

**OpenCV**

# OpenCV 개요

강사 양석환



# 영상 처리와 컴퓨터 비전, 그리고 AI



## • 인간의 시각 정보 처리

- 인간(동물 포함)은 복잡한 신경체계를 통해 다양한 정보를 수집, 처리하면서 생존함
- 인간이 수집하는 정보의 가장 많은 부분은 시각 정보(영상 정보) → AI의 많은 응용이 시각 처리와 연관됨
  - 눈으로 영상 정보를 획득하고 뇌를 이용하여 시각 정보를 처리함

## • 컴퓨터의 시각 정보 처리

- 비전 관련 자동화 기기 및 기술
  - “눈 → 카메라, 뇌 → 컴퓨터”를 사용하여 시각 정보 처리
- 관련 기술
  - 영상 처리, 컴퓨터 비전, 인공지능

인공지능 기술은 수많은 영역으로 구성되어 있으며  
그 중에서 컴퓨터 비전과 관련된 분야는  
패턴인식, 머신러닝, 딥러닝, 로봇틱스 등 많은 분야가 있음

## • 영상 처리 (Image Processing)

- 컴퓨터를 사용하여 영상 데이터를 다루는 분야
- 영상 처리 기술의 응용 예
  - 영상에 포함된 잡음(Noise) 제거
  - 영상의 화질 개선
  - 영상 내에서의 관심 영역(Region of Interest, Roi) 검출
  - 영상 분류(Image Classification)
  - 영상 인식, 객체 인식, 객체 검출(Image Object Detection)
  - 영상 영역 분할(Image Segmentation)
  - 영상 검색(Image Retrieval) 등

의료 영상 처리(Medical Image Processing)  
위성 영상 처리(Satellite Image Processing)  
등과 같이 분야를 명시하여 사용하기도 함

- **컴퓨터 비전 (Computer Vision)**

- 카메라에 의해 캡처 된 영상 프레임에서 의미 있는 정보를 추출하는 분야. 주로 실시간 응용을 다룸
- 사람은 눈을 통해 많은 정보를 인식 → 컴퓨터에게도 사람과 같은 시각처리 능력을 부여하고자 하는 기술
- 컴퓨터 비전 기술의 응용 예
  - 제품 결함 검사(Industrial Inspection)
  - 문자 인식(Character Recognition)
  - 얼굴 인식(Face Recognition)
  - 지문 인식(Fingerprint Recognition)
  - 물체 검출(Object Detection)
  - 물체 추적(Object Tracking) 등



## • 인공지능 (Artificial Intelligence)

- 언어, 인지, 학습 등의 지능적인 처리를 대신할 수 있는 컴퓨터와 관련된 모든 기술
- 자연어 처리, 음성인식, 전문가 시스템, 인공신경망, 퍼지로지, 로봇틱스, 컴퓨터 비전, 패턴인식, 머신러닝, 딥러닝 등 다양한 분야를 포함함
  - 머신 러닝 (Machine Learning)
    - 인공지능 기술 중에서 학습 알고리즘과 관련된 분야
    - 주로 지도학습(회귀, 분류 등), 비지도학습(군집화 등), 강화학습(Q-Learning, DQN 등)으로 구분함
  - 딥러닝 (Deep Learning)
    - 머신 러닝 기술 중에서 심층 신경망(Deep Neural Network, DNN)을 이용한 학습 알고리즘과 관련된 분야
    - DNN(ANN, MLP), CNN, YOLO, RNN 등 다양한 모델이 있음

