
파이썬

OpenCV 프로그래밍 1

컴퓨터 비전이란?



WIKIPEDIA
The Free Encyclopedia

컴퓨터 비전은 컴퓨터가 디지털 이미지 또는 비디오에서 높은 수준의 이해를 위한 결과를 얻을 수 있는 방법을 다루는 과학 분야입니다. 공학적 관점에서 볼 때, 인간의 시각 시스템이 할 수 있는 작업을 이해하고 자동화하려고 연구하는 분야입니다.

컴퓨터 비전 작업에는 디지털 이미지를 획득, 처리, 분석 및 이해하는 방법, 수치 또는 기호 정보(예: 의사 결정 형태)를 생성하기 위해 현실 세계에서 고차원 데이터를 추출하는 방법이 포함됩니다. 이러한 맥락에서 이해한다는 것은 시각 이미지(망막의 입력)를 사고 과정에 이치에 맞고 적절한 행동을 유도할 수 있는 세계에 대한 묘사로 변형시키는 것을 의미한다. 이러한 이미지 이해는 기하학, 물리학, 통계 및 학습 이론의 도움으로 구성된 모델을 사용하여 이미지 데이터에서 기호 정보를 분리하는 것으로 볼 수 있습니다.

컴퓨터 비전의 과학적 분야는 이미지에서 정보를 추출하는 인공지능 시스템 이론과 관련이 있습니다. 이미지 데이터는 비디오 시퀀스, 여러 카메라의 뷰, 3D 스캐너의 다차원 데이터 또는 의료 스캐닝 장치와 같은 다양한 형태를 취할 수 있습니다. 컴퓨터 비전의 기술 분야는 컴퓨터 비전 시스템 구축에 이론과 모델을 적용하려고 합니다.

컴퓨터 비전의 하위 영역에는 장면 재구성, 객체 감지, 이벤트 감지, 비디오 추적, 객체 인식, 3D 포즈 추정, 학습, 인덱싱, 동작 추정, 시각적 서빙, 3D 장면 모델링 및 이미지 복원이 포함됩니다.

컴퓨터 비전이란?

컴퓨터 비전(computer vision)은 디지털 이미지와 동영상 속 사물을 식별하는 기술입니다. 생명체는 시각 피질로 이미지를 처리하기 때문에, 많은 연구자들이 포유류의 시각 피질 구조를 모델로 삼아 이미지 인식을 수행할 신경망을 설계했습니다. 이러한 생물학적 연구는 1950년대로 거슬러 올라갑니다.

지난 20년간 컴퓨터 비전 기술은 놀랍게 발전했습니다. 아직 완벽하지는 않지만 컴퓨터 비전 시스템 중에는 99% 정확성을 보이는 것도 있고, 모바일 장치에서 제법 잘 실행되는 것도 있습니다.

시각 구현을 위한 신경망 분야의 돌파구는 1998년 얀 르쿤(Yann LeCun)의 르넷-5(LeNet-5)였습니다. 르넷-5는 7단계 콘볼루션 신경망(convolutional neural network, 이하 CNN)으로서 32x32픽셀 이미지로 디지털화된 손글씨 숫자를 인식할 수 있었습니다. 이보다 더 높은 해상도의 이미지를 분석하려면 르넷-5 네트워크가 더 많은 신경세포와 계층으로 확장해야 했습니다.

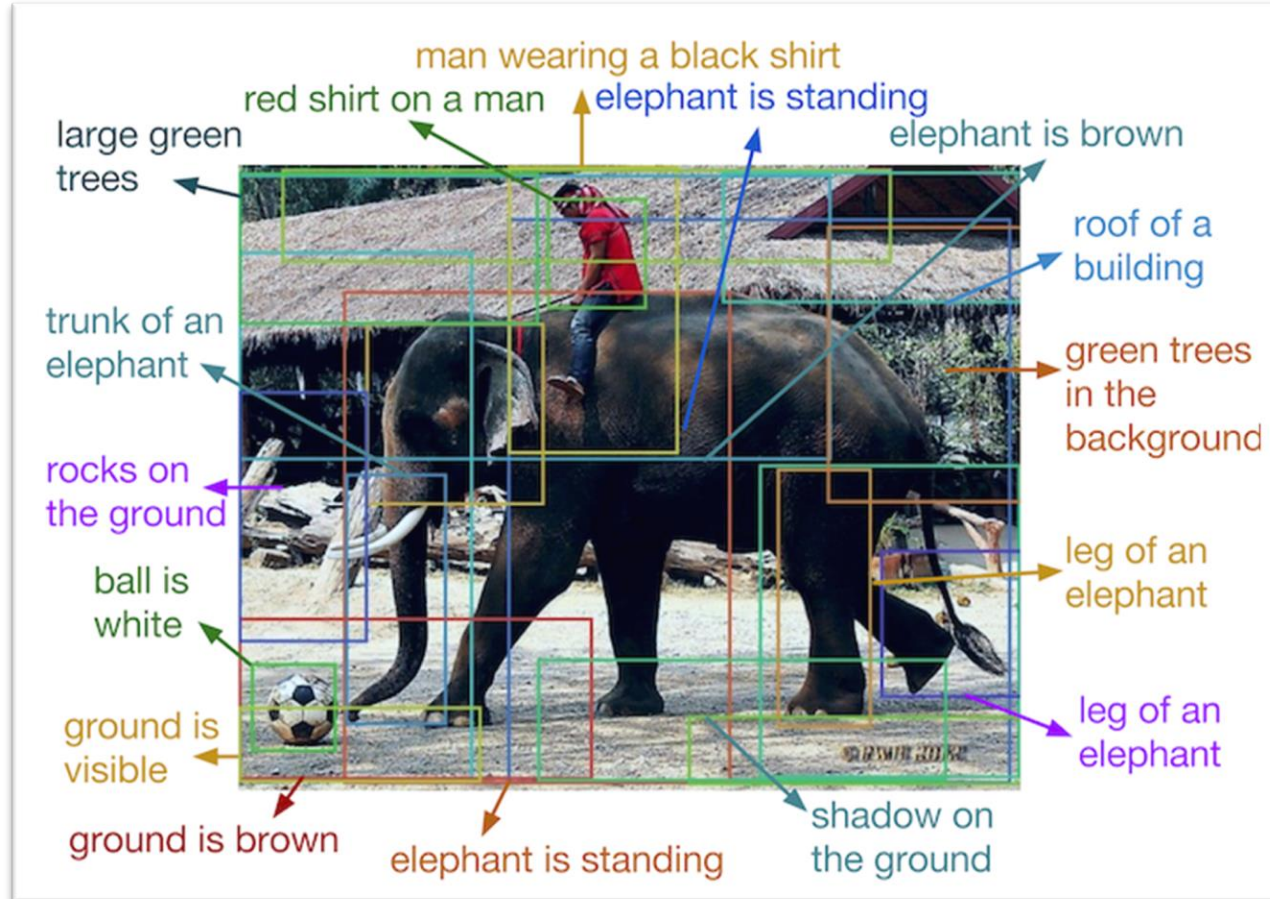
오늘날 가장 뛰어난 이미지 분류 모델의 경우, 다양한 사물을 컬러 HD 해상도로 식별 가능합니다. 순수 심층 신경망(deep neural networks, DNN)뿐만 아니라 혼합형 시각 모델도 가끔 사용됩니다. 혼합형 모델은 특정 하위 작업을 수행하는 고전적인 머신러닝 알고리즘에 딥 러닝을 결합한 것입니다.

시각 구현에 있어서 기본적인 이미지 분류 이외의 다른 문제들(예: 국소화로 이미지 분류, 사물 탐지, 사물 구획화, 이미지 형식 전송, 이미지 컬러화, 이미지 재구성, 이미지 초해상, 이미지 합성)은 딥 러닝으로 해결되었습니다.

컴퓨터 비전이란?



WIKIPEDIA
The Free Encyclopedia

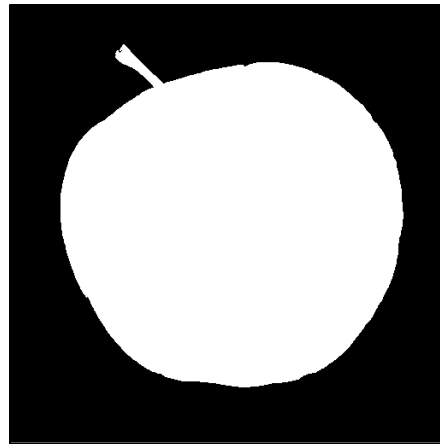


DenseCap "고밀도 캡션" 소프트웨어의 출력, 코끼리를 타는 남자의 사진을 분석

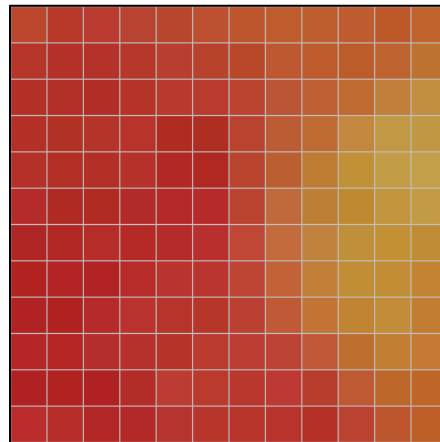
컴퓨터 비전이란?



사과?



둥글다?



빨간색이다?



꼭지 모양으로 구분?



사과가 몇 개??

컴퓨터 비전 vs. 영상 처리

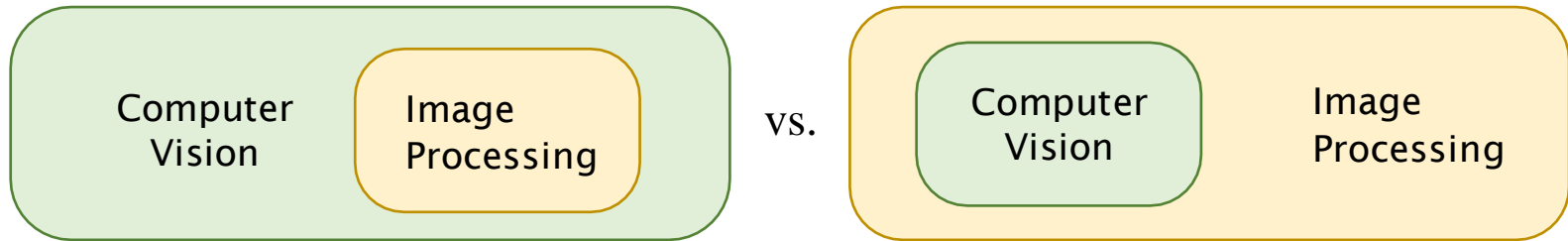


Image processing is processing of images using mathematical operations by using any form of signal processing for which the input is an image, a series of images, or a video, such as a photograph or video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image.

Computer vision, on the other hand, is often considered *high-level image processing* out of which a machine/computer/software intends to decipher the physical contents of an image or a sequence of images.

https://en.wikipedia.org/wiki/Image_processing

컴퓨터 비전 작동

컴퓨터 비전은 카메라, 에지 기반 또는 클라우드 기반 컴퓨팅, 소프트웨어 및 인공 지능(AI)을 결합하여 시스템이 사물을 "확인"하고 식별할 수 있게 합니다. 다양한 환경에서 가치를 발휘하는 컴퓨터 비전 시스템은 사물과 사람을 빠르게 인식하고, 고객 통계 자료를 분석하고, 공산품을 검사하는 등의 작업을 수행할 수 있습니다.

