[붙임4]

캡스톤 디자인 II 최종결과 보고서

프로젝트 제목(국문): 딥러닝 기반 병리사진 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

지도교수: 김태훈 교수

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

결과 데이터 추가

단순한 세그멘테이션(Segmentation) 결과뿐만 아니라, 원본 이미지와의 비교 및 정량적 분석 결과의 시각화가 필요함.

→ 4분할 결과 뷰어(Quad-View) 구현

단일 이미지만 보여주던 기존 방식에서 벗어나, 다이얼로그 창을 4개 영역으로 분할하여 다양한 분석 정보를 동시에 제공하도록 개선함.

좌상: 원본 이미지 (Original)
우상: Contour 오버레이 (contours)
좌하: 히스토그램(Histogram)
우하: 블러링 이미지 (blurrd)

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

가. AI 모델 및 알고리즘 (Model & Training)

1. **모델 아키텍처 (U-Net++):**
   - 세포 이미지와 같이 경계가 모호하고 크기가 다양한 객체를 정밀하게 탐지하기 위해 **U-Net++ (Nested U-Net)**구조를 채택함.
   - **Dense Skip Pathways**를 통해 인코더와 디코더 사이의 의미적 격차(Semantic Gap)를 줄여 미세한 경계선 검출 성능을 극대화함.
   - Encoder 백본으로 **EfficientNet-B2**를 사용하여 정확도와 연산 속도의 균형을 맞춤.
2. **데이터 증강 (Albumentations):**
   - 의료 데이터 부족 문제를 해결하고 모델의 강건성(Robustness)을 확보하기 위해 **Albumentations**라이브러리를 활용함.
   - **기하학적 변환:**Horizontal/Vertical Flip, ShiftScaleRotate (회전 불변성 학습).
   - **조명/화질 변환:**CLAHE(대비 강화), RandomBrightnessContrast, Blur, Gaussian Noise 등을 적용하여 다양한 촬영 환경에 대응.
3. **대용량 이미지 처리 (Tiling):**
   - 고해상도 이미지를 한 번에 처리할 수 없으므로, **512x512 픽셀 크기의 타일(Tile)**로 분할하여 추론한 뒤 다시 원본 크기로 병합(Stitching)하는 알고리즘 적용.

나. 시스템 설계 및 아키텍처 (System Architecture)

1. **하이브리드 아키텍처 (C++ & Python):**
   - **UI/UX (View):**안정적인 Windows API 제어를 위해 **MFC (C++)**사용.
   - **AI Engine (Model):**강력한 딥러닝 라이브러리 활용을 위해 **Python**사용.
2. **MVC 패턴 적용:**
   - 시스템을 **Model(FastAPI Server)**, **View(MFC)**, Controller(Python Bridge)로 분리하여 유지보수성 및 확장성 확보.
3. **성능 최적화 기술:**
   - **JIT (Just-In-Time) 컴파일:**PyTorch 모델을 TorchScript로 변환하여 모델 로딩 속도 및 추론 속도 향상.
   - **Job Object:**Windows Kernel Object를 사용하여 메인 프로세스 종료 시 서버 프로세스까지 안전하게 자동 종료되도록 생명주기 관리.

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

가. 전체 시스템 아키텍처 (On-Premise MVC 패턴)

본 프로젝트는 외부 인터넷 연결 없이도 로컬 PC에서 즉시 구동 가능한 **독립 실행형(Standalone)**시스템으로 구현되었습니다.

- **VIEW (MFC Client - InnoMedics.exe):**메인 실행 파일 (사용자 진입점)
  - 사용자 인터페이스 담당. 파일 탐색, 분석 시작/취소 제어, 결과 이미지(4분할) 출력.
  - **Job Object 관리:**프로그램 실행 시 백그라운드 AI 서버(server.exe)를 자동으로 실행하고, Windows Job Object에 등록하여 메인 프로그램 종료 시 서버 프로세스 트리(Tree)가 안전하게 자동 소멸되도록 생명주기를 관리함.
  - Direct2D/GDI+를 활용한 고품질 이미지 렌더링.

CONTROLLER (Python Bridge - controller/controller.exe):클라이언트 브릿지 폴더

- MFC와 AI 서버 간의 비즈니스 로직 중계.
- 이미지 전처리(Tiling) 수행 후 로컬 서버(127.0.0.1)와 고속 HTTP 통신 수행하며, 진행률(Progress) 및 남은 시간(ETA) 계산하여 View에 전달.

