

Digital  
Image  
Process  
Report #1



Survey related with DIP

로봇기계공학과  
21721160  
최병희

# 목차

1. 현재 영상 데이터 처리의 활용분야
  - Deep Learning
  - 자율주행 자동차
  - 실내 물류 로봇(모바일 로봇)
  - 3D 모델링 구축
  - 파노라마 이미지 생성
2. 그 분야에 활용되는 영상처리 기술
  - Convolutional Nerual Network
  - Object Detection
  - Descriptor Matching
3. 영상처리에 사용되는 프로그래밍 언어
  - C/C++
  - Python
  - MATLAB
4. 자료 출처

## 1. 현재 영상 데이터 처리의 활용분야

영상 처리 기술은 컴퓨터가 시각적인 데이터를 분석하기 위해 수많은 데이터 처리 방법이 개발되어 왔습니다. 그 중 저는 아래 5가지 영상처리 응용 분야에 대해서 정리해 보고자 합니다.

- Deep Learning

4차 산업 혁명과 지난 2017년 알파고의 등장으로 부상한 키워드인 Deep Learning입니다. Deep Learning은 다량의 데이터를 기반으로 모델을 학습시키는 기법인데, 이미지 처리 뿐만 아니라 학습시킬 데이터의 종류에 따라 수많은 학습 방법이 있습니다.

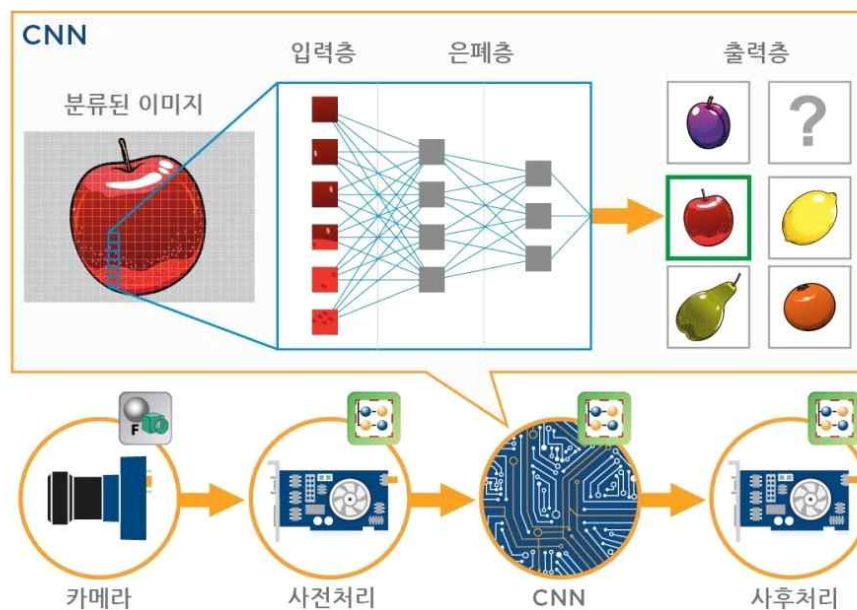


그림 2 이미지 딥러닝 과정의 예시

Deep Learning은 어떤 분야에 이미지 처리를 직접적으로 활용한다기보단, 이미지 처리 기법을 사용하여 Deep Learning Nerual Network에 입력할 데이터를 가공하고, 이 데이터를 학습시킨 모델을 특정 분야에 적용하는 방식입니다. 위의 그림 2는 이미지 데이터를 통해 데이터의 종류를 분류하는 방법의 대표적인 예시인 CNN입니다. 이 Network에 대해서는 2-(가)에서 조금 더 자세히 서술하겠습니다.

이 이미지 처리 기반 Deep Learning 기술을 활용하여 대표적으로는 물체 탐지 (Object Detection)이 있습니다. 물체 탐지는 카메라로부터 받아온 이미지를 처리하여 학습된 모델을 바탕으로 인식된 물체를 탐지하는 것입니다. CNN 같은 모델은 분류를 통해 컴퓨터는 이 물체가 사람인지, 다른 어떤 물체인지 인식할 수 있습니다.

