

구글 맵 API를 이용한 딥러닝 기반의 드론 자동 착륙 기법 설계

이지은¹, 문형진^{2*}

한세대학교 대학원 IT융합전자공학과 석사과정, 성결대학교 정보통신공학과 조교수

Design of Deep Learning-Based Automatic Drone Landing Technique Using Google Maps API

Ji-Eun Lee¹, Hyung-Jin Mun^{2*}

¹Graduate Student, IT Convergence and Electronics Engineering, Hansei University

²Assistant Professor, Department of Information and Communication Engineering, SungKyl University

요 약 최근 원격조종과 자율조종이 가능한 무인항공기(RPAS:Remotely Piloted Aircraft System)가 택배 드론, 소방드론, 구급 드론, 농업용 드론, 예술 드론, 드론 택시 등 각 산업 분야와 공공기관에서의 관심과 활용이 높아지고 있다. 자율조종이 가능한 무인드론의 안정성 문제는 앞으로 드론 산업의 발달과 함께 진화하면서 해결해야 할 가장 큰 과제이기도 하다. 드론은 자율비행제어 시스템이 지정한 경로로 비행하고 목적지에 정확하게 자동 착륙을 수행할 수 있어야 한다. 본 연구는 드론의 센서와 GPS의 위치 정보의 오류를 보완하는 방법으로서 착륙지점 영상을 통해 드론의 도착 여부를 확인하고 정확한 위치에서의 착륙을 제어하는 기법을 제안한다. 서버에서 도착지 영상을 구글맵 API로부터 수신받아 딥러닝으로 학습하고, 드론에 NAVIO2와 라즈베리파이, 카메라를 장착하여 착륙지점의 이미지를 촬영한 다음 이미지를 서버에 전송한다. Deep Learning으로 학습된 결과와 비교하여 임계치에 맞게 드론의 위치를 조정한 후 착륙지점에 자동으로 착륙할 수 있다.

키워드 : 지상관제 소프트웨어, GPS, 딥러닝, 구글맵 API, 자동 착륙, 이미지매칭

Abstract Recently, the RPAS(Remote Piloted Aircraft System), by remote control and autonomous navigation, has been increasing in interest and utilization in various industries and public organizations along with delivery drones, fire drones, ambulances, agricultural drones, and others. The problems of the stability of unmanned drones, which can be self-controlled, are also the biggest challenge to be solved along the development of the drone industry. Drones should be able to fly in the specified path the autonomous flight control system sets, and perform automatically an accurate landing at the destination. This study proposes a technique to check arrival by landing point images and control landing at the correct point, compensating for errors in location data of the drone sensors and GPS. Receiving from the Google Map API and learning from the destination video, taking images of the landing point with a drone equipped with a NAVIO2 and Raspberry Pi, camera, sending them to the server, adjusting the location of the drone in line with threshold, Drones can automatically land at the landing point.

Key Words : Ground control software, GPS, Deep learning, Google Maps API, Automatic landing, Image matching

1. 서론

드론은 무인항공기(UAV)로 초기에는 군사적 목적인 사격연습용에서 점차 정찰, 감시, 공격 용도로 사

용되고 있다. 최근에는 4차산업혁명으로 정보통신기술의 발전과 함께 다양한 민간의 부분에서 활용되고 있다. 대표적인 것이 화산 분화구 촬영처럼 사람이 직접

*Corresponding Author : 문형진(jinmun@gmail.com)

