



# 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



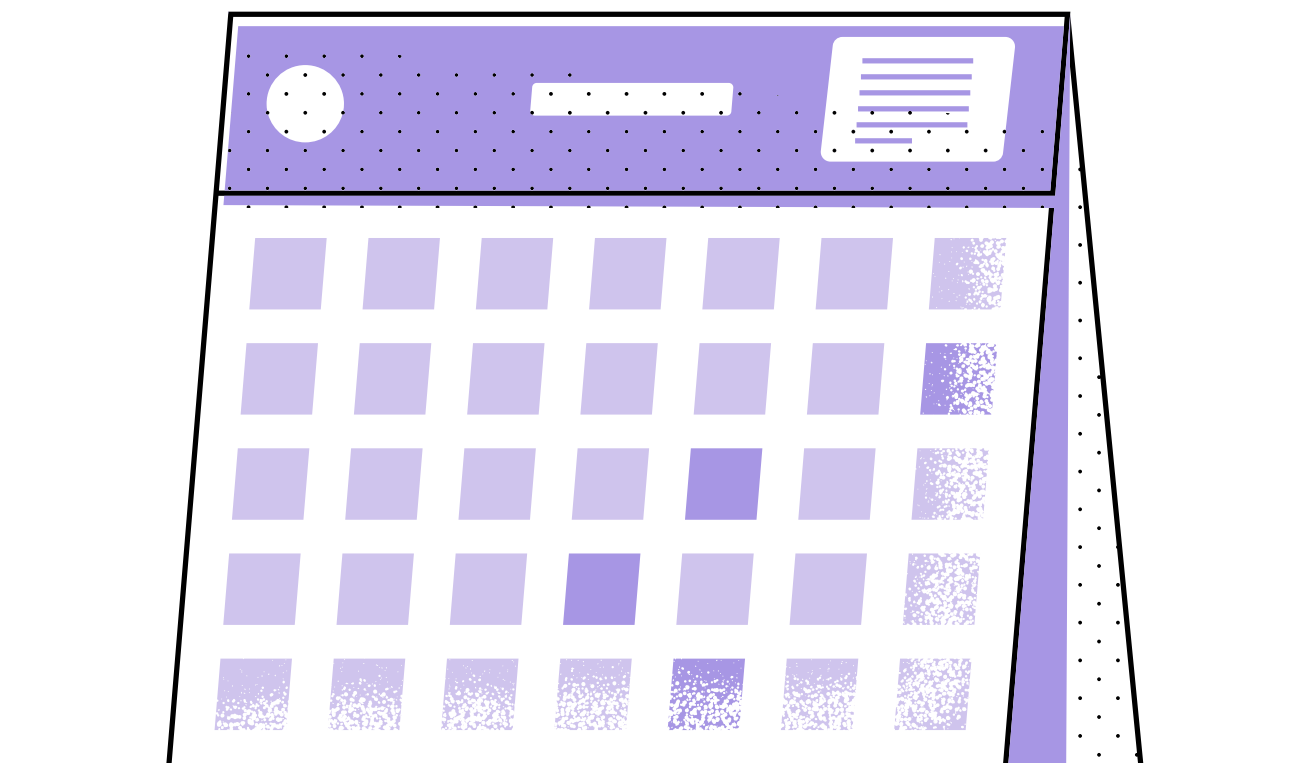
Webcam

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

# CoAtNets

## -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

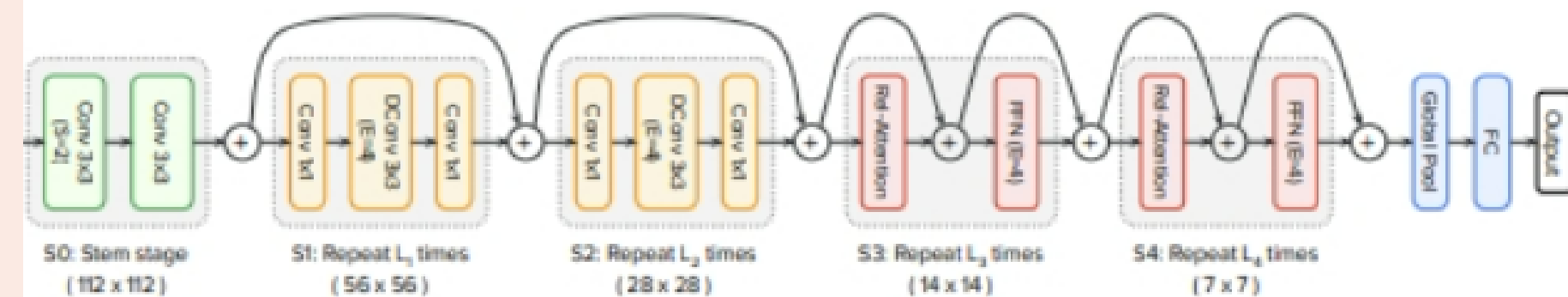


Figure 4: Overview of the proposed CoAtNet.



