 보낼 때 메시지는 CoMirror 내에 보관되지만, 다른 CoMirror 사용자에게 보낸 메시지는 서버에 의해 임시 보관된 후 서버에 의해 다른 사용자에게 전송된다. 다섯째, CoMirror 사용자들끼리 화상 통화를 지원한다.

## 2. CoMirror 클라이언트와 서버

본 논문에서 구현된 CoMirror 클라이언트는 그림 2와 같다. 거울 효과를 만들기 위해 아크릴 판에 거울 필름을 붙이고 디스플레이 위에 아크릴판을 쌓았다. 그리고 다시 그 위에 적외선 터치 패널을 쌓아 터치 입력이 가능하도록 제작하였다.

디스플레이는 라즈베리파이의 HDMI에, 터치 패널과 웹 카메라는 USB 단자에 연결하였다. 라즈베리파 모델은 Raspberry PI 3 B+이며 Bullseye 리눅스 운영체제를 탑재하였다.

서버는 Windows 11 운영체제에 2.30Ghz의 11세대 인텔 i7-11800H CPU와 16GB RAM을 가진 MSI 노트북을 사용하여 구현하였다. 서버는 다음 3가지 목적으로 사용된다. 첫째, CoMirror들 사이에 주고받는 메시지의 임시 저장소의 역할을 한다. 사용자는 다른 CoMirror의

사용자에게 메시지를 보내려면 서버에 전송한다. 수신자가 현재 CoMirror를 사용 중이면 서버는 수신자에게 메시지가 바로 전달하지만, 그렇지 않은 경우 서버는 메시지를 저장해두었다가 수신자가 로그인할 때 전송한다. 둘째, CoMirror는 사용자의 얼굴을 촬영하여 서버로 보내는 역할만 하고, 얼굴 학습과 얼굴 인식은 서버에서 처리한다. 셋째, 서로 다른 CoMirror의 사용자들이 화상 통화를 할 때 이들을 연결하는 작업은 서버에 의해 이루어진다.

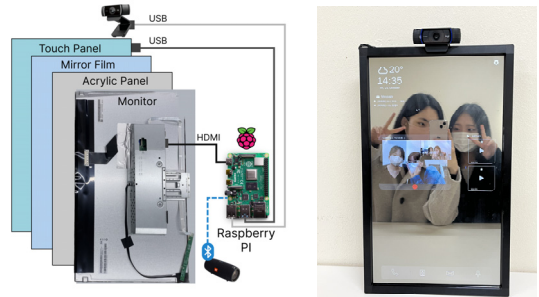
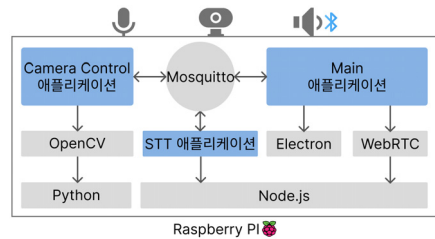


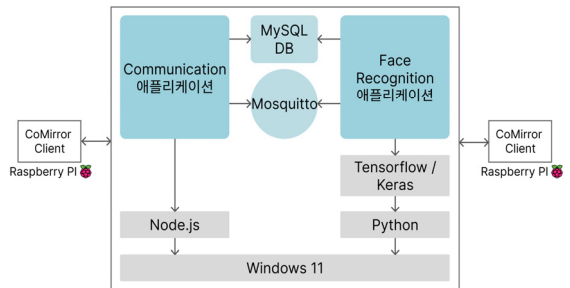
그림 2. CoMirror 클라이언트 하드웨어  
 Fig. 2. Hardware of CoMirror client

## 3. 소프트웨어 전체 구성

그림 3은 CoMirror 시스템의 소프트웨어의 전체 구성을 보여준다. CoMirror에는 독립적인 4개의 애플리케이션이 실행된다.



(a) CoMirror 클라이언트의 소프트웨어 구성



(b) CoMirror 서버의 소프트웨어 구성

그림 3. CoMirror 시스템의 전체 소프트웨어 구성  
 Fig. 3. Overall software structure of CoMirror system

이들은 CoMirror를 제어하고 GUI를 수행하는 Main 애플리케이션, 마이크로부터 음성 입력을 받아 텍스트로 변환하는 STT(Speech To Text) 애플리케이션, 카메라를 제어하고 영상을 입력받는 Camera Control 애플리케이션, 마지막으로 3개의 애플리케이션이 서로 데이터를 주고받도록 중계 역할을 하는 Mosquitto 브로커이다. Mosquitto를 제외한 3개의 애플리케이션은 본 연구에서 구현되었다.