또한 최근 유튜브 알고리즘에 많이 올라오는 딥페이크 기술 역시 GAN이라는

Learning 알고리즘을 통해 만들어 졌습니다. 원래 Deep Network를 사용하는 Network는 아니지만 이미지를 기반으로 학습한다는 점에서 이미지 처리로 볼 수 있다고 생각하여 서술했습니다.

- 자율주행 자동차

두 번째 활용사례는 자율주행 자동차입니다. 자율주행 자동차 역시 최근 떠오르고 있는 키워드입니다. 대표적으로 테슬라의 여러 자율주행 차량이나 구글의 웨이모 같은 차량에 이미지 처리 기법이 적용되었습니다. 제 수준에선 이해할 수 없는 엄청나게 복잡한 기술들이 적용되었을 테지만, 대표적으로 차선 검출이나 물체 탐지 같은 기술이 포함되어 있을 것입니다.

차량의 자율주행을 위해서는 첫 번째로 차선을 인식하여 그에 따라 움직이는 기능이 필수입니다. 직선 차선을 따라 주행하고, 곡선 차선에서는 그에 맞게 좌/우회전 실행하는 기능입니다. 사람은 눈으로 차선을 보고 따라가는 행위를 자연스럽게 하고 있지만, 컴퓨터는 이미지에서 차선을 검출, 그에 맞는 속도로 운행하기까지의 과정을 모두 이미지 처리 기법을 사용하여 알고리즘으로 넣어줘야 카메라로 차선을 보고 주행하는 행위가 완성될 수 있습니다.

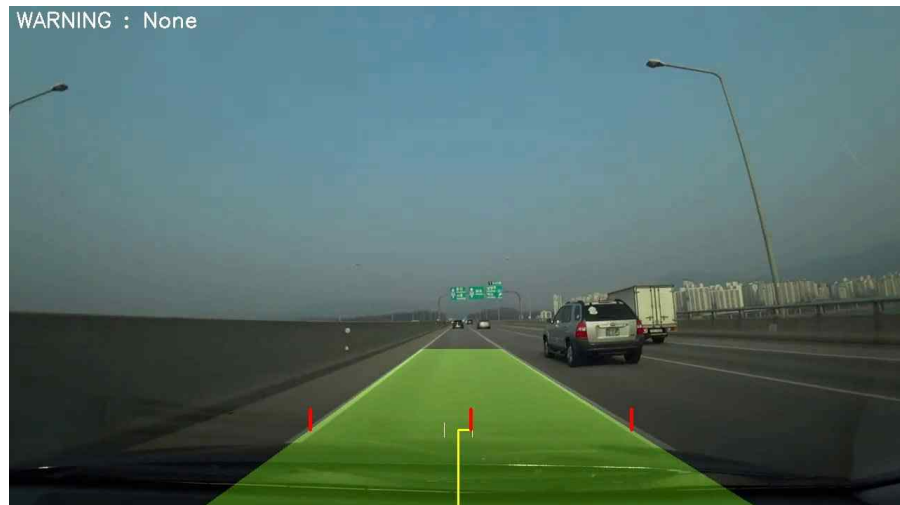


그림 3 이미지(영상)로부터 차선을 인식하는 모습

위 그림 3은 자동차가 주행하면서 수집한 데이터에서 차선을 검출한 모습입니다. 이러한 방식으로 자율주행 자동차는 차선을 검출합니다.

