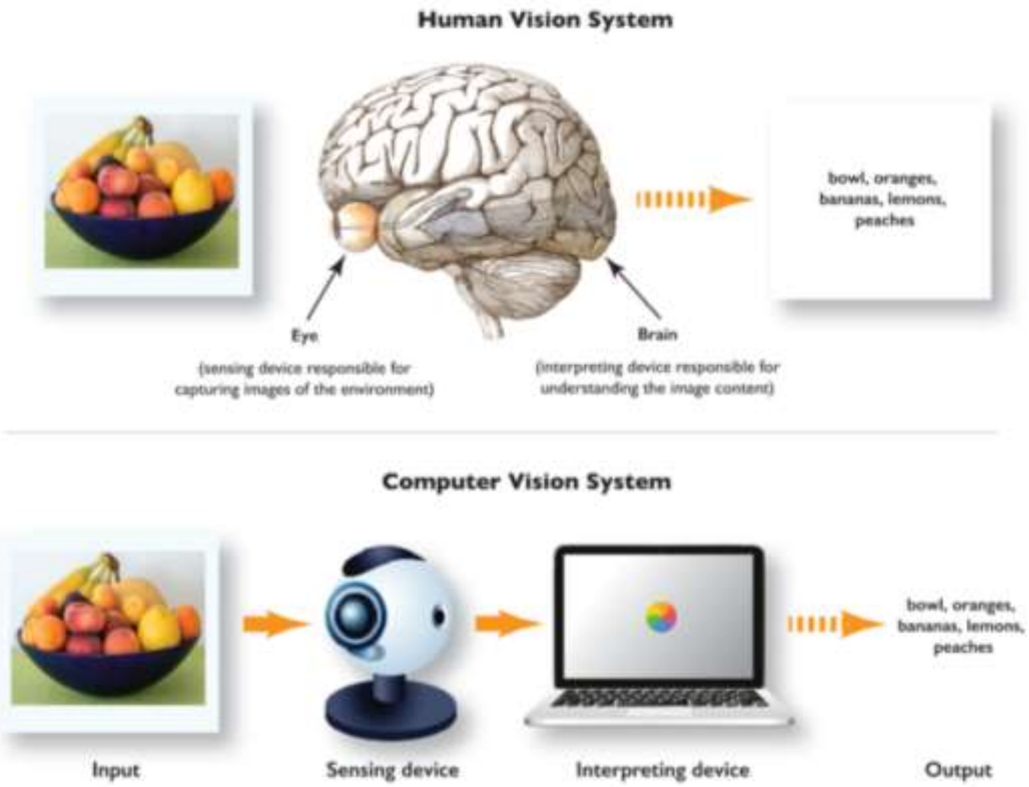
Όραση Υπολογιστών (Computer Vision)



"Just like to hear is just not to same as to listen, to take pictures is not the same as to see" ~ Fei Fei Le (Director of AI laboratory at Stanford).

Vision begins with
the eyes, but truly takes
place in the brain.

Όραση Υπολογιστών (Computer Vision)

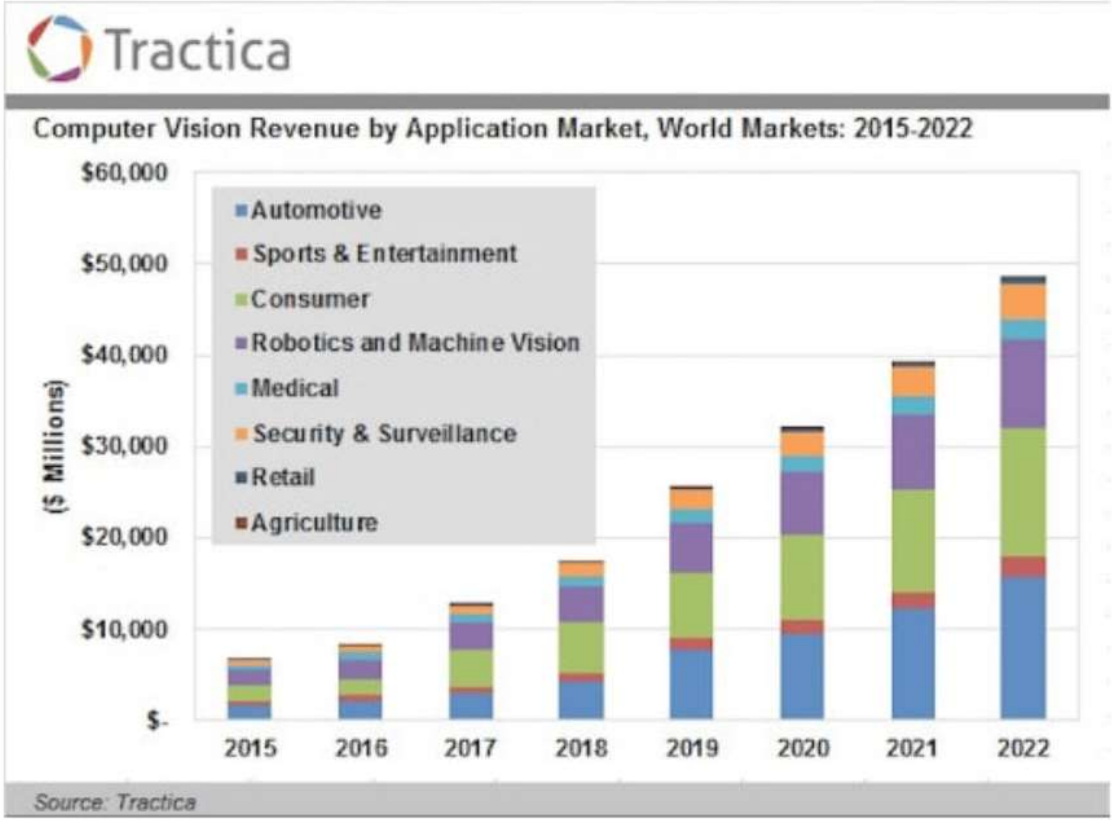
Computer Vision, also known as **CV** is a field of Computer Science.

