

Digital Mirror System with Machine Learning and Microservices

Myeong Ho Song[†] · Soo Dong Kim^{††}

ABSTRACT

Mirror is a physical reflective surface, typically of glass coated with a metal amalgam, and it is to reflect an image clearly. They are available everywhere anytime and become an essential tool for us to observe our faces and appearances. With the advent of modern software technology, we are motivated to enhance the reflection capability of mirrors with the convenience and intelligence of realtime processing, microservices, and machine learning. In this paper, we present a development of Digital Mirror System that provides the realtime reflection functionality as mirror while providing additional convenience and intelligence including personal information retrieval, public information retrieval, appearance age detection, and emotion detection. Moreover, it provides a multi-model user interface of touch-based, voice-based, and gesture-based. We present our design and discuss how it can be implemented with current technology to deliver the realtime mirror reflection while providing useful information and machine learning intelligence.

Keywords : Digital Mirror, Face Recognition, Age Detection, Emotion Detection, Microservice, Machine Learning

머신 러닝과 Microservice 기반 디지털 미러 시스템

송 명 호[†] · 김 수 동^{††}

요 약

거울은 일반적으로 아말감으로 코팅된 물리적 반사 표면으로 거울 앞의 상을 선명하게 반사한다. 이것은 언제 어디서나 사용이 가능하며 사용자의 얼굴이나 외모를 확인하기 위한 필수적인 도구이다. 현대 소프트웨어 기술의 출현으로 사람들은 실시간 처리, Microservice 및 머신 러닝이 적용된 편의성과 지능성을 통해 거울 반사 기능을 향상시킬 수 있다. 본 논문에서는 거울로써 실시간 반영과 동시에 사용자 맞춤 정보 조회, 공공 정보 조회, 외모를 통한 나이와 감정 탐지 등의 기능을 가진 디지털 거울 시스템 개발을 제안한다. 더불어, 본 시스템은 터치 기반, 음성 인식 기반, 제스처 기반의 Multi-Modal 사용자 인터페이스를 제공한다. 본 논문에서는 이 시스템에 대한 디자인을 제시하고 현재 기술을 이용하여 실시간 거울 반영과 동시에 유용한 정보 제공 및 지능형 머신 러닝 기술을 제공하는 구현 방법을 제안한다.

키워드 : 디지털 거울, 얼굴 인식, 나이 탐지, 감정 탐지, Microservice, 머신 러닝

1. 서 론

거울은 아말감 등의 금속으로 코팅된 유리를 이용해 물체의 모양이나 형상을 비추어 볼 수 있게 하는 물건이다. 이러한 거울은 언제 어디서나 사용이 가능하며 사용자의 얼굴이나 외모 등을 확인하기 위한 필수적인 도구이다.

최근 소프트웨어 기술의 발전으로 실시간 처리, Microservice 및 머신 러닝과 같은 기술의 발달로 거울의 기능을 향상시킬 수 있다. 현재 실시간 처리를 이용하여 아날로그식 거울에 디지털 요소를 추가하는 연구와 실험이 활발히 진행중이다[1-8]. 뿐만

아니라 IoT 시스템에 다양한 기능 제공을 위한 Microservice 아키텍처 설계 및 연구[9-16]와 IoT 시스템에서 머신 러닝 모델을 적용한 설계 및 연구[17-23]가 활발히 진행되고 있다. 이와 같이 디지털 거울 시스템의 구현과 해당 시스템에 Microservice와 머신 러닝을 접목하여 사용자 편의성을 높인 시스템 개발에 대한 가능성과 필요성을 보여주고 있다.

본 논문에서는 머신 러닝과 Microservice 기반 디지털 미러 시스템을 제안한다. 본 시스템은 본래의 물리적 거울의 기능인 사물의 실시간 반영을 제공하면서 머신 러닝과 Microservice를 이용하여 사용자의 얼굴을 통한 나이와 감정 탐지 및 사용자 맞춤 정보 조회, 공공 정보 조회 등의 기능을 제공한다. 또한 얼굴 인식을 통한 자동 세션 관리 기능을 제공한다. 그리고 Multi-Modal 사용자 인터페이스(User Interface)를 제공하여 사용자에게 다양한 방법의 상호작용을 제공한다.

위의 기능을 만족할 때 발생하는 요구사항과 이를 해결하

[†] 준 회원 : 송실대학교 소프트웨어학과 박사과정

^{††} 종신회원 : 송실대학교 소프트웨어학과 교수

Manuscript Received : May 21, 2020

First Revision : July 27, 2020

Accepted : July 30, 2020

* Corresponding Author : Soo Dong Kim(sdkim777@gmail.com)

기 위한 시스템 디자인을 제시한다. 더불어 현재 기술을 이용하여 센서를 통한 실시간 거울 반영 기능과 Microservice 및 머신 러닝 모델을 통한 유용한 정보를 제공하는 구현 방법을 제안하고 제안한 설계에 따라 구현 및 평가를 진행한다.

2. 관련 연구

Derrick의 연구[24]는 라즈베리 파이 (Raspberry Pi)를 이용하여 기존의 거울을 디스플레이로 사용하여 Web 기반 사용자 인터페이스를 이용한 스마트 거울을 제작하였다. 본 연구는 스마트 거울 기능의 확장성을 높였지만 다양한 사용자에게 맞는 데이터를 보여주기 힘들다는 단점이 있다.

Mohammed의 연구[25]는 스마트폰 앱을 통하여 스마트 미러에 사용자별 다른 화면을 볼 수 있게 하였다. 본 연구는 사용자별로 한 스마트 미러에서 다른 사용자 인터페이스나 기능들을 제공하는 것을 보여줬지만 스마트폰을 이용하여 해당 사용자에게 대한 세션을 관리하고 스마트 미러에 접근해야 된다는 어려움이 있다.

Ayushman의 연구[26]는 기존의 거울과 디스플레이를 이용하여 음성인식 기반의 스마트 미러를 제안하였다. 해당 기능은 Google Assistance를 이용하여 제안하였다. Adokiye 연구[27] 또한 음성인식 기반의 시스템을 제안하였다. 해당 연구에서는 Amazon Alexa와 AWS를 기반으로 하는 시스템을 제안하였다.

Raju의 연구[28]는 라즈베리 파이에 카메라를 설치하여 사람을 감지하는 스마트 미러를 제안하였다. 일반적인 스마트 미러에 사람 감지 머신 러닝 모델을 적용함으로써 보안에 강점을 두었다. Pedro의 연구[29]는 스마트 미러 형태의 건강 상태 모니터링 시스템을 제안하였다. 여러 개의 카메라를 설치하여 사람의 얼굴의 요소 변화 등을 감지하여 현재 사람의 건강 척도(Wellness Index)를 예측하는 연구를 진행하였다.

Table 1은 위의 연구들과 본 논문의 시스템을 기능성 단위로 비교한 것이다.

Table 1에서의 기능성은 디지털 미러 시스템의 기능성들을 나열하였다. 대부분의 연구에서 카메라 센서 등을 사용하지 않고 기존의 거울을 사용하여 제작하였기 때문에 적은 수의 연구만 실시간 이미지 반영을 보였다. 더불어 대부분의 연구들은 세션관리를 고려하지 않았고, 세션 관리가 보장되는 연구의 경우도 사용자의 직접적인 입력을 통해 세션을 생성할 수 있다. 공공 정보의 경우 대다수의 연구들이 제공을 하였지만 개인 정보의 경우 세션 관리를 포함하는 연구에서만 제공되었다. 머신 러닝 모델 또한 일부 연구에서 건강 정도 확인 및 사람 검출에 대해 적용되었다. 특히 사용자와 상호작용은 많은 연구들이 사용자의 입력을 음성 또는 터치 기반 중 한가지로 제한을 두거나 사용자와 다른 상호작용 없이 정보를 제공하는 경우가 많았다. 디지털 미러 시스템을 이용할 시, 일반적인 정보 제공 뿐 만 아니라 각 사용자에게 맞는 정보 제공이 필요하다. 이 때, 세션 관리가 복잡하게 되면 해당 거울에 사용성이 떨어질 수 있기 때문에 자동 세션 관리 기능이 필요하다. 또한 사용자가 거울을 사용할 때 다양한 활동을 할 수 있으므로 상호작용 또한 한가지 방법으로 제한을 두면 안된다. 이러한 방법을 위해 자동 세션 관리나 개인 정보 제공, 사용자와의 상호작용 등에 머신 러닝 모델이 필요하다.

본 시스템은 카메라 기반의 실시간 이미지 반영을 기반으로 한다. 해당 사용자가 거울을 사용할 때 불편하지 않게 얼굴 인식 기반 자동 세션관리를 제공하고 이에 따라 개인 정보와 공공 정보를 제공한다. 이를 위해 다수의 머신 러닝 모델이 적용된다. 또한 해당 사용자가 거울을 사용할 시 편의성이 중요하게 고려되어야 하기 때문에 사용자와 상호작용할 수 있는 방법을 터치 또는 음성 인식 기능에 제한하지 않고 Multi-Modal 기반의 상호작용을 제공한다. 본 논문에서는 카메라를 이용한 디지털 거울 시스템에 여러 가지 기능들을 관리 및 추가할 수 있고, 다양한 입력 방법을 이용한 시스템을 제안한다. 또한 센서를 이용하여 수집된 데이터를 통해 머신 러닝 모델들을 통해 해당 사용자에게 대한 세션 관리나 정보를 제공하는 시스템을 제안한다.

Table 1. Comparison of Related Works

Study	Derrick [24]	Mohammed [25]	Ayushman [26]	Adokiye [27]	Raju [28]	Pedro [29]	Our
Real Time Reflection	X	X	X	X	○	○	○
Auto Session Management	X	△ (Applying Smart Phone)	X	X	X	X	○
Providing Public Information	○	○	○	○	○	X	○
Providing Personal Information	X	○	X	X	X	○	○
Applying ML Model	X	X	X	X	○	○	○
Multi-Modal User Interaction	△ (Touch)	△ (Touch)	△ (Voice)	△ (Voice)	○ (Voice, Touch)	X	○ (Touch, Voice, and Etc.)

