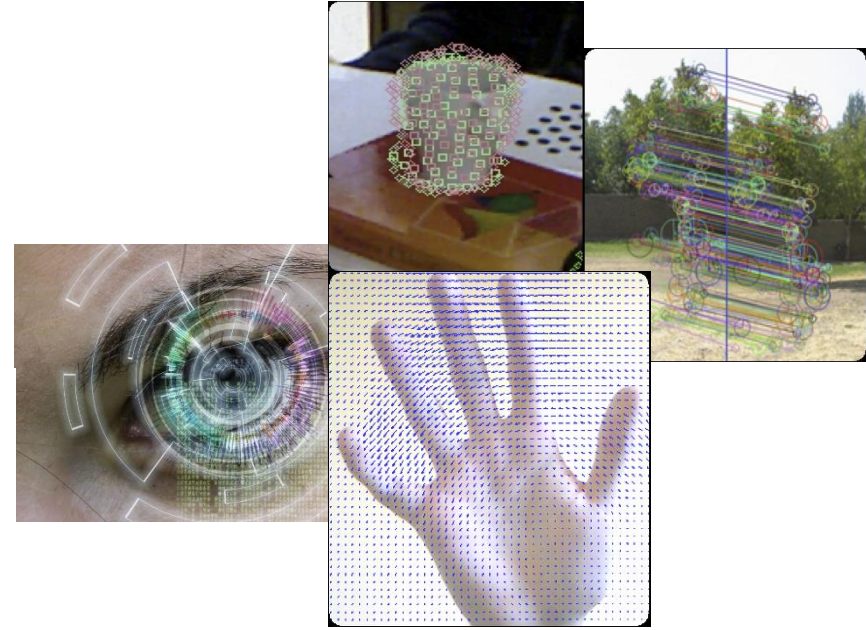


2023 Autumn

COMPUTER VISION

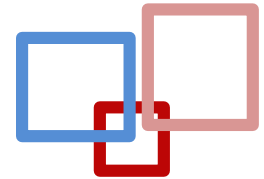
비전
프로그래밍



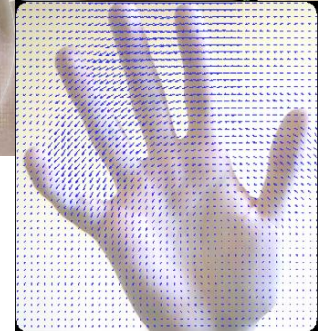
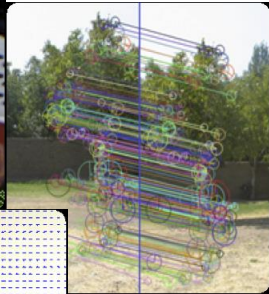
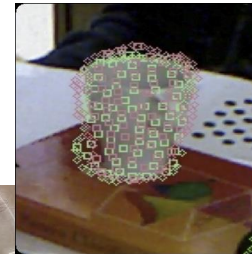
Dept. of IT Engineering, Sookmyung Women's University
Prof. Byung-Gyu Kim

1장. 교과목 소개

COMPUTER VISION 비전 프로그래밍



교과목 소개



Introduction (교과목 소개)

- 교과목 명: 컴퓨터비전(Computer Vision) (3/2/2)

- 담당 교수

- 이 름: 김 병 규 (IT공학과)
- 사 무 실: 새힘관 102호실
- 연락방법: 010-5037-3452, bg.kim@sm.ac.kr 또는 bg.kim@ieee.org
- 연구실 홈페이지: <http://ivpl.sookmyung.ac.kr>

- 강의 시간 및 실습:

- 초반 : 이론 2시간+실습 2시간 = 총 4시간/주
- 이론 수업: 이론 강의실에서 진행
- 실습 수업: 주로 수요일 수업으로 개인 노트북 으로 실습 진행 예정

- 강의 방법

- 기본 이론 강의/구현 실습 수업 병행
- 기본 이론 강의 후 실습을 통해 확인하는 방식으로 진행하여 질문을 통해 다양한 생각들을 도출하도록 함

Introduction (교과목 소개)

■ 교과목 개요

- 영상 및 비디오 데이터를 처리하여 의미 있는 정보를 추출하는 과정을 이해하기 쉽게 개념적으로 학습한다.
- 기초 이론 강의를 통해 기본 원리를 이해하고 실습을 통하여 다양한 영상에 적용하여 결과를 확인한다.
- 오픈 소스인 최신 OpenCV 4.4.0를 활용하여 이론적 내용을 확인하고 다양한 기하학적 특징 추출 및 인식 등 기술을 개념적으로 이해하기 쉽게 강의한다.

■ 교육목표

- 디지털 영상처리 및 컴퓨터 비전 기술 분야의 이해 능력
- 다양한 응용분야의 이해 및 실무 기술의 이해 능력
- OpenCV 기반의 개발 환경 설정 및 기본 모델링 이해 능력
- 다양한 특징 기반의 응용 SW 개발 능력

■ 강의 자료

- E-강의동 강의 게시판에서 매주 강의자료 업로드함

Checking Point (강의수강 관련 점검 사항)

■ 요구되는 사전 지식

- 열정(passion)..!!!
- 기본 프로그래밍 능력(sense)
- 수업에 열심히 참여하는 성실함 등등
- 가능하면 기초부터 하나 하나 모두 소개를 할 예정이므로 즐겁게 수업 시간에 충실해야 함

■ 준비물 (실습 수업 관련)

- **HW**
 - 개인 노트북 지참 (가능하면)
 - 실습실 PC에 환경 설정도 가능하나 권장하지 않음
- **SW**
 - **Visual Studio 2017 이상 버전** (C/C++ 프로그래밍 가능한 환경)
 - OpenCV 4.1.X ~ 이상 설정 ← 수업 시간에 직접 진행할 예정임

Evaluation (평가 방법)

■ 평가 항목

■ 정량 평가 (상대평가)

- 중간고사: 25% 내 외
- 기말고사: 35% 내 외
- 프로젝트: 20% (팀 단위 소규모 프로젝트 포함)
- 출석: 10%
- 과제: 10%
- 기타: 수업 참여도 등

■ 정성평가

- 항목1
 - 컴퓨터 비전의 기본 개념 및 이론적 모델링을 잘 이해하는가?
 - OpenCV 실습 환경 구축 및 기본 SW 개발 능력이 있는가?
- 항목2
 - 컴퓨터 비전의 응용 구조를 잘 이해하는가?
 - 다양한 응용 SW 개발 능력이 있는가?
 - Open 소스 활용 능력이 있는가?

■ 이론/실습 숙제

- 약 2회 내외

Text book (강의 교재 소개)

- 주로 강의 자료와 기타 공식 매뉴얼을 활용할 예정
- 아래 주교재를 중심으로 활용 예정



도서명 | OpenCV 2 Computer Vision Application Programming
Cookbook (ppt)

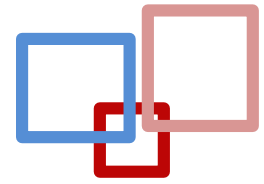
저자 | Robert Laganière

출판사 | -

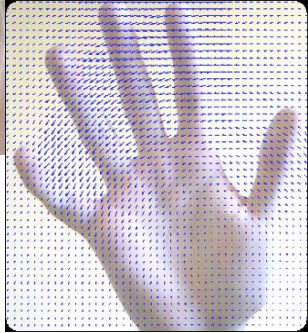
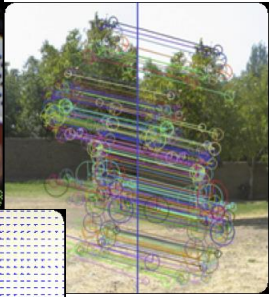
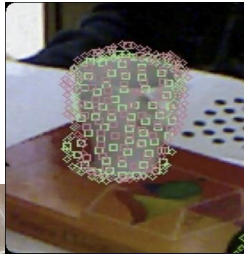
(번역판 : OpenCV를 활용한 컴퓨터 비전 프로그래밍(개정판) - 로버트 라가니에)

