

2024 DSC 리빙랩 지원사업 최종 발표

Presentation with **KURO**

과제책임자: 김 휘

팀장: 김준원

혁신가: 서성국

팀원: 박성배, 이영기, 송태용, 유영찬

2024.12.05

Contents

01 팀원 소개

02 사업 소개

03 활동 내용 소개

04 성과 소개

01 팀원 소개



김 휘

과제 책임자

고려대학교

전자 및 정보공학 교수

01 팀원 소개



김준원



박성배



서성국



이영기



송태용



유영찬

팀장

2017390316

기획 총괄
사업비 집행관리

팀원

2023270759

H/W 지원
대외 활동

혁신가

2021270830

H/W 지원
대외 활동

팀원

2021270836

S/W 개발
사업비 집행관리

팀원

2023270730

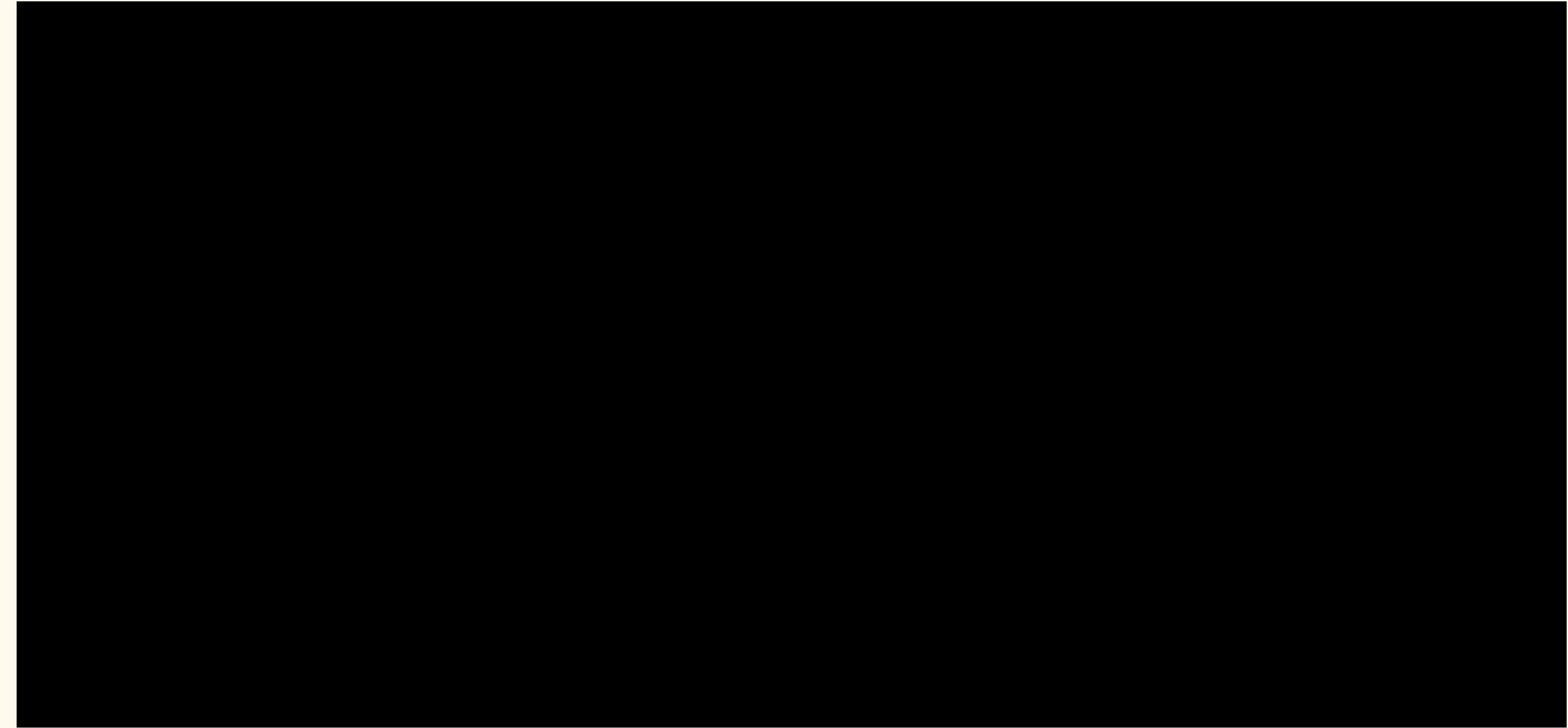
H/W PCB 설계
구매 자재 관리

팀원

2021270141

S/W
인공지능개발

02 사업 소개



02 사업 소개

사회 환경

모기 출현 두 달 빨라졌다...온난화 지속 땀 '덴기열' 토착화 우려

모기 첫 등장 시기 5월 말→3월 말로 당겨져

기자 정봉비

수정 2024-04-12 06:55 등록 2024-04-11 18:54



유인호 세종시의원 “예산만 주는 모기 방제사업, 관리 소홀해”

강대묵 기자 | 승인 2023년 06월 15일 09시 33분 온라인 2023년 06월 15일 목요일

요약 f t l d e m s

[충청투데이 강대묵 기자] 세종시의회 유인호 의원(더불어민주당·보람동)은 최근 진행된 행정복지위원회 보건소 소관 행정사무감사에서 “모기 방제사업의 관리가 소홀하다”며 효율적인 방제체계 마련을 촉구했다.