3. 지능형 디지털 거울시스템의 개요

본 시스템은 상용화를 목적으로 기획 및 개발되었으며, 본 장에서는 이 시스템 개발을 위한 핵심적인 요구사항을 열거한다. 본 시스템은 Microservice 및 머신 러닝을 적용한 실시간 처리 기반 디지털 거울이다. 하드웨어를 통해 현재 보이는 상황을 반영하면서 사용자가 필요한 정보를 제공하는 기능을 한다. 아래는 본 시스템의 높은 품질을 위해 요구되는 요구사항이다.

- 이미지 반영 기능의 실시간성
- 지능형 자동 세션 관리
- Microservice 기반 공공서비스 정보 제공
- 맞춤형 PIM 정보 제공
- Multi-Modal 기반 사용자 상호 연동 및 확장성

이러한 요구사항을 만족하기 위한 하드웨어는 Fig. 1과 같이 구성되어 있다.

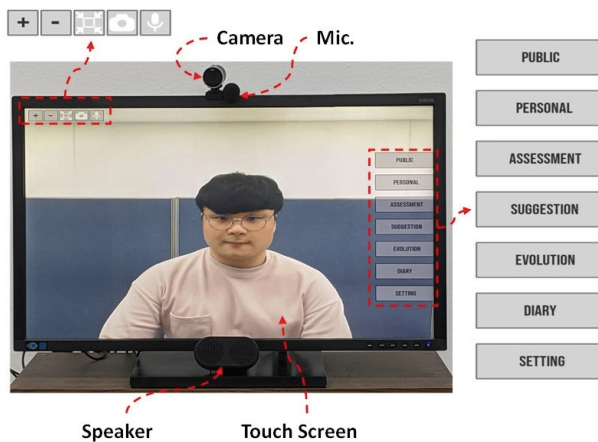


Fig. 1. Prototype of Digital Mirror System with User Interface

본 시스템은 카메라(Camera), 마이크(Mic), 스피커(Speaker), 터치 스크린(Touch Screen)으로 이루어져 있다. 센서들을 통해 사용자의 데이터를 실시간으로 수집하고 머신 러닝 모델과 Microservice를 적용하여 데이터를 수집 및 분석하여 해당 사용자에게 필요한 정보를 반환한다. 우선 터치 스크린을 통하여 현재 시스템이 카메라를 통하여 수집하는 이미지 데이터를 반영하고 사용자와 상호작용을 하게 된다. 또한 마이크를 통해 사용자가 시스템을 조작할 수 있으며 해당 입력에 대한 결과물을 스피커를 통해 반환하게 된다.

Fig. 1의 좌측 상단의 버튼들을 통해 화면의 확대, 축소, 버튼과 화면을 원래 크기로 되돌리기 위한 버튼, 현재화면을 캡처 하기 위한 버튼, 음성인식 기능을 활성화하기 위한 버튼이 있다.

Fig. 1의 우측 버튼들은 현재 사용자에게 정보를 제공하는 기능들로 사용자의 시스템 가입유무에 따라 달라진다. Personal과 Evolution, Diary 기능은 가입된 사용자가 사용을 할 때 나타나는 버튼들로, 가입하지 않은 사용자들에게는 제공되지 않

는다. Public버튼을 통해 날씨, 대기 정보, 현재 시간 등 일반적인 정보를 제공한다. Personal 버튼의 경우, 현재 사용자가 등록해둔 정보를 이용하여 일정과 이메일을 보여주는 기능을 제공한다. Assessment 기능을 통해 현재 사용자의 얼굴을 분석하여 나이와 감정을 보여주는 기능을 하고, Evolution 버튼은 분석된 결과를 시간변화에 따라 보여준다. Setting 버튼은 현재 시스템의 상태를 설정한다.

3.1 이미지 반영 (Reflection) 기능의 실시간성

본 시스템의 기본 목적은 디지털 거울이다. 거울로써 기본 기능을 하기 위해서는 1초당 약 30개의 프레임(Frame) 이상이 터치 스크린에 반영되어야 기존의 물리적인 거울과 같은 사용성을 가질 수 있다.

그러나 이 시스템은 수집된 이미지를 반영하는 것 이외에도 실행되어야 하는 복수개의 작업(Task)들이 존재한다. 특히, 특정 작업들은 높은 런타임 오버헤드를 가진다. 따라서 이러한 작업들과 이미지 반영이 동시에 진행될 경우, 실시간 반영에 어려움이 생긴다.

그러므로 수집된 이미지가 복수개의 작업들에 사용되면서도 실시간으로 반영될 수 있어야 한다. 즉, 이미지를 반영하는 쓰레드를 메인 쓰레드로 그 외의 작업이 진행되는 쓰레드를 서비스 쓰레드라 할 때 메인 쓰레드는 서비스 쓰레드들과 독립성이 유지가 되어 실시간성을 띄어야 한다. 이를 위해 메인 쓰레드로부터 얻어진 이미지는 쓰레드간 서로 공유할 수 있게 메모리에 설정된다. 메인 쓰레드의 경우 매 프레임당 해당 이미지를 변경하고, 나머지 서비스 쓰레드들은 해당 이미지를 읽어 사용을 하기 때문에 상호 배제(Mutual Exclusion) 이 성립되지 않는다. 따라서 쓰레드 간의 정보 공유가 가능하고 더불어 서로 성능 저하를 발생시키지 않는다.

3.2 얼굴인식 기반 자동 세션관리

본 시스템은 사용자에게 터치스크린만 입력장치로 제공된다. 그러므로 사용자가 본 시스템 사용 중 로그인 등의 방법을 통해 사용자가 직접적으로 세션을 생성할 경우, 거울 시스템임에 반해 번거로움이 발생한다.

본 시스템은 사용자가 카메라에서 감지가 되었을 경우, 수집된 이미지만을 이용하여 세션이 생성 및 유지, 종료된다. 이때 사용자에게서 다른 입력을 받지 않는다. 또한 생성된 세션을 통해 현재 사용자에게 최적화된 기능들을 제공한다. 따라서, 현재 수집된 이미지 내에 사람이 있는 지 여부를 판단하는 사용자 유무 확인과 현재 이미지 내에 있는 사람이 누구인지를 판단하는 사용자 신분 확인을 통해 세션 관리를 한다.

한 사용자의 다양한 모습에도 같은 사용자로 판별하기 위해 사용자 탐지를 위한 모델과 사용자 분류를 위한 모델을 적용한다. 해당 모델들을 통해 다양한 모습의 사용자를 탐지할 수 있고, 높은 정확도로 해당 사람의 신분을 구분할 수 있다. 본 시스템에서는 두 모델을 이용하여 사용자의 유무 확인 및 사용자 신분 확인을 주기적으로 진행하여 현재 사용자의 세션을 관리한다.

해당 모델들은 미흡한 훈련이나 사용자의 달라진 외모 등으로 인해 잘못된 결과를 반환할 수 있다. 이러한 경우, 사용자의 개인 정보 유출이 될 수도 있는 상황이 발생한다. 이 경우에 해당 모델이 사용자의 신분을 반환할 시, 결과에 대한 신뢰도(C Confidence Score)가 비교적 낮은 값으로 반환되게 된다. 따라서 신분 확인 시, 신뢰도에 대한 평가를 진행하여 일정 수치(Threshold) 보다 높을 경우 자동 세션관리 기능을, 낮을 경우 ID와 비밀번호 입력을 기반으로 한 수동 세션관리 기능을 제공한다.

3.3 Microservice 기반 공공서비스 정보 제공

본 시스템은 사용자가 일상 생활을 할 때 필요한 정보들을 제공하여야 한다. 그 중 날짜, 시간, 당일의 날씨나 대기 정보 등의 정보들은 해당 사용자의 시스템 회원가입 유무를 가리지 않고 다양한 사람들에게 제공이 되어야 한다. 이러한 정보들은 개인적인 정보를 담지 않고 해당 정보를 제공하는 외부 서버나 센서를 통해 얻어진 정보들을 통해 수집된다. 이 때 각 정보들의 소스가 다를 수 있기 때문에 각각의 정보들이 수집될 때 다른 정보 수집에 영향을 미치게 되면 안된다.

기본적으로 제공되는 기능 이외에도 추가가 필요한 기능들이 생길 수 있으므로 높은 확장성이 요구된다.

3.4 맞춤형 PIM 정보 제공

본 시스템은 공공 서비스 정보 뿐 만 아니라 일정, 이메일, 수집된 이미지를 통해 얻어진 개인 정보 등 사용자 맞춤 정보(Personal Information Management, PIM)도 제공을 해야 한다. 이러한 정보들은 해당 사용자에게 맞는 정보가 제공되어야 하고 타인에게 제공되어서는 안된다. 따라서 이러한 정보들은 현재 본 시스템을 사용하고 있는 사용자의 세션의 상태와 해당 사용자의 신원을 확인 후 정확한 정보를 전달한다.

이러한 기능들은 사용자의 세션을 이용하여 미리 저장된 사용자 키를 통해 해당 서비스에 접근, 필요한 정보를 수집하여 제공한다. 또한 사용자의 요구사항에 따라 기능이 추가 또는 수정될 수 있다.

3.5 Multi-Modal 기반 사용자 인터페이스

본 시스템은 키보드나 마우스 같은 입력장치를 제공하지 않고 터치 기반의 인터페이스를 갖고 있다. 하지만 사용자가 항상 터치 인터페이스를 사용할 수 없기 때문에 복합적인 방법의 사용자 상호연동 기능이 구현된다. 이러한 상호연동들은 다른 작업이 진행되는데 지연 등을 발생시키지 않으면서 사용자의 입력에 따라 해당 작업을 진행 또는 중지시키는 역할을 해야 한다.

각각의 사용자 상호 연동은 특정 입력에 대해 낮은 정확도를 보일 수 있다. 예를 들어, 음성 인식 모델을 사용하는 음성 인식 기반의 사용자 상호 연동은 사용되지 않은 단어나 음성에 대해서 정확도가 낮게 나올 수 있고, 제대로 된 상호 연동을 지원하지 못할 수 있다. 이렇게 해당 입력을 통해 제대로 상호

연동을 수행하지 못할 경우, 사용자에게 해당 입력에 대한 상호 연동을 다른 정확도 높은 방식의 인터페이스를 이용하여 입력을 받게 한다. 이렇게 모인 데이터를 이용하여 해당 모델을 재 훈련 및 최적화하여 높은 정확도를 보일 수 있게 한다.