주	Contents	주	Contents
1	<ul style="list-style-type: none"> - 교과목 소개 및 기본 요소 . 교과목 로드맵(read map) 확인 . 컴퓨터 비전이란? . 기본 실습 환경 설정: OpenCV extra module 	9	<ul style="list-style-type: none"> - Deep learning Basics(1): . Neural Network (신경망) 기초 이론 및 학습 알고리즘 . Single/Multi-layer Perceptron 이해
2	<ul style="list-style-type: none"> - 영상처리 및 컴퓨터비전 소개 . 다양한 영상처리 분야 . 고급 컴퓨터 비전 소개 - OpenCV 기본 기능 실습: 변수 및 데이터 구조 등 . 클래스 기반 SW 구조 이해 . 디자인 패턴 이해 	10	<ul style="list-style-type: none"> - Deep learning Basics(2): . Error correction learning rule (에러정정 학습 기법) . Deep learning frameworks 이해 . Back-propagation Algorithm 개념 이해
3	<ul style="list-style-type: none"> - 영상 화소 처리 구조 이해 - 화소 처리 실습 . 화소처리 구조 및 실습 . 관련 API 실습 - 비디오 시퀀스 처리 구조 이해 및 API 실습 	11	<ul style="list-style-type: none"> - Deep learning Basics(3): . Anaconda 3 기반 Keras 실습 환경 설정 . Keras 기반 deep learning framework 구축 구조 . Keras 기반 deep learning programming 실습: mnist 기반으로 . Keras 기반 deep learning programming 확장 실습
4	<ul style="list-style-type: none"> - 히스토그램 처리 . 히스토그램 정의 및 처리 알고리즘 - 히스토그램 처리 API 활용 실습 . Mean-Shift 알고리즘 이해 및 응용 - 형태학 연산기반 영상변환: 침식/팽창 등 기본 이론, 영상분할 강의 - 형태학 연산기반 API 실습 및 응용 	12	<ul style="list-style-type: none"> - Deep learning Basics(4): . 영상 인식 및 분류 기술의 이해 . Tensorflow 기반 데이터 학습 . Python 기반 데이터 구조 학습 및 이해
5	<ul style="list-style-type: none"> - 기본 특징 추출 학급: 선, 라인, 외곽선 등 . Hough 변환 기법 - 기본 특징 추출 AIP 학습 및 실습 . hough 변환을 통한 라인 검출 . 영상 내 원 검출 등 	13	<ul style="list-style-type: none"> - Deep learning Basics(4): . 전이 학습(transfer learning)의 개념 및 실제 . 얼굴 표정 인식 기반 사용자 감성인식 기술의 이해 . 표정 인식 관련 deep learning (CNN) 구조 구현 . 표정인식 관련 데이터 만들어서 학습 하기
6	<ul style="list-style-type: none"> - 기하학적 특징 추출: . 관심점 정의 및 모델링 . 관심점 추출 방법: Harris corner, SURF 등 . 매칭을 통한 인식 - 관심점 실습을 위한 API 이해 . 다양한 추출 결과 확인 및 응용 실습 	14	<ul style="list-style-type: none"> - Deep learning Basics(5): . Object Detection 기본 원리 이해 . AutoEncoder 구조 이해 . Object 검출 실습 및 Autoencoder 응용 실습
7	<ul style="list-style-type: none"> - 얼굴 특징에 의한 얼굴 검출 . 주요 얼굴의 특징들, 특징 기반 얼굴 검출 	15	기말발표/기말고사(대면)
8	중간고사 (대면)	16	

COMPUTER VISION 비전 프로그래밍



Preview

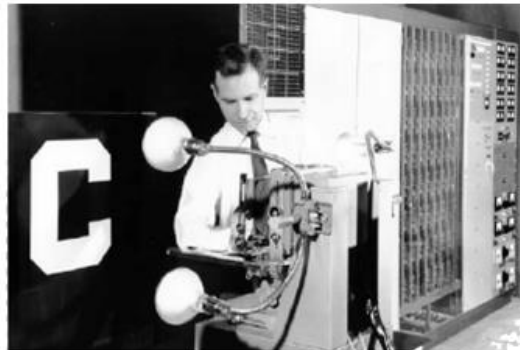


PREVIEW

- 오감 중에 **시각**은 가장 강력한 인지 기능
- 컴퓨터비전(Computer Vision)은 **컴퓨터를 이용하여 시각 기능을 갖는 기계 장치를 만드는 기술 분야**



(a) 사진의 태동



(b) 초창기 컴퓨터 비전



(c) 현재 컴퓨터 비전



(d) 미래 컴퓨터 비전

그림 1-1 인공 시각을 만들기 위한 인류의 노력

PREVIEW

- 컴퓨터비전(Computer Vision)와 영상처리(Digital Image Processing)의 차이점

Computer Vision

- 인공지능 (Artificial Intelligence)의 한 분야로서, 어떤 영상에서 장면이나 특징(scene or features) 들을 " 이해 (Understanding)" 하는 컴퓨터를 프로그램 하는 것
- 기계에 인간의 시각적 지능을 부여하는 것을 목표로 함
- 영상처리를 통해 정제된 데이터를 통해 다양한 특징 및 장면 이해를 함

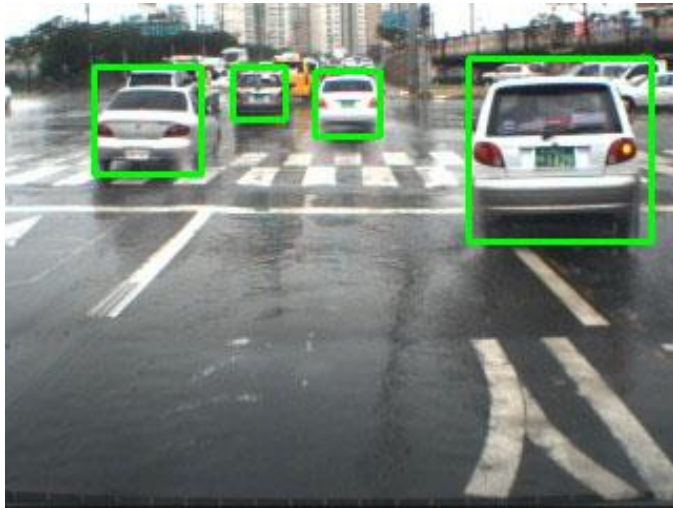
Image Processing

- 영상처리는 사진 또는 그림 등을 디지털화하여 컴퓨터에서 이를 처리하는 기술
- 화상을 장면해석 단계에서 이용하기 쉬운 형태로 변형시키는 단계
- 일반적으로 컴퓨터비전의 전처리 과정의 개념

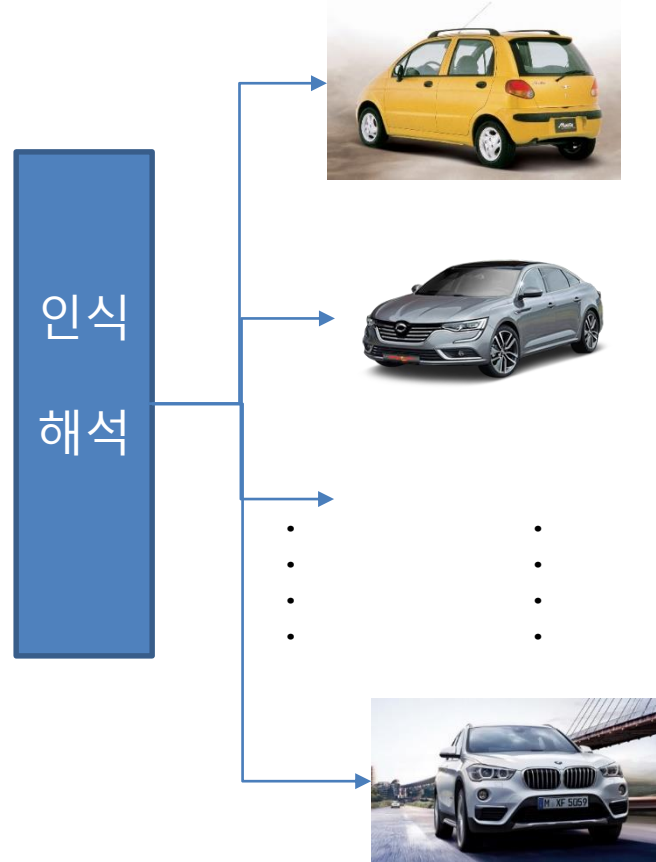
Visual Intelligence Processing
(시각지능 처리 과정)

PREVIEW

■ 컴퓨터비전(Computer Vision) VS. 영상처리(Digital Image Processing)



Edge, color, filtering, etc.



