



# 위험비행물(드론) 감지시스템

실시간 카메라로 확인하여 위험비행물(드론)과 새를 이진 분류하여  
위험물 발견 시 위험물의 GPS 정보를 경보메세지와 함께 이메일로 사용자에게 제공



# 01. 진행 상황

데이터 수집

Drone, Bird 데이터 수집 및 분류



모델 훈련 및 테스트

Swin\_t, ResNet, DeiT, VGG



모델 논문 리뷰

위의 이미지 분류 모델을 제외한 모델



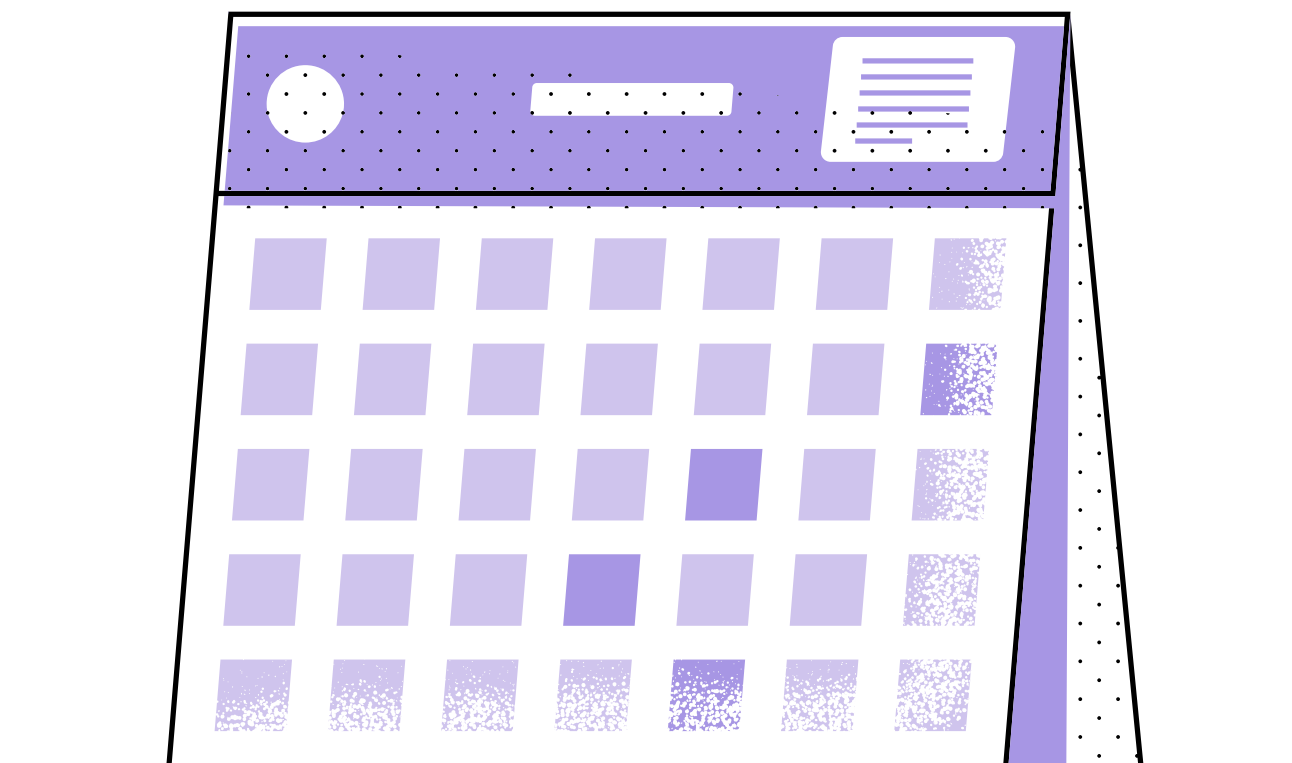
Webcam

위험물(드론) 감지를 위한 threshold 설정



서비스 구현

GPS, 경보 메시지 이메일로 전송



# ReXNet

## - 특징 + 기존 모델과 차이점

1. 활성화 함수로 ReLU가 아닌 Swish, SiLU를 사용(실험해보니 가장 성능이 좋았다는 차트 제시)
2. 기존엔 대량의 fully connected layer를 한 번에 작게 줄여왔었음(ex. vgg16 : 4096 --> 1000 features)
3. ReXNet모델에서는 차원(총)-차원(총)을 넘어가는 단계에서 기존의 최솟값에서 상관성(rank)이 높은 것이 채택되도록 함