## • 컴퓨터 비전이 어려운 이유 1

### • 인간과 컴퓨터가 영상을 인식하는 방법의 차이가 원인

- 인간과 달리 컴퓨터는 영상을 숫자(정확하게는 0/1)로 인식함
- 컴퓨터가 인식할 수 있는 것은 전기의 On/Off 상태 뿐임
  - 사람의 이해를 돕기 위하여 On/Off  $\rightarrow$  1/0(2진수)의 형태로 변경하여 표시
  - 2진수  $\rightarrow$  10진수 또는 16진수로 변경하고 각 숫자 코드에 다양한 명령어(Command)를 매칭 시켜서 프로그래밍
  - 따라서 컴퓨터는 입력되는 모든 신호를 숫자의 배열로 인식함
- 따라서 인간의 정보처리체계가 수행하는 일을 모두 분석하여 수치로 된 데이터 구조에 적용할 수 있도록 수학적으로 분석하고 알고리즘화 한 후 프로그래밍의 과정을 거쳐야 함
  - 그러나 인류는 아직 인간의 정보처리체계를 완전히 이해하지 못하고 있다는 것이 결정적인 원인임

사람에게 쉬운 일은 컴퓨터에게 어려울 수 있다.  
꼭 영상이 아니더라도...





## • 컴퓨터 비전이 어려운 이유 2

- 인간은 모든 신경체계가 유기적으로 연결된 거의 완성형에 가까운 구조이지만 컴퓨터는 아직 미완성
  - 예를 들면
    - 눈은 빛을 자연스럽게 조절하며, 초점을 자동으로 조절하지만(SW에 해당하는 인식 처리까지 모두 연동되어 있음)
    - 컴퓨터+ 카메라는 이와 관련된 SW와 HW를 일일이 제어함으로써 맞춰주어야 함

## • 해결 방안은 없는가?

- 다양한 영상 정보 처리 과정을 구현, 지원하는 라이브러리의 활용 → OpenCV 등
- 그래도 부족한 부분은 어떻게 보완할까?
  - 인간이 아직 모르는 부분들에 대한 해결, 개선을 그냥 AI에 맡겨 버리자!!! → 딥러닝 프레임워크가 주목받게 됨

# OpenCV 개요



**AiDA**  
Lab.

- **OpenCV(Open Source Computer Vision Library)**

- 오픈소스로 개발되고 있는 컴퓨터 비전 및 머신 러닝 라이브러리

OpenCV의 적용 라이선스

버전 4.4.0 이하: BCD / 버전 4.5.0 이상: Apache2  
(모두 상업적 이용이 허용되며 소스 공개의 의무가 없는 라이선스임)

- **2500개 이상의 알고리즘으로 구성됨**

- 영상 처리, 컴퓨터 비전과 관련된 전통적인 알고리즘
  - 얼굴 검출과 인식, 객체 인식, 객체 3D 모델 추출, 스테레오 카메라에서 3D 좌표 생성
  - 고해상도 영상 생성을 위한 이미지 스티칭, 영상 검색, 적목 현상 제거, 안구 운동 추적
- 최근에는 딥러닝과 관련된 기능도 제공하기 시작함
- 처음부터 실시간 처리를 고려하여 만들어졌기 때문에 다양한 하드웨어 플랫폼에서 매우 빠르게 작동함
- C, C++, Python, Java, Matlab 인터페이스, CUDA와 OpenCL 인터페이스 제공

- **OpenCV(Open Source Computer Vision Library)**

- **OpenCV의 사용 범위**

- 자율주행 자동차, 로봇틱스 분야의 영상 인식 시스템
- 스마트팩토리 등 공장에서의 제품 검사 및 영상 기반 관리 시스템
- 의료 영상 처리 및 보정, 판단 시스템
- 스마트폰, 카메라, CCTV 등
- **카메라로 찍어서 할 수 있는 모든 일**