← Digital Image Processing →

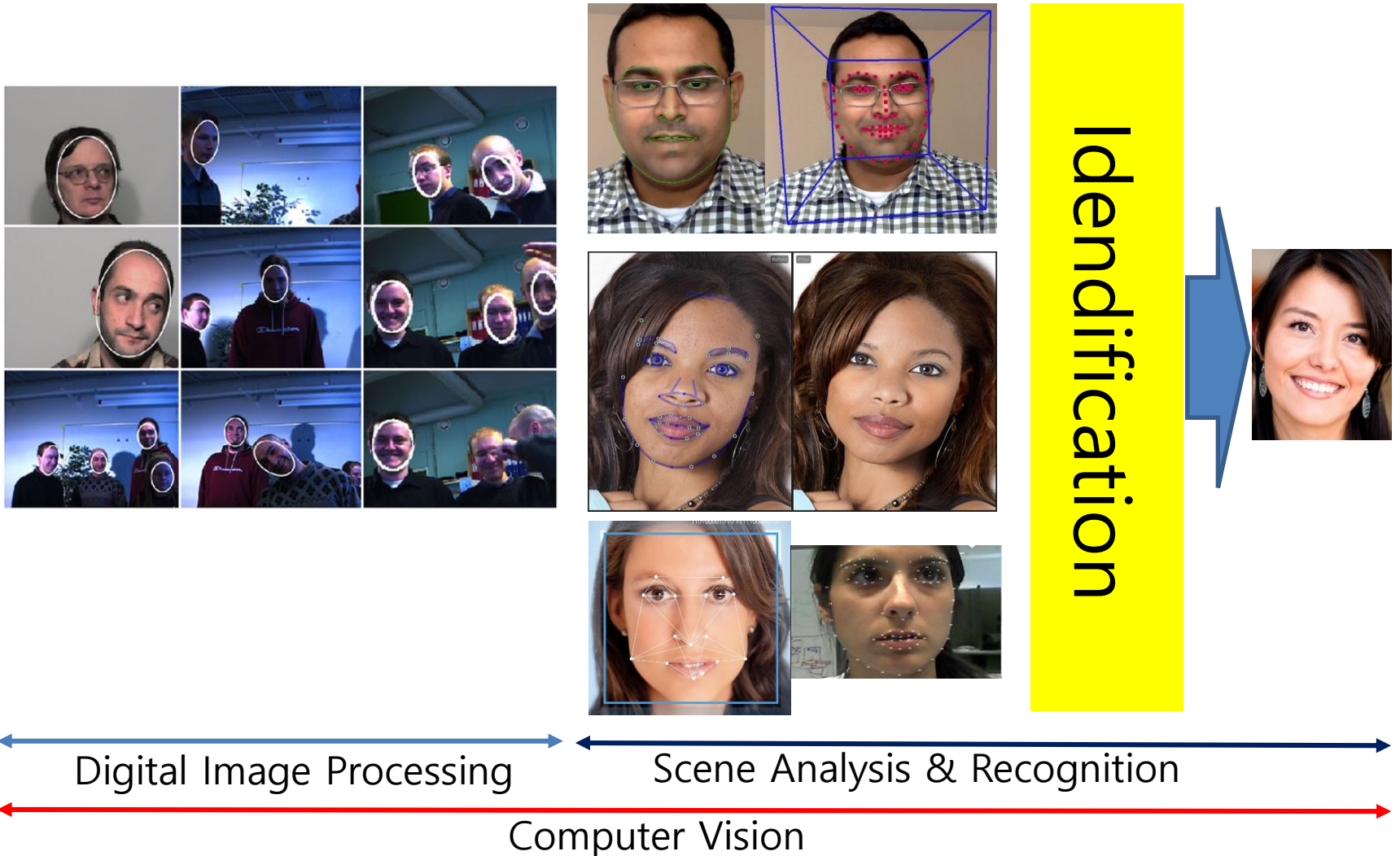
← Scene Analysis & Recognition →



Computer Vision

PREVIEW

- 컴퓨터비전(Computer Vision) VS. 영상처리(Digital Image Processing)



주요 내용

1. 왜 컴퓨터 비전인가?

→ 흥미로운 응용 사례를 통해 컴퓨터 비전을 공부하는 이유를 설명한다.

2. 컴퓨터 비전 문제는 어떻게 해결하나?

→ 컴퓨터 비전이 왜 어려운지, 연구자들이 취하는 접근 방법, 문제 해결에 사용하는 주요 도구들을 살펴본다.

3. 시스템 설계

→ 컴퓨터 비전 시스템을 설계하고 구현할 때 거치는 과정을 자세히 설명한다.

4. 인접 학문

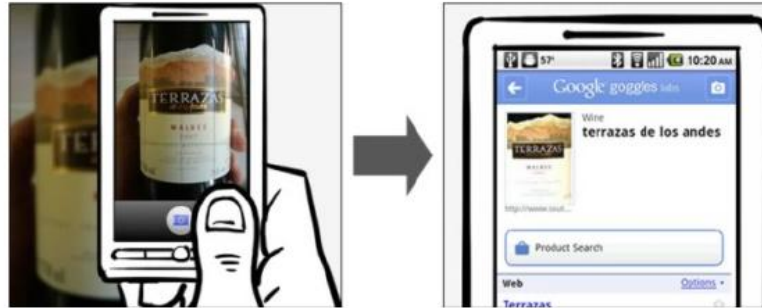
→ 컴퓨터 비전과 인접한 학문인 영상 처리, 컴퓨터 그래픽스, 그리고 패턴 인식을 대비시켜 서로 어떻게 협력하고 어떻게 다른지 설명한다.

5. 학습을 위한 자원

→ 컴퓨터 비전을 공부하는 데 도움이 될 책, 학술지와 학술대회, 웹 포털을 소개한다.

1.1 왜 컴퓨터 비전인가?

■ 생활 속의 응용 예



(a) 구글 앱



(b) Xbox 게임기

그림 1-2 와인 병을 인식하는 스마트폰 앱과 사용자의 동작을 인식하는 게임기

1.1 왜 컴퓨터 비전인가?

■ 팡창하는 응용

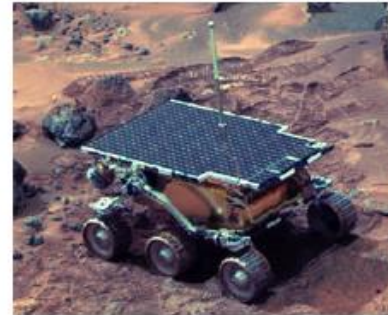
- 오락, 교통, 보안, 산업, 계산 사진학, 의료, 과학, 농업, 군사, 모바일 등



(a) 구글 자동차(자율 주행)



(b) 수술용 로봇 다빈치



(c) 화성 탐사선



(d) 딸기 따는 로봇



(e) 군사용 로봇 빅독



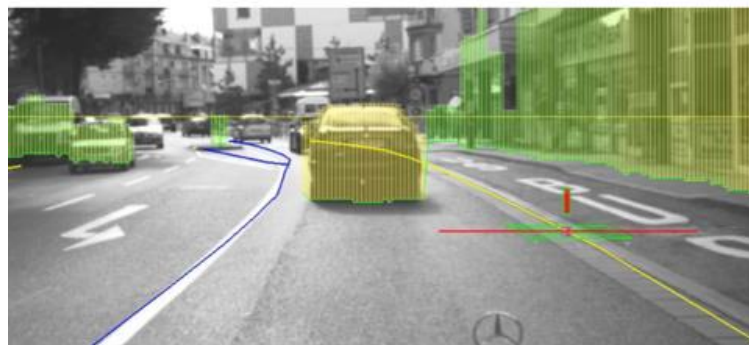
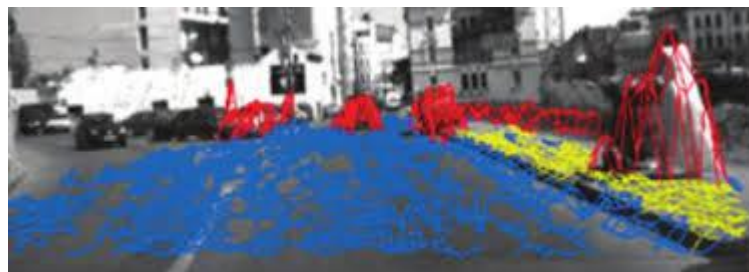
(f) 리프스냅

그림 1-3 컴퓨터 비전의 응용 사례

1.1 왜 컴퓨터 비전인가?

■ 최근의 hot issue

- 자동차 융합 기술



1.1 왜 컴퓨터 비전인가?

■ 최근의 hot issue

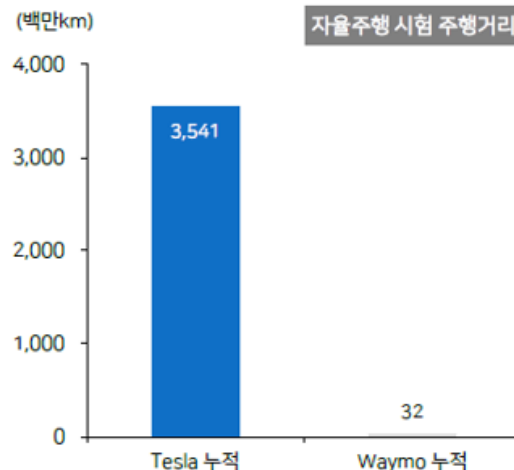
- 자율주행 자동차 기술 <https://www.youtube.com/watch?v=CgJ2FDP36gl>



1.1 왜 컴퓨터 비전인가?

- Tesla의 자율주행 시스템 (autopilot)
 - 8대의 카메라/1대의 Laser 센서
 - 카메라 이미지 기반 -> 3D 라벨링 기술로 고도화를 통한 자율주행 -> 가격이 저렴해서 전기차 가격을 낮춤
 - 초정밀지도와 라이다 필요 없다고 주장 -> 소프트웨어가 너무 무거워져 소비전력이 높아져 연비를 떨어트림. 라이다와 슈퍼컴퓨터급의 GPU는 가격도 매우 비싸 상용화하기 어려움. 이에 따른 가격 상승: 아반떼 수준차를 1억에 팔면 누가사겠는가? -> **테슬라는 철저히 자율주행차의 상용화 관점.**
 - 웨이모(Google)는 연구소 단계의 자율주행 소프트웨어적 관점으로만 개발 중.
 - 테슬라는 철저히 경험에 의한 학습에 집중/웨이모는 시험에 집중