Received January 28, 2019

Revised February 6, 2020

Accepted February 20, 2019

Published February 29, 2020

가서 촬영하기 어려운 장소를 촬영하거나, 인공위성을 이용해 위치를 확인하는 GPS(위성항법장치) 기술을 활용하여 서류, 책, 피자 등을 개인에게 배달하는 인터넷 쇼핑물의 무인 택배 서비스이다. 또한, 최근 드론 택시 개발과 함께 상용화를 위해 자율주행시스템이 탑재된 자율비행 드론이 개발되었다. 하지만 다양한 목적의 드론이 개발되고 동시에 상용화를 위해 넘어야 할 장애물들이 있는데 그중 가장 큰 부분이 안전한 드론 착륙이다. 즉 원하는 위치에 돌발상황에도 불구하고 안전하게 착륙하는 것이다. 자율비행을 위해선 기본적으로 GPS를 활용하는데 일반적으로 GPS 위치 측정의 정확성을 떨어뜨리는 요소들을 3가지로 나눌 수 있다. 첫째, 인공위성 시간 오차, 인공위성 위치 오차, 전리층과 대류층의 굴절 및 잡음(Noise), 다중 경로 등 구조적 요인으로 생기는 오차가 있다. 둘째, 위성의 위치 상황에 따른 기하학적 오차가 있으며 마지막으로 가장 큰 오차의 원인인 SA(Selective Availability)가 있다[1]. 이 요소들이 낳는 오차들은 시간과 장소에 따라서 매우 크게 변한다. GPS의 위치 측정의 정확성을 높이기 위해 다양한 연구들이 시행되어 몇 m 수준의 위치 정확도를 지니는 보정 위성항법 시스템(DGPS)에서 몇 cm 수준의 위치 정확도를 지니는 반송파 보정 위성항법 시스템(CDGPS)으로 발전하고 있다. 하지만 몇 cm 정도의 위치 정확도를 기대하기 위해선 코드를 사용한 위치 계산보다 계산량이 많고 시스템의 무결성이 저하되어 이용되는 분야가 좁은 영역으로 제한적이다. 그러므로 드론의 자율비행은 이런 GPS의 오류로 인해 착륙목표 지점에서 몇 미터 떨어진 지점에서 착륙하게 만든다. 이 GPS의 오류로 인해 사용자는 드론을 찾아야 하는 불편함을 겪어야 한다. 무인항공기인 드론의 사용이 다양한 분야에서 확대되고 있으므로, 드론 자율비행의 정확도를 높이기 위한 연구할 필요가 있다. 이 논문에서 드론 자율비행에서 정확한 착륙을 제어하기 위해 정확도를 높이기 위한 드론의 착륙제어 방식으로 구글맵 API를 통해 얻은 이미지와 드론 카메라로 촬영한 이미지를 딥러닝을 이용하여 비교한다. 두 이미지를 매칭한 영상처리 기법을 이용하여 두 이미지의 임계치가 90% 이상 매칭된 위치를 찾을 수 있도록 연구되었다. 오차범위를 최소화한 목표지점에 드론이 정확한 착륙을 할 수 있도록 설계하였다.

2. 관련연구

2.1 드론 착륙 기법

드론의 이착륙과 관련하여 RC(remote control), 착륙지의 H 마커(marker) 및 GPS(global positioning system)를 이용한 방법 등이 연구됐다. RC를 통한 수동 이착륙 방법은 조종자가 드론의 상태(고도, 속도, 방향 등)를 확인하면서 RC를 통하여 드론을 조종하며 이착륙을 진행한다. 조종자가 드론의 상태를 확인하는 방법으로는 드론을 조종자의 육안으로 확인하는 방법, 드론에 설치된 카메라를 통하여 드론의 상태를 확인하는 방법, 드론에서 GCS(ground control system)로 전송하는 정보를 통하여 확인하는 방법 등이 있다. 착륙지의 H 마커를 추적하여 자동 착륙을 진행하는 기술은 드론에 저장해둔 마커와 드론에 탑재된 카메라의 영상을 비교하여 영상 내에 마커가 존재하는지 확인하고, 마커를 발견할 시 영상 내의 마커의 위치, 크기, 방향 등을 바탕으로 현재 드론의 위치, 방향, 높이를 계산하며 착륙을 진행한다. 마커를 인식하기 위해서 이진화 기법, 레이블링(Labeling) 기법을 사용하고 레이블마다 마커의 모양이 정사각형인지 마커의 외곽선을 이용하여 마커인지 아닌지 판단하는 검증 과정을 진행한다. 또 마커의 꼭짓점을 활용하여 카메라와 마커 간의 상대적 거리를 계산하고 기울어진 각도를 추정된 후 자동 착륙을 시도한다. 하지만 마커가 없는 지역에서는 자동 착륙이 불가능하다는 문제점을 가지고 있다[2]. GPS를 통한 드론 착륙 방법은 GPS를 통하여 드론의 현재 위치를 확인하고 사용자가 요청한 지점으로 이동하여 착륙지점에 도달하면 GPS를 통하여 드론의 고도와 좌표를 확인하며 착륙을 진행한다. 하지만 주위 환경을 고려하지 않고 오직 사용자가 지정한 경로를 따라 이륙하기에 환경에 따라 수 미터의 오차가 발생하고 실내에서는 작동하지 않는다는 점에 불안정하다[3]. 이러한 불확실한 상황에서 최적의 제어를 위해 자율 착륙을 위한 학습 기반 제어 방법들이 연구되었다. Polvara et al은 다른 단계에서 내비게이션에 대한 고급 제어 정책으로 사용할 수 있는 Deep Q-Networks(DQNs)의 계층을 기반으로 한 접근 방식을 제안했다. 최적의 정책으로, 대규모의 시뮬레이션 된 환경에서 자율적으로 착륙하는 것을 보여주었다. 또한, 다양한 장애와 불확실성이 존재하는 상황에서 관제사가 항공기를 안전한 착륙으로 유