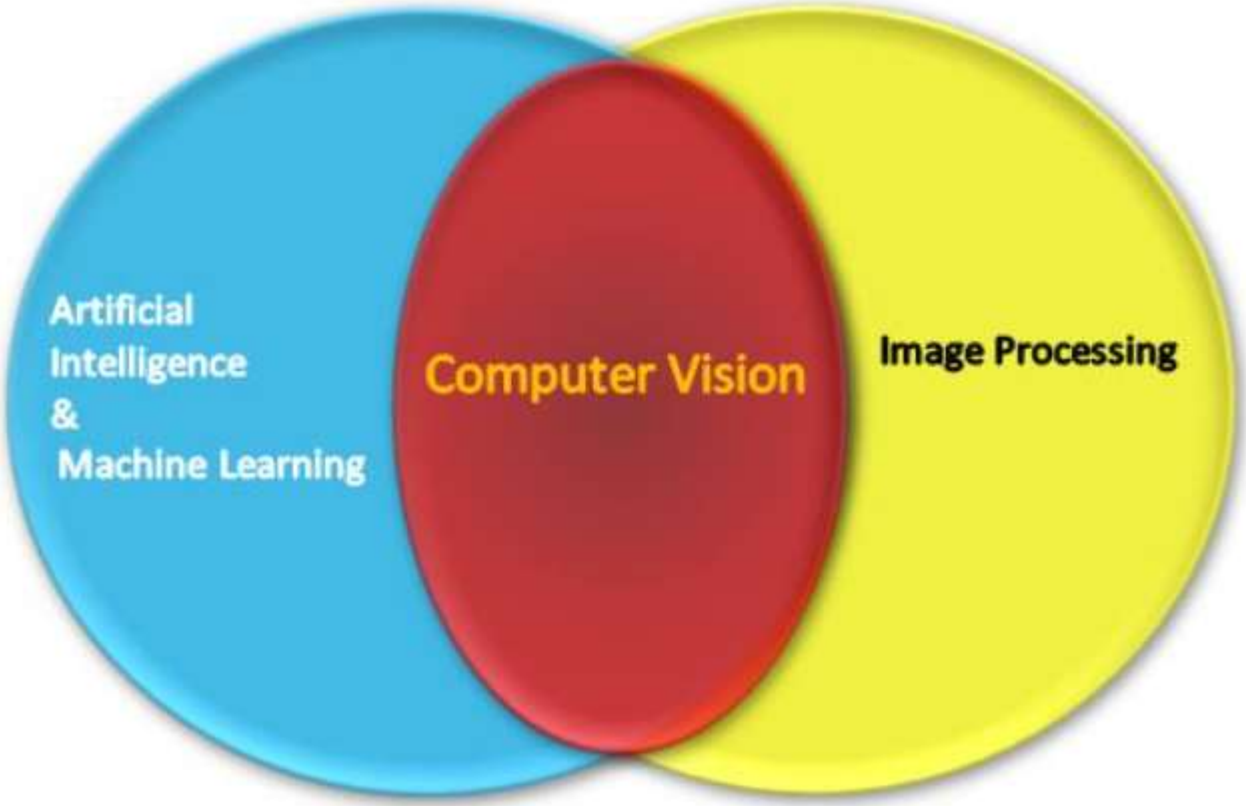
The main goal of computer vision is to give computers the ability to have a high-level understanding of digital images and videos.

In short, it is a field of computer science that seeks to develop techniques, and tools to help computers “SEE” and understand the content of digital images and videos.

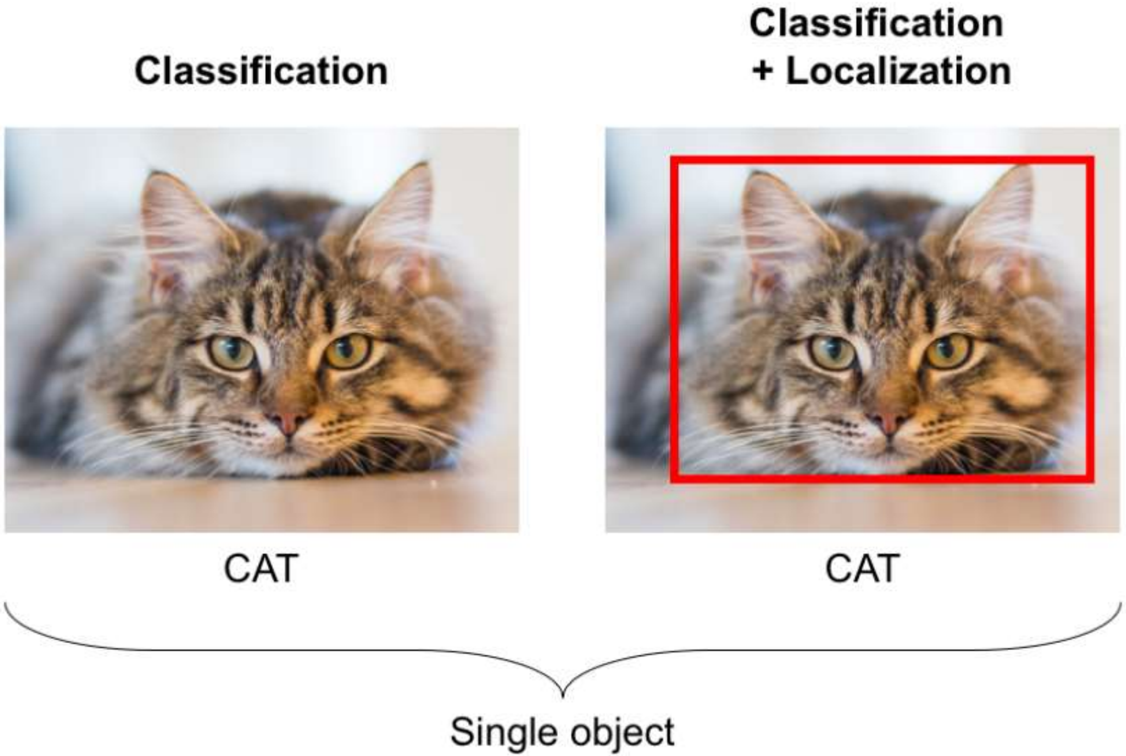
Computer Vision focuses on the implementation of the complex human vision system inside computers making them perform intelligent tasks as humans do.

