**계획서**

**분리수거 척척박사 - 2조**

****

|  |  |
| --- | --- |
| **과목명** | 졸업프로젝트 2 |
| **담당교수** | 김두현 교수님 |
| **학과** | 공과대학 컴퓨터공학부 |
| **학번 및 이름** | 201814116 권태윤 |
| 201613187 송용수 |
| 201613188 송태인 |

**목차**

1. **시작 반 작품에 대한 재분석**
2. **변경 (개선 내역)**
3. **대표 시연 시나리오**
4. **문제 식별 및 해결 전략**
5. **시스템 구축 계획**
6. **SW 구현 계획 (활용도구, 오픈소스, 데이터, 알고리즘 포함)**
7. **시험 계획 (시험 항목, 통과 조건)**
8. **일정**
9. **팀원 역할 분담**
10. **시작 반 작품에 대한 재분석**

프로토타입의 주안점으로는 첫째, 재활용 쓰레기 사진 데이터 셋 부족 둘째, Object Detection 모델의 정확도 셋째, 모델과 안드로이드와의 연동 넷째, 애플리케이션내에서의 카메라와 갤러리 동작 작동이었다. 프로토타입 모델에서 대부분의 문제점을 해결하였다. 하지만 캔이나 건전지와 같은 비슷한 모양의 분류결과 또는, 빨대와 같은 분류결과에서는 높지 않은 정확도를 보여줬다. 해당 카테고리에 대해서는 추가적인 데이터셋을 수집하고 다양한 각도에서 바라본 물체의 사진을 수집할 생각이다.

또한, 1학기에 진행됐던 프로토타입의 경우 6가지 항목에 대해서만 분류하였지만 이번 완성품에 대해서는 10가지 모델에 대해서 진행하므로 더 많은 데이터셋이 필요할 것이다.

분류 모델의 경우에도 6가지보다 4가지 카테고리가 더 늘었기 때문에 이전 모델보다 비슷한 크기를 유지하거나 가벼운 모델을 만들기 위해서는 더 많은 시간과 노력이 필요할 것으로 예상된다.

데이터셋이 부족한 경우 직접 사진을 찍어 부족한 부분을 채우거나 사진을 회전시켜 추가로 학습시킬 생각이다.

1. **변경 (개선 내역)**

1학기와 큰 틀은 변화 없이 분리수거 항목 분류 개수만 6개에서 10개로 확장 시킬 계획이다. 기존에는 빨대, 컵라면, 페트병, 캔, 종이컵, 건전지 6가지에 항목이었지만 젖병, 칫솔, 우유팩, 아이스팩과 같이 일반인들이 잘 알지 못하는 분리수거 항목들에 대해서 추가했다.

기존 프로토타입 모델에서는 분류 결과만 나왔지만 이번 완성품에서는 분류 결과 뿐만 아니라 자세한 분리수거 방법을 제공해 완벽한 분리수거를 도와주는 분리수거 척척박사가 될 예정이다.

기존 6개에서 10개로 카테고리가 추가돼 모델도 무거워질것으로 예상한다. 따라서 모델을 변경하거나 학습시키는 데이터셋을 정규화시켜 모델을 가볍게 만들 예정이다.

1. **대표 시연 시나리오**
2. 앱 실행 후 "사진 업로드" 버튼을 클릭한다.
3. 재활용품 이미지를 업로드할 업로드 방식을 선택한다.

(새로운 사진 촬영 or 앨범에서 불러오기)

1. 업로드한 이미지의 분류 결과를 확인한다.
2. "상세한 분리수거 Tip“ 버튼을 클릭해 해당 항목에 대한 분리수거 방법을 확인한다.

**4. 문제 식별 및 해결 전략**

문제점

- 재활용 쓰레기 사진 데이터셋이 부족하다.

- Object Detection 모델의 정확도가 높아야 한다.

- 학습된 모델과 안드로이드 어플리케이션이 잘 연결되어야 한다.

- 어플리케이션 카메라와 갤러리를 사용해 입력된 사진을 Object Detection 할 수 있어야한다.

- 비슷한 모양의 분리수거 항목에 대해서는 정확도가 낮게 나왔으며, 가끔은 다른 물체로 인식한다.

- 하나의 분류항목 내에서도 모양이 여러가지 존재해 특정 모양에서는 높은 정확도를 보였으나, 또 다른 모양에서는 낮은 정확도

해결 방법

- 하나의 사진에 대해 여러 조도, 명도의 사진을 촬영한다.

- 많은 데이터셋 확보(기존 6개 항목에 대해서도 좀 더 추가 필요하면 해야할 듯?)