도할 수 있도록 보장하기 위해 적응형 신경 네트워크에 기초한 다수의 접근 방식들이 채택되었다[4].

2.2 이미지 매칭(Image Matching)

이미지 매칭(matching)은 컴퓨터 비전의 하위 영역으로, 이미지 세트 간의 유사성 또는 다중 유사성을 찾아내고, 두 가지 이미지가 유사한지 아닌지를 확인하는 것을 목표로 한다. 유사한 이미지를 일치시키는 이 특별한 작업은 다양한 알고리즘을 사용하여 수행되었다. 이미지매칭 기법으로 OpenCV를 활용한 Feature matching이 있다. 영상에서 특징이 될만한 지점이나 영상의 중요한 정보를 포함하고 있는 지점을 특징점(keypoint, feature point)이라 한다. 이미지의 수많은 피쳐(Feature)를 추출하여, 해당 부분이 회전되거나, 확대/축소되거나, 찌그러져도 동일한 피쳐(Feature)로 인식하도록 보정 하는데 이런 방식으로 추출된 피쳐(Feature)들은 다른 이미지의 피쳐셋(Feature set)과 비교하면서 유사성을 검사하여 두 개의 이미지에서 추출한 피쳐(Feature)들이 높은 비율로 일치한다면 동일한 또는 유사한 이미지로 볼 수 있다[5]. 좋은 특징점 검출 방법은 영상의 다양한 변화요소인 카메라의 위치 및 관찰 방향 변화, 조명 변화, 물체의(위치, 크기, 형태) 변화, 물체의 회전등에 강인하고 불변한 검출이 가능해야 한다. 영상 매칭에는 여러 가지 기법이 있지만, 대표적인 특징점 검출 알고리즘인 SIFT(Scale Invariant Feature Transform) 기법을 가리킨다. 기존 영상의 스케일 변화에 민감한 문제를 해결하기 위해, SIFT 기법은 DoG(가우스 차이)를 기반으로 영상에서 스케일 축의 정점으로부터 큰 값을 찾는다. OpenCV 라이브러리를 사용하여 실시간 이미지 추적을 수행하고 사용된 라이브러리 패키지는 FindObject2D[6]이다.

KNN(K-Nearest Neighbor Algorithm)은 거리 측정 방법에 따라 그 결과가 크게 달라지는 알고리즘으로 가장 흔히 사용하는 거리 척도로서 두 관측치 사이의 직선 최단거리를 의미한다. Best와 Second를 비교해 필터링하면 신뢰할 수 없는 특징점을 걸러내기 때문에 매칭이 줄어들게 된다. 또한, k가 1인 1-NN의 오차 범위에서는 모델 성능을 어느 정도 신뢰할 수 있다[7].

2.3 영상처리 딥러닝(Deep Learning)

최근 인물사진을 통해 사용자인증을 하는 데 딥러닝 기술을 활용한다[8]. 주어진 인물사진으로부터 사람의 얼굴을 찾고, 얼굴에서 특징점을 도출하여 학습한다. 학습된 상태에서 새로운 인물사진을 입력하면 입력된 사진으로부터 사람의 얼굴에서 특징점을 찾아 학습된 이전의 사진의 인물과 동일임을 확인하는 기술들이 보편화 되고 있다.

