

실시간 미니드론 카메라 영상을 기반으로 한 얼굴 인식 시스템 개발

김성호
상지대학교 컴퓨터공학과 교수

Development of Face Recognition System based on Real-time Mini Drone Camera Images

Sung-Ho Kim
Professor, Department of Computer Science and Engineering, Sangji University

요약 본 논문에서는 미니 드론을 조종하면서 드론에 부착된 카메라가 촬영하는 영상을 실시간으로 받아들이며 특정인의 얼굴을 인식하여 확인시켜주는 시스템 개발 방법론을 제안한다. 본 시스템의 개발을 위해서는 OpenCV, Python 관련 라이브러리 및 드론 SDK 등을 사용한다. 실시간 드론 영상으로부터 특정인의 얼굴 인식 비율을 높이기 위해서는 딥러닝 기반의 얼굴 인식 알고리즘을 사용하며 특히 Triples 원리를 활용한다. 시스템의 성능을 확인하기 위해 저자 얼굴을 기준으로 30회 동안 얼굴 인식 실험을 수행한 결과 약 95% 이상의 인식률을 보여주었다. 본 논문의 연구 결과물은 관광지, 축제 행사장 등에서 특정인을 드론으로 빠르게 찾기 위한 목적으로 사용할 수 있을 것으로 판단된다.

주제어 : 드론, 실시간 드론 영상, 드론 제어, GUI, 얼굴 인식, 딥러닝

Abstract In this paper, I propose a system development methodology that accepts images taken by camera attached to drone in real time while controlling mini drone and recognize and confirm the face of certain person. For the development of this system, OpenCV, Python related libraries and the drone SDK are used. To increase face recognition ratio of certain person from real-time drone images, it uses Deep Learning-based facial recognition algorithm and uses the principle of Triples in particular. To check the performance of the system, the results of 30 experiments for face recognition based on the author's face showed a recognition rate of about 95% or higher. It is believed that research results of this paper can be used to quickly find specific person through drone at tourist sites and festival venues.

Key Words : Drone, Real-time Drone image, Drone Control, GUI, Face Recognition, Deep Learning

1. 서론

1.1 연구배경

최근 4차 산업혁명 시대를 맞이하여 드론(Drone)에 대한 관심이 증폭되고 있다. 드론은 과거에 군사용으로 개발되었으나 최근에는 민간분야에서 다양하고

활발하게 활용 가능성이 높아지면서 빠르게 확산되고 있다. 특히 최근에는 드론을 사용하여 사람의 얼굴을 인식할 수 있어서 기존의 셀카봉을 대신하는 기능을 가진 얼굴 인식 셀카용 드론까지 등장하고 있다. 결국, 가까운 미래의 드론 산업은 1인 1드론 시대를 맞이하게 될 것이고, 조종기 없이 자유롭게 드론을 조종

*This research was supported by Sangji University Research Fund, 2018.

*Corresponding Author : Sung-Ho Kim(kimsh1204@sangji.ac.kr)

Received October 11, 2019

Accepted December 20, 2019

Revised November 6, 2019

Published December 28, 2019

할 수 있는 드론 제품들이 주류를 이룰 것이다. 이와 같은 기능들이 주류가 될 가까운 미래 드론 산업의 육성을 위한 핵심 기술 중의 하나는 바로 사람의 얼굴을 인식하는 영상 처리 기술이다. 드론 영상으로부터 얼굴 인식 기술이 필요한 이유는 드론이 소유자를 인식하고 소유자로부터 명령을 받아 수행할 수 있어야 하기 때문이다, 특히 대중 속에서 사람의 얼굴을 인식하여 특정인을 찾을 수 있어야 한다. 그러므로 본 논문에서는 최근 시중에 판매되고 있는 소형 카메라가 장착된 미니 드론들을 사용하여 실시간 드론 카메라 영상으로부터 인간의 얼굴을 인식하고 식별할 수 있는 시스템을 설계하고 개발하고자 한다. 본 논문의 결론물은 드론 영상으로부터 인간의 얼굴을 인식한 후 인식한 사람이 누구인지 이름을 알려주는 과정으로 처리되는 시스템이다. 2장에서는 본 논문에서 실험에 사용할 드론에 대한 개요, 제어 시스템 및 사용자인터페이스 등에 대하여 기술하고, 3장에서는 얼굴 인식 시스템에 대하여 기술한다. 4장에서는 실시간 드론 카메라 영상을 기반으로 한 얼굴 인식 시스템의 구현 결과를 기술하고 결론 및 향후 연구결과를 기술한다.

