**AI신발인식 및 3D좌표추출**

**1. 프로젝트 개요**

최근 인공지능 기술의 발전으로 의료데이터의 공개와 활용에 대한 패러다임이 급격히 변화하고 있습니다. 그 동안 비공개로만 취급되었던 의료 정보가 데이터 3법 등 법적 개선 및 패러다임의 변화로 인해 새로운 가능성으로 떠오르고 있습니다.

또한 최근 ICT 기술의 발전으로 인해 대용량 데이터의 축적이 가능해졌습니다. 한국은 세계 최고의 정보기술 인프라를 바탕으로 다양한 의료정보를 축적하고 있으며 이러한 데이터 분석과 변화 추이는 깊은 통찰력을 제공하고 있습니다.

AI 기술의 발전에 따라 의료산업 내 AI 도입의 가능성도 커져가고 있습니다. 데이터 공개에 따른 AI 개발이 가속화됨에 따라 다양한 분야에서 AI를 의료 산업 내 도입하고자 하는 다양한 시도가 잇따르고 있습니다.

AI 영상을 분석하고 질병을 예측하는 알고리즘을 Tensorflow 및 Keras 라이브러리를 통해 개발하고 있으며 본 실습 프로젝트에서는 치아를 찾아내어 분석하는 알고리즘 및 일반 질환 들을 분석하여 개발하는 알고리즘을 구현하고자 하였습니다.

**2. 프로젝트 목적**

구현하고자 한 알고리즘은 1) 치아의 번호를 분석하여 제공하는 알고리즘 및 2) 질환을 분석하여 예측해주는 알고리즘입니다. 본 알고리즘의 개발을 통하여 저희는 치과 의사선 생님들의 판독시간을 줄이고 판독의 정확도를 올리는 일을 수행하는 것을 목적으로 합니다.

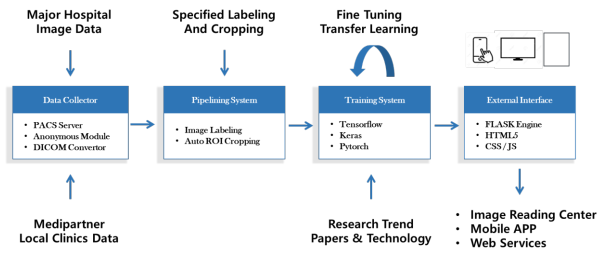


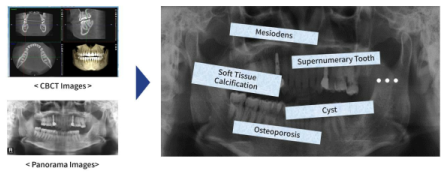
[그림1] AI기반의 의료영상 판독의 미래

**3. 프로젝트 내용**

**3-1. 프로젝트 구성도**

본 시스템은 다음과 같은 4가지 프로세스를 통해 구성되어 있습니다. 먼저 로컬 치과에서 데이터를 수집하고 이를 정제하고 레이블링하여 학습을 시키며 이를 다시 서비스화 시키는 일련의 과정을 구형하였습니다.

[그림2] 프로젝트 개발 모듈 도식도



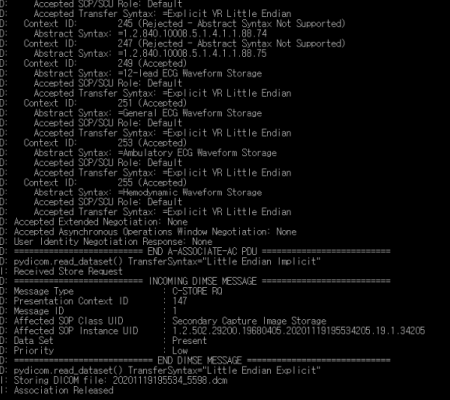
[그림3] 치과 파노라마 이미지 및 CBCT 이미지의 질병 진단 컨셉

**3-2. 데이터 정제**

- Store SCU : 로컬 치과병원의 파노라마 촬영장비에서 생성되는 DICOM 파일을 읽어내고 decompression을 한 후 분석용 서버로 전송하는 프로그램입니다. 본 프로그램의 DICOM 정보를 읽기 위해 파이썬의 PYDICOM 및 PYNETDICOM 라이브러리가 사용되었습니다.