컴퓨터 비전은 딥 러닝을 사용하여 이미지 처리 및 분석 시스템을 안내하는 신경망을 형성합니다. 충분히 훈련된 컴퓨터 비전 모델은 사물을 인식하고 사람을 감지하거나 인식하며 움직임도 추적할 수 있습니다.

컴퓨터 비전 알고리즘은 주로 CNN에 의존합니다. CNN은 보통 컨볼루션 계층, 풀링 계층, ReLU 계층, 전체 연결 계층, 손실 계층을 이용해 시각 피질을 비슷하게 구현해 냅니다. 컨볼루션 계층은 기본적으로 작은 중복 구역 여러 개의 적분을 취합니다. 풀링 계층은 비선형 다운 샘플링을 수행합니다. ReLU 계층은 비포화 활성화 함수($f(x) = \max(0, x)$)를 적용합니다.

전체 연결 계층에서는 신경세포가 이전 계층의 모든 활성화에 연결되어 있습니다. 손실 계층은 망 훈련으로 인해 예측 라벨과 진짜 라벨의 편차가 어떻게 분리해지는지 계산합니다. 이때 소프트맥스(Softmax) 또는 크로스엔트로피 손실(cross-entropy loss)을 분류에 활용합니다.

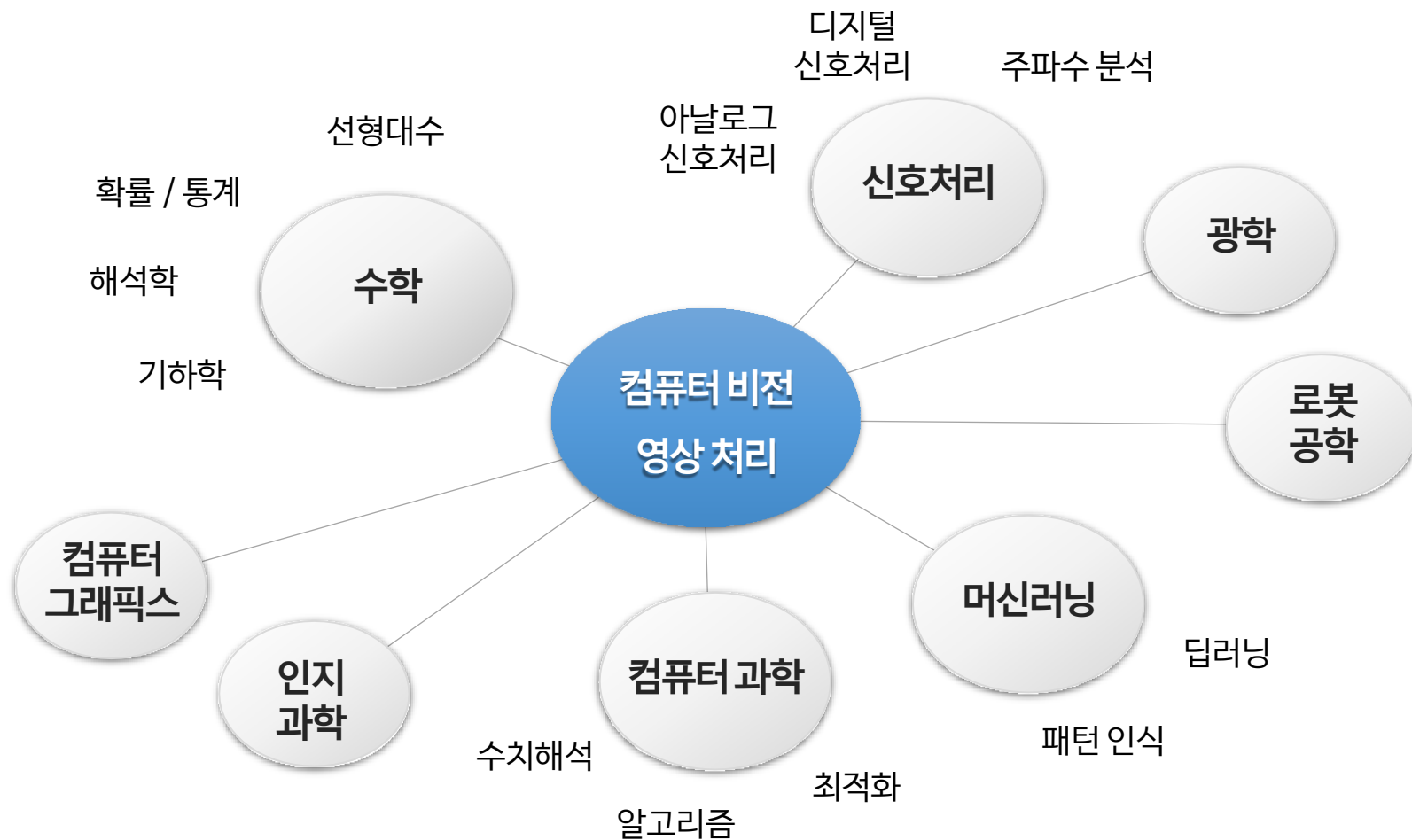
컴퓨터 비전 훈련 데이터집합

비전 모델 훈련에 유용한 공개 이미지 데이터집합이 많이 있습니다. 그 중에서 가장 단순하고 가장 오래된 MNIST에는 10개 등급의 손글씨 숫자 7만 개가 있습니다. 6만 개는 훈련용, 1만 개는 테스트용입니다. MNIST는 모델화 하기 쉬운 데이터 집합입니다. 심지어 가속 하드웨어가 없는 노트북으로도 할 수 있습니다. CIFAR-10과 패션-MNIST는 비슷한 10등급 데이터집합들입니다. SVHN(스트리트 뷰 가옥 번호)는 구글 스트리트 뷰에서 추출한 실제 가옥 번호 이미지 60만 개를 모아 놓은 것입니다.

COCO는 사물 탐지와 구획화, 자막 삽입을 위한 대규모 데이터집합입니다. 80가지 사물 범주의 이미지 33만 개가 있습니다. 이미지넷(ImageNet)에는 상자와 라벨 이미지, 워드넷(WordNet)의 10만 개 이상의 구절 일러스트 등이 포함되어 있습니다. 오픈 이미지(Open Images)에는 이미지 URL 약 9백만 개와 라벨 약 5천 개가 있습니다.

구글, 애저, AWS는 모두 방대한 이미지 데이터베이스로 훈련된 자체 비전 모델이 있습니다. 그대로 사용해도 되고 전이 학습(transfer learning)을 실행해 본인의 이미지 데이터베이스에 맞게 적응시킬 수도 있습니다. 전이 학습은 이미지넷과 오픈 이미지에 기반한 모델을 이용해 수행할 수도 있습니다. 전이 학습의 장점은 처음부터 모델을 구축하는 것(몇 주 소요)에 비해 훨씬 빠르고(몇 시간 소요)와 정확도가 더 높은 모델을 얻을 수 있다는 점입니다. 최상의 결과를 얻으려면 여전히 라벨당 1,000개의 이미지가 필요하지만, 가끔은 라벨당 10개 정도의 이미지만으로도 충분한 경우가 있습니다.

컴퓨터 비전 관련 분야



컴퓨터 비전 프레임워크와 모델 I

컴퓨터 비전은 대부분의 딥 러닝 프레임워크에서 지원됩니다. 예를 들면, 파이썬 기반 프레임워크들인 텐서플로우(TensorFlow, 프로덕션을 위한 최고의 선택), 파이토치(PyTorch, 학문 연구에 최고의 선택), 엠엑스넷(MXNet, 아마존이 선택한 프레임워크) 등이 있습니다. 오픈 CV(OpenCV)는 컴퓨터 비전에 특화된 라이브러리입니다. 실시간 비전 애플리케이션에 치우쳐 있으며 MMX 및 SSE 명령어가 이용 가능하면 이를 활용합니다. CUDA, 오픈CL, 오픈GL, 벌컨(Vulkan)을 이용한 가속도 지원됩니다.

아마존 레코그니션(Amazon Rekognition)은 물체, 사람, 문자, 장면, 활동을 식별할 수 있는 이미지 및 동영상 분석 서비스로서 안면 분석 기능과 사용자 정의 라벨이 제공됩니다. 구글 클라우드 비전 API(Google Cloud Vision API)는 미리 훈련을 거친 이미지 분석 서비스입니다. 물체 및 사람 얼굴 탐지 기능과 인쇄 및 손글씨 문자 판독 기능이 있으며 사용자의 이미지 카탈로그 내에 메타데이터를 구축할 수 있습니다. 구글 오토ML 비전은 사용자가 스스로 정의한 이미지 모델을 훈련하게 해 줍니다. 아마존 레코그니션 커스텀 라벨과 구글 오토ML 비전은 둘 다 전이 학습을 수행합니다.

마이크로소프트 컴퓨터 비전 API(Microsoft Computer Vision API)는 25개 언어로 된 라벨이 붙은 사물 1만 개 카탈로그에서 사물을 식별할 수 있습니다. 식별된 사물에 경계 상자도 반환합니다. 애저 페이스 API(Azure Face API)는 사람 얼굴 탐지를 수행하여 이미지 내의 얼굴과 특성을 인식합니다. 사람 식별도 수행하여 개인 보관소에 있는 최대 1백만 명 중 일치하는 사람 한 명을 찾아냅니다. 인지 감정 인식도 수행합니다. 페이스 API는 클라우드에서 또는 컨테이너 엣지에서 실행 가능합니다.

컴퓨터 비전 프레임워크와 모델 II

IBM 왓슨 비주얼 레코그니션(IBM Watson Visual Recognition)은 미리 훈련된 모델로부터 이미지를 분류할 수 있습니다. 또한, 사용자 정의 이미지 모델을 전이 학습으로 훈련하게 해 줍니다. 사물 집계로 사물 탐지를 수행하고 시각 검사 훈련을 할 수 있게도 해 줍니다. 왓슨 시각 인식은 클라우드에서 실행할 수 있으며 코어ML을 사용해 iOS 장치에서 실행할 수도 있습니다.

데이터 분석 패키지 매트랩(Matlab)은 머신 러닝 및 딥 러닝을 이용한 이미지 인식을 수행할 수 있습니다. 선택 사용할 수 있는 컴퓨터 비전 툴박스가 있고 오픈CV와 통합 가능합니다.

컴퓨터 비전 모델은 르넷-5 이후 많은 발전을 했으며 대부분 CNN입니다. 예를 들면, 알렉스 넷(AlexNet)(2012), VGG16/옥스포드 넷(OxfordNet)(2014), 구글르 넷(GoogLeNet)/인셉션 V1(InceptionV1)(2014), 레즈넷50(Resnet50)(2015), 인셉션V3(2016), 모바일넷(MobileNet)(2017-2018) 등이 있습니다. 모바일넷 시각 신경망은 모바일 장치를 염두에 두고 설계되었습니다.

애플 비전(Apple Vision) 프레임워크는 사람 얼굴 및 안면 표지점 탐지와 문자 탐지, 바코드 인식, 이미지 등록, 일반 특징 추적을 수행합니다. 분류나 사물 탐지와 같은 작업에 사용자 정의 코어 ML 모델을 사용할 수 있게도 해 줍니다. iOS와 맥OS에서 실행됩니다. 구글 ML 키트 SDK는 유사한 기능이 있고 안드로이드 장치와 iOS 장치에서 실행됩니다. ML 키트는 자연어 API를 추가로 지원합니다.

