

개발완료 보고

	참여인원	
참여자	박민건, 박장미, 이	
개발프로젝트 소개		
프로젝트 명	로봇암 활용 불량품 자	
활동일시	'24. 07. 09. 14:00~ '24. 07. 13. 21:00	장소
개발목적	C++ 및 C# 프로그래밍 능력 향상 프로젝트의 주요 로직을 C++과 C#으로 구현함으로써 고, 다양한 프로그래밍 기법과 최적화 방법을 습득하	
	GUI 프로그래밍 경험 축적 C#과 WPF를 사용하여 사용자 친화적인 그래픽 사용 자 인터페이스 설계 및 구현 능력을 배양하고, 실무	
	로봇 제어 기술 습득 DOBOT 로봇암을 활용하여 불량품을 처리하는 기능 쌓고, 파이썬을 사용한 로봇 제어 기술을 습득하고자	
	이미지 처리 및 데이터 통신 기술 향상 OpenCV를 사용한 이미지 처리 기술을 적용하여 불 량품 이미지 데이터 송수신을 구현함으로써, 실시간 데이터 도를 높이하고자 함.	
	시스템 통합 및 최적화 능력 함양 다양한 기술 스택(C++, C#, 파이썬)을 통합하여 하 위 시스템과의 연동을 최적화함으로써, 시스템 통합 및 최적화 능력	
개발환경	- Window10, Ubuntu22.4	
	- Visual Studio 2022 WPF, Visual Studio Code, do	
	- mariaDB (DBeaver, HeidiSQL)	
서버	PJ_DETECT	
클래스	class Df	

멤버변수 및 함수	sql::Connection* conn	더
	void connect()	데이터베이스
클래스	Class_Task_i	
멤버변수 및 함수	uchar protocol[5] ={red,green,blue,yellow,postp one};	바이너리 프로토콜 unsigned char 배열 각각 0,1,2,3,10으
	void handling_image()	함수 로직의 순서 핸들러 함수이다. Receive_image, c pretreatment_image _check() 함수 순
	void receive_image()	이미지를 클라이언
	void open_image()	수신한 이미지를
	void pretreatment_image()	이미지를 HSV로 inrange 함수를 통
	bool contours_fill_check(int clnt_sock, cv::Mat HSV_image, std::vector<std::vector<Point >> contours)	컨투어를 사용하 색상이 끼어 있으
	bool contours_color_check(int clnt_sock, cv::Mat HSV_image, std::vector<std::ve ctor<Point>> contours, int color)	보류가 아닌 경우 초록색, 파란색, 노 빨간색이 추출될 결과에따라 클라 DB에 결과를 저장

	json data_send()	DB에 검사결과 테이블 가져와서 json 시켜 송신한다.
메인	int main	
	통신을 위한 준비를 하는 로직이다. Socket을 생성, 바인딩, 웨이트, 액셉트를 통해 클 통신이 이루어지고, 클라이언트 소켓 디스크립터 전달인자로 보낸다.	
백그라운드 스레드	void * backgroundthr	
	각 클라이언트에서 받은 토큰값을 통해 전달인자 소켓디스크립터를 프로토콜을 사용하여 핸들링 우선적으로 DB 클래스를 인스턴스 하여 DB객체 Task_image 클래스의 핸들러 함수에 진입한다. 클라이언트에게 그래프화할 데이터를 요청 받은 일별 불량품백분율 환산하기 위한 데이터를 전송	
사용자 정의 라이브러리	cpp_tcp_lib	
네임스페이스	cpp_tcp_	
클래스	public class cpp_V	
생성자	public cpp_Write_Read()	cpp서버S Network
	public void w_msg (string msg)	데이

멤버변수 및 함수	public void iw_msg (byte[] imageBytes)	이미지 데이터
	public string r_msg()	데이터 수신
	public string rb_msg()	서버에서 바이너리 데이터 수신
	public void close_cpp()	TcpClient 객체 소멸
사용자 정의 라이브러리	python_tcp_client	
네임스페이스	python_tcp_client	
클래스	public class python_tcp_client	
생성자	public Write_Read()	파이썬 클라이언트 생성
		NetworkClient
멤버변수 및 함수	public void w_msg (string msg)	데이터 전송
	public string r_msg()	데이터 수신
	public void close_python()	TcpClient 객체 소멸
프로젝트	Project_16	
네임스페이스	Project_16	
xaml/cs	MainWindow.xaml / MainWindow.cs	
클래스	public partial class MainWindow	

