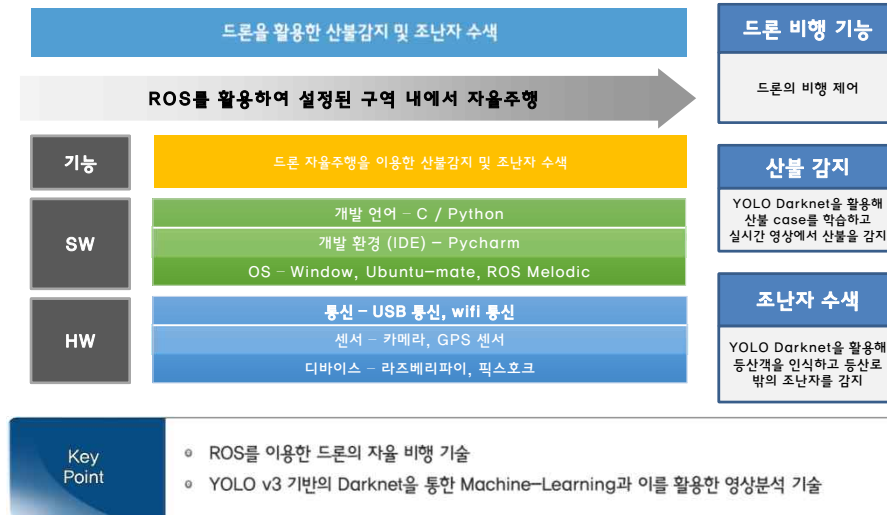


1



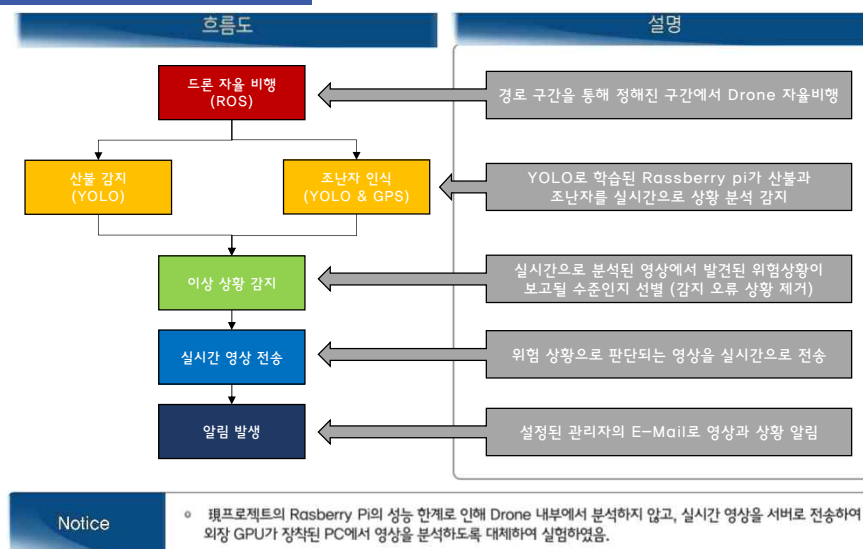
2

## | 1. 시스템 구성도



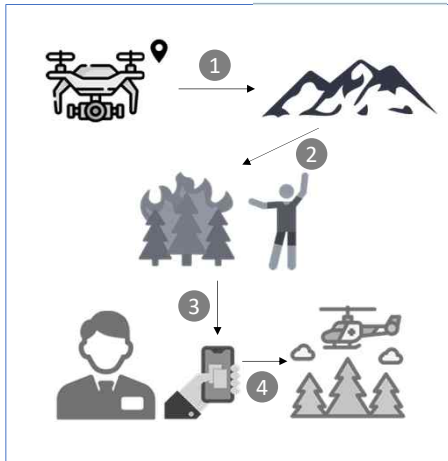
3

## | 2. 시스템 흐름도



4

## | 3. 