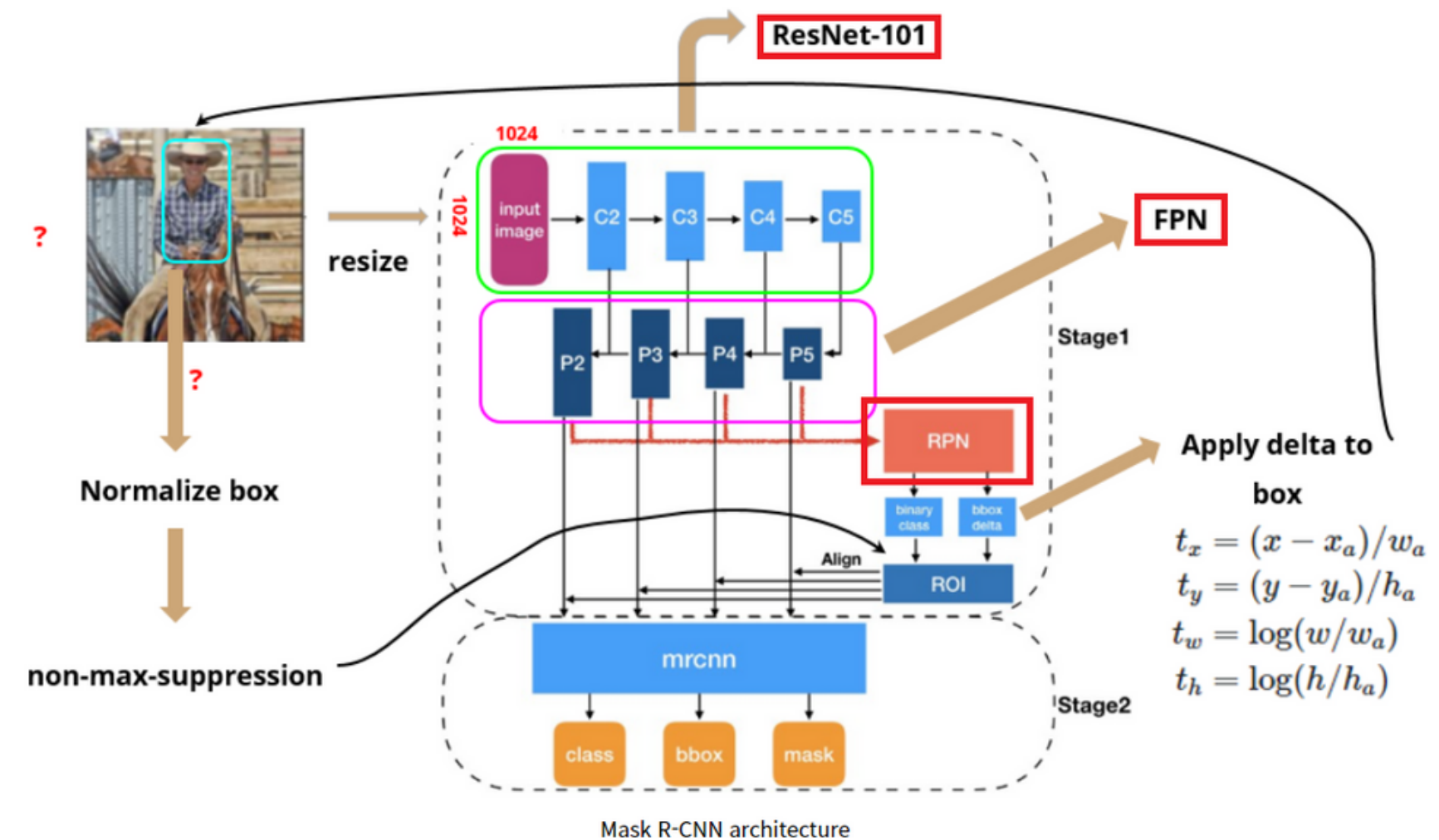
# Mask R-CNN

## - 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



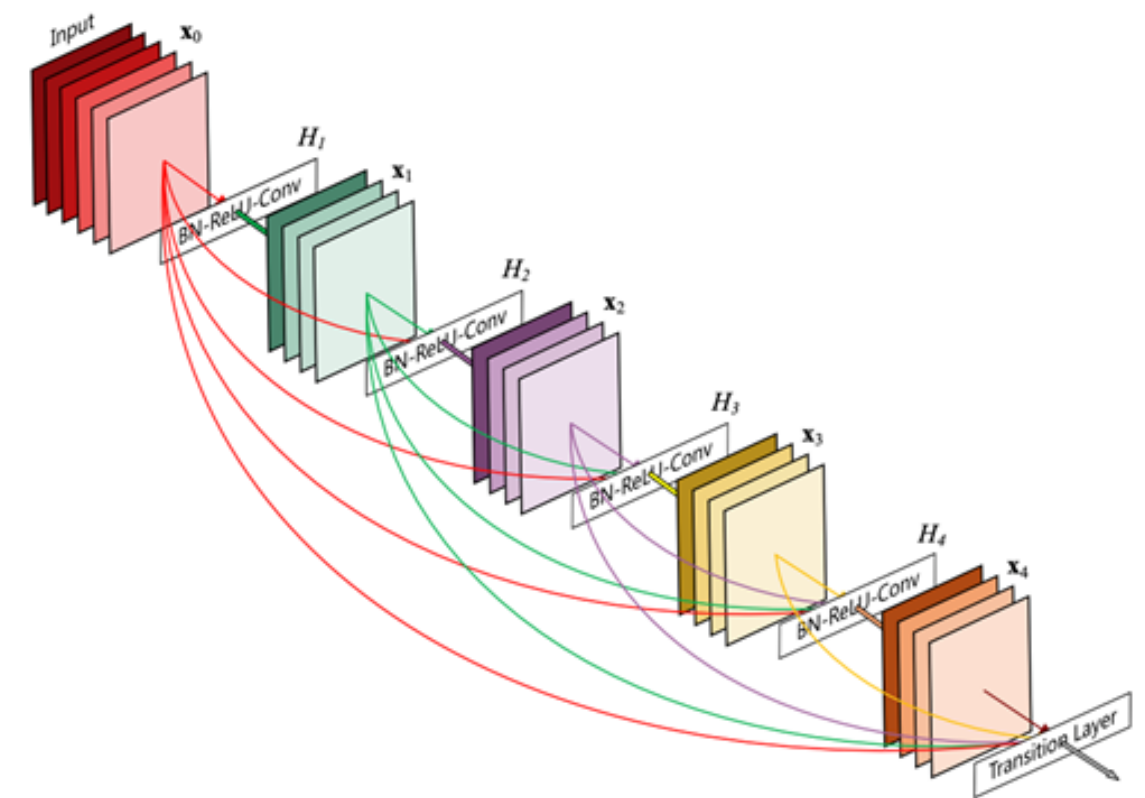
# DenseNet

## - 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.

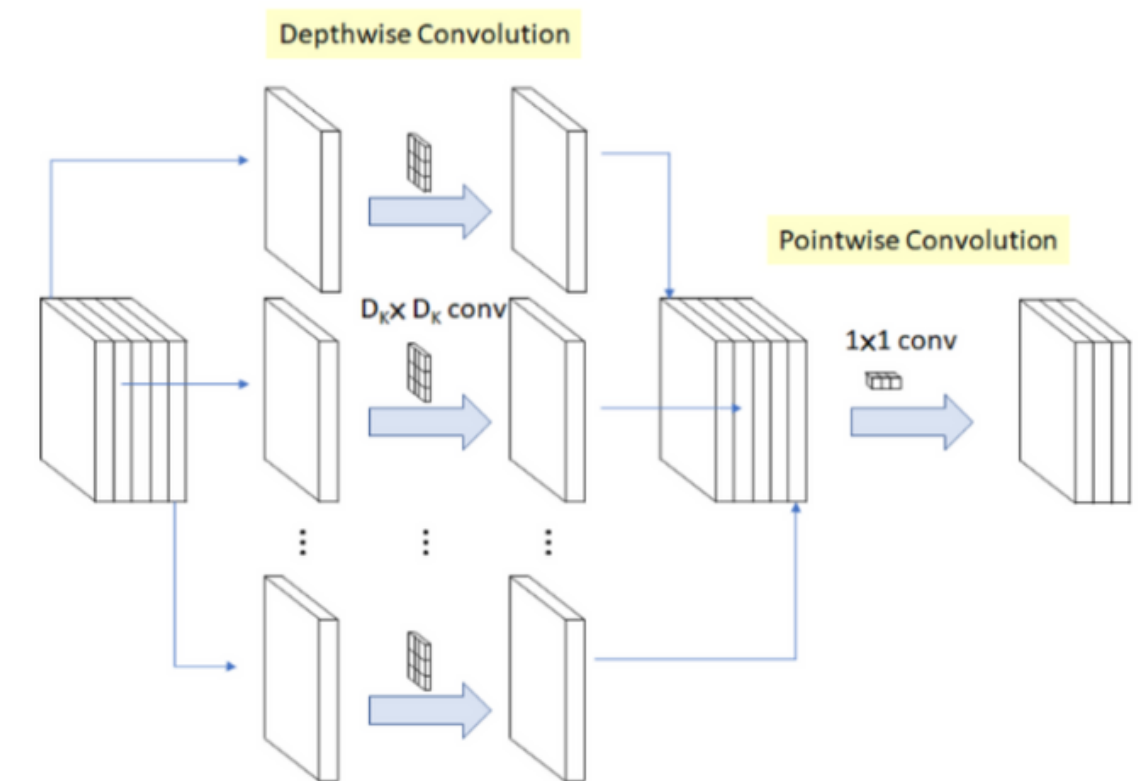
# MobileNet

## - 특징 + 기존 모델과 차이점

1. MobileNet은 컴퓨터 성능이 제한되거나 배터리 퍼포먼스가 중요한 곳에서 사용될 목적으로 설계된 CNN 구조
2. Depthwise Separable Convolutions 과 Width Multiplier & Resolution Multiplier 파라미터를 사용해서 모델의 경량화
3. MobileNet은 그 자체로도 이미 경량화 되어 있지만 상황에 따라 더 경량화하기 위해 두 개의 hyper-parameter 활용해 네트워크 크기를 감소.

## - 앞으로 계획

- 모델 경량화를 위한 다양한 Convolution을 학습해 Width Multiplier 에 대한 이해
- Width Multiplier과 Resolution의 차이에 따른 Accuracy와 연산량의 차이 확인



# 이미지 분류 모델 테스트 결과

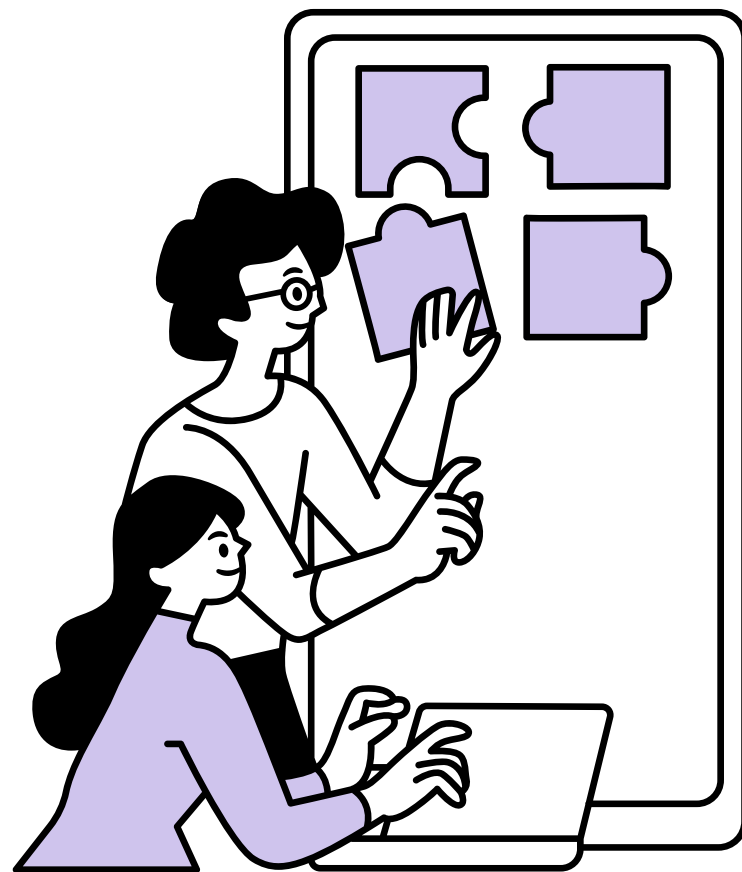
model name	optimizer	lr	batch size	epochs	train loss	train acc	val acc	test acc	.pt size
