성장이 빠른
인재 박우영입니다.

Links

기술 블로그

깃허브 페이지

CONTACT

uyoung@snu.ac.kr

010 2102 9596

어떤 능력을 가지고 있을까?

학부(숭실대) 관련 수강 과목

- 2019 선형대수(A+)
- 2019 기초전자공학(A+)
- 2020 기계학습(A0)
- 2020 데이터베이스(A+)
- 2020 데이터베이스 응용 및 프로그래밍(A-)
- 2020 운영체제(A-)

석사과정(서울대) 관련 수강 과목

- 2022 머신러닝을 위한 기초 수학 및 프로그래밍 실습(A-)
  앰비언트 AI 부트캠프(P)
  컴퓨터 비전(A-)
  미래도시 설계(A0)

이 외 학습 과목

- Udemy - Deployment of Machine Learning Models

ML Production Cycle

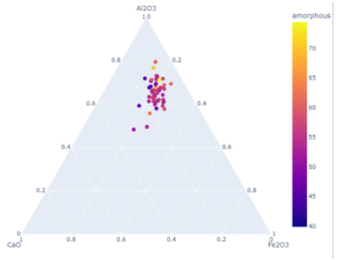
- Kubernetes 사용 경험
- Docker, 형상관리 툴 사용
- Restful API 설계 경험
- CircleCI를 통해 CI/CD 경험

Trouble Shooting

- 예외처리 및 디버깅으로 문제 원인 파악
- 공학수학을 바탕으로 한 문제 상황 해결
- 도메인 연구를 기반으로 해결방안 제시

데이터 분석

- 웹크롤링, 논문 데이터 및 실험 데이터를 통해 데이터 수집 및 변환
- EDA(Exploratory Data Analysis)를 통해 논리적 상상을 통해 가설을 세우고 확인하면서 데이터 컨텍스트(context)를 이해



AI Research

- 데이터 변환을 통해 새로운 데이터 생성
- ML개발에 사용할 데이터 베이스 구축
- ML 및 컴퓨터비전을 이용한 다양한 리서치 경험

< Lucas-Kanade를 구현하여 Car recognition >

PROJECT.1

구글 CORAL 개발보드와 센서를 사용한 AI 플랫폼 개발

"모바일 이미지에서 얼굴 탐지(detection) 및 심박수 예측 "

- 모듈 구성 : (1) Face Detection, (2) 심박수 예측
- 알고리즘 : ResNet, TS-CAN (temporal shift convolution attention network)
- 의의 : 본 프로젝트는 TS-CAN 사전학습(pre-trained) 모델 통해 32.07%의 경량화(2.34MB->0.75MB) 를 이루면서 성능을 유지함 (경량화 전 후 MAE 13.74, 14.06)

1. 도메인 : 의용생체 공학

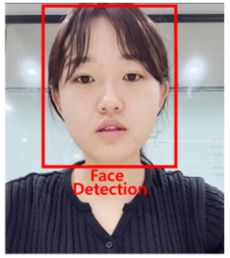
원격 광혈류 측정법(rPPG) 원리를 이용

혈류량의 변화는 심박수의 변화를 의미하며, 혈류량의 변화는 말단 조직(손 끝, 귀)의 빛 반사량을 통해 확인이 가능함. 이러한 원리를 이용하여 혈류량의 변화를 측정하는 것을 remote photoplethysmography 라고 함.

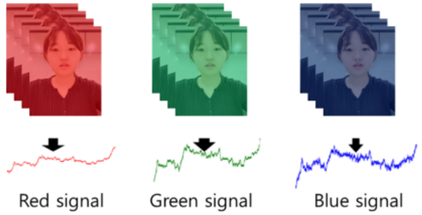
2. 프로세스

① 자신의 기기를 이용하여 15초 동안 자신의 얼굴 영상을 촬영

② Face detection 모듈이 영상 속의 얼굴영역을 감지하고 추출함.

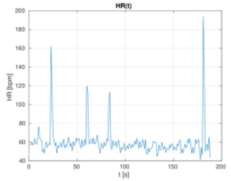


③ 얼굴 영역에서 RGB 신호 얻음.