컴퓨터 비전 연구 분야

□ 영상의 화질 개선



Filtering App



HDR



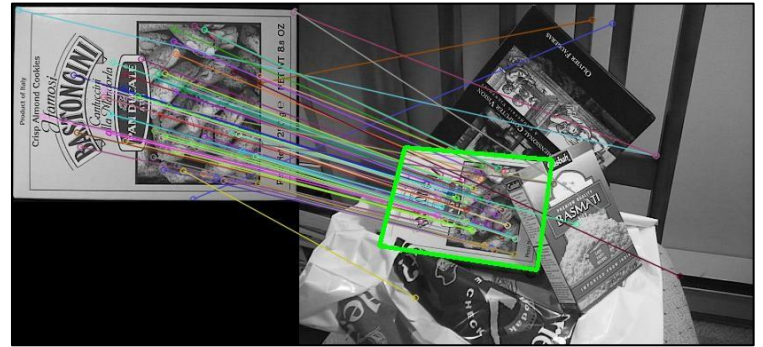
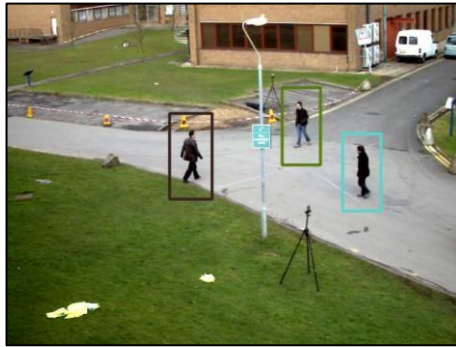
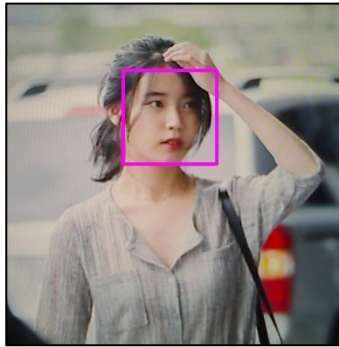
Image Noise Remove



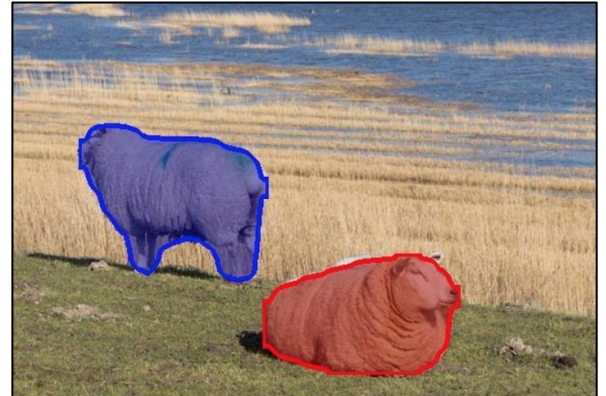
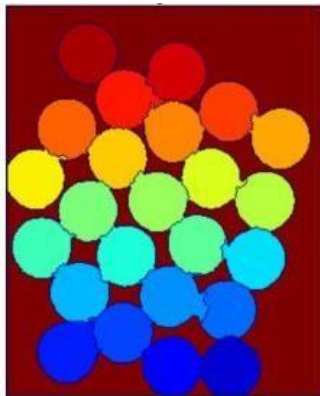
Super Resolution

컴퓨터 비전 연구 분야

□ 객체 검출(Object detection)

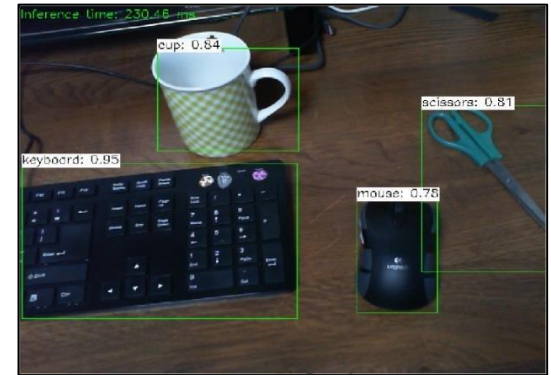
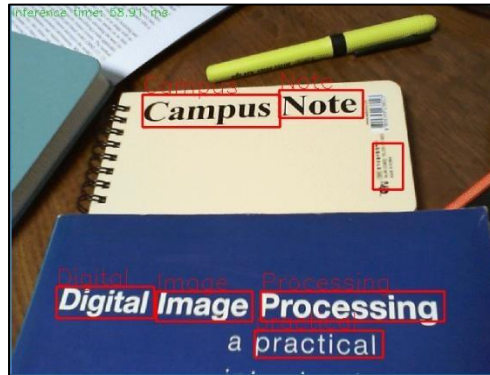


□ 분할(Segmentation)

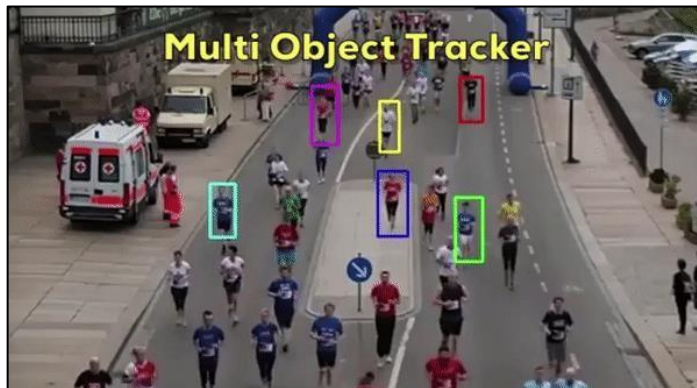


컴퓨터 비전 연구 분야

□ 인식(Recognition)



□ 움직임 정보 추출 및 추적



컴퓨터 비전 응용 분야

□ 머신 비전(machine vision)

- ✓ 공장 자동화: 제품의 불량 검사, 위치 확인, 측정 등
- ✓ 높은 정확도와 빠른 처리 시간 요구
- ✓ 조명, 렌즈, 필터, 실시간 (Real-time) 처리

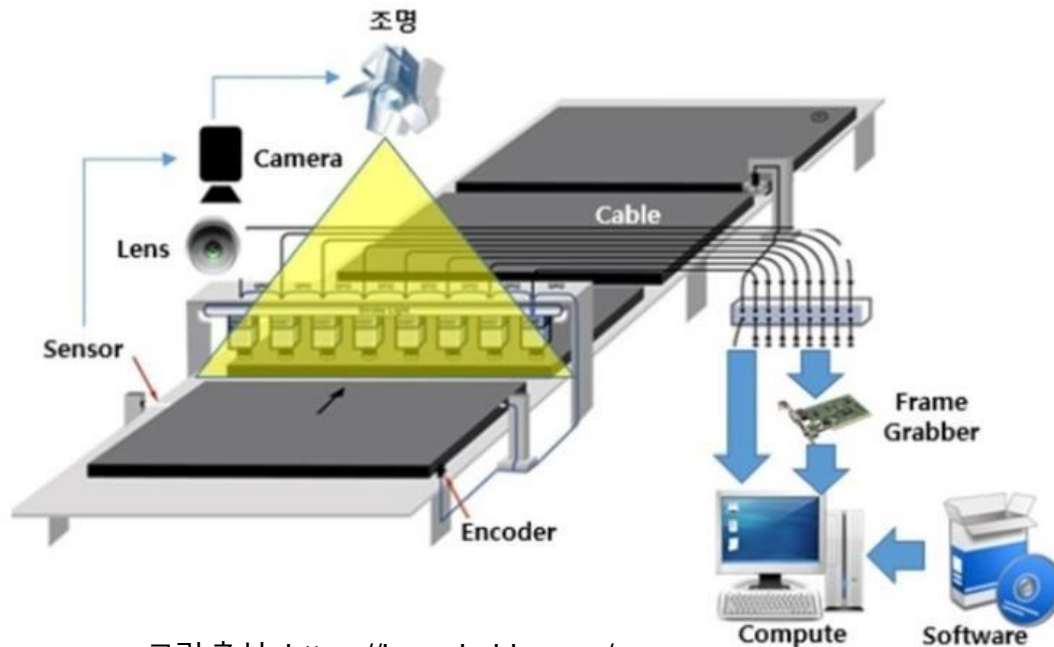


그림 출처: <https://laonple.blog.me/>

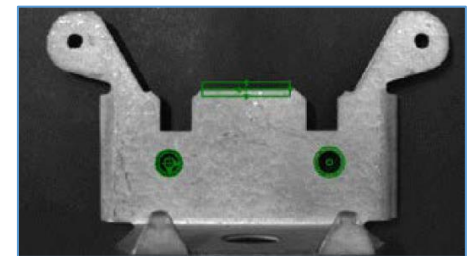


그림 출처: <http://www.cognex.com>

컴퓨터 비전 응용 분야

□ 인공지능(AI) 서비스

- ✓ 인공지능 로봇과 자율 주행 자동차
- ✓ 입력 영상을 객체와 배경으로 분할 ⑦ 객체와 배경 인식 ⑦ 상황 인식 ⑦ 로봇과 자동차의 행동 지시
- ✓ Computer Vision + Sensor Fusion + Deep Learning
- ✓ Amazon Go / 구글, 테슬라의 자율 주행 자동차



<https://www.youtube.com/watch?v=NrmMk1Myrxc>



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

컴퓨터 비전 애플리케이션 I

컴퓨터 비전은 완벽하지는 않지만 실제로 쓸 만한 정도는 되는 경우가 많습니다. 자율 주행 자동차에서의 활용이 대표적입니다.

원래 구글 자율 주행 차량 프로젝트로 시작한 웨이모(Waymo)는 공공 도로에서 700만 마일의 주행 시험을 거쳐 안전한 일상 운행이 가능하다고 합니다. 웨이모 승합차가 연루된 사건이 한 건 이상 있었는데 경찰에 따르면 소프트웨어의 문제는 아닌 것으로 알려졌습니다.

테슬라는 자율 주행 차량 모델이 3가지 있습니다. 2018년 테슬라 SUV 한 대가 자율 주행 모드로 달리던 중 사망 사고에 연루되었습니다. 사고 보고서에 따르면 운전자(사망)는 콘솔에서 경고가 여러 번 나왔음에도 불구하고 운전대에서 손을 떼 상태였으며, 콘크리트 장벽을 들이받을 상황이었는에도 운전자도 소프트웨어도 제동을 걸지 않았다고 합니다. 이 사건 이후에는 아예 운전자가 운전대에서 손을 떼지 못하도록 소프트웨어가 업그레이드되었습니다.

아마존 고(Go) 매장은 계산대가 없는 셀프서비스 소매점입니다. 쇼핑객이 물건을 집어 들거나 돌려 놓으면 매장 내 컴퓨터 비전 시스템으로 탐지됩니다. 스마트폰 앱을 통해 쇼핑객 신원 파악과 결제도 이뤄집니다. 아마존 고 소프트웨어에서 놓친 물건은 무료로 가져갈 수 있습니다. 가져간 물건의 계산이 잘못된 경우에 쇼핑객은 해당 물건을 표시해 두고 잘못 청구된 금액을 환불 받을 수 있습니다.