각각의 상호연동은 메인 쓰레드에 영향을 주지 않기 위해 서비스 쓰레드로 구현이 되어야 하고 사용자의 입력에 따라 다른 서비스 쓰레드에 속해 있는 작업을 실행시켜야 한다. 따라서 해당 서비스 쓰레드는 이벤트 기반으로 구성되어 다른 서비스 쓰레드에 이벤트를 전송해야 한다. 또한 상호연동은 제스처 기반의 상호연동 등 추가적인 상호연동 구현이 가능하다.

4. 지능형 디지털 거울 시스템 설계

3장의 요구사항 해결을 위해 아래와 같이 5가지 설계 방법(Tactic)을 제시한다. 각각의 설계 방법은 Fig. 2와 같이 요구사항과 연관을 갖는다.

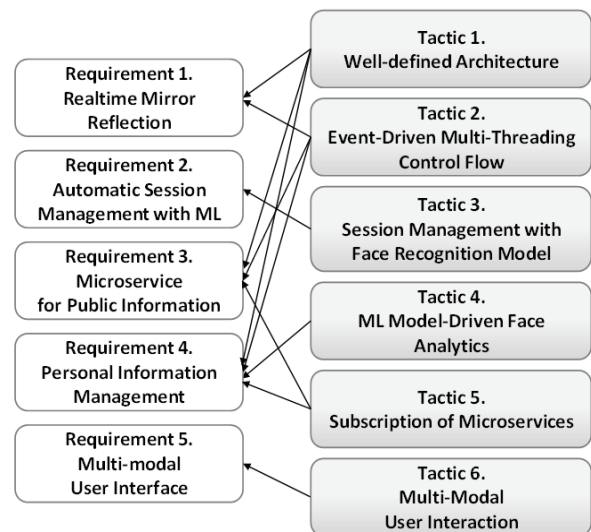


Fig. 2. Design Tactics Satisfying Requirements

정제된 아키텍처를 적용하여 이미지 반영 기능의 실시간성, 공공 정보 및 개인 맞춤 정보 제공을 보장한다. 이벤트 기반 다중 쓰레드 제어 흐름을 통해 이미지 반영 기능의 실시간성, 공공 정보 및 개인 맞춤 정보를 제공한다. 얼굴 인식 모델 기반 사용자 세션 관리를 이용하여 머신 러닝 기반 세션 관리를 보장한다. 머신 러닝 기반의 얼굴 분석 모델과 Microservice 적용 및 관리를 통해 공공 정보 제공과 개인 맞춤 정보 제공을 보장된다. Multi-Modal 사용자 상호작용을 통해 Multi-Modal 기반 사용자 인터페이스를 보장한다.

4.1 정제된 아키텍처 적용

본 시스템은 위에 제시한 대로 머신 러닝 모델들과 Microservice 들을 기반으로 한 임베디드 시스템이다. 임베디드 시스템에서 머신 러닝 모델을 훈련하거나 복수개의 Microservice들을

동시에 실행시켜 시스템에서 결과를 얻는 등의 작업을 진행하게 될 경우, 많은 리소스가 필요하기 때문에 해당 환경에 적합하지 않다. 따라서 본 시스템에 맞는 아키텍처 설계를 적용한다. Fig. 3은 본 시스템의 아키텍처를 보여주고 있다.

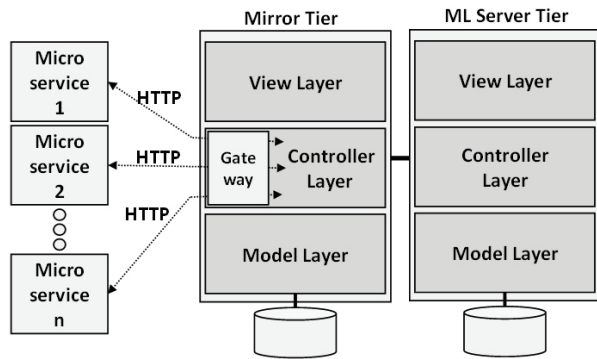


Fig. 3. Architecture of Digital Mirror System

해당 아키텍처는 Multi-Tier 아키텍처, Layered 아키텍처, Microservice Architecture를 혼합하여 적용되었다. 우선, Mirror Tier와 ML Server Tier로 구분하고 각 Tier가 하는 역할을 분리하였다. Mirror Tier는 사용자가 사용하게 되는 부분으로 사용자와 상호연동을 통해 데이터 수집 및 정보 전달 등을 한다. ML Server Tier는 수집한 데이터를 보관하고 해당 데이터를 이용하여 새로운 머신 러닝 모델을 생성 및 배포한다.

해당 시스템의 Mirror Tier는 뷰 레이어(View Layer), 컨트롤러 레이어(Controller Layer), 모델 레이어(Model Layer)로 구성되어있다. 뷰 레이어는 사용자와 시스템이 상호 작용 부분으로 사용자에게 필요한 이벤트를 입력 받고 결과를 보여준다.

컨트롤러 레이어는 뷰 레이어에서 입력된 이벤트를 받아 수행 및 결과를 전송하는 레이어이다. 뿐만 아니라 외부 Microservice와의 연결을 위한 게이트웨이(Gateway)를 포함하여 필요한 정보를 수집 및 제공한다. 게이트웨이는 HTTP 통신을 이용하여 외부 서비스에서 정보를 수집하는 통로로 사용된다. 이때, 한 외부 서비스는 컨트롤러 레이어 내부의 한 쓰레드와 연결이 된다. 외부에서 입력 받은 데이터는 해당 쓰레드 내에서 처리 및 제공된다. 컨트롤러 레이어는 뷰 레이어에서 생성된 이벤트를 받아 해당하는 쓰레드를 작동하게 되고, 각 쓰레드 내부에서 외부 서비스 또는 내부 데이터 베이스와의 연결이 필요할 경우 HTTP 또는 모델 레이어의 API등을 호출하여 필요한 정보를 수집 및 처리하게 된다. 이 레이어 내부에서는 모든 작업들이 쓰레드로 구현되어 작업 중 서로에게 영향을 미치지 않고 새로운 기능들을 추가하는데 용이하다. 이 때 컨트롤러 레이어에 연결된 Microservice들은 자체 서버 상태나 데이터 업데이트 주기 등의 이유로 실시간성이 보장될 수 없다. 더불어 게이트웨이의 쓰임에 맞춰 Microservice들과 게이트웨이의 실시간성은 고려되지 않는다. 하지만 컨트롤러 레이어에서 수집된 데이터를 이용하여 사용자에게 제공하는 과정의 경우, 요구사항에 따라 실시간적으로 정보를 제공해야

된다. 따라서 컨트롤러 레이어의 실시간성 보장만 고려된다.

모델 레이어는 데이터베이스와 연결되어 컨트롤러 레이어에서 필요한 질의(Query)를 제공한다. 해당 질의는 API 형식으로 제공된다.

ML Server Tier 또한 3가지 레이어로 구성된다. 본 Tier의 뷰 레이어도 시스템 관리자에게 현재 시스템 서버, 머신러닝 모델, 수집된 데이터의 상태를 보여준다.

본 Tier의 컨트롤러 레이어는 주로 머신 러닝 모델 품질을 위한 작업을 진행한다. Mirror Tier는 머신 러닝 모델을 직접 훈련하기에 적합하지 않은 하드웨어를 갖고 있기 때문에 Mirror Tier의 모델이 일정 정확도 이하를 보일 때 또는 일정 주기가 지났을 때 수집된 데이터를 전송하여 필요한 모델 학습을 요청한다. ML Server Tier의 컨트롤러 레이어에서는 전송된 데이터와 요청을 받아 필요한 모델을 훈련 및 평가하여 Mirror Tier의 컨트롤러 레이어에 전달하게 된다.

ML Server Tier의 모델 레이어의 경우 모델 훈련에 대한 로그, 데이터 저장 및 모델 훈련 시 제공하는 역할을 한다. 이 레이어 또한 데이터베이스와 연결이 되어 필요한 질의를 API 형식으로 제공한다.

따라서 해당 아키텍처를 이용하여 제시된 요구사항들을 해결할 수 있다. 특히, 요구 사항 중 이미지 반영 기능의 실시간성은 Mirror Tier와 ML Server Tier로 구분하면서 Mirror Tier에서 생길 수 있는 런타임 오버헤드를 해결할 수 있게 된다. 더불어, Mirror Tier의 컨트롤러 레이어를 통해 내부 기능들과 외부 Microservice들의 사용성을 높인다.

4.2 이벤트 기반의 다중 쓰레드 제어 흐름

4.1에서 제시된 아키텍처를 기반으로 하여 본 시스템은 Fig. 4와 같은 액티비티 다이어그램(Activity Diagram)이 적용된다.

해당 액티비티 다이어그램은 다수개의 쓰레드로 구성이 되어있으며 가장 왼쪽에 있는 쓰레드 1은 메인 쓰레드, 나머지 쓰레드 2~7를 서비스 쓰레드이다. 쓰레드 2는 사용자와 상호 작용을 하는 부분으로 사용자에게 다양한 방법으로 입력을 받아 해당 기능을 활성화 또는 비활성화한다. 쓰레드 3~7은 각 이벤트의 활성화 유무를 확인하여 해당 쓰레드의 작업을 진행한다. 쓰레드 3은 Personal 이벤트에 대한 쓰레드로 맞춤형 PIM을 수집 및 반영하고, 쓰레드 4는 Public 이벤트에 대한 쓰레드로 공공서비스에 대한 데이터를 수집 및 반영한다. 쓰레드 5는 Assessment 이벤트에 대한 쓰레드로 사용자 얼굴 데이터를 통해 나이와 감정을 탐지하고 반영한다. 쓰레드 6은 Evolution 이벤트에 대한 쓰레드로 현재 사용자의 최근 얼굴 이미지를 통해 예측된 감정 및 나이의 변화도를 보여준다. 쓰레드 7은 Setting 이벤트에 대한 쓰레드로 시스템 설정 기능을 제공한다.

각 쓰레드는 서로 독립적으로 작동되나, 쓰레드 간의 필요한 정보나 연산 결과 등은 공유되어야 한다. 이는 쓰레드 간 이벤트 전송 및 공유를 통해 해결한다.

