
최성민

Email : loye2002@naver.com

Phone : 010-2546-7234

경력 요약	2018.12 ~	LG전자	Firmware 개발자
	2018.06 ~ 2018.08	의료IT 스타트업 인턴	
	2017.12 ~ 2018.02	의료IT 스타트업 인턴	
학력	2013.03 ~ 2019.02	인하대학교	컴퓨터공학과 학사
자격증	2018.05	정보처리기사 취득	

Skill Set

C, Java, Python, UEFI, ACPI, Network Socket, , PyQt5

포트폴리오 요약

C를 이용한 Firmware 신규 플랫폼 개발 (C, UEFI, ACPI)

: CPU Vendor사에서 Reference code를 전달받아 개발하는 제품에 맞게 적용하고, 필요한 기능을 개발, 추가

이미지 편집 프로그램 제작 (Python, PyQt5, DICOM, JPEG, TIFF)

: DICOM 포맷 이미지에서 원하는 ROI를 설정하여 TIFF 포맷으로 저장할 수 있는 편집 프로그램 제작

안드로이드 앱 <-> 서버간의 통신 개발 (Python, Java, Network Socket)

: 딥러닝을 활용한 실시간 사람 표정 인식 프로그램 제작

C를 이용한 Firmware 신규 플랫폼 개발

2018.12

1. 개발 업무

CPU와 BIOS vendor사에서 Reference Code를 전달받아 저희 제품의 HW 보드에 맞게 적용하고, 펌웨어 내에서 기능을 개발하기도 하고, Windows App ACPI를 통해서 기능을 개발하는 업무를 맡고 있습니다.

중요 업데이트가 있을 때마다 Unit Test를 통하여 시스템의 안정성을 확인합니다.

팀원들과 Git을 이용하여 코드 리뷰하면서 문제를 찾기도 하고, 더 나은 코드를 작성하기 위해서 함께 고민합니다.

제품 생산 라인에서 자동화 생산을 위해, UEFI 스펙과 C 언어를 사용한 툴을 제작하고 전달하여 생산 라인을 지원하고 있습니다.

BIOS Setup menu에 USB / Bluetooth / WLAN 등 여러 장치를 Enable / Disable 할 수 있는 기능을 구현했습니다.

하는 업무가 검색으로는 정보를 얻기 어렵다 보니, Vendor사에서 제공해주는 문서를 분석하고, 이해하면서 코드를 작성하고 있습니다.

2. 디버깅

Debug message를 활용하여 개발한 기능을 검토하고, 문제 원인을 파악합니다. 코드에는 특정 모듈마다 출력하는 메시지가 있고, 호출되는 순서가 있습니다. 이러한 메시지를 보고, 흐름을 따라가면서 문제의 원인을 찾습니다.

Windows Dump file

Window OS는 블루스크린이 발생하면 자체적으로 Dump File을 생성합니다. 이 파일을 활용하면, 시스템이 어떠한 문제로 블루스크린이 발생했는지에 대한 원인을 찾을 수 있는 데 큰 도움이 됩니다. 특정 HW(CPU, Memory)의 문제인지, 적용했던 코드의 문제인지, Call 스택을 통하여 문제가 발생한 원인을 찾습니다.

3. 툴 개발

[제품 생산 공정 Tool 개발]

공장 생산라인의 자동화 생산을 위해 필요한 툴을 제작하여 전달합니다. 라인 담당자에게 생산 프로세스에 대한 정보를 듣고, 필요한 툴이 어떤 것인지, 어떤 결과를 원하는지에 대한 회의를 합니다. 이렇게 회의를 마치면 저는 해당 결과를 도출해낼 수 있는 알고리즘에 대해 고민합니다.

보통 공정에서 요청하는 툴은 EFI Shell 환경에서 동작하는 툴이기 때문에, UEFI 문서를 참조하여 C언어를 통해서 개발합니다.

이렇게 툴을 제작하여 전달하면 공정에서는 확인 후, 생산 라인에 적용합니다. 이렇게 툴이 적용되면서 해당 작업에 필요했던 인원들은 다른 업무에 배정할 수 있게 되었고, 공정을 좀 더 간소화할 수 있었습니다.

이미지 편집 프로그램 제작

2018.06

- 내용 요약 : DICOM 포맷 이미지에서 원하는 ROI를 설정하여 TIFF 포맷으로 저장할 수 있는 편집 프로그램 제작
- 사용한 기술 스택 및 지식 : Python, PyQt5, nibabel, medcon, dcm2nii, DICOM, PNG, JPEG, TIFF 포맷
- 결과 : 딥러닝 데이터셋 제작을 위한 이미지 편집 프로그램 제작, 다양한 이미지 포맷(PNG, JPEG, TIFF, DICOM)에 대한 이해, 프로젝트의 설계부터 개발까지 모든 의사 결정을 경험

스타트업의 요청으로 딥러닝 연구에 필요한 이미지 편집 프로그램을 개발했습니다.