유 의원은 “보건소에서 관리하는 모기 방제사업에는 10억 원이라는 큰 예산이 지출되고 있다”며 “하지만 보건소는 이같이 예산을 각 읍·면·동에 배분만 해줄 뿐, 사후 효과 분석 및 데이터 관리 등 지휘 기관의 역할을 못 하고 있다”며 소홀함을 지적했다.

더불어 “주요 감염병 매개 모기 방제 지침에 따르면 유충의 경우, 1월부터 12월까지 연중 관리가 필요하고, 성충도 기후변화 등에 따라 시기를 앞당겨 방제할 필요가 있다”고 시기 문제를 언급했다.

또한 “현재 각 읍·면·동별로 진행되는 용역에서는 방제 시기 및 일자 등에 대한 자료확인도 어렵고, 실제 방제가 어떻게 효과가 있는지조차 확인할 수가 없다”며 “방제 후 7일 이내 재조사 및 추가 방제가 필요함에도 보건소는 해당 사항에 대해 실시한 바가 없으며, 방제 효과 분석 등에 대한 표본조사조차 이루어지지 않았다”고 지적했다.

“모기 발생 빈도 증가 및 전염병 매개 모기 개체수 증가”




유형	질병별 통계 - 연도별	감염병 급/명	제1급 외, 일본뇌염 외	구분	발생수	환자분류	전체
조회기간	2021년 ~ 2023년	시도/시군 구	전체, 전체	연령단위	-	감염지역	전체

1

구분		2021	2022	2023
제3급	일본뇌염	23	11	17
	말라리아	294	420	747
	뎅기열	3	103	206


03 활동 내용 소개



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

등록특허 10-2078264



(45) 공고일자

2020년02월18일

(11) 등록번호

10-2078264

(24) 등록일자

2020년02월11일

(51) 국제특허분류(Int. Cl.)

A01M 1/22 (2006.01)

A01M 1/02 (2006.01)

A01M 25/00 (2006.01)

B05C 1/00 (2006.01)

(52) CPC특허분류

A01M 1/222 (2013.01)

A01M 1/02 (2013.01)

(21) 출원번호

10-2018-0090410

(22) 출원일자

2018년08월02일

(23) 심사청구일자

2018년08월02일

(65) 공개번호

10-2020-0015088

(43) 공개일자

2020년02월12일

(56) 선행기술조사문헌

KR1020090056763 A*

KR1020110038890 A*

KR200481174 Y1*

*는 심사관에 의하여 인용된 문헌

전체 청구항 수 : 총 4 항

심사관 : 김상민

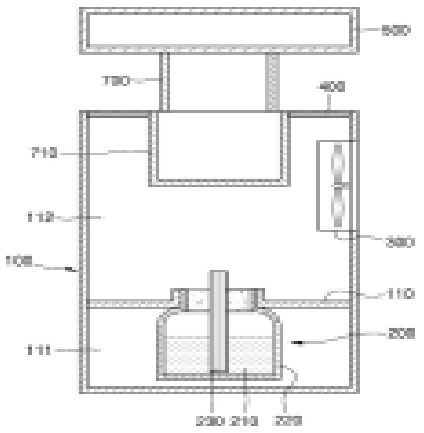
(54) 발명의 명칭

모기 퇴치기

(57) 요약

모기 퇴치기에 관해 개시되어 있다. 개시된 모기 퇴치기는 모기가 좋아하는 모기 유인력을 방출하고 전기 방출량을 이용하여 모기를 제거할 수 있는 것으로서, 열 에시에 따른 모기 퇴치기는 상부가 개방되도록 마련된 하우징, 상기 하우징의 하부에 배치되는 모기 유인제, 상기 하우징의 측면에 배치되어 상기 모기 유인제를 상기 하우징의 내부에 분산시키는 분산 팬 및 상기 하우징의 상부에 배치되어 소정의 전압을 구비하는 전류가 흐르는 전기 방출량:을 포함할 수 있다.