Tesla, 시험이 아닌 경험에 집중하는 유일 업체



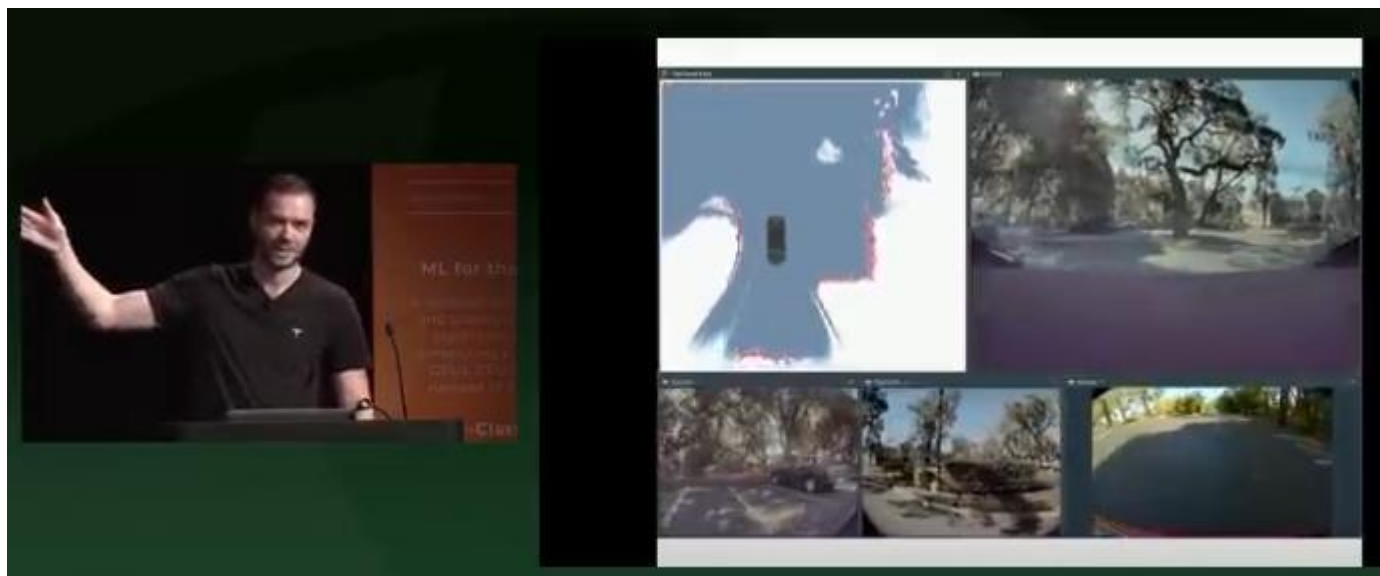
자료: Tesla, Waymo, 메리츠증권 리서치센터

1.1 왜 컴퓨터 비전인가?

8대의 카메라로부터 받은 이미지 영상을 합쳐서 3차원 입체영상화



3D 라벨링된 영상을 기반으로 실시간으로 자체지도를 제작하며 주행

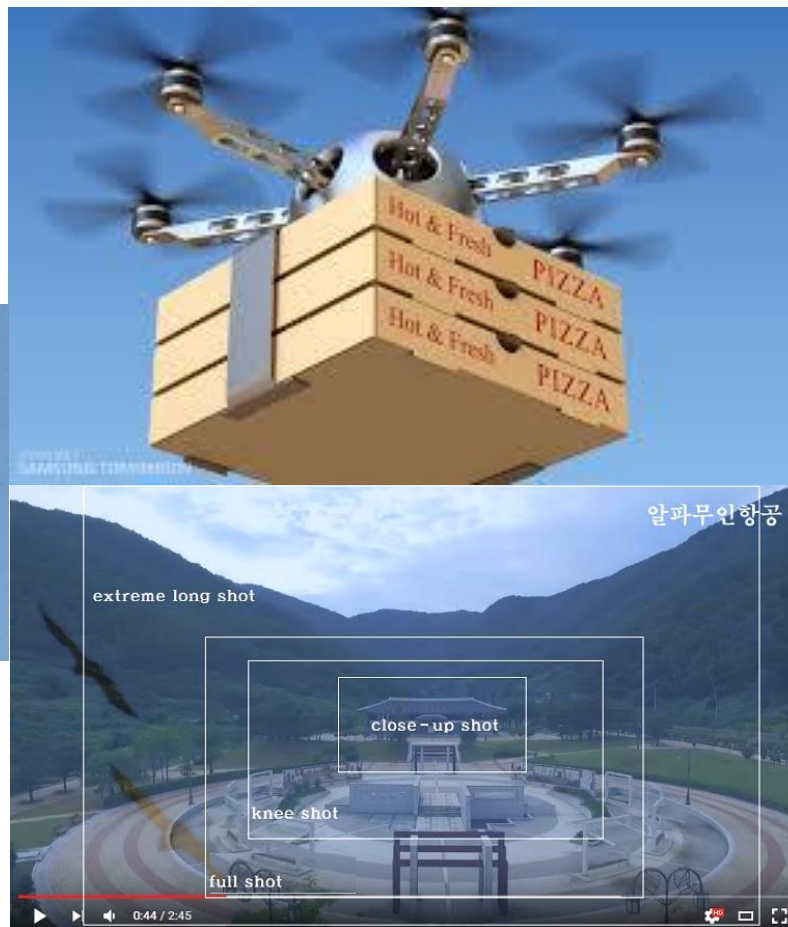


1.1 왜 컴퓨터 비전인가?



1.1 왜 컴퓨터 비전인가?

- 차세대 항공 기술 - 무인 드론 기술



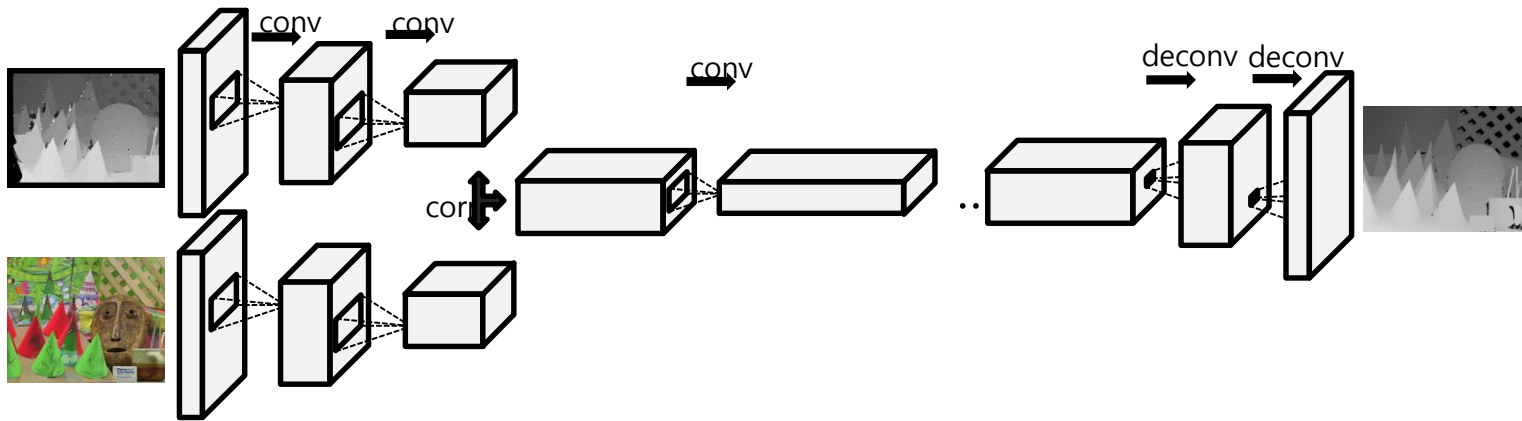
1.1 왜 컴퓨터 비전인가?



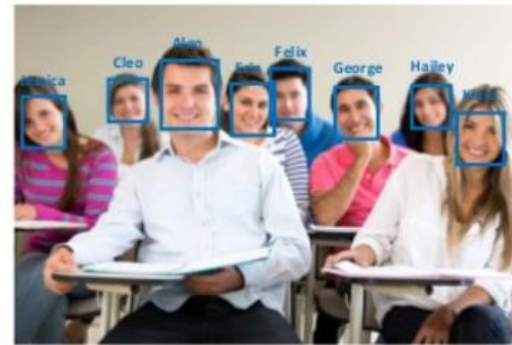
관련사이트 <https://www.youtube.com/watch?v=neQU6Ovlees>

1.1 왜 컴퓨터 비전인가?

- **Deep Learning** is very hot..!!!
 - Image-based machine learning technology



Face Detection

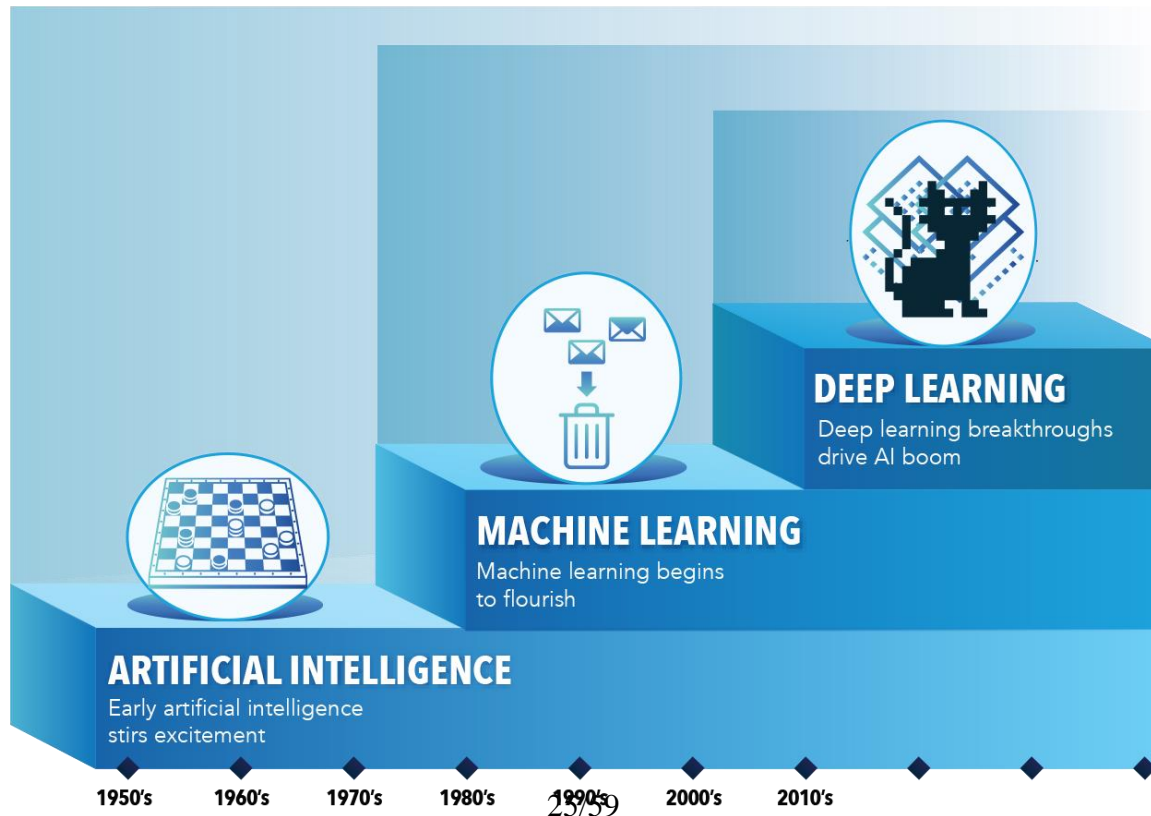


Face Recognition

1.1 왜 컴퓨터 비전인가?