## - 앞으로 계획

논문의 수학적까지는 안 되더라도 모델의 전체적인 구조와 어떤 점이 구체적으로 개선된 것인지 이해 필요

# CoANet

## -기존 모델과의 차이점

1. Convolutional Networks +Transformer ,  
두 아키텍처의 장점을 결합한 하이브리드 모델
2. 컨볼루션층과 어텐션층을 수직으로 쌓아 일반화와 용량, 효율성을 대폭 개선

## -앞으로의 계획

- 심층적인 논문 리뷰를 위한 inverted bottleneck, Attention 매커니즘에 대한 학습

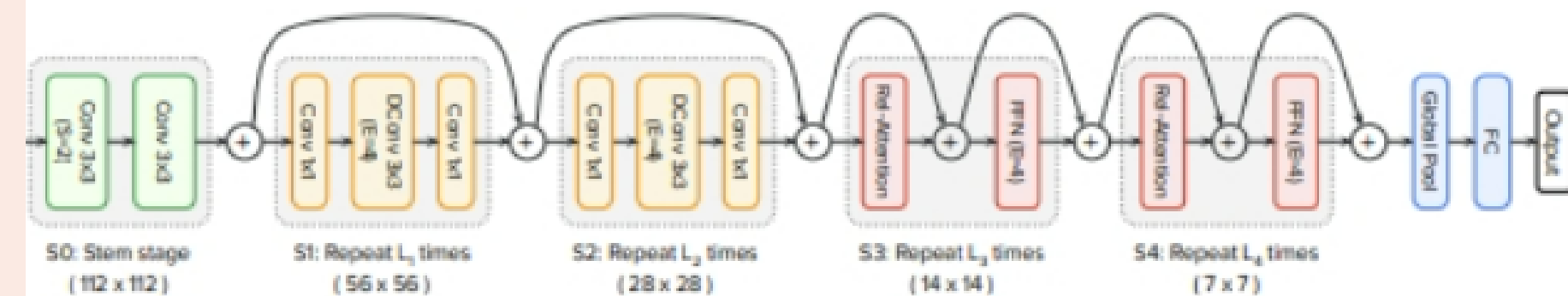


Figure 4: Overview of the proposed CoAtNet.