대표도 - 도3



모기 퇴치기 관련 특허 보유

모기 퇴치기 프로토타입 제작 완료

세종시와의 모기 방제 사업 관련 컨택

모기 검출기 관련 전문가 자문 예정

대학교 內 설치

03 활동 내용 소개

방역 솔루션

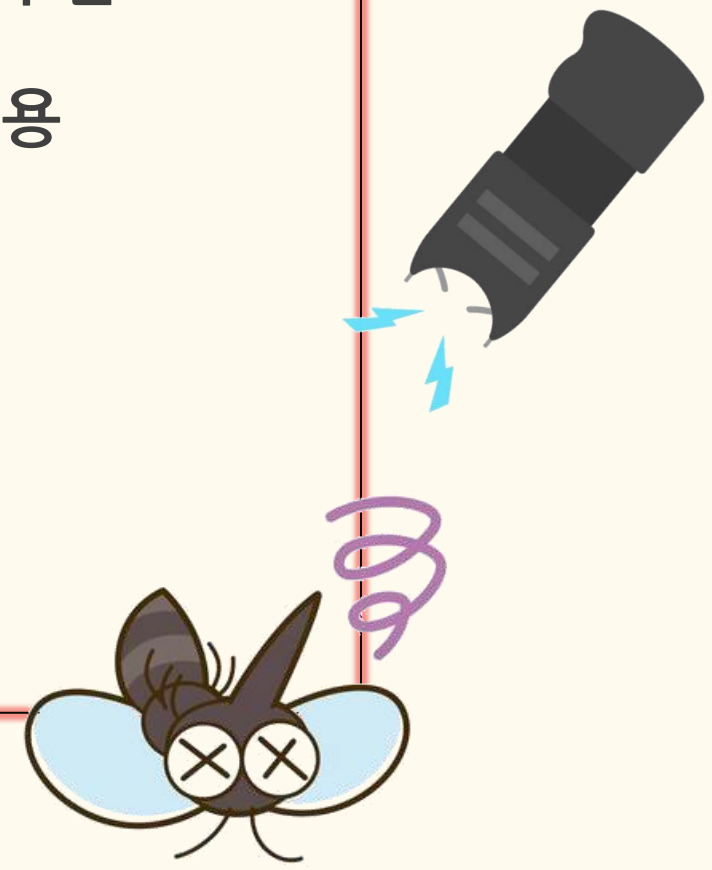
검출

- ✓ 축사 및 공원 등의 모기 밀집 지역에 설치
- ✓ 실시간 모니터링 시스템 구축을 통한 2차 방역
- ✓ 퇴치기와 분리, AI 검출 후 2차 분석



퇴치

- ✓ 학교, 공공시설 등의 모기 출현 장소에 설치
- ✓ 방역 모니터링보다 쾌적함 제공이 우선
- ✓ 이산화탄소와 365nm UV LED 병용
- ✓ 모기 살상용 전기 그물망 도입



03 활동 내용 소개

방역 솔루션

◦ 모기퇴치기, 모기감지기 설치 및 운영

-> 운영 효율성: 주 1~2회 정도 설치 구역 Patrol 실시 하여 상황 전파 및 데이터 수집 및 유지보수



고려대학교 교내 영화제 시제품 테스트

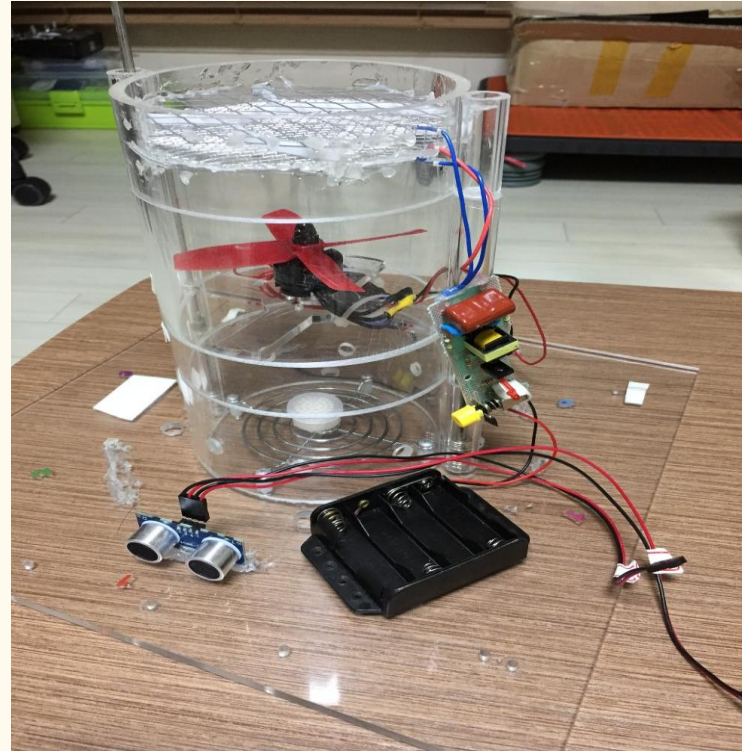


고려대학교 야자수 컨셉 시제품 테스트

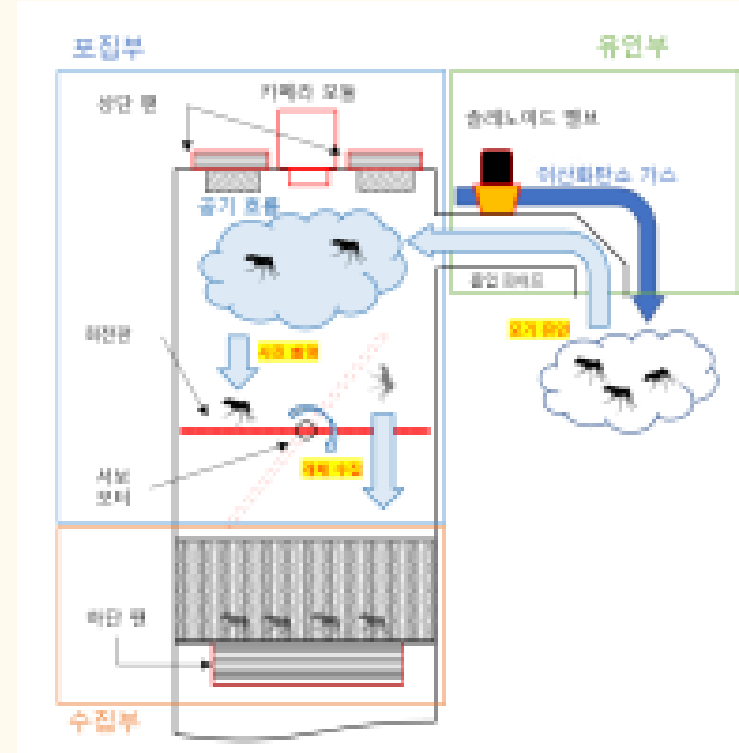


고려대학교 교내 야자수 컨셉 시제품 테스트

04 성과 소개



(모기 퇴치기 1차 시제품)