두 번째로는 물체 탐지입니다. 물체를 탐지할 때 Deep Learning 기반 기법을 사용하기도 하지만 그 이전에 feature detection 방법을 사용하기도 합니다. Deep Learning 기반 방법은 학습에 필요한 데이터가 많이 필요하기도 하고, 연산량도 많이 때문에 임베디드 시스템에 넣기는 조금 불리한 편입니다. 따라서 feature detection 방법을 사용하여 넣는 방법도 있습니다. 자동차가 자율주행을 할 때, 사람을 인식하여 멈추거나 피하는 기능 뿐만 아니라 도로교통 표지판, 또는 신호등을 인식해야 할 필요가 있으므로 물체 탐지 기능이 필요하다고 할 수 있습니다.

- 실내 물류 로봇(모바일 로봇)

세 번째로 이미지 처리 기법을 실내 물류 로봇에 적용한 사례가 있습니다. 실내 물류 로봇은 주로 창고 내의 물류들을 분류/운반하는 기능을 하는데, 실내 자율주행 기능을 수행할 때 마커를 인식하여 자신의 위치를 추정하거나 해당 마커에 match된 기능을 수행할 수 있습니다.



그림 5 마커를 사용하여 해당 기능을 수행하는 실내 물류 로봇

이때 마커를 인식하기 위해 여러 이미지 처리 기법을 사용할 수 있습니다. 정확하게 사용되는 방법은 알 수 없지만, corner를 검출하는 feature detection 방법을 사용한 것으로 추측하고 있습니다.

또한 로봇이 실내에서 자율주행 할 때 필요한 Map을 작성하기 위한 방법인 VisualSLAM에도 이미지 처리 기법을 사용할 수 있습니다. 카메라로 데이터를 수집하여 이미지의 feature와 odometry를 검출한 후 이를 바탕으로 Map을 작성하는 것입니다. 이 작성된 Map을 바탕으로 이후 이 로봇은 Map이 작성된 공간에서 자율주행을 할 수 있습니다.

- 3D 모델링 구축

이미지의 기하학적 정보를 이용하여 이미지를 3D 상으로 표현하는 방법입니다. 이미지에서 같은 평면을 구하고, 사람이나 물체와의 거리를 기하학적인 방법으로 추측함으로써 2D 이미지의 요소들을 입체적으로 표현할 수 있습니다. 아래 그림 6에서의 명화를 3차원 모델로 구현한 것이 좋은 예시라고 할 수 있습니다.

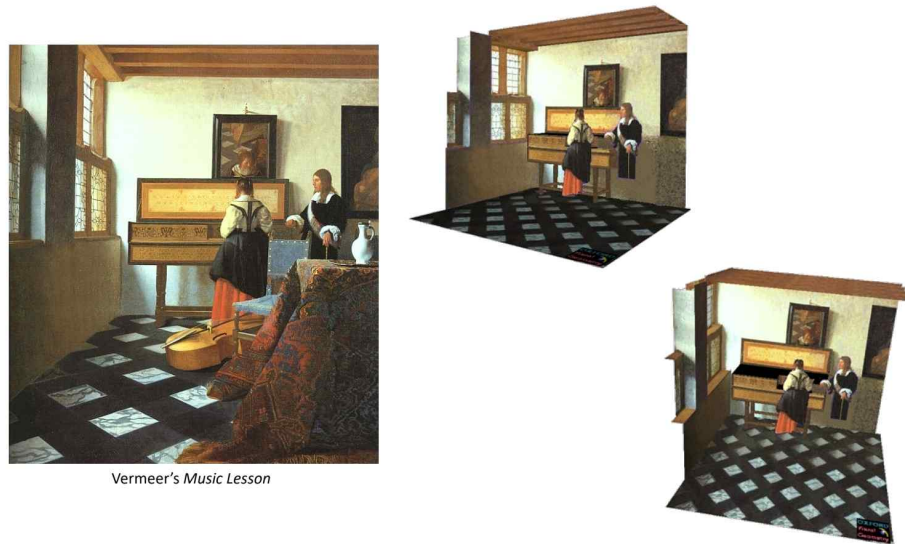


그림 6 projective geometry를 사용하여 명화를 3차원상에서 복구한 모델

- 파노라마 이미지 생성

몇 년 전부터 스마트폰의 기본 카메라에서도 볼 수 있는 파노라마 이미지 생성 기능입니다. 주로 주변의 360도에 가까운 풍경을 촬영하기 위해 사용됩니다. 이미지들 사이의 공통적인 특징을 찾아서 사진끼리 이어 붙이는 방식의 기술을 사용합니다.