- 가장 높은 정확도 모델 사용(이피션트v4)

- 촬영 가이드라인 사진 제

**5. 시스템 구축 계획**

One Android phone

**6. 소프트웨어 구현 계획**

1. 활용도구 : Android Studio / Google Colab / Git / TensorFlow Lite
2. 오픈소스 : Tensorflow Lite (https://www.tensorflow.org/lite/guide?hl=ko)

TensorFlow Lite는 개발자가 모바일, 내장형 기기, IoT 기기에서 모델을 실행할 수 있도록 지원하여 기기 내 머신러닝을 사용할 수 있도록 하는 도구 모음입니다.

주요 특징

* 기기 내 머신러닝에 최적화됨, 5가지 핵심 제약사항 해결: 지연 시간(서버까지의 왕복 없음), 개인 정보 보호(기기에 개인 정보를 남기지 않음), 연결성(인터넷 연결이 필요하지 않음), 크기(모델 및 바이너리 크기 축소), 전력 소비(효율적인 추론 및 네트워크 연결 불필요)
* 여러 플랫폼 지원: Android 및 iOS 기기, 내장형 Linux 및 마이크로 컨트롤러 등
* 다양한 언어 지원: 자바, Swift, Objective-C, C++, Python 등
* 고성능: 하드웨어 가속 및 모델 최적화 사용
* 포괄적인 예: 다양한 플랫폼에서의 일반적인 머신러닝 작업(예: 이미지 분류, 객체 감지, 자세 추정, 질문 답변, 텍스트 분류 등)

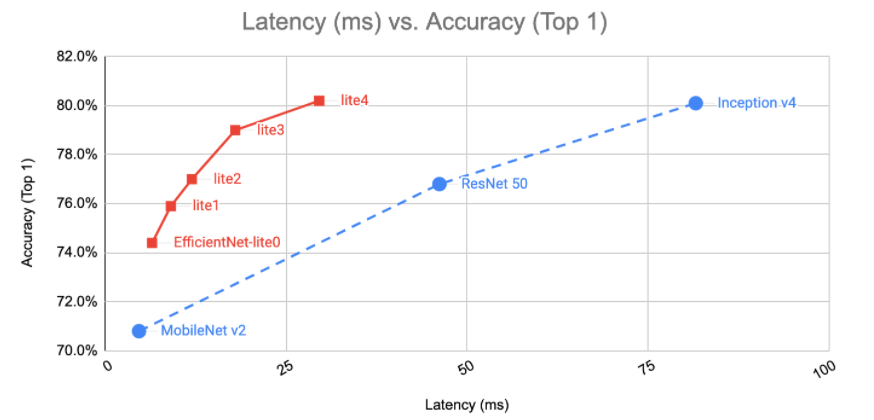
1. 데이터

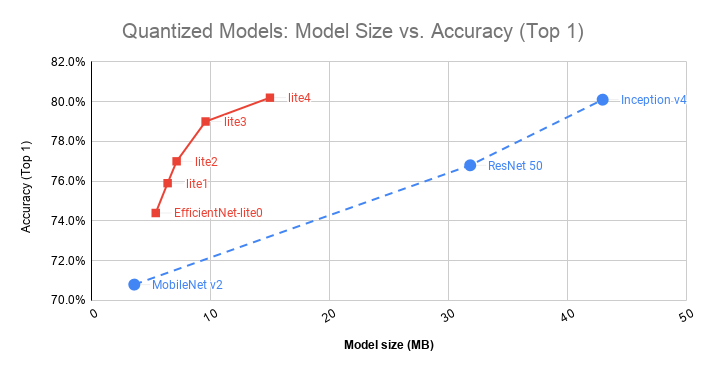
- 구글링을 통한 데이터셋 수집

- 부족한 사진의 경우 직접 촬영

1. 알고리즘 : EfficientNet-Lite V4

EfficientNet-Lite는 계산량과 매개변수 수를 현격하게 줄이면서도 최고의 정확도를 달성하는 이미지 분류 모델 모음이다. EfficientNet-Lite 모델은 양자화를 포함한 TensorFlow Lite에 맞게 최적화되어 있어 정확도 손실을 무시할 수 있을 정도의 수준에서 더 빠르게 추론할 수 있고 CPU, GPU 또는 Edge TPU에서 실행 가능하다.





테이블이(가) 표시된 사진

자동 생성된 설명

MobileNetV2, ResNet-50 및 Inception-V4와 비교시 Efficientnet-lite4 모델이 정확도와 용량에서 더 나은 결과를 가져와 기존 ResNet에서 Efficientnet-lite모델로 변경하였다.

또한, 모든 EfficientNet-Lite모델들을 사용해본 결과 표와 달리 분리수거 데이터셋에서는 EfficientNet-Lite V4가 다른 모델에 비해 속도도 빠르고 정확도가 높게 나와 EfficientNet-Lite V4를 사용하였다.