(모기 감지기 플로우)

2차 모기퇴치기 3D 모델링 및 실물



<모기 퇴치기 운영 방식>

1. 모기를 팬으로 흡입
2. 카메라 모듈로 촬영
3. 촬영사진 첫 번째 라즈베리파이와 AI키트로 처리.
4. 서보모터를 이용해 가림막 개방
5. 모기는 킬링존으로 이동되어 사살 및 제거.
6. 두 번째 라즈베리파이가 제어
7. 배터리, BMS, 충전기, 커스텀 PCB가 보조.

04 성과 소개

<지자체 협력기관 세종시 질병관리청 매개체분석과 내방 자문 회의>

자문 회의 내용 Q&A

Q. 국내 모기 출현시기가 여름에서 가을로 방대해졌을 때의 차이?
A. 큰 차이는 없음

Q. 채집기 설치 시기와 포집량은 어땠나요?
A. 모기 종류가 바뀌진 않음 (흰줄숲모기 0.05 ↓ 최근 4~6% ↑)
(낮에 활동하는 모기) 과거엔 사용 안해서 수치상으로
설명이 어렵고, 고정적이지 않다

Q. 국내 여러 종이 있는 걸로 알고 있는데, 새로운 종이 발견되었나요?
A. 1종 원충은 검출 되지 않았다. 하지만 7종은 몸속에서 검출 된 이력이 있다.

Q. 뉴스에 자주 뜨는데 왜 그런가요?
A. 말라리아 관련 보도 (모니터링)을 많이 하기 때문에 홍보용으로 뉴스기사
활용함



04 성과 소개

그림 1-1. AI-DMS 작동원리



그림 2. AI-DMS 현장설치 모습

그림 3. AI-DMS 모기 포집 및 촬영

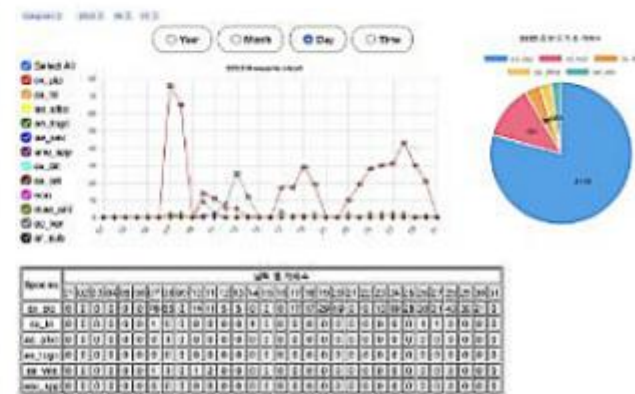
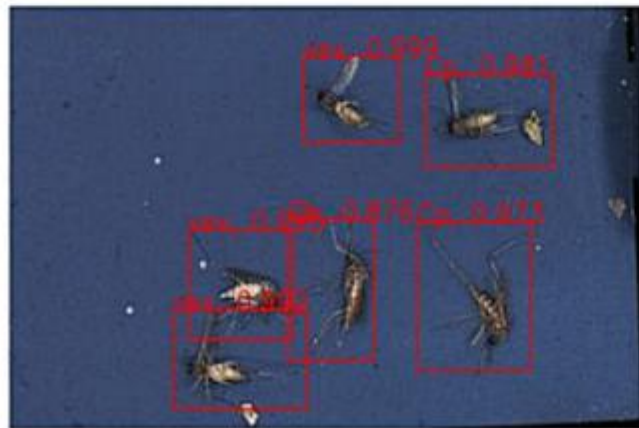


그림 4. 촬영된 모기 영상에 대한 모기 종·동정

그림 5. 홈페이지를 통해 모기 발생수 및 종 확인

Q. 인공지능 vs 기존 검출방식- 장단점설명 부탁드립니다.

A. 인공지능: 자동 분류가 잘되고,일정함 유지, 자유롭다

기존 검출방식: 인력 (고도기술 필요)노동력, 분류, 정확도, 공무원 (순환보직) 오차 ↑

Q. 야외테스트 학습데이터 정확도는 어떻게 되나요? 그리고 살상 후 어떻게 하시는지?

A. 정확도 95% → 살상 → 학습데이터 수집 → 폐기처리

머리 몸통 날개 다리 보존/ 살상에 유리 (전압 미세조절 전기파리채 보다 낮게)

Q. 모기 유인 (옥테놀) 줄어든 이유:

A. 유인률 ↓ co2 비교시 많이 떨어짐 → 10m이내 유인

365nm 파장 EUV: 5m 이내 유인 500 파장 기피함.