xaml	<Frame Source="Mainmenu.xaml" NavigationUIVisibility="Hidden"/>	xaml을
cs	static public cpp_Write_Read cpp_con = new cpp_Write_Read();	cpp를 데이터
xaml/cs	start.xaml / start	
클래스	public partial class	
메서드	private void btn_account_Click(object sender, RoutedEventArgs e)	버튼 클릭 이벤트 Camera.xaml로
xaml/cs	Camera.xaml / Camera	
클래스	public partial class Camera : System	
멤버변수	Mat frame Mat roiFrame	videocapture로 저장한뒤 관심영역
	BitmapSource bitmapSource	이미지 도구에 바
	python_Write_Read python_con	파이썬 클라이언트 의 객체 인스턴스
멤버함수	private void Page_Loaded()	videocapture로 저장한뒤 관심영역 이미지 도구에 영 UI 블로킹을 방지 wait_python_msg 비동기 함수로 호출
	private void wait_python_msg()	무한루프로 클라 roiFrame을 바이너리 서버에게 결과를 또한 이 결과를 클
	private void btn_log_Click(object sender, RoutedEventArgs e)	LogCheck.xaml 파일
	private void load_canvas_Click(object sender, RoutedEventArgs e)	버튼 클릭 이벤트 파이썬 클라이언트 그 후 Page_Load
xaml/cs	LogChjeck.xaml / Log	
클래스	public partial class LogChjeck : Sys	

멤버함수	public void ShowChart()	색상별로 구해진 해당 수치를 표시
	public void ShowBar()	전체 합계에서 불막대그래프에 표시
	private void Deserial()	C++ 서버와 프로부 수신 받아 읽으
	private void LogData(JToken column)	역직렬화된 데이터 분류하여 카운트한 체 합계를 구한다
	private void print_result()	색상별로 구해진
프로젝트	DOBOT.p	
멤버변수	pick = () place = {}	물체를 들어올릴 에 따라 딕셔너리
	set_flag = 0	초기 기계 작동시
	sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)	C#서버와 연결하 의 통신.
	sock.connect((TCP_IP, TCP_PORT))	서버의 IP주소와
	count = N	지정된 N번의 작
멤버함수	def move_pick(api, pick_x, pick_y, pick_z):	지정된 좌표로 이 수. z축의 좌표를

코딩	def move_place(api, place_x, place_y, place_z):	들어올린 물체를 착을 종료후 1초
박민건 후기	<p>이번 불량품 자동검출 프로젝트는 여러 기술과 많은 경험을 얻었습니다. 특히, 프로젝트의 핵심 로직을 구현하는 과정에서 기술적 역량을 크게 향상시킬 수 있었습니다.</p> <p>우분투 환경에서 C++을 사용하여 OpenCV 라이브러리를 이용하여 이미지 검출 기능을 구현했습니다. 처음에는 OpenCV의 디테일한 색상 검출 알고리즘과 이미지 전처리 과정을 이해하고 이미지를 처리하고 이벤트를 감지하는 과정에서 시스템을 구축할 수 있었습니다.</p> <p>윈도우 환경에서는 C#과 WPF를 사용하여 사용자 친화적인 인터페이스를 설계하고 구현하면서 GUI를 구현했습니다. 특히, 서버-클라이언트 구조를 통해 이미지 데이터를 주고받는 이벤트 처리 로직을 구현하면서 많은 경험을 얻었습니다.</p> <p>DOBOT 로봇암과의 연동을 통해 불량품을 물리적으로 제거하는 로직을 사용하여 로봇 제어 로직을 작성하면서, 프로젝트를 완성할 수 있었습니다. C++ 서버, C# 클라이언트, DOBOT 로봇암과의 연동을 통해 시스템을 완성하는 과정에서 시스템 통합 및 최적화를 경험했습니다.</p>	