1.2 관련 연구

본 논문 주제와 관련된 드론 영상 관련 연구 동향을 살펴보면, 미니 드론의 영상을 기반으로 한 자동 비행 제어와 관련된 연구[1] 등이 있었고, 인공지능 기반 드론 목표물 추적 시스템의 설계 및 구현[2] 등이 있었다. 또한 [3]과 같이 얼굴 인식 셀카용 드론인 호버 카메라가 Best of CES2017의 Top 10에 선정되기도 하였다. 이외에도 다양한 드론 관련 연구들이 있는데, 드론 조종이나 장애물 회피 등과 같이 드론 자체의 성능 및 기능 등과 관련된 연구결과들이 대부분이었다. 본 논문의 한 분야인 얼굴 인식, 검출 등과 관련된 국외 관련 연구 동향을 살펴보면, [4-6]은 OpenCV 및 Python을 사용하여 얼굴 인식(검출) 기능을 구현한 것으로서 얼굴 검출을 위해 Haar Classifier 및 LBP Classifier를 사용하여 장단점을 비교 분석하고 있다. 본 논문에서 참고하여 사용할 가장 핵심 기술인 얼굴 인식 시스템과 관련된 해외 연구 동향으로는 OpenCV, Python 및 Deep learning을 사용한 얼굴 인식 연구결과[7]가 있다. 이 연구결과는 본 논문에서 구현하고자 하는 얼굴 인식 시스템에서 사용할 기능과 동일한 부분이다.

그리고 [8]은 Python에서 얼굴 인식을 구현하기 위한 기본적인 소개를 하고 있다. [9]는 웹캠을 사용하여 Python으로 얼굴 검출을 수행하는 사례를 보여주고 있다. [10]은 얼굴 인식을 위해 연구 개발하고 있는 대표적인 몇 개의 회사 Kairos, Microsoft, Google, Amazon 및 OpenCV 등을 비교 분석하고 있다. [11]은 OpenCV와 Python을 사용하여 실시간 얼굴표정 인식을 위한 기술을 설명하고 있다. [12]는 초고속 R-CNN을 이용한 얼굴 영상에서 눈 및 입술영역을 검출하는 방법에 대하여 소개하고 있다. [13]은 본 연구에서도 사용하고자 하는 라이브러리 dlib로부터 CNN 기반 얼굴 검출기를 소개하고 있다. [14]는 15개의 효율적인 얼굴 인식 알고리즘 및 기술들에 대한 간략한 소개를 하고 있다. [15]는 Python으로 Live Video에서 빠르고 정확한 얼굴을 추적하는 사례를 보여주고 있다. [16]은 AR Drone Quadrotor를 사용하여 얼굴 검출 및 인간 추적 기술을 소개하고 있다. 얼굴 인식, 검출 등과 관련된 국내 연구 동향을 살펴보면, [17-21]은 얼굴 검출, 인식 및 추적 등에 한정된 국내 연구들이 있다.

본 논문이 제안하는 기존연구[1-21]와의 차이점은 드론의 실시간 영상으로부터 사람의 얼굴을 인식하는 기능에 특정인의 얼굴을 정확하게 찾아낼 수 있다는 것이다.

2. 드론 제어 시스템 설계

2.1 드론 코딩 제어

본 논문에서는 코딩이 가능한 드론 중 DJI Tello[22]를 실험에 사용한다. Tello 드론은 최대 비행시간이 13분이며, 약 100m 이내에서 사진 및 동영상 전송이 가능하고, 720p의 해상도를 지원한다. 그리고 Tello SDK 2.0을 Python 언어에 적용하여 코딩으로 제어가 가능하며, Tello SDK 2.0에서 지원해주는 기본 명령어들을 사용한다. 드론은 드론을 조종하고 제어하기 위한 별도의 조종기나 스마트폰 앱을 활용하여 조종 및 제어하는 것이 일반적이다. 즉, 조종기는 Fig. 1 (A)와 같이 스마트폰 앱을 통해 블루투스로 연결하고, 스마트폰과 드론 사이의 통신 방식은 드론이 가지고 있는 와이파이 통신 방식을 사용한다.