VGG16	AdamW	0.0001	128	5	0.217	0.992	0.973	99.19	512MB
Swin-T	AdamW	0.0001	32	10	0.227	0.985	0.986	99.73	107.8MB
Swin-B	AdamW	0.0001	16	10	0.217	0.99	0.989	99.73	339.4MB
ReXNet 1.0	AdamW	0.001	128	10	0.243	0.977	0.984	99.72	13.7MB
ResNet50	AdamW	0.0001	64	10	0.242	0.977	1	99.46	92.2MB
ResNet18	AdamW	0.0001	128	10	0.255	0.973	0.986	97.83	42.7MB
EfficientNetB3	AdamW	0.001	16	10	0.262	0.967	0.993	99.77	41.3MB
DeiT	Adam	0.0001	16	10	0.002	0.945	0.984	97.27	21.8MB
ConvNeXT_large	NAdam	0.0001	16	10	0.218	0.99	0.986	100.00	748MB
MobileNetV3_large	AdamW	0.0001	256	10	0.246	0.977	0.984	99.19	16.6 MB
MobileNetV4_large	Adam	0.0001	128	10	0.102	0.991	0.992	0.99	16.6MB
MobileNetV5_large	Adam	0.0001	256	10	0.027	0.992	0.981	0.98	16.6MB
MobileNetV6_large	Adam	0.001	128	10	0.046	0.991	0.992	0.99	16.MB