DICOM 확장자의 이미지 데이터셋 제작하기 위해서, 기존에 C#으로 제작된 JPEG 편집 프로그램에 DICOM 확장자를 사용할 수 있도록 수정해달라는 요청을 받았습니다. 프로그램이 어떻게 동작하는지 몰랐기 때문에 코드 분석부터 시작했습니다. 코드 흐름을 분석한 후, 수정하기 시작했습니다.

C#에서는 DICOM 확장자를 사용할 수 있는 라이브러리가 부족하여 DICOM → JPEG로 저장했습니다. 결과물을 확인하는 과정에서 원본 데이터의 손실을 발견했습니다. 원인을 분석해보니 이미지 포맷이 변환되면서 발생한 문제였습니다.

65536 color를 표현하는 DICOM을 256 color를 표현하는 JPEG로 변환하면서 원본 데이터의 손실이 발생한 것이었습니다.

이 문제를 해결하기 위해서 조사를 해보니, Python에서 이와 관련된 라이브러리가 많다는 것을 찾았습니다.

C#으로 제작된 프로그램을 포기하고, DICOM 포맷을 활용할 수 있는 Python으로 변경하여 처음부터 다시 설계했고, GUI를 위하여 PyQt library를 사용했습니다.

DICOM 이미지의 원본 데이터 손실이 없는 이미지 포맷을 조사를 했고, 찾은 포맷은 TIFF, PNG 두 가지였습니다. 사용 목적상 웹에서 사용할 일이 없고, 최대한 원본 데이터를 유지하기 위해서 TIFF 확장자를 선택했습니다.

원하는 데이터셋을 위해선 CT image와 MRI image가 같은 zoom 비율을 가져야하는 문제가 있었습니다. 이작업을 하기 위해선 python의 'nibabel' library를 사용해야 하는데 이때 nii 포맷이어야 가능했습니다. 그래서 'dcm2nii' 프로그램을 사용하여 DICOM → nii로 변환을 한 후, zoom 비율을 수정하여 nii 포맷으로 저장했습니다. 저장된 nii 이미지를 다시 DICOM 포맷으로 변환하고, 원하는 축 추출 및 slice를 위해서 'medcon' 프로그램을 사용했습니다.

라이브러리와 프로그램의 사용은 관련 문서를 보면서 테스트 코드를 작성하여 결과값을 확인 후 사용했습니다.

이러한 작업이 모두 끝난 DICOM 이미지를, GUI를 통해서 원하는 좌표를 설정하고, 작업자가 원하는 ROI (600 * 700 / 700 * 800)으로 Crop하여 저장하도록 하는 프로그램을 완성했습니다.

안드로이드 앱 <-> 서버간의 통신 개발

2018.03

- 내용 요약 : 딥러닝을 활용한 실시간 사람 표정 인식 프로그램 제작
- 기여한 점 : 핸드폰 카메라 장치(App)와 노트북(Server)의 통신 담당
- 사용한 기술 스택 및 지식 : Python, Java, network socket
- 결과 : 네트워크 통신 속도 개선, 서로 다른 IDE 에서 파일 전송 구현, Network socket 에 대한 이해

외부 카메라 장치를 이용하여 server 에 이미지를 전송하면 사람이 존재할 경우, 사람 표정(Happy, sad, angry 등)을 표시해주는 프로그램을 제작했습니다.

외부 카메라 장치로는 핸드폰 카메라를 이용하였고, Server 로는 노트북을 이용했습니다. 핸드폰 카메라 사용을 위하여 안드로이드 스튜디오로 앱을 제작했습니다.

앱에서 전송된 이미지 파일을 노트북에서 받아, 표정을 분석하여 결과값을 얼굴 주위에 표시합니다. 이때 앱과 노트북의 통신을 담당하여 개발했습니다.

서로 다른 IDE 에서 데이터 송수신이 가능하도록 Socket 을 이용했고, 세부 설정을 하여 기능을 구현했습니다.

성능을 확인해보니 이미지 처리속도가 사진마다 4~5 초 이상의 시간 지연이 발생했습니다. 원인을 분석해보니 사용하는 송수신 알고리즘의 문제였습니다.

기존 전송 알고리즘 :

1 장마다 socket open -> 전송 및 수신 -> close

socket open 을 사진 1 장 전송마다 호출하다 보니 시간 지연이 점차 누적되었고, 그 결과 이미지 처리속도에도 시간 지연이 발생했습니다.

변경한 전송 알고리즘 :

앱 시작 시 socket open -> 이미지 1 장의 size 전송 -> size 만큼 수신 -> 앱 종료 시 socket close

위와 같이 변경하여 사진 전송마다 호출하던 socket open 을 최초 앱 실행 시 1 회만 호출하여 시간 지연을 해결했습니다.