Q. 추가로 더 도와줄 수 있는 부분은 없나요?

A. 모기 공문시 (사육장) 모기 받을 수 있음

→ 모기 관련 기업체 연결 가능

→ 메일/유선상으로 기타 질의응답 대응

04 성과 소개 (정량 지표)

프로그램	정량지표	목표치	산정기준
목업 제작	퇴치기,감지기 제작 대수	각 5대	시제품 개수
고려대 內 설치 테스트	고려대 교내 야외 설치	5곳	모기 출몰 지역
모기 인식 S/W 개발	모기 인식 이미지 기반 S/W 개발 건수	50%	S/W 저작물
퇴치기 배터리 전환	태양광 충전량 대비 Cap축전 배터리 용량	12hr	용량수치

04 성과 소개 (정량 지표)

목업 제작	퇴치기,감지기 제작 대수	각 5대	시제품 개수
-------	---------------	------	--------

송태용

모기퇴치기 설치 사례



교내 CELL 영화제



석원 경상관



학생 회관



석원 경상관 외부 벤치

04 성과 소개 (정량 지표)

모기 인식 S/W 개발	모기 인식 이미지 기반 S/W 개발 전수	1건 달성	S/W 저작물
--------------	------------------------	-------	---------

PROBLEMS

OUTPUT

DEBUG/CONSOLE

TERMINAL

PORTS

powerlshet + v [icon] -- v x

With torch.cuda.amp.autocast(amp):
0/99 3.566 0.1263 0.0268 0 48 340: 41%
..) is deprecated. Please use 'torch.amp.autocast('cuda', args...)' instead.
With torch.cuda.amp.autocast(amp):
0/99 3.566 0.1261 0.0267 0 50 340: 42%
..) is deprecated. Please use 'torch.amp.autocast('cuda', args...)' instead.
With torch.cuda.amp.autocast(amp):
0/99 3.566 0.1259 0.02657 0 70 340: 44%
..) is deprecated. Please use 'torch.amp.autocast('cuda', args...)' instead.
With torch.cuda.amp.autocast(amp):
0/99 3.566 0.1256 0.02648 0 47 340: 45%
..) is deprecated. Please use 'torch.amp.autocast('cuda', args...)' instead.
With torch.cuda.amp.autocast(amp):
0/99 3.566 0.1255 0.02642 0 71 340: 47%
..) is deprecated. Please use 'torch.amp.autocast('cuda', args...)' instead.
With torch.cuda.amp.autocast(amp):
0/99 3.566 0.1252 0.02625 0 51 340: 48%
..) is deprecated. Please use 'torch.amp.autocast('cuda', args...)' instead.
With torch.cuda.amp.autocast(amp):
0/99 3.566 0.1251 0.0261 0 64 340: 50%
..) is deprecated. Please use 'torch.amp.autocast('cuda', args...)' instead.
With torch.cuda.amp.autocast(amp):
0/99 3.566 0.1248 0.02608 0 46 340: 52%
..) is deprecated. Please use 'torch.amp.autocast('cuda', args...)' instead.
With torch.cuda.amp.autocast(amp):
0/99 3.566 0.1245 0.02597 0 51 340: 53%
..) is deprecated. Please use 'torch.amp.autocast('cuda', args...)' instead.
With torch.cuda.amp.autocast(amp):
0/99 3.566 0.1243 0.02598 0 69 340: 55%
..) is deprecated. Please use 'torch.amp.autocast('cuda', args...)' instead.
With torch.cuda.amp.autocast(amp):
0/99 3.566 0.124 0.02575 0 30 340: 56%
..) is deprecated. Please use 'torch.amp.autocast('cuda', args...)' instead.
With torch.cuda.amp.autocast(amp):
0/99 3.566 0.1239 0.0256 0 50 340: 58%
..) is deprecated. Please use 'torch.amp.autocast('cuda', args...)' instead.
With torch.cuda.amp.autocast(amp):
0/99 3.566 0.1236 0.02553 0 49 340: 59%
..) is deprecated. Please use 'torch.amp.autocast('cuda', args...)' instead.
With torch.cuda.amp.autocast(amp):
0/99 3.566 0.1235 0.02539 0 52 340: 61%
..) is deprecated. Please use 'torch.amp.autocast('cuda', args...)' instead.
With torch.cuda.amp.autocast(amp):
0/99 3.566 0.1232 0.02530 0 65 340: 62%
..) is deprecated. Please use 'torch.amp.autocast('cuda', args...)' instead.
With torch.cuda.amp.autocast(amp):
0/99 3.566 0.123 0.02523 0 48 340: 64%
..) is deprecated. Please use 'torch.amp.autocast('cuda', args...)' instead.
With torch.cuda.amp.autocast(amp):
0/99 3.566 0.1227 0.02512 0 35 340: 65%
..) is deprecated. Please use 'torch.amp.autocast('cuda', args...)' instead.
With torch.cuda.amp.autocast(amp):
0/99 3.566 0.1224 0.02515 0 60 340: 67%
..) is deprecated. Please use 'torch.amp.autocast('cuda', args...)' instead.
With torch.cuda.amp.autocast(amp):

DDP mode
if cuda and RANK != -1:
| model = smart_DDP(model)

Model attributes
nl = de_parallel(model).model[-1].nl # number of detection layers (to scale hyps)
hyp["box"] *= 3 / nl # scale to layers
hyp["cls"] *= nc / 80 * 3 / nl # scale to classes and layers
hyp["obj"] *= (imgsz / 640) ** 2 * 3 / nl # scale to image size and layers
hyp["label_smoothing"] = opt.label_smoothing
model.nc = nc # attach number of classes to model
model.hyp = hyp # attach hyperparameters to model
model.class_weights = labels_to_class_weights(dataset.labels, nc).to(device) * nc # attach class weights
model.names = names

