벽면이동로봇에서의 머신러닝을 이용한 벽면 균열 검출에 관한 연구

박재민*, 김현섭*, 신동호*, 김상훈*
*한경대학교 전기전자제어공학과
e-mail : kimsh@hknu.ac.kr

A Study on Wall-Crack Detection Using Machine Learning in Wall-Climbing Robot

Jae-Min Park*, Hyun-Seop Kim*, Dong-Ho Shin*, Sang-Hun Kim*
*Department of Electrical, Electronic and Control Engineering, Hankyong National University

요 약

본 논문은 진공을 이용한 흡착방식과 바퀴형 이동방식을 사용하는 벽면이동로봇의 구성 및 벽면 균열 검출 알고리즘에 관한 연구로써, 카메라와 함께 임베디드 시스템을 구성하였으며 Convolutional Neural Network를 이용한 머신러닝 알고리즘을 통해 균열을 감지하고 검출된 균열의 영상과 위치정보를 서버(관리자 장치)로 전송하는 통신 환경을 구축하였다. 균열 검출 성능을 검증하기 위해 균열 데이터를 이용하여 실험하고 결과를 제시하였다

1. 서론

본 논문에서는 벽면 이동로봇에 고성능 Embedded Software Platform과 Webcam을 탑재한 지능 로봇의 머신러닝을 통한 벽면 균열 검출을 목표로 연구가 수행되었다. 과거뿐 아니라 현재에도 많은 벽면의 균열을 탐지하고 보수하는 작업이 사람에 의해 직접 수행되고 있지만, 대부분의 검사 및 보수작업에서 위험성이 따르고 효율성이 부족하기 때문에 이 점들을 고려하여 연구를 진행하였다.

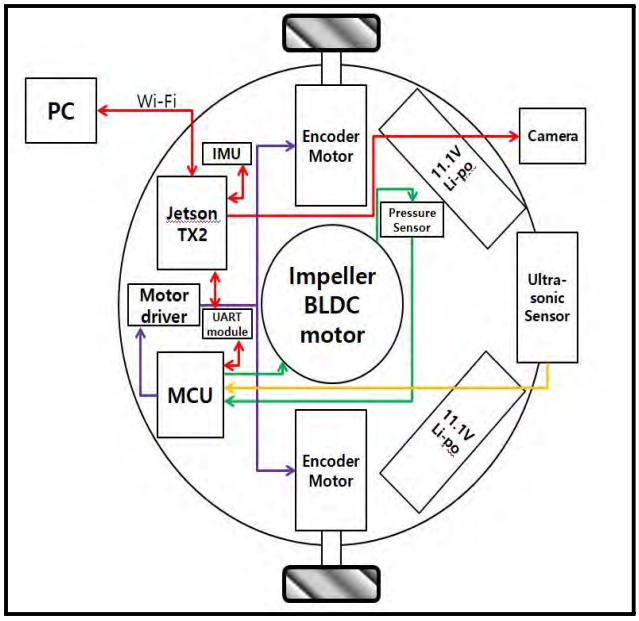
본 논문에서는 기존의 연구들에서 제안한 여러 벽면 부착방식 중, 진공 흡착 방식[4][8]을 선택하였다. 적재 하중에 유리하여 여러 부품을 탑재할 수 있고 한정된 재질의 벽에 부착하여 이동할 수 있다는 환경의존성 또한 개선할 수 있다는 이유로 선택하였다. 이동방식에 있어서는 위치인식과 기동성을 고려해 Encoder Motor를 이용한 바퀴이동방식을 채택하였다. 특히 본 논문은 머신러닝에 특화된 고성능 Embedded Software Platform인 Jetson TX2를 이용하여 영상처리 및 머신러닝을 로봇 내에서 구현하여 균열을 검출하고 이에 대한 위치와 영상을 서버로 전송하는 방법을 연구하였다.