Όραση Υπολογιστών Εφαρμογές


Tractica
Computer Vision Revenue by Application Market, World Markets: 2015-2022
$60,000
$50,000
$40,000
$30,000
$20,000
$10,000
$-
($ Millions)
Automotive
Sports & Entertainment
Consumer
Robotics and Machine Vision
Medical
Security & Surveillance
Retail
Agriculture
2015
2016
2017
2018
2019
2020
2021
2022
Source: Tractica

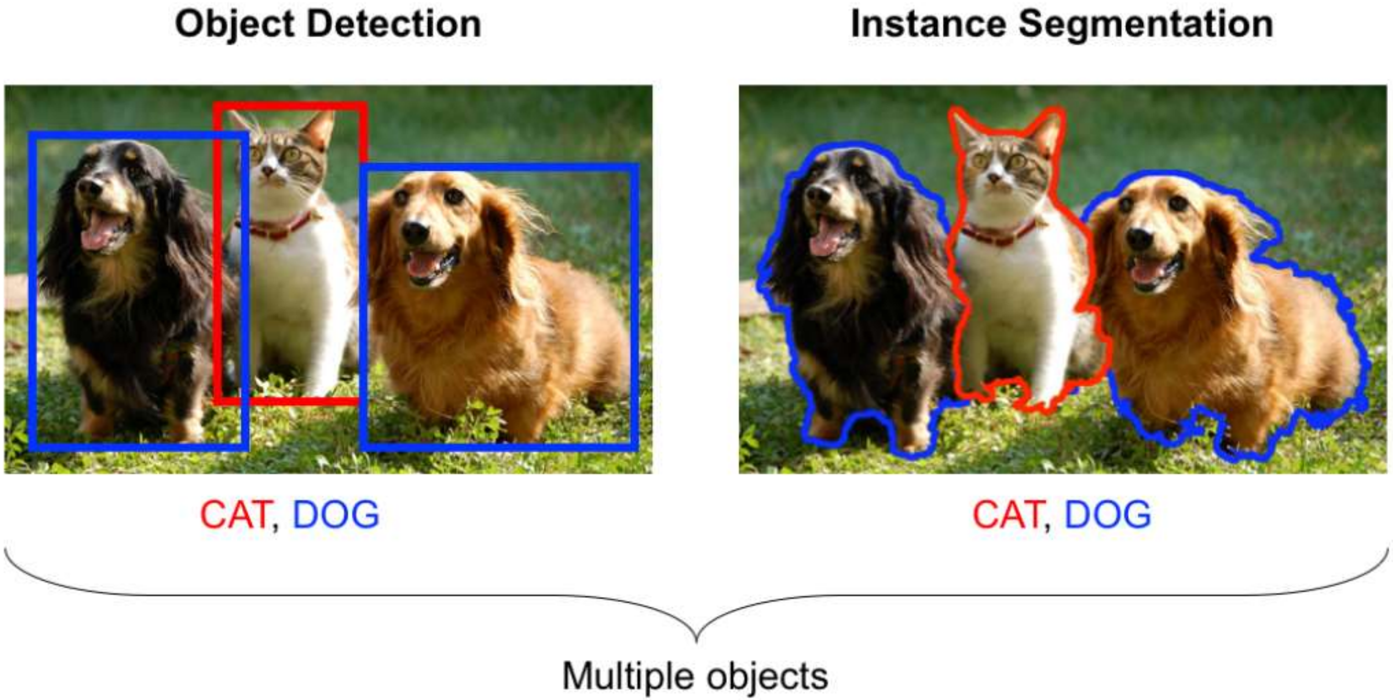

Artificial
Intelligence
&
Machine Learning
Computer Vision
Image Processing

Όραση Υπολογιστών

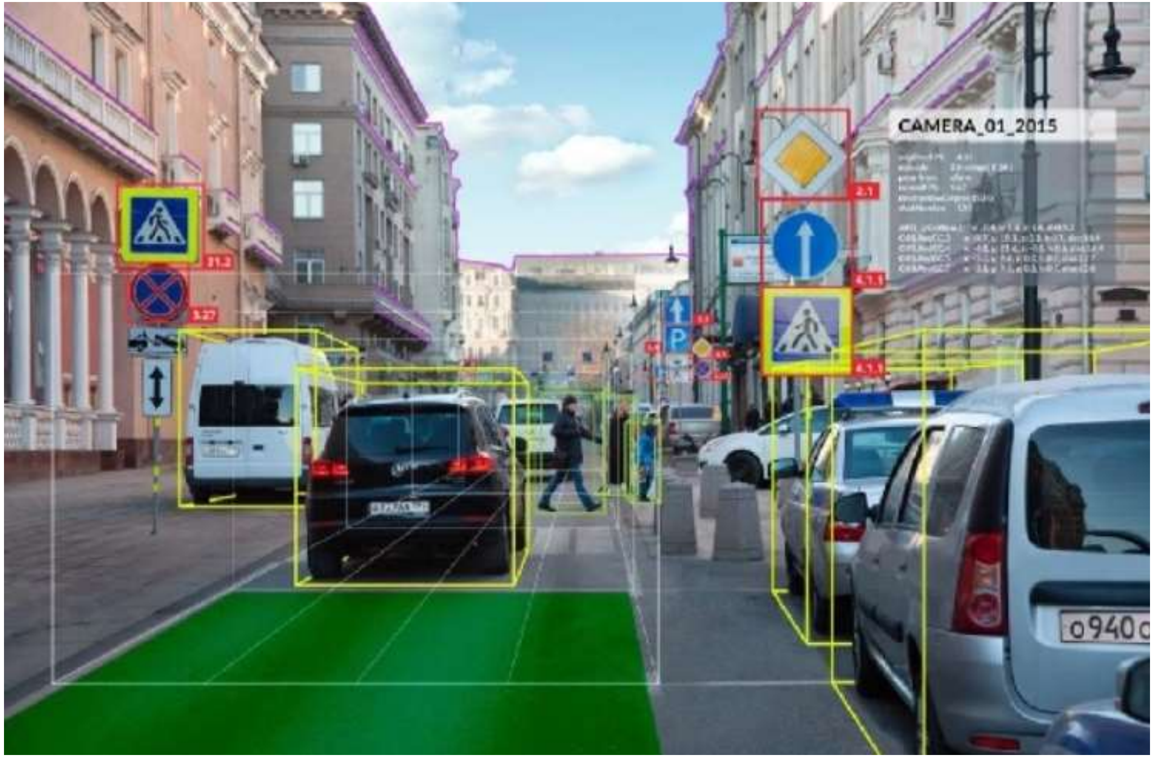


Democritus University of Thrace
IEEE Student Branch

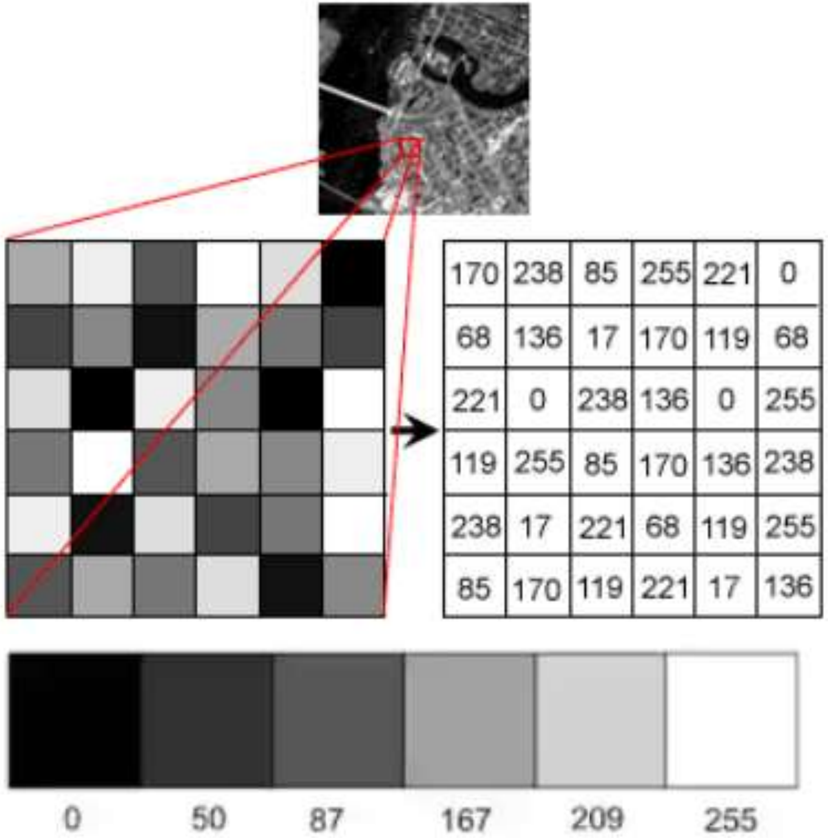
Όραση Υπολογιστών



Object detection

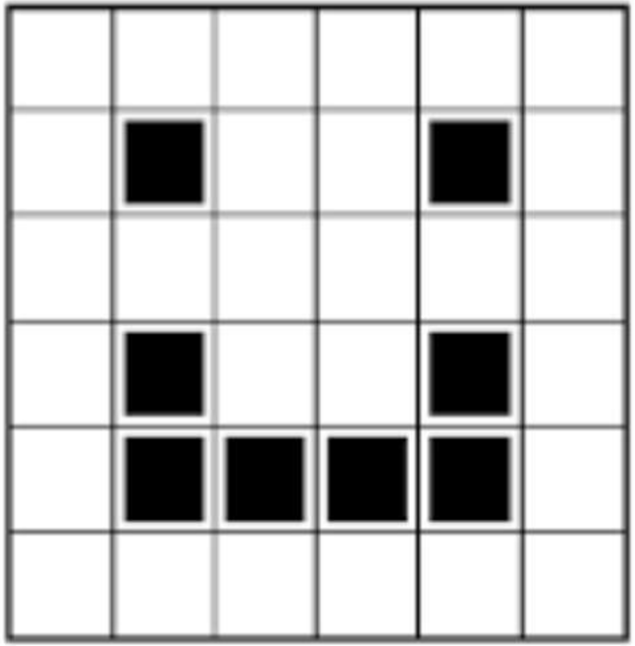


Images



170 238 85 255 221 0
68 136 17 170 119 68
221 0 238 136 0 255
119 255 85 170 136 238
238 17 221 68 119 255
85 170 119 221 17 136
0
50
87
167
209
255

Binary image (Black and White)



1	1	1	1	1	1
1	0	1	1	0	1
1	1	1	1	1	1
1	0	1	1	0	1
1	0	0	0	0	1
1	1	1	1	1	1