**OpenCV 사용 기업들**

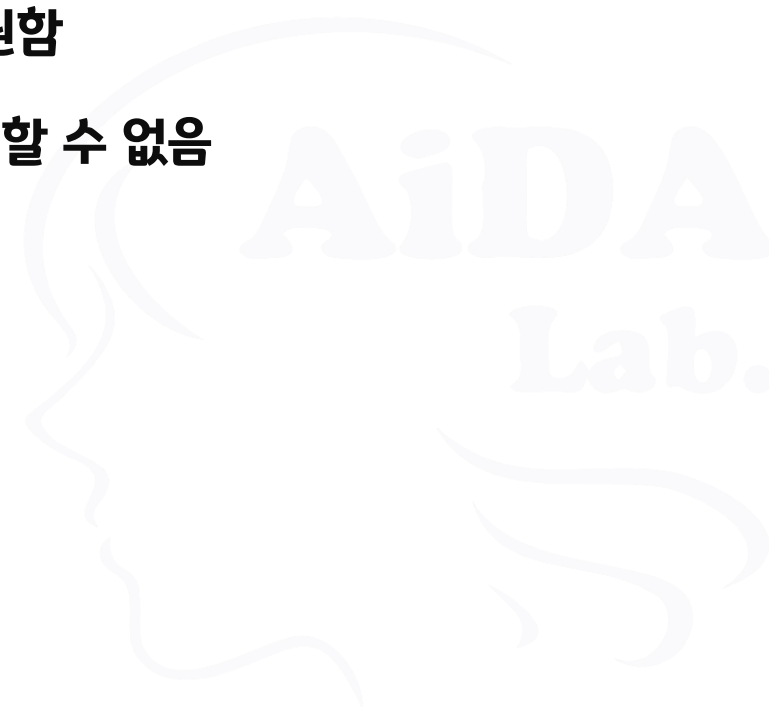
구글, 야후, 마이크로소프트, 인텔, IBM, 소니, 혼다, 도요다와 같은 대기업부터  
Applied Minds, Videosurf 및 Zeitera와 같은 신생 기업들까지 폭넓게 사용 중

- 1999년, 인텔에서 개발된 IPL(Image Primitive Library)을 기반으로 개발 시작
- 2000년, 일반에 공개되어 오픈 소스로서 개발 진행
- 2006년, OpenCV 1.0 버전 정식 배포
  - C 언어를 기반으로 구현
  - 많은 컴퓨터 비전 알고리즘이 주로 구조체와 함수로 구현됨
  - 영상 데이터는 IplImage라는 이름의 구조체를 이용하여 표현

