무니레고(Let’s go) 프로그램
프로젝트 개발 완료 보고서

제출일 **: 2024.03.13**

작성자 **:** 김규비 김민형 손웅철

기간	2024.03.08(금) ~ 2024.03.13(수) [06일]	
개발 환경	실습실/개발 서버/Windows/Visual Studio/C#/WPF	
소개	C# WPF를 기반으로 웹캠을 연결하며, 영상에서 실시간으로 레고의 도형과 색상을 검출하여 파일 송수신을 하는 프로젝트입니다.	
담당 구현 파트	김규비	**[**클라이언트**]** **MainWindow.xaml.cs //** 웹캠과 검출 로직 파일 *함수 이름: Window_Loaded() 기능: MainWindow화면을 home화면에서 호출하는 메서드로 웹캠 영상 호출과 관리자모드의 색상 선택 기본값을 지정해주는 로직 *함수 이름: End_btn_Click() 기능: 종료버튼 클릭 시 저장된 웹캠에서 저장된 이미지 디렉토리를 압축해서 송신하는 함수 호출 후 프로그램을 종료하는 로직 *함수 이름: Com_1_SelectionChanged() 기능: 검사화면에서 콤보박스 선택 시 선택된 텍스트 불러와 Txt_box변수에 저장하는 로직 **home.xaml.cs //** 시작화면 로직 파일 *함수 이름: File_mode() 기능: 웹캠에서 검사 후 저장될 이미지의 디렉토리 생성 및 오픈하는 로직 *함수 이름: Start_Btn_Click 기능: 홈화면에서 시작버튼 클릭 시 검사화면을 띄우고 홈화면을 끄는 로직 **Tcp.cs //** 비동기 클라이언트 파일 *함수이름: Send() 기능: 파일 송신하는 로직 *함수이름: GetFileList() 기능: 해당 디렉토리의 하위 모든 파일을 찾아 압축하는 로직 *함수이름: Tcp_main() 기능: 현재 클래스의 함수를 MainWindow에서 호출 시 디렉토리 경로와 서버의 ip, port로 클라이언트 네트워크의 순서대로 다른 메서드를 호출하는 로직 **[**서버**]** **Program.cs //** 비동기 서버 파일

		class File // 파일 클래스 class Client : File // 파일 클래스를 상속 받은 클라이언트 클래스 class Program : Socket // 소켓 클래스를 상속 받은 프로그램 클래스 *함수이름: Client() 기능: 생성자로 소켓받고, 버퍼로 메세지 받고 메세지 대기 하는 로직 *함수이름: Receive() 기능: 데이터 수신(파일이름, 파일이름크기, 파일크기, 파일저장)하는 로직 *함수이름: program() 기능: 생성자로 포트 리슨 서버 대기 요청으로 대기하는 로직 *함수이름: Accept() 기능: 클라이언트 접속 요청시 로직 *함수이름: main() 기능: 프로그램 시작하며 서버 네트워크의 순서대로 다른 메서드를 호출하는 로직 **[WPF]** **home.xaml //** 홈화면 기능: 상단바에 아이콘과 제목 변경, gif 무한 재생, 시작 버튼 구현한 파일 **Mainwindow.xaml //** 검사화면 기능: 상단바에 아이콘과 제목 변경, 웹캠 영상 출력, 검사 결과 출력, 관리자 모드 색상 선택 콤보박스, 종료 버튼 구현한 파일

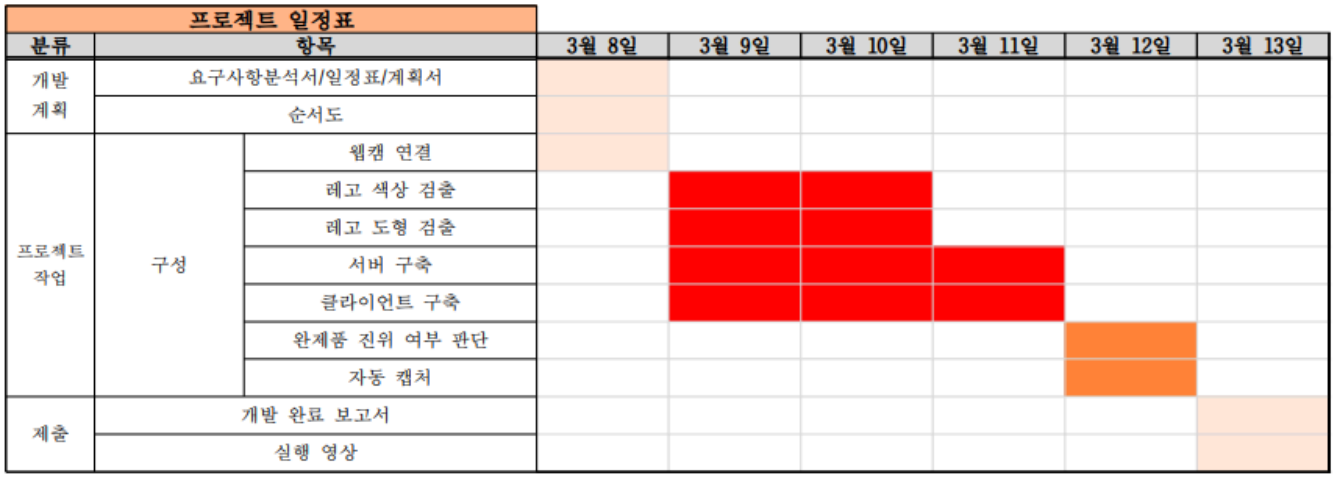
	김민형	**[**클라이언트**]**
		MainWindow.xaml.cs // 웹캠과 검출 로직 파일
		List<string> now_color = new List<string>(); / now_color.Add(lego_color1);
		기능: 색상을 담을 리스트
		GetShape(OpenCvSharp.Point[] c)
		기능: 도형 검출을 위한 도형 함수
		var m = Cv2.Moments(c); / var x = (int)(m.M10 / m.M00); / var y = (int)(m.M01 / m.M00);
		기능 : 도형이나 색상의 중심 좌표
		Cv2.Circle(frame, new OpenCvSharp.Point(x, y), 5, Scalar.Red, -1);
		기능 : 도형이나 색상의 중심 좌표를 동그랗게 그려주는 함수
		var area = Cv2.ContourArea(c);
		기능 : 윤곽선의 면적을 계산
		Cv2.DrawContours(frame, new_contours1, -1, Scalar.Red, 2, LineTypes.Link8, null, 1);
		기능 : 윤곽선을 그려주는 함수
		DateTime.Now.ToString("yyyy.MM.dd - hh시mm분ss초") / Cv2.ImWrite(path, frame);
		기능 : 캡처 및 저장