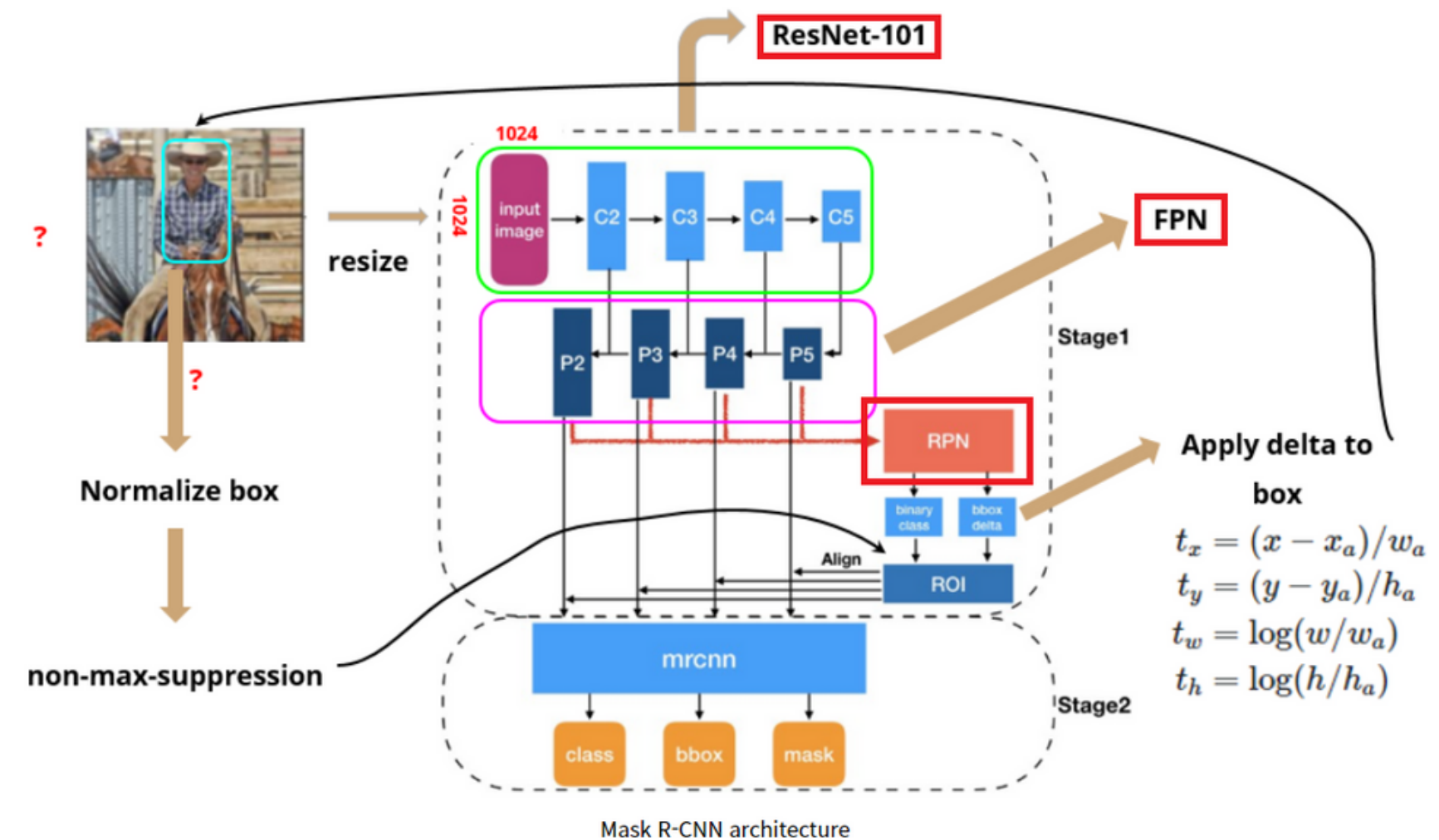
# Mask R-CNN

## - 특징 + 기존 모델과 차이점

1. Fast R-CNN의 classification, localization(bounding box regression) branch에 새롭게 mask branch가 추가
2. RPN 전에 FPN(feature pyramid network)가 추가
3. Image segmentation의 masking을 위해 RoI align이 RoI pooling을 대체