Main 애플리케이션은 HTML5, CSS3, 자바스크립트를 사용하여 Electron 애플리케이션으로 개발되었으며, STT 애플리케이션은 자바스크립트 언어로 개발되어 Node.js에 의해 실행되고, Camera Control 애플리케이션은 OpenCV를 활용하며 파이선으로 개발되었다. 화상 통화는 WebRTC 라이브러리를 이용하여 구현되었다.

서버 소프트웨어는 Communication 애플리케이션과 Face Recognition 애플리케이션, Mosquitto 브로커로 구성된다. Face Recognition 애플리케이션은 등록하는 사용자의 얼굴 학습과 로그인 시 얼굴 인식을 수행한다. Communication 애플리케이션은 서로 다른 CoMirror 사용자들끼리 메시지를 보내거나 화상 통화 시 초기 셋업 작업을 실행한다. Mosquitto 브로커는 CoMirror의 Main 애플리케이션이 얼굴 학습과 얼굴 인식을 위해 Face Recognition 애플리케이션에 얼굴 이미지를 보내고 결과 값을 받을 때와, Main 애플리케이션과 다른 CoMirror의 Main 애플리케이션 사이에 메시지를 중계해주는 Communication 애플리케이션과 메시지를 주고받을 때 이용된다.

#### 4. 회원 가입과 얼굴 학습

그림 4는 회원 가입 과정을 보여준다. CoMirror의 Main 애플리케이션은 사용자의 '이름'을 Mosquitto로 전송한다. Camera Control 애플리케이션이 이 메시지를 받으면 사용자의 전면 사진을 20장 촬영하고 얼굴 부분만 파일에 저장한 다음, 사용자 이름과 CoMirror ID와 함께 서버의 Mosquitto로 전송한다.

Face Recognition 애플리케이션이 Mosquitto로부터 얼굴 이미지가 들어 있는 MQTT 메시지를 받으면 얼굴 이미지를 벡터로 가공하는 전처리 과정을 실행한다. 이 과정에서 keras의 FaceNet 라이브러리를 사용되며 얼굴 이미지로부터 얼굴 특징을 추출되며 128개의 특징 벡터가 생성된다. 그 후 Tensorflow의 SVM(Support Vector Machine) 모델을 사용하여 얼굴을 학습시킨다. 학습이 끝나면 서버의 Mosquitto를 통해 CoMirror의

Main 애플리케이션에게 알린다. 이후부터 얼굴로 로그인이 가능해진다.

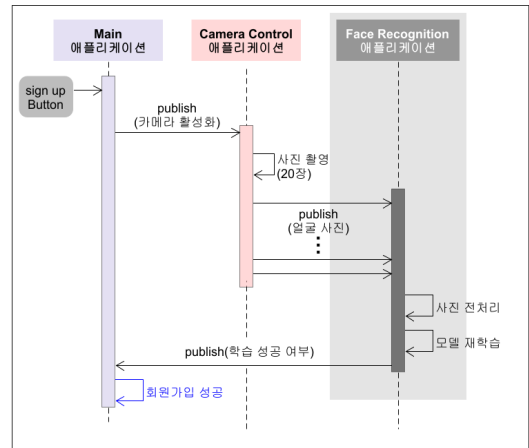


그림 4. 회원 가입 시 얼굴 학습 과정  
Fig. 4. Process of face learning on sing up

#### 5. 로그인과 얼굴 인식

Camera Control 애플리케이션이 10장의 사진을 찍고 얼굴 부분만 잘라 파일로 저장한 다음, 서버의 Mosquitto를 경유하여 Face Recognition 애플리케이션으로 보낸다. Face Recognition 애플리케이션은 받은 사진들을 이용해 얼굴을 인식한다. 인식 결과는 서버의 Mosquitto를 경유하여 CoMirror의 Main 애플리케이션으로 보내진다. 그림 5는 로그인 과정을 보여준다.