<p>박장미 후기</p>	<p>이번 프로젝트는 처음으로 두봇이라는 외부 디비 업해 보았는데, 처음 직면한 문제임에도 불구하고 로젝트를 마무리할 수 있었습니다. 이번에는 Op 에서 캠 화면을 연동한 뒤, 관심 영역을 설정해 보았고, 서버 담당과 협업하면서 이미지를 보내고 송하는 로직을 구현했는데 이전 프로젝트에서 이 많아 헛갈리는 개념을 한 번 더 짚고 넘어갈 수</p>
<p>이동준 후기</p>	<p>파이썬을 이용하여 DOBOT 로봇 팔을 제어해 여러 시퀀스처리를 거치며 계산하면서 수학 공부 로 제어하기 위해 X축, Y축, Z축의 계산뿐만 아니 각도 계산도 필수적이라는 것을 깨달았습니다. 이 같아 조금씩 학습해보려 합니다. 파이썬을 이용한 이 있었지만, 여러 자료를 찾아보며 다시 공부해</p> <p>C++, C#, Python 세 언어의 연동을 이용한 로 를 제어하여 프로그램을 제작하고 사용할 때보다 게 느껴졌습니다. 그래서 OpenCV의 기능을 C++ Python이 담당하도록 제작하여 C#의 부담을 최</p> <p>이번 프로젝트를 통해 다양한 언어와 기술을 수 있었고, 앞으로의 공부 방향에 대해 많은 생각</p>

**김혜빈
후기**

이번 프로젝트는 Python, C#, C++에 DB까지 연동할 수 있게 하는 프로젝트였습니다. 석션을 사용하도록 옮기는 게 신기하고 재밌었습니다.

무엇보다도 이번 프로젝트를 진행하면서 TCP/IP가 되었다는 점에서 매우 수확이 큼니다. 다른 팀원에게도 급했던 이전 프로젝트들과 달리 서버-클라이언트 법까지 익힐 수 있는 매우 유익한 시간이었습니다.

막히는 부분이 있을 때면 언제든지 물어보라며 친절함이 크고 덕분에 한층 성장할 수 있었다고 자부합니다.

이번에 알게 된 내용을 기반으로 다음 프로젝트들에서 모르고 넘어갔던 부분들을 되짚어보겠습니다.

서

원
동준, 김혜빈
동검출 프로그램
광주인력개발원 공학1관
로써 두 언어에 대한 실무 능력을 향상시키고자 함.
용자 인터페이스(GUI)를 개발함으로써, 사용에서의 응용 능력을 높이하고자 함.
을 구현함으로써, 로봇 제어에 대한 경험을 다 함.
를량품을 검출하고, 서버-클라이언트 간의 이 처리 및 네트워크 프로그래밍에 대한 이해
나의 시스템을 완성하고, 각 구성 요소 간의 역할을 배양하고자 하였습니다.
obot(PYTHON)
.cpp
B

||이터베이스와의 연결객체

|스와의 연결객체(conn)를 생성한다.

mage

콜에 사용할 enum으로 구성된
|열이다.
로 구성되어있다.

|를를 명시적으로 나타내주는

open_image(),
age(),contours_fill_check(),contours_color
서로 진입한다.

엔트로 부터 수신하는 작업을 거친다.

전처리를 위해 Mat 객체에 저장한다.

색상공간을 변환하고
통해 추출할 색상영역을 특정한다.

여 색상 중간에 다른
.면 보류판단을 내린다.

. 양품/불량품을 판단한다.
o 란색이 추출될 경우, 양품
경우 불량품으로 특정지었으며
이엔트에게 프로토콜을 송신하며
장한다.

해이블에서 기계명과 양호/불량, 색, 개수 배열로 저장한 뒤 string 타입으로 덤프

o()

클라이언트와의
를 백그라운드 스레드의

read(void* arg)

으로 받은
한다.

에 참조할 변수를 생성한다.

경우, 현재 DB에 저장되어 있는 결과를
한다.

o.sln

lib

Write_Read

과 통신하기 위한 Tcpclient 객체의
인스턴스 생성 후,
데이터 송수신을 위해
kStream 객체를 GetStream 한다.

터를 송신하기 위한 함수이다.

데이터를 송신하기 위한 함수이다.
데이터를 수신하기 위한 함수이다.
저장된 데이터를 수신하기 위한 함수이다.
로 NetworkStream객체를 닫는 함수이다.
lib.sln
o_lib
o_Write_Read
엔트와 통신하기 위한 Tcpclient 객체의 인스턴스 생성 후, 데이터 송수신을 위해 NetworkStream 객체를 GetStream 한다.
데이터를 송신하기 위한 함수이다.
데이터를 수신하기 위한 함수이다.
로 NetworkStream객체를 닫는 함수이다.
s.sln
16
inWindow.xaml.cs
: System.Windows.Window

을 통해 Page1.xaml로 이동한다.

통신 라이브러리의 연결 객체와
송수신 객체를 인스턴스한다.

rt.xaml.cs

Start : Page

. 함수이다.
네비게이트 한다.

nera.xaml.cs

em.Windows.Controls.Page

블러들인 이미지행렬값을 frame에
역을 지정해서 roiFrame에 복사한다.