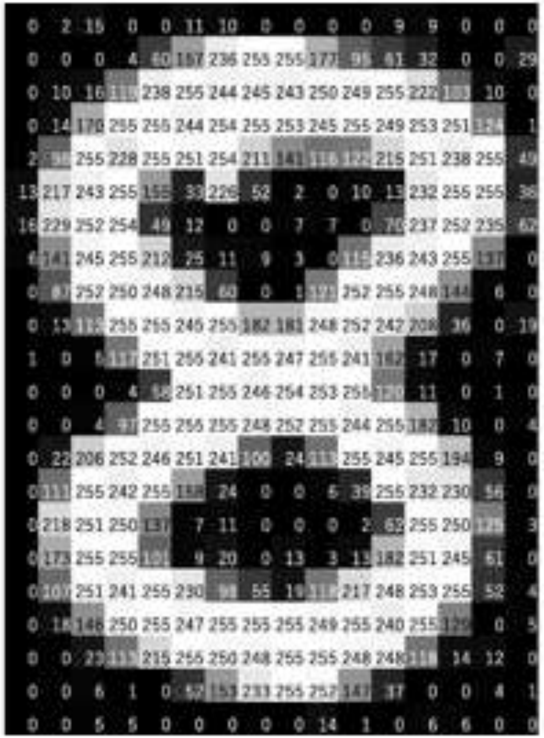
Grayscale image

25	43	11	04	70	87	12	31	43	10	05	77	12	06	45	09	29	30	02
56	22	75	03	22	96	45	12	23	03	77	67	81	45	22	04	90	22	21
32	45	41	91	87	62	35	02	00	11	62	25	43	11	04	70	87	12	61
31	43	10	05	77	12	06	45	09	29	30	56	22	75	03	22	96	45	05
12	23	03	77	67	81	45	22	04	90	22	32	45	41	91	87	62	35	44
02	00	11	62	25	43	11	04	70	87	12	31	43	10	05	77	12	06	10
45	09	29	30	56	22	75	03	22	96	45	12	23	03	77	67	81	45	55
22	04	90	22	32	45	41	91	87	62	35	02	00	11	62	25	43	11	80
04	70	87	12	31	43	10	05	77	12	06	45	09	29	30	56	22	75	08
03	22	96	45	12	23	03	77	67	81	45	22	04	90	22	32	45	41	99
91	87	62	35	02	00	11	62	22	01	00	72	65	23	01	00	22	04	30
90	22	32	45	41	91	87	62	35	02	00	11	62	25	43	11	04	70	42
87	12	31	43	10	05	77	12	06	45	09	29	30	56	22	75	03	22	91
96	45	12	23	03	77	67	81	45	22	04	90	22	32	45	41	91	87	40
62	35	02	00	11	62	22	01	00	72	65	23	01	00	56	22	75	03	67
22	96	45	12	23	03	77	67	81	45	22	04	90	22	32	45	41	91	22