Start training
t0 = time.time()
nb = len(train_loader) # number of batches
nw = max(round(hyp["warmup_epochs"] * nb), 100) # number of warmup iterations, max(3 epochs, 100 iterations)
nw = min(nw, (epochs - start_epoch) / 2 * nb) # limit warmup to < 1/2 of training
last_opt_step = -1
maps = np.zeros(nc) # mAP per class
results = (0, 0, 0, 0, 0, 0, 0) # P, R, mAP@.5, mAP@.5-.95, val_loss(box, obj, cls)
scheduler.last_epoch = start_epoch - 1 # do not move
scaler = torch.cuda.amp.GradScaler(enabled=amp)
stopper, stop = EarlyStopping(patience=opt.patience), False
compute_loss = ComputeLoss(model) # init loss class
callbacks.run("on_train_start")
LOGGER.info(
f'Image sizes {imgsz} train, {imgsz} val\n'
f'Using {train_loader.num_workers * WORLD_SIZE} dataloader workers\n'
f'logging results to {colorstr("bold", save_dir)}\n'
f'Starting training for {epochs} epochs...'
)
for epoch in range(start_epoch, epochs): # epoch -----
| callbacks.run("on_train_epoch_start")
| model.train()

Update image weights (optional, single-GPU only)
if opt.image_weights:
| cw = model.class_weights.cpu().numpy() * (1 - maps) ** 2 / nc # class weights
| iw = labels_to_image_weights(dataset.labels, nc=nc, class_weights=cw) # image weights
| dataset.indices = random.choices(range(dataset.n), weights=iw, k=dataset.n) # rand weighted idx

opt=opt,
hyp=hyp,
logger=LOGGER,
include=tuple(include_loggers),
)

Register actions
for k in methods(loggers):
| callbacks.register_action(k, callback=getattr(loggers, k))

Process custom dataset artifact link
data_dict = loggers.remote_dataset
if resume: # If resuming runs from remote artifact
| weights, epochs, hyp, batch_size = opt.weights, opt.epochs, opt.hyp, opt.batch_size

Config
plots = not evolve and not opt.noplots # create plots
cuda = device.type != "cpu"
init_seeds(opt.seed + 1 + RANK, deterministic=True)
with torch.distributed_zero_first(LOCAL_RANK):
| data_dict = data_dict or check_dataset(data) # check if None
train_path, val_path = data_dict["train"], data_dict["val"]
nc = 1 if single_cls else int(data_dict["nc"]) # number of classes
names = {0: "item"} if single_cls and len(data_dict["names"]) != 1 else data_dict["names"] # class names
is_coco = isinstance(val_path, str) and val_path.endswith("coco/val2017.txt") # COCO dataset

Model
check_suffix(weights, ".pt") # check weights
pretrained = weights.endswith(".pt")
if pretrained:
| with torch.distributed_zero_first(LOCAL_RANK):
| | weights = attempt_download(weights) # download if not found locally
| ckpt = torch.load(weights, map_location="cpu") # load checkpoint to CPU to avoid CUDA memory leak
| model = Model(cfg or ckpt["model"].yaml, ch=3, nc=nc, anchors=hyp.get("anchors")).to(device) # create
| exclude = ["anchor"] if (cfg or hyp.get("anchors")) and not resume else [] # exclude keys
| csd = ckpt["model"].float().state_dict() # checkpoint state_dict as FP32
| csd = intersect_dicts(csd, model.state_dict(), exclude=exclude) # intersect
| model.load_state_dict(csd, strict=False) # load
| LOGGER.info(f"Transferred {len(csd)}/{len(model.state_dict())} items from {weights}") # report
else:
| model = Model(cfg, ch=3, nc=nc, anchors=hyp.get("anchors")).to(device) # create
amp = check_amp(model) # check AMP

Freeze
freeze = [f"model.{x}." for x in (freeze if len(freeze) > 1 else range(freeze[0]))] # layers to freeze
for k, v in model.named_parameters():
| v.requires_grad = True # train all layers
| # v.register hook(lambda x: torch.nan to num(x)) # NaN to 0 (commented for erratic training results)