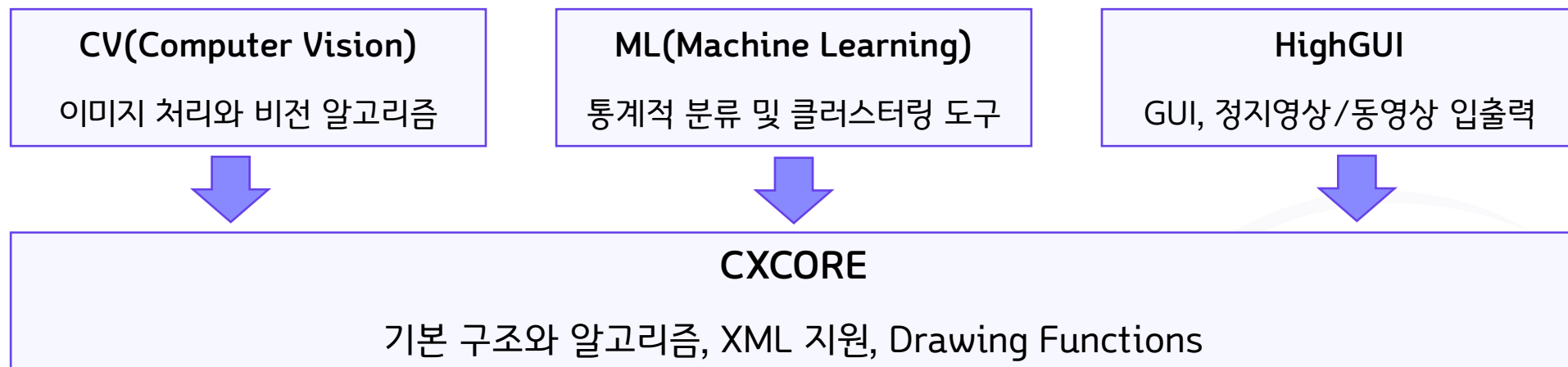


- **2009년, OpenCV 2.0 발표**
  - C 언어 대신 C++ 인터페이스 채택
  - C++클래스를 사용함으로써 메모리 관리가 좀 더 수월해지고 소스 코드 작성이 더욱 편리해짐
  - Mat라는 이름의 클래스를 사용하여 영상 데이터를 표현하기 시작
- **2015년 6월, OpenCV 3.0 발표**
  - OpenCV 프로젝트 구조가 크게 개선되고 전반적인 성능이 향상됨
  - OpenCL 사용성을 크게 확대한 T-API(Transparent API) 지원 시작
  - 2017년 8월 발표한 OpenCV 3.3부터 DNN 모듈이 기본 소스에 포함되기 시작

- **2018년 11월, OpenCV 4.0 발표**
  - C++11 문법 필수 지원
  - DNN 모듈 기능이 강화되어 AlexNet, Inception v2, Resnet, VGG 같은 영상 분류기 뿐만 아니라 Mask-RCNN, tiny YOLO 같은 최신 딥러닝 네트워크 구조를 지원함
  - 오래된 C API 지원이 종료되어 더 이상 IplImage 구조체 등을 사용할 수 없음
- **현재의 최신 버전: 4.8.0 (2023.07.02 Released)**



## • OpenCV의 기본 구조



- CV (Computer Vision)
  - 기본 구성 요소
  - 기본 이미지 처리와 고급 컴퓨터 비전 알고리즘을 포함
- ML (Machine Learning)
  - 머신 러닝 모델과 관련된 기능을 포함
  - ML을 활용할 수 있는 기본 통계 분류기와 클러스터링 도구 등
- HighGUI
  - OpenCV에 기본으로 포함되어 있는 GUI
  - 비디오와 이미지의 I/O 루틴
  - CV 기본 요소에서 지원하고 있지만 더 범용적으로 Qt를 사용
- CXCORE
  - 기본 데이터 구조와 여러 content가 구성되어 있음



## • OpenCV 모듈

- OpenCV 라이브러리는 다수의 모듈로 구성되어 있음

### 모듈

OpenCV에서 제공하는 다양한 클래스와 함수를  
기능과 성격에 따라 모아서 만들어 놓은 OpenCV의 부분 라이브러리

## • OpenCV의 주요 모듈

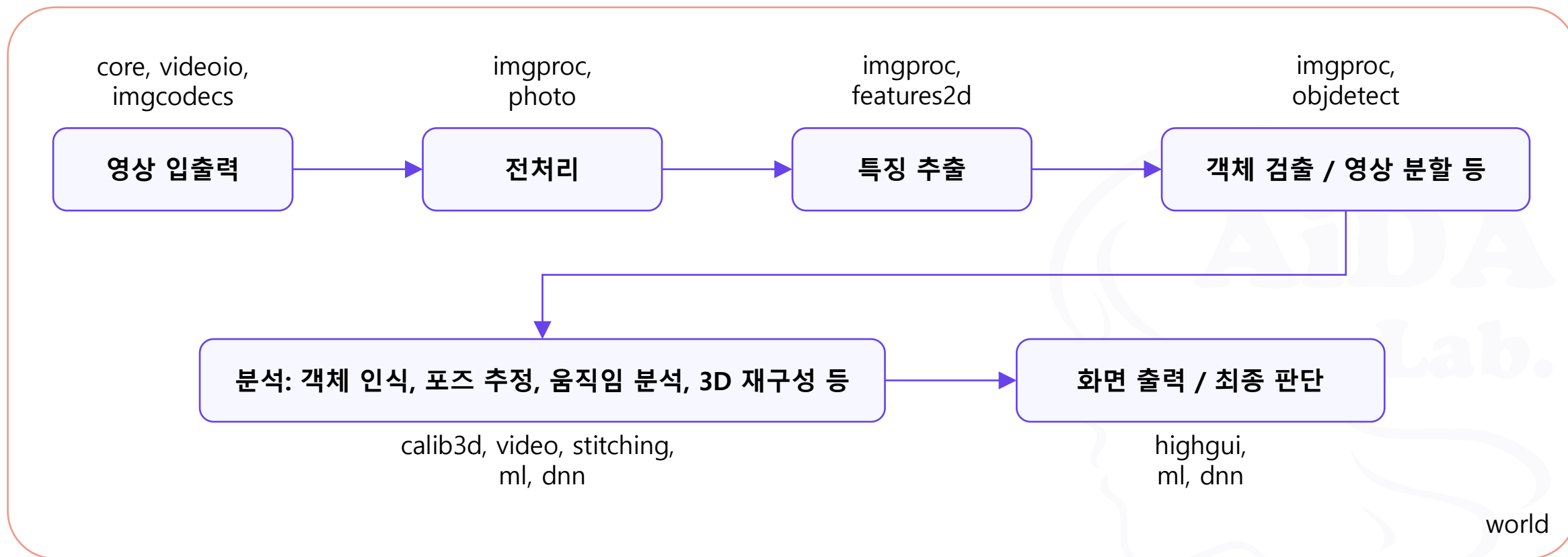
모듈 이름	설명
calib3d	카메라 캘리브레이션과 3차원 재구성
core	행렬, 벡터 등 OpenCV 핵심 클래스와 연산 함수
dnn	심층신경망 기능
features2d	2차원 특징 추출과 특징 벡터 기술, 매칭 방법
flann	다차원 공간에서 빠른 최근접 이웃 검색
highgui	영상의 화면 출력, 마우스 이벤트 처리 등 사용자 인터페이스
imgcodecs	영상 파일 입출력
imgproc	필터링, 기하학적 변환, 색 공간 변환 등 영상 처리 기능