알고리즘 설명서



## YOLO 기반의 AI DRONE 활용 알고리즘

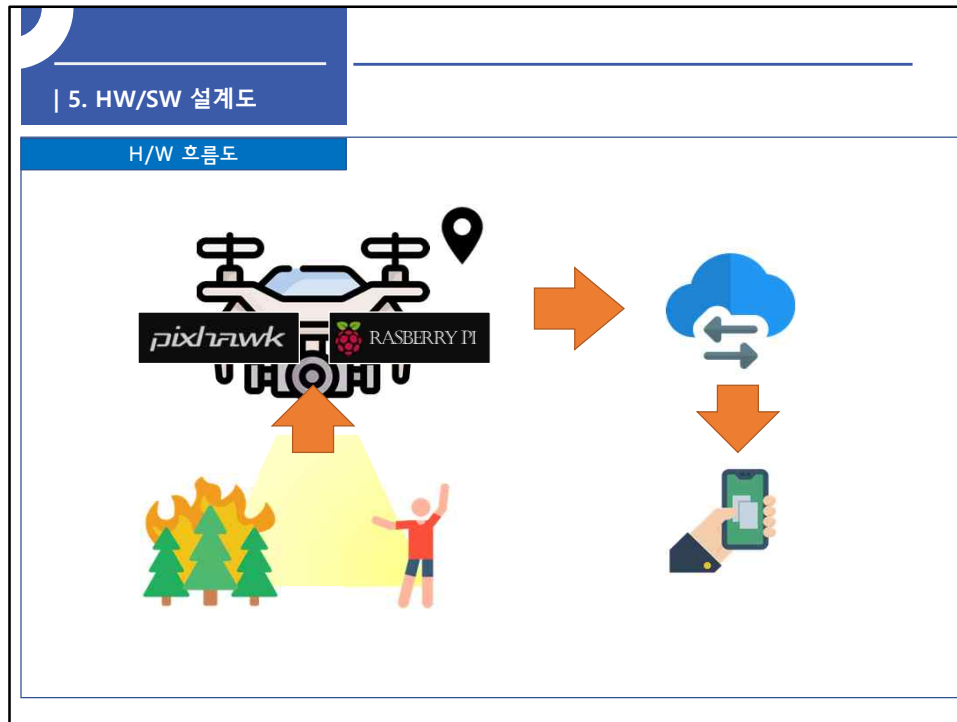
- ① 드론이 수색할 지역을 선택하고 자율비행할 구역을 미리 지정한다.
- ② 위험상황을 학습된 데이터를 바탕으로 식별한다.
- ③ 위험상황이 판단되면 관리자에게 알림이 전송되고 관리자는 실시간 영상으로 확인할 수 있다.
- ④ 산악구조대와 사전에 협력하면 빠른 시간에 출동할 수 있다.