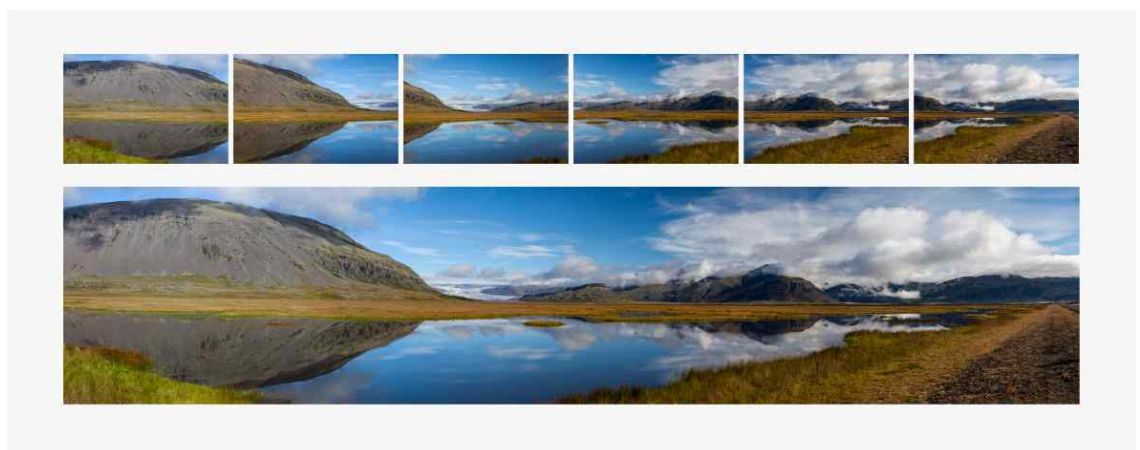


그림 7 파노라마 이미지 생성 과정



## 2. 그 분야에 활용되는 영상 처리 기술

- Convolutional Neural Network(CNN)

Deep Learning 이미지 처리 분야에 많이 사용되는 합성곱 신경망(Convolutional Neural Network)입니다. 1번 항목의 첫 번째 문단의 그림 2에서 볼 수 있듯이, 이미지의 정보 손실·연산량 최소화를 위한 사전 처리를 한 후 이미지에 필터를 사용한 합성곱 연산을 통해 feature map을 작성하고, 작성된 feature map을 바탕으로 신경망을 계속해서 거쳐 학습합니다.

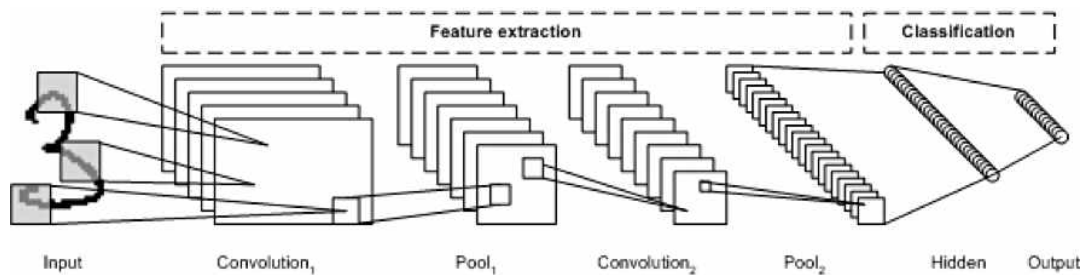


그림 8 LeNet-5 네트워크

위 그림 8은 가장 기초적인 형태의 CNN인 LeNet-5 네트워크입니다. Convolution layer들에서 학습 후 classification layer들에서 softmax 분류를 통해 input된 이미지가 어떤 이미지인지를 분류합니다.