-인딩할 BitmapSource 인스턴스이다.

트와 통신할 서버를 여는 라이브러리
: 이다.

블러들인 이미지행렬값을 frame에
역을 지정해서 roiFrame에 복사한다.

!상을 바인딩하기 시작하면
|하면서 프로토콜을 수신하기 위해
} 함수를 Task를 사용하여
호출한다.

이언트에게 프로토콜을 수신할 때마다
너리화 하여 서버에 송신한다.
수신하여 텍스트박스에 바인딩한다.
클라이언트에게 송신한다.

페이지로 네비게이트 한다.

. 함수이다.
트와의 통신을 할 서버를 연다.
ed함수를 호출한다.

gChjeck.xaml.cs

stem.Windows.Controls.Page

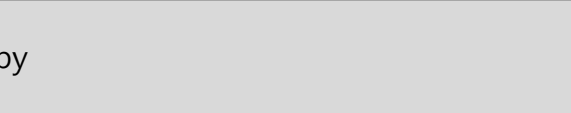
개수를 구한 후 원형그래프에 바인딩하여
한다.

량품(빨간색) 개수와 양품 개수를 구한 후
인딩하여 해당 수치를 표시한다.

!토콜을 실행하여 DB에 저장된 정보를 전
거들이고 이를 역직렬화한다.

터를 각 변수에 저장하고 이를 색상별로
한 후 Values에 카운트 수를 추가하고 전
.

개수를 xaml에 텍스트로 출력한다.



pick의 좌표와 양품, 불량, 재검토의 분류
| place의 좌표로 이동,

| 서버와의 작동 간격을 맞추기 위한 flag

기 위한 변수, IPv4를 이용하며 TCP방식

PORT번호로 DOBOT을 연결

동을 한번의 루틴으로 설정.

|동하여 물체를 흡착하고 1초 대기하는 함
리턴한다.

분류에 따라 알맞은 좌표로 이동하고 흡 대기하는 함수. z축의 좌표를 리턴한다.

도구를 활용하여 실제 문제를 해결하는 소
직을 C++와 C#으로 전담하여 구현하면서

브러리를 통해 이미지 처리와 불량품 검
|양한 기능을 익히는 데 시간이 걸렸지만,
을 최적화할 수 있었습니다. 실시간으로
여러 테스트와 디버깅을 거쳐 안정적인

다 인터페이스를 개발했습니다. 사용자 친
프로그래밍에 대한 이해가 크게 향상되었
|지 데이터를 송수신하고, 사용자 입력에
것을 배울 수 있었습니다.

적으로 처리하는 기능을 구현했습니다. 파
로봇 제어 기술과 관련된 새로운 경험을
그리고 로봇 제어를 통합하여 하나의 시
화의 중요성을 깊이 이해하게 되었습니다.

라이스를 사용해 코드와 로봇을 연결해 작
고 좋은 팀원들 덕분에 큰 어려움 없이 프
enCV로 캠과 코드를 연결해 WPF의 화면
고 영역만을 송출할 수 있도록 작업을 해
고, 서버에서 검출한 색상을 파이썬으로 전
이미 경험했던 개념들을 복습하는 부분이
있었습니다.

보았습니다. 로봇의 제어를 위해 좌표를
필요성을 느꼈습니다. 로봇을 효과적으
라 회전(Roll), 피치(Pitch), 요(Yaw)와 같은
이러한 공부들이 미래에 필요하게 될 것
한 작업이 오랜만이라 버벅이고 낯선 부분
볼 수 있는 기회가 되었습니다.

로봇 제어 및 DB 활용을 해보면서 두 언어
가 중간 역할을 C#이 담당할 때 부담이 크
다 서버로 나누고, 로봇 제어 기능은
대한 덜어내려 노력하였습니다.

연동하여 로봇을 제어하는 경험을 쌓을
것을 얻을 수 있었습니다.

동하여 로봇팔을 이동시키고 원하는 동작
중하여 설정해둔 좌푯값대로 움직이며 블

의 통신구조를 조금 더 명확히 이해하게
들이 작성한 로직을 보고 이해하기에 급
의 통신구조와 과정, 그리고 사용하는 방
다.

설하게 설명해주는 팀원들에게 감사한 마
합니다.

때 로직을 구상해보고 싶고, 이전의 프로
며 제대로 이해할 수 있도록 공부하고자