5

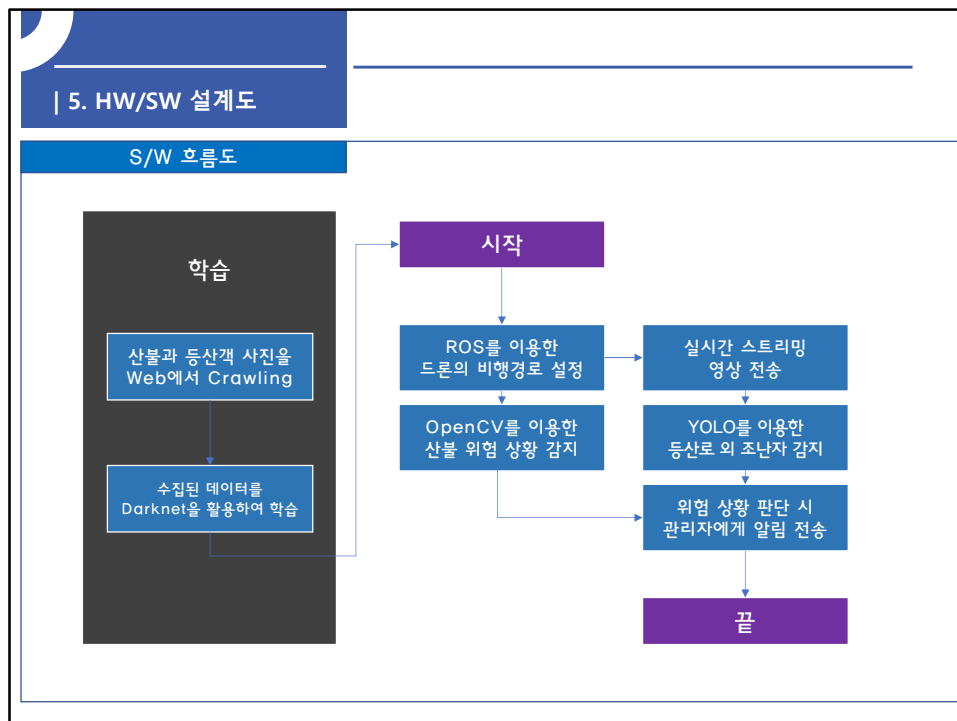
## | 4. 프로그램 목록

구분	기능	설명	구분	기능	설명
S/W	Drone 자율비행	- 로봇용 공개소스 운용체제인 ROS가 설치된 Raspberry pi를 Drone에 부착하여 Pixhawk를 이용해 자율적으로 비행 - 회피센서를 이용해 물체가 감지되면 비행속도를 늦추거나 정지	H/W	Raspberry Pi용 카메라	Drone의 정면 상황을 촬영
	Wi-Fi 통신	Drone에 부착한 Wi-Fi Module을 이용해 실시간 영상을 스트리밍하여 전송		카메라 영상 스트리밍	Camera로 촬영한 영상을 실시간으로 전송. 관리자는 현 상황을 직접 확인 가능
	Darknet YOLO 학습	Darknet을 활용하여 산불과 조난자에 대한 학습 가중치 파일 생성		Drone의 자율 비행 및 수동비행	Pixhawk와 Raspberry Pi의 ROS를 이용해 Drone이 입력된 경로를 따라 자율적으로 비행
	실시간 영상 분석	Drone의 Camera를 통해 촬영되는 영상을 실시간으로 분석하고 목표물 식별			
	알림 전송	상황 발생 시, 사전에 설정되어 있는 관리자의 계정으로 E-Mail 전송			

6



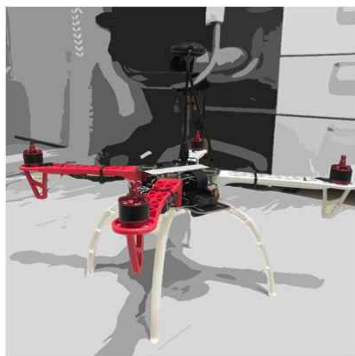
7



8

## | 5. HW/SW 설계도

### 메인 장치



구분	장치	기능
메인장치	쿼드콥터 f450 드론	드론의 기체로 모터, 변속기, 프로펠러 등을 통해 비행이 가능하게 된다.

9

## | 5. HW/SW 설계도

### 핵심 장치



라즈베리파이



픽스호크

구분	장치	기능
핵심 장치	라즈베리파이	다양한 센서를 연결하고 명령을 수행한다.
핵심장치	픽스호크	드론 제어, GPS설정, 비행모드 설정

10

## | 5. HW/SW 설계도

## 주변 장치



구분	장치	기능
주변 장치	리본케이블	카메라가 Drone의 전면 하단부를 바라보게 촬영하게 하기 위한 연장 케이블이다.
주변 장치	Raspberry Pi Camera Module V2	접근이 어려운 지형에 Drone을 날려 상황을 확인하고 분석할 수 있도록 영상을 촬영하는 역할을 수행한다.
주변 장치	GPS	자율비행을 하기 위해 Drone의 위치 정보 신호를 수신하여 지상 혹은 공중의 위치를 파악할 수 있는 시스템이다.
주변 장치	Micro SD card 16GB	Raspberry Pi의 메모리를 늘려 OS를 설치하기 위한 장치이다.
주변 장치	USB to TTL	Raspberry Pi의 pixhawk를 연결하여 비행의 동작을 제어하기 위한 장치이다.

