

개발 실패 보고서

제출일 : 2023년 3월 28일

팀명		참여 인원	
이상복명과외		박의용, 최지혁, 이여름	
개발프로젝트 소개			
프로젝트 명	OpenCV를 활용한 동영상 송수신 프로그램		
활동일시	3/20(월) ~ 3/28(화)	장소	광주인력개발원 공학1관 드론융합실
주요 주제	주차 관리 프로그램		
개발목적	<ul style="list-style-type: none">- OpenCV를 활용한 인식 프로그램 개발- 서버, 클라이언트 구조 간 학습- C#, winform, MySQL 학습		
파트별 임무분담	박의용	- C# 클라이언트	
	최지혁	- C# 서버	
	이여름	- C# OpenCV	
개발환경	PC, window OS, C#, Visual Studio, winform, MySQL		
일정표	별도 첨부		
요구분석서	별도 첨부		
코드 설명 (서버)	public static void insert_sql(string sql)	mysql에 insert 쿼리문 실행 메소드	
	public static int select_sql(string select)	mysql에 select쿼리문 실행 메소드	
	public static void update_sql(string update)	mysql update 쿼리문 실행 메소드	
	public static void join(TcpListener server)	클라이언트 접속을 위한 메소드	
	static void Main()	main 실행 메소드	
코드 설명	ExitLabel_Click()	X(종료)를 클릭할 경우 모든 쓰레드와 연결을 종료한다.	

(클라이언트)	parking_Load()	폼 로드 시에 webcam 연결 객체 생성후, 날짜.시간 타이머 생성한다.
	btnServerConnect_Click()	TcpClient 객체 생성 후, 서버와 연결 함수를 실행한다
	connectTcp()	연결 성공 시, 메시지를 받을 쓰레드를 생성한다.
	setLabel()	연결 상태에 따른 라벨 텍스트를 설정한다.
	streaming()	웹캠을 띄우기 위한 이미지와 비트맵 생성.
	btncctv_Click()	웹캠을 띄우기 위한 버튼 클릭 메서드.
	btncapture_Click()	웹캠 이미지 캡처.
	exitbtn_Click()	서버와의 연결 종료 버튼.
	Display(string msg)	서버에서 받은 메시지 띄우는 함수.
	Sender(string server)	서버에 메시지 보내는 함수.
	Receiver()	서버에서 메시지 받는 함수.
	Classify(string server)	서버에 보낼 메시지 분류 하는 함수.
	isClassifyMsg(string msg)	서버에서 메시지를 받은 이후, 주차관련 메시지 분류하는 함수.
	entrancebtn_Click	입차요청
	carExitbtn_Click	출차요청
	parkingstatusbtn_Click	주차현황 요청
코드 설명 (OpenCV)	public Bitmap SelectImg() 이미지 선택 메서드	멤버변수인 processedImg 전처리 이미지가 Null 일 경우 NowImg 현재 이미지를 반환하고, 그렇지 않을 경우 processedImg 전처리 이미지를 반환한다. 이미지 전처리 메서드(가우시안블러, 그레이스케일, 이진화)에 사용할 이미지를 반환하는 메서드.
	전처리 메서드 (가우시안, 흑백화, 이진화)	OpenCv 를 활용하여 이미지를 편집하는 메서드 멤버변수에 저장된 현재(기존) 이미지 / 전처리 된 이미지를 가져와서 이미지 처리를 거치고 결과를 화면에 노출한다. 메모리 누수가 발생하지 않도록 모든 메서드의 끝에 사용한 이미지 변수의 메모리를 해제했다.
	텍스트 인식	테서렉트 엔진을 이용하여 이미지상의 글자를 식별해내는 코드 <pre>try { // 경로 (폴더 내 tessdata폴더), traineddata 파일명, 모드 using (var engine = new TesseractEngine(@".\tessdata", "kor", EngineMode.Default)) using (Pix pix = PixConverter.ToPix(SelectImg()))</pre>

		<pre> using (var page = engine.Process(pix)) { this.carnumbertext.Text = page.GetText(); Console.WriteLine(page.GetText()); } } catch (Exception ex) { this.carnumbertext.Text = ex.ToString(); Console.WriteLine(ex.ToString()); } </pre>
후기	박의용	<p>Opencv를 활용한 프로젝트였는데, 여러모로 실패한 점이 많다</p> <ol style="list-style-type: none"> 1. 주제선정 부분에서 처음에 동영상 송수신을 고려하고, 여러모로 학습할 자료들을 찾아보고, 진행했는데 쉽지 않았고, 이후에 주차관리시스템이 괜찮을 것이라고 판단했지만, 그마저도 쉽지 않았다. 2. 클라이언트 부분을 완성한 이후에 서버 쪽을 같이 함께 작업하여 마무리 한 후, opencv 또한 다 같이 하려고 했지만 C++ 서버가 너무 어려워 끝내 완성하지 못하였다. 3. 본인의 학습역량, 업무분담, 프로젝트 진행 절차 전반에 반성한다. 할 수 있는게 많았다면.. 끝까지 포기는 하지 않았지만 결과를 낼 수 없는 상황에 아쉬움이 많이 남는다.
	최지혁	<p>이번 프로젝트를 하면서 어려운 점이 많았다.</p> <ol style="list-style-type: none"> 1. c++로 server를 구현하고, c#클라이언트로 통신할때 식별자를 인식을 제대로 하지못함. 2. 인코딩문제 : 한글 데이터를 수신하면 인코딩이 잘 안되서 깨졌고 결국 해결하지 못해서 c#으로 서버를 구현함 3. tcp/ip 이해도 부족 : 마음이 급한 나머지 구글링을 통해 검색한 예제로 어떻게든 하려다보니 기본 개념이 부족해서 오류가 발생해도 왜 발생하는지 몰랐고, 메시지 송수신이 이상하게 되면 원인을 찾아서 해결하기 힘들었다. <p>이러한 어려움이 있었고 다음 프로젝트를 대비해서 c++로 tcp/ip를 구현하는 개념을 차근차근 공부해 봐야겠다고 느꼈다.</p>
	이여름	<p>이번 프로젝트 진행을 위해 학습한 사항은 다음과 같다.</p> <ul style="list-style-type: none"> - C# file IO 학습 - 프로젝트에 사용된 라이브러리 환경 설정 - OpenCV의 사용법 - 이미지 자료형의 기초와 메서드를 활용하지 않은 기본 수정 원리 - tesseract 사용법 - OCR traineddata를 생성하는 법 <p>이미지의 전처리와 이미지로부터 텍스트 인식 기능은 구현하였으나 인식률을 높이기 위한 텍스트 영역(차량 번호판)의</p>

		<p>사전 검출을 해내지 못했다. 다양한 자료와 예제를 찾아 참고하였으나 진행 시간을 우려해 다소 ‘쉬운 길’을 찾으려던 것이 실패의 원인이 되었다. 차라리 고민할 시간에 어렵고 복잡해보이는 방법을 붙들고 차근히 시도해보았으면 완성할 수 있지 않았을까 아쉬움이 남는다. 지금까지 중 가장 다양한 고민과 시도를 투자했는데 정작 결과로 남는 코드는 몇 줄 되지 않아 안타깝다. C++, C#에 대한 이해도가 보다 높았더라면 진행이 수월했을 것 같다. 부족함을 느낀다.</p>
비고		