What we see **What computers see**

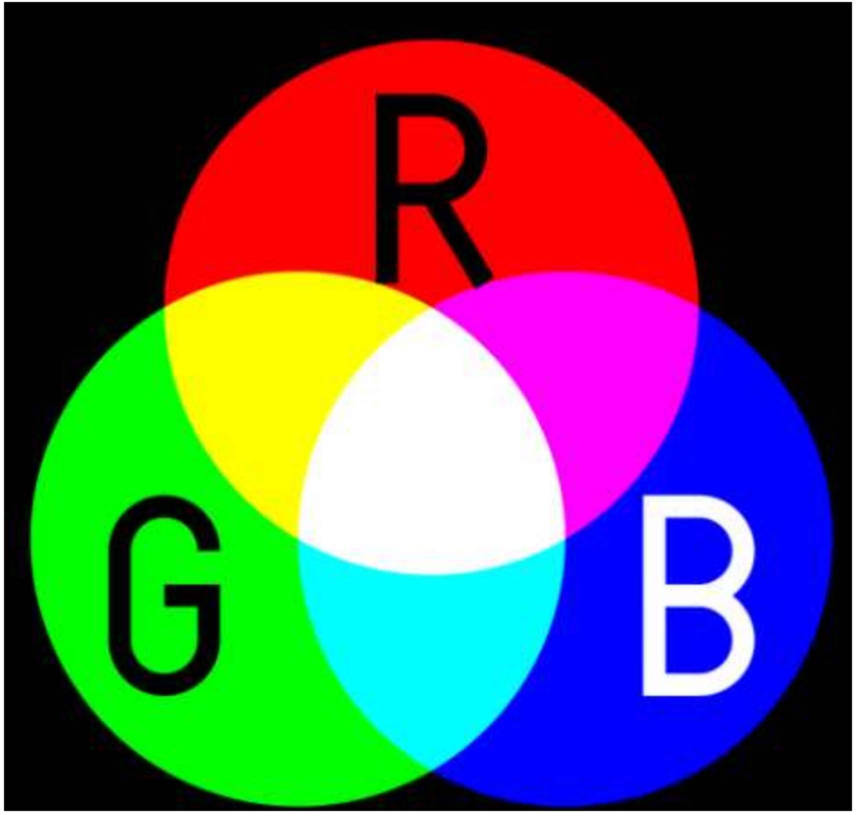
Democritus University of Thrace
IEEE Student Branch

What Computer Sees



0	2	15	0	0	11	10	0	0	0	0	9	9	0	0	0
0	0	0	4	60	157	236	255	255	177	95	61	32	0	0	29
0	10	16	119	238	255	244	245	243	250	249	255	222	103	10	0
0	14	170	255	255	244	254	255	253	245	255	249	253	251	124	1
2	98	255	228	255	251	254	211	141	116	122	215	251	238	255	49
13	217	243	255	155	33	226	52	2	0	10	13	232	255	255	36
16	229	252	254	49	12	0	0	7	7	0	70	237	252	235	62
6	141	245	255	212	25	11	9	3	0	115	236	243	255	137	0
0	87	252	250	248	215	60	0	1	121	252	255	248	144	6	0
0	13	113	255	255	245	255	182	181	248	252	242	208	36	0	19
1	0	5	117	251	255	241	255	247	255	241	162	17	0	7	0
0	0	0	4	58	251	255	246	254	253	255	120	11	0	1	0
0	0	4	97	255	255	255	248	252	255	244	255	182	10	0	4
0	22	206	252	246	251	241	100	24	113	255	245	255	194	9	0
0	111	255	242	255	158	24	0	0	6	39	255	232	230	56	0
0	218	251	250	137	7	11	0	0	0	2	62	255	250	125	3
0	173	255	255	101	9	20	0	13	3	13	182	251	245	61	0
0	107	251	241	255	230	98	55	19	118	217	248	253	255	52	4
0	18	146	250	255	247	255	255	255	249	255	240	255	129	0	5
0	0	23	113	215	255	250	248	255	255	248	248	118	14	12	0
0	0	6	1	0	52	153	233	255	252	147	37	0	0	4	1
0	0	5	5	0	0	0	0	0	14	1	0	6	6	0	0

RGB



How to create colors with RGB?

Combine parts of the three primary colors **red**, **green** and **blue**.

Each of the primary colors can have a value in the range from 0 to 255.

R:	255	0	0	0	255
G:	0	255	0	0	255
B:	0	0	255	0	255

Images

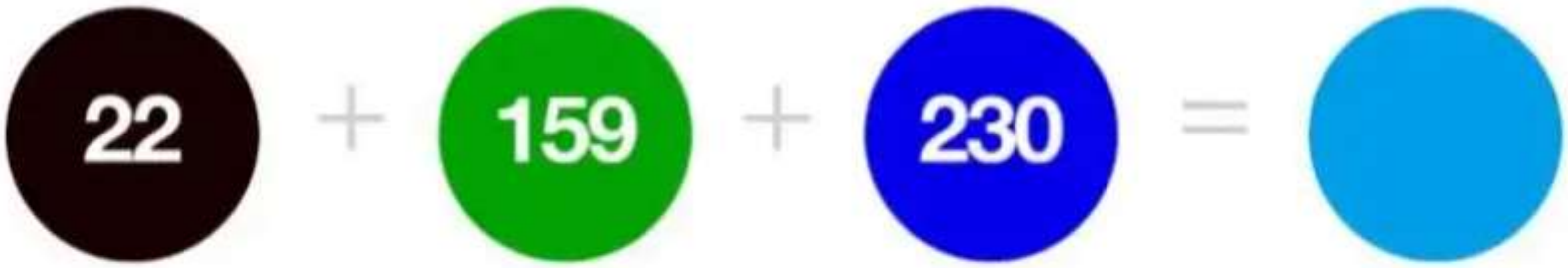


Fig 1.4 R has value 22, G has value 159, B has value 230 combined in an additive manner to result in Navy Blue color component