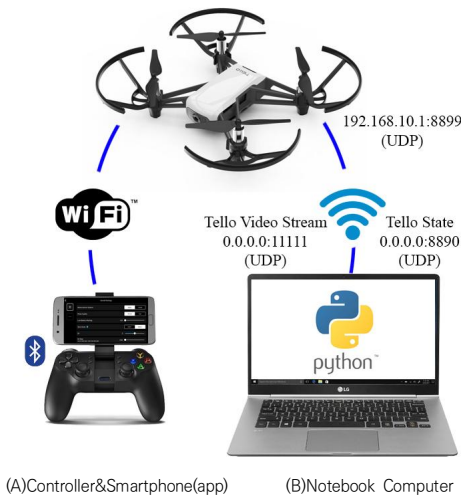


Fig. 1. Two ways to control a drone

본 논문에서는 Fig. 1 (B)와 같이 제어용 프로그램을 직접 개발하여 와이파이가 가능한 노트북 컴퓨터에서 원격 조정한다. Fig. 1의 노트북 컴퓨터와 드론 사이의 통신 구성은 다음과 같다. 첫째, 노트북 컴퓨터에 UDP 클라이언트를 설정하여 동일한 포트를 통해 드론으로부터 메시지를 주고받는다. 둘째, 다른 명령을 전송하기 전에 “command”을 UDP PORT 8899를 통해 드론에 전송하여 SDK 모드를 시작한다. 셋째, 노트북 컴퓨터에 UDP 서버를 설정하고 UDP PORT 8890을 통해 IP 0.0.0.0에서 메시지를 확인한다. 넷째, 노트북 컴퓨터에 UDP 서버를 설정하고 UDP PORT 1111을 통해 IP 0.0.0.0에서 메시지를 확인한다. 마지막 다섯째, 스트리밍을 시작하기 위해 UDP PORT 8889를 통해 드론으로 “streamon”을 보낸다. 노트북 컴퓨터에서 와이파이 통신 방식으로 드론을 제어하기 위해서는 Python 언어를 사용하였으며, Tello SDK 2.0을 함께 사용한다.

2.2 드론 제어를 위한 GUI 설계

드론을 조종하기 위한 프로그램은 Python을 사용하여 개발하며, GUI를 위해서는 PyGame을 사용한다. PyGame은 비디오 게임 제작용으로 설계된 Python 모듈 세트이다. PyGame은 Python과 함께 사용 가능하며 컴퓨터 그래픽스 및 사운드 라이브러리 등이 포함되어 있다. 드론을 조종하기 위해서는 별도의 조종기를

사용할 경우 DJI Go를 주로 사용하거나 드론 개발 업체에서 별도로 제공하는 스마트폰 앱을 사용하는데, 본 논문에서는 별도의 조종기나 스마트폰 앱 등을 전혀 사용하지 않고 직접 개발한 프로그램을 사용하기 때문에 Fig. 2와 같이 드론 개발사에서 제공하는 스마트폰 앱의 범용 GUI 형식을 그대로 따르려고 한다. 왜냐하면, 사용자들에게는 이미 기존의 GUI가 익숙해져 있기 때문이다.



Fig. 2. Original GUI of Tello App

즉, Fig. 2의 좌하부 원형 제어부와 같이 드론의 상승, 하강, 좌회전 및 우회전 기능과 우하부 원형 제어부와 같이 전진, 후진, 좌로 이동 및 우로 이동 등과 같은 기능을 기본적으로 가진다. 그리고 좌상부와 같이 이륙 (Take off) 및 착륙(Land) 기능을 가진 On/Off 아이콘을 별도로 배치한다. 그리고 드론이 장착한 배터리의 잔량을 실시간으로 확인할 수 있도록 비율(%)로 표시하며, 와이파이 통신 상태도 나타낸다. 또한, 미니 드론으로부터 실시간 영상을 와이파이로 받아들이는 양은 FPS로 확인할 수 있는데, 본 논문에서는 FPS를 표시하여 확인할 수 있도록 한다.