■ Artificial Intelligence (AI)

- The simulation of human intelligence processes by machines, especially computer systems.
- Specific applications of AI include **expert systems, natural language processing, speech recognition and machine vision.**



1.1 왜 컴퓨터 비전인가?

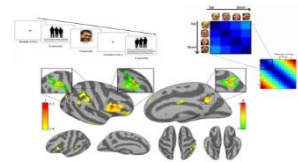
▪ How does AI work?

- AI systems work by ingesting **large amounts of labeled training data**, **analyzing the data for correlations and patterns**, and using these patterns to **make predictions** about future states.
- AI programming (software) focuses on **three cognitive skills: learning, reasoning and self-correction**.

1] **Learning processes:** acquiring data and creating rules for how to turn the data into actionable information. The rules, which are called **algorithms**, provide computing devices with step-by-step instructions for how to complete a specific task.



2] **Reasoning processes:** **choosing the right algorithm** to reach a desired outcome.



logical + inferential reasoning

3] **Self-correction processes:** This aspect is designed to continually **fine-tune algorithms** and ensure they provide the most accurate results possible.



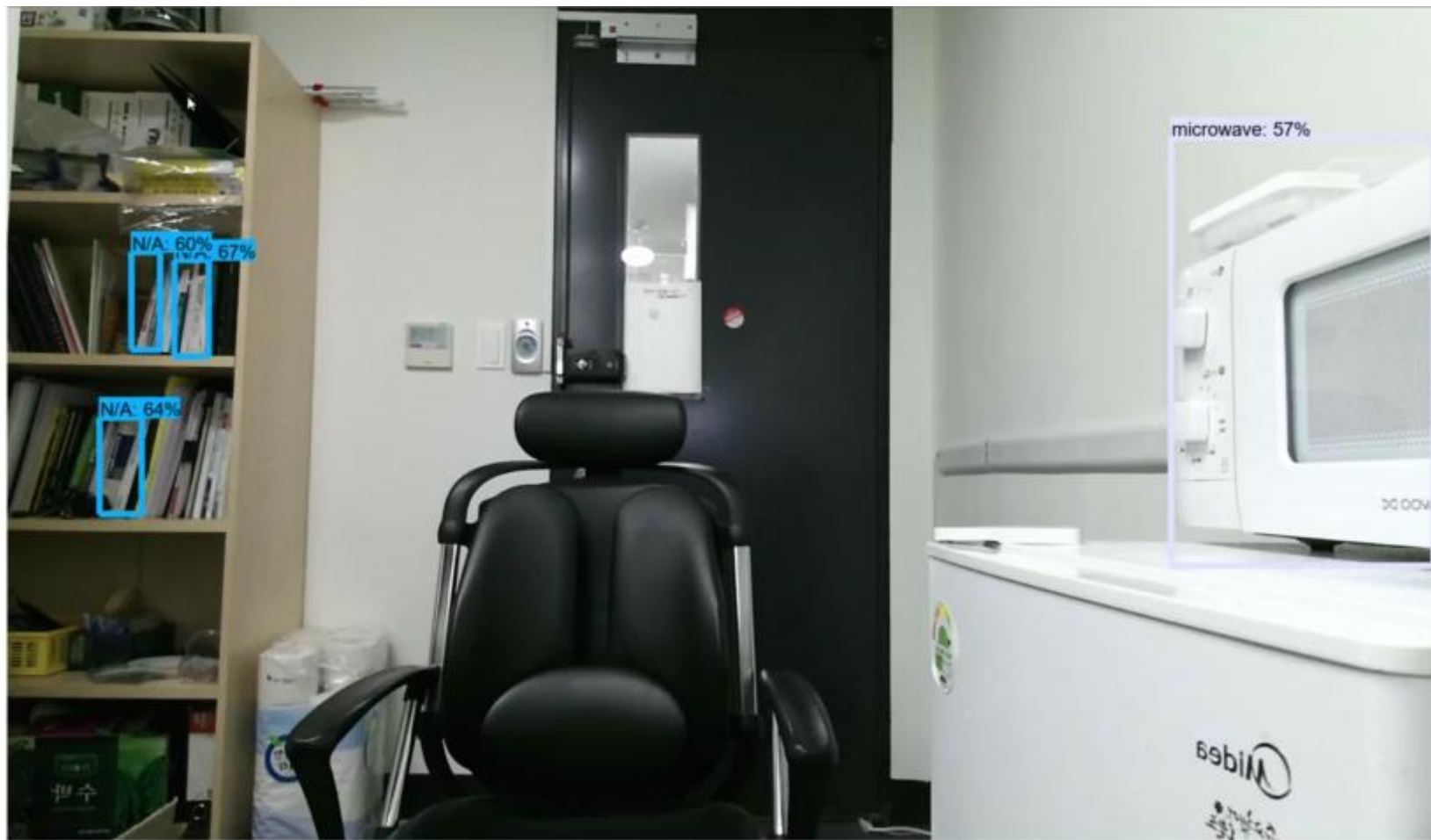
1.1 왜 컴퓨터 비전인가?

- Tensorflow 기반 Object 검출 및 인식 기술 (캡스톤디자인 과제)



1.1 왜 컴퓨터 비전인가?

- Deep learning 기반 장면 인식 및 해석 기술 (캡스톤디자인 과제)



1.2 컴퓨터 비전 문제는 어떻게 해결하나?

1.2.1 과학적 접근과 공학적 접근

1.2.2 계층적 처리

1.2.3 문제 해결 도구

1.2 컴퓨터 비전 문제는 어떻게 해결하나?

- 사람의 시각은 빠르고 매우 강건
- 컴퓨터로 사람 수준을 달성하는 목표는 현재 기술로 불가능
 - 왜 어려운가?
 - 실용적인 시스템은 어떻게 만드나?

1.2.1 과학적 접근(목표1)과 공학적 접근(목표2)

- 목표 1 : 사람의 시각에 거의 유사하게 동작하는 인공 시각을 만든다.



(a) 목표1 - 제약이 없는 상황

- 목표 2 : 한정된 범위에서 특정한 임무(응용)를 달성하는 인공 시각을 만든다.



(b) 목표2 - 제약이 있는 상황

그림 1-4 컴퓨터 비전의 목표

1.2.1 과학적 접근과 공학적 접근

■ 과학적 접근

- 목표 1을 달성하려는 노력
- **사람 시각의 원리를 밝혀낸 다음 컴퓨터로 모방**
- 뇌 과학의 주요 관심사
- **지식 표현, 학습, 추론, 창작 같은 인공지능이 필수**

■ 어려운 이유

- 불량 문제 \leq 모든 경우의 수를 감안할 수 있는가?
- 다양한 변형 발생 (기하학적 변환, 광도 변환)

1.2.1 과학적 접근과 공학적 접근

■ 공학적 접근

- 목표 2를 달성하려는 노력
 - 특정 상황에서 특정 임무를 수행하는 실용 시스템 구축
- 성공적인 시스템
 - 많은 응용 현장에서 쓰고 있음
 - 컴퓨터 비전이 사람보다 뛰어난 경우 (예, 엔진 실린더 정밀 측정, 칩 검사 등)

■ 실용적인 성능 달성의 어려움

- 여전히 역 문제, 불량 문제, 다양한 변환 발생
- 영상은 숫자 배열 형태 (센싱된 데이터 이후의 과정을 주로 개발)

125 134 125 122 127 127 120 130 139 135 139 140 133 127 127 130 133 135 138 133 137 139 134 130 125 121
117 123 114 116 120 122 118 120 122 117 122 126 124 117 106 100 99 102 105 120 118 113 109 105 106 111
109 110 105 102 112 123 130 135 147 171 191 184 183 174 157 139 124 107 90 92 87 88 92 93 88 89
108 105 100 116 117 129 163 195 210 217 205 215 211 198 185 176 167 143 117 91 80 77 88 91 84 79
107 103 102 120 146 173 200 193 172 165 138 141 135 123 118 125 139 143 137 121 99 84 85 88 82 81
104 107 115 134 159 171 170 136 115 129 107 83 83 82 80 83 90 103 113 125 108 93 91 90 86 83
107 120 137 160 150 125 139 150 167 174 115 99 94 93 98 98 89 87 91 104 103 99 97 95 94 95
111 133 156 134 151 157 189 206 216 212 136 114 92 83 97 110 108 100 98 97 101 101 95 92 103 120
130 145 164 165 185 213 219 210 212 196 158 108 123 137 137 123 111 121 134 145 132 130 147 159 163 171
138 151 170 185 195 215 222 211 214 218 209 160 152 151 157 163 166 167 166 159 155 160 180 193 195 193
142 153 171 190 190 204 218 213 207 214 218 213 204 195 192 189 183 178 173 161 159 163 171 183 189 187
141 151 164 188 178 180 197 204 201 197 196 196 193 190 187 176 163 157 156 156 161 163 166 174 186 192
144 151 160 185 183 176 176 187 192 191 188 193 184 178 177 174 165 156 151 148 163 177 182 188 200 203
152 160 168 176 193 193 182 180 180 174 172 164 161 159 154 146 140 143 149 173 184 190 190 193 199 205
159 168 178 178 202 206 197 194 187 175 175 167 172 179 180 176 176 188 203 215 212 206 204 202 204 205
161 171 185 197 210 204 199 211 210 206 212 219 210 206 215 225 226 220 215 214 209 210 214 216 211 200