컴퓨터 비전 애플리케이션 II

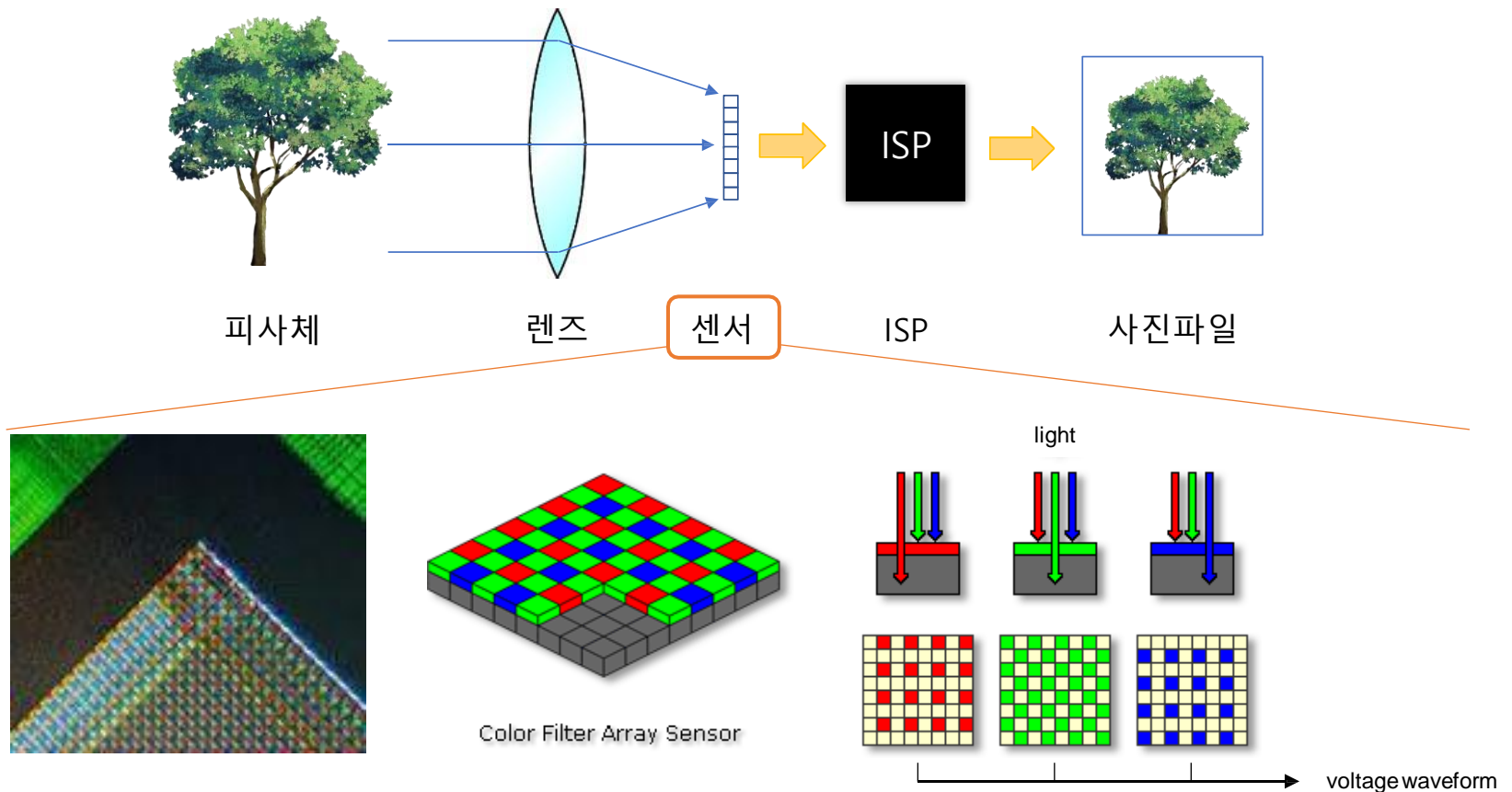
의료 분야에서는 병리학 슬라이드, 흉부 방사선 사진 등 의료 영상 시스템에서 구체적인 특징을 분류하기 위한 비전 애플리케이션들이 있습니다. 그 중에는 숙련된 인력과 비교했을 때 가치가 입증된 것도 몇 가지 있고 규제 승인을 받기에 충분한 것도 있습니다. 수술실이나 분만실에서 환자의 혈액 손실량을 추산하기 위한 실시간 시스템도 있습니다.

이 밖에도 유용한 비전 애플리케이션으로는 농업용(농업 로봇, 작물 및 토양 모니터링, 예측 분석), 은행 업무용(사기 간파, 서류 인증, 원격 입금), 산업 모니터링용(원격 유전, 현장 보안, 작업 활동) 등이 있습니다.

컴퓨터 비전 애플리케이션 중에는 논란이 되거나 심지어 심한 반대에 부딪히는 것도 있습니다. 그중 한 가지는 사람 얼굴 인식 기능입니다. 정부에서 사용할 경우 사생활 침해의 소지가 있고 훈련 편향이 생기는 경우가 많아서 백인이 아닌 사람의 얼굴은 식별을 잘못하는 경향이 있습니다. 합성 영상 제작도 논란과 반대의 대상입니다. 외설물이나 장난 등 가짜 사진 제작에 사용될 경우에는 단순히 조금 오싹한 단계를 넘어서기 때문입니다.

영상의 획득

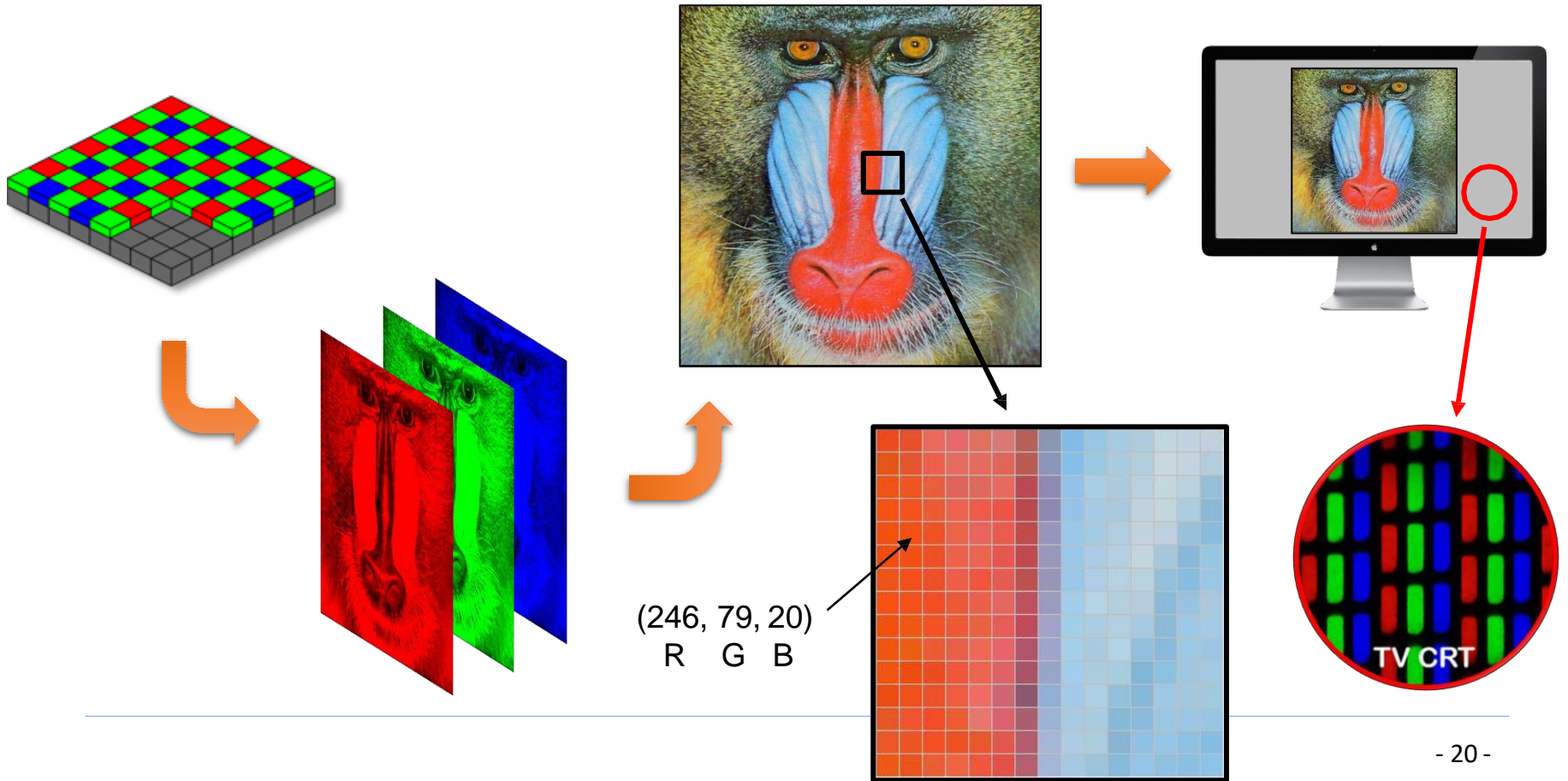
□ 디지털 카메라에서 영상의 획득 과정



영상의 표현 방법

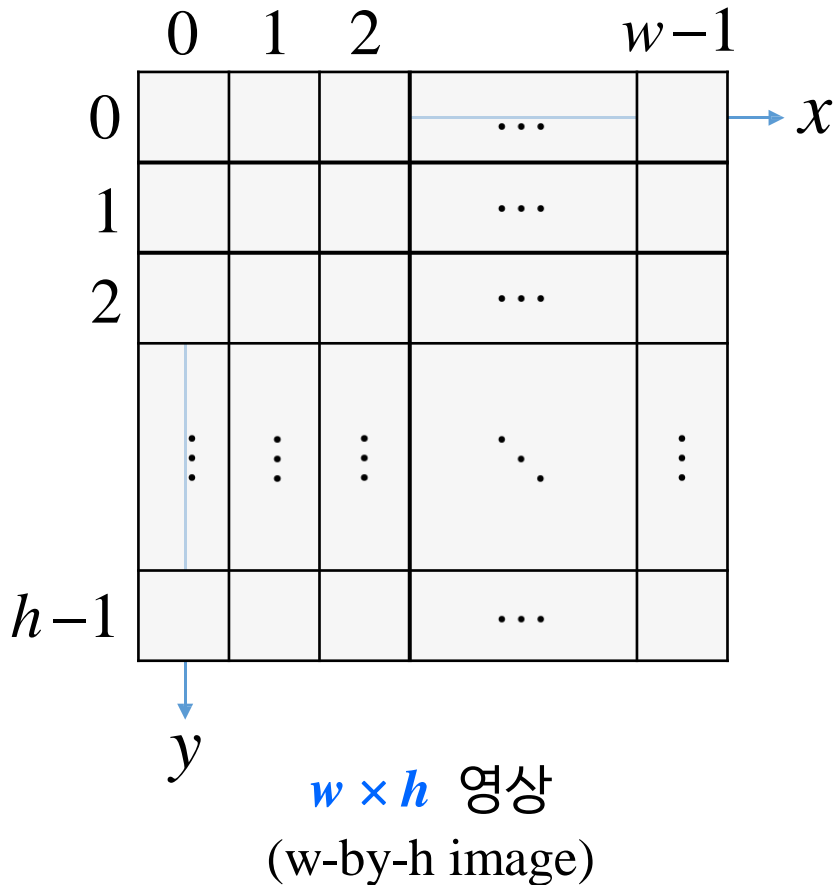
□ 영상(image)이란?