11

## | 6. 개발 환경 및 설명

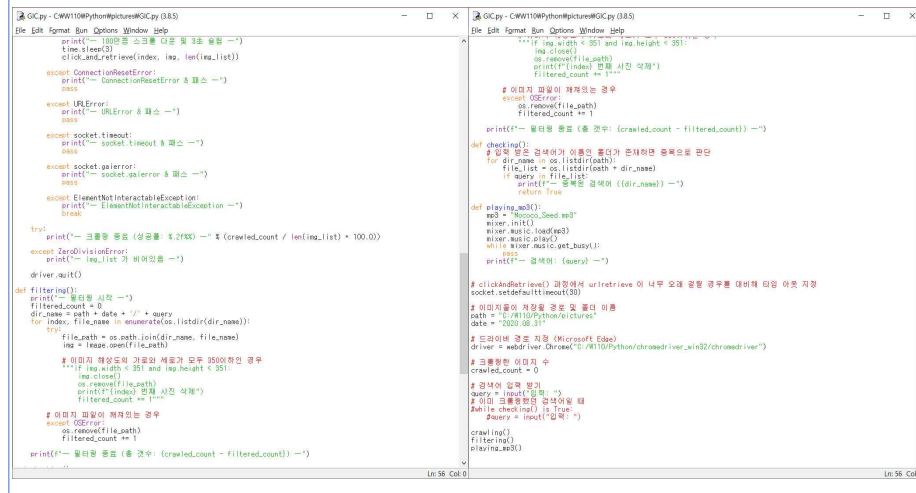
구분	상세내용
S/W 개발환경	OS Ubuntu MATE 16.04 LTS, ROS
	개발환경(IDE) Python IDE, PyCharm, Terminal(nano, vi editor)
	개발도구 Nodejs, Darknet-YOLO
	개발언어 C, Python, Linux
	기타사항 -
H/W 구성장비	디바이스 RaspberryPi, F450 QuadCopter, Pixhawk
	센서 RPI 8MP Camera Module V2
	통신 이론상 HTTP(장비상 wifi)
	언어 C, Python, Linux
	기타사항 -
프로젝트 관리환경	형상관리 <b>Git, Gitlab:</b> 버전관리시스템인 git을 이용하여 open source인 YOLO 코드를 잘못 수정한 경우, git reset 명령어를 이용하여 버전을 이전으로 이동하여 주로 사용하였다. 최종코드는 Gitlab에 올려 모두가 볼 수 있도록 공유하였다.
	의사소통관리 <b>Notion:</b> 팀원들이 개별적으로 작업한 과정들과 부품조립 등을 칸반보드를 이용하여 혼용되지 않도록 관리하였다. <b>카카오톡 채팅방 및 공지방:</b> Notion의 주요 사항들을 확인할 수 있도록 수시로 공지하고, 회의 내용을 빠르게 접근하기 위해 사용되었다.
	기타사항 -