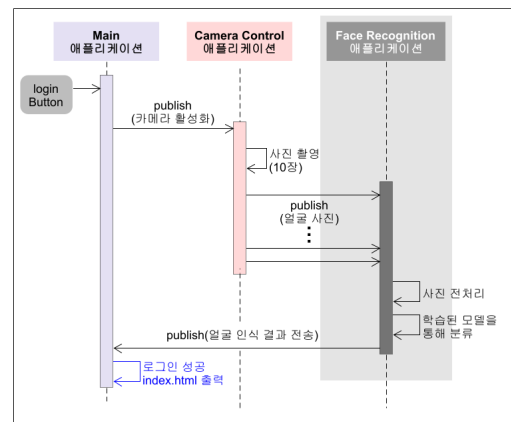


그림 5. 얼굴 인식을 이용한 로그인 과정  
Fig. 5. Process of login using face recognition

## 6. 메시지 전송

CoMirror에서 메시지를 전달하고 수신하는 기능은 Main 애플리케이션에서 이루어진다.

### 가. 수신자가 온라인 상태일 때

사용자 A가 온라인 사용자 B에게 메시지를 보내는 과정은 그림 6과 같다. A는 송신자 ID, 수신자 ID, 메시지 타입, 메시지 내용, 보낸 시간으로 구성된 MQTT 메시지를 만들어 Communication 애플리케이션에게 보낸다. Communication 애플리케이션은 받은 MQTT 메시지를 B에게 보내고, 사용자 B의 Main 애플리케이션은 MQTT 메시지에서 사용자 A로부터 받은 메시지를 분리하여 자신의 DB에 저장해두고, 디스플레이에 메시지가 왔음을 알려 사용자 B가 메시지를 보게 한다.

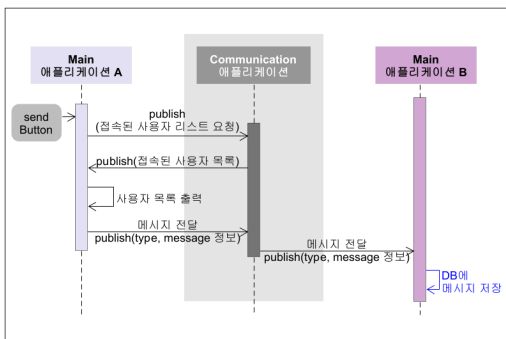


그림 6. 온라인 사용자 B에게 메시지를 전송할 때  
 Fig. 6. When a message is sent to online user B

### 나. 수신자가 오프라인 상태일 때

사용자 A가 오프라인 사용자 B에게 메시지를 보내는 과정은 그림 7과 같다.

A는 송신자 ID, 수신자 ID, 메시지 타입, 메시지 내용, 보낸 시간으로 구성된 MQTT 메시지를 만들어 Communication 애플리케이션에게 보낸다. Communication 애플리케이션은 받은 MQTT 메시지 중에서 메시지를 분리하여 DB에 저장하고, 임시 보관중인 메시지를 나타내는 DB 테이블에 B가 수신할 메시지가 대기 중임을 기록해둔다. 사용자 B가 로그인하면 Main 애플리케이션은 Communication 애플리케이션에게 대기 중인 메시지를 받아와 DB에 저장해두고, 디스플레이에 메시지가 왔음을 알려 사용자 A가 메시지를 선택해 보게 한다.

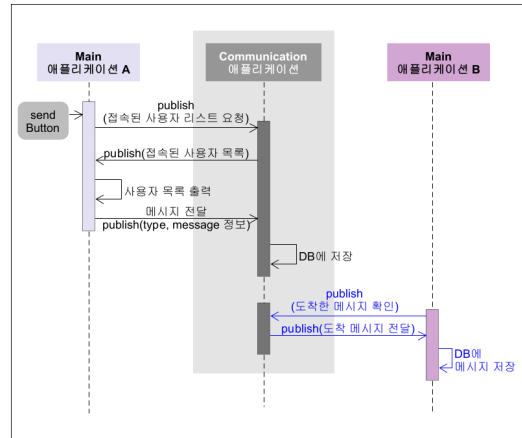


그림 7. 오프라인 사용자 B에게 메시지를 전송할 때  
 Fig. 7. When a message is sent to offline user B

## 7. 화상 통화

그림 8은 사용자 A와 B의 화상 통화 과정을 보여준다. 사용자들이 자신의 CoMirror에 로그인 하면 Main 애플리케이션이 Communication 애플리케이션에 접속하고 Mirror ID로 룸(room)이 만들어진다.