✓ 픽셀(pixel)이 바둑판 모양의 격자에 나열되어 있는 형태 (2차원 행렬)



영상의 표현 방법

□ 영상에서 사용되는 좌표계



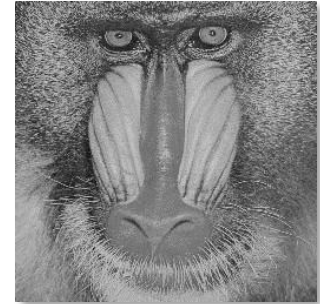
$$\mathbf{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,N} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M,1} & a_{M,2} & \cdots & a_{M,N} \end{bmatrix}$$

$M \times N$ 행렬
(m-by-n matrix)

영상의 표현 방법

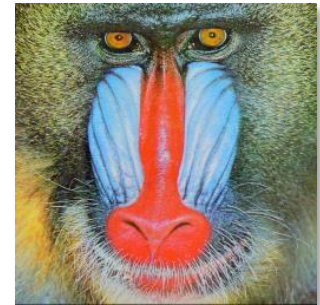
□ 그레이스케일(grayscale) 영상

- ✓ 색상 정보가 없이 오직 밝기 정보만으로 구성된 영상
- ✓ 밝기 정보를 256 단계로 표현



□ 트루컬러(truecolor) 영상

- ✓ 컬러 사진처럼 색상 정보를 가지고 있어서 다양한 색상을 표현할 수 있는 영상
- ✓ Red, Green, Blue 색 성분을 256 단계로 표현
 - ⑦ $256^3 = 16,777,216$ 색상 표현 가능

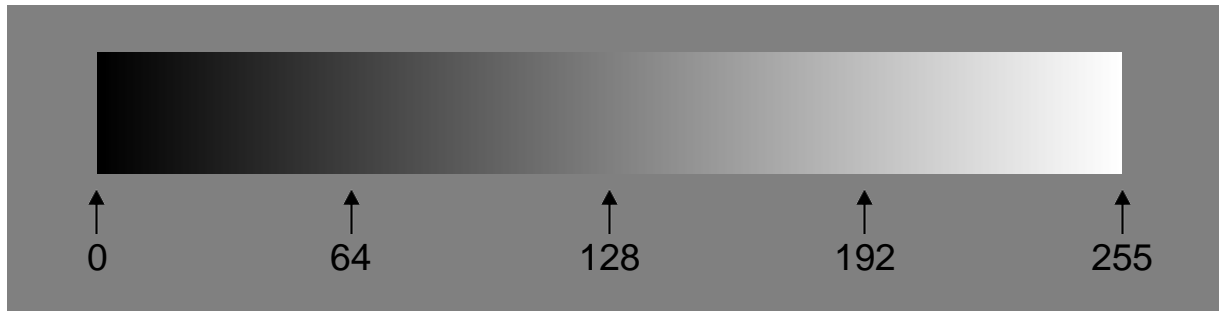


□ 픽셀(pixel)

- ✓ 영상의 기본 단위, picture element, 화소(畫素)

그레이스케일 영상과 컬러 영상

- 그레이스케일 값의 범위(Grayscale level)
 - ✓ 그레이스케일 영상에서 하나의 픽셀은 0부터 255 사이의 정수 값을 가짐.
 - 0 : 가장 어두운 밝기(검정색)
 - 255 : 가장 밝은 밝기(흰색)



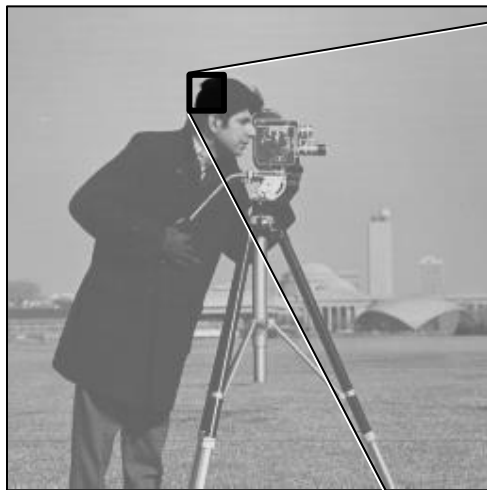
- ✓ C/C++ 에서 **unsigned char** 로 표현 (1 byte)

```
typedef unsigned char BYTE;           // Windows
typedef unsigned char uint8_t;        // Linux
typedef unsigned char uchar;          // OpenCV
```

- ✓ Python에서는? ➔ **numpy.uint8**

그레이스케일 영상과 컬러 영상

□ 그레이스케일 영상에서 픽셀 값 분포의 예

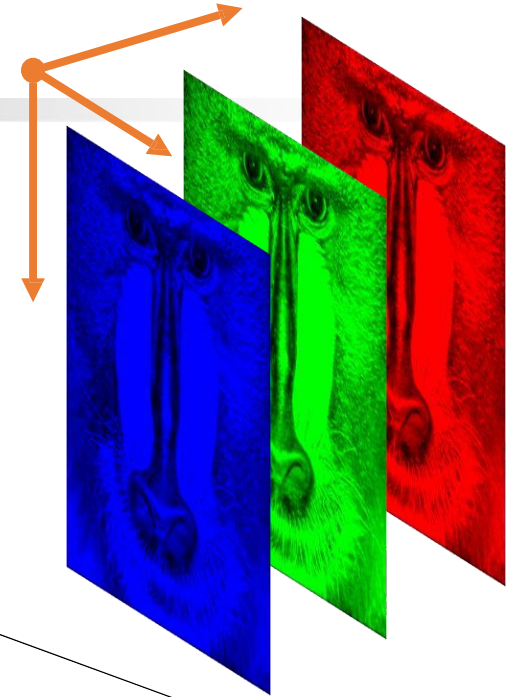
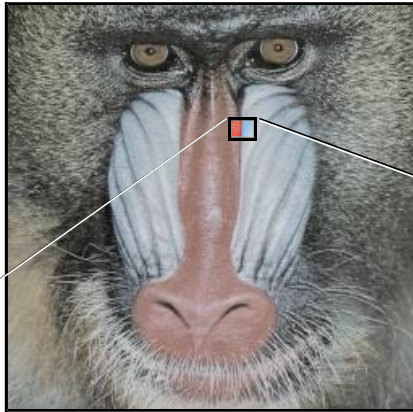


187	187	187	194	197	173	77	25	19	19
190	187	190	191	158	37	15	14	20	20
187	182	180	127	32	16	13	16	14	12
184	186	172	100	20	13	15	18	13	18
186	190	187	127	18	14	15	14	12	10
189	192	192	148	16	15	11	10	10	9
192	195	181	37	13	10	10	10	10	10
189	194	54	14	11	10	10	10	9	8
189	194	19	16	11	11	10	10	9	9
192	88	12	11	11	10	10	10	9	9

픽셀

그레이스케일 영상과 컬러 영상

□ 트루컬러 영상에서 픽셀 값 분포의 예



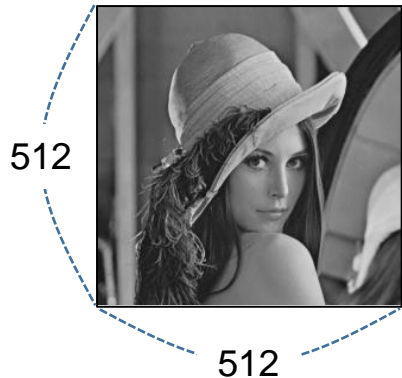
231	103	92	228	118	86	220	122	122	213	92	59	166	136	164	131	185	228	142	192	227	145	195	228
229	104	92	221	114	72	226	109	109	208	101	51	150	146	189	137	189	232	132	189	229	143	195	229
228	109	103	229	110	90	228	107	107	206	86	64	144	165	206	138	187	230	137	188	227	142	195	230
229	110	104	231	108	102	226	117	102	206	81	45	134	155	199	135	187	230	139	193	231	156	199	229
224	106	93	219	132	114	208	118	137	189	88	70	133	166	211	127	184	229	145	192	229	165	201	228
231	111	98	227	102	70	209	110	103	178	84	65	121	157	210	131	183	229	145	192	228	161	199	228

픽셀

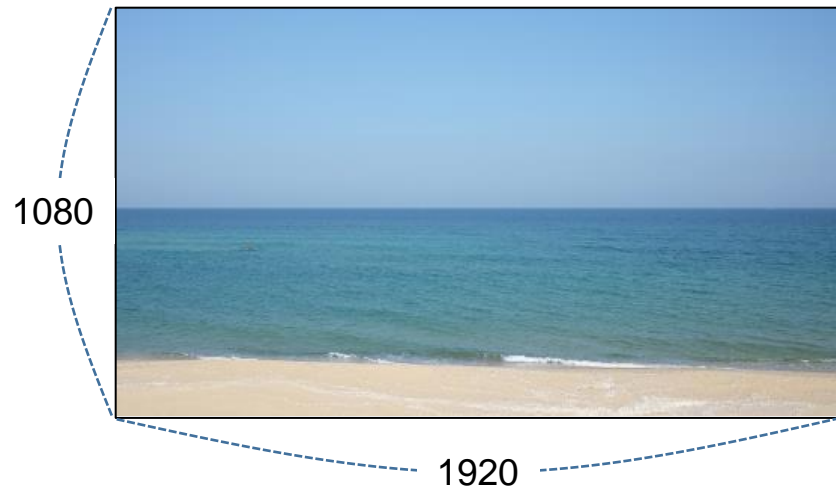
영상의 표현 방법

□ 영상 데이터 크기 분석

- ✓ 그레이스케일 영상: (가로 크기) × (세로 크기) Bytes
- ✓ 트루컬러 영상: (가로 크기) × (세로 크기) × 3 Bytes



$$512 \times 512 = 262144 \text{ Bytes}$$



$$1920 \times 1080 \times 3 = 6220800 \text{ Bytes}$$
$$\approx 6 \text{ MBytes}$$

영상 파일 형식 특징

BMP

- 픽셀 데이터를 압축하지 않고 그대로 저장, 파일 용량이 큰 편
- 파일 구조가 단순해서 별도의 라이브러리 도움 없이 직접 파일 입출력 프로그래밍 가능

JPG

- 주로 사진과 같은 컬러 영상을 저장하기 위해 사용
- 손실 압축(lossy compression)
- 압축률이 좋아서 파일 용량이 크게 감소, 디지털 카메라 사진 포맷으로 주로 사용됨

GIF

- 256 색상 이하의 영상을 저장
- 일반 사진을 저장 시 화질 열화가 심함
- 무손실 압축(lossless compression)
- 움직이는 GIF 지원

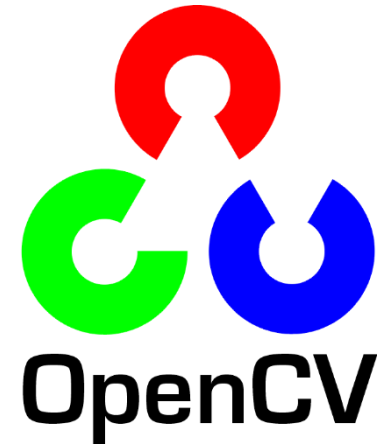
PNG

- Portable Network Graphics
- 무손실 압축 (트루컬러 영상도 무손실 압축)
- 알파 채널(투명도)을 지원

OpenCV 개요

☐ What is OpenCV?