Image with color

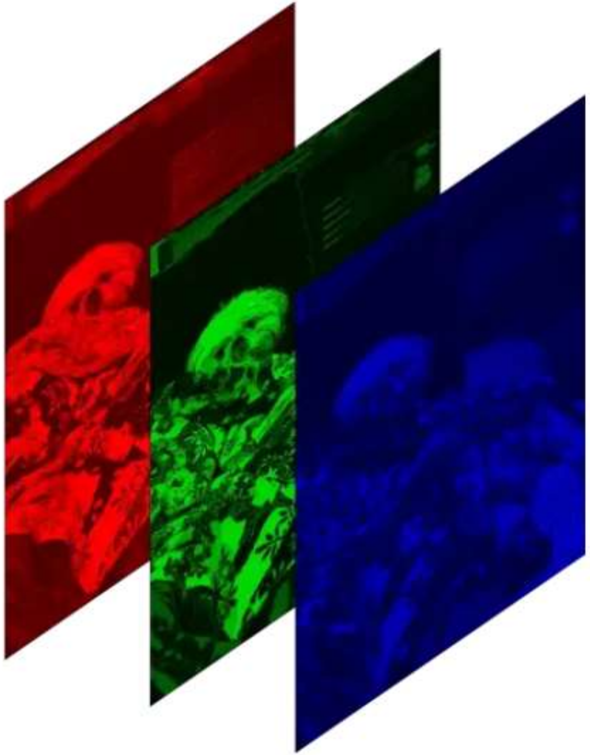
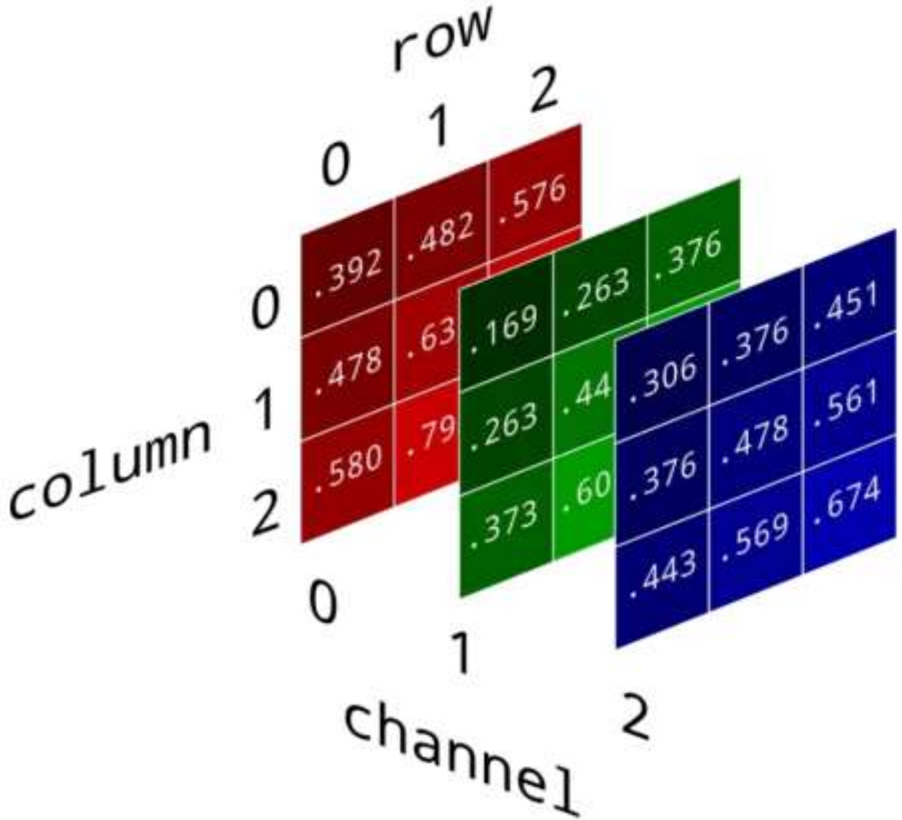