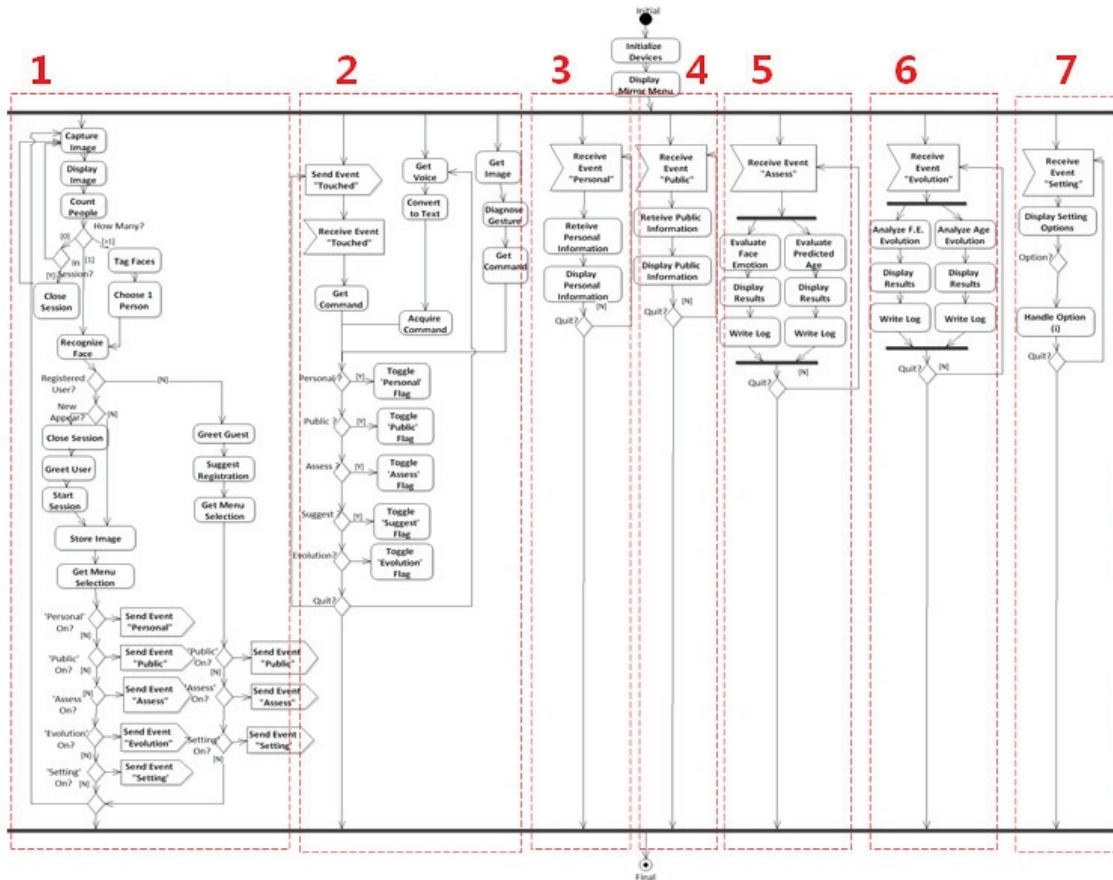


Fig. 4. Parallel Processing in Activity Diagram

본 시스템은 총 5가지(Personal, Public, Assessment, Evolution, Setting)의 이벤트가 있으며 이벤트를 통해 활성화되는 구조이다. 메인 스크린에서 해당 이벤트에 대한 토글(Toggle)을 확인하여 활성화 유무에 따라 해당 이벤트를 발생 여부를 결정한다.

Fig. 5는 본 시스템 액티비티 다이어그램의 일부로 이벤트 기반의 설계를 보여주는 부분이다.

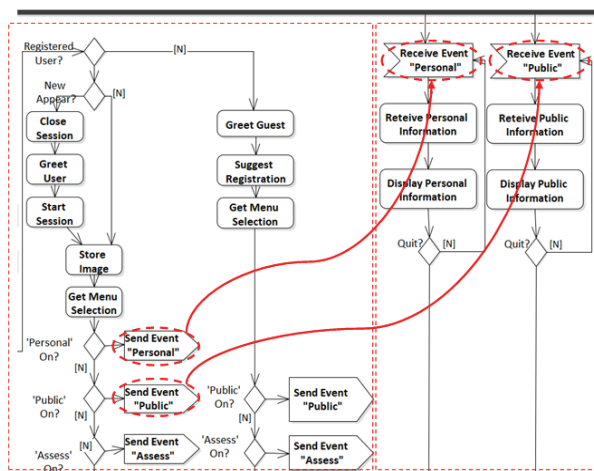


Fig. 5. Event-Driven Asynchronous Control Flow

Fig. 5의 왼쪽 부분에 있는 스크린은 메인 스크린이다. 또한 오른쪽의 서비스 스크린은 Fig. 4의 스크린 3과 스크린 4이다. 이 두 스크린은 메인 스크린에서 Personal이나 Public에 대한 이벤트가 발생할 시, 각 스크린이 해당 이벤트를 전달 받아 각 이벤트 관련 정보들을 수집 후 스크린에 디스플레이 한다. 해당 작업은 메인 스크린에서 해당 이벤트 발생을 종료할 때까지 지속된다.

Table 2는 본 시스템에서 적용한 이벤트 기반 다중 스크린이 작동하는 방식에 대한 알고리즘을 설명한다.

Table 2는 메인 스크린과 공공 정보를 처리하는 스크린의 관계를 나타낸다. 메인 스크린에서는 수집된 이미지를 통해 현재 사용자 유무와 몇 명이 있는지를 연산한다(3~6줄). 탐지된 사람의 없을 경우 이미지 수집을 반복하고(8~9줄), 탐지된 사람이 1명 초과일 경우, 사용자에게 세션 생성을 위한 사람을 선택하게 한다(10~11줄). 탐지된 사용자가 1명일 경우, 해당 사용자에 대해서 정보를 제공하게 된다(12~13줄). 선택된 사용자에 대해 신원 확인 후 (16줄), 사용자가 본 시스템에 가입이 되어 있다면 세션 생성 후 토글들의 활성화 여부를 확인한다(17~22줄). 가입이 되어있지 않다면, 세션 생성 후 해당 사용자가 사용 가능한 토글들의 활성화를 확인한다(23~26줄). 이때 Public 이벤트 토글이 활성화될 경우, Public 이벤트에 해당되는 서비스 스크린이 해당 메시지를

Table 2. Algorithm of Multi Thread based on Event

Algorithm. Algorithm of Handling Multi Thread based on Event	
Input: isActive: Boolean, whether activate the system or not, Default is True; RegisterUser: Dictionary, user information of registered MIAS System	
1:	thread main {
2:	while isActive
3:	img ← getImg()
4:	// Display Image
5:	people ← {p _n dataofdetectedperson}
6:	numPeople ← people
7:	
8:	if numPeople = 0
9:	continue
10:	else if numPeople > 1
11:	selectPerson ← p _x // Choose person x on people
12:	else if numPeople = 1
13:	selectPerson ← p ₀
14:	endif
15:	
16:	selectPerson = getIdentification() //detect Identification
17:	if selectPerson ∈ RegisteredUser
18:	// Greet and Start Session
19:	toggles ← {toggle t of functions for registered user}
20:	if toggles["Public"]
21:	SEND "Public"
22:	// Check other toggles and Send event
23:	else
24:	// Greet, suggest registration, and create session
25:	// Check toggles and Send events
26:	endif
27:	endWhile
28:	}
29:	
30:	thread publicInfo {
31:	listService ← {s Service s related to perblic information}
32:	while isActive
33:	RECEIVE "Public"
34:	img ← getCurImage() // get current image
35:	results ← {}
36:	
37:	for s in listService
38:	s.perform()
39:	results[s.name] ← s.getResult()
40:	endfor
41:	// Display result at current image
42:	endWhile
43:	}

전달받는다(33줄). 해당 기능에 대한 쓰레드의 작업이 진행되고 기능 내 서비스들의 결과를 수집하여 정보를 사용자에게 제공한다(37~41줄).

이 설계 방법을 통하여 시스템에서 제공하는 정보들의 사용성을 높이고 관리를 수월하게 할 수 있다. 그리고 쓰레드를 통해 각 기능 간 영향을 미치지 않기 때문에 실시간 디스플레이를 보장할 수 있다. 또한 Multi-Modal 기반의 사용자 상호연동을 가능하게 한다. 더불어 해당 방법을 이용하여 새로운 기능이나 상호연동 방법에 대한 높은 확장성도 보장된다.

4.3 얼굴 인식 모델 기반 사용자 세션 관리

본 시스템은 사용자의 세션을 관리할 때 사용자에게 직접적으로 입력을 받지 않는다. 따라서 Table 2의 메인 쓰레드 알고리즘과 같이 수행하는 사용자 세션 관리를 위해 사용자

유무, 찾아진 사용자의 수, 세션 유지를 위한 목표 사용자 설정, 해당 사용자의 사용 유무를 판별해야 한다.

사용자 세션 관리를 위해 사용자를 탐지하는 부분과 현재 세션을 활성화할 사용자에게 대한 신분을 탐지하는 부분에 각각 머신 러닝 모델이 요구된다. 해당 모델은 CNN(Convolutional Neural Network) 기반의 딥러닝 모델로 이미지를 입력 데이터로 사용한다. 본 시스템에서는 사용자의 신분을 확인하기 위해 세 가지의 CNN 모델을 합쳐 사용한다. Fig. 6은 해당 CNN 구조를 나타낸다.

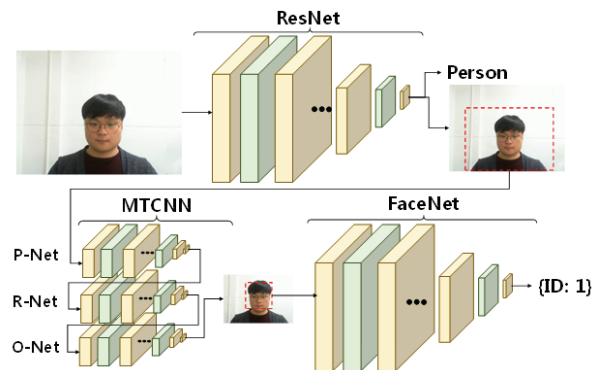


Fig. 6. CNN Structure of Classifying Identification

첫 번째로 사용되는 모델은 ResNet이다. 이 모델은 이미지 내 사물을 찾는 모델로서, 수집된 이미지에서 사람이 있는지 여부를 확인하기 위해 사용된다. 만약 수집된 이미지 내에 사람이 있는 경우 해당 사람이 있는 이미지상의 위치와 레이블을 반환하게 된다.