현재 활발히 연구가 진행되는 드론(UAV)을 이용한 균열감지 기법[5]의 경우, 실내 환경 구동에 대한 위험성이 존재하고, 벽과의 일정거리를 유지해야한다는 점에서 드론은 지속적으로 Hovering을 하여야하지만 주변 환경 및 설계에 영향을 많이 받아 제작 과정에서 고도의 기술과 노력이 필요하다. 또한 벽에 근접하여 촬영할 수 없기 때문에 미세균열 검출에 있어 불리하다. 드론보다는 안전성에서 유리한 지면에서 이동하는 모바일로봇을 이용한 벽면 균열 감지 방법도 연구[6]되었지만 벽면높이에 따른 한계와 대형구조물과 해저면에 기반을 둔 대형 교량 등에선 구동할 수 없다. 그러나 벽면에 부착하여 이동하는 로봇의 경우, 드론과 다르게 벽면에 부착하여 이동하기 때문에 미세 균열 검출을 더 정밀하게 할 수 있고 실내 및 교량 등에서 구동이 가능하여 위험성이 적고 효율이 좋다. 특히 압력센서를 통한 Chamber 내부의 압력을 측정하여 임무 수행 불가 상황에서 스스로 작업을 중단하는 환경을 쉽게 구축할 수 있기 때문에 안전성면에서 더욱 유리하다.

2. 로봇의 구성

MCU와 Jetson TX2는 UART module을 통해 연결하여 위치 정보와 센서에서 얻은 정보로 주변 환경에 대하여 통신을 하도록 구성하고 Jetson TX2와 서버(관리자 장치)는 소켓 통신 기반 연결망(Wi-Fi, Wi-Fi Direct 등)으로 연결하여 관리자에게 필요한 정보를 전송하도록 하였다.

Jetson TX2는 머신러닝을 통해 카메라에서 균열을 감지하고 이에 대한 Image와 위치정보를 서버에 전송하며 IMU를 통해 위치를 추정한다. MCU는 초음파 센서를 통해 주행을 제어하고 Encoder Motor로 위치를 추정하여 Jetson TX2에 전송하도록 구성한다. '벽면 이동을 위한 지능형 로봇' 논문[10]을 참고하여 MCU에 BLDC모터와 압력센서 또한 연결하여 Chamber 내부의 압력을 측정한 뒤, 상황에 맞게 BLDC모터를 제어하여 흡착세기를 조절하도록 설계하였다. [Fig.1]은 로봇의 각 센서 및 부품에 대한 구성 및 데이터 흐름을 나타낸 그림이다.



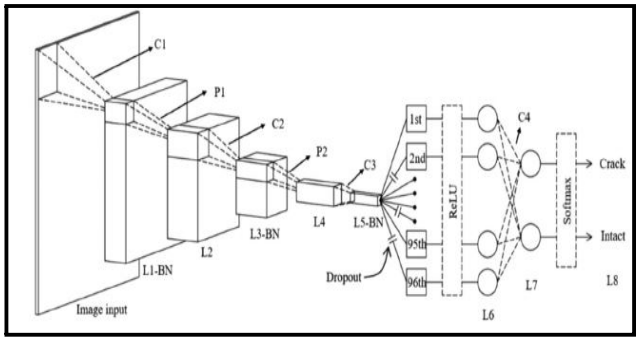
[Fig.1] Robot System Configuration Diagram

3. 균열 검출 구현

CNN(Convolutional Neural Network)를 이용한 머신러닝을 구현하기 위해 이에 최적화된 고성능 임베디드 플렛폼인 NVIDIA사의 'Jetson TX2'를 사용하였으며 부피 및 무게의 적합성을 고려하여 기존 NVIDIA사에서 제공하는 Develop Board를 이용하지 않고 ConnectTech사에서 제공하는 Custom Board인 Orbitty Carrier Board를 이용하였다. 로봇의 본체에 Webcam을 벽면과 마주보는 상태로 설치하고 Jetson TX2와 연결하여 한 벽면의 모든 부분을 돌아다니며 균열 검출 임무를 수행한다.

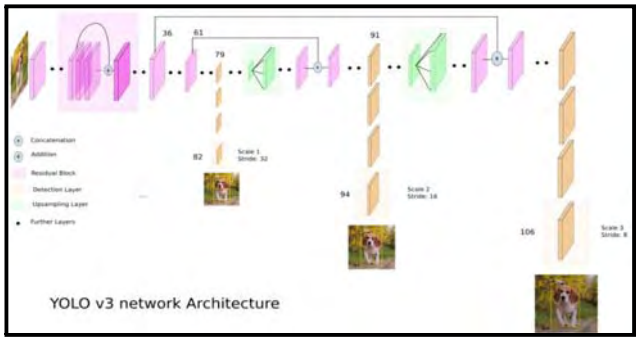
본 논문에서는 Young-Jin Cha & Wooram Choi의 'Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks'의 연구[1]를 인용하여 구현한 것과 'YOLO : Real time Object Detection'[2](이하 YOLO)을 사용하여 Wall Crack Detection을 구현하였다.

