

AI·빅데이터 기술을 이용한 가로수 조경관리 플랫폼 '나무지킴e'



Team 나무

김시은, 오준학, 이서현, 정재은, 조예진



Part 1.

필요성 및 배경





CO₂ 흡수 및 O₂ 배출

도시 그늘 제공

**가로수
기능 및
현황**



획일적인 수종 식재

과도한 가지치기



환경부 가이드라인과의 충돌

1 2023년 3월, 환경부와 국립생물자원관은
'도시 내 녹지관리 개선방안' 발표

2 생물다양성 증진을 위한 원칙, 식재 식물 선정 절차 개선, 가지치기 개선 등이 내용

3 그러나 현재 많은 지자체가 **가이드라인 미준수**

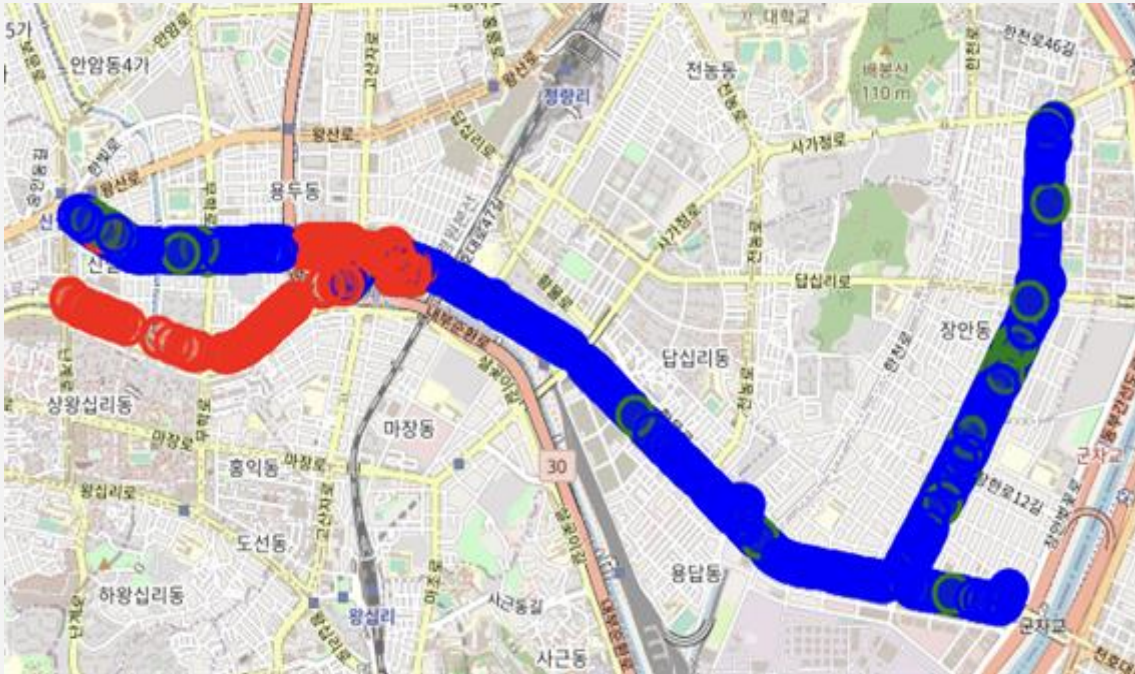


출처 : storyset on freepik

Part 2. 디지털 기술을 기반으로 한
사회 현안 해결책 제안 
- 동대문구를 중심으로 -



데이터에 기반한 가로수 수종 추천 알고리즘



- 대로(폭 25~40미터)는 자동차 통행량이 많아 이산화탄소가 다량 배출됨
- 현재, 동대문구의 대로는 느티나무(빨간색), 양버즘 나무(파란색)가 가로수의 대부분

〈표 9〉 최종 선정 수종 및 수종선정의 원칙

CO ₂ 흡수능력우수 수종	튤립나무, 메타세쿼이아, 양버즘나무, 은행나무, 느티나무, 회화나무, 칠엽수, 참나무류(상수리나무)
수종선정 원칙	<ul style="list-style-type: none"> • 수목은 아름다운 경관측면보다는 이산화탄소 저감기능에 중점 • 조경용으로 사용되는 수종보다는 탄소흡수능이 우수한 자생수종 도입 • 침엽수종보다는 낙엽활엽수종을 다양하게 활용 • 단기 및 장기적으로 탄소 흡수기능을 높일 수 있는 수종조합을 고려