# MobileNet 모델 테스트 결과

model name	optimizer	lr	batch size	epochs	train loss	train acc	val acc	test acc	.pt size
MobileNetV2	AdamW	0.001	200	10	0.246	0.977	0.992	0.9891	8.72MB
MobileNetV3_large	Adam	0.0001	64	10	0.052	0.989	0.992	0.989	16.6MB
MobileNetV3_large	AdamW	0.0001	256	10	0.246	0.977	0.984	99.19	16.6 MB
MobileNetV4_large	Adam	0.0001	128	10	0.102	0.991	0.992	0.9892	16.6MB
MobileNetV5_large	Adam	0.0001	256	10	0.027	0.992	0.981	0.9783	16.6MB
MobileNetV6_large	Adam	0.001	128	10	0.046	0.991	0.992	0.9864	16.MB

# 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

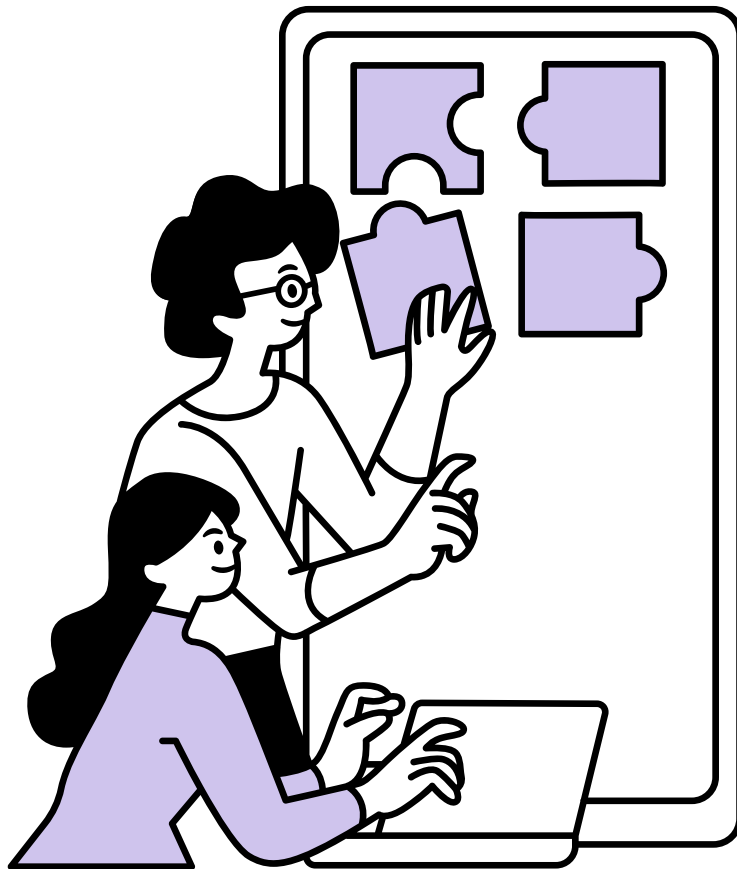
## 4. 드론 관측 이메일 전송

- 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

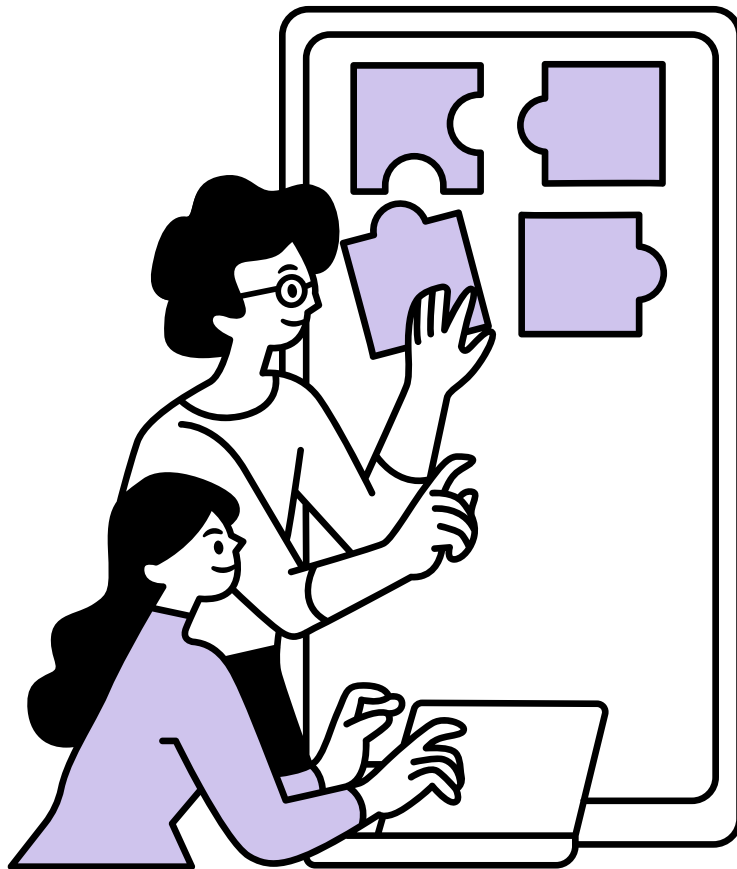
def get_location():
    crd = current_location()