위에서 언급한 'Young-Jin Cha & Wooram Choi의 알고리즘'은 Tensorflow Framework를 바탕으로 구성되어 있으며 CNN Architecture는 [Fig.2]와 같이 Test Image를 Convolution layer, Polling layer, ReLU layer, softmax layer로 이루어진 총 8개의 layer를 거쳐 최종적으로 두 개의 Class(Crack, Intact)로 출력되어 Crack 존재 여부를 확인한다고 한다. 본 논문에서는, 벽면 이동로봇에서 이를 구현하기 위해 입출력 부분을 다음과 같이 수정하였다. Jetson TX2는 Webcam에서 출력되는 화면을 30Frames마다 Capture하고 벽면을 왼쪽부터 차례대로 탐색하며 이동한다. 시간이 지나 벽면의 모든 부분을 탐색했다고 판단되면 벽면을 이동하며 Capture한 Image를 [Fig.2]의 구성을 통해 분석한다. 최종적으로, 균열이 있다고 판단되는 Image를 관리자 장치에 전송한다.



[Fig.2] CNN architecture of Algorithm by Young-Jin Cha & Wooram Choi

YOLO는 실시간 객체탐지에 적합한 Object Detection 도구이며 수정을 거치지 않은 기본적인 YOLO v3의 Network 구성은 [Fig.3]과 같다. 본 논문에서는, AlexeyAB의 Yolo_Mark 프로그램에 함께 구성되어있는 yolo-obj.cfg 파일의 CNN[3]구성을 이용하여 구현하였다. 먼저, 이를 Crack Detection에 적용하기 위해선 Training에 쓰일 Crack Image와 더불어 각 Image마다 Bounding Box의 좌표가 입력된 Text파일을 구성해야 한다. 위와 같이 Dataset을 구성하고 학습을 진행하여 weights파일을 생성한 후, Jetson TX2에 Webcam을 연결하고 YOLO를 실행하면 해당 weights파일을 통해 Webcam 화면에서의 Crack을 실시간으로 검출하게 된다. 최종적으로 관리자에게 전송할 Image는 Crack이 Image의 중앙에 위치하도록 하기위해 로봇의 이동속도를 고려하여 최초 검출 후 5 Frames 뒤에 Frame을 Image파일로 Capture한 후 전송하도록 구성하였다.



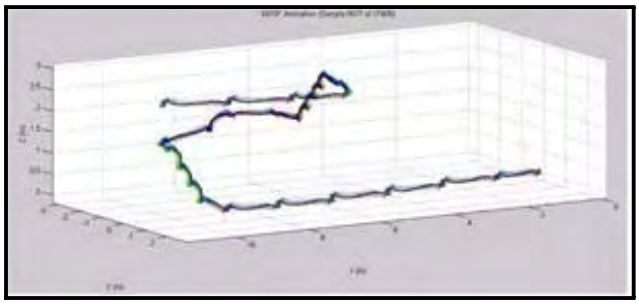
[Fig.3] YOLO v3 network Architecture

4. 균열 위치 전송

균열의 위치는 Encoder Motor와 IMU를 이용하여 초기 위치에 대한 상대적인 로봇의 위치를 통해 추정[9]한다. 로봇의 위치는 Encoder Motor의 Hall Sensor의 Pulse를 이용하여 이동거리를 계산하고 IMU에서 얻은 이동거리를 Kalman filter를 통해 오차를 보정한다. 이 두 센서에서 얻은 값을 비교한 뒤 최종 이동거리의 오차를 최소화한다.