- 익명화 모듈 : DICOM 파일 내에는 환자 정보가 포함되어 있는 경우가 있습니다. 기본적 으로 환자 동의를 구하고 난 후 학습 혹은 서비스 제공을 하게 되지만 그럼에도 불구하고 원칙적으로 모든 환자 정보는 익명화되어 이미지 정보만 전송하게 됩니다.

- DICOM CONVERTOR : DICOM 파일은 의료용 영상을 저장하는 일종의 파일 형식으로 이 DICOM 자체로는 학습 목적으로 이미지를 사용하는 것이 매우 까다롭습니다. 이를 변환 하기 위한 CONVERTOR 라이브러리를 구현 및 사용하였습니다.



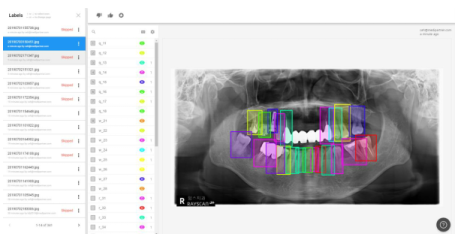
[그림4] 실제 DICOM 수신에 따른 프로세스 화면

**3-3. 파이프라이닝 시스템**

- LABEL ANNOTATION SYSTEM : 학습을 위해서는 각각의 이미지들이 통일된 규격으로 질환 및 위치 정보가 저장되어야 합니다. 저희 AIQUB는 학습 데이터의 저장, 학습을 위해

coco 데이터셋 기반 2 의 표준으로 저장을 하고 있습니다. 레이블링을 위해서는 bounding box를 그릴 수 있는 자바스크립트 기반의 오픈소스 3 를 참조하여 사용하였습니다.

- AUTO LABELING SYSTEM : 또한 학습된 모델을 탑재하여 Prediction을 통해 재학습을 시킬 수 있도록 하는 프로그램을 구현하고 탑재하였습니다.



[그림5] AUTO LABEING에 따른 포로세스 화면

**3-4. FINETUING TRAINING 시스템**

**- KERAS & TENSORFLOW 학습 서버 :** 학습을 원격에서 독립적으로 시키기 위한 학습 서버를 구축하였습니다. 이 학습 서버는 Keras 및 Tensorflow 라이브러리(아키텍쳐)를 사용 하고 있으며, 탑재되는 알고리즘으로는 Inception V3, Resnet, Dense Block 등 다양한 모델 들을 탑재할 수 있습니다. 다양한 알고리즘을 탑재하여 성능을 비교 분석함으로써 높은 정확도를 끌어올리는 과정을 수행합니다.

**- FINETUNING SYSTEM :** 본 학습 모델은 Transfer Learning 및 Finetuning 기법을 사용하여 정확도를 높이는 과정을 수행하고 있습니다. Transfer Learning을 위해서는 Imagenet 에서 트레이닝 시킨 데이터를 사용하였습니다.



[그림6] 이미지넷 기반의 데이터 셋

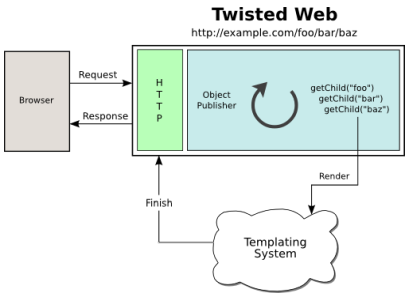
**3-5. FINETUING TRAINING 시스템**

**- HTML5, CSS, JAVASCRIPT :** 서비스 프레임 제작 용도로 활용하였습니다. 내부적으로는 이미지 로딩 및 레이블링을 위해 Javascript를 사용하였고, CSS를 통해 내부 시스템의 디자인을 구체화시켰습니다.

**- BOOTSTRAP :** 추가적인 디자인 및 반응형에 맞는 페이지 구현을 위하여 부트스트렙을 사용하여 디자인을 추가하였습니다.

**- MONGO DB :** 시스템의 DB 연동을 위해 MONGO DB를 사용하였습니다.

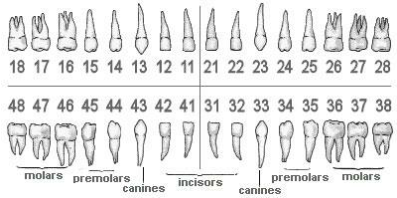
**- FLASK :** 이미지를 불러오고 파이썬 학습 라이브러리와의 연동을 자연스럽게 하기 위해 Flask 서버를 사용하였습니다. Flask 서버 사용성 및 Handling Load Balance를 유연하게 하기 위하여 Twisted, Nginx 엔진을 추가로 덧대어 사용하였습니다. Twisted 를 사용함에 따라 다중 접속에 대해서도 유연하게 대응할 수 있도록 하였습니다.