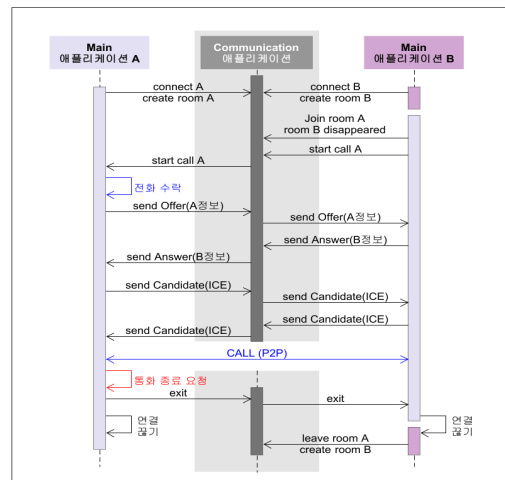


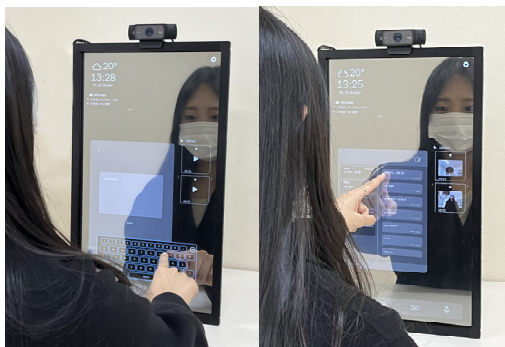
그림 8. 화상 통화 과정  
 Fig. 8. Process for video call

B가 A와 화상통화를 원하면, A의 ID, 자신의 ID, 통화 방법(화상 통화인지 단순 전화인지)을 Communication 애플리케이션에 전송하여 A의 룸에 참가한다. 현재 로그인된 모든 사용자에 관한 정보는 Communication 애플리케이션으로부터 받을 수 있다. Communication 애플리케이션은 A에게 B의 통화 요청을 알린다. A가 통화 요

청을 수락하면 자신의 세션 정보를 서버에게 전달하고, 서버는 A의 세션 정보를 B에 전달한다. B는 A의 세션 정보를 받으면, 역시 자신의 세션 정보를 서버에게 전달하고, 서버는 다시 A에게 B의 세션 정보를 전달한다. 두 사용자가 서로 세션 정보를 주고받으면, 서로의 ICE(P2P 통신 식별번호)를 서버를 거쳐 상대방에게 전달하며, 이로써 A와 B는 화상 통화가 진행된다.

## 8. 시연

그림 9는 메모를 시연하는 과정으로 홈 화면의 하단에는 화상통화, 메모, 연락처 조회, 메시지 기능을 위한 버튼들이 있으며, 그림 10(a)은 메시지 함에서 수신한 메시지를 보는 화면이며 그림 10(b)은 실시간 화상 통화가 작동 중인 화면으로, 작게 보이는 화면이 본인의 카메라 출력 화면이며, 큰 화면은 상대방 카메라로부터 받은 화면이다.