    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()
```

# 서비스 구현 이슈

## - 서비스에 사용할 최종 모델 선정

- 현재까지 실험에 사용한 모델
  - MobileNet, Convnetx, Efficient, Resnet, ReXNet, swin, Vgg 계열류 총 16종 모델
- 이 중 최종적으로 MobileNet V2, V3 계열로 선정
  - 채택 이유 1. 전체적으로 매개변수 및 optimizer를 가리지 않고 균일하게 97%를 넘는 Test Accuracy를 보여주고 있음
  - 채택 이유 2. 서비스 제공 상황이 보통 좋은 성능의 환경이라고 기대하기 어려움(CCTV, 기타 관측 장비)

이에 최종 출력되는 모델이 가벼워야 할 것으로 판단(경량화된 모델 필요)

## - 앞으로 계획

- 기기 단에서 실제로 배포, 구동하기 위한 방법 고려
- 현재 위치 호출 관련해서 한국내 주소가 정확히 잡히지 않는 문제가 있어 이에 대한 해결 필요

# 감사합니다

Team 02.

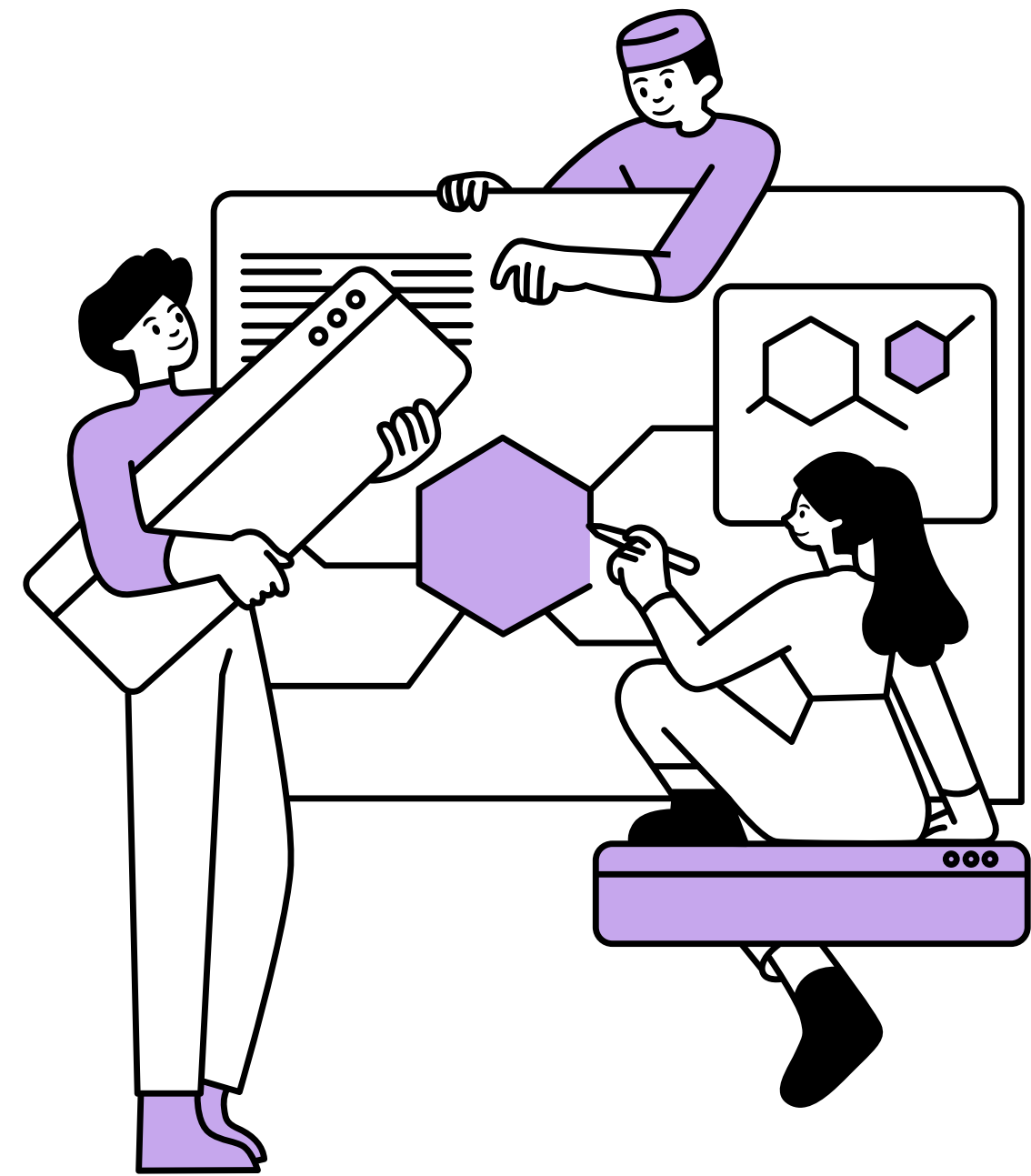
여익수, 권태윤, 이승윤, 손병구, 최유연





# 목차

- 01. 기획 의도 & 목적