그림 1-5 숫자 배열 형태의 영상⁹

1.2.2 컴퓨터비전 분야의 계층적 처리

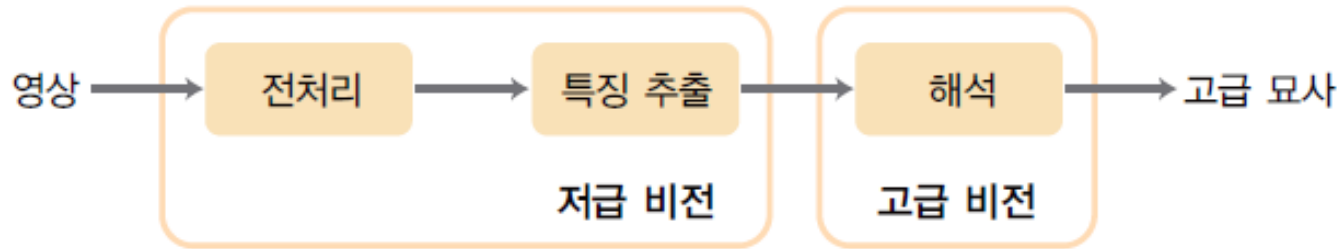


그림 1-6 컴퓨터 비전의 계층적 처리

■ 전처리

- 주로 영상 필터링 관련 이슈

■ 특징 추출

- 에지, 선분, 영역, 텍스처, 지역 특징 등을 검출하고 특징 벡터 추출

■ 해석

- 추출된 특징을 기반으로 사물에 대한 속성 인식 메커니즘 사용
- 응용에 따라 다양한 형태

1.2.3 문제 해결 도구

■ 자료 구조와 알고리즘

- 배열, 트리, 그래프, 힙, 해싱, 탐색 트리 등
- 탐욕 방법, 동적 프로그래밍, 한정 분기 등
- 고속 처리가 주된 관심

■ 수학

- 선형 대수, 미적분학, 확률과 통계 등
- 최적화 문제 풀이

■ 기계 학습

- 심층신경회로망 학습(deep learning)을 활용하는 사례 급증
- 신경망, SVM, AdaBoost, 임의 숲 등의 분류기 (전통적 기계학습)

1.3 시스템 설계

1.3.1 문제 이해

1.3.2 데이터베이스 수립

1.3.3 알고리즘 설계와 구현

1.3.4 성능 평가

1.3 시스템 설계

■ 순차 처리와 피드백

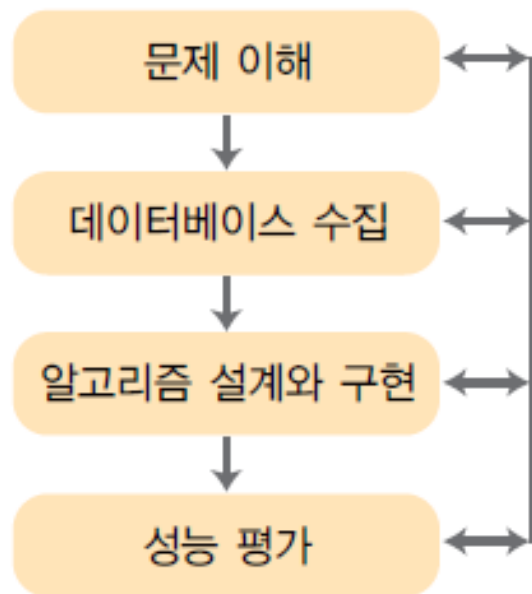


그림 1-8 컴퓨터 비전 시스템 설계 과정

1.3.1 문제 이해

- 주어진 문제에 대한 직관적이고 철저한 이해 중요
 - 합리적 제약 조건 수립 필요
 - 예) 얼굴 인식기
 - 크기의 일정한 정면 얼굴 vs. 자연 영상 속의 얼굴
 - 전자는 제작 쉽지만 응용 분야 한정. 보안 장치에 활용 가능하지만 사진 분류 응용에 적용 불가능



VS.



1.3.2 데이터베이스 수집

■ 데이터베이스

- 질적으로 양적으로 우수해야 고성능 시스템 제작 가능
- 데이터베이스=학습 집합+테스트 집합
- 수집 방법
 - 직접 수집 (많은 비용 부담해야 하지만 개발자에게 자산)
 - 인터넷에서 다운로드 (고품질의 데이터베이스 풍부함. 부록 B 참조)



(a) 물체 인식



(b) 차량 번호판 인식



(c) 영역 분할

그림 1-9 여러 가지 용도의 데이터베이스

1.3.3 알고리즘 설계와 구현

■ 알고리즘 선택의 중요성과 어려움

- 새로운 알고리즘 개발 또는 기존 알고리즘 중에서 주어진 문제에 적합한 것 선택
- 예) 손 동작 인식기 제작
 - 에지, 영역, 지역 특징 중에 어떤 것을 사용할까?
영역을 사용한다면, 어떤 영상 분할 알고리즘을 사용할까? 어떤 추적 알고리즘을 사용할까?

■ 선택 방법

- 데이터베이스를 이용하여 성능 실험을 해 봄 (설계자의 경험과 직관이 중요)
- 성능 비교 분석 논문을 참조

1.3.3 알고리즘 설계와 구현

■ 프로그래밍

- OpenCV (부록 A)
 - 국내 커뮤니티 <http://cafe.naver.com/opencv>
 - 공식 포럼 <http://opencv.org>
- Matlab이 제공하는 IPT (Image Processing Toolbox)
- 오픈 소스 (부록 B 참조)

1.3.4 성능 평가: 분류 및 인식

■ 인식 성능 측정

$$\text{정인식률} = \frac{c}{N}, \text{기각률} = \frac{r}{N}, \text{오류율} = \frac{e}{N}$$

$$\begin{aligned} \text{이때 } c &= \text{맞는 샘플수}, r = \text{기각한 샘플수}, e = \text{틀린 샘플수} \\ (N &= c + r + e) \end{aligned} \quad (1.1)$$

- 부류가 심한 불균형일 때 부적절
(예, 칩 검사에서 불량률이 0.1%라면, 임의 짐작 분류기의 정인식률은 99.9%)

■ 혼동 행렬(Confusion Matrix)

- 오류 경향을 세밀하게 분석하는데 사용

표 1-1 부류가 두 개인 경우의 혼동 행렬

참부류 \ 분류 결과	ω_1	ω_2
	ω_1	$n_{11}(\text{TP})$
ω_2	$n_{21}(\text{FP})$	$n_{22}(\text{TN})$

1.3.4 성능 평가

- 참/거짓 긍정률, 참/거짓 부정률, 재현률과 정확률, F 측정

$$\begin{aligned} \text{거짓 긍정률} \left(\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \right) &= \frac{n_{21}}{(n_{21} + n_{22})} \\ \text{거짓 부정률} \left(\text{FNR} = \frac{\text{FN}}{\text{TP} + \text{FN}} \right) &= \frac{n_{12}}{(n_{11} + n_{12})} \\ \text{참 긍정률} \left(\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \right) &= \frac{n_{11}}{(n_{11} + n_{12})} \\ \text{참 부정률} \left(\text{TNR} = \frac{\text{TN}}{\text{FP} + \text{TN}} \right) &= \frac{n_{22}}{(n_{21} + n_{22})} \end{aligned} \quad (1.2)$$

$$\begin{aligned} \text{Precision} \quad \text{정확률} &= \frac{n_{11}}{(n_{11} + n_{21})} \\ \text{Recall} \quad \text{재현율} &= \frac{n_{11}}{(n_{11} + n_{12})} \end{aligned} \quad (1.3)$$

$$\begin{aligned} F_\beta &= (1 + \beta^2) \frac{\text{정확률} \times \text{재현율}}{\beta^2 \times \text{정확률} + \text{재현율}} \\ F_1 &= \frac{2 \times \text{정확률} \times \text{재현율}}{\text{정확률} + \text{재현율}} \end{aligned} \quad (1.4)$$

1.3.4 성능 평가

예제 1-1 얼굴 검출의 성능 측정

[그림 1-10]은 세 개의 영상을 가진 데이터베이스에서 얼굴을 검출한 결과를 보여준다. 앞에서 다룬 정확률, 재현율, F_1 측정을 구해보자.