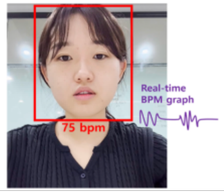


④ RGB 신호를 합쳐서 신호 재구성

⑤ 심장박동수를 예측



⑥ 실시간으로 자신의 심박수 모니터링



3. Pre-trained Model 사용

Github 오픈소스를 이용하여 Tensorflow로 Pre-trained Model을 Edge device 환경에서 구동하기 위해서 TensorflowLite로 변환하여, Coral 보드에서 구동시킴.
두 종류의 새로운 데이터셋에 적용하여 성능 비교함.

4. Architectural Design

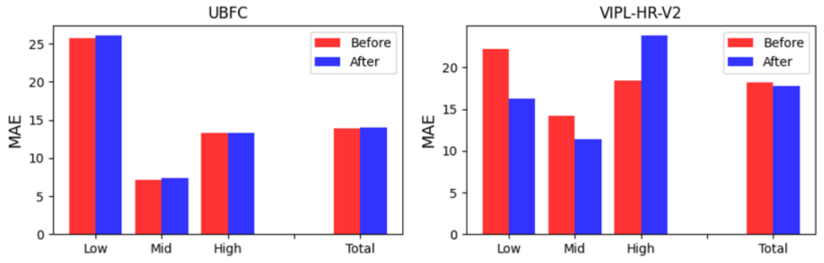
Face detection in real time — **ResNet + SSD**

+

Estimate HR values in real time — **TS-CAN**

(1) 사용자가 촬영한 영상에서 얼굴 부분을 탐지하여 해당 부분만 자름. 자른 부분만 모델의 Input Image로 사용함.
(2) 이미지에서 혈류량의 변화를 학습하여, 사용자의 심박수를 실시간으로 예측함.

5. 모델 결과



<UBFC, VIPL-HR-V2 데이터셋의 경량화에 따른 모델 성능 변화>

성능의 저하없이, 경량화를 이루어 기존에는 제공하지 않은 Edge device 환경에서 실시간 심박수 예측이 가능함.

6. 데모 영상 (유투브 링크)

7. 나의 트러블 슈팅

Tensorflow(TF)로 pre-trained 된 모델 TensorflowLite(TFL)로 변환하는 과정에서 RuntimeError

- 문제 원인 : TFL는 TF와 달리, batch inputs을 지원하지 않기 때문에 런타임 에러 발생
- 해결 : batch를 input 배열의 차원으로 추가하여, 4차원 input으로 모델링 함.

기여도 (해당 깃허브 레파지토리 링크)

기획	30
개발	30

PROJECT.2

토양의 질(SOIL ORGANIC CARBON, SOC) 예측 모델 성능 향상

"항공 이미지 및 정형 (tabular)데이터를 통해 SOC 함량 예측 모델"

- 데이터의 생성주기가 다른 점을 고려하여, 데이터 주기가 빠른 특성을 사용하여 데이터 증강을 함.
- Input features를 제거 및 추가하여 기존 논문의 예측 성능을 60% (결정계수 0.52>0.83) 높임.
- Support Vector Machine, Partial least Squares Regression, Random Forest 모두 기존 논문보다 성능이 뛰어남.

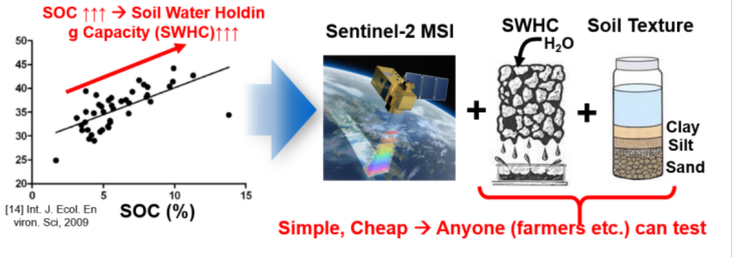
1. 도메인 : 지반공학

분광 광도계(Spectrometer)의 원리를 이용

분광 광도계는 샘플을 통과한 빛을 측정하여 샘플의 구성 및 농도 등을 측정함. 이는 토양 품질의 지표가 되는 토양탄소(soil organic corbon, SOC)의 Intensity(함유량)을 측정하는 데 사용할 수 있음.