- Object Detection

두 번째는 1번 항목에서 많이 언급되었던 물체 탐지입니다. 물체 탐지는 사람이나 사물 등의 객체를 탐지하는 기능으로 딥러닝을 통한 방법, 물체의 특징을 통해 탐지하는 방법 등이 있습니다.

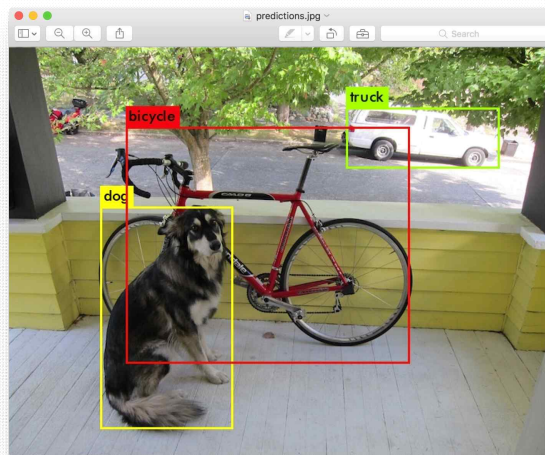


그림 9 Yolo Object Detection의 결과



먼저 Deep Learning 방법을 사용한 Object Detection인 Yolo입니다. Yolo는 You look only once의 줄임말로, 학습 데이터를 생성하기 위해 이미지에 대한 처리를 단 한번만 해도 되는 것이 특징입니다. 학습할 데이터와 알고리즘이 판단한 일치 정도가 높은 가중치를 향해 계속해서 가중치를 갱신합니다. 이미지를 한번 훑은 후 그 정보를 바탕으로 딥러닝을 진행하는 느낌입니다.

다른 물체 탐지 방법으로는 물체의 특징을 기반으로 검출하는 방법이 있습니다. 그 중 한 방법은 Histogram of Gradient(HoG)입니다.

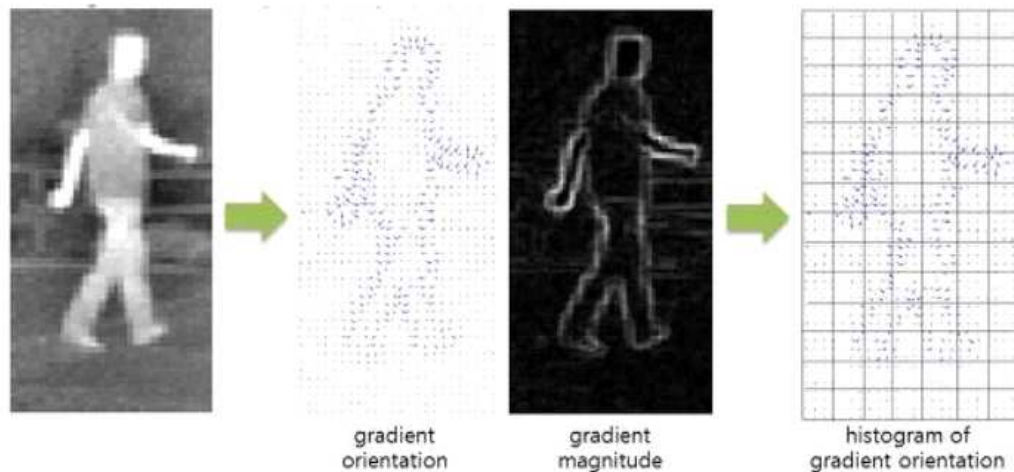


그림 10 HoG Object Detection

사람의 실루엣처럼 독특한 edge를 가진 경우 주위 배경과의 gradient 패턴이 일정한 경우가 많습니다. 이러한 점을 이용하여 이미지상에서의 gradient를 계산하여 이미지의 어느 부분이 사람인지를 판단할 수 있는 방법이 바로 HoG입니다.