두 번째로 사용되는 모델은 MTCNN이다[30]. 이 모델은 이미지에서 얼굴을 찾는 모델이다. 해당 모델은 이전 단계 모델의 결과 중 사람이 있다고 예측되는 부분의 이미지를 입력으로 받는다. 이 이미지에서 얼굴과 비슷하게 생긴 부분을 찾는다. 해당 모델은 P-Net, R-Net, O-Net 3가지의 구조로 이루어져 있다. 각각의 CNN 네트워크를 통해 얼굴 형태와 유사한 부분 탐색, 얼굴에 대한 공간적 정보 확인 등을 진행한다. 그 후 해당 부분의 얼굴 여부, 얼굴의 위치를 나타내는 박스, 얼굴 내 랜드마크 위치 정보를 반환한다.

세 번째로 사용되는 모델은 FaceNet이다[31]. 해당 모델은 이전 단계 결과물을 이용하여 해당 이미지의 특징들이 어떤 사용자의 이미지와 유사도가 높은지를 평가한 후, 해당 이미지에 맞는 사용자 정보를 반환한다.

위의 모델들을 통해 사용자 유무와 사용자 신분 확인을 함으로써 사용자에게 다른 입력을 받지 않고 세션 생성 및 유지, 종료한다. 따라서 해당 설계 방법을 통해 머신 러닝 기반 자동 세션 관리를 할 수 있게 된다.

세션 관리의 경우, 사람이 이미지에 존재하지 않을 때 매 프레임마다 사용자 유무 확인을 진행하게 된다. 하지만 시스템이 한 사용자에게 세션을 생성한 후부터 일정 개수의 프레임만 세션 관리에 사용한다. 즉, 사용자가 해당 거울을 벗

어나 세션이 제거될 수 있을 시간을 t 초라 하였을 때, 세션 생성 후 사용자 유무와 신분 확인은 t 초 마다 진행하게 된다. 이를 통해서 불필요한 사용자 신분 확인을 줄일 수 있다.

또한 모델이 특정 사람들에 대해 정확도가 낮게 나올 수 있다. 이는 모델 훈련 시, 그 사람에 대한 적은 수의 데이터를 이용하거나 해당 사람의 이미지에 대해 모델이 최적화가 되어있지 않은 상태이다. 따라서 이러한 경우, 해당 사용자에 대해 이미지를 추가 수집하여 하이퍼파라미터(Hyperparameters) 최적화, 데이터 정제화(Data Purification)을 한 후 모델을 최적화한다.

4.4 머신 러닝 기반의 얼굴 분석 모델

4.3 절의 설계 방법에서 시스템은 사용자의 얼굴 데이터를 이용하여 세션을 유지하였다. 이 얼굴 데이터를 이용하여 얼굴에서 드러나는 시각적인 정보를 수집할 수 있다. 예를 들어, 해당 사용자의 시각적으로 보이는 나이[32-36]와 감정[37-39] 등을 예측할 수 있게 된다. 해당 모델의 합성곱 레이어(Convolutional Layer)와 풀링 레이어(Pooling Layer)를 이용하여 얼굴 이미지에서 시각적 정보를 추출한다. 시각적 정보가 레이어들을 통해 추출이 된 후, 완전 연결 레이어(Fully Connected Layer)를 이용하여 추출된 정보의 수를 줄이면서 대표적인 정보를 추리게 된다. 출력 레이어(Output Layer)의 뉴런의 개수를 분류할 나이의 개수와 일치하게 설정한 후, 추려진 정보들을 통해 입력된 얼굴이 각 나이로 예측될 정도를 반환하게 된다. 해당 결과 중 가장 높은 예측 정도를 보이는 종류를 해당 얼굴에서 얻어진 값으로 설정한다. Fig. 7은 얼굴 이미지를 통해 나이를 판단하는 CNN 모델의 구조를 나타낸다.

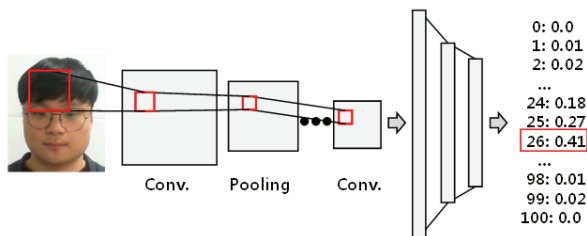


Fig. 7. CNN Structure of Estimating Age

세션 관리를 위해 탐지된 얼굴 이미지를 입력데이터를 사용한다. 이 데이터에서 드러나는 시각적 정보를 추출하기 위해 합성곱 레이어와 풀링 레이어를 반복하여 사용한다. 이후 완전 연결 레이어를 통해 추출된 정보의 수를 분류할 나이를 고려하여 101개로 반환하게 된다. 각 데이터의 값은 해당 얼굴 이미지가 각 나이로 예측될 정도를 나타낸다. Fig. 7에서는 26세의 예측 정도가 가장 높으므로 해당 얼굴은 26세로 설정이 된다.

나이 판단하는 CNN 모델과 감정을 판단하는 CNN 모델은 얼굴 이미지를 통해 특정 결과를 예측하기 때문에 유사한 CNN 구조를 가지게 된다. 본 시스템에서는 얼굴 이미지의 시각적 정보를 추출하기 위해서 ResNet[40]을 적용하였다. ResNet 모델은 다른 CNN 기반의 이미지 분류 모델들 보다

많은 레이어를 갖고 있으며 높은 정확도로 분류할 수 있어 적용되었다.

따라서 각 모델은 입력된 얼굴의 시각적 정보를 사용하여 각 감정이나 나이로 예측이 되었을 때 신뢰할 수 있는 정도를 반환하게 된다. 이 값을 이용하여 가장 높은 값을 갖고 있는 나이와 감정을 해당 얼굴에서 판단할 수 있는 나이와 감정으로 판단한다. 이 과정을 통해 항상 모델을 높은 정확도를 보장할 수 있게 하고 머신 러닝 모델을 이용한 Microservice를 제공한다.

4.5 Microservice 적용 및 관리

본 시스템에서 제공하는 기능들은 각 기능들의 특성에 따라 묶어 쓰레드로 제공된다. 이 때 각 쓰레드 내부의 기능들은 외부에서 제공되는 Microservice이거나 시스템에 연결된 카메라, 마이크, 터치 스크린을 통해 감지된 결과를 가지고 예측 및 판단하는 기능 등 다양한 기능을 가진다. Fig. 8은 본 시스템에서 Public과 Personal 기능을 위해 기상청과 Google에서 Microservice 적용하는 부분을 나타낸다.

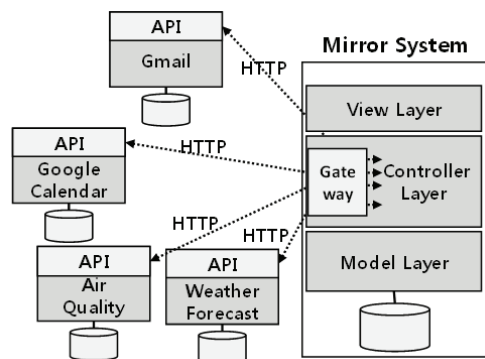


Fig. 8. Application of Microservices

본 시스템의 Mirror 티어가 가지고 있는 게이트웨이는 각 기능이 실행되었을 때 필요한 Microservice에 대한 연결을 활성화하고 데이터를 수집 및 스크린에 반영하게 된다. Public 기능이 활성화될 경우, Public 기능 중 일기예보 기능을 위해 기상청에서 제공하는 일기예보에 대한 Microservice에 접근한다. Personal 기능이 활성화될 경우, 해당 사용자의 이메일과 일정을 보여주기 위해 Google에서 제공하는 이메일 정보와 일정 정보에 관련된 Microservice에 접근한다.

Table 3은 본 시스템의 Public 기능 중 일기 예보의 Microservice를 접근 및 데이터 수집에 대한 코드이다.

해당 코드는 기상청에서 일기예보에 대한 정보를 수집하기 위한 코드로 일기예보 API를 접근하기 위한 키와 API 객체 및 위치를 설정한다(3~7줄). Public 기능의 쓰레드를 정의하는 메소드(16~27줄), 쓰레드를 생성하는 메소드(9~10줄), 쓰레드를 종료하는 메소드(12~14줄)로 구성된다. 이 쓰레드는 일기예보 Microservice에서 현재 설정된 지역의 최저 기온, 최고기온과 구체적인 날씨를 가져온다(18~23줄). 그 후, 이 정보들을 스크린에 반영하기 위해 설정한다(24~25줄).

Table 3. Accessing Microservice on Public Service

```

Code. Code of Accessing Microservice on Public Function
1: class Public:
2:     # Initialize key
3:     weather_key = '...'
4:
5:     owm = OWM(weather_key)
6:     city = 'Seoul,kr'
7:     obs = owm.weather_at_place(city)
8:
9:     def turn_on(self):
10:         self.activate_public()
11:
12:     def turn_off(self):
13:         # Reset variables
14:         ...
15:
16:     def activate_public(self):
17:         def th():
18:             self.w = self.obs.get_weather()
19:             self.s = self.w.get_detailed_status()
20:             self.temp = self.w.get_temperature(unit='celsius')
21:             self.c = self.temp['temp']
22:             self.min_temp = self.temp['temp_min']
23:             self.max_temp = self.temp['temp_max']
24:             self.temp_info = 'Temperature: Current {C}, Max {C}, Min
{|C'.format(round(self.c, 1), self.max_temp, self.min_temp)
25:             self.weather_info = 'Weather: {}'.format(self.s.title())
26:
27:         threading.Thread(target=th).start()
    
```

Mirror tier의 컨트롤러 레이어 구성과 서비스 쓰레드들을 통해 독립적인 Microservice 적용 및 관리가 가능하고 높은 사용성을 보장한다.

4.6 Multi-Modal 기반 사용자 상호작용

본 시스템을 사용자가 제어하기 위한 상호연동 또한 서비스 쓰레드로 구성되게 된다. Fig. 9는 전체 액티비티 다이어그램 중 상호연동 부분을 담당하는 쓰레드 2를 보여준다.