텍스트이(가) 표시된 사진

자동 생성된 설명

CNN의 성능을 높일 수 있는 요소로는 Depth (d), Width (w), Resolution (r) 3가지가 있다.

EfficientNet 모델은 기본적으로 width scaling과 depth scaling, resolution scaling 3가지의 방식을 통해 모델을 학습시키며 EfficientNet의 알파, 베타, 감마 값은 간단한 grid search를 통해 구하는 방식을 제안하고 있으며, 처음 단계에서는 파이를 1로 고정한 뒤, 타겟 데이터셋에서 좋은 성능을 보이는 알파, 베타, 감마 값을 찾아낸다.

테이블이(가) 표시된 사진

자동 생성된 설명

위 표는 ImageNet에 대한 실험 결과이다. 기존 ConvNet들에 비해 EfficientNet 모델은 비슷한 정확도를 보이면서 parameter수와 FLOPS 수를 굉장히 많이 절약할 수 있는 것을 알 수 있다. 또한, 기존에 ImageNet 데이터셋에서 가장 높은 정확도를 달성했던 GPipe 보다 EfficientNet 모델은 더 높은 정확도를 달성하는 것을 확인할 수 있다. 동시에 parameter 수와 FLOPS수도 굉장히 많이 절약할 수 있어 굉장히 좋은 결과를 보여주는 것을 바로 알 수 있다.

1. **시험 계획 (시험 항목, 통과 조건)**

<기능>

1. 처음 사용하는 사용자도 쉽게 사용할 수 있는가?

-> 임의의 3명을 뽑아 테스트 진행

2. 이미지를 불러올 수 있는가?

3. 이미지를 직접 촬영해서 할 수 있는가?

4. 분류 결과와 분리수거 방식이 일치하는가?

->

<모델 성능>

항목으로는 이미지 분류 시 오 분류될 확률이 높은 항목들로 정했다

1. 찌그러진 캔과 정상적인 캔 모두 캔으로 인식하는가?

2. 다양한 모양의 페트병을 모두 페트병으로 인식하는가?

3. 다양한 모양의 컵라면을 모두 컵라면으로 인식하는가?

4. 캔의 색상이 달라도 모두 캔으로 인식하는가?

5 투명한 페트병도 페트병으로 인식하는가?

6. 컵라면 용기와 종이컵이 비슷하게 생겼는데 이를 구분하여 인식하는가?

7. 구부러진 빨대와 구부러지지 않은 빨대를 모두 빨대로 인식하는가?

8. 건전지와 캔이 비슷하게 생겼는데 이를 구분하여 인식하는가?

9. 뚜껑을 씌운 젖병의 경우 페트병과 비슷한데 잘 구분할 수 있는가?

10. 여러가지 종류의 우유팩을 인식할 수 있는가?

11. 길쭉한 모양으로 비슷한 칫솔과 빨대를 잘 구분할 수 있는가?

12. 뚱캔 인식 가능?

13. 아이스팩도 종류 다양…?

14. 다양한 건전지….?

// . 캔을 위에서 찍을 때와 옆에서 찍을 때 모두 캔으로 인식하는가?  
// 페트병을 위에서 찍을 때 페트병으로 인식할 수 있는가?

1. **일정**

1주차 08-30 ~ 09-04 데이터 셋 수집

2주차 09-06 ~ 09-11 데이터 셋 라벨링

3주차 09-13 ~ 09-18 분류 모델 학습 + 계획서 작성

4주차 09-20 ~ 09-25 분류 모델 학습 + 모델 경량화 + 계획서 작성

5주차 09-27 ~ 10-02 분류 모델 학습 + 모델 안드로이드 탑재

6주차 10-04 ~ 10-09 자세한 분리수거 항목 추가

7주차 10-11 ~ 10-16 오류 난 부분 해결

8주차 10-18 ~ 10-23 어플리케이션 UI개선 + 중간 발표

9주차 10-25 ~ 10-30 어플리케이션 1차 수정

10주차 11-01 ~ 11-06 어플리케이션 2차 수정

11주차 11-08 ~ 11-13 어플리케이션 3차 수정

12주차 11-15 ~ 11-20 시스템 테스트

13주차 11-22 ~ 11-27 시스템 성능 개선 + 오류 수정

14주차 11-29 ~ 12-04 최종 보고서 작성 + 시스템 안정화

15주차 12-06 ~ 12-11 최종 보고서 작성 + 최종 발표 준비

16주차 12-13 ~ 12-18 최종 발표 및 시연 + 최종 보고서 마무리

1. **팀원 역할 분배**

- 권태윤 : 데이터셋 구축, 모델 설계, 모델 학습

- 송용수 : 데이터셋 구축, 라벨링 작업, 모델 학습

- 송태인 : 데이터셋 구축, 안드로이드 어플리케이션 개발, UI/UX 디자인