3. 얼굴 인식 시스템 설계

본 논문에서는 얼굴 인식 시스템 개발을 위해 프로 그래밍 언어 중의 하나인 Python을 사용하며, 추가로 PyGame, dlib, face_recognition 등의 Python 라이브러리와 OpenCV, Deep Learning, 그리고 드론을 제어하기 위한 미니 드론 SDK 2.0 등을 사용한다. 여기서 dlib는 실제 얼굴 인식 과정에서 사용되는 딥 메트릭 학습(Deep Metric Learning)을 포함하며, face_recognition은 dlib의 얼굴 인식 기능을 더 쉽게 해주는 얼굴 인식 모듈이다. dlib 얼굴 인식 네트워크의 경우 출력 특징 벡터는 사람 얼굴 이미지로부터 128개

실수로 구성된 값 리스트를 구성하여 양자화 한다. 얼굴 인식 네트워크를 구성하기 위해서는 그림 Fig. 3과 같이 Triplets를 사용한다. Fig. 3에서는 서로 다른 2명의 얼굴 사진 3장을 사용하는데, 왼쪽 첫째 사진과 오른쪽 2개의 사진(동일 인물)은 다른 사람 사진이다. 얼굴 인식 네트워크의 얼굴을 정량화하기 위해서 각 사진에 대해 128개 실수 값으로 구성된 리스트를 구성하여 양자화 한다.

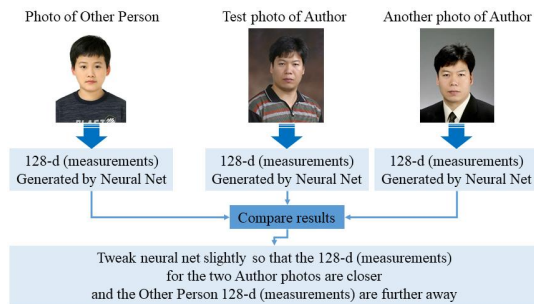


Fig. 3. A single 'Triplet' training step[23]

그런 다음, 신경망의 중량을 조정된 양자화 결과에 의하면 오른쪽 두 개의 사진이 더 가깝고 왼쪽 하나의 사진이 더 멀어진다는 것을 알게 된다. 이와 같은 Deep Metric Learning에 대한 자세한 내용은 dlib의 창시자인 Davis King의 "High Quality Face Recognition with Deep Metric Learning"과 face_recognition 모듈의 저자인 Adam Geitgey의 "Modern Face Recognition with Deep Learning[24]"을 참조하며 시스템 구현에 그대로 활용한다.

4. 시스템 구현결과

4.1 드론 제어를 위한 환경 설정 및 GUI 구축

본 논문에서 실험에 사용하는 미니 드론을 제어하기 위해서는 기본적으로 노트북 컴퓨터에 설치된 최신 버전의 Python(3.7) 언어를 기반으로 하며, Tello SDK(tellopy), DJI Tello(djitellopy), OpenCV(cv2), PyGame(pygame) 및 Numpy 등의 라이브러리를 추가로 설치하여 사용한다. 드론과의 통신을 위해서는 와이파이를 사용하지만, 시스템 개발을 위해서는 Tello SDK(tellopy), DJI Tello(djitellopy)이 주로 사용된다. 그리고 OpenCV(cv2), PyGame(pygame) 및 Numpy

등의 라이브러리는 GUI 등과 관련된 기능을 구현하기 위해서 주로 사용된다. Fig. 4에서 보는 바와 같이 본 논문에서는 드론을 제어하기 위한 GUI를 단순화시켰다. 좌상단에는 배터리 잔량(%), 와이파이 수신율(%), FPS를 나타내고 있으며, 우상단에는 드론 카메라로 찾고자 하는 사람의 사진을 보여주고 있다. 사진 아래에는 버튼을 하나 배치하여 찾고자 하는 사진의 사진을 변경하고자 할 경우에 사용할 수 있도록 하였다. 좌하단부와 우하단부에는 기존의 미니 드론 앱이 가지고 있는 GUI 기능을 그대로 구현하였다. 하단 가운데에는 이륙(Take off) 및 착륙(Land) 기능을 가진 아이콘 버튼을 삽입하여 사용자가 쉽게 이륙과 착륙을 할 수 있도록 하였다.