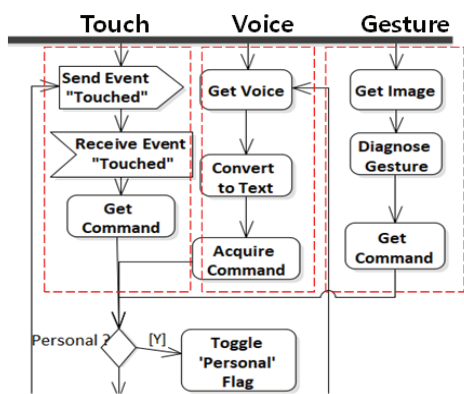


Fig. 9. User Interaction Based on Multi-Modal

위의 액티비티 다이어그램은 Multi-Modal 기반 상호연동을 나타낸다. Fig. 9의 왼쪽 부분은 터치 기반 상호연동에 대한 부분으로 사용자로부터 이벤트를 전달받아 해당 명령을 획득 후 수행한다. Fig. 9의 가운데 부분은 음성 인식 기반 상호연동을 나타낸다. 시스템에서 사용자의 목소리를 텍스트

로 변경 후 해당 텍스트에서 해당 명령을 획득 후 수행한다. Fig. 9의 오른쪽 부분은 제스처 기반 상호연동을 나타내며 현재 수집된 이미지에서 사용자의 행동을 파악하여 사용자가 어떤 명령을 요구하는 지 확인한다. 확인된 명령에 따라 사용자가 필요한 기능에 대한 토글을 활성화 및 비활성화 하게 된다.

Fig. 9의 Modal들은 입력 받은 데이터를 통해 데이터 분석 후 사용자가 원하는 명령을 찾게 되는 구조이다. 현재 제스처 기반의 상호연동은 구현이 되지 않았지만, 상호연동 관련 쓰레드에 적용되어 명령 파악 및 토글 활성화를 할 수 있음을 보인다. 이를 통해 다양한 상호작용들을 통해 같은 기능을 실행할 수 있으며, 본 구조를 통해 새로운 상호작용 기능을 추가할 수 있다.

5. 시스템 구현

5.1 시스템 구현 환경

본 시스템은 Python 언어를 이용하여 작성되었다. 25개의 클래스들과 약 4,000줄의 코드로 구성되어 있다. 시스템의 사용자 인터페이스를 구현하기 위해 Tkinter GUI 툴킷(Toolkit)을 이용하였다. Micro-service에서는 날씨, 대기 정보, 사용자의 일정 및 이메일 등의 외부 정보를 가져오기 위해 OpenWeather-Map API와 pypubdata API, Google Calendar API, Gmail API를 사용하였다. 머신 러닝 기반의 사용자 얼굴 분석을 위해 Keras와 TensorFlow를 사용하였고, 사용자 상호연동 중 음성 인식 기반의 상호연동을 위해 Microsoft Azure voice recognition이 사용되었다.

5.2 실험 시나리오 및 결과

1) 머신 러닝 기반 자동 세션 관리

본 시스템의 자동 세션관리는 수집된 이미지에서 사용자 탐색, 사용자 신분 탐지, 사용자의 회원가입 유무 확인, 해당 사용자의 사용 여부 확인 총 4단계로 구분된다. 따라서 본 실험에서는 사용자가 시스템을 사용할 때 사용자의 세션이 관리되는 과정을 확인한다.

Fig. 10은 세션이 생성되고 해당 사용자가 시스템에 가입되어있는 경우의 화면을 보여준다.

Fig. 10에서는 사용자가 수집된 이미지에 존재하기 때문에



Fig. 10. Situation of Session-on in System

사용자를 찾는 사용자의 신원 판단과 시스템에서 가입이 되어있다 라는 것을 확인된 후에 해당 사용자의 이름을 담은 인사를 띄운다. 또한 해당 사용자는 시스템에 가입되어 있으므로 추가 메뉴를 디스플레이 한다.

2) 다중 스크레드를 통한 Microservice 적용

본 시스템은 다수의 Microservice를 제공 및 기능에 따라 분류하여 제공한다. 따라서 본 실험에서는 세션 생성 후 해당 사용자가 기능들을 실행시켜 각각의 service들이 실행되는 것을 확인한다. Fig. 11은 각 기능들의 토글을 활성화했을 때의 디스플레이 모습이다.

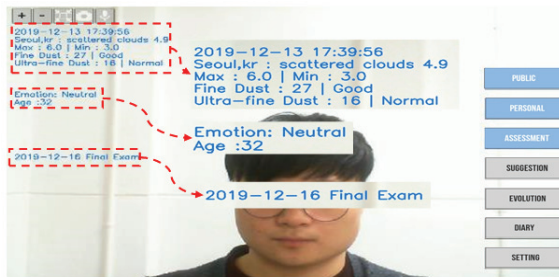


Fig. 11. Pop-up Windows Following Selected Buttons

Fig. 11에서는 Public과 Personal, Assessment 기능을 활성화하였다. Public에는 현재 시간, 날씨, 대기 정보를 반환하는 Microservice들을, Personal에는 현재 사용자가 등록해둔 계정에 일정 중 다가오는 일정을 반환하는 Microservice를 갖고 있다. 또한 Assessment의 경우, 해당 사용자의 얼굴에서 얻을 수 있는 나이 정보와 감정 정보를 반환하는 머신 러닝 기반 모델을 갖고 있다. 따라서 이 3가지의 기능을 활성화하였을 때, 각 기능의 결과들을 확인할 수 있었으며 서로 독립적이게 활동하는 것을 확인할 수 있었다.

3) 머신 러닝 기반 사용자 얼굴 분석 모델 적용

본 시스템의 기능 중 Assessment 기능에는 사용자 얼굴을 이용하여 나이와 감정을 예측하는 모델이 적용이 되었다. 현재 사용자에 대해 해당 모델이 실시간으로 예측하는 지 여부와 해당 모델들의 정확도를 평가한다. 세션 유지를 위해 탐지된 얼굴 이미지를 이용하여 나이와 감정을 예측하였을 때 Fig. 12와 같은 결과를 얻었다.

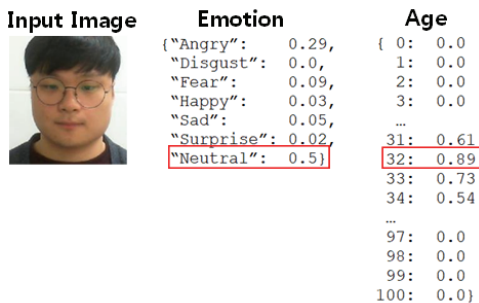


Fig. 12. Results of ML Model Related on Session Management

Fig. 12 좌측에 있는 얼굴 이미지를 두 모델에 입력하였을 때, 각 모델은 해당 얼굴 이미지가 모델이 분류할 수 있는 나이와 감정의 종류에 대해 신뢰도를 반환하였다. 나이의 경우, 32세일 때 가장 높은 신뢰도를 보였다. 감정의 경우 무표정일 때 가장 높은 신뢰도를 보였다.

5.3 실험 평가

1) 머신 러닝 기반 자동 세션 관리

자동 세션 관리를 위해 사용한 사용자 신원 판단 모델을 평가한다. 해당 사용자 신원 판단 모델은 가입한 사용자들의 신원과 가입되지 않은 사람을 포함하여 분류를 하게 된다. 따라서 전체 신원의 리스트를 iw, 가입된 사용자의 리스트를 ir, 가입되지 않은 사용자의 리스트를 in이라 하고 Num(x)함수를 리스트 x의 인자 개수를 반환하는 함수라 할 때 Equation (1)을 만족한다.

$$\begin{aligned}
 Num(iw) &= Num(ir) + Num(in) \\
 Num(in) &= 1
 \end{aligned}
 \tag{1}$$

전체 신원의 수는 가입된 사용자의 수와 가입되지 않은 사용자의 수를 합한 값이다. 이 때 모든 가입하지 않은 사람은 해당 신원을 가입되지 않음(None)으로 분류하므로 가입되지 않은 사용자의 수는 항상 1이다.

해당 모델을 평가하기 위해 Accuracy Equation (2)와 Precision Equation (3) 두 가지의 평가 방법을 사용하였다.

$$Accuracy = \frac{TP + TN}{TP + FN + FP + FN}
 \tag{2}$$

$$Precision = \frac{TP}{TP + FP}
 \tag{3}$$

Fig. 13은 해당 모델을 Equation (2)와 Equation (3)을 적용한 결과를 나타낸다.

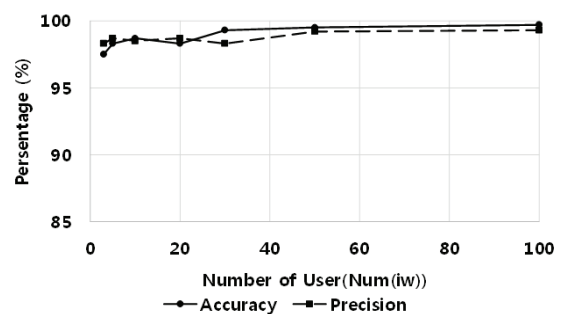


Fig. 13. Performance of User Classification Model Considering Number of User

각 Num(iw)의 값이 3, 5, 10, 20, 30, 50, 100인 경우에 대해 실험을 진행하였다. iw 내의 각 사람에 대한 데이터는 총 50개의 이미지로 구성되어 되어있다. 해당 데이터셋은 Num(iw)에 따라 150, 250, 500, 1,000, 1,500, 2,500, 5,000 이 된다. 각 데이터는 80%와 20%로 나누어 각각 훈

련 데이터(Training Set)와 테스트 데이터(Test Set)로 할당된다. 해당 모델은 훈련에 사용된 데이터 내 사용자의 수와 관계없이 항상 98%이상의 정확도를 나타냈다.

해당 결과를 통해 새로 추가되는 사용자가 늘어나도 각기 다른 사용자들을 구분할 수 있다는 것을 확인하였다. 특히 Precision이 높은 값을 보이면서, 해당 모델을 사용하여 사용자의 신원을 분류할 때 해당 사람이 정확하게 분류될 수 있다는 것을 보장한다.

그러나 해당 모델들의 경우 약 2% 정도의 실패율을 보인다. 해당 모델들을 적용할 때 결과의 신뢰도 평가 또한 진행하게 되고 신뢰도가 낮게 나왔을 경우 해당 사용자의 이미지를 추가 수집하여 모델을 최적화한다. 이 과정을 반복하여 실제 시스템에서 사용자의 신분이 잘못 식별되는 과정을 최소화한다.