- 02. 프로젝트 계획
- 03. 데이터 현황
- 04. 시스템 구성도
- 05. 팀원 역할 분담
- 06. 모델 선정, 모델 테스트 결과
- 07. 간단한 시연
- 08. 추후 서비스 방향성



# 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



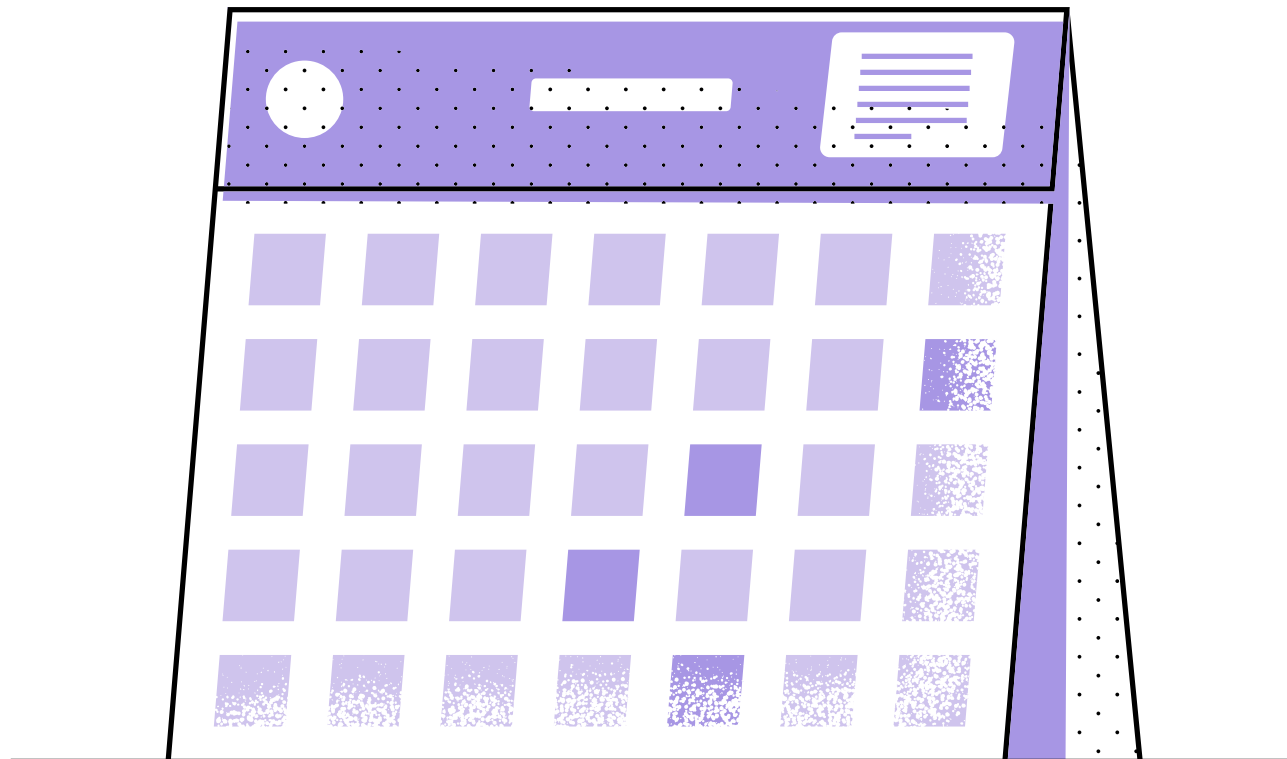
## Webcam

위험물(드론) 감지를 위한 threshold 설정



## 서비스 구현

GPS, 경보 메시지 이메일로 전송



# 01. 기획 의도 & 목적



## 국가 안보

---

위험물을 탐지하여  
빠르게 위험 사항에 대처



## 실시간 탐지

---

실시간 카메라 탐지를  
통해 효율성 향상

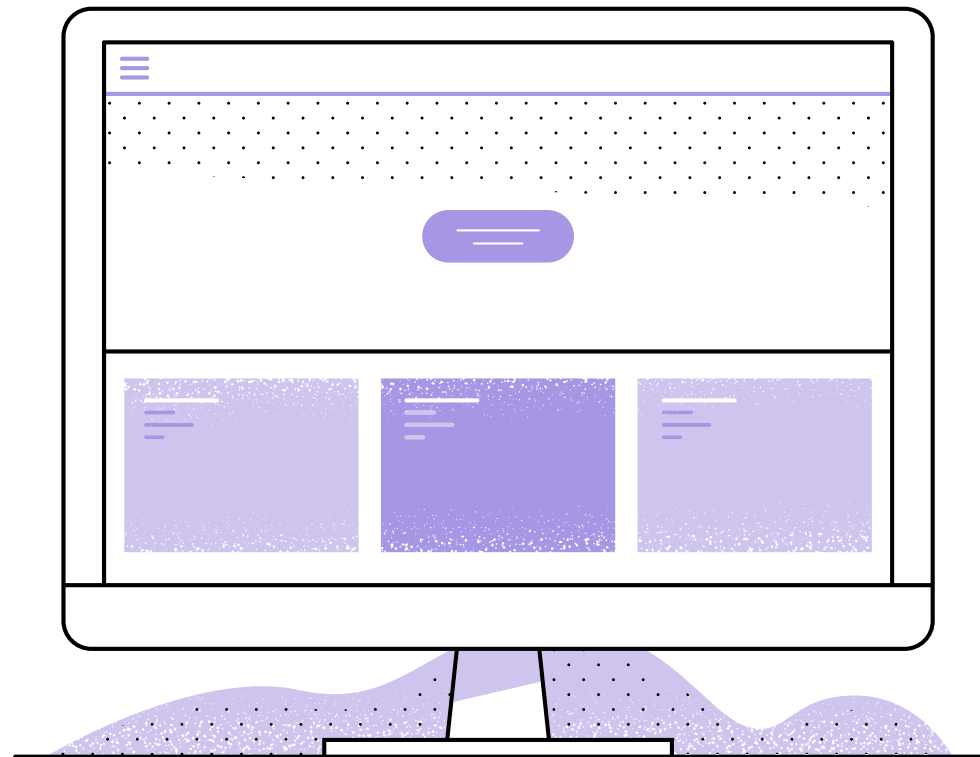


## 경보 시스템

---

GPS를 경보메세지와 함께  
이메일로 전송하여  
사용자에게 정보 제공

## 02. 프로젝트 계획



### 1. 데이터 수집 & 정제화

- 드론과 새 데이터 수집, resize, split
- 1월 10일(화) 오전

### 2. 이진 분류

- 드론과 새를 분류하는 이진 분류 모델훈련