3. 제안기법

3.1 하드웨어 및 소프트웨어 구성 요소

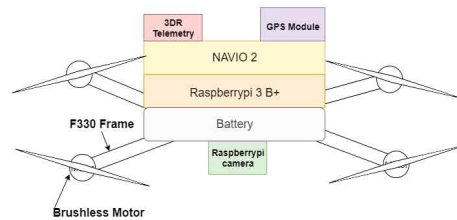


Fig. 1. Components of a drone

Fig. 1 제안기법에서 UAV는 Flight F330 프레임에 빠른 속도와 무선 LAN, Bluetooth가 내장되어있는 프로세서로 라즈베리파이 카메라 모듈을 부착하여 OpenCV와 연동해서 영상을 제어할 수 있는 라즈베리파이와 드론 컨트롤러인 NAVIO2를 올린다. NAVIO2의 특징으로는 GNSS 수신기, 방향 및 모션 감지를 위한 가속도계, 자이로스코프 및 자력계인 듀얼 IMU, RC I / O 보조 프로세서, 고해상도 기압계, 센서와 라디오를 위한 노출 된 ADC, I2C 및 UART 인터페이스 확장 포트, 트리플 리턴턴트 전원이 있다. NAVIO2 위에 매우 작은 스마트 안테나가 일체화된 모듈을 제공하며 -163dBm 추적 및 -160dBm의 내비게이션 감도, 48개 이상의 위성 확인 채널을 가지고 있는 GPS 수신기와 수 킬로미터에 이르는 소형 전방향 안테나를 사용하고 더 넓은 범위에서 사용할 수 있는 양방향 증폭기인 라디오 데이터 전송 모듈 3DR Telemetry를 장착한다. 서버는 드론이 움직이는 경로와 GPS 정보를 수신하여 그것을 기반으로 방향을 잡고, 드론의 카메라로 찍은 영상을 서버로 전송하여 구글맵 API로부터 온 영상과 비교하여 정확한 위치인지를 확인한다. UAV의 맨 아래에는 고정된 위치에서

사용 가능한 카메라로 5백만 화소, 시야각 75.7도 최고의 센서, 적외선 램프 액세스를 지원하는 라즈베리파이 카메라를 장착하고 그 외에 Brushless 모터와 프로펠러, ESC를 장착하였다.



Fig. 2. Mission Planner [9]

Fig. 2는 미션플래너(Mission Planner)의 화면을 캡처한 그림이다. 미션플래너는 ArduPilot 오픈소스 자동 파일럿 프로젝트를 위한 모든 기능을 갖춘 지상국 애플리케이션이다. Ardupilot에서 지원하는 모든 기능을 갖춘 GCS(Ground Control System)로 하드웨어, 맞춤형 스크립팅 및 시뮬레이션과의 Point and click 상호 작용을 제공하며 GCS를 통해 사용자는 (비행기) 차량을 설정, 구성, 테스트 및 튜닝을 할 수 있다. Calibration 또한 고급 패키지를 통해 자율적인 Mission Plan, 운영 및 사후 분석이 가능하다.

3.2 제안 프로토콜

지상관제 S/W인 Mission Planner를 이용하여 드론을 원하는 위치로 이동(컨트롤)한 다음 드론이 착륙 목표지점에 도착한 후에는 GPS 정보를 착륙지점과 비교하여 같으면 카메라로 착륙을 위한 사진을 찍고 서버로 이미지 전송한다. 착륙목표 지점의 구글맵 API로부터 내려받은 이미지를 딥러닝으로 매칭하여 드론의 방향과 위치 등 조정이 가능하다.

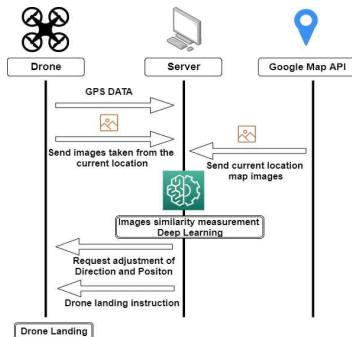


Fig. 3. Automatic Landing Protocol

Fig. 3은 자동 착륙 프로토콜로 다음과 같다.