MODEL (FastAPI Server - server/server.exe): AI 엔진 폴더 (서버 실행 파일 및 모델)

- 핵심 AI 추론 엔진. 로컬 호스트(Localhost)에서 구동되며, JIT 컴파일된 모델을 메모리에 상주(Memory Resident)시켜 **지연 없는(Zero-Latency)**응답 환경 구축.
- (확장성) 아키텍처 상 REST API를 사용하므로, 필요시 코드 수정 없이 Docker/클라우드 환경으로 확장 가능함을 실증함. → 실제로 캡스톤 발표 때는 학교 GPU 서버의 Docker로 진행

나. 기능별 상세 완료 내역

1. **이미지 분석:**Tiff 등 고용량 의료 이미지 로드 및 딥러닝 기반 세그멘테이션 완료.
2. **실시간 모니터링:**분석 진행률 및 예상 소요 시간(남은 시간) 실시간 텍스트 출력 기능 구현.
3. **프로세스 제어:**'초기화' 버튼을 통한 Soft Reset(분석 프로세스 즉시 중단 및 재대기) 기능 구현.
4. **결과 리포팅:**원본, 블러링, 컨투어, 히스토그램의 4가지 관점 분석 결과 자동 저장 및 시각화 완료.

다. 설계 모델 (폴더 및 파일 구조)

- **실행 파일:**InnoMedics.exe(Main), python/server/server.exe(AI Engine), python/controller/controller.exe(Bridge)
- **데이터 흐름:**MFC → (Pipe) → Client → (HTTP) → Server → (Inference) → Controller → (File Save) → MFC View

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)
		해당사항 없음

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

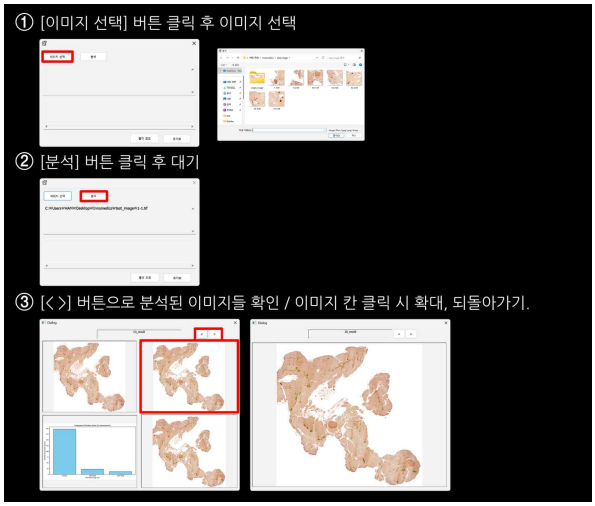
6. 최종 완성된 프로젝트 결과물(소프트웨어, 하드웨어 등)을 설치하여 사용하기 위한 사용자 매뉴얼을 작성하시오.

1. 시스템 설치 및 구동
- 별도의 설치 과정 없이 첨부한 폴더 내의 InnoMedics.exe파일만 실행하면 됩니다.
- **(자동화 기능)**사용자가 별도로 서버를 켤 필요가 없습니다. 메인 프로그램 실행 시 **내부적으로 AI 추론 서버(server.exe)가 백그라운드에서 자동 실행**되며, 최적화된 상태로 대기합니다.

2. 이미지 로드 및 분석
- **[파일 선택]**버튼을 클릭하여 분석할 의료 이미지(Tiff, Png 등)를 선택합니다.
- **[분석 시작]**버튼을 클릭하면 로컬 서버를 통해 즉시 분석이 수행됩니다. 하단 로그 창을 통해 진행 상황이 실시간으로 표시됩니다.

3. 분석 중단 (Soft Reset)
- 분석 도중 **[초기화]**버튼을 누르면, 현재 분석 중인 프로세스만 즉시 중단됩니다. 서버는 백그라운드에서 계속 유지되므로, 재시작 대기 시간 없이 바로 다음 이미지를 분석할 수 있습니다.

4. 시스템 종료
- 프로그램 우측 상단의 **[X]**버튼을 눌러 종료합니다.
- **(안전 종료)**Windows Job Object 기술이 적용되어 있어, 사용자가 종료하는 즉시 백그라운드의 AI 서버 프로세스까지 **잔여물(Zombie Process) 없이 깔끔하게 일괄 종료**됩니다.



7. 캡스톤디자인 결과의 활용방안

가. 기술적 파급효과
- **OS 레벨의 프로세스 제어 기술 구현:**단순히 AI 모델만 돌리는 것이 아니라,

Windows Job Object를 활용하여 다중 프로세스(Multi-Process)의 생명주기를 동기화하는 시스템 소프트웨어 기술을 확보함.

- **유연한 하이브리드 아키텍처:**기본적으로는 보안과 속도를 위해 **로컬(On-Premise) 서버**로 동작하지만, 통신 규격이 표준 HTTP(REST)로 설계되어 있어 향후 대규모 처리가 필요할 경우 **Docker 컨테이너 기반의 클라우드/GPU 서버**로 즉시 전환 가능한 확장성(Scalability)을 가짐.

나. 사회적/경제적 기대효과

- **중소 병원 맞춤형 솔루션:**고가의 서버 장비나 클라우드 구독료 없이, 일반적인 고사양 PC 한 대만 있으면 **독립적으로 운영 가능한 AI 분석 시스템**을 제공하여 도입 비용을 획기적으로 절감함.
- **보안성 강화:**모든 데이터가 로컬 PC 내부(Loopback)에서만 처리되므로, 민감한 의료 데이터의 외부 유출 우려가 없음.