12



## | 7. 핵심소스코드 (2)

## [Python] Web Crawling (2)



```

G:\Python - CWW10MPython\picture\GCPy (3.8.5)
File Edit Format Run Options Window Help
print("1000원 소액 결제 성공 및 3초 뒤 결제")
time.sleep(3)
click_and_enter(index, img, len(img_list))

except ConnectionResetError:
    print("ConnectionResetError & 패스")
    pass

except URLError:
    print("URLError & 패스")
    pass

except socket.timeout:
    print("socket.timeout & 패스")
    pass

except socket.gaierror:
    print("socket.gaierror & 패스")
    pass

except ElementNotInteractableException:
    print("ElementNotInteractableException")
    break

try:
    print("크롤링 종료 (성공률: %.2f%%) - %d (crawled_count / len(img_list) * 100.0)"
    except ZeroDivisionError:
        print("img_list 가 비어있음")

driver.quit()

def filtering():
    print("필터링 시작")
    filtered_count = 0
    dir_name = path + date + "/" + query
    for index, file_name in enumerate(os.listdir(dir_name)):
        file_path = os.path.join(dir_name, file_name)
        img = image.open(file_path)
        # 이미지 해상도의 가로와 세로가 모두 3500 이하인 경우
        if img.width < 351 and img.height < 351:
            img.close()
            os.remove(file_path)
            print("이미지 해상도 문제 삭제")
            filtered_count += 1
        # 이미지 파일이 깨져있는 경우
        except OSError:
            os.remove(file_path)
            filtered_count += 1
    print("필터링 종료 (총 갯수: (crawled_count - filtered_count))")

G:\Python - CWW10MPython\picture\GCPy (3.8.5)
File Edit Format Run Options Window Help
print("1000원 소액 결제 성공 및 3초 뒤 결제")
time.sleep(3)
click_and_enter(index, img, len(img_list))

except ConnectionResetError:
    print("ConnectionResetError & 패스")
    pass

except URLError:
    print("URLError & 패스")
    pass

except socket.timeout:
    print("socket.timeout & 패스")
    pass

except socket.gaierror:
    print("socket.gaierror & 패스")
    pass

except ElementNotInteractableException:
    print("ElementNotInteractableException")
    break

try:
    print("크롤링 종료 (성공률: %.2f%%) - %d (crawled_count / len(img_list) * 100.0)"
    except ZeroDivisionError:
        print("img_list 가 비어있음")

driver.quit()

def filtering():
    print("필터링 시작")
    filtered_count = 0
    dir_name = path + date + "/" + query
    for index, file_name in enumerate(os.listdir(dir_name)):
        file_path = os.path.join(dir_name, file_name)
        img = image.open(file_path)
        # 이미지 해상도의 가로와 세로가 모두 3500 이하인 경우
        if img.width < 351 and img.height < 351:
            img.close()
            os.remove(file_path)
            print("이미지 해상도 문제 삭제")
            filtered_count += 1
        # 이미지 파일이 깨져있는 경우
        except OSError:
            os.remove(file_path)
            filtered_count += 1
    print("필터링 종료 (총 갯수: (crawled_count - filtered_count))")

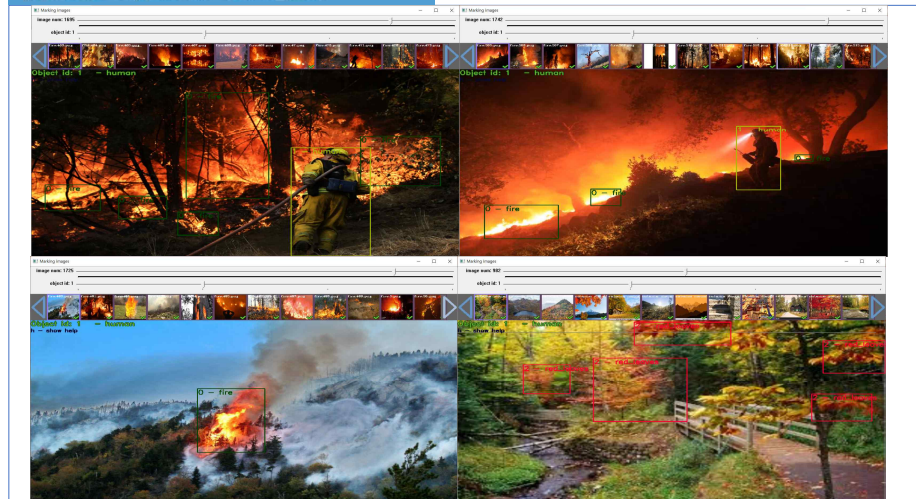
def checking():
    # 검색 결과 검색어가 포함된 폴더가 존재하면 중복으로 판단
    for dir_name in os.listdir(path):
        file_list = os.listdir(path + dir_name)
        if query in file_list:
            print("중복 검색어 (%s) - %s" % (dir_name, query))
            return True
    return False

def playing_music():
    mp3 = "music\\Seed.mp3"
    mixer.init()
    mixer.music.load(mp3)
    mixer.music.play(1)
    while mixer.music.get_busy():
        print("음악 재생 중")
    # clickAndRetrieve() 과정에서 urllibretrieve 이 너무 오래 실행 경우를 대비해 타임 아웃 지정
    socket.setdefaulttimeout(30)
    # 이미지가 저장될 경로 및 폴더 이름
    path = "E:/100Python/chromedriver_win32/chromedriver/"
    date = "2020.08.31"
    # 드라이버 로드 지정 (Microsoft Edge)
    driver = webdriver.Chrome("E:/100Python/chromedriver_win32/chromedriver")
    # 크롤링할 이미지 수
    crawled_count = 0
    # 검색어 입력 받기
    query = input("검색어: ")
    # 이미 크롤링한 것 체크할 때
    while checking() is True:
        query = input("검색어: ")
    crawling()
    filtering()
    playing_music()
  