〈표 10〉 탄소흡수 효율이 높은 주요수종과 환경적용성 비교

수종	CO ₂ 흡수량 (kgCO ₂ /tree/y)	생장속도***	도로 적용성			
			내공해성	내음성	내한성	내건성
느티나무*	38.8	+	중	중	강	중
은행나무*	39.7	+	강	약	강	강
메타세쿼이아*	35.5	+++	중	약	약	약
양버즘나무*	54.1	+++	강	중	강	중
회화나무*	67.8	++	강	중	강	중
튤립나무*	99.1	+++	강	중	강	중
칠엽수**	54.0	++	중	중	강	중
상수리나무**	51.0	+	중	중	강	중

* 박은진과 강규이(2010). 경기도 도시가로수의 탄소저장량과 연간 이산화탄소 흡수량산정(15~24년생 기준)
 ** 산림청홈페이지(<http://forestlove.forest.go.kr>), 탄소나무스쿨 탄소계정(흉고직경 20cm 기준)산출치 적용
 *** 생장속도: + 느림, ++ 보통, +++ 빠름

- 튤립나무의 이산화탄소 흡수율(99.1%)은 느티나무(38.8%)와 양버즘나무(54.1%)에 비해 높음
- 따라서 대로에 기존 가로수 대비 **이산화탄소 흡수율이 높은 수종을 추천**하는 알고리즘을 개발



데이터에 기반한 가로수 수종 추천 알고리즘



<동대문구 도로별 가로수 분포도>

<https://wcuznrk8rhub0o8gtsa2na.on.driv.tw/tree/%EB%8C%80%EB%A1%9C1.html>

<https://wcuznrk8rhub0o8gtsa2na.on.driv.tw/tree/%EC%A4%91%EB%A1%9C1.html>

<https://wcuznrk8rhub0o8gtsa2na.on.driv.tw/tree/%EC%86%8C%EB%A1%9C1.html>



<서울시 자동차 밀집도 데이터를 활용한 단계구분도>

<https://wcuznrk8rhub0o8gtsa2na.on.driv.tw/tree/map.html>



Vision Transformer 모델을 이용한 가로수 건강분류기 개발

정상적인 가로수 사진, 가지치기
되어있는 가로수 사진 **200개를 2개
클래스로** 분류
(class0 : 건강 / class1 : 불량)

학습데이터, 검증데이터, **평가데이
터를 70%/15%/15%**로 분류
(train : 140, val : 30, test : 30)