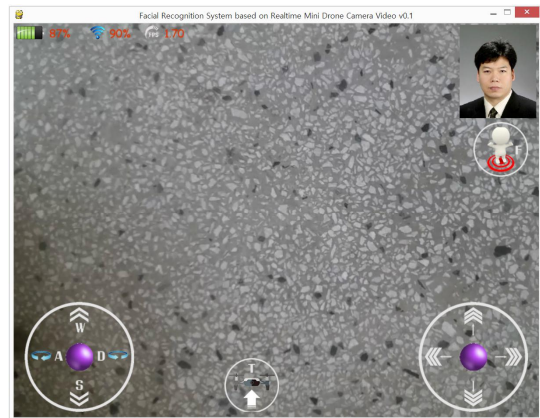


Fig. 4. GUI to control a drone

4.2 얼굴 인식을 위한 데이터 셋

얼굴 인식을 위해서는 특정 인물의 다양한 얼굴 사진들로 구성된 데이터 셋이 필요하다. 데이터 셋을 구성하기 위해서는 최소 33장 이상의 얼굴 사진이 필요하며 각각의 사진마다 128개의 실수 값으로 구성된 양자화가 필요하다. 데이터 셋을 구성하기 위한 얼굴 사진은 얼굴 인식률을 최대한 높이기 위해 카메라가 정면 얼굴을 향하게 삼각대로 고정시키고 상하좌우 및 그 사이사이 등 8각형을 암시하는 8개 방향으로 최소 0도에서 최대 40도까지 10도씩 각각 한 번씩 고개를 돌리면서 촬영한다. 만약 특정 인물을 대상으로 얼굴 인식을 하고자 하는데 특정 인물의 다양한 얼굴 사진들을 쉽게 구하지 못할 경우, 딥러닝 이미지 데이터 셋을 신속하게 구축하기 위해서는 Microsoft의 Bing Image Search API를 활용하는 방법[25]도 있다. 그러나 본 논문에서는 저자

의 얼굴을 기반으로 데이터 셋을 구성하였기 때문에 저자의 다양한 얼굴 사진으로 데이터 셋을 구축하여 실험에 사용하였다. 실험에 사용할 얼굴 인식용 데이터 셋은 저자의 이름으로 된 폴더를 생성하고 33장의 사진을 datasets 폴더에 삽입하면 된다.

4.3 OpenCV와 딥러인을 사용한 얼굴 인코딩

얼굴 인식을 위해서는 앞서 언급한 것과 같이 각각의 얼굴 이미지를 128개 실수 값으로 구성된 양자화를 생성하도록 훈련해야 한다. 또한, 얼굴 인식 네트워크를 제대로 훈련시키기 위해서는 가능한 많은 이미지가 필요하기도 하다. 그리고 사전에 훈련된 네트워크를 실험에 사용하는 것도 매우 효율적이기도 하다. 이와 같은 과정을 위해서는 앞서 설치한 dlib, face recognition 및 OpenCV와 imutils 등의 패키지를 사용한다.

4.4 드론 영상으로부터 얼굴 인식 결과

소형 카메라가 장착된 미니 드론으로부터 실시간 입력된 영상을 기반으로 특정인의 얼굴을 인식하기 위한 실험은 실내외에서 30회 진행하였다. 특정인의 얼굴 인식 실험을 위한 환경적 요인으로는 사람이 볼 때 드론 영상에 사람의 얼굴이 제대로 확인될 수 있는 정도의 자연 및 인공조명이 있는 장소이면 충분하다. 그러므로 상대적으로 주변에 조명이 없는 야간에는 얼굴 인식이 거의 되지 않는 문제점은 있다. 물론 이와 같은 문제점은 드론에 일정한 밝기를 가진 전 방향 조명을 설치하여 해결할 수도 있지만 본 논문에서는 다루지 않는다. 미니 드론 카메라 영상으로부터 인식된 얼굴 영역들은 PyGame에서 제공하는 Circle 함수를 사용하여 (0, 0, 255) 색상의 원형으로 표시하도록 하였다. 본 논문에서는 저자의 얼굴을 특정인으로 설정하고, Fig. 5와 같이 특정인을 실시간으로 찾은 결과를 모니터를 통하여 확인할 수 있었다. 또한, 본 시스템에서 특정인을 제대로 찾은 경우에는 특정인의 얼굴 영역 아래에 이름을 출력해 주고, 특정인이 아닌 경우에는 얼굴 영역은 찾아주지만 이름을 출력하지는 않도록 하였다. 본 시스템의 성능평가를 위해서 30회의 실험을 한 결과 특정인의 얼굴을 제대로 찾아주는 비율은 약 95%를 넘어서는 정확도를 나타내었다. 나머지 5%에 해당하는 오류는 특정인이 드론 카메라를 정면으로 바라보지 않고 고개를 옆으로 40도에서 45도 이상 돌리는 경우로서 양쪽 눈 모

두가 카메라 영상에 나타나지 않는 경우 즉, 한쪽 눈만 영상에 나타나는 경우 등이 있었다.