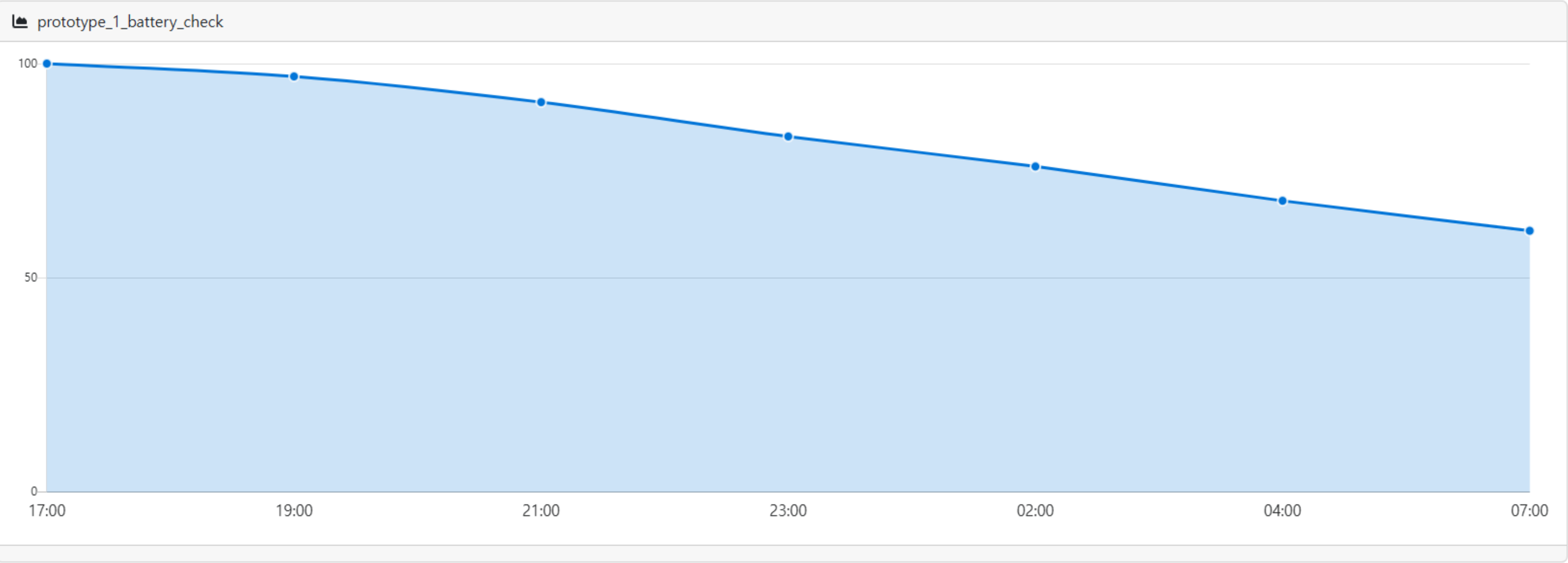
학습 코드

학습 진행중

가중치확인함수

04 성과 소개 (정량 지표)

퇴치기 배터리 전환	태양광 충전량 대비 Cap축전 배터리 용량	12hr 달성	용량수치
------------	----------------------------	---------	------



04 성과 소개 (정성 지표)

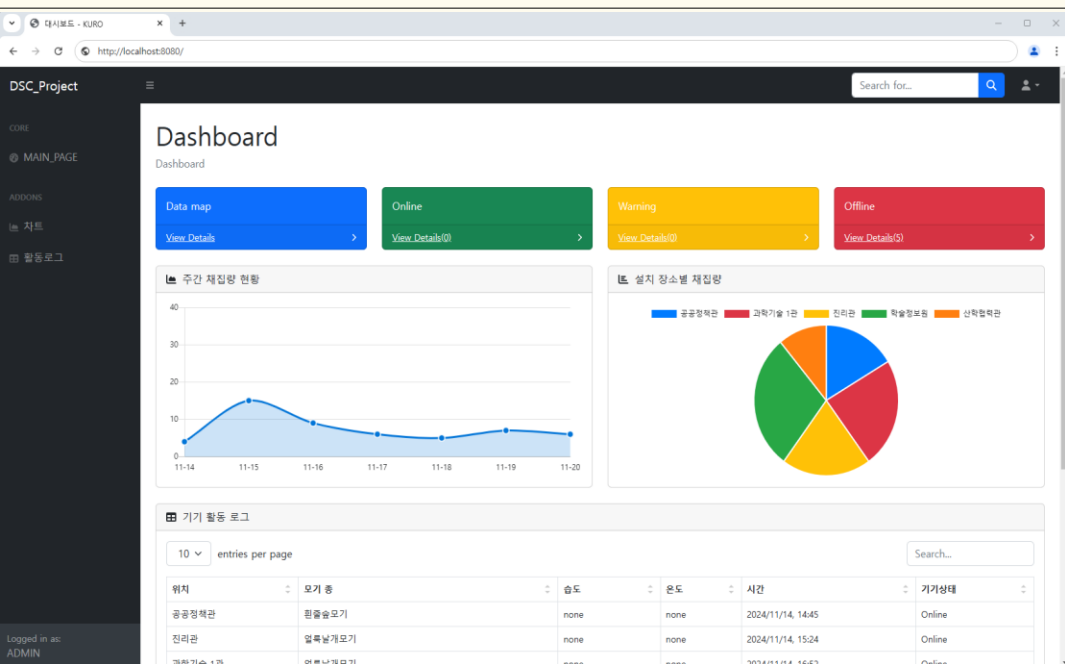
프로그램	정성지표	목표내용
감지기 정확도 향상	이미지 기반 인식 정확도 수치	70%
분석 및 시각화	교내 모기 밀집도 지도 업데이트	고려대 모기 데이터 지도화
지역 사회 협력	지역 문제 해결을 위한 지자체 협력	2개 이상 기업 및 기관 협력
기술 사업화	포집 및 데이터 지도 기술 문서 작성	기술 문서 작성

04 성과 소개 (정성 지표)

감지기 정확도 향상	이미지 기반 인식 정확도 수치	82% 달성
		
작은 빨간집 모기	흰줄 숲 모기	중국 얼룩 날개 모기
지역 사회 협력	지역 문제 해결을 위한 지자체 협력	3개 이상 기업 및 기관 협력
	(주)세이프티솔라텍 기업정보 교통 신호장치 제조업	 고려대학교 세종캠퍼스 세종산학협력단 KOREA UNIVERSITY SEJONG CAMPUS

04 성과 소개 (정성 지표)