네이버 view를 통해 수집된 정상적
인 가로수 5장(g1 - g5), 가지치기 된
가로수 5장(b1 - b5)을 **딥러닝 분류
기에 적용**

딥러닝 모델이 10장 모두 올바르게
분류함





나무지킴e 플랫폼

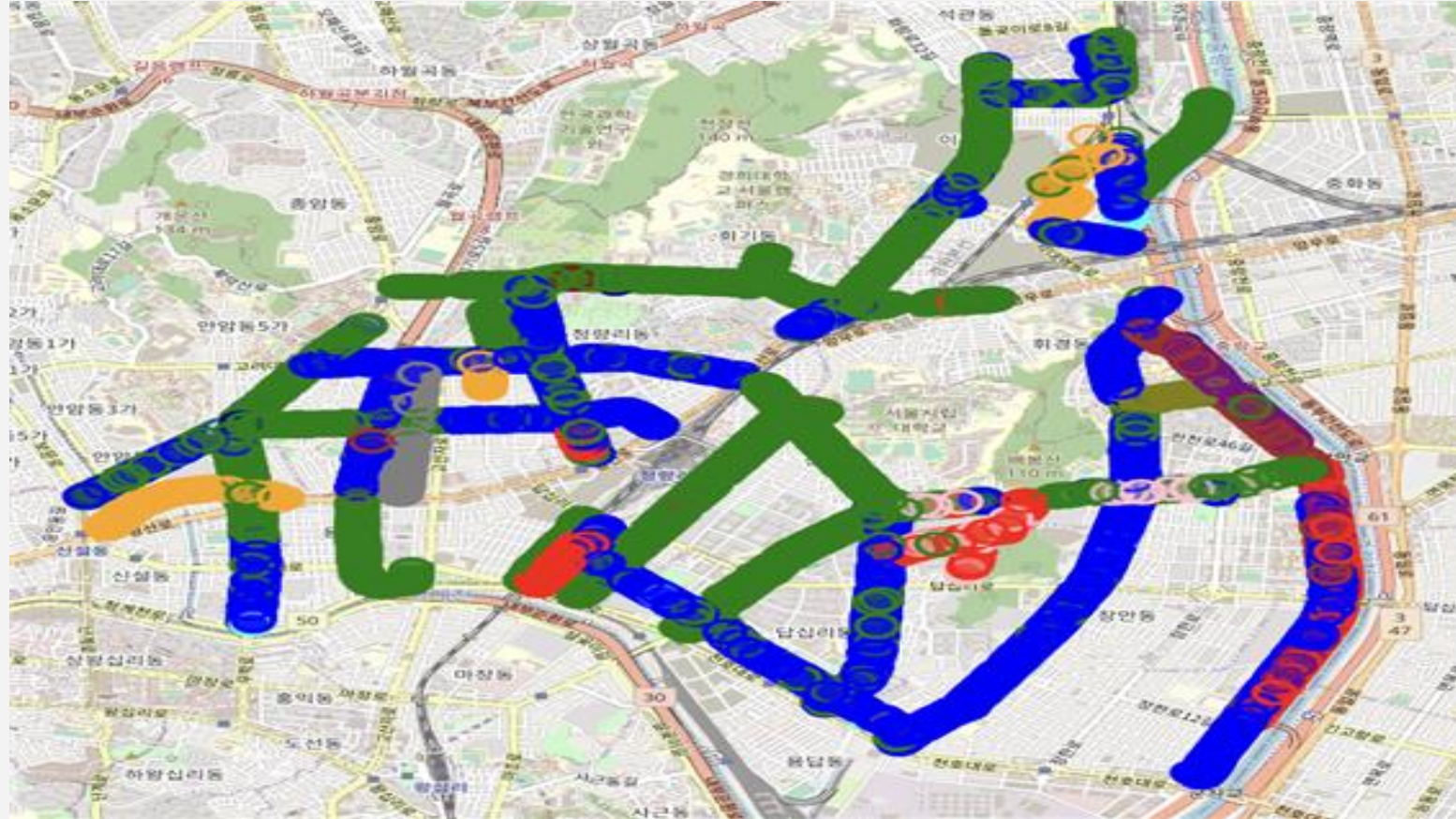
현실성, 효과성 파악을 위해 실제 앱 프로토타입 구현

- 가로수 현황 조회 및 지도 기반 수종 추천
- 가지치기 가로수, 불량 가로수 제보
- 가로수 봉사 참여 신청
- K-PaaS 이용하여 개발자와 공무원의 소통 활성화





나무지킴e 플랫폼



<동대문구 예시>

https://wcuznrk8rhub0o8gtsa2na.on.driv.tw/tree/fohium_%EA%B0%80%EB%A1%9C%EC%88%981.html

https://wcuznrk8rhub0o8gtsa2na.on.driv.tw/tree/fohium_%EA%B0%80%EB%A1%9C%EC%88%982.html

추진 필요성, 확산가능성
Part 3. 파약을 위해 구체적인
예상 시나리오 구현 



예상 시나리오 1

-'닭발 가지치기' 목격한 시민A씨

- ① 사진 촬영 후 '나무지킴e(environment)' 플랫폼에 신고 진행
- ② 플랫폼에 탑재된 <Vision transformer> 기술을 활용한 나무 건강 상태 분류기 모델로 신고 내용의 정합성 판단
- ③ 정합성 판단 절차에 의해 훼손 나무로 판단 시, 지방자치단체 도시건설국 공무원에게 해당 민원이 전달
- ④ 민원 처리 과정을 실시간으로 서비스 향유자인 시민들이 조회할 수 있도록 정보 공개
- ⑤ K-PaaS 이용하여 제보 데이터를 관리 및 활용해 해당 민원을 다음 분기 가지치기 계획에 반영