그림 1-10 얼굴 검출 성능

총 15개의 얼굴 중 12개를 옳게 검출했으므로 참 긍정 $n_{11}=12$, 세 개의 얼굴을 못 찾았으므로 거짓 부정 $n_{12}=3$, 그리고 얼굴 아닌 곳을 얼굴로 검출한 것이 두 개이므로 거짓 긍정 $n_{21}=2$ 이다. 따라서 정확률은 $12/14$ 이고 재현율은 $12/15$ 이다. F_1 측정은 $24/29$ 이다.

1.4 인접 학문

- 상호 협력이 강해지는 추세(분야융합에 의한 시너지)
 - 예) 영화 제작, 계산 사진학 등

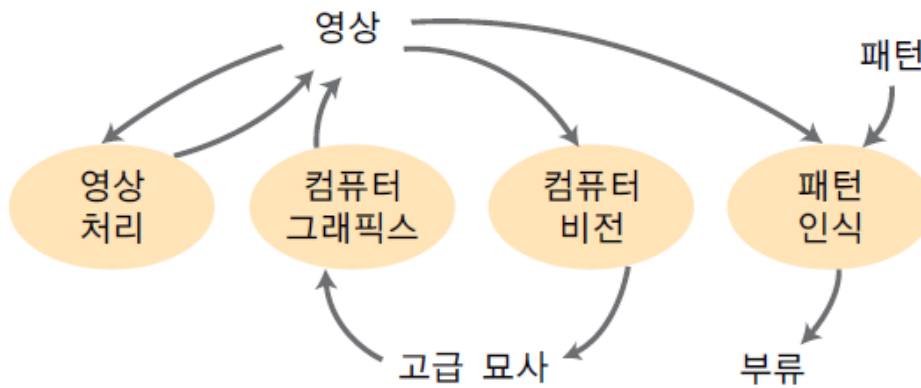


그림 1-11 인접 학문 간의 관계

1.5 학습을 위한 자원

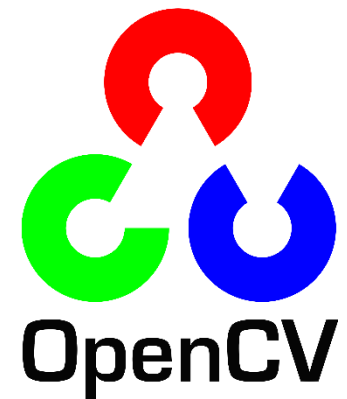
■ 웹 사이트(부록 B)

- Cvonline (<http://homepages.inf.ed.ac.uk/rbf/CVonline/>)
 - 컴퓨터 비전의 주제를 계층적으로 나누고, 그들 주제에 대한 설명을 위키피디아를 이용하여 제공한다.
 - 데이터베이스, 문헌, 소프트웨어, 교육을 위한 최신 자료도 풍부하다.
- VisionBib.Com (<http://www.visionbib.com/bibliography/contents.html>)
 - 컴퓨터 비전과 관련된 논문을 망라하여 제공한다.
 - 주제어, 저자, 시기, 학술지, 학술대회에 따라 검색이 가능하다.
- Computer Vision Online (<http://www.computervisiononline.com/>)
 - 소프트웨어, 데이터베이스, 그리고 책에 대한 최신 정보를 제공한다.
 - 컴퓨터 비전 분야에서 주목할 만한 뉴스도 알려준다.

2 Getting started with Open CV (Computer Vision)

■ What is Open CV??

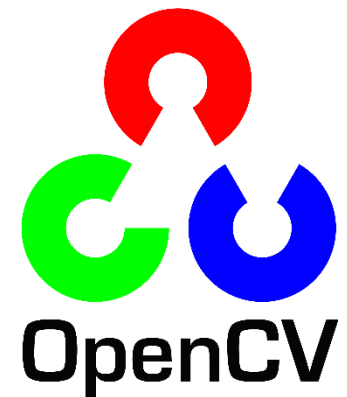
- An open source **Berkeley Software Distribution (BSD) licensed** computer vision library.
 - Patent-encumbered code isolated into “non-free” module.
 - SIFT, SURF, some of the Face Detectors, etc.
- Available on all major platforms
 - Android, iOS, Linux, Mac OS X, Windows
- Written primarily in C++
 - Bindings available for Python, Java, even MATLAB (in 3.0).
- Well documented at <http://docs.opencv.org>
- Source available at <https://github.com/Itseez/opencv>



2. Getting started with Open CV (Computer Vision)

■ History of Open CV

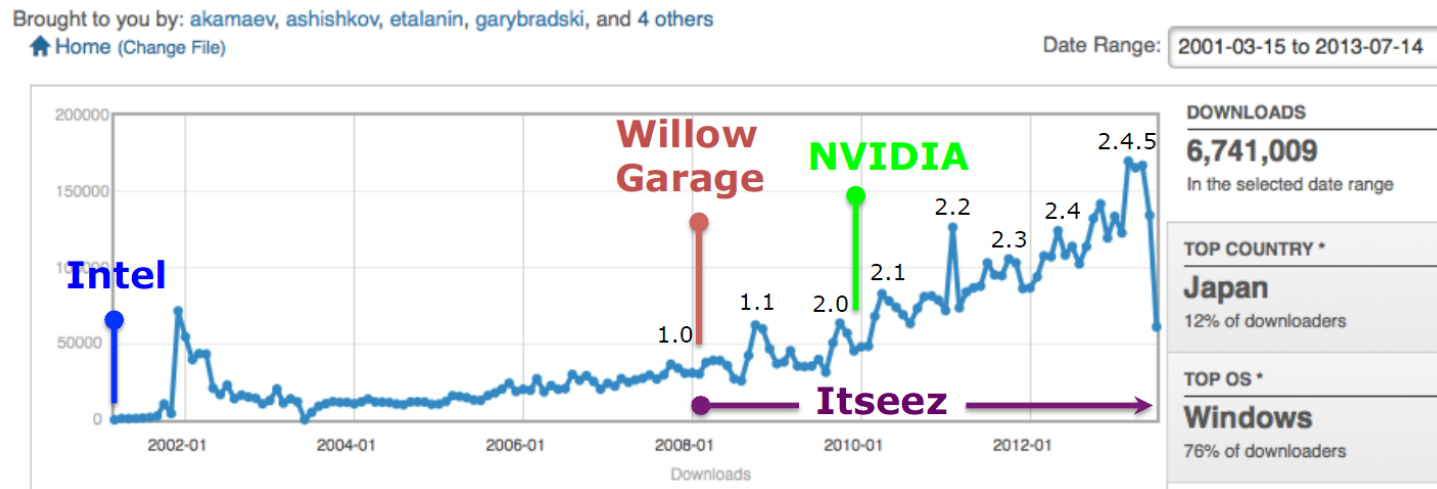
- Open CV started by Intel Research in 1998.
- Goals originally were:-
 - Advance vision research by providing not only open but also optimized code for basic vision infrastructure. No more reinventing the wheel.
- Disseminate vision knowledge by providing a common infrastructure that developers could build on, so that code would be more readily readable and transferable.
- **Advance vision-based commercial applications** by making portable, performance optimized code available **for free with a license** that did not require to be open or free themselves.
- Originally released at **CVPR (Computer Vision and Pattern Recognition) 2000**.



2. Getting started with Open CV (Computer Vision)

■ Open CV then and now.....

- Version 1.0 was released in 2006.
- In 2008 obtained corporate support from Willow Garage (Robotics Company).
- Open CV 2 was released in 2009.
 - Included major changes for C++ (mostly C beforehand).
- In 2012 support for Open CV was taken over by a non-profit foundation **OpenCV.org**.
- Open CV 3 was released in 2014. Now, Open CV 4.5.3 released.....!!!!
 - Seems to be under corporate support from Itseez.
 - More on these changes soon.