	손웅철	**[**클라이언트**]** **MainWindow.xaml.cs //** 웹캠과 검출 로직 파일 Cv2.CvtColor(frame, frame, ColorConversionCodes.BGR2HSV)

		기능: 색상공간변환 함수 호출로 색상 인식을 위해 BGR 값을 HSV 값으로 변경 Cv2.InRange(frame, new Scalar(0, 50, 100), new Scalar(10, 255, 255), mask1) 기능: 원하는 범위의 색상을 출력하여 보여주는 함수 Cv2.FindContours(mask1, out var contours1, out var hierarchy1, RetrievalModes.Tree, ContourApproximationModes.ApproxTC89KCOS); 기능:원하는 색상 범위의 윤곽선을 보여주는 함수
개발 후기	김규비	C# 및 WPF를 이용하여 웹캠을 연결하고, 영상에서 실시간으로 레고 도형과 색상을 검출하여 파일 송수신하는 프로젝트였습니다. 앞서 5번의 다양한 언어로 네트워크를 다룬 경험을 토대로 시작했지만, 파일 송수신 부분에서 어려움을 겪었습니다. 코드를 여러 차례 삭제하고 처음부터 다시 작성하는 과정에서 놓친 부분이 없는지 확인하고, 메서드로 나눌 수 있는 부분은 체크하며 문제를 해결하려 노력했습니다. 이 과정에서 3일 동안 네트워크에만 집중하며 기본서를 참고하고 손 코딩을 통해 부족한 부분을 보완했습니다. 결과적으로는 디렉터리의 하위 파일을 찾아 압축하는 형태로 송수신하는 것에 성공하게 되어 만족스러운 경험이었습니다. C# 및 WPF를 이용한 프로젝트는 처음이었기 때문에, Qt를 배울 때와 같이 하나하나 검색하고 찾아가는 과정이 필요했습니다. 특히 콤보 박스의 배경을 변경하는 부분에서는 버전 차이로 인한 문제로 인해 어려움을 겪었습니다. 여러 정보를 찾고 교수님에게 질의를 해도 원하는 결과를 얻지 못했지만, 이런 상황에서도 끈기 있게 문제를 해결하려는 자세를 기를 수 있었습니다. 이번 프로젝트에서는 과거의 경험을 되돌아보며 다양한 언어와 네트워크를 다루며 정리해둔 기록을 남긴 나를 칭찬하게 되었습니다. 덕분에 이번 프로젝트에서의 아이디어와 네트워크 기술을 다지는 계기가 되었습니다. 앞으로도 정리하는 습관과 꾸준함을 잃지 않도록 노력하겠습니다. 이번 경험이 미래의 프로젝트에 큰 자신감과 도움이 될 것이라 기대합니다.
	김민형	1주일간 팀원들과 스터디를 한 후 하는 C# 첫 프로젝트였다. C#만 사용하는게 아닌 C#을 기반으로한 WPF를 활용하여 주어진 프로젝트를 만들어야했다. 웹캠을 이용한 사물의 색상 검출과 모형 검출이 필수 구현 사항이었다. 처음에는 마냥 필수 구현 사항만 생각하고 여러 자료들을 서칭하고 수집하였다. 수집하며 알게 된 내용들은 Opencv를 활용하여 구현해야 한다는 것이고, Winform에 대한 자료는 많았지만 WPF에 대한 자료는 많지 않아 Winform의 예제들을 수정하며 프로젝트를 진행하였다. Opencv는 생소했기에 무작정 예제를 가져다 쓰는게 아닌 '이 곳에 이 함수가 왜 쓰였고, 어떤 기능을 하는지' 이해하고 학습해가며 개발해 나갔다. 제조업 컨베이어벨트에서 불량 검사로 쓰이는걸 모티브로 레고를 활용해 완제품의 기준을 정하고 기준 미달인 것들은 불량품으로 구분하여 로직을 구현하였고 웹캠에 따라 색상이나 도형 검출이 달라져서 어려움을 겪었다. 그리고 색상, 모형, 좌표를 출력하는 과정에 있어 겹쳐서 출력되는 경우가 생겨 bool형식으로 한번만 출력되게 해결하였다. 벌써 12번째 프로젝트이고 꾸준히 여러 언어를 다루다 보니 생각하는 방식이 다양해지고 접근하는 방법도 처음과는 많이 달라진거 같다. 앞으로 있을 프로젝트 또한 문제 해결 능력을 중점으로 두고 진행해 나가야겠다.