분석 및 시각화



교내 모기 밀집도 지도 업데이트

차트

대시보드 / 차트

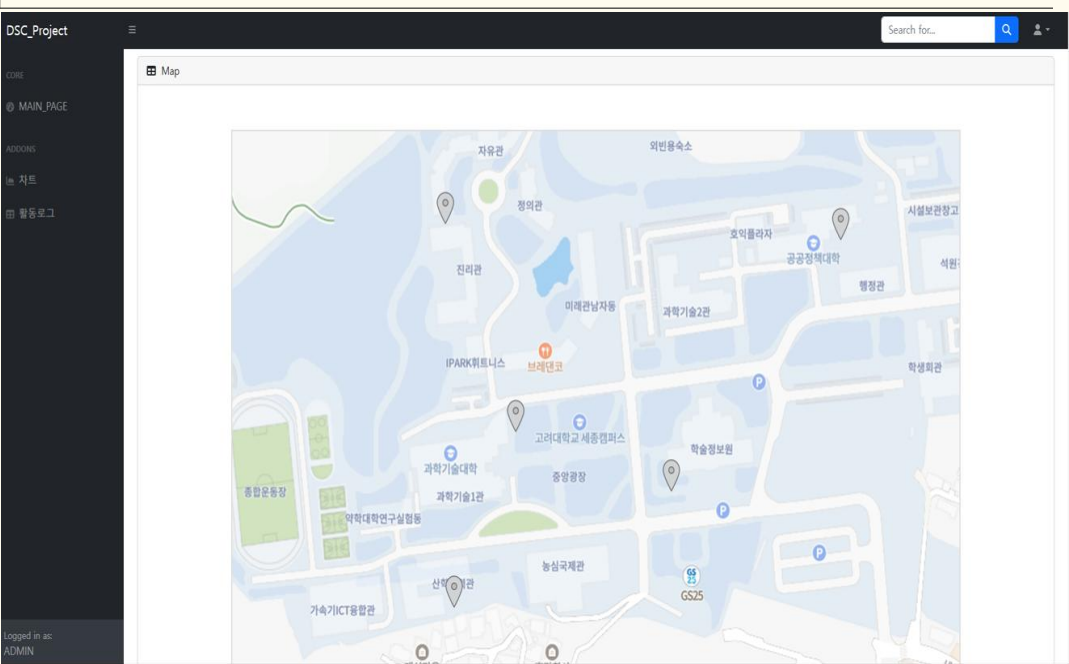
기기의 포집 기록과 모기 포집량을 나타내는 차트입니다. 기기의 시간과 위치별 포집기록을 출력합니다.

주간 채집량 현황



Updated 2024-11-27, 18:00 PM

고려대 모기 데이터 지도화



YOLOv5

1. One-Stage Detector 방식의 빠른 객체 탐지기능과 라즈베리파이 AI카메라의 AI가속기를 활용.
2. 포집기 내부에 포집된 모기를 빠르게 인식하여 데이터 베이스에 전송함.
3. 데이터 베이스는 PostgreSQL을 활용하여, 웹 서버에 저장됨.
4. 부트스트랩 프레임 워크를 활용하여 웹사이트 대시보드를 제작함.
5. 대시보드에는 설치 구역별, 시간별, 주간 채집량이 출력됨.
6. 실시간 연결 현황과 설치 상태 등을 확인 가능함.

04 성과 소개 (정성 지표)

기술 사업화				포집 및 데이터 지도 기술 문서 작성				기술 문서 작성					
DSC 지역 <u>리빙랩</u> 지원사업 개발 기술서													
프로젝트	태양광 기반 모기 퇴치 솔루션	개발자	KURO										
SDK/TOOL	<u>VScode</u> / <u>YOLOv5</u> , <u>Raspberry PI</u> , <u>Bootstrap</u> , <u>PostgreSQL</u>												
개발 기간	2024.08.20 ~2024.12.01												
개발 목적	태양광 기반 친환경 방역체계 구축 및 방역 빅데이터 구축												
주요 기능	모기 객체 탐지, 객체 정보 주간/구역/시간별 분류 및 조회, 장비 연결 현황 확인, 실시간 데이터 지도 조회												
상세기술	모기 객체 탐지 1-1. <u>Raspberry PI</u> 의 AI카메라가 포집 된 모기의 사진을 촬영함. 1-2. 촬영된 모기는 내장된 <u>YOLOv5</u> 의 학습모델이 객체를 인식함. 1-2-1. 인식된 객체 정보는 웹 서버로 전송되어 데이터베이스에 저장됨 객체 정보 주간/구역/시간별 분류 및 조회 2-1. 데이터베이스에 저장된 객체 정보는 위치, 시간, 구역별 정보를 같이 저장함. 2-2. 저장된 정보는 <u>JavaScript</u> 의 함수를 활용하여 차트를 출력함. 오라클 <u>vpn</u> 을 활용하여 실시간 연결 상태를 데이터베이스에 저장함. 3-1 저장된 정보를 테이블 형태로 실시간 출력. 지도 <u>api</u> 를 활용, 웹 페이지 상에 지도를 출력 4-1 <u>Raspberry PI</u> 의 위치 정보를 지도 <u>api</u> 에 출력 4-2 데이터베이스에 저장된 실시간 포집 정보를 지도 <u>api</u> 에 출력												

기술 문서화 작업 80% 달성.

개발 기술서 작성 완료

세종시 방역 시스템에 적용하기
위한

맞춤화 연구는 아직 착수되지
못한 상태

감사합니다.
