



부산대학교

PUSAN NATIONAL UNIVERSITY

OpenCV/CNN 기반 페인팅로봇 작업경로 생성

담당 교수님 : 김원석

팀명 : 톰

201824451 김정호

201824581 정제영

201724601 최성렬

목차

1. 과제 배경 및 목표

1.1 과제 배경.....	3
----------------	---

1.2 과제 목표.....	3
----------------	---

2. 제약 사항 및 대책

2.1 제약 사항.....	5
----------------	---

2.2 대책.....	5
-------------	---

3. 과제 수행 방법

3.1 개발 환경.....	7
----------------	---

3.2 사용 기술.....	8
----------------	---

3.3 시스템 구성도 및 순서도.....	10
------------------------	----

4. 개발 일정 및 역할 분담

4.1 개발 일정.....	12
----------------	----

4.2 역할 분담.....	12
----------------	----

5. 참고 문헌

1. 과제 배경 및 목표

1.1 과제 배경

최근 스마트 야드 라는 개념이 등장하여 선박의 설계, 생산, 조달, 관리 등 다양한 선박 건조 분야에 자동화를 비롯한 IT 기술이 접목하여 조선소를 스마트 야드화 하고 있다. 또한 조선 업계에서 인력 문제로 인해 자동화 도입을 시도하고 있으며 조선소 내부에서 많은 연구가 이루어지고 있지만, 도장 작업은 자동화 비율이 높지 않다. 도장 작업은 작업 시 발생하는 분진으로 인해 작업자의 건강 및 안전상의 문제가 존재하며, 블록 하부 도장 작업은 숙련된 전문인력이 직접 작업 차량을 조작하는 형태로 상당한 시간이 소모된다. 이 같은 상황에 작업 차량의 무인화를 이루어 낸다면, 작업 효율성 증대, 작업 안정성 및 자동화로 인해 산업재해 감소를 기대해볼 수 있다. 따라서 Vision 과 CNN 을 활용하여 하여 도장 로봇의 자율주행을 위한 작업경로 생성을 해보고자 한다.

1.2 과제 목표

본 과제에서 도장 로봇에서 주어진 이미지를 Calibration 하여 edge detection 을 수행, 도장 시 장애물과 경계선 등을 파악하여, 자동으로 경로를 생성하고 경로 이탈 여부를 파악하여 보고자 한다.

1. Calibration

- 보다 넓은 범위를 촬영하기 위해 4 개의 카메라를 이용하여 하나의 큰 이미지를 생성한다.
- Calibration 을 통해 4 개의 이미지를 하나의 큰 이미지로 만든다. 이 때 카메라의 내부 파라미터 행렬을 이용하여 카메라의 왜곡을 보정하고, 외부 파라미터 행렬을 조정하여 각 카메라의 방향에 맞게 transportation 하여 calibration 을 수행한다.

2. Edge detection

- 도장 작업을 위해 기준이 될 경계면을 파악하기 위해 edge detection 을 활용한다.
- Edge detection 을 수행해서 경계면을 찾아내고, YOLOv5 모델을 이용해 더 빠르고 정확한 경계면을 인식한다.

3. 작업 경로 생성 알고리즘 개발

- Edge detection 을 통해 판별된 Edge 를 이용하여 작업 경로를 생성한다.
- 도장을 수행하기 위해 경계선 탐색 후 최초 진입 점 도출 및 판별된 edge 및 장애물 정보를 기반으로 작업 영역을 구분하여 로봇의 이동경로를 생성하는 알고리즘을 개발한다.

4. 경로 이탈 감지 알고리즘 개발

- 다양한 외부 환경 요인으로 인해 경로를 이탈한 상태로 도장이 이루어질 경우 도장 퀄리티가 떨어질 가능성이 있으므로 기준선과 용접선을 이용하여 경로 이탈을 감지한다.
- 기준선과 용접선의 틀어진 각도를 비교하여 로봇의 경로 이탈 정도를 파악한다.