모듈 이름	설명
ml	통계적 분류, 회귀 등 머신러닝 알고리즘
objdetect	얼굴, 보행자 검출 등 객체 검출
photo	HDR, 잡음 제거 등 사진 처리 기능
stitching	영상 이어 붙이기
video	옵티컬 플로우, 배경 차분 등 동영상 처리 기술
videoio	동영상 파일 입출력
world	여러 OpenCV 모듈을 포함하는 하나의 통합 모듈

- **OpenCV 모듈 사용 시 참고 사항**

- OpenCV 모듈은 각각 별도의 \*.lib 파일과 \*.dll 파일로 만들어짐
- OpenCV를 이용하는 프로그램을 개발하려면 여러 모듈 중에서 자신에게 필요한 모듈을 모두 선택하여 프로젝트에 포함시켜야 함
- core 모듈은 OpenCV에서 행렬 및 영상을 저장하는 용도로 사용되므로 항상 필요함
- 동영상 또는 정지 영상 파일을 불러오기 위해 videoio, imgcodecs 모듈이 필요함
- 기본적인 영상 처리 기능을 이용하기 위해 imgproc 모듈도 필요함
- 이외에도 특징점 검출, 객체 검출, 움직임 분석 등 특화된 컴퓨터 비전 기능을 사용하기 위해 features2d, objdetect, video 등 모듈이 필요함
- 영상 인식 또는 판단과 같은 고수준의 작업을 위하여 ml 또는 dnn 모듈을 사용할 수도 있음

- 일반적인 컴퓨터 비전 문제의 해결 과정과 관련 OpenCV 모듈



- Python 환경을 위한 설치

- OpenCV의 개발 환경(OS, HW, 개발언어 등)에 따라 설치 방법이 다름
- 본 과정에서는 Python을 사용하므로 Python 환경을 위한 설치만 설명함

## Google Colab

개인 시스템에서 환경을 설정하여  
개발을 진행할 수 있으며  
Google Colab을 이용하여  
개발을 진행할 수도 있음

OpenCV의 설치 및 사용 방법은  
두 환경 모두 동일함

### Installation

...

Select your preferences and run the install command.

Operating System:	Linux	macOS	Windows			
Building From Source:	Yes		No			
Language:	Python	C++	Java	Android	iOS	JavaScript
Run this Command:	<pre>pip3 install opencv-python</pre>					

**THANK  
YOU**