여기서, 우리는 항공 이미지인 다중분광 데이터(MSI)에 실험 데이터(SWHC, ST)를 추가하여 정확성을 60% 향상하여 SOC를 예측함.

2. 프로세스



(1) k-means clustering을 통해 SOC 함량의 variance를 고려한 토양 채취 실험을 계획함.

(2) SNAP, QGIS 프로그램을 이용해 해당 지역의 MSI 데이터를 얻음 (input features 유형1)

(3) 간단한 실험을 통해 SWHC, ST 실험 데이터를 얻음 (input features 유형2)

3. 모델 선정 : Random Forest

(1) 입력 특성(MSI 및 실험데이터)과 타겟특성(SOC) 간의 관계가 비선형임.
(2) 데이터의 수가 적음 (대략 200개)

=> (2)를 고려했을 때, 인공신경망 모델을 사용하기 어렵고, (1)을 고려했을 때 비선형 모델을 사용해야하므로 Support vector machine(SVM), Partial Least-Squares Regression(PLSR), Random Forest(RF) 모델을 시도하였으며, RF가 가장 높게 나왔음.

4. 나의 트러블 슈팅

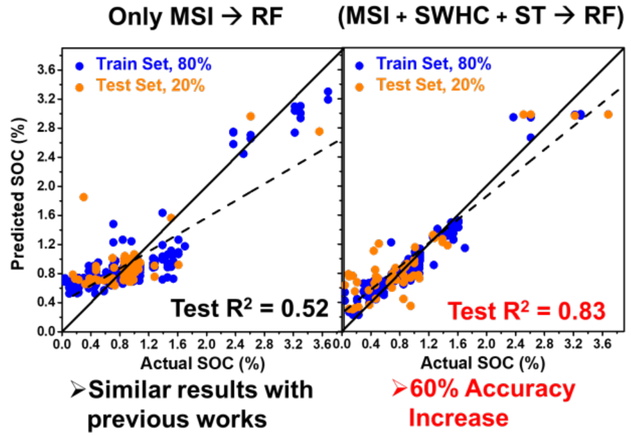
데이터에 대한 이해를 바탕으로 문제를 해결함.

- 문제 상황 : 훈련데이터가 적어서 모델이 과적합되는 문제가 발생함.
- 해결 방법 : 데이터의 생성주기가 다른 점을 고려하여, 데이터 주기가 빠른 특성을 사용하여 데이터 증강을 함.

기여도 (해당 깃허브 레파지토리 링크)

기획	30
개발	80

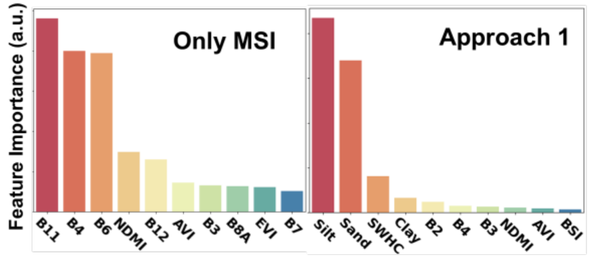
5. 모델 결과 및 분석



➢Similar results with previous works

➢60% Accuracy Increase

Top 10 most important features for SOC prediction



➢Top 3 features have wavelengths 1610nm, 665nm, 740nm
➢Associated with SOC chemistry

➢Top 4 features show predictive power of ST and SWHC
➢ST importance > SWHC importance

기존 연구는 실험적인 의의만을 갖고 있었지만, 실험 데이터를 추가 및 가공하여 성능을 향상시켜 실질적인 의의를 갖음. 성능이 60% (결정계수 0.52>0.83) 향상 되었음. 이는 항공데이터와 간단한 실험을 통해 토양의 품질을 모니터링 할 수 있음을 보여줌.