[그림7] Twisted 기반의 플라스크 서버

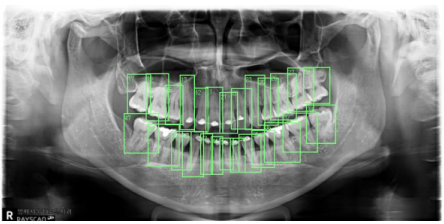
**3-6. 알고리즘 모델 및 데이터**

본 시스템의 데이터셋은 303개의 파노라마 이미지로 구성되어 있으며, 각각의 파노라마 이미지의 치아는 위치에 대한 정보 값과 각각의 치아 넘버 데이터로 구성되어 있습니다. 각각의 치아번호에 담긴 의미는 그림과 같습니다.



[그림8] ISO 표준 규격의 치아 Number

**3-7. 학습 Detection 결과**



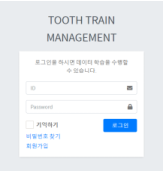
[그림9] Tooth Detection 결과



[그림10] 질병 Detection 결과

**3-8. 웹페이지 기획 및 구성**

- LOGIN : 아무나 사용하는 것이 아닌 관리자만 레이블링을 할 수 있도록 로그인 시스템을 구현하였습니다. Flask의 Session 기능 및 Permanent 옵션을 추가하여 영구 로그인이 가능하도록 하였습니다.



- TRAIN : 학습을 시키기 위한 LABEL 입력 및 PREDICTION 기능을 추가하였습니다. 우측 상단의 PREDICTION 버튼을 누르면 Prediction 이 되며 Reset 버튼을 누르면 리셋 기능이 발동됩니다.



**3-9. 결론**

전체 질환에 대한 Detection 기능의 강화 및 다양한 질환에 대한 높은 정확도를 보여주는 것으로 프로젝트를 고도화하면 좋을 것 같습니다. 또한 다양한 장비의 데이터를 고루 수집하여 장비에 따른 성능 격차를 줄이는 것도 하나의 목표로 염두하고 개선하면 좋을 것 같습니다.

**4. 주요기술**

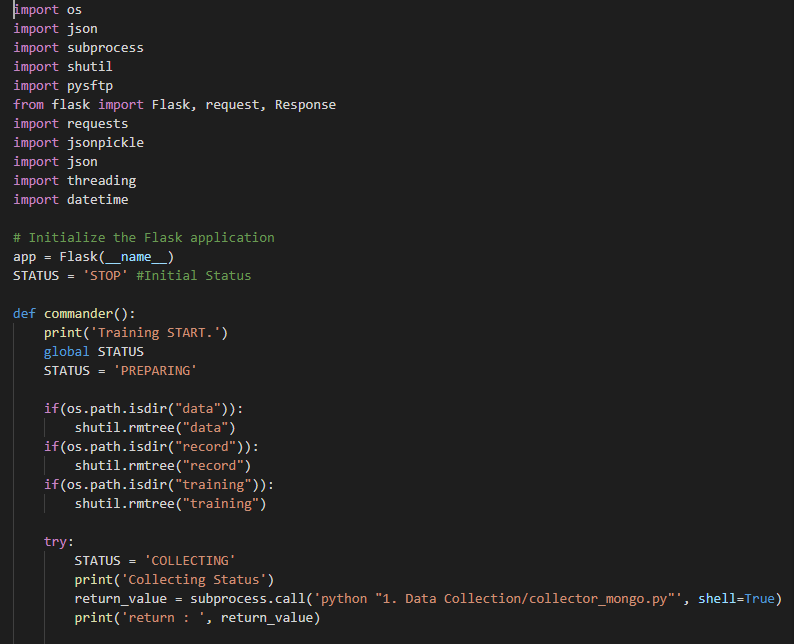
- 언어 : Python, Tensorflow

- 모델 : Inception V3를 기반으로 약간 변형

- 프레임워크 : Tensorflow, Keras

**5. 주요 소스코드**

- Training을 총괄적으로 진행해주는 커맨드 서버



- Web 쪽 Viewer 구동부