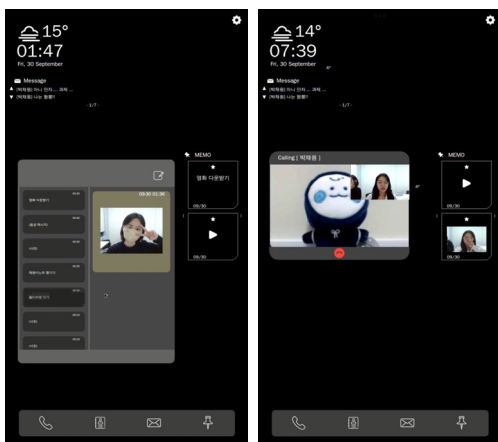


(a) 터치로 메모 작성

(b) 메모 보기

그림 9. 메모 시연

Fig. 9. Memo demonstration



(a) 메시지 보기

(b) 화상 통화 화면

그림 10. 메시징과 화상 통화 시연

Fig. 10. Demonstration of messaging and video calling

## VI. 결 론

본 논문은 스마트 미러를 서버 클라이언트 모델로 구현하여 스마트 미러 사이에 메시지를 주고받고 화상 통화를 할 수 있는 시스템 CoMirror를 설계 구현한 내용을 소개하였다. 클라이언트에는 라즈베리파이를 사용하고 터치 패드를 붙여 터치 입력이 가능하도록 제작하였으며, 서버는 얼굴 학습 및 얼굴 인식, 사용자들 사이의 메시지 중계, 화상 통신 셋업 등의 작업을 수행하도록 구현하였다. 본 논문은 스마트 미러들이 서로 정보를 공유하고 교환할 수 있는 네트워크 시스템의 아키텍처를 보여주며 실제 구현을 통해 실증하였다는 점에서 의미가 있다. 하지만, 이 논문에서는 얼굴 학습이나 얼굴 인식, 메시지 전송, 화상 통화 셋업 등에 걸리는 시간에 대한 성능 평가가 이루어지지 않았으며 이들은 추후 연구를 통해 달성하고자 한다.

## References

- [1] S. S. I. Samuel, "A review of connectivity challenges in IoT-smart home", 2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC), pp. 1-4, 2016. doi: 10.1109/ICBDSC.2016.7460395.
- [2] H. Thapliyal, "Internet of Things-Based Consumer Electronics: Reviewing Existing Consumer Electronic Devices, Systems, and Platforms and Exploring New Research Paradigms", IEEE Consumer Electronics Magazine, Vol. 7, No. 1, pp. 66-67, 2018.
- [3] <https://www.etnews.com/20210812000195>
- [4] Max Braun. Smart Mirror, <https://experiments.withgoogle.com/smart-mirror>, March 2016
- [5] Un-Chan Yeo, Sin-Hoo Park, Jin-Wan Moon, Seong-Won An, Yeong-Oh Han, "Smart Mirror of Personal Environment using Voice Recognition, Journal of the KIECS", Vol. 14, No. 1, pp. 199-204, 2019.
- [6] M. M. Yusri et al., "Smart mirror for smart life", 6th ICT International Student Project Conference, pp. 1-5, 2017.
- [7] <https://www.samsung.com/us/explore/family-hub-refrigerator/overview/>
- [8] Kitae Hwang, Jae-Moon lee, Inhwan Jung, "Remote Control System using Face and Gesture Recognition based on Deep Learning", The Journal of The Institute of Internet, Broadcasting and Communication (IIBC), Vol. 20, No. 6, pp. 115-121, 2020.



- [9] [http://digitalchosun.dizzo.com/site/data/html\\_dir/2015/05/08/2015050812733.html](http://digitalchosun.dizzo.com/site/data/html_dir/2015/05/08/2015050812733.html)
- [10] <https://www.marketsandmarkets.com/Market-Reports/smart-mirror-market-71622395.html>
- [11] <https://www.electronjs.org/docs/latest>

### 저 자 소개

#### 황 기 태(중신회원)



- 서울대학교 컴퓨터공학과 학사
- 서울대학교 컴퓨터공학과 석사
- 서울대학교 컴퓨터공학과 박사
- 1994년 ~ 현재 한성대학교 컴퓨터공학과 교수
- 경력  
University of Florida 방문 교수
- 관심분야 : 모바일 시스템, IoT, 인공지능

#### 김 경 미(비회원)



- 한성대학교 컴퓨터공학부 재학
- 관심분야 : 웹 공학, 분산 시스템

#### 김 유 진(비회원)



- 한성대학교 컴퓨터공학부 재학
- 관심분야 : 웹 공학, 게임 프로그래밍

#### 박 채 원(비회원)



- 한성대학교 컴퓨터공학부 재학
- 관심분야 : 웹 공학, UI/UX

#### 유 송 연(비회원)



- 한성대학교 컴퓨터공학부 재학
- 관심분야 : 웹 공학, 분산 시스템

#### 정 인 환(정회원)



- KAIST 정보및통신공학과 박사
- 삼성전자 수석연구원
- 한성대학교 컴퓨터공학부 교수
- 관심분야 : IoT, 분산시스템, 모바일 시스템

#### 이 재 문(정회원)



- 한양대학교 전자공학과(학사)
- 한국과학기술원 전기및전자공학과(석사)
- 한국과학기술원 전기및전자공학과(박사)
- 경력  
한국통신 연구개발단
- 관심분야 : 기계학습, 게임프로그래밍, 감성컴퓨팅

※ 본 연구는 한성대학교 교내 학술 연구비를 지원받았음