

캡스톤 디자인의 발표 요령

프로젝트 발표는 10분(데모포함), 질의응답 10분등, 총 20분에 발표 및 데모가 이루어져야 한다. 다음은 캡스톤 디자인 발표 자료에 포함될 내용이며 발표 자료(ppt)의 작성은 최종보고서 및 데모 결과를 토대로 작성한다.

- 프로젝트 제목, 프로젝트 목표 및 개요
- 아키텍처 또는 전체 시스템 구성도 제시
- 주요 기능목록(캡스톤 디자인의 범위) 및 설명
- 시스템의 성능, 품질 및 제약 요구사항의 테스트 결과
 - 성능 테스트 결과 등
- 주요 알고리즘 및 설계 방법(또는 문제해결 방법)
- 팀원(또는 멘토)의 역할
- 데모

- 질의 응답

캡스톤 디자인 I 최종결과 보고서

프로젝트 제목(국문): 딥러닝 기반 질병사진 분석 도구 개발

프로젝트 제목(영문): Development of Deep Learning-Based Disease Photo Analysis Tools

프로젝트 팀(원): 학번: 20222030 이름: 양예은

프로젝트 팀(원): 학번: 20222014 이름: 함승희

1. 중간보고서의 검토결과 심사위원의 '수정 및 개선 의견'과 그러한 검토의견을 반영하여 개선한 부분을 명시하시오.

없음

2. 기능, 성능 및 품질 요구사항을 충족하기 위해 본 개발 프로젝트에서 적용한 주요 알고리즘, 설계방법 등을 기술하시오.

2-1. U-Net 기반 Segmentation 알고리즘

베타세포의 형태적 특성을 추출하기 위한 dataset.

IoU 및 Dice Score 기준으로 성능 측정.

2-2. 전처리 모듈

OpenCV 및 Albumentations를 통해 이미지 증강 및 정규화 수행.

2-3. MFC-Python 연동 설계

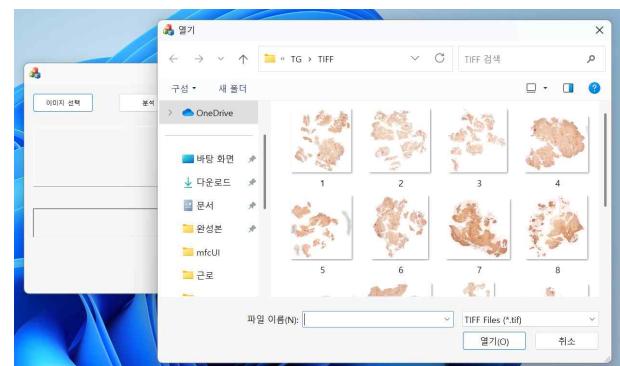
GUI 클릭 이벤트 → Python 분석 요청 구조로 설계하여 직관적 사용자 경험 제공.

3. 요구사항 정의서에 명세된 기능 및 품질 요구사항에 대하여 최종 완료된 결과를 기술하시오.

전체 시스템 구성

1. GUI(MFC) → 클릭 이벤트 → Python 분석

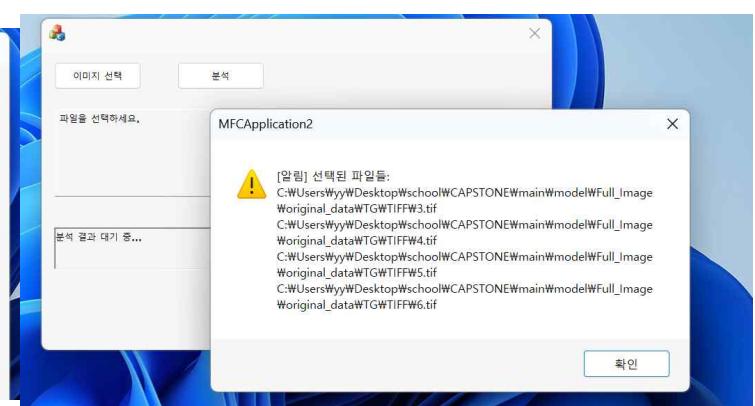
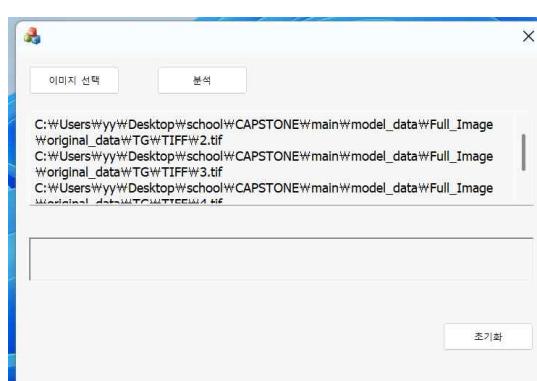
2. 분석 결과 → 이미지 마스크