출처 : storyset on freepik



예상 시나리오 2

-가로수 봉사참여 희망자 B씨

- ① 평소 가로수 건강 및 환경문제에 관심이 많은 B씨는 '나무 지킴e(environment)' 통해 가로수 봉사활동을 신청
- ② 가로수 지도를 활용해 주기적으로 **가로수 건강 상태 점검 봉사**에 참여하여 데이터(가로수 종류, 위치, 높이, 흉고, 건강상태)의 유용성을 제고하고 실시간 업데이트
- ③ 해당 데이터를 관련 **부처와 공공 대중에 공개하여 가로수 조성 연구에 활용함으로써 지속가능한 기후 대응 기능 신장**



출처 : storyset on freepik



예상 시나리오 3

-도시관리국 공무원 C씨

- ① 가로수 식재 계획 수립 전, '나무 지킴e(environment)' 플랫폼을 통해 가로수 정보를 수집 및 조회
- ② 전체 가로수 수종 분포 인터랙티브 지도, AI 빅데이터 기술을 통합적으로 활용하여 가로수 당 이산화탄소 저감 수치, 동별 자동차 통행량, 도로폭에 따른 수종 추천 목록 확보
(예시) 대로(천호대로, 청계천로)에는 현재 느티나무와 양버즘 나무가 약 70%로 대부분을 차지하는데, 온실가스 흡수량이 많은 회화나무(CO2 흡수량 67.8%), 튼린 나무(CO2 흡수량 99.1%) 등을 혼합하여 식재
- ③ 식재 계획 공고 및 실행