## - 앞으로 계획

1. Mask R-CNN 모델의 개선점을 학습하기 위해 이 전 모델인 Fast R-CNN, RPN, FPN 학습 필요
2. Making의 한 종류인 RoI pooling 학습 필요



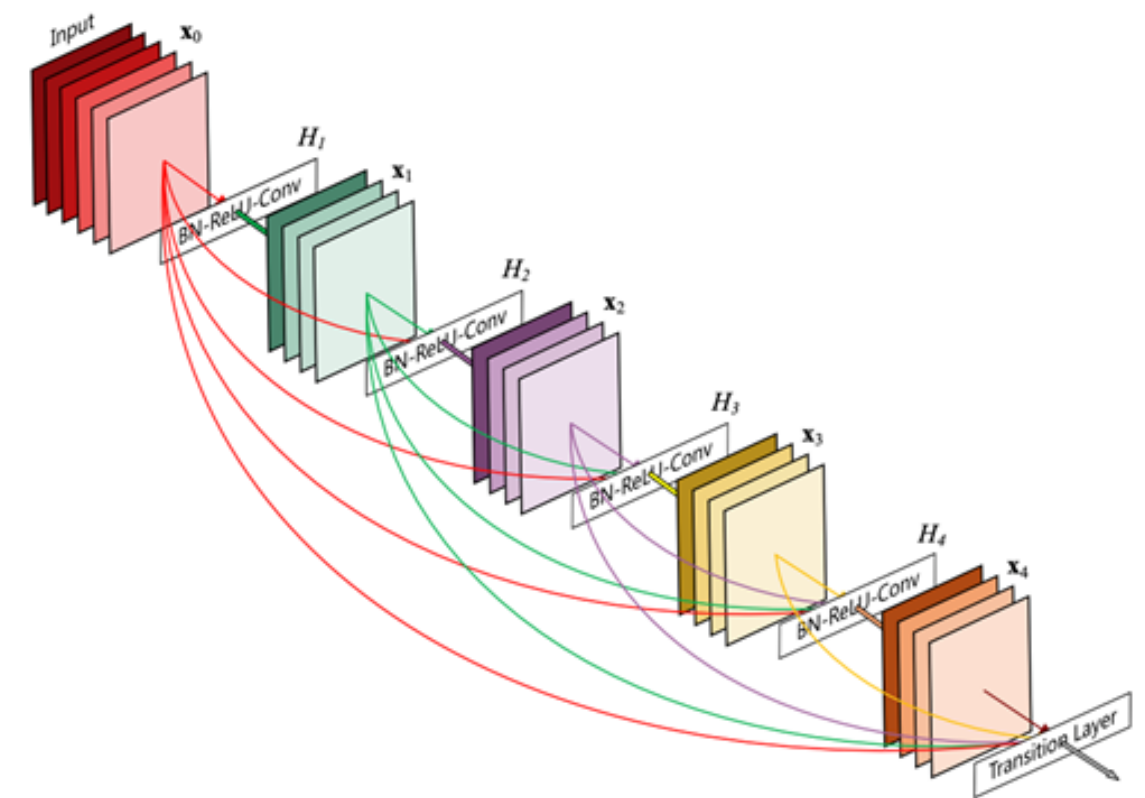
# DenseNet

## - 특징 + 기존 모델과 차이점

1. DenseNet은 ResNet과 Pre-Activation ResNet보다 적은 파라미터 수로 더 높은 성능을 가진 모델
2. 이전 layer들의 feature map을 계속해서 다음 layer의 입력과 연결하는 아이디어는 ResNet과 같은데 ResNet에서는 feature map끼리 더하기를 해주는 방식이었다면 DenseNet에서는 feature map끼리 concatenation 한다.
3. DenseNet 장점  
vanishing-gradient 개선 / feature propagation 강화 / Feature Reuse  
parameter 의 수 절약