- ✓ Open source
- ✓ Computer vision & machine learning
- ✓ Software library



<http://www.opencv.org/>

☐ Why OpenCV?

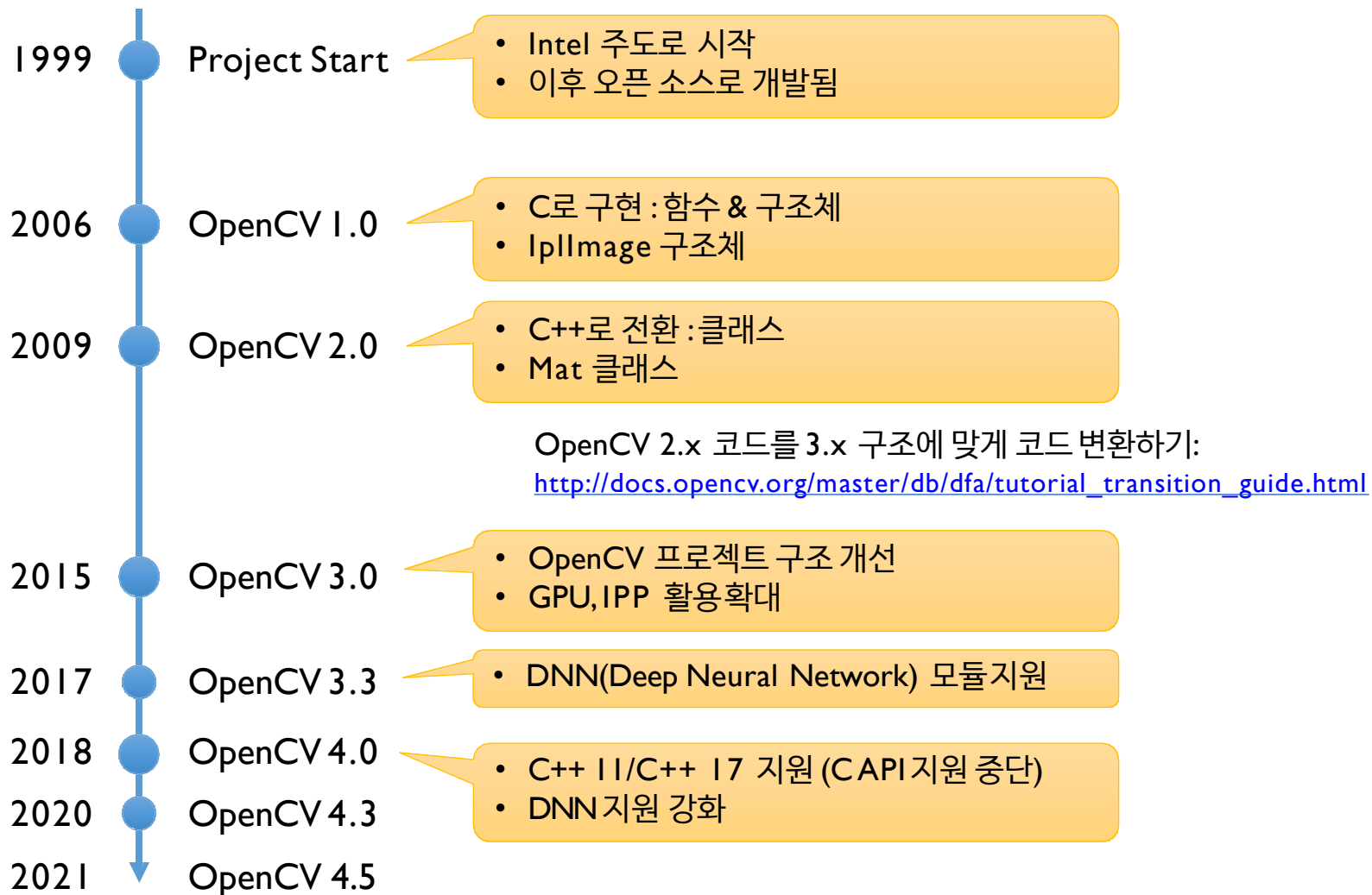
- ✓ BSD license ...Free for academic & commercial use
- ✓ Multiple interface ...C, C++, Python, Java, JavaScript, MATLAB, etc.
- ✓ Multiple platform ...Windows, Linux, Mac OS, iOS, Android
- ✓ Optimized ...CPU instructions, Multi-core processing, OpenCL, CUDA
- ✓ Popular ...More than 15 million downloads
- ✓ Usage ... Stitching streetview images, detecting intrusions, monitoring mine equipment, helping robots navigate and pick up objects, Interactive art, etc.

OpenCV 소개

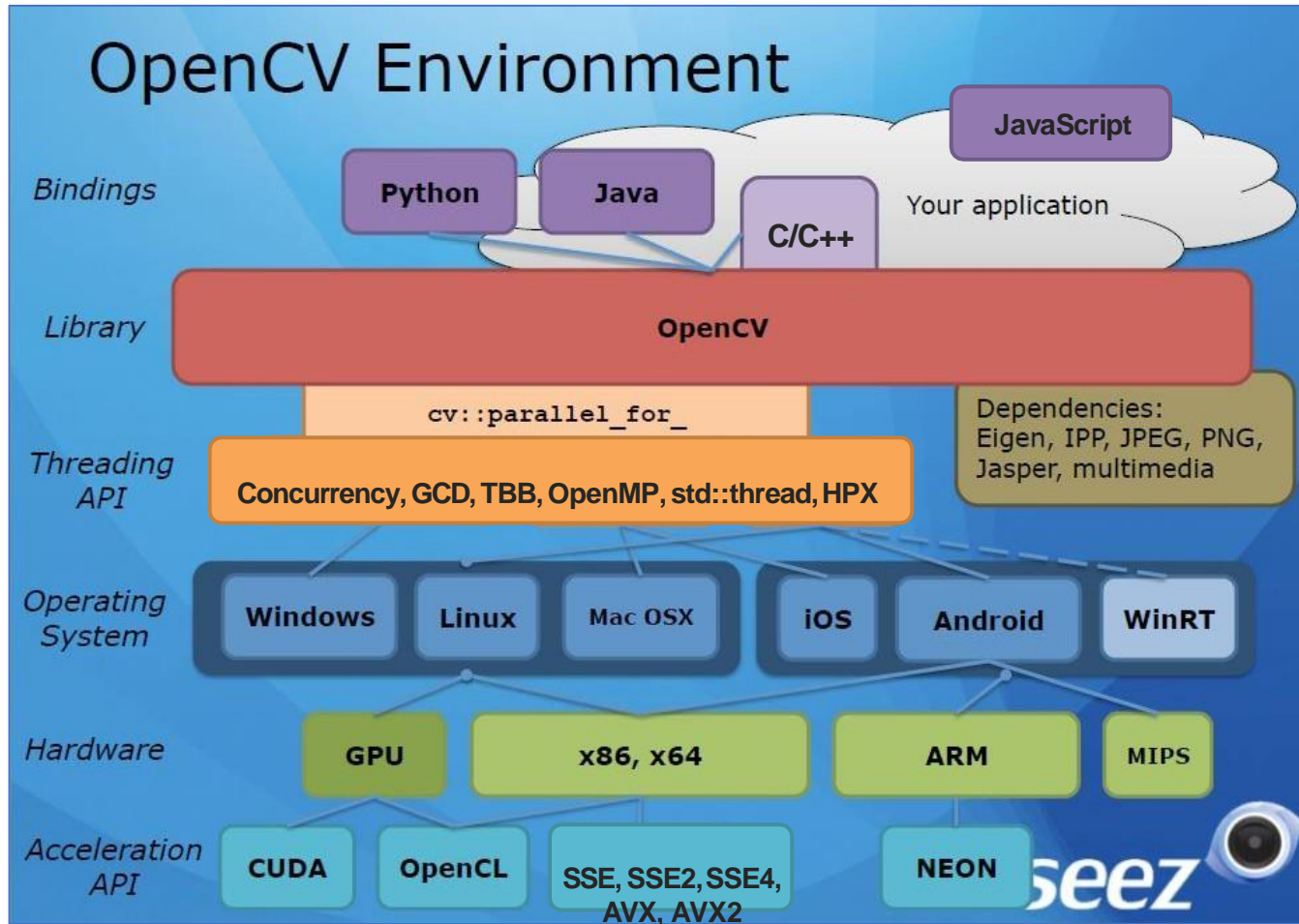
OpenCV(Open Source Computer Vision Library: <http://opencv.org>)
는 수백 개의 컴퓨터 비전 알고리즘을 포함하는 오픈 소스 라이브러리입니다.

C 기반 OpenCV 1.x API(C API는 더 이상 사용되지 않으며 OpenCV 2.4 릴리스 이후 "C" 컴파일러로 테스트되지 않음)와 달리 본질적으로 C++ API인 OpenCV 2.x API를 제공합니다.

OpenCV 역사



OpenCV 구성



OpenCV 모듈

OpenCV는 모듈식 구조를 가지고 있습니다. 즉, 패키지에는 여러 공유 또는 정적 라이브러리가 포함됩니다. 다음 모듈을 사용할 수 있습니다.

- **핵심 기능 (core)** - 조밀한 다차원 배열 Mat 및 다른 모든 모듈에서 사용되는 기본 기능을 포함하여 기본 데이터 구조를 정의하는 소형 모듈입니다.
- **이미지 처리 (imgproc)** - 선형 및 비선형 이미지 필터링, 기하학적 이미지 변환(크기 조정, 아핀 및 원근 변형, 일반 테이블 기반 재매핑), 색 공간 변환, 히스토그램 등을 포함하는 이미지 처리 모듈입니다.
- **비디오 분석 (비디오)** - 모션 추정, 배경 빼기 및 객체 추적 알고리즘을 포함하는 비디오 분석 모듈입니다.
- **카메라 보정 및 3D 재구성 (calib3d)** - 기본 다중 뷰 기하학 알고리즘, 단일 및 스테레오 카메라 보정, 물체 포즈 추정, 스테레오 대응 알고리즘 및 3D 재구성 요소.
- **2D 기능 프레임워크 (features2d)** - 두드러진 기능 감지기, 설명자 및 설명자 일치자.
- **객체 감지 (objdetect)** - 미리 정의된 클래스(예: 얼굴, 눈, 머그, 사람, 자동차 등)의 객체 및 인스턴스 감지.
- **고급 GUI (highgui)** - 간단한 UI 기능에 대한 사용하기 쉬운 인터페이스입니다.
- **비디오 I/O (videoio)** - 비디오 캡처 및 비디오 코덱에 대한 사용하기 쉬운 인터페이스입니다.
- ... FLANN 및 Google 테스트 래퍼, Python 바인딩 등과 같은 기타 도우미 모듈.