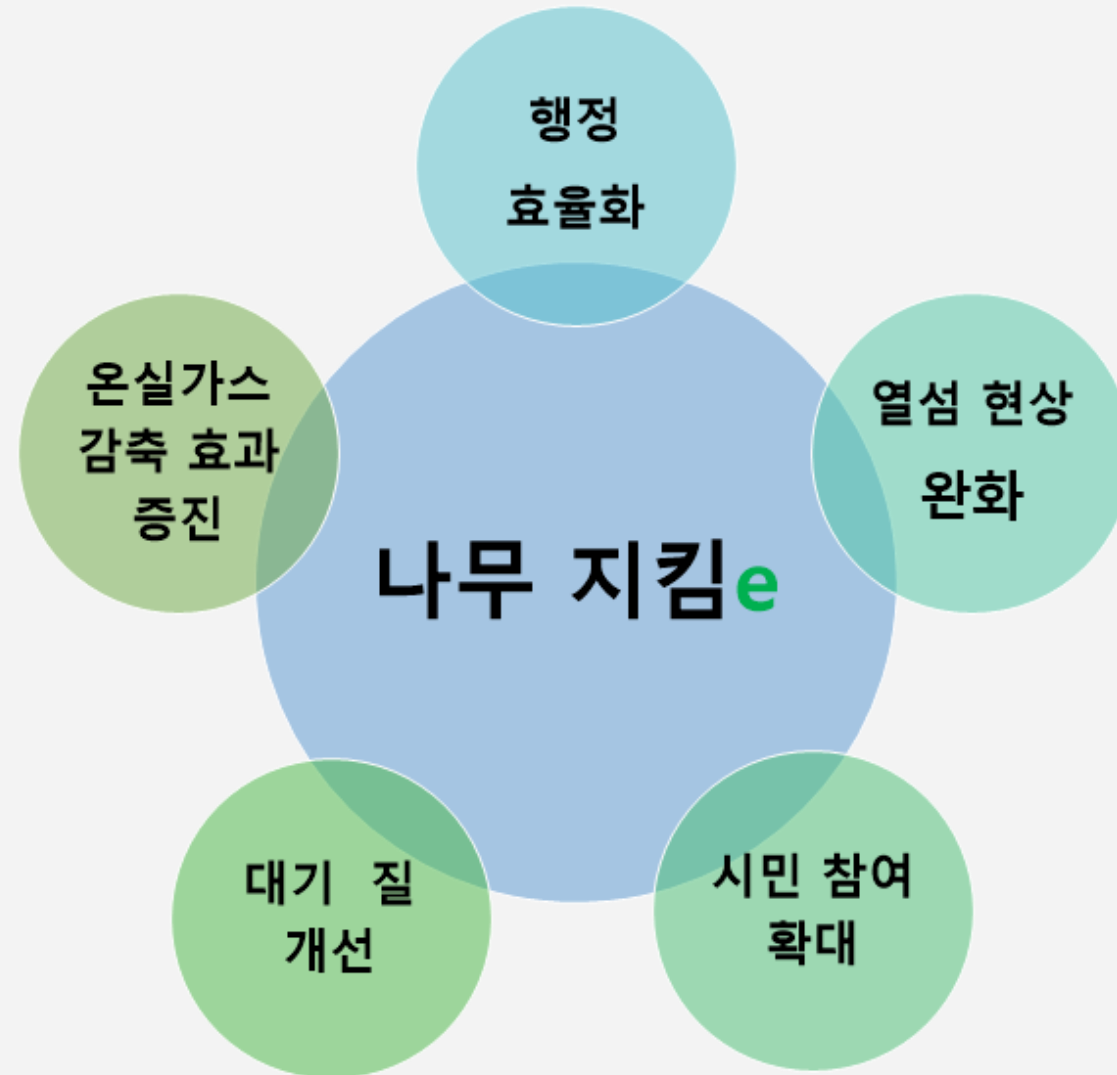
출처 : storyset on freepik

Part 4. 차별점 및 기대효과

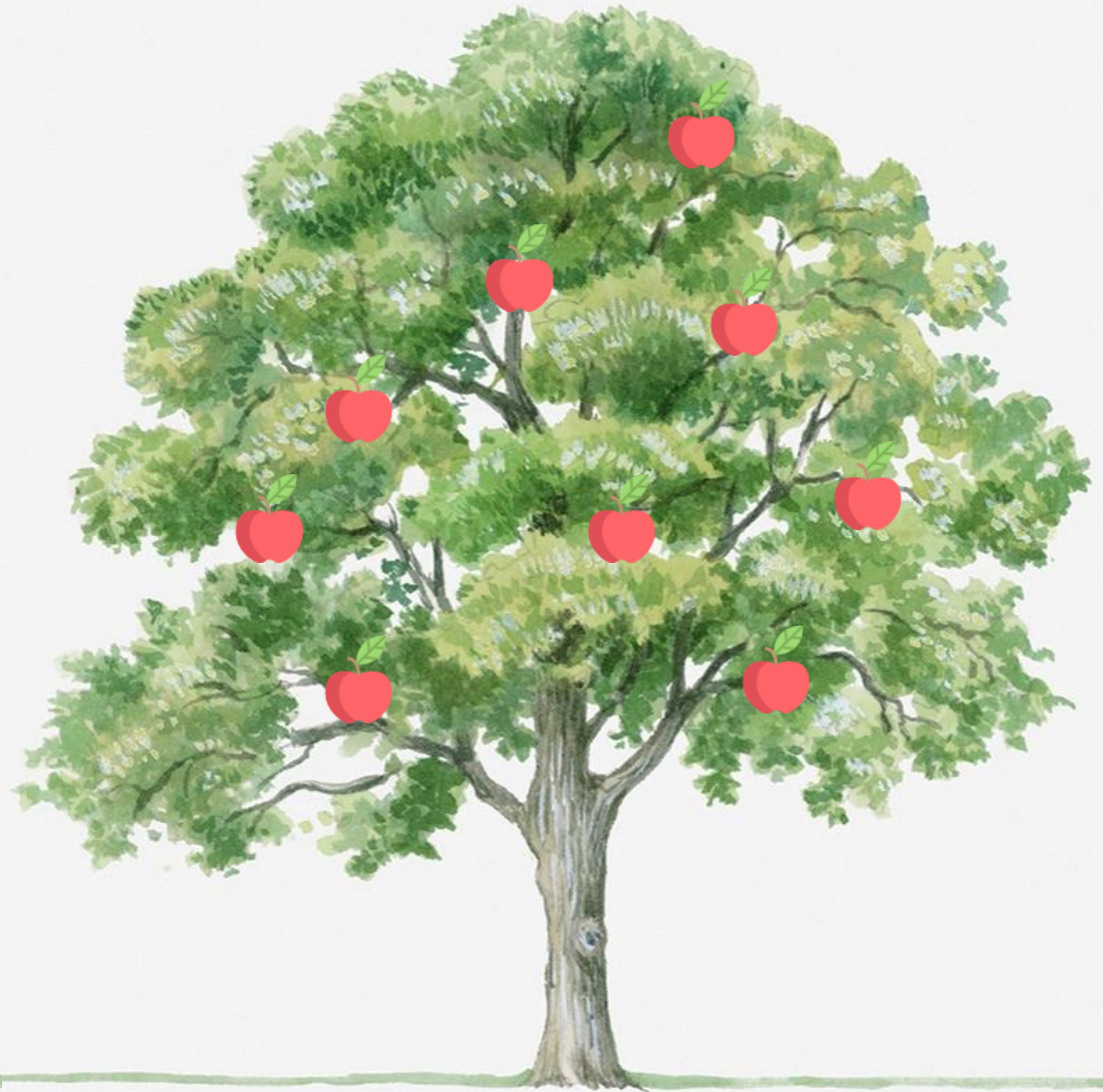


차별점

구분	기존(AS-IS)	나무지킴e(TO-BE)
스마트 서울맵	<ul style="list-style-type: none"> - 서울 지역에 한정 - 가로수 현황을 기록하는 것에 그침 - 시민돌봄사업에 대한 안내 부족 	<ul style="list-style-type: none"> - 전국 단위 서비스 시행 - 수집된 가로수 현황을 향후 사업에 반영 - 시민봉사활동 참여 및 관리 매뉴얼 제공
네이처링의 가지치기 프로젝트	<ul style="list-style-type: none"> - 적은 유인으로 인한 참여 부족 - 지역별 현황에 대한 시각자료 부족 - 2021년 이후 사실상 중단 	<ul style="list-style-type: none"> - 녹색생활실천포인트와 연계하여 유인 확보 - 가로수 현황 지도 등 시각자료 제공 - 지속적 관리를 통한 사업의 계속성 확보



감사합니다





```

num_epochs = 10
# Adjust this based on the number of training epochs you want
best_val_accuracy = 0.0
transfer_model.train()

for epoch in range(num_epochs):
    running_loss = 0.0
    progress_bar = tqdm(enumerate(train_loader), total=len(train_loader))
    for i, data in progress_bar:
        #for images, labels in train_loader:
            images, labels = data
            images, labels = images.to(device), labels.to(device)
            optimizer.zero_grad()