2. 제약 사항 및 대책

2.1 제약 사항

1. 도로에서의 경로이탈 감지는 차선이라는 명확한 기준이 있기 때문에 차선을 기준으로 경로이탈을 감지할 수 있다. 하지만 이번과제에서는 도로와 달리 기준선으로 활용할 수 있는 요소인 용접선이 소실되거나 잘 보이지 않는 경우가 존재하기 때문에 그에 따른 대책을 필요로 한다.
2. 영상 합성 시 카메라의 위치, 각도에 따른 시차 발생으로 스티칭 과정에서 왜곡이 발생해 영상이 일관적이지 않을 수 있다. 따라서 영상 정보를 바탕으로 경로를 탐색하는 이번 과제의 특성상 큰 문제가 될 가능성이 있다.
3. 온 습도 및 조명환경의 변화는 같은 위치의 카메라에서도 영상 간의 차이를 만들어 영상분석의 어려움을 야기한다. 주변환경의 변화에도 안정적으로 인식가능한 영상을 검출해내야 한다.
4. 모델 및 경로 생성 알고리즘을 완성하여도 실적용이 원활히 이루어지는지 알기 어렵다.

2.2 대책

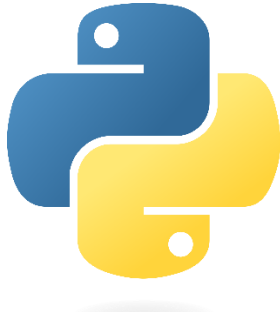
1. 평판에 기준선이 존재하지 않기 때문에 비교가능한 기준선과 경계선을 설정해야 한다. OpenCV의 edge detection 기능을 이용하여 도장, 비 도장 경계선을 검출한다. 경계선과 비교할 기준선을 카메라에 비치는 로봇의 일부로 정한다. 경계선과 기준선의 거리 차이 및 틀어짐 정도를 비교해 경로 이탈을 감지한다.1)
2. 이번과제의 영상처리에서 높은 우선순위를 가지는 것은 페인트 도장/비도장의 경계선이라고 판단된다. 따라서 경계선의 왜곡을 최소화하기 위해 경계선을 객체로 설정하여 영상 합성 시 시접선이 객체를 회피하여 추정될 수 있도록 한다. YOLOv5의 object detection 기능을 활용하여 경계선을 객체로 설정한다. OpenCV를 활용한 영상 스티칭 과정에서 설정된 객체에 가중치를 부여하여 시접선이 객체를 회피하여 추정될 수 있도록 한다. 또한 이전 영상에서 추정된 시접선과 새로 추정된 시접선의 비교를 통해 영상의 왜곡을 보정하고 일관적인 영상을 생성한다 2)

3. 큰 특징이 없는 평평한 평판에서 객체를 구별가능한 요소는 페인트로 구분가능한 색이다. 따라서 물체 인식의 정확도를 높이기 위해 색상정보를 보정하는 것이 중요하다. 주변 조도의 영향을 줄이기 위해 명도를 보정하여 선명도를 일정한 수치로 변환한다. OpenCV의 `cvtColor` 함수를 사용해 기존의 RGB 이미지로 표현된 영상을 LAB 색상공간으로 변환시킨 후 명도를 보정한다. 과반수 페인트는 붉은색을 띄므로 Red- Green 색상 정보를 가진 A 성분을 기준으로 색상 정보를 보정하는 것이 유리할 것으로 판단된다.3)
4. Unity를 활용해 시뮬레이션을 구현하는 것으로 모델 및 알고리즘을 검증해 볼 수 있다.

3. 과제 수행 방법

3.1 개발 환경

- 개발 언어 : Python



Python은 높은 확장성과 많은 표준 라이브러리를 제공하는 개발 언어이며 특히 머신러닝에서 강점을 보이는 언어이다. 영상처리와 머신러닝을 위한 라이브러리를 활용하기 위해 개발 언어로 Python을 선택했다.

- 개발 언어 : C#



C#은 마이크로소프트에서 개발한 프로그래밍 객체 지향 프로그래밍 언어이다. C#은 Unity의 스크립트 작성에 사용되며, 게임 오브젝트의 동작을 제어하고 상호작용을 구현하는데 사용된다. 따라서 이번 과제에서 시뮬레이션 구현을 위해 C#을 사용하게 됐다.

3.2 사용 기술

- OpenCV



OpenCV는 실시간 컴퓨터 비전을 목적으로 한 프로그래밍 라이브러리이며 실시간 이미지 프로세싱에 중점을 둔 라이브러리이다. 4개의 카메라로 촬영되는 영상의 합성과 edge detection을 위해 OpenCV를 사용한다.