OpenCV-Python 설치

□ pip 명령으로 OpenCV-Python 설치하기

✓ 설치 명령

```
> pip install opencv-python
```

- `<PYTHON_PATH>\Lib\site-packages` 아래에 `cv2`, `opencv_python` - 4.5.5.64.dist-info 폴더가 생성되고, 그 아래에 `pyd` 파일이 저장됨. (`cv2.pyd`)
- 설치되는 OpenCV 버전은 <https://pypi.org/> 에서 확인 가능
- `numpy` 패키지도 함께 설치됨
- OpenCV 추가 모듈도 함께 사용하려면 `opencv-contrib-python` 패키지를 설치

- OpenCV 4.2.0 버전에서 `cv2.imread()` 함수를 사용하여 컬러 영상을 그레이스케일 형식으로 불러올 때 픽셀 값이 잘못 설정되는 문제가 있음
- `cv2.imread()` 함수가 정상 동작하는 OpenCV를 설치하려면 아래 명령 사용
> `pip install opencv-python==4.1.0.25`

OpenCV-Python 설치

- OpenCV 설치 배포 프로그램을 이용하여 설치하기 (Optional)
 - ✓ <https://opencv.org/releases/> 에서 OpenCV 설치 프로그램 다운로드 (e.g. opencv-4.5.5-vc14_vc15.exe)
 - ✓ 다운로드 받은 EXE 파일을 실행하여 압축 해제
 - "Extract to" 입력 창에 C:\ 폴더 지정
 - ➔ C:\opencv 폴더가 새로 생성되면서 그 아래에 OpenCV 관련 파일이 저장됨
 - ✓ C:\opencv\build\python\cv2\python-3.9\cv2.pyd 파일을 <PYTHON_PATH>\Lib\site-packages 폴더로 복사
 - ✓ C:\opencv\build\x64\vc15\bin 폴더를 시스템 환경 변수 PATH에 추가

OpenCV-Python 코딩 작업 환경

□ 메모장 + 명령프롬프트

□ 주피터 노트북(Jupyter Notebook)

- ✓ 웹 브라우저에서 파이썬 코드를 작성 & 실행
- ✓ 블록 단위 코딩
- ✓ 마크업 언어와 그림 등을 활용한 설명 추가가 쉬움
- ✓ matplotlib 패키지를 이용하여 영상을 웹 브라우저에 출력 가능



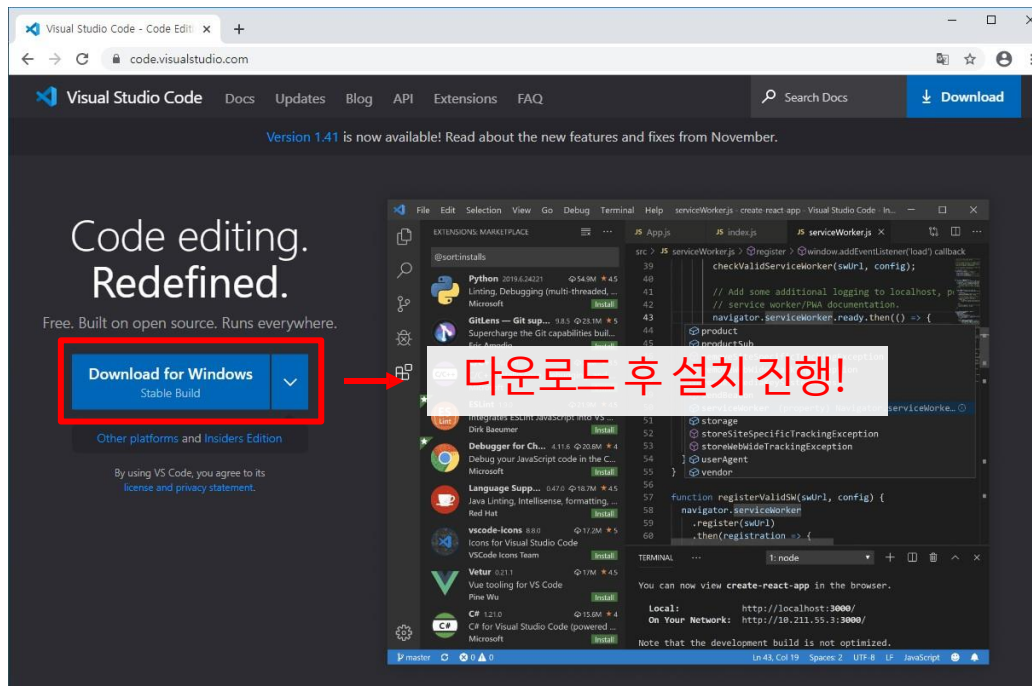
□ 파이썬 IDE

- ✓ Visual Studio Code, PyCharm, Spider 등
- ✓ OpenCV에서 제공하는 GUI 기능 사용 가능
- ✓ 편리한 디버깅

VS Code 설치

□ Visual Studio Code

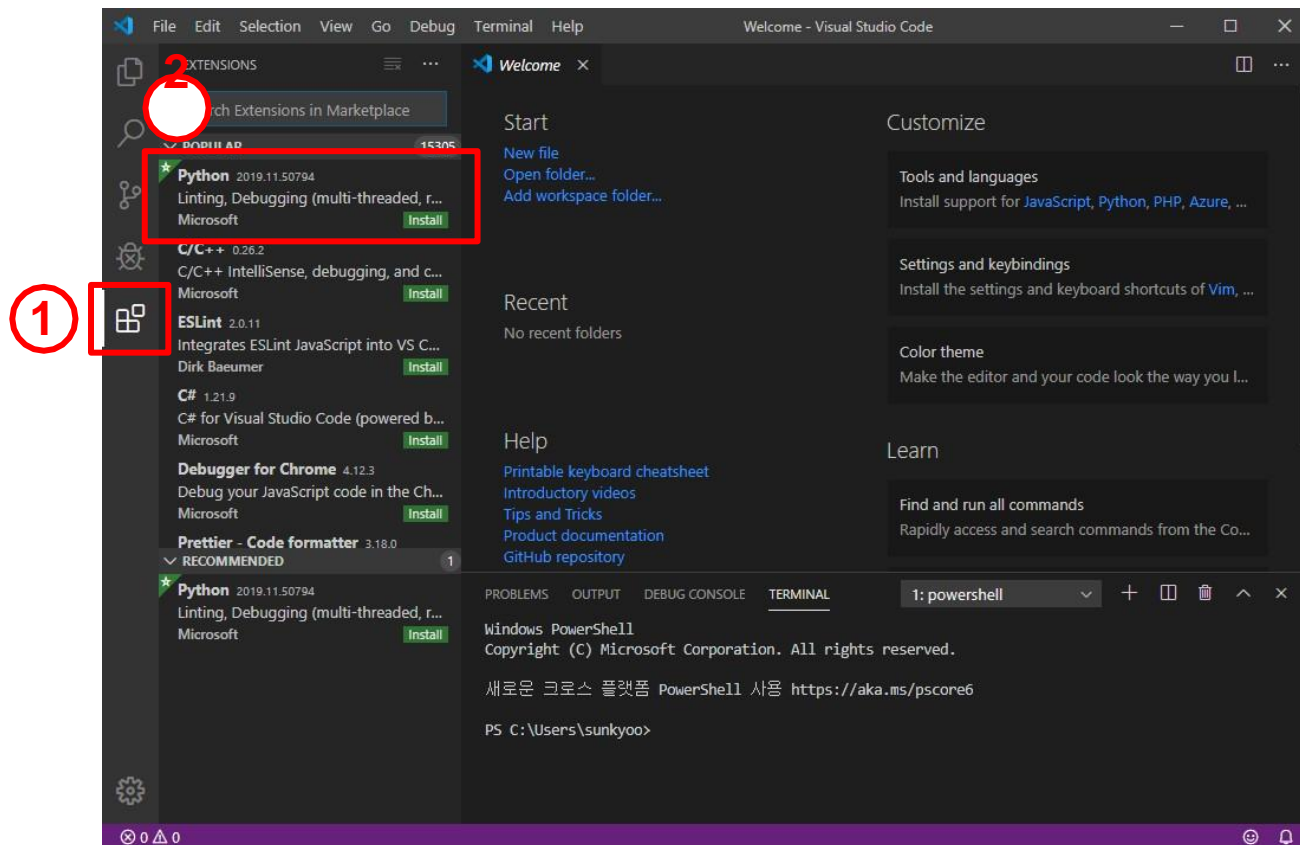
- ✓ Microsoft에서 개발한 범용 소스 코드 편집기 (Windows, MacOS, Linux)
- ✓ 설치 프로그램 다운로드: <https://code.visualstudio.com/>
- ✓ 튜토리얼: <https://code.visualstudio.com/docs>



VS Code 설치

□ Python 확장 설치

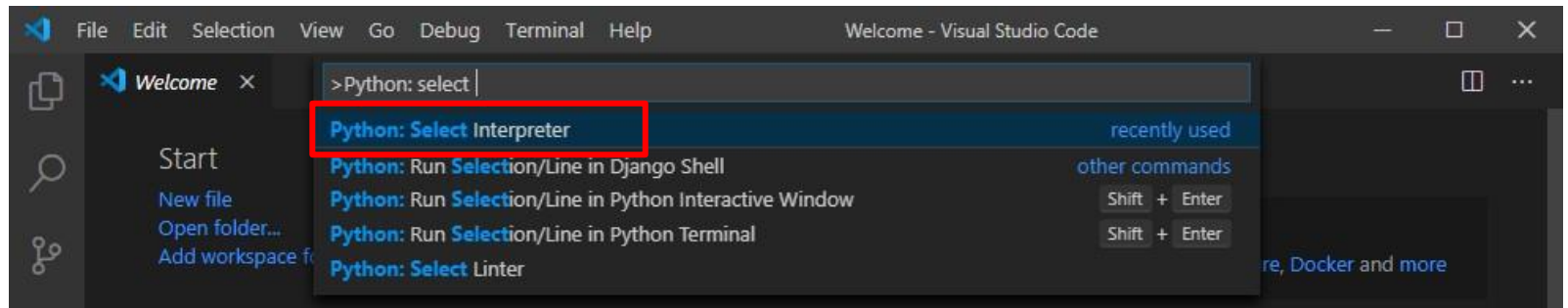
- ✓ Extensions 뷰에서 "Python (by Microsoft)" 설치



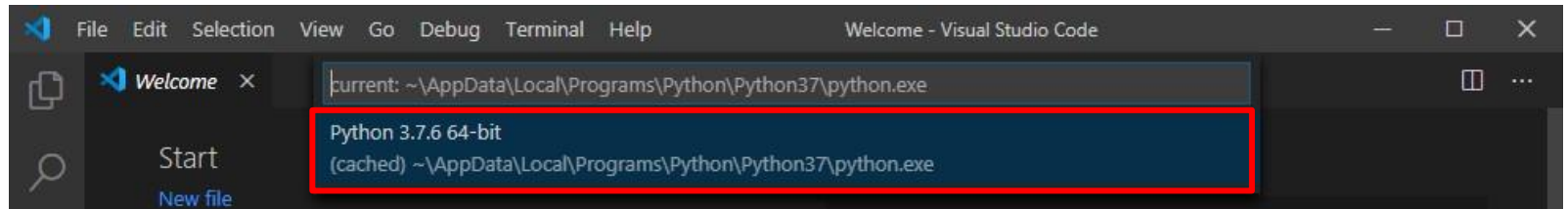
VS Code 설치

□ Python 인터프리터 선택

- ✓ [View] → [Command Palette...] (Ctrl + Shift + P) 선택 후,
"Python: Select Interpreter" 입력 & 선택