	손웅철	웹캠을 이용해서 사진이나 영상을 인식하고 그것을 통해 모양이나 색상같은 것을 구분해야한다고 했을때 처음 든 생각은 '과연 이것을 내가 할 수 있을까' 였습니다. OpenCV라는 라이브러리가 있다는 걸 몰라서 웹캠을 인식하는 것 부터 로직을 모두 구현 할 수 있을까하고 겁이 났기 때문입니다. OpneCV 덕분에 하던 걱정은 필요 없었지만 물체를 인식시키고 화면에서 색을 구분하게 하는 것은 쉽지 않았습니다. 예제를 참고해 로직을 만들어도 정확도가 많이 부족했기 때문입니다. 다큐멘터리에서 나온 이미지 검사 프로그램들은 정말 어마어마한 성능의 프로그램이라는 것을 알 수 있었습니다. 내가 만들어본 것으로 나온 결과와 내가 알고 있던 화면은 너무나도 달랐기 때문입니다. 다큐멘터리를 보면 나오던 이미지 인식 검사와 같은 인식률을 얻기 위해서는 많은 과정이 있었다는 것을 깨달을 수 있었습니다. 어려웠지만 어찌어찌 팀원들의 많은 도움으로 이번 프로젝트를 마칠수 있었습니다. 저를 도와주신 팀원분들 감사합니다.

일정표



요구 분석서			
유형	분류	세부내용	담당자
홈 화면	화면	아이콘과 gif 출력 및 시작 버튼구현	김규비
	기능	페이지의 상단에 아이콘 출력 구현	김규비
		gif 출력이 무한 재생되도록 구현	김규비
		시작버튼 클릭 시 검사화면으로 전환되며 완제품 사진을 저장할 디렉토리 생성 로직 구현	김규비
검사 화면	화면	웹캠과 모형/색상 검사결과 출력 및 종료 버튼과 정상품 선택 항목 출력	김규비
	기능	컴퓨터와 연결된 웹캠 연결 후 화면에 출력 구현	전원
		웹캠에 연결된 영상에서 실시간으로 윤곽선 추출 로직 구현	김민형 손웅철
		추출한 윤곽선을 영상에 출력 구현	김민형 손웅철
		추출된 윤곽선에서 모형 검출 로직 구현	김민형
		추출된 윤곽선에서 색상 검출 로직 구현	손웅철
		추출된 모형과 색상 좌표 추출 로직 구현	김민형
		추출된 좌표를 영상에 출력 구현	김민형
		정상품인지 불량품인지 확인하는 로직 구현	김민형 손웅철
		정상품일 경우 캡처 후 사진 저장하는 로직 구현	김민형
		관리자 기능으로 정상품 조건 변경하는 로직 구현	손웅철
		정상품 목록에서 선택 시 선택된 해당 문장 저장하는 로직 구현	김규비
		종료 버튼 클릭 시 네트워크 함수 호출하여 송수신 종료 시 프로그램 종료하는 로직 구현	김규비
[네트워크] 클라이언트	기능	클라이언트는 지정된 서버의ip와 port로 연결요청을 하고 연결된 서버로부터 메세지를 비동기적으로 송수신하여 처리하는 로직 구현	김규비
		검사화면에서 종료 버튼 클릭 시 호출되며 프로그램 시작 시 생성된 디렉토리의 하위 파일을 확인한 후 압축파일의 형태로 만드는 로직 구현	김규비
		압축된 파일을 서버에 송신하는 로직 구현	김규비
[네트워크] 서버	기능	서버는 지정된 port에서 클라이언트의 연결 요청을 수신 대기하고, 연결된 클라이언트로부터 메시지를 비동기적으로 송수신하여 처리하는 로직 구현	김규비
		서버는 다중 클라이언트 형태가 가능하도록 구현되었으며, 각 클라이언트에 고유 ID를 할당하는 로직 구현	김규비
		수신받은 압축파일을 풀고 확장자를 지정하는 로직 구현	김규비
		정리된 데이터들을 지정된 경로에 파일 형태로 저장하는 로직 구현	김규비