2) 다중 쓰레드를 통한 Microservice 적용

본 시스템에서 제공되는 7개의 쓰레드가 연산 과정에 생기는 지연으로 인해 다른 쓰레드의 연산, 특히 메인 쓰레드에 영향을 미치는 지 확인한다.

Fig. 14는 일반적인 상황(Normal)과 특정 서비스 쓰레드가 지연이 된 상황(Delay)의 소요시간을 나타낸다.

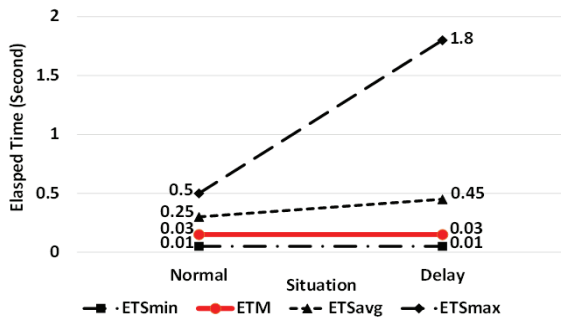


Fig. 14. Comparison of Elapsed Time Threads

메인 쓰레드에서 수집된 이미지에 서비스 쓰레드의 연산 결과를 추가하여 스크린에 반영하는데 소요 시간을 ETM이라 한다. 서비스 쓰레드의 경우, 모든 서비스 쓰레드에 대해 소요시간을 계산한다. 그 중 소요시간이 가장 긴 시간을 ETS_{max}로 가장 짧은 시간을 ETS_{min}, 모든 서비스 쓰레드의 평균 소요시간을 ETS_{avg}라 한다.

이때 메인 쓰레드는 0.03초로 0.03초마다 이미지를 1장씩 스크린에 반영하였다. 서비스 쓰레드 중 쓰레드 4번이 해당 서버 관련한 이슈로 지연이 발생하여 데이터를 수집하는 데에 1.8초가 소요되었다. 다른 서비스 쓰레드는 일반적인 상황과 같은 시간을 소요하여 평균적으로 0.45초를 소요하였다. 하지만 이경우도 메인 쓰레드의 소요시간은 0.03초로 일반적인 상황과 같은 결과를 보였다.

따라서 다중 쓰레드를 통해 각 Microservice를 구현한 후 기능을 활성화시켰을 때 서로 독립적으로 작동되는 것을 확인할 수 있었다. 여러 개의 Microservice를 포함하고 있는 기능에서는 각 Microservice가 해당 서비스의 서버 이상 또

는 연산 지연 등에도 다른 Microservice에 영향을 미치지 않고 디스플레이 되는 것을 확인할 수 있었다.

3) 머신 러닝 기반 사용자 얼굴 분석 모델 적용

사용자 얼굴을 이용하여 나이와 감정을 예측한 모델을 평가한다. 나이에측 모델은 UTKFace 데이터셋을 이용하여 학습하였다. 해당 데이터셋은 0세부터 116세까지의 사람에 대해 약 20,000개의 얼굴 이미지로 구성되어 있다. 이 데이터셋을 사용하여 0세에서 100세까지 분류하는 모델을 생성하였다. 해당 데이터 중 80%는 훈련 데이터로 20%는 테스트 데이터로 사용되었다.

해당 모델을 평가하기 위해 사용자의 나이대를 고려한 나이와 차이를 계산하는 평가를 하였다. 해당 모델을 평가하기 위해 Equation (4)의 수식을 적용하였다.

$$diffAge_n = \frac{\sum_{i=0}^m (age_{real} - age_{dti})}{m} \quad (4)$$

실제 나이를 age_{real}이라 하고 모델을 통해 예측된 나이를 age_{dti} 하였을 때 실제 나이가 속한 나이대인 n의 오차율은 해당 나이대의 데이터에서 예측된 모든 나이의 차이를 평균 낸 값이다. 해당 결과를 diffAge_n이라 하고 예측되는 나이의 범위가 1세부터 100세까지 이므로 총 10개의 diffAge값이 존재한다. Fig. 15는 Equation (4)를 사용하여 얻은 결과를 그래프로 표현한 것이다.

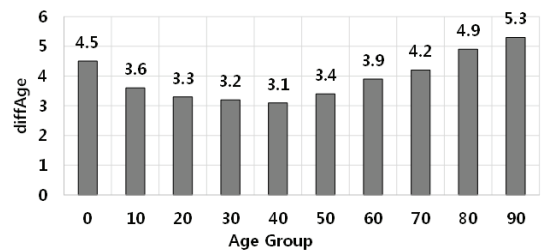


Fig. 15. Difference of Age Considering Age Group

전체적인 나이 그룹에서 diffAge는 3~5세 정도 차이가 났다. 훈련 데이터가 가장 많은 나이 대인 20~40대의 경우, 3.1~3.3세의 차이를 보였고 데이터가 적은 80~90대의 경우, 4.9~5.5세의 차이를 보였다.

감정을 예측하는 모델은 FER2013 데이터셋을 사용하였다. 해당 데이터셋은 약 29,000여개의 얼굴 이미지를 화남 (Angry), 싫음(Disgust), 두려움(Fear), 행복함(Happy), 슬픔(Sad), 놀람(Surprise), 무표정(Neutral) 분류하였다. 해당 데이터는 각 레이블을 고려하여 80%를 훈련 데이터와 20%를 테스트 데이터로 나뉘어 사용한다. 따라서 훈련 데이터는 23,000여 개, 테스트 데이터는 6,000여 개의 데이터를 갖고 있다.

해당 데이터셋을 사용하여 입력된 7가지의 감정으로 분류하는 모델을 생성하였다. 해당 모델을 평가하기 위해 Equation (2)를 이용한 Confusion Matrix를 사용하였다. Table 4는 해당 결과를 보여준다.

Table 4. Confusion Matrix of Emotion Detection

Predicted Value \ Real Value	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
Angry	0.73	0.12	0.14	0.01	0	0	0
Disgust	0.1	0.75	0.12	0	0	0.03	0
Fear	0.1	0.08	0.78	0.01	0.01	0.02	0
Happy	0.04	0.01	0	0.83	0	0.12	0
Sad	0	0.01	0.05	0	0.79	0	0.15
Surprise	0.02	0	0.01	0.11	0.01	0.85	0
Neutral	0.01	0.01	0.01	0.02	0.12	0	0.83

테스트에서 사용된 대부분의 데이터는 다소 높은 정확도로 예측이 되었다. 가장 낮은 정확도를 보인 것은 화남으로 73%의 정확도를 보였으며 가장 높은 정확도를 보인 감정은 놀람으로 85%의 정확도를 보였다.

Fig. 16은 해당 모델을 이용하여 성별에 따른 감정을 예측한 결과를 평가한 그래프이다.

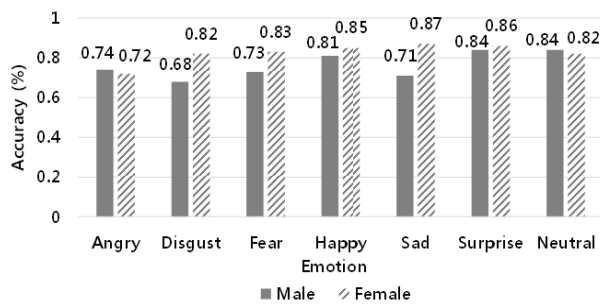


Fig. 16. Performance of Detecting Emotion Considering Gender

두 성별의 결과 차이가 크게 나지 않으나, 전체적으로 남성의 데이터가 여성의 데이터보다 정확도가 떨어졌다. 차이가 가장 큰 감정은 슬픔으로 16%의 차이를 보였고, 차이가 가장 적은 감정은 화남, 놀람, 무표정으로 2%의 차이를 보였다. 특히, 무표정의 경우, 남성의 결과가 여성의 결과보다 높은 것을 확인할 수 있었다.

본 실험을 통해 높은 정확도의 나이 및 감정 예측을 보였고, 본 시스템에 적용된 머신 러닝 모델을 통해 높은 정확도의 예측결과를 얻을 수 있음을 보였다.

6. 결 론

디지털 거울 시스템은 아날로그식 거울의 기능인 실시간 반영을 보장하면서 사용자에게 필요한 정보를 제공해야 한다. 따라서 머신 러닝 모델과 Microservice를 이용한 정보 제공이 필요하다. 더불어 사용자 편의를 고려하여 자동 세션 관리 Multi-Modal 상호작용이 보장되어야 한다.

본 논문에서는 머신 러닝과 Microservice 기반의 지능형 디지털 거울 시스템을 제안하였다. 또한 본 시스템에서 생길 수 있는 요구사항들을 제시했다. 이러한 문제점들을 해결할

수 있는 시스템 설계를 제안하고 해당 설계를 따라 시스템을 제작하여 설계의 유효성을 검증하였다. 또한 제작된 시스템의 성능을 평가함으로써 각 기능들이 높은 품질을 보이는 것을 확인하였다.

References

- [1] B. Cvetkoska, N. Marina, D. C. Bogatinoska, and Z. Mitreski, "Smart mirror E-health assistant — Posture analyze algorithm proposed model for upright posture," In *proceedings of IEEE EUROCON 2017. -17th International Conference on Smart Technologies*, Ohrid, Macedonia, pp.507-512, Jul. 2017.
- [2] K. Jin, X. Deng, Z. Huang, and S. Chen, "Design of the smart mirror based on raspberry PI," In *Proceedings of 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC 2018)*, Xi'an, China, pp.1919-1923, May 2018.
- [3] Y. Sun, L. Geng, and K. Dan, "Design of smart mirror based on raspberry PI," In *Proceedings of International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS 2018)*, Xiamen, China, pp.77-80, Jan. 2018.
- [4] M. M. Yusri, S. Kasim, R. Hassan, Z. Abdullah, H. Ruslai, K. Jahidin, and M. S. Arshad, "Smart mirror for smart life," *6th ICT International Student Project Conference (ICT-ISPC 2017)*, Skudai, Malaysia, pp.1-5, May 2017.
- [5] V. Singh and D. Singh, "Smart Interactive Mirror Display," In *Proceedings of International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon 2019)*, Faridabad, India, pp.140-145, Feb. 2019.
- [6] P. Mathivanan, G. Anbarasan, A. Sakthivel, and G. Selvam, "Home automation using smart mirror," In *Proceedings of IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)*, Pondicherry, India, pp.1-4, Mar. 2019.
- [7] R. M. A. Latif, L. B. Imran, M. Farhan, M. J. Bah, G. Ali, and Y. A. Abid, "Real-time simulation of IoT based smart home live mirror using WSN," In *Proceedings of International Conference on Frontiers of Information Technology (FIT 2019)*, Islamabad, Pakistan, pp.49-54, Dec. 2019.
- [8] O. Gomez-Carmona and D. Casado-Mansilla, "SmiWork: An interactive smart mirror platform for workplace health promotion," In *Proceedings of 2nd International Multi-disciplinary Conference on Computer and Energy Science (SpliTech 2017)*, Split, Croatia, pp.1-6, Jul. 2017.
- [9] W. Dai, P. Wang, W. Sun, X. Wu, H. Zhang, V. Vyatkin, and G. Yang, "Semantic integration of plug-and-play software components for industrial edges based on microservices," *IEEE Access*, Vol.7, pp.125882-125892, Aug. 2019.