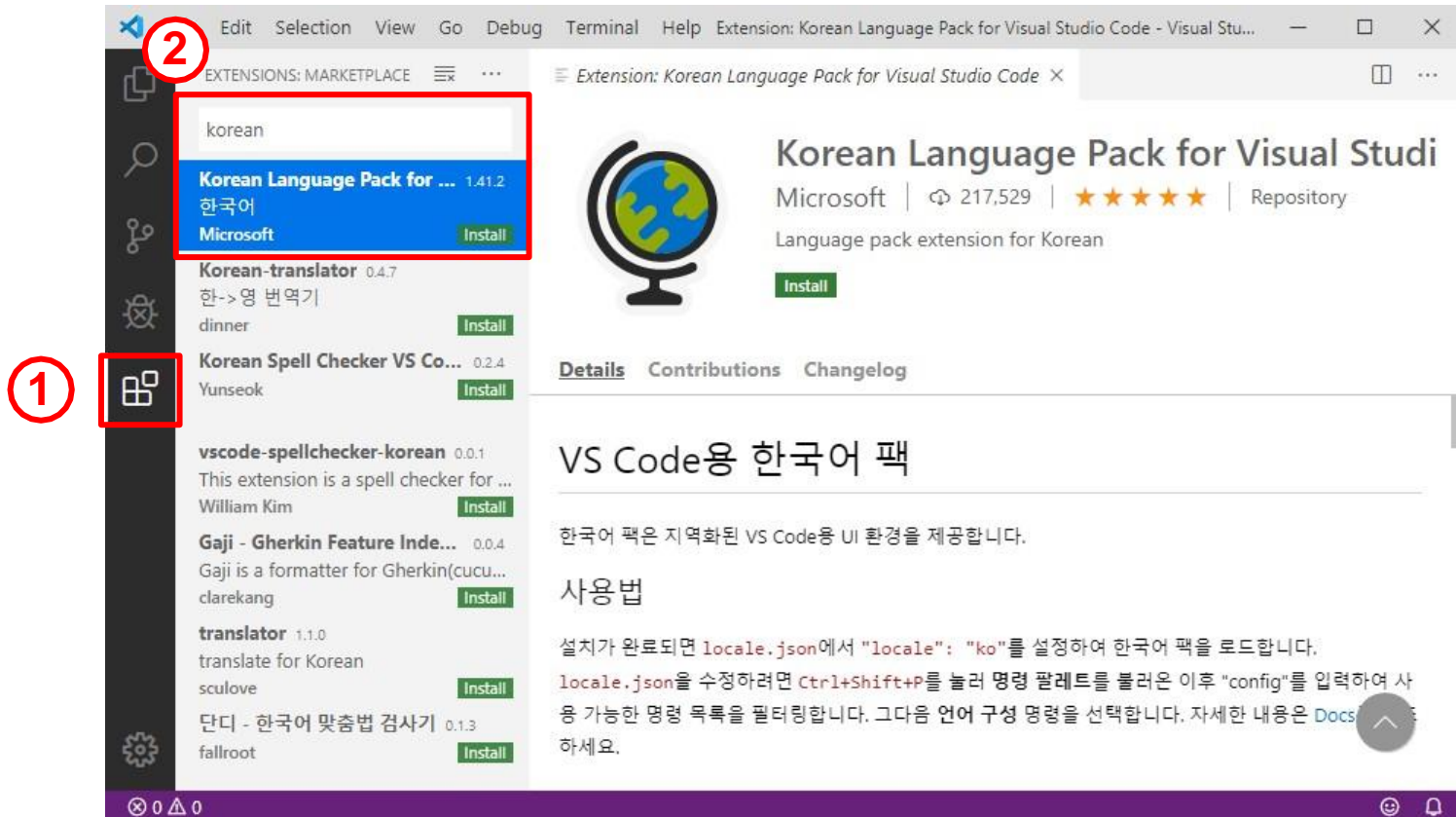
- ✓ 설치된 Python 프로그램을 선택



VS Code 설치

□ 한글 언어팩 설치 (Optional)

- ✓ Extensions 뷰에서 "Korean Language Pack" 검색하여 설치



HelloCV.py 프로그램 만들기

- VS Code에서 새 Python 프로그램 만들기
 - ✓ VS Code에서 [파일] ⑦ [폴더 열기] 메뉴 선택한 후, 임의의 폴더 선택
 - (e.g.) C:\coding\python\opencv\t-academy
 - ✓ 아래 그림처럼 [새 파일] 버튼 클릭 후, **HelloCV.py** 파일 이름 입력



- ✓ 편집창에 소스 코드 입력

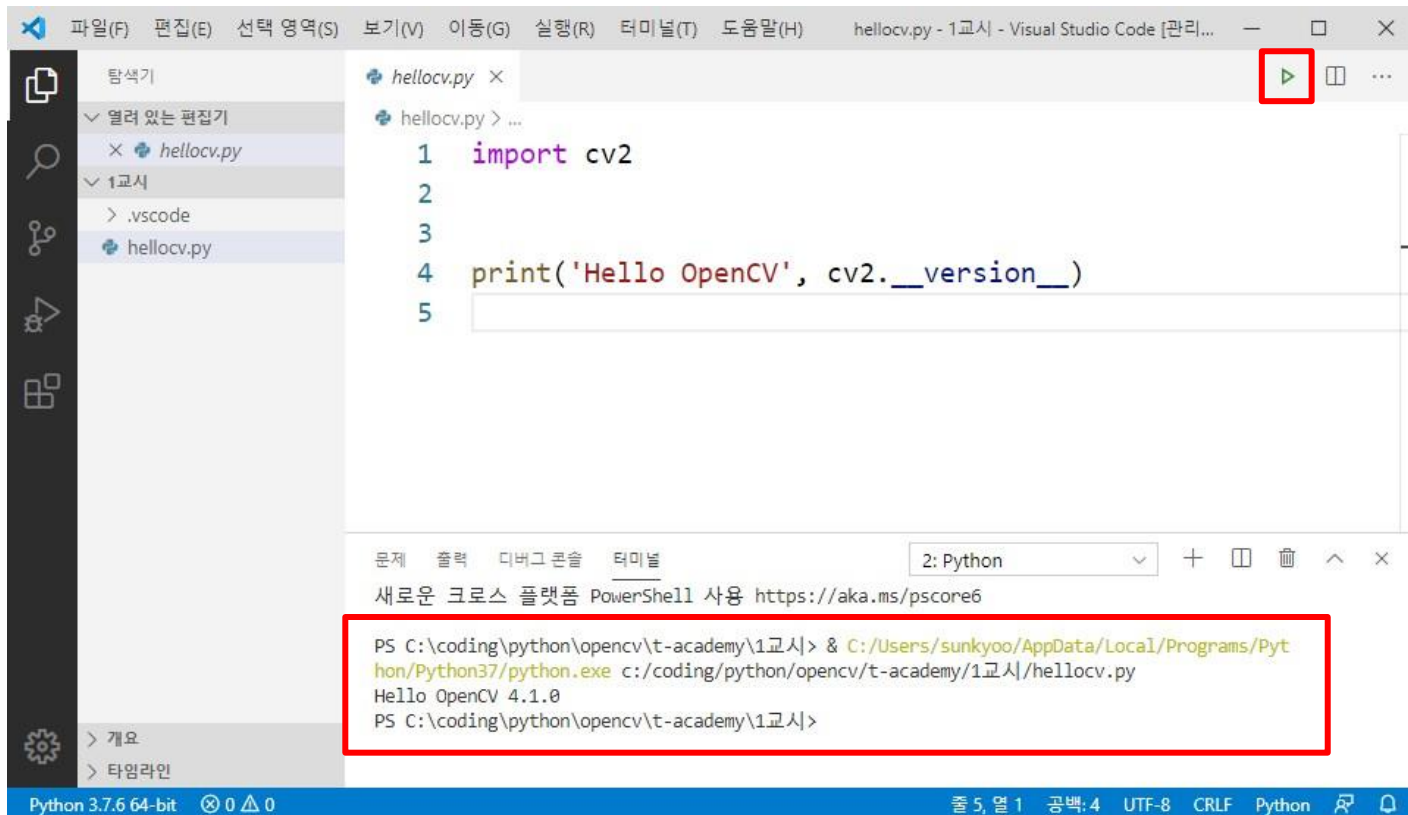
```
import cv2

print('Hello OpenCV', cv2._version_)
```

HelloCV.py 프로그램 만들기

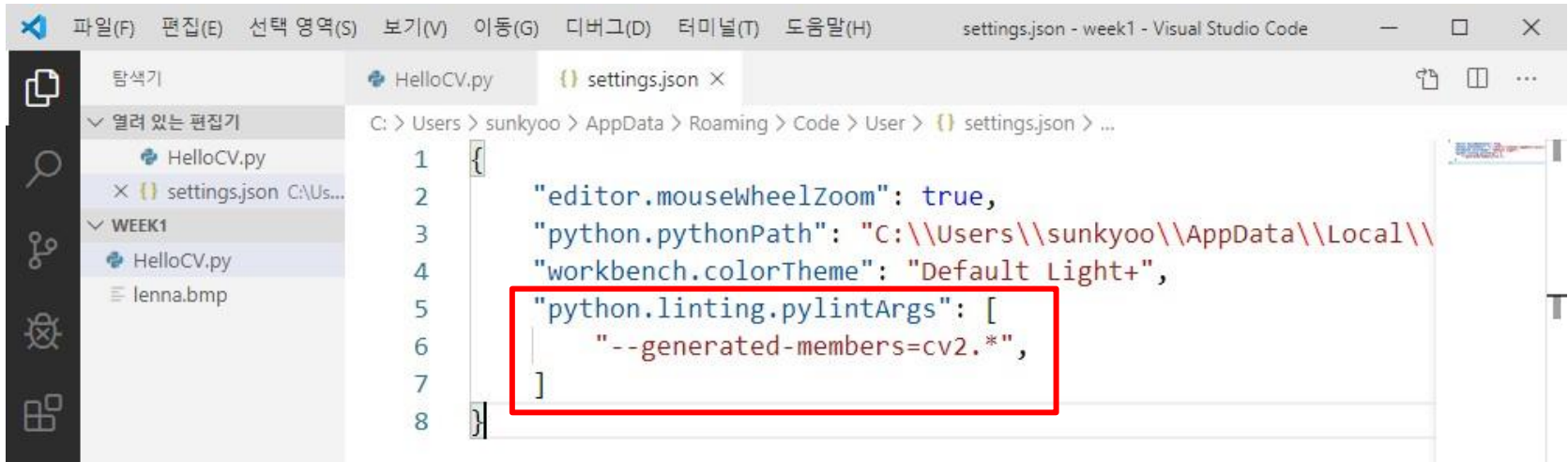
□ VS Code에서 새 Python 프로그램 실행하기

- ✓ 편집창 우측 상단 "Run Python File in Terminal" 삼각형 버튼 클릭!



HelloCV.py 프로그램 만들기

- VS Code 편집창에서 cv2 에러 없애기
 - ✓ VS Code pylint에서 cv2 모듈의 멤버를 제대로 인식하지 못하는 문제가 있음 (실행은 정상적으로 동작함)
 - ✓ [보기] → [명령 팔레트] (Ctrl + Shift + P)
 - ✓ "Preferences: Open Settings (JSON)" 선택
 - ✓ "python.linting.pylintArgs": ["--generated-members=cv2.*"] 문장 추가



The screenshot shows the Visual Studio Code interface with the settings.json file open. The file path is C:\Users\sunkyo0\AppData\Roaming\Code\User\settings.json. The JSON content is as follows:

```
1 {  
2     "editor.mouseWheelZoom": true,  
3     "python.pythonPath": "C:\\Users\\sunkyo0\\AppData\\Local\\  
4     "workbench.colorTheme": "Default Light+",  
5     "python.linting.pylintArgs": [  
6         "--generated-members=cv2.*",  
7     ]  
8 }
```

The line 5 configuration is highlighted with a red box.