Image with color



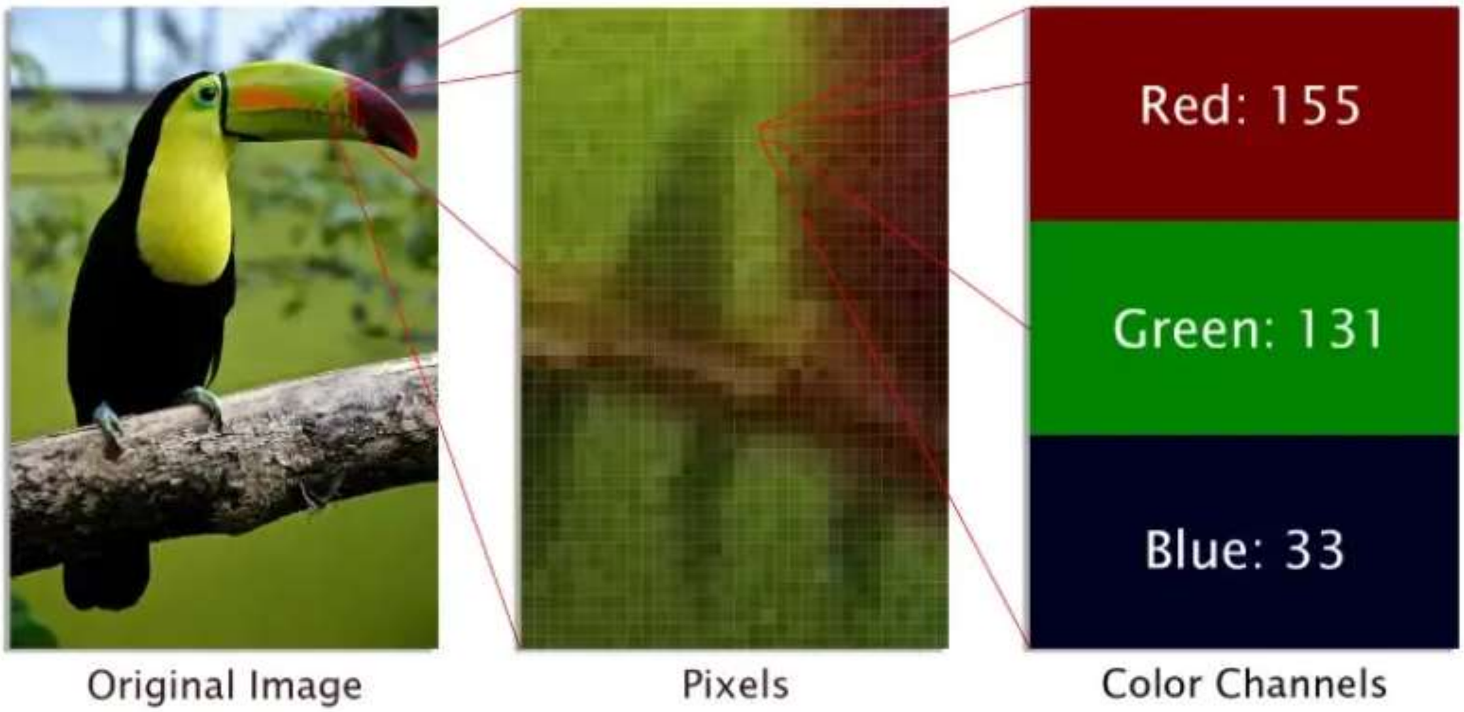
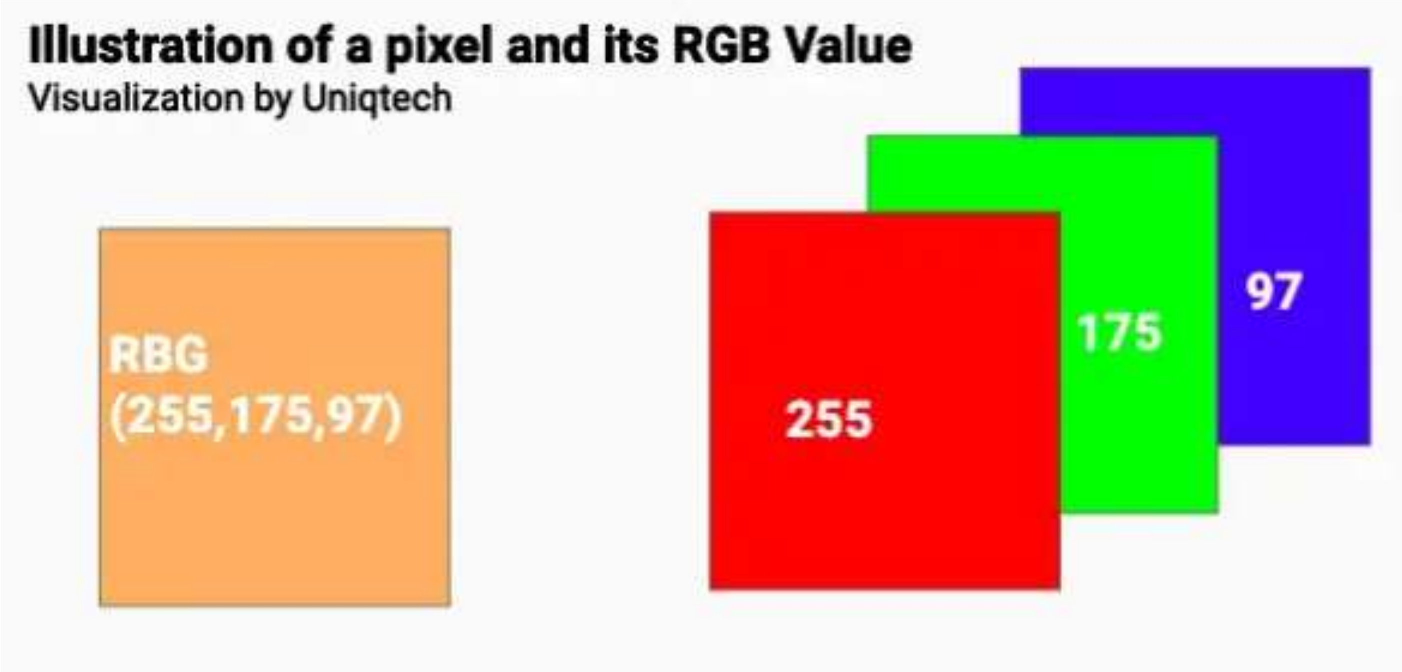
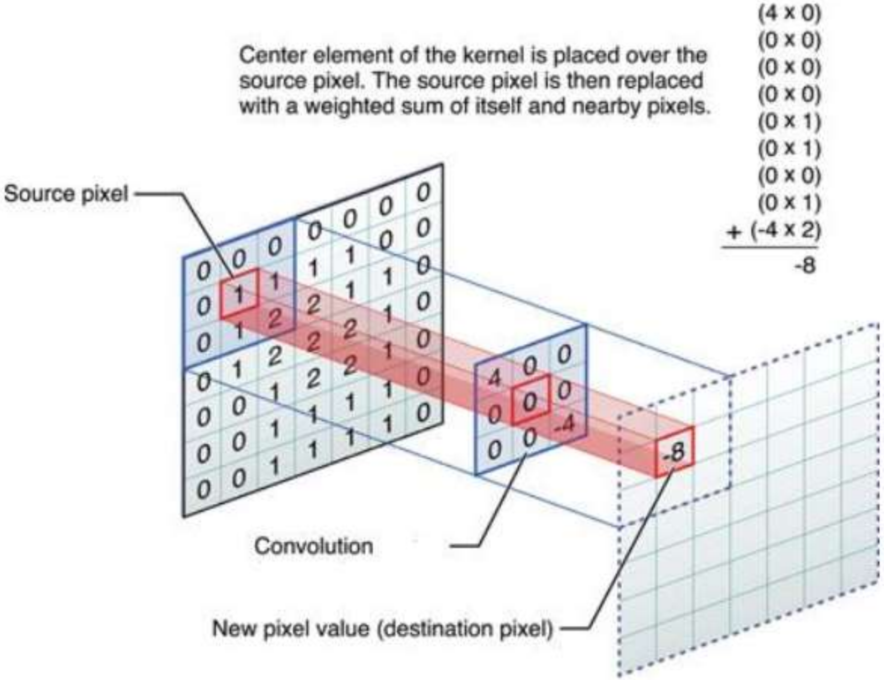