PROJECT.3

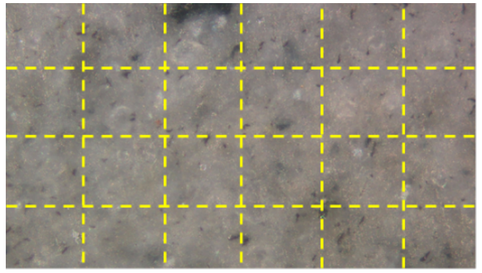
CV기반 탄소나노튜브(CNT) 혼입 재료 품질 평가

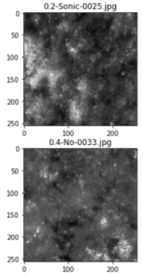
"CNT의 분산처리 및 함유량 분류"

- 광학 현미경 이미지를 통해 CNT의 (1) 분산 처리 여부와 (2) 분말 함유량에 대해 예측함.
  - 데이터 전처리(image cropping, histogram stretching)를 함.
  - CNN 모델을 통해 모델링 함.
- 본 연구는 이미지만을 통해 정량적 추적이 가능하다는 점에서 콘크리트의 품질 관리에 있어서 의의가 있음.

1. 데이터 전처리

(1) 데이터 증강 : 이미지 자르기



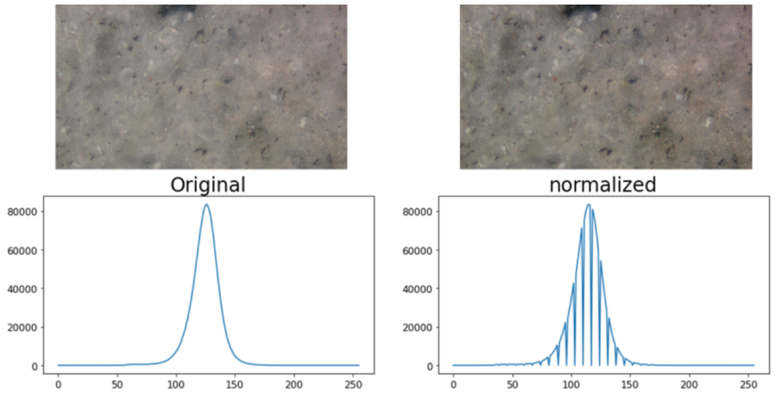


(4 by 6) * 원본이미지 322개 = 7,728개의 이미지

(2) 샘플링 : 데이터 변환은 샘플링 후 해야함.

7,728개의 이미지 데이터는 (85%, 15%) 비율로 훈련 데이터와 테스트 데이터로 나누었음.

(3) 히스토그램 스트레칭 : 명암비 높이기



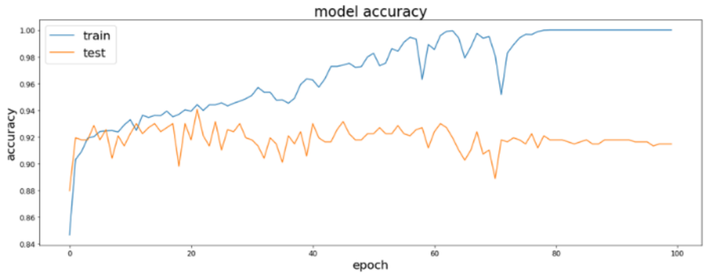
$$dst(x,y) = \frac{src(x,y) - G_{min}}{G_{max} - G_{min}} \times 255$$

2. 모델 선정 : CNN

convolution layer를 이중으로 배치하여 비교적 적은 파라미터 수로 비선형성을 증가하여 패턴을 찾음.
이미지 데이터를 히스토그램 1D 데이터로 변환하여 1D-CNN 구조를 사용함.