```

15

## | 7. 핵심소스코드 (3)

## [Python/C++] Darknet : YOLO\_mark



16



18

한이음

## | 7. 핵심소스코드 (5)

### [C] 실시간 영상 Streaming & 위험상황 선별 (2)

```

82 pt_text.y = top - 4 * font_size;
83 pt_text_bg1.x = left;
84 pt_text_bg1.y = top - (3 * 18 * font_size);
85 pt_text_bg2.x = right;
86 if (right - left) < text_size.width pt_text_bg2.x = left +
87   text_size.width;
88 pt_text_bg2.y = top;
89 cv::Scalar color;
90 color.val[0] = red * 255;
91 color.val[1] = green * 255;
92 color.val[2] = blue * 255;
93 cv::rectangle(show_img, pt1, pt2, color, width, 8, 0);
94 if (text_output)
95   printf("\t\tleft: %d top: %d width: %d height: %d\n",
96     (float)left, (float)top, b.w * show_img->cols, b.h * show_img->rows);
97   else
98     printf("\n");
99 cv::rectangle(show_img, pt_text_bg1, pt_text_bg2, color, width, 8, 0);
100 cv::rectangle(show_img, pt_text_bg1, pt_text_bg2, color, CV_FILLED, 8, 0);
101 // filled
102 cv::Scalar black_color = CV_RGB(0, 0, 0);
103 cv::putText(show_img, labelstr, pt_text,
104   cv::FONT_HERSHEY_COMPLEX_SMALL, font_size, black_color, 2 *
105   font_size, CV_AA);
106 // cv::FONT_HERSHEY_COMPLEX_SMALL, cv::FONT_HERSHEY_SIMPLEX
107 }
108 }
109 if (text_output) {
110   fflush(stdout);
111 }
112 } catch (...) {
113   cerr << "OpenCV exception: draw_detections_cv_v01\n";
114 }
115 }

```

산불 인식 구현은 동영상으로 대체했습니다.

19

한이음

## | 7. 핵심소스코드 (6)

### [C] 상황 발생 시 E-Mail을 통한 알람 Service

```

1 import smtplib
2 import time
3 from email.mime.text import MIMEText
4 from datetime import datetime
5
6 def send_email():
7     #이메일 발송
8     sendEmail = "withfavor@naver.com"
9     recvEmail = "withfavor@naver.com"
10    password = "withfavor881"
11
12    #smtp 서버 주소
13    smtpName = "smtp.naver.com"
14    #smtp 포트 번호
15    smtpPort = 587
16
17    #이메일 내용(오늘 날짜)
18    text = "" + str(datetime.today()) + "InForest Fire Detection!A dangerous situation has
19    been reported to 119."
20    msg = MIMEText(text) #MIMEText(text, charset="utf8")
21
22    #이메일 제목
23    msg['Subject'] = "FIRE WARNING from DOLO"
24    msg['from'] = sendEmail
25    msg['to'] = recvEmail
26    print(msg.as_string())
27
28    #이메일 서버 연결
29    s = smtplib.SMTP(smtpName, smtpPort)
30    s.starttls()
31    s.ehlo()
32    s.login(sendEmail, password)
33
34    #이메일 전송, 수신명 변경
35    s.sendmail(sendEmail, recvEmail, msg.as_string())
36    #smtp 서버 연결 종료
37    s.close()
38
39 pre = 0
40 cur = 0
41

```

필요한 라이브러리 삽입

이메일 보내기 함수 구현

3초마다 상황 파악

```

while True:
    #이메일 발송
    f = open("C:/Users/withfavor/Desktop/PythonProjects/SMTP_Test/text.txt", "r")
    cur = f.readline()
    res = int(cur)-int(pre)
    print("noopen")
    if res >= 10:
        send_email()
    pre = cur
    f.close()
    time.sleep(3)

```

20



21