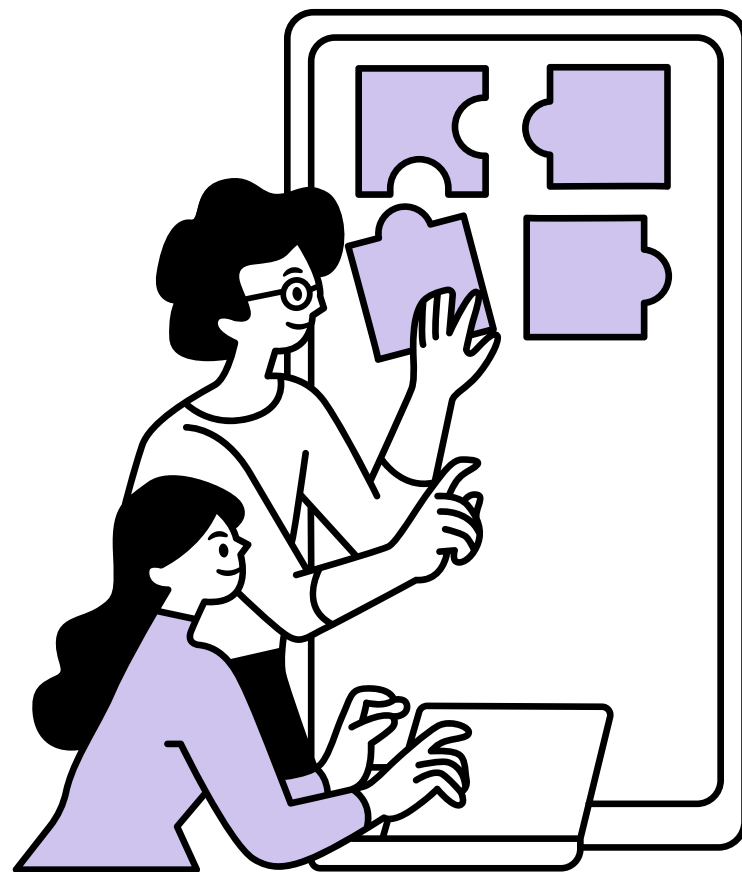
## - 앞으로 계획

DenseNet에서 사용하기 위해 BottleNeck와 Transition block를 정의하고 모델을 구현해본다.



**Figure 1:** A 5-layer dense block with a growth rate of  $k = 4$ . Each layer takes all preceding feature-maps as input.