Step 1. 드론은 목적지로 도착하면 자신의 GPS 정보를 서버로 전송한다.

Step 2. 드론 장착된 카메라로 착륙목표 지형을 영상을 찍어 서버로 전송한다.

Step 3. 서버는 드론으로부터 받은 GPS 정보를 구글맵 API에 전송하여 구글맵으로부터 영상을 수신한다.

Step 4. 서버는 드론으로부터 받은 영상과 구글맵 API로부터 받은 영상을 입력으로 같은 지역인지 Deep learning으로 이미지 Matching(유사도 측정)을 한다.

Step 5. 서버는 정확한 위치를 파악하여 드론의 방향, 위치조정을 요청한다.

Step 6. 목표지점이 맞는지 확인하고 드론에 착륙 지시를 한다.

Step 7. 드론은 지시받은 목표지점에 착륙한다.

3.3 딥러닝을 이용한 두 영상 이미지 매칭

Google Map은 구글에서 제공하는 지도 서비스로 구글 지도는 위성 사진, 스트리트 뷰, 360° 거리 파노라마 뷰, 실시간 교통 상황 (구글 트래픽), 그리고 도로, 자동차, 자전거(베타), 대중교통의 경로를 제공한다. 구글에서 제공하는 라이브러리를 활용하여 구글맵 영상을 전송받아 영상처리(딥러닝)를 한다.

드론 착륙 기술 중 하나인 GPS의 문제점을 보완하기 위해 드론에 부착된 라즈베리파이에 카메라를 연결하고 라즈베리파이에 OpenCV를 설치하여 카메라로부터 받은 영상 이미지와 구글맵 API를 통해 얻은 지도의 이미지를 딥러닝 지도학습을 통해 두 이미지 유사도를 측정한다. 두 이미지의 크기가 다르거나 회전되어 있을 때 이미지의 특징에 대한 정보를 추출하고, 이 특징을 비교해야 하는데 이미지의 특징을 추출하는 알고리즘 중에서 SIFT 알고리즘과 KNN 알고리즘을 이용해 이미지에서 특징점을 추출하고, 두 이미지를 비교한다.

```

import numpy as np
import cv2

img1 = cv2.imread('C:\pywork\images\camera_hanse.png',0)
img2 = cv2.imread('C:\pywork\images\googlemap_hanse.png',0)
# Initiate SIFT detector
sift = cv2.xfeatures2d.SIFT_create()
# find the keypoints and descriptors with SIFT
kp1, des1 = sift.detectAndCompute(img1,None) #1
kp2, des2 = sift.detectAndCompute(img2,None) #2
# BFMatcher with default params
bf = cv2.BFMatcher()
matches = bf.knnMatch(des1,des2, k=2) #3
# Apply ratio test
good = []
for m,n in matches:
    if m.distance < 0.75*n.distance: #4
        good.append([m])
# cv2.drawMatchesKnn expects list of lists as matches.
img3 = np.array([])
img3 = cv2.drawMatchesKnn(img1, kp1, img2, kp2, good,
                           img3, flags=2)

cv2.imshow('aaa', img3)
cv2.waitKey(0)
    
```

Fig. 4. Image matching source using SIFT and KNN[10]

Fig. 4에서는 SIFT 알고리즘은 이미지 하나를 입력하고 이미지에서 특징이 될만한 위치를 찾는데, 이를 특징점(모서리, 꼭짓점 등 변화가 크게 나타나는 부분)이라 한다. 각 특징점에서 특징을 뽑아내서 128 차원의 벡터로 Descriptor를 추출한 다음 이미지 전체에 대해 특징점을 찾고, Descriptor를 추출한다. 다음 단계로 특징을 가진 두 이미지를 비교하기 위해선 Matching이라는 순서가 기다린다. 대표적인 Matching 알고리즘인 BFMatcher(Brute-Force Matcher)를 기준으로 두 이미지를 비교한다.