- [10] J. Innerbichler, S. Gonul, V. Damjanovic-Behrendt, B. Mandler, and F. Strohmeier, "NIMBLE collaborative platform: Microservice architectural approach to federated IoT," In *Proceedings of Global Internet of Things Summit (GloTS 2017)*, Geneva, Switzerland, pp.1-6, Jun. 2017.
- [11] S. K. Datta and . Bonnet, "Next-generation, data centric and end-to-end iot architecture based on microservices," In *Proceedings of IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia 2018)*, Jeju, South Korea, pp.1-4, Jun. 2018.
- [12] M. A. Jarwar, S. Ali, and I. Chong, "Exploring Web Objects enabled Data-Driven Microservices for E-Health Service Provision in IoT Environment," In *Proceedings of International Conference on Information and Communication Technology Convergence (ICTC 2018)*, pp.112-117, Oct. 2018.
- [13] M. De Sanctis, H. Muccini, and K. Vaidhyanathan, "Data-driven adaptation in microservice-based IoT architectures," In *Proceedings of IEEE International Conference on Software Architecture Companion (ICSA-C 2020)*, Salvador, Brazil, pp.59-62, Mar. 2020.
- [14] M. A. A. Mostafa and A. M. Khater, "Horizontal offloading mechanism for IoT application in fog computing using microservices case study: Traffic management system," In *Proceedings of IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT 2019)*, Amman, Jordan, pp.640-647, Apr. 2019.
- [15] E. Al-Masri, "Enhancing the microservices architecture for the internet of things," In *Proceedings of IEEE International Conference on Big Data (Big Data 2018)*, WA, USA, pp.5119-5125, Dec. 2018.
- [16] C. Lai, F. Boi, A. Buschetti, and R. Caboni, "IoT and microservice architecture for multimobility in a smart city," In *Proceedings of 7th International Conference on Future Internet of Things and Cloud (FiCloud 2019)*, Istanbul, Turkey, pp.238-242, Aug. 2019.
- [17] W. Kang and D. Kim, "DeepRT: A predictable deep learning inference framework for IoT devices," In *Proceedings of IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI 2018)*, FL, USA, pp.279-280, Apr. 2018.
- [18] J. Tang, D. Sun, S. Liu, and J. L. Gaudiot, "Enabling deep learning on IoT devices," *Computer*, Vol.50, No.10, pp.92-96, Oct. 2017.
- [19] S. Durga, R. Nag, and E. Daniel, "Survey on machine learning and deep learning algorithms used in internet of things (IoT) healthcare," In *Proceedings of 3rd International Conference on Computing Methodologies and Communication (ICCMC 2019)*, Erode, India, pp.1018-1022, Mar. 2019.
- [20] M. Song, K. Zhong, J. Zhang, Y. Hu, D. Liu, W. Zhang, J. Wang, and T. Li, "In-Situ AI: Towards autonomous and incremental deep learning for IoT systems," *IEEE International Symposium on High Performance Computer Architecture (HPCA 2018)*, Vienna, Austria, pp.92-10, Feb. 2018.
- [21] M., Jin, K., Park, and M. Song, "An intelligent CMOS image sensor with a deep learning algorithm for smart internet of things," In *Proceedings of 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 2019)*, Jeju, South Korea, pp.1-2, Jun. 2019.
- [22] S. Yao, Y. Zhao, A. Zhang, S. Hu, H. Shao, C. Zhang, L. Su, and T. Abdelzaher, "Deep learning for the internet of things," *Computer*, Vol.51, No.5, pp.32-41, May 2018.
- [23] J. S. Shin, U. T. Kim, D. K. Lee, S. J. Park, S. J. Oh, and T. J. Yun, "Real-time vehicle detection using deep learning scheme on embedded system," In *Proceedings of Ninth International Conference on Ubiquitous and Future Networks (ICUFN 2017)*, Milan, Italy, pp.272-274, Jul. 2017.
- [24] D. Gold and D. Sollinger, "SmartReflect: A modular smart mirror application platform," In *Proceedings of IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON 2016)*, Vancouver, Canada, pp.1-7, Oct. 2016.
- [25] M. Ghazal, T. Al Hadithy, Y. Al Khalil, M. Akmal, and H. Hajjdiab, "A mobile-programmable smart mirror for ambient IoT environments," In *Proceedings of 5th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW 2017)*, Prague, Czech Republic, pp.240-245, Aug. 2017.
- [26] A. Johri, S. Jafri, R. N. Wahi, and D. Pandey, "Smart mirror: A time-saving and affordable assistant," In *Proceedings of 4th International Conference on Computing Communication and Automation (ICCCA 2018)*, Greater Noida, India, pp.1-4, Dec. 2018.
- [27] A. C. Njaka, N. Li, and L. Li, "Voice controlled smart mirror with multifactor authentication," In *Proceedings of IEEE International Smart Cities Conference (ISC2 2018)*, MO., USA, pp.1-8, Sept. 2018.

[28] R. Nadaf, and V. Bonal, "Smart mirror using raspberry pi as a security and vigilance system," In *Proceedings of 3rd International Conference on Trends in Electronics and Informatics (ICOEI 2019)*, Tirunelveli, India, pp.360-365, Apr. 2019.

[29] P. Henriquez, B. J. Matuszewski, Y. Andreu-Cabedo, L. Bastiani, S. Colantonio, G. Coppini, M. D'Acunto, R. Favilla, D. Germanese, D. Giorgi, P. Marraccini, M. Martinelli, M.-A. Morales, M. A. Pascali, M. Righi, O. Salvetti, M. Larsson, T. Strömberg, L. Randeberg, A. Bjorgan, G. Giannakakis, M. Padiaditis, F. Chiarugi, E. Christinaki, K. Marias, and M. Tsiknakis, "Mirror mirror on the wall... an unobtrusive intelligent multisensory mirror for well-being status self-assessment and visualization," *IEEE Transactions on Multimedia*, Vol.19, No.7, pp.1467-1481, Jul. 2017.

[30] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multi-task cascaded convolutional networks," *IEEE Signal Processing Letters*, Vol.23, No.10, pp.1499-1503, Oct. 2016.

[31] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015)*, MA, USA, pp.815-823, Jun. 2015.

[32] R. Rothe, R. Timofte, and L. Van Gool, "Deep expectation of real and apparent age from a single image without facial landmarks," *International Journal of Computer Vision*, Vol. 126, pp.144-157, Apr. 2018.

[33] Z. Hu, Y. Wen, J. Wang, M. Wang, R. Hong, and S. Yan, "Facial Age Estimation With Age Difference," *IEEE Transactions on Image Processing*, Vol.26, No.7, pp.3087-3097, Jul. 2017.

[34] K. H. Liu and T. J. Liu, "A structure-based human facial age estimation framework under a constrained condition," *IEEE Transactions on Image Processing*, Vol.28, No.10, pp.5187-5200, Oct. 2019.

[35] R. Jana and A. Basu, "Automatic age estimation from face image," In *Proceedings of International Conference on Innovative Mechanisms for Industry Applications (ICIMIA 2017)*, Bangalore, India, pp.87-90, Feb. 2017.

[36] M. T. B. Iqbal, M. Shoyab, B. Ryu, M. Abdullah-Al-Wadud, and O. Chae, "Directional age-primitive pattern (DAPP) for human age group recognition and age estimation," *IEEE Transactions on Information Forensics and Security*, Vol.12, No.11, pp.2505-2517, Nov. 2017.

[37] A. M. Cadayona, N. M. S. Cerilla, D. M. M. Jurilla, A. K. D. Balan, and J. C. de Goma, "Emotional state classification: An additional step in emotion classification through face detection," In *Proceedings of IEEE 6th International Conference on Industrial Engineering and Applications (ICIEA 2019)*, Tokyo, Japan, pp.667-671, Apr. 2019.

[38] H. Zhang, A. Jolfaei, and M. Alazab, "A face emotion recognition method using convolutional neural network and image edge computing," *IEEE Access*, Vol.7, pp. 159081-159089, Oct. 2019.

[39] M. S. Hossain and G. Muhammad, "An emotion recognition system for mobile applications," *IEEE Access*, Vol.5, pp. 2281-2287, Feb. 2017.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, NV, USA, pp.770-778, Jun. 2016.



송 명 호

<https://orcid.org/0000-0001-8581-3382>
 e-mail : songmho@gmail.com
 2017년 연세대학교 컴퓨터공학부(학사)
 2020년 송실대학교 컴퓨터학과(석사)
 2020년~현 재 송실대학교
 소프트웨어학과 박사과정

관심분야 : 머신 러닝 기반 자율 제어 시스템(Machine Learning based Autonomous System), 의료 영상 분석 시스템 (People Safety System)



김 수 동

<https://orcid.org/0000-0001-7585-1801>
 e-mail : sdkim777@gmail.com
 1984년 Northeast Missouri State
 University 전산학(학사)
 1988년 The University of Iowa
 전산학(석사)

1991년 The University of Iowa 전산학(박사)
 1991년 ~ 1993년 한국통신 연구개발단 선임연구원
 1994년 ~ 1995년 현대전자 소프트웨어연구소 책임연구원
 1995년 ~ 현 재 송실대학교 소프트웨어학과 교수
 관심분야 : SW Architecture, Machine Learning Systems, Medical Analytics Software