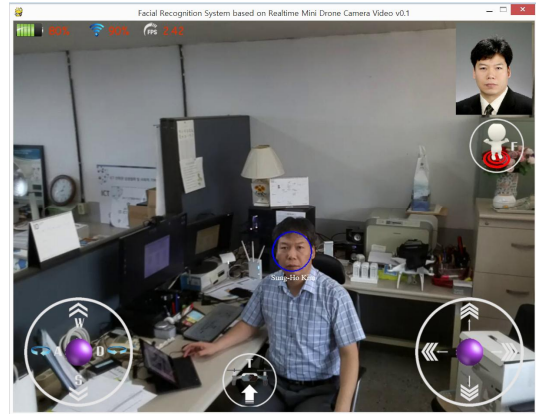


Fig. 5. Real time face recognition result from drone image

또한, 33장 미만과 이상의 얼굴 사진으로 구성된 데이터 셋에 대한 비교 실험에서는 특정인의 얼굴 인식에 전혀 문제가 없었다. 왜냐하면, 본 논문에서는 얼굴 인식 오류를 최소화하기 위해서 33장이라는 얼굴 사진의 장수에 대한 수치를 논리적으로 적용한 것이고 실제로는 딥러닝 기술을 활용하였기 때문에, 사람의 양쪽 두 눈이 드론 영상에 제대로 촬영만 되면 얼굴 인식에 문제가 없기 때문이다.

5. 결론 및 향후 연구

본 논문에서는 미니 드론을 와이파이가 가능한 노트북 컴퓨터에 연결하고 실시간으로 촬영된 드론 영상을 받아들이며 특정인의 얼굴을 인식해서 알려주는 시스템을 개발하였다. 드론을 와이파이로 제어하기 위해서는 드론 개발사에서 제공하는 SDK 2.0을 기본적으로 사용하였지만, 드론에 장착된 카메라 영상으로부터 특정인의 얼굴을 인식하는 것은 디지털 영상 처리 알고리즘을 사용하였다. 특히, 드론 제어와 특정인의 얼굴 인식 등을 위해서는 Python 및 Python에서 사용 가능한 다양한 라이브러리들을 추가로 설치하여 사용하였다. 드론 영상으로부터 얼굴 인식을 효율적으로 하기 위해서는 OpenCV와 딥러닝 등의 알고리즘을 사용하였다. 그러므로 본 시스템을 상용화하게 된다면 사람들이 많이 붐비는 관광지나 축제장 등에서 특정인을 잃어버렸

을 때 드론으로 상공에서 빠르고 쉽게 찾을 수 있을 것으로 판단된다.

그러나 본 논문에서 사용한 미니 드론의 배터리가 최대 13분이므로 실제 외부 비행에서 특정인을 찾기 위해서는 10분 이내의 시간밖에 사용할 수 없다. 그러므로 더 긴 시간을 사용할 수 있는 배터리를 사용할 필요성이 있다. 그리고 본 시스템은 와이파이기가 가능한 노트북 컴퓨터에서 실험을 하였기 때문에 실제로 이동하면서 사용하기에는 불편한 점이 있었다. 그러므로 향후 연구에서는 노트북 컴퓨터를 사용하지 않고 스마트폰 앱에서 실행이 가능하도록 시스템을 개선 및 보완하여 개발할 필요가 있다.