### ● Descriptor Matching

Descriptor Matching은 파노라마 이미지 같은 사례에 사용되는 이미지와 이미지 간의 일치성을 조사하는 영상 처리 기술입니다. corner같은 특징적인 feature를 찾아서 이미지의 어떤 부분이 일치하는지를 알 수 있습니다. 알고리즘은 대표적으로 SIFT, SURF, ORB 등이 있습니다.

## 3. 영상처리에 사용되는 프로그래밍 언어

제가 조사한 영상처리에 사용되는 프로그래밍 언어는 C/C++, Python, MATLAB입니다. 하지만 대부분의 언어는 OpenCV 라이브러리를 사용하기 때문에 프로그래밍 언어 간의 차이가 덜합니다. 하지만 언어별로 사용되는 플랫폼이 다르기 때문에 그 차이점을 정리하도록 하겠습니다.

### ● C/C++

C/C++ 언어는 다른 언어에 비해 기계어에 좀 더 가까운 Low level 언어이기 때

문에 최대한 데이터 처리량이 적어야 하는 임베디드 시스템에서 사용하기 좋습니다. 하지만 Python에 비해 개발 난이도가 높고, 이미지 처리의 중요한 분야인 Deep Learning에 대한 라이브러리가 Python에 비해 조금 부족하다는 단점이 있습니다.

이미지 처리를 위해서는 행렬 연산 기능이 구현되어 있어야 하는데, C++에는 Eigen Library를 사용하여 행렬 연산을 할 수 있습니다.

- Python

Python은 배우기가 가장 쉽고 라이브러리가 잘 구현되어 있는 특징을 갖고 있어 입문하기에 아주 편합니다. 행렬 연산 역시 numpy 라이브러리를 통해 가능하고, Deep Learning 라이브러리 역시 Tensorflow, Keras 등 많은 API를 지원합니다. 단점으로는 C++보다 High Level 언어라서 처리하는데 시간이 조금 더 걸리거나 데이터 처리량이 조금 늘 수는 있지만, 최근 프로세서의 발달로 임베디드 시스템에도 많이 사용되는 모습을 볼 수 있습니다.

- MATLAB

MATLAB은 알고리즘보다 계산에 특화되어 있는 프로그래밍 언어로서 행렬 연산에 특히 편리한 모습을 보여줍니다. 또한 Addon 기능을 사용하여 Deep Learning Toolbox, Image Process Toolbox 등 다양한 라이브러리를 손쉽게 다운로드하여 사용할 수 있습니다. 하지만 개발환경이 MATLAB 프로그램에 한정되어 있기 때문에 오프라인 환경이나 라이선스가 없는 경우 사용하기 힘들다는 단점을 갖고 있습니다.

#### 4. 자료 출처

그림 2: BASLER, FPGA 프레임 그래버 기반 딥 러닝,

<https://www.baslerweb.com/ko/sales-support/knowledge-base/deep-learning/frame-grabber-based-deep-learning/>

그림 3: kiwan, OpenCV를 이용한 고속도로 차선검출,

<https://www.youtube.com/watch?v=aln5Q2e1he8>

그림 4: 로봇신문, DB첼커, 체코 물류창고에 자율이동로봇(AMR) 도입 운영,

<https://www.irobotnews.com/news/articleView.html?idxno=22854>

그림 6: Computer Vision ppt

그림 7: Adobe, 파노라마 만들기, 2021.04.29.,

<https://helpx.adobe.com/kr/camera-raw/using/create-panoramas.html>

그림 8: Yann Lecun, Gradient-based learning applied to document recognition, December 1998, p.7

그림 9: yolo, YOLO: Real-Time Object Detection

<https://pjreddie.com/darknet/yolo/>

그림 10: HDLY, OpenCV, Object Detection - HOG, 사람 검출,

<https://m.blog.naver.com/handuelly/221833630061>