- Step 1. img1에 대해서 SIFT Descriptor를 추출한다.
- Step 2. img2에 대해서 SIFT Descriptor를 추출한다.
- Step 3. img1의 특징점 개수와 img2의 특징점 개수를 각각 비교해 거리를 구한다.
- Step 4. 가까운 거리부터 정렬해서 사용한다.
- Step 5: camera Image와 googlemap Image에 대해 각각 key point와 descriptor를 구한다. (#1, #2)
- Step 6: 여기에서는 camera_hanse 이미지로부터 각각의 descriptor 별로 2개의 (K=2) Best 매치를 구한다. (#3)
- Step 7: 각 descriptor 별로 best 매치가 다른 매치들에 비교해 압도적인지 0.8이 아닌, 0.75를 사용해서 계산한다. (#4) 이처럼 KNN Match를 사용하면

googlemap image에 camera_hanse.png과 관련 없는 다른 이미지를 입력할 때 신뢰할 수 없는 특징점을 걸러내기 때문에 매칭이 줄어들게 된다[11].

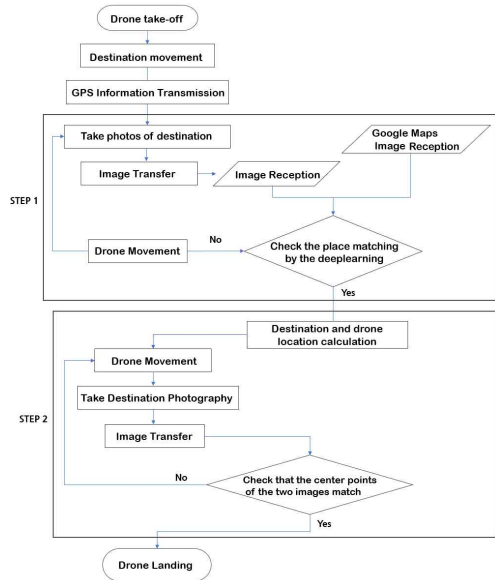


Fig. 5. Drone Automatic Landing flowchart

4. 분석 및 평가

Fig. 5는 드론이 정확한 위치에 착륙하기 위한 과정을 나타낸다. 드론은 이륙한 후 GPS 정보를 이용해 착륙지점으로 이동하여 주기적으로 GPS 정보를 서버에 전달한다. 목적지인 착륙지점에 도착했을 때 드론에 부착된 카메라로 해당 지형을 촬영하고, 이미지를 서버로 전달한다. 서버에서는 정확한 착륙지점을 지정할 때 구글맵 API로부터 수신한 이미지와 드론으로부터 받은 이미지를 딥러닝으로 학습된 모델을 이용하여 비교한다. 구글로부터 받은 이미지의 중심을 드론으로부터 받은 이미지와 비교하여 방향을 조정하고, 떨어진 정도만큼 이동한다. 2개의 이미지의 중심이 일치할 때 드론이 착륙한다.

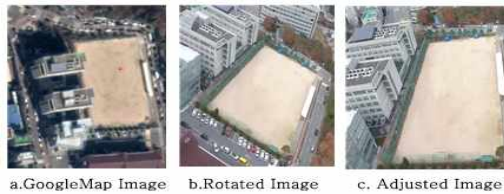


Fig. 6. Google Map Image and Camera Image

Fig. 6(a)는 구글맵으로부터 온 이미지이고, Fig. 6(b), Fig. 6(c)는 드론이 촬영한 이미지이다. 구글맵으로부터 온 이미지는 몇 년 전에 촬영된 이미지이므로 현재 지형 이미지와 다를 수 있다. 딥러닝 학습을 통해 같은 지역인지를 확인하고 같은 영상이라고 판단이 된다면 [Fig. 5]의 자동 착륙 알고리즘에 따라 드론의 정확한 위치를 찾아서 이동한 후 착륙을 한다. 본 제안기법을 통해 정확한 위치를 구글맵에서 얻은 이미지를 통해 파악하고, 그 위치에 착륙할 수 있게 되었다. 드론의 착륙 위치와 구글맵의 중심점(GPS 정보)간의 약간의 오차는 발생할 수 있다.