mfc 첫 화면

->

이미지 업로드



이미지 다중 선택

->

이미지 분석

기능별 상세 요구사항:

- 이미지 업로드 → 분석 요청 → 자동 segmentation → 결과 시각화 및 저장

설계 모델:

- 클래스: ImageDialog.cpp, MFCApplication2Dig.cpp
- 모듈: model.ipynb, preprocessing.py, server_module.py, analyze.py

4. 구현하지 못한 기능 요구사항이 있다면 그 이유와 해결방안을 기술하시오,

최초 요구사항	구현 여부(미구현, 수정, 삭제 등)	이유(일정부족, 프로젝트 관리미비, 팀원변동, 기술적 문제 등)
결과 시각화	미구현	MFC와 모델간 연동이 미흡

5. 요구사항을 충족시키지 못한 성능, 품질 요구사항이 있다면 그 이유와 해결방안을 기술하시오

분류(성능, 속도 등) 및 최초 요구사항	충족 여부(현재 측정결과 제시)	이유(일정부족, 프로젝트 관리미비, 팀원변동, 기술적 문제 등)
		해당사항 없음

6. 최종 완성된 프로젝트 결과물(소프트웨어, 하드웨어 등)을 설치하여 사용하기 위한 사용자 매뉴얼을 작성하시오.

환경 준비:

1. PYTHON 3.8.16
2. GPU: NVIDIA A5000 x 2
3. CPU AMD Ryzen 3 5300U with Radeon Graphics
4. pip install :
torch torchvision torchaudio opencv-python albumentations
segmentation-models-pytorch numpy matplotlib pandas scikit-learn

실행 파일:

* 제출 파일을 "root:W"로 놓고 설명

- python

1. root:Wmodel_train\Full_train_Model(autolabeling_with_no_background).ipynb
: segmentaion 모델 학습 코드
: 전처리부터 테스트까지 들어감
: 이를 기반으로 preprocessing.py, server_module.py 작성
: root:Wmodel_train\Full_Image\Autolabeling_dataset\ckpt\Full_Image_model.pth

2. preprocessing.py

: 전처리 모듈 코드
: 입력 구조에 맞도록 512x512로 크롭

3. server_module.py

: 도커에서 학습시킨 모델 로드

```
model = torch.load(os.path.join(
    valid: 100%
Evaluation on Test Data:
Mean IoU Score: 0.9362
Mean Dice Loss: 0.0339
[ WARN:00152:2741 global load]
```

- “root:result(prediction”, “root:result(compare(original-label-img)” 파일은 server_module에서 모델로드를 test검증 하며 생성된 데이터
- MFC
 - 1. ImageDialog.cpp,
: 이미지 분석 후 결과 시각화하기 위한 코드
 - 2. MFCApplication2Dig.cpp
: 이미지를 분석하기 위해 analyze.py와 연동되는 코드
- 실험을 위한 tif 파일은 “main\model\Full_Image\original_data\TG\TIFF”
-

7. 캡스톤디자인 결과의 활용방안

- **기술적 파급효과:** 수동 분석 방식의 자동화로 의료 분석 효율성 향상
- **사회적 파급효과:** 연구자 및 의료진의 업무 부담 감소 및 신뢰도 향상
- **경제적 기대효과:** 고가 상용 소프트웨어 대체 가능성 확보 및 오픈소스 활용으로 개발비 절감
- **확장성:** 다른 조직(간, 폐 등) 및 세포 분석 시스템으로 확장 가능