REFERENCES

- [1] E. H. Sun, T. H. Luat, D. Y. Kim & Y. T. Kim. (2015). A Study on the Image-based Automatic Flight Control of Mini Drone. *Journal of Korean Institute of Intelligent Systems*, 25(6), 536-541.
- [2] D. W. Kim et al. (2017). AI-Based Drone Object Tracking System: Design and Implementation. *The Journal of Korean Institute of Communications and Information Sciences*, 42(12), 2391-2401.
- [3] Zero Zero Robotics. (2018). *HOVER*. <https://gethover.com/?d=pc&c=kr>
- [4] Super Data Science. (2019). *Face detection using OpenCV and Python: A beginner's guide*. <https://www.superdatascience.com/opencv-face-detection>
- [5] Mike Driscoll. (2018). *Face Detectioin using Python and OpenCV*. <https://dzone.com/articles/face-detection-using-python-and-opencv>
- [6] Adrian Rosebrock. (2015). *Creating a face detection API with Python and OpenCV*. <https://www.pyimagesearch.com/2015/05/11/creating-a-face-detection-api-with-python-and-opencv-in-just-5-minutes>
- [7] Adrian Rosebrock. (2018). *Face recognition with OpenCV, Python and Deep learning*. <https://www.pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning>
- [8] Elliot Forbes. (2017). *An Introduction to Face Recognition in Python*. <https://tutorialedge.net/python/intro-face-recognition-in-python>
- [9] Shantnu Tiwari. (2012-2019). *Face Detection in Python Using a Webcam*. <https://realpython.com/face-detection-in-python-using-a-webcam>
- [10] Ben Virdee. (2017). *Face Recognition : Kairos vs Microsoft vs Google vs Amazon vs OpenCV*. <https://www.kairos.com/blog/face-recognition-kairos-vs-microsoft-vs-google-vs-amazon-vs-open-cv>
- [11] Sefik Ilkin Serengil. (2018). *Real Time Facial Expression Recognition on Streaming Data*. <https://sefiks.com/2018/01/10/real-time-facial-expression-recognition-on-streaming-data>
- [12] J. H. Lee. (2018). A Method of Eye and Lip Region Detection using Faster R-CNN in Face Image. *Journal of the Korea Convergence Society*, 9(8), 1-8.
- [13] Arun Ponnusamy. (2018). *CNN based face detector from dlib*. <https://towardsdatascience.com/cnn-based-face-detector-from-dlib-c3696195e01c>
- [14] Varun Kumar. (2017). *15 Efficient Face Recognition Algorithms And Techniques*. <https://www.rankred.com/face-recognition-algorithms-techniques>
- [15] Glenn. (2018). *Fast and Accurate Face Tracking in Live Video with Python*. <https://www.codemade.io/fast-and-accurate-face-tracking-in-live-video-with-python/>
- [16] M. Gemici & Y. Zhuang. (2011). *Autonomous Face Detection and Human Tracking using AR Drone Quadrotor*. http://www.cs.cornell.edu/courses/cs4758/2011sp/final_projects/spring_2011/Gemici_Zhuang.pdf
- [17] H. K. Kim, J. M. Moon & J. Y. Park. (2018). Research Trends for Deep Learning-Based High-Performance Face Recognition Technology. *ETRI Electronics and Telecommunications Trends*, 33(4), 43-53.
- [18] O. S. Kwon. (2018). Face recognition Based on Super-resolution Method Using Sparse Representation and Deep Learning. *Journal of Korea Multimedia Society*, 21(2), 173-180.
- [19] W. J. Hwang. (2017). Trends in Deep Learning-based Face Detection, Landmark Detection and Face Recognition Technology. *The Korean Society Of Broad Engineers, Broadcasting and Media Magazine*, 22(4), 41-49.

- [20] T. Y. Ko et al. (2017). Real-time face recognition and tracking system using deep learning in various environments. *The Institute of Electronics and Information Engineers*, 643-646.
- [21] S. Y. Kang et al. (2017). Real-time Missing Persons Recognition System through CCTV based on Deep learning. *Conference of the Korea Information Science Society*, 1941-1943.
- [22] RAZE. (2019). *Tello*.
<https://www.ryzerobotics.com/kr/tello>
- [23] A. Geitgey. (2016). *Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning*.
<https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78>
- [24] A. Geitgey. (2019). *Face Recognition Documentation*.
<https://media.readthedocs.org/pdf/face-recognition/latest/face-recognition.pdf>
- [25] A. Rosebrock. (2018). *How to build a deep learning image dataset*.
<https://www.pyimagesearch.com/2018/04/09/how-to-quickly-build-a-deep-learning-image-dataset/>

김 성 호(Sung-Ho Kim)

[정회원]



- 1998년 8월 : 숭실대학교 컴퓨터학과(공학석사)
- 2005년 2월 : 숭실대학교 컴퓨터학과(공학박사)
- 2006년 3월 : 상지대학교 컴퓨터정보공학부 교수

- 2018년 3월 ~ 현재 : 상지대학교 컴퓨터공학과 교수
- 관심 분야 : 컴퓨터그래픽스, 컴퓨터 애니메이션, 모션 캡처, 가상현실/증강현실, 드론, 컴퓨터게임, Web3D, 멀티미디어, 지리정보시스템
- E-Mail : kimsh1204@sangji.ac.kr