3. 모델 결과 (성능)

(1) 분산 처리 여부 판별 모듈



<Fig1. Model accuracy per epoch>

```
# Accuracy
Train_Accuracy = tf.model.evaluate(x_histo, y_train)
Test_Accuracy = tf.model.evaluate(x_histo_test, y_test)
```

```
206/206 [==============================] - 1s 3ms/step - loss: 0.0765 - accuracy: 0.9915
37/37 [==============================] - 0s 3ms/step - loss: 0.8280 - accuracy: 0.9033
```

<Fig2. Model accuracy per epoch>

90%정답

(2) 함유량 분류 모듈

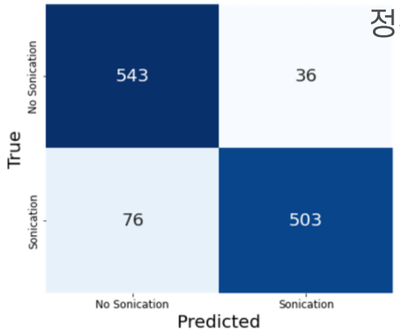
```
# Accuracy
Train_Accuracy = tf.model.evaluate(x_histo, y_train)
Test_Accuracy = tf.model.evaluate(x_histo_test, y_test)
```

```
102/102 [==============================] - 1s 6ms/step - loss: 0.0809 - accuracy: 0.9899
19/19 [==============================] - 0s 5ms/step - loss: 1.6182 - accuracy: 0.8862
```

<Fig3. Model accuracy per epoch>

89% 정답

4. 모델 평가



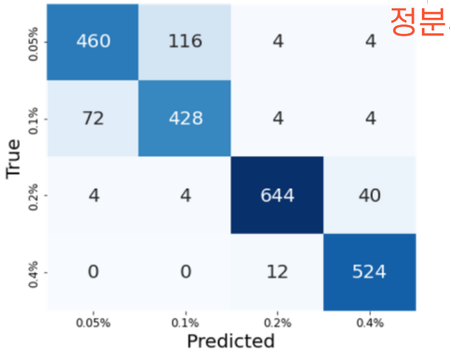
정분류율(Accuracy) : 90%

정확도(Precision) : 93%

재현율(Recall) : 87%

F1 Score : 90%

<Fig4. Confusion matrix of distributed processing module>



정분류율(Accuracy) : 89%

<Fig5. Confusion matrix of content module>

본 모델은 Accuracy가 높을 뿐만 아니라 분류마다 데이터의 수가 균형적이므로, F1 Score도 높다. 결론적으로, 성능 뿐만 아니라 공정성도 높은 모델임을 알 수 있음.

기여도 (해당 깃허브 레파지토리 링크)

기획 95

개발 95

토이 프로젝트.

OPENCV 내부 함수 구현

- cv2의 getPerspectiveTransform() 구현 : 이미지의 특징점을 이용하여 Mosaic. 일부 데이터에서 성능이 더욱 좋음.
- Lucas-Kanade 구현 : 움직이는 자동차를 탐지함.

1. Warping을 통한 Mosaic 구현 (블로그 해당페이지 링크)

이미지의 특징점(Feature point)을 이용하여 이미지를 Warping 하여 Mosaic 알고리즘을 구현하여 성능을 개선함.

Mosaicing

- 구현 방법 :

(1) 두 이미지의 Homography 행렬을 구하고, 비교 이미지와 같은 특징점(Feature Point)을 Homography 행렬을 통해 Warping 하여 두 이미지간의 특징점을 기준으로 Mosacing 하였습니다.

- 성능 개선

cv2의 getPerspectiveTransform()는 특징점을 4개만 입력 받기 때문에 본인이 구현한 알고리즘은 특징점에 대한 개수 제한이 없으므로 복잡한 이미지에서도 Mosacing이 가능하다는 점에서 성능을 높였습니다.

2. Object MotionDetection 구현 (깃허브 링크)

Lucas-Kanade 알고리즘을 구현하여 움직이는 물체를 감지

- 구현 방법 :

(1) Sequence 프레임 간의 affine parameter vector 를 구함. 이는 global optical flow라 생각할 수 있음.
(2) 주요 움직임(dominent motion)을 제거한 후, 다음 프레임 I(t+1)을 이전 프레임 I(t)에 맞추어 이동시키고, 두 이미지를 뺌.
(3) 이동 중인 객체가 있는 위치에서 높은 차이가 생기며, 이를 통해 움직이는 차를 탐지할 수 있음.

감사합니다,
박우영 지원자였습니다.

Links

기술 블로그

깃허브 페이지

CONTACT

uyoung@snu.ac.kr

010 2102 9596