5. 결론

본 연구는 드론의 센서와 GPS의 위치 정보의 오류를 보완할 방법으로 착륙지점 영상을 기준으로 도착 여부를 확인하고 정확한 지점에 착륙을 제어하는 기법을 제안하였다. 서버에서 도착지 영상을 구글맵 API로부터 수신받은 이미지와 드론에 NAVIO2와 라즈베리파이, 카메라를 장착하여 착륙지점의 영상을 촬영하고 서버에 전송하여 수신받은 두 이미지의 크기가 다르거나 회전되어 있을 때 이미지의 특징에 대한 정보를 추출하고, 이 특징점을 비교하기 위해서 이미지의 특징을 추출하는 SIFT 알고리즘과 이미지를 매칭 하는 KNN 알고리즘을 이용해 이미지 매칭(딥러닝)을 하고 정확한 착륙지점에 드론이 학습된 결과와 비교하여 임계치에 맞게 드론의 위치를 마지막으로 다시 한번 조정하도록 하여 착륙지점에 자동으로 착륙할 수 있도록 한다[12]. 향후 연구로는 제안된 기법을 적용하여 정확도를 높이고, 구글맵 영상 이미지에서 매칭 하는 개선된 알고리즘 개발에 관한 연구가 필요하다.

REFERENCES

- [1] Y. K. Ju, H. J. Mun & K. H. Han. (2019). Image Processing Based Drone Landing Technique Considering GPS error and wind Direction, *IJITEE*, 8(2), 245-254
- [2] J. M. Jeong, J. S. Kim, K.W. Lee, N.W. Hwang, T.S. Yoon & J. B. Park. (2015). Development and Validation of Image Sensor Based Landing Site Recognition Algorithm for Autonomous Landing of the Unmanned Aerial Vehicle. *Journal of The Korean Society for Aeronautical & Space Sciences*, 1547-1550
- [3] H. N. Kim & Y. J. Jung. (2016). Automatic take-off and landing method and device of drones, KR. Patent No. 10-1749578 “<https://patents.google.com/patent/KR101749578B1/ko>”
- [4] Y. Feng, C. Zhang, S. Baek, S. Rawashdeh & A. Mohammadi (2018). Autonomous landing of a UAV on a moving platform using model predictive control. *Drones*, 2(4), 34. DOI: 10.3390/drones2040034
- [5] N. Jayanthi & S. Indu. (2016). Comparison of image matching techniques. *International Journal of Latest Trends in Engineering and Technology*, 7(3), 396-401 DOI: 10.21172/1.73.552
- [6] OpenCV Library Package-”IntRoLab [internet]. Available
- [7] KNN Euclidean Distance, “<https://ratsgo.github.io/machine%20learning/2017/04/17/KNN/>”
- [8] H. J. Mun & G. H. Kim. (2019). A Survey on Deep Learning based Face Recognition for User Authentication, *Journal of Industrial Convergence*, 17(3), 23-29. DOI: 10.22678/jic.2019.17.3.023
- [9] MissionPlanner, <https://ardupilot.org/planner/docs/mission-planner-overview.html>
- [10] Comparison of Features Using SIFT Technicians. “<https://github.com/yildbs/SearchImageWithPy>”

thon/blob/master/Example-OpenCV/brute_force_matching_knn.py”

- [11] D.G Lowe. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 91-110.
DOI:10.1023/b:visi.0000029664.99615.94
- [12] J.P. Lee, J.W. Lee, & K.H. Lee. (2016). A Scheme of Security Drone Convergence Service using Cam-Shift Algorithm. *Journal of the Korea Convergence Society*, 7(5), 29-34.
DOI: 10.15207/JKCS.2016.7.5.029

이지은(Ji-eun Lee)

[정회원]



·2006년 2월 : 서울디지털대학교
(e-business 학사)
·2019년 3월 ~ 현재 : 한세대학교
IT융합전자공학과(공학석사과정)

- 관심분야 : 딥러닝, 로봇제어, 영상처리
- E-Mail : jebs2u@gmail.com

문형진(Hyung Jin Mun)

[종신회원]



·2008년 2월 : 충북대학교 전자계산
학과(이학박사)
·2017년 3월 ~ 현재 : 성결대학교
정보통신공학과 조교수

- 관심분야 : 사용자 인증, 딥러닝, 빅데이터분석
- E-Mail : jinmun@gmail.com