- PyTorch



Pytorch는 Torch를 기반으로 한 Python 오픈소스 머신 러닝 라이브러리이다. 자연어 처리와 같은 애플리케이션을 위해 사용된다. 전처리된 데이터의 학습을 위해 사용한다.

- YOLOv5



YOLOv5는 object detection 분야에서 잘 알려진 모델이며 실시간으로 객체를 탐지가능한 특징이 있다. 카메라를 통해 인식되는 영상을 인식 및 분류하여 평판의 도장유무를 판별하기 위해 사용한다.

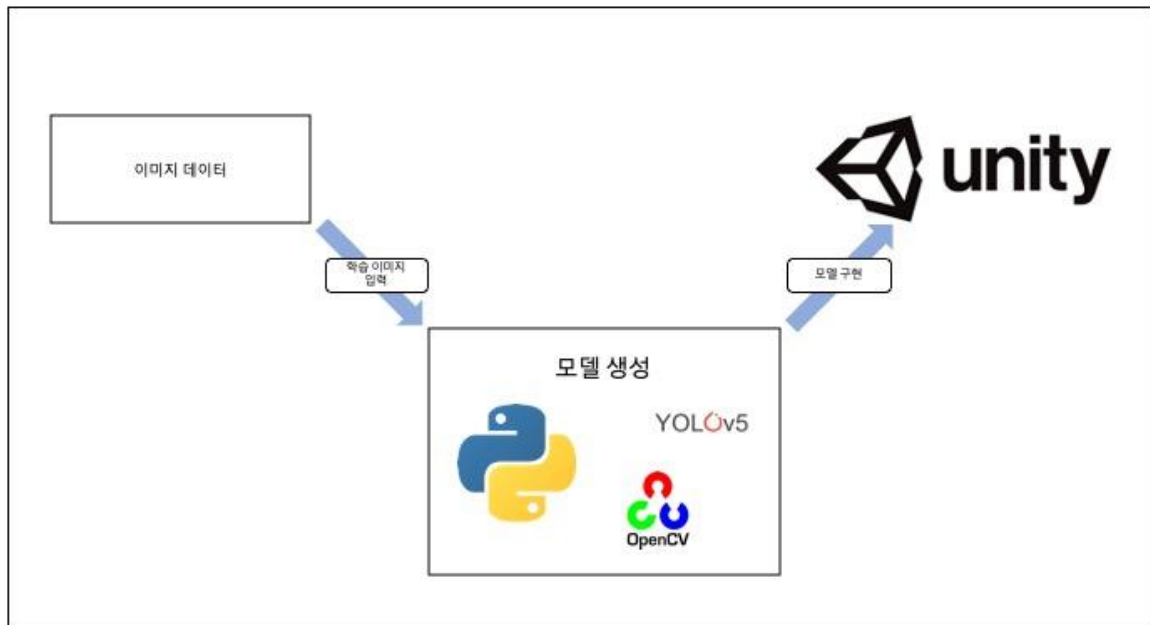
- Unity



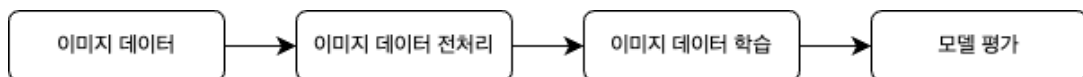
Unity는 3D 및 2D 비디오 게임의 개발 환경을 제공하는 게임 엔진이자, 3D 애니메이션과 건축 시각화, 가상현실 등 인터랙티브 콘텐츠 제작을 위한 통합 제작 도구이다. 크로스 플랫폼환경에서 작동하여 Window, Mac, 모바일 환경 등 다양한 플랫폼에서 실행 가능한 어플리케이션을 개발할 수 있으며, 사용자 친화적 인터페이스를 가져 직관적이고 배우기 쉽다.