            outputs = transfer_model(images)
            loss = criterion(outputs, labels)
            loss.backward()
            optimizer.step()
            running_loss += loss.item()
            progress_bar.set_postfix({'loss': running_loss / (i + 1)})

    print(f"Epoch {epoch + 1}/{num_epochs}, Loss: {running_loss / len(train_loader)}")

```

```

b2.jpeg: 1 [0.20607899129390717, 0.7939210534095764]
b3.jpeg: 1 [0.21417230367660522, 0.78582763671875]
g1.jpeg: 0 [0.7452677488327026, 0.25473225116729736]
g5.jpeg: 0 [0.5066187381744385, 0.4933812618255615]
g4.jpeg: 0 [0.6874135732650757, 0.3125864863395691]
g2.jpeg: 0 [0.8111068606376648, 0.1888931542634964]
g3.jpeg: 0 [0.6647684574127197, 0.33523157238960266]
b5.jpeg: 1 [0.4579228162765503, 0.5420771837234497]
b4.jpeg: 1 [0.18944789469242096, 0.8105521202087402]
b1.jpeg: 1 [0.10374567657709122, 0.896254301071167]

```

Pre-training-model : vit_base_patch16_224 /이미지 크기
(IMAGE_SIZE) = 224 X224 (pixels)

배치 크기(BATCH_SIZE) = 32 /학습률(lr)=0.00005 / 에포크
(num_epochs) = 10

최종 정확도(accuracy)=93.33%