처음 로봇이 출발한 지점을 벽면의 왼쪽 하단으로 지정하고 그로부터 이동한 거리를 3D Tracking하여 Graph로 표현한다. 1초마다 그 지점을 지나간 시간을 기록하고

Webcam을 Capture한 Image의 생성 시간과 비교하면 어느 지점에서 균열이 발생했는지 알 수 있다. (30frames/sec를 이용한다.) [Fig.4]은 x-io Technologies Limited사의 IMU를 이용하여 실시간 3D Tracking을 구현[7]한 3D Graph 출력 영상 중 일부이다.



[Fig.4] 3D Tracking with IMU

5. 실험 및 분석

본 논문에서 학습에 사용된 Crack Image Dataset 총 60,412장의 Image로 구성되어 있다. Crack Image는 총 11,704장이고 Training Accuracy 및 AP측정에 사용되는 Valid Image는 156장이다. Intact로 구분되는 Non-crack Image는 총 48,708장이고 이 중 Valid Image는 75장이다. 이 때 YOLO는 1개의 Class(Crack)만 검출하도록 설계되어 있어 Non-crack Image를 제외한 11,704장의 Crack Image만 학습 및 AP분석에 이용되었다. [Table.1]은 알고리즘 학습 및 성능분석에 이용된 Dataset의 구성을 정리한 표이다.

Crack	Training image	11,548장
	Valid image	156장
Intact (Non-crack)	Training image	48,633장
	Valid image	75장

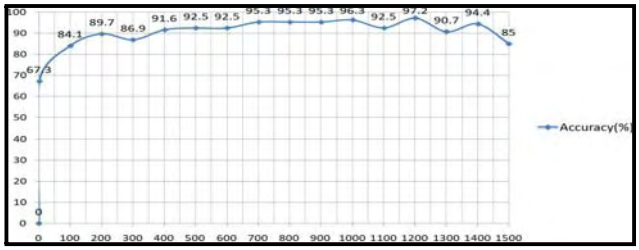
[Table.1] Desktop Specification

구동은 Jetson TX2에서 진행하였고 Jetson Tx2의 OS는 Ubuntu 16.04를 설치하였다. Webcam은 1280*720의 HD 해상도를 가진 'ODROID USB-CAM 720P'를 사용하였으며 한경대학교 제1공학관 407-2호 앞 복도의 벽면 균열을 대상으로 실험을 진행하였다. Dataset Training에 사용된 Desktop의 사양은 [Table.2]과 같다.

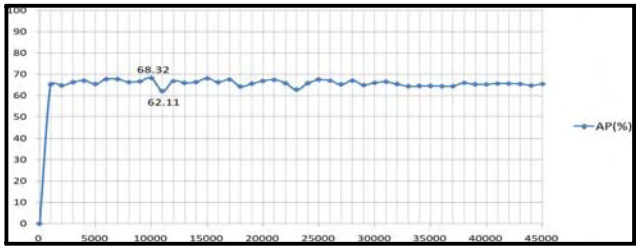
CPU	Intel Core i7-8700K CPU @ 3.7GHz
GPU	NVIDIA TITAN Xp
Memory	32GB
OS	Ubuntu 16.04.5 LTS

[Table.2] Desktop Specification

[Fig.5]와 [Fig.6]는 각 알고리즘을 Training하면서 Valid Images로 측정한 Training 횟수에 따른 성능분석을 나타낸 그래프이다.



[Fig.5] Accuracy of Algorithm by Young-Jin Cha & Wooram Choi

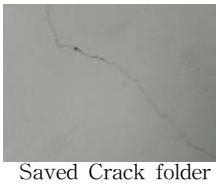
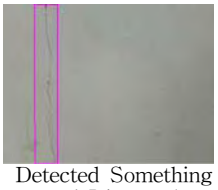


[Fig.6] AP(Average Precision) of the YOLO

학습이 종료된 후, Webcam의 영상을 이용하여 실험을 진행하였다. 'Young-Jin Cha & Wooram Choi'의 알고리즘은 30Frames당 한번씩 Webcam의 영상을 Image로 저장하여 연산하였고 YOLO 알고리즘은 실시간 탐지로 6.0전후의 FPS를 보였다.