2. Getting started with Open CV (Computer Vision)

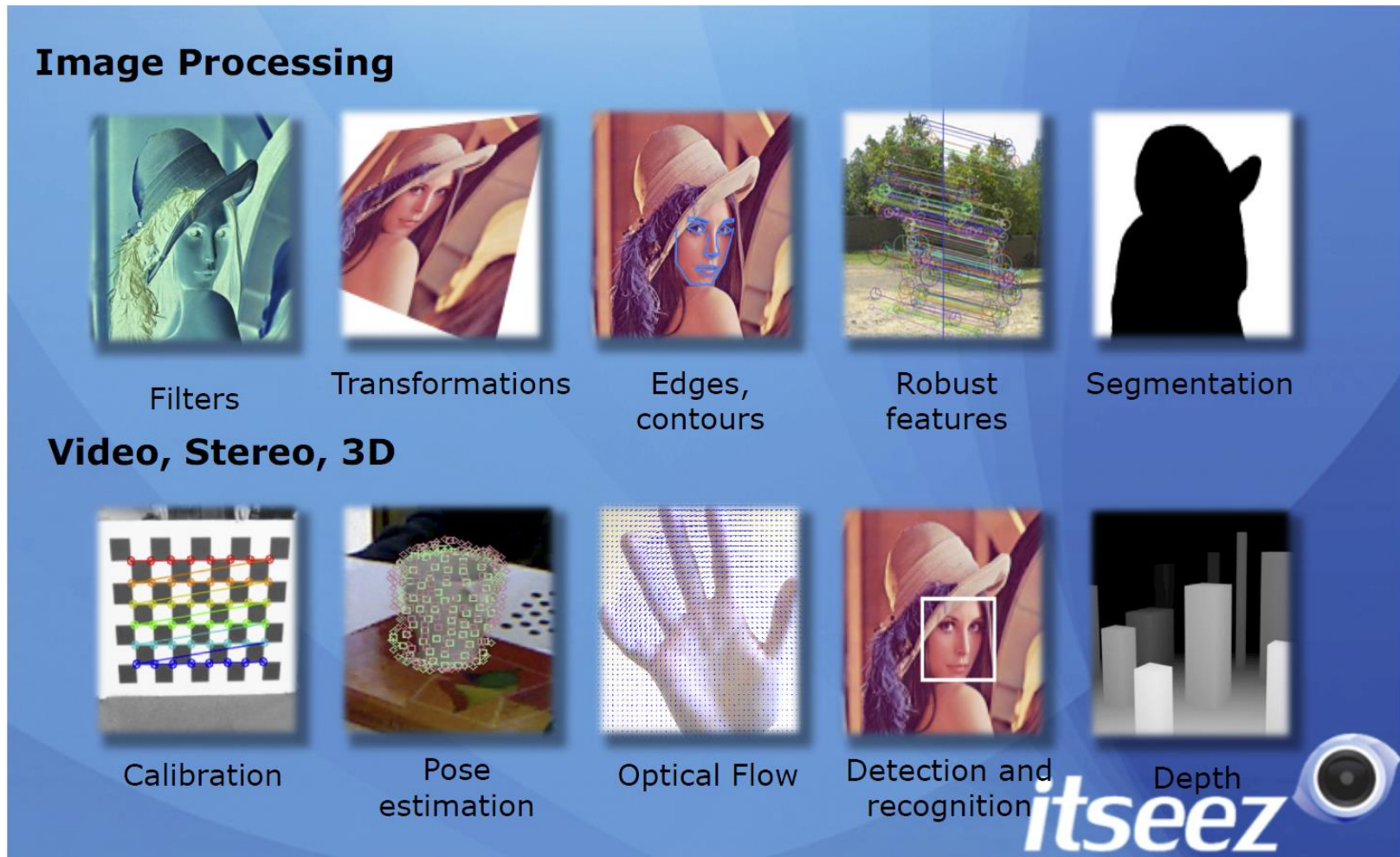
■ What can Open CV do (in 2018)?

Image Processing

Filters Transformations Edges, contours Robust features Segmentation

Video, Stereo, 3D

Calibration Pose estimation Optical Flow Detection and recognition Depth



itseez

- Taken from Open CV 3.0 latest news and the roadmap.

2. Getting started with Open CV (Computer Vision)

■ What can Open CV do, now in 2021?



OpenCV 4.5.3 has been released!

Release highlights

- Added support for multiple *highgui* backends
- Implemented HW-accelerated decoding and encoding using FFmpeg
- [DaSiamRPN](#) tracker is implemented as OpenCV algorithm
- Supported OpenVINO 2021.4 release in *dnn* module
- G-API module: improved Python bindings and DL inference capabilities, added instrumentation support
- 1D Barcode support

More details can be found in the [Changelog](#).

2. Getting started with Open CV (Computer Vision)

■ Open CV 3.0 ~

Migration is relatively smooth from 2.4

- Mostly cleanings
 - Refined **C++ API**
 - **Python API** available
 - Use **cv::Algorithm** everywhere
- API changes
 - C API will be marked as deprecated
 - Old **Python API** will be deprecated
 - Monstrous modules will be split into micromodules – Extra modules
- Sufficiently **improved CUDA and OpenCL** modules
 - Mobile CUDA support
 - Universal OpenCL binaries (CPU, GPU)
- Hardware Abstraction Layer (HAL)
 - IPP, FastCV-like low-level API to accelerate Open CV on different HW.

2. Getting started with Open CV (Computer Vision)

■ Open CV 3.x~

- Open-source NEON optimizations
 - iOS, Android, Embedded.
 - Latest NEWS - 40 NEON optimized functions after 4.0.
- Check out the transition guide.

2. Getting started with Open CV (Computer Vision)

■ Key Open CV Classes

<code>Point_</code>	Template 2D point class
<code>Point3_</code>	Template 3D point class
<code>Size_</code>	Template size (width, height) class
<code>Vec</code>	Template short vector class
<code>Matx</code>	Template small matrix class
<code>Scalar</code>	4-element vector
<code>Rect</code>	Rectangle
<code>Range</code>	Integer value range
<code>Mat</code>	2D or multi-dimensional dense array (can be used to store matrices, images, histograms, feature descriptors, voxel volumes etc.)
<code>SparseMat</code>	Multi-dimensional sparse array
<code>Ptr</code>	Template smart pointer class

2. Getting started with Open CV (Computer Vision)

Matrix Basics

Create a matrix

```
Mat image(240, 320, CV_8UC3);
```

[Re]allocate a pre-declared matrix

```
image.create(480, 640, CV_8UC3);
```

Create a matrix initialized with a constant

```
Mat A33(3, 3, CV_32F, Scalar(5));  
Mat B33(3, 3, CV_32F); B33 = Scalar(5);  
Mat C33 = Mat::ones(3, 3, CV_32F)*5.;  
Mat D33 = Mat::zeros(3, 3, CV_32F) + 5.;
```

Create a matrix initialized with specified values

```
double a = CV_PI/3;  
Mat A22 = (Mat_<float>(2, 2) «  
cos(a), -sin(a), sin(a), cos(a));  
float B22data[] = {cos(a), -sin(a), sin(a), cos(a)};  
Mat B22 = Mat(2, 2, CV_32F, B22data).clone();
```

3. OpenCV Modules (Packages)

OpenCV Libs.

<https://docs.opencv.org/3.1.0/>

Main modules

- core. **Core functionality**
- imgproc. **Image processing**
- imgcodecs. **Image file reading and writing**
- videoio. **Media I/O**
- highgui. **High-level GUI**
- video. **Video Analysis**
- calib3d. **Camera Calibration and 3D Reconstruction**
- features2d. **2D Features Framework**
- objdetect. **Object Detection**
- ml. **Machine Learning**
- flann. **Clustering and Search in Multi-Dimensional Spaces**
- photo. **Computational Photography**
- stitching. **Images stitching**
- cudaarithm. **Operations on Matrices**
- cudabgsegm. **Background segmentation**
-
-
-

Extra modules

- aruco. **ArUco Marker Detection**
- bgsegm. **Improved Background-Foreground Segmentation Methods**
- bioinspired. **Biologically inspired vision models and derivated tools**
- ccalib. **Custom Calibration Pattern for 3D reconstruction**
- cvv. **GUI for Interactive Visual Debugging of Computer Vision Programs**
- datasets. **Framework for working with different datasets**
- dnn. **Deep Neural Network module**
- dpm. **Deformable Part-based Models**
- face. **Face Recognition**
-
-
-
- xfeatures2d. **Extra 2D Features Framework**
- ximgproc. **Extended Image Processing**
- xobjdetect. **Extended object detection**
- xphoto. **Additional photo processing algorithms**

3. OpenCV Modules (Packages)

■ Main modules

- **core** - a compact module defining basic data structures, including the dense multi-dimensional array **Mat** and basic functions used by all other modules.
- **imgproc** - an image processing module that includes linear and non-linear image filtering, geometrical image transformations (resize, affine and perspective warping, generic table-based remapping), color space conversion, histograms, and so on.
- **video** - a video analysis module that includes motion estimation, background subtraction, and object tracking algorithms.
- **calib3d** - basic multiple-view geometry algorithms, single and stereo camera calibration, object pose estimation, stereo correspondence algorithms, and elements of 3D reconstruction.
- **features2d** - salient feature detectors, descriptors, and descriptor matchers.
- **objdetect** - detection of objects and instances of the predefined classes (for example, faces, eyes, mugs, people, cars, and so on).
- **highgui** - an easy-to-use interface to simple UI capabilities.
- **videoio** - an easy-to-use interface to video capturing and video codecs.

⋮

3. OpenCV Modules (Packages)

- **Extra Modules**
 - aruco. **ArUco Marker Detection**
 - bgsegm. **Improved Background-Foreground Segmentation Methods**
 - bioinspired. **Biologically inspired vision models and derivated tools**
 - ccalib. **Custom Calibration Pattern for 3D reconstruction**
 - **⋮**
 - stereo. **Stereo Correspondance Algorithms**
 - structured_light. **Structured Light API**
 - surface_matching. **Surface Matching**
 - text. **Scene Text Detection and Recognition**
 - tracking. **Tracking API**
 - xfeatures2d. **Extra 2D Features Framework**
 - ximgproc. **Extended Image Processing**
 - xobjdetect. **Extended object detection**
 - xphoto. **Additional photo processing algorithms**

다음 수업 준비 (실습 준비) 내용

■ OpenCV 다운로드하여 준비하기

- OpenCV 공식 사이트: <https://opencv.org/>
 - Version: 4.7.x self-extraction file (not source file) (4.7.x 이상 version 권장)
- OpenCV extra module 소스: https://github.com/opencv/opencv_contrib/releases
 - Version 4.7.x zip 파일 다운로드 (4.7.x 이상 version 권장)

(윈도우용으로 다운로드하여 여러분들 노트북 또는 USB에 카피하여 준비해 오시기 바랍니다.)

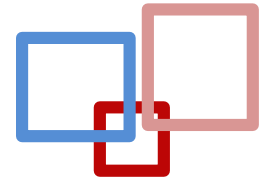
■ Cmake 설치하기

- <https://cmake.org/>

■ Visual Studio 2017 community 이후 버전 노트북 또는 시스템에 설치해 오기

- Win32 console 응용 패키지 사용할 예정임

COMPUTER VISION 비전 프로그래밍



Thank you and question?