3.3 시스템 구성도 및 순서도

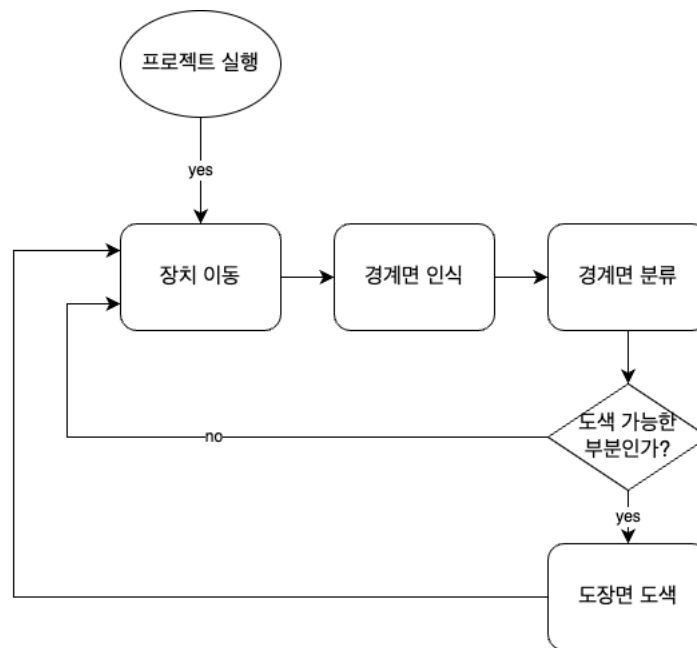
3.3.1 시스템 구성도



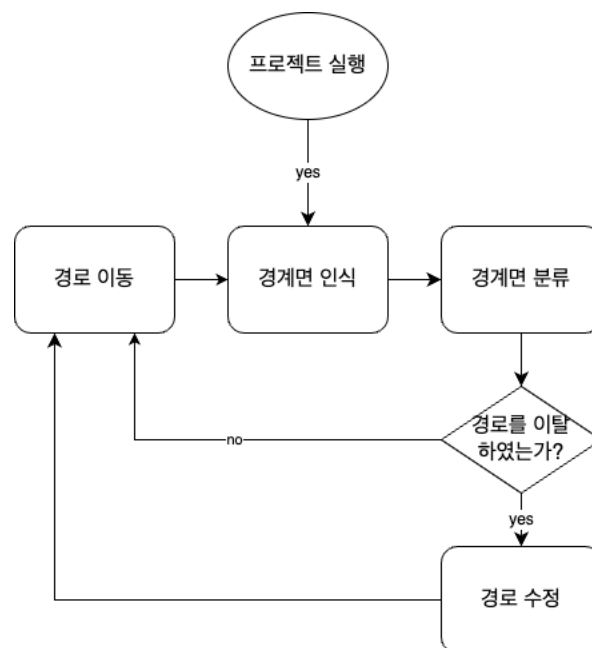
3.3.2 데이터 학습 순서도



3.3.3 장치 도색 순서도



3.3.4 장치 이동 순서도



4. 개발 일정 및 역할 분담

4.1 개발 일정

5월			6월					7월				8월					9월			
4주	5주		1주	2주	3주	4주	5주	1주	2주	3주	4주	1주	2주	3주	4주	5주	1주	2주	3주	4주
관련	기술	지식																		
학습																				
			데이터 수집 및 전처리																	
					Edge detection 모델 구 현 및 학습															
						경로 생성 알고리 즘 개발														
								중간보고서 작성												
									시뮬레이터 구현											
												안정성 및 성능 평가								
														문제점 파악 및 오류 수정						
																	최종보고서 작성 및 심사준비			

4.2 역할 분담

조원	역할
김정호	작업 경로 알고리즘 개발, Unity 시뮬레이터 개발
정제영	YOLOv5 로 도장면 분류 모델 개발
최성렬	OpenCV 로 도장면 영상 합성 알고리즘 개발
공통	필요 지식 습득, 데이터 수집, 보고서 작성, 발표, 장치 구현

5. 참고 문헌

1. 서다솜, 원진호, 양창주, 김국환, 권경도, 김경철, 홍영기, 류희석.(2021).『 자율주행 트랙터 경로 추종을 위한 영상 기반 경계검출기술 개발. 』한국통신학회논문지,46(11),2078-2087.
2. 김태하.(2022),『 객체 분할 정보를 활용한 객체 보존 스티칭 기법 연구』,서울과학기술대학교,1-43.
3. 김은경, 조현학, 김종근, 김성신.(2017).가우시안 분포 기반 제어 함수와 영상 밝기 정보를 이용한 명도 보정 기법.한국지능시스템학회 논문지,27(5),382-387.