- 1월 10일(화) 오후 ~ 1월 11일(수) 오전

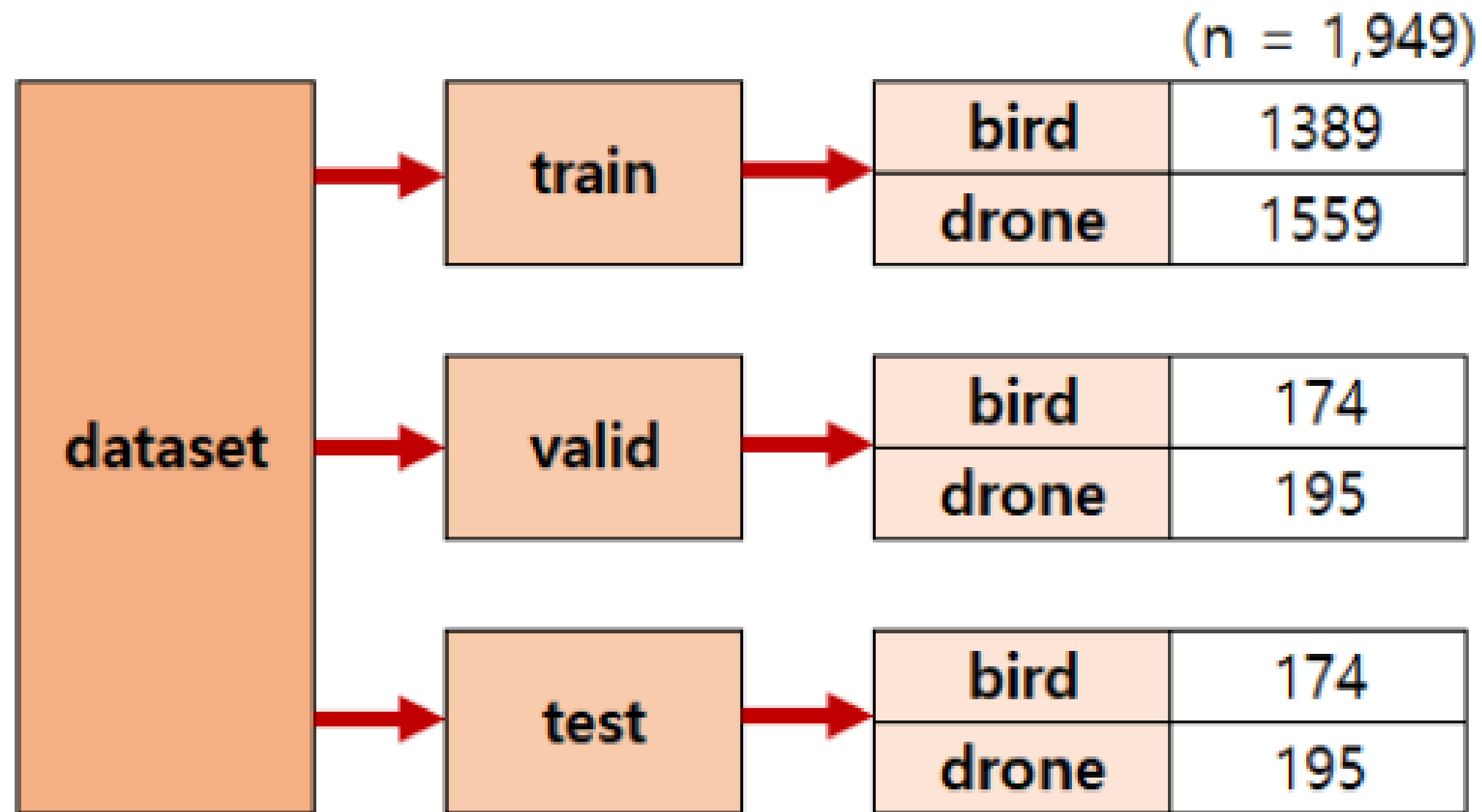
### 3. 실시간 위험물 탐지

- Webcam으로 실시간 위험물 탐지
- 1월 11일(수) 오후

### 4. 경보 메시지

- 위험물의 GPS를 Email로 전송
- 1월 12일(목) 오전 오후

# 03. 데이터 현황



## 04. 시스템 구성도



# 05. 역할 분담



**여익수**

발표 자료 준비, Augmentation 수정,  
Train & Test 스크립트 수정

**권태운**

파이썬 스크립트를 사용하여 Webcam  
(CCTV)의 GPS를 이메일로 전송

**이승윤**

RAW 데이터셋 구축, pretrain (resize,  
split), Web 캠(CCTV) 스크립트 수정

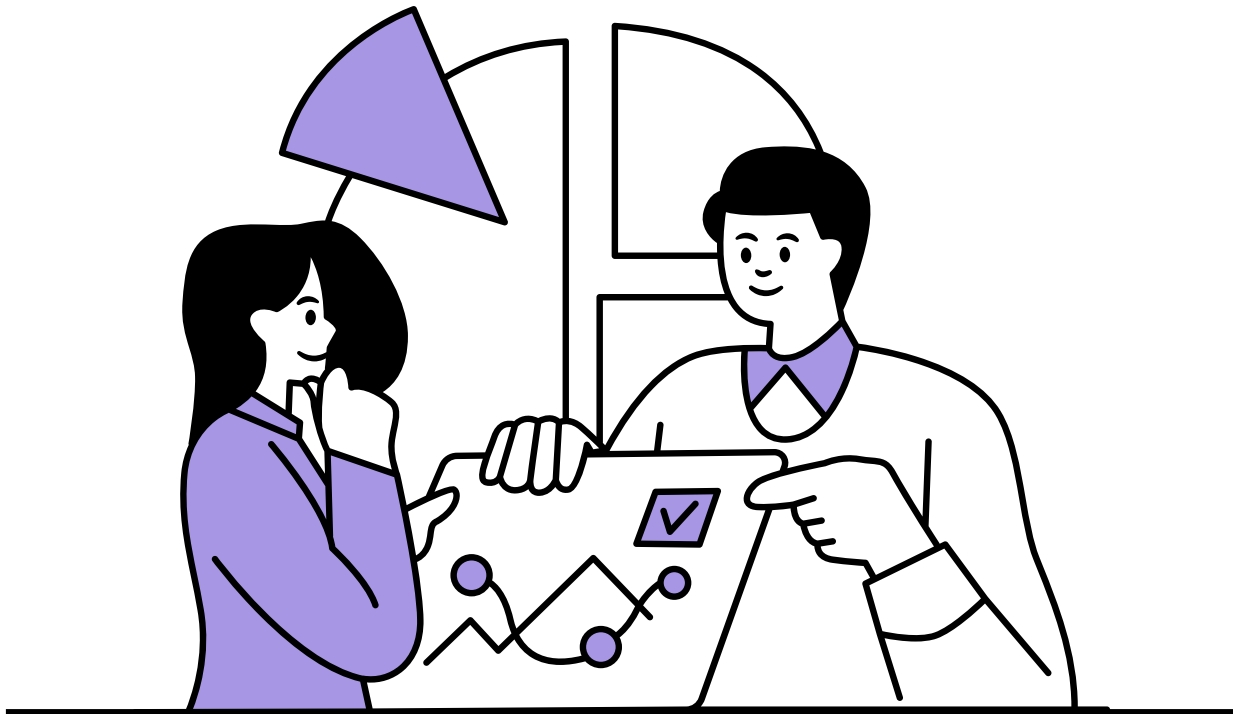
**손병구**

이미지 분류 모델 조사,  
하이퍼 파라미터 최적화 값 조사

**최유연**

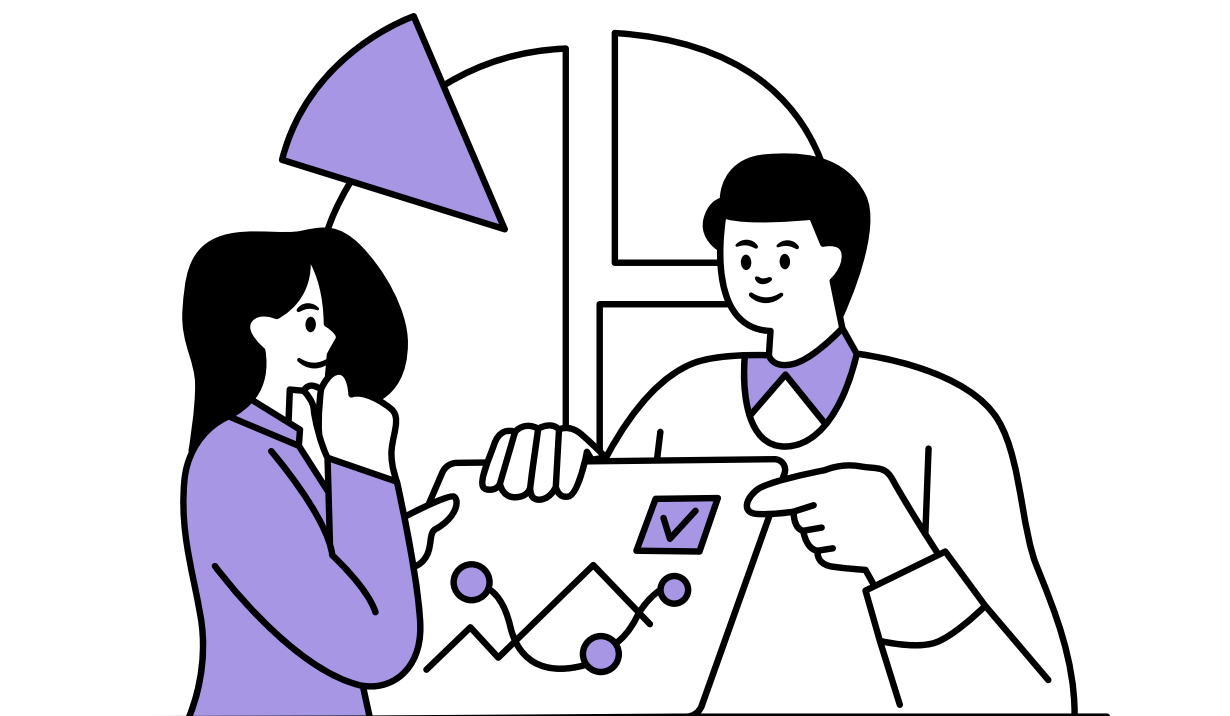
이미지 분류 모델 조사,  
Custom dataset 구현

# 06. Aug 및 Hyperparameter