# 이메일 및 GPS 위치 서비스 현황



## 1. 이용자 데이터 읽기

- .csv파일로 저장된 서비스 이용자 이메일 수집

## 2. CCTV를 통한 드론이 관측

- 이진분류를 통해 상공에 떠 있는 물체 확인

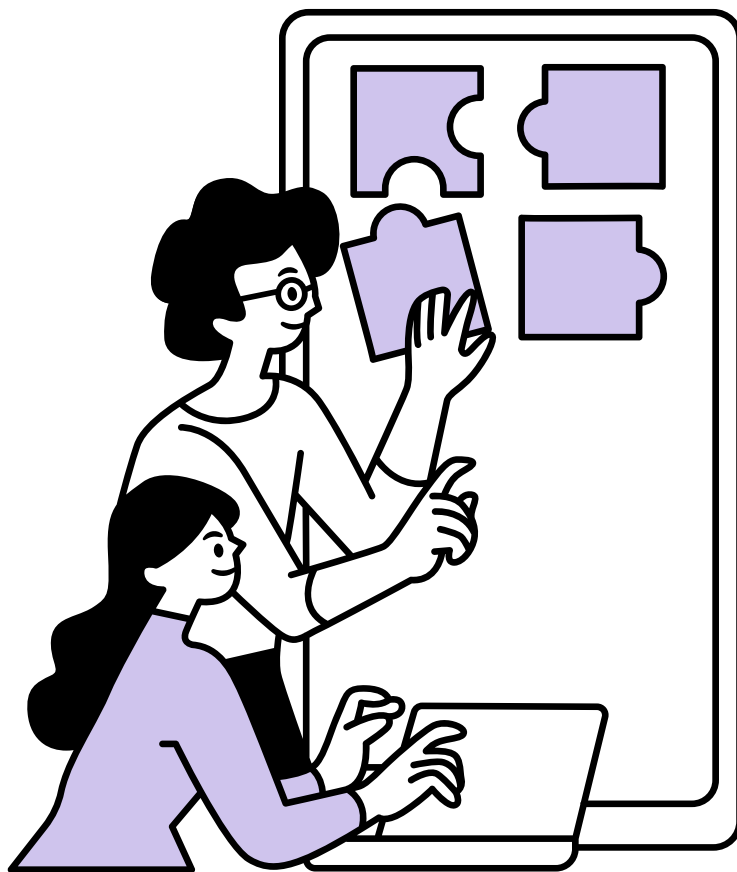
## 4. 드론 관측 이메일 전송

- 드론이 관측된 GPS 값과 경고 메시지를 이용자 Email로 전송
- smtplib 라이브러리 사용

## 3. CCTV IP정보로 주소 획득

- requests와 json, geopy 라이브러리들을 통해 드론이 관측된 CCTV의 IP 주소 및 실주소 획득

# 이메일 및 GPS 위치 서비스 현황



```
import pandas as pd

from email.mime.text import MIMEText
from geopy.geocoders import Nominatim

# 현재 좌표값을 얻기 위한 함수
def current_location():
    here_req = requests.get("http://www.geoplugin.net/json.gp")

    if (here_req.status_code != 200):
        print("unknown address")
    else:
        location = json.loads(here_req.text)
        crd = {"lat": str(location["geoplugin_latitude"]), "lng": str(location["geoplugin_longitude"])}

    return crd

def geocoding_reverse(lat_lng_str):
    geocoder = Nominatim(user_agent = 'South Korea', timeout=None)
    address = geocoder.reverse(lat_lng_str)

    return address

def get_location():
    crd = current_location()

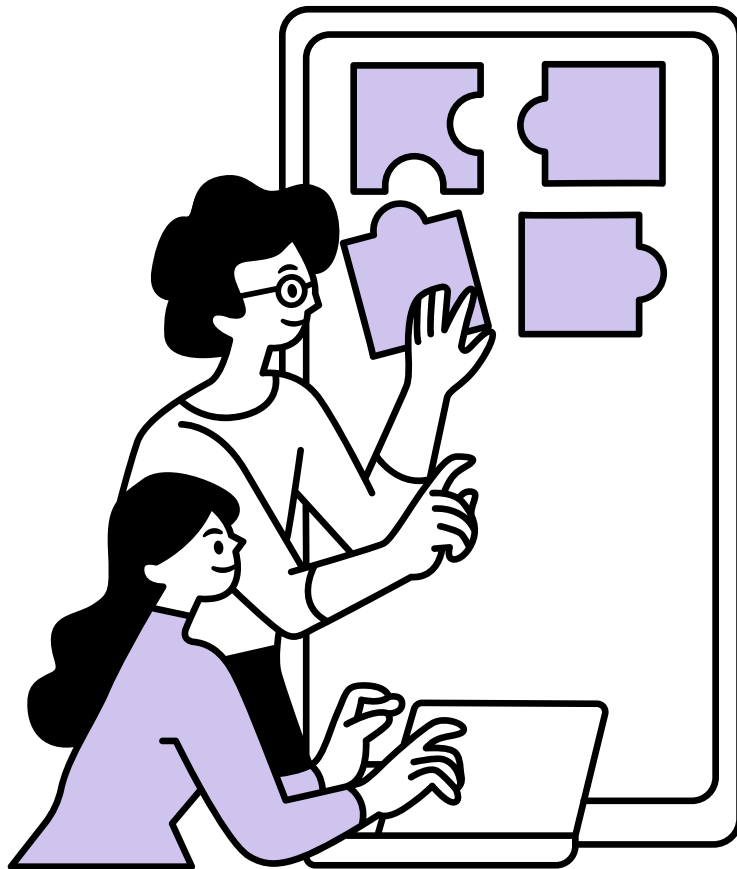
    lat = crd['lat']
    lng = crd['lng']
    location = lat+', '+lng

    address = geocoding_reverse(location)

    return address
```



# 이메일 및 GPS 위치 서비스 현황



```
# 현재 좌표와 경고 문구를 이메일로 보내주는 함수
def get_customer():
    data = []
    customer_data = './customer.csv'
    data = pd.read_csv(customer_data)
    print( data )
    return data

def send_alarm():
    # 이메일 로그인
    s = smtplib.SMTP('smtp.gmail.com', 587)
    s.starttls()
    private_data = './userdata.svm'
    sender_account = private_data[0]
    password = private_data[1]
    s.login(sender_account, password)

    # 고객 이메일 정보와 드론 발견 주소 입력

    address = get_location()

    customers = get_customer()
    for users in customers:
        # 보낼 메시지 설정
        msg = MIMEText(f"***경고***\n현재 아래 좌표에서 드론이 관측되었습니다.\n {address}")
        msg['Subject'] = '제목 : 드론 경고 알람입니다.'

        # 메일 보내기
        s.sendmail(sender_account, users, msg.as_string())

    # 세션 종료
    s.quit()
```

# 감사합니다

Team 02.

여익수, 권태윤, 이승윤, 손병구, 최유연