Young-Jin Cha & Wooram Choi의 알고리즘의 경우, Input Image를 Crack으로 판단하는 경우엔 Crack폴더에 Image를 저장하고 Crack으로 판단되지 않는 Image의 경우엔 Non_Crack이미지 폴더에 저장되게 재구성하였으나 Crack에 대한 영역 검출은 할 수 없다. YOLO는 실시간으로 Crack의 영역만 잡아내도록 구성되어 있어 Crack 영역이 감지된 경우에만 Crack폴더에 해당 Frame의 5Frames 뒤의 Frame을 저장하도록 재구성하였다.

[Table.3]는 위의 학습된 데이터를 가지고 각각 실제 벽면을 촬영한 영상을 통해 실험을 진행한 결과이다.

	Algorithm by Cha & Choi	YOLO
검출 성공	 Saved Crack folder and It's true image	 Detected Something and It's crack
검출 실패	Saved Crack folder but It isn't true image	Detected Something but It isn't crack

[Table.3] Result of Crack Detection

결과적으로 두 알고리즘 모두 눈에 잘 띄는 두꺼운 Crack은 잘 검출되었고 아주 얇은 Crack도 두꺼운 Crack만큼은 아니지만 대체적으로 잘 검출되었다. YOLO로 구현한 Crack Detection은 벽에 존재하는 낙서나 이물질도 Crack으로 분류하였다. 그러나 모든 낙서나 이물질 등을 Crack으로 분류하진 않았고 영상의 밝기나 객체의 형태나 크기에 영향을 받아 특정 객체만 오검출 되었다. 'Young-Jin Cha & Wooram Choi의 알고리즘'은 Crack이 포함되지 않은 Image를 Crack으로 분류하는 결과가 다소 발생 하였다.

6. 결론 및 향후 연구

일반적인 Object Detection과 다르게, Crack은 일정한 형태가 없이 어떤 부분이 손상되었다는 것을 특징으로 분류되는 객체이기 때문에 성능 개선을 위해 이번 실험에서 사용한 Dataset보다 품질이 좋아야하고 Vision에만 의존하여 Crack을 Detection하기 보단 센서 등을 함께 이용하는 것이 더 좋은 결과를 얻을 수 있을 것 같다는 결론을 내렸다.

균열을 검출하는 벽면이동로봇의 목적은 향후 시설물에 대해 위험성이 있어 보수가 필요한 Crack을 찾는 것이기 때문에 낙서나 이물질을 Crack으로 분류하는 오검출이 발생하더라도 많은 Crack이 될 수 있는 객체들의 후보를 검출하는 것이 로봇의 임무수행 목적에 더 적합하다고 생각한다. 이러한 점들을 고려하여 향후에 Crack Detection을 위한 벽면이동로봇의 목적에 적합한 Model 성능지표에 대해 연구해서 그에 맞는 성능지표로 두 알고리즘을 비교할 필요가 있다.

참고문헌

[1] Young-Jin Cha & Wooram Choi, "Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks", 2017.

[2] Redmon, Joseph and Farhadi, Ali , "YOLOv3: An Incremental Improvement", 2018.

[3] AlexeyAB. (2016). "GUI for marking bounded boxes of objects in images for training neural network Yolo v3 and v2". Available at: https://github.com/AlexeyAB/Yolo_mark (Accessed: 1 April 2019).

[4] William Morris, Jizhong Xiao, "City-Climber: Development of a Novel Wall-Climbing Robot", 2008.

[5] Jong-Woo Kim, Sung-Bae Kim, Jeong-Cheon Park, Jin-Won Nam, "Development of Crack Detection System with Unmanned Aerial Vehicles and Digital Image Processing", 2015.

[6] 안성열, "An inspection system for detection of cracks on the concrete structures using a mobile robot", 2015.

[7] Sebastian O. H. Madgwick, Andrew J. L. Harrison, Ravi Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm", 2011.

[8] 구익모, 송영국, 문형필, 박선규, 최혁렬, "임펠러를 이용한 벽면이동로봇의 설계 및 제어", 로봇학회 논문지 제5권 제3호, 2010.

[9] 김경환, 박계도, 조재한, 이장명,. "Encoder와 IMU 센서를 이용한 이동로봇의 실외 위치 추정에 대한 연구", 제어로봇시스템학회 각 지부별 자료집, 9-12, 2011.

[10] 은영민, 김민성, 오세엽, 신선웅, 김상훈, "벽면 이동을 위한 지능형 로봇", 정보 및 제어 논문집, 442-443, 2012.