```
HALF_PERCENT = 0.5
FULL_PERCENT = 1.0
train_aug = A.Compose([
    A.SmallestMaxSize(max_size= 224),
    A.RandomCrop(width= 200, height= 200),
    A.HorizontalFlip(p= HALF_PERCENT),
    A.ShiftScaleRotate(shift_limit= 0.05, scale_limit= 0.06,
                       rotate_limit=20, p= HALF_PERCENT),
    A.RGBShift(r_shift_limit=10, g_shift_limit=10, b_shift_limit=10, p = 1),
    ## 낮은 확률
    A.RandomBrightnessContrast(p= 0.2),
    A.VerticalFlip(p= 0.2),      # 뒤집어 있을 확률 낮음
    A.ToGray(p= 0.333),
    A.OneOf([ ## 날씨 상황
        A.RandomFog(fog_coef_lower= 0.3, fog_coef_upper= 0.8,
                    alpha_coef= 0.03, p= FULL_PERCENT),      # 안개
        A.RandomSunFlare(flare_roi= (0, 0, 0.05, 0.001),
                         angle_lower= 0.1, p= FULL_PERCENT),  # 눈썹
        A.RandomSnow(brightness_coeff= 2.5, snow_point_lower= 0.1,
                     snow_point_upper= 0.3, p= FULL_PERCENT), # 눈 (눈만 0.2)
        A.RandomRain(brightness_coefficient= 0.7, drop_width= 1,
                     blur_value= 3, p= FULL_PERCENT),          # 비
    ], p = HALF_PERCENT),
    A.Normalize(mean=(0.485, 0.456, 0.406), std= (0.229, 0.224, 0.225)),
    ToTensorV2()
])
```

## 06. 모델 훈련 결과



model name	optimizer	lr	batch size	epochs	train loss	train acc	val acc	test acc
swin_t	AdamW	0.0001	32	10	0.227	0.985	0.986	99.73%
	AdamW	0.0001	64	10	0.212	0.994	0.989	99.46%
	AdamW	0.0001	128	10	0.21	0.994	0.984	99.45%
	AdamW	0.001	128	10	0.457	0.849	0.886	86.72%

model name	optimizer	lr	batch size	epochs	train loss	train acc	val acc	test acc
vgg16	AdamW	0.0001	128	5	0.217	0.992	0.973	99.19%
	AdamW	0.0001	128	10	0.236	0.981	0.981	99.18%
	AdamW	0.001	128	10	0.533	0.777	0.829	84.28%

model name	optimizer	lr	batch size	epochs	train loss	train acc	val acc	test acc
resnet18	AdamW	0.0001	128	10	0.255	0.973	0.986	97.83%
	AdamW	0.001	128	10	0.358	0.903	0.919	94.57%

model name	optimizer	lr	batch size	epochs	train loss	train acc	val acc	test acc
ResNet50	AdamW	0.0001	64	10	0.242	0.977	1	99.46%
	AdamW	0.0001	32	10	0.24	0.98	0.995	99.46%
	AdamW	0.0001	128	10	0.236	0.981	0.986	99.19%

# 감사합니다

Team 02.

여익수, 권태윤, 이승윤, 손병구, 최유연



# Mask R-CNN

## - Process

1. Resize (800 ~ 1024 사이) → Padding (1024 x 1024)
2. ResNet-101을 통해 feature map (C1, C2, C3, C4, C5)를 생성
3. FPN을 통해 이전에 생성된 feature map에서 P2, P3, P4, P5, P6 feature map을 생성
4. 최종 생성된 feature map에 각각 RPN을 적용하여 classification, bbox regression output값을 도출
5. output으로 얻은 bbox regression값을 원래 이미지로 projection시켜서 anchor box를 생성
6. Non-max-suppression을 통해 생성된 anchor box 중 score가 가장 높은 anchor box를 제외하고 모두 삭제
7. 각각 크기가 서로다른 anchor box들을 RoI align을 통해 size 통일
8. Fast R-CNN에서의 classification, bbox regression branch와 더불어 mask branch에 anchor box값을 통과