Red: 155
Green: 131
Blue: 33
Original Image
Pixels
Color Channels

Image with color

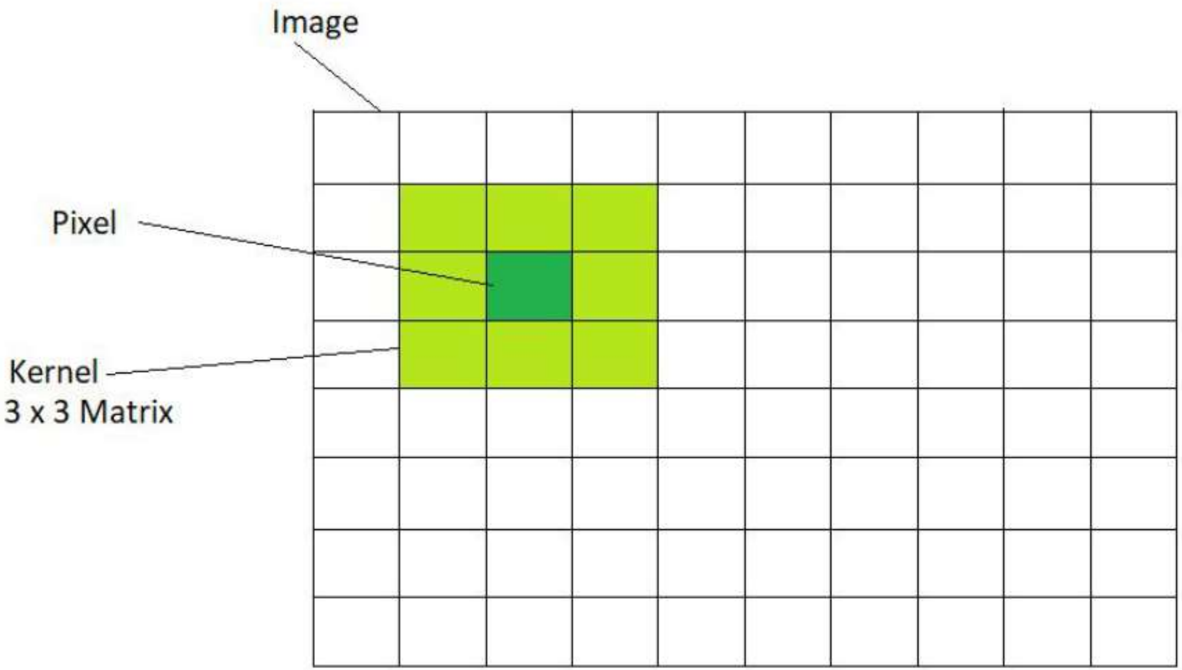


Democritus University of Thrace
IEEE Student Branch

Image filtering



Convolution Operation on a 7×7 matrix with a 3×3 kernel



2D Convolution

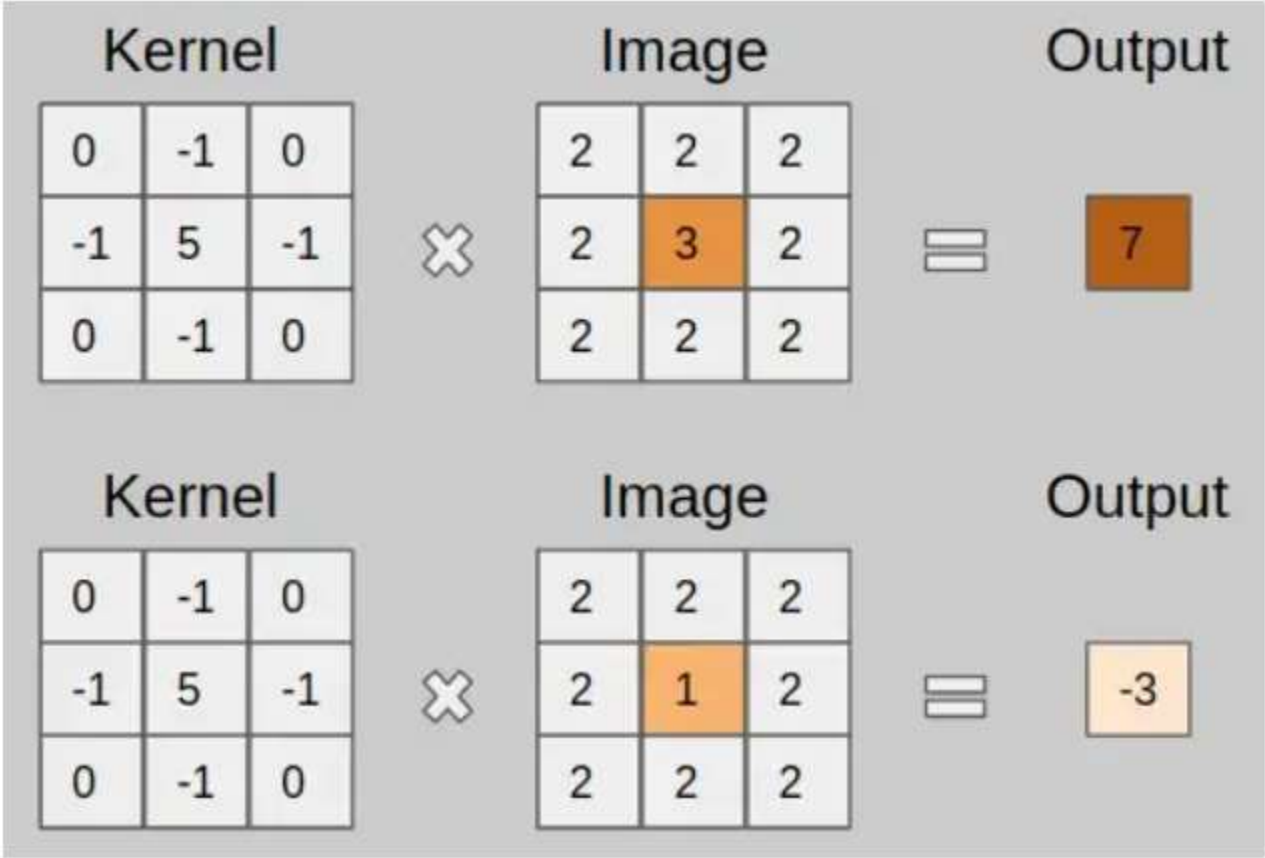

Kernel
0 -1 0
-1 5 -1
0 -1 0
Image
2 2 2
2 3 2
2 2 2
Output
7
Kernel
0 -1 0
-1 5 -1
0 -1 0
Image
2 2 2
2 1 2
2 2 2
Output
-3

Image Filtering

- https://www.youtube.com/watch?v=6v8dNtknOSM
- https://i.stack.imgur.com/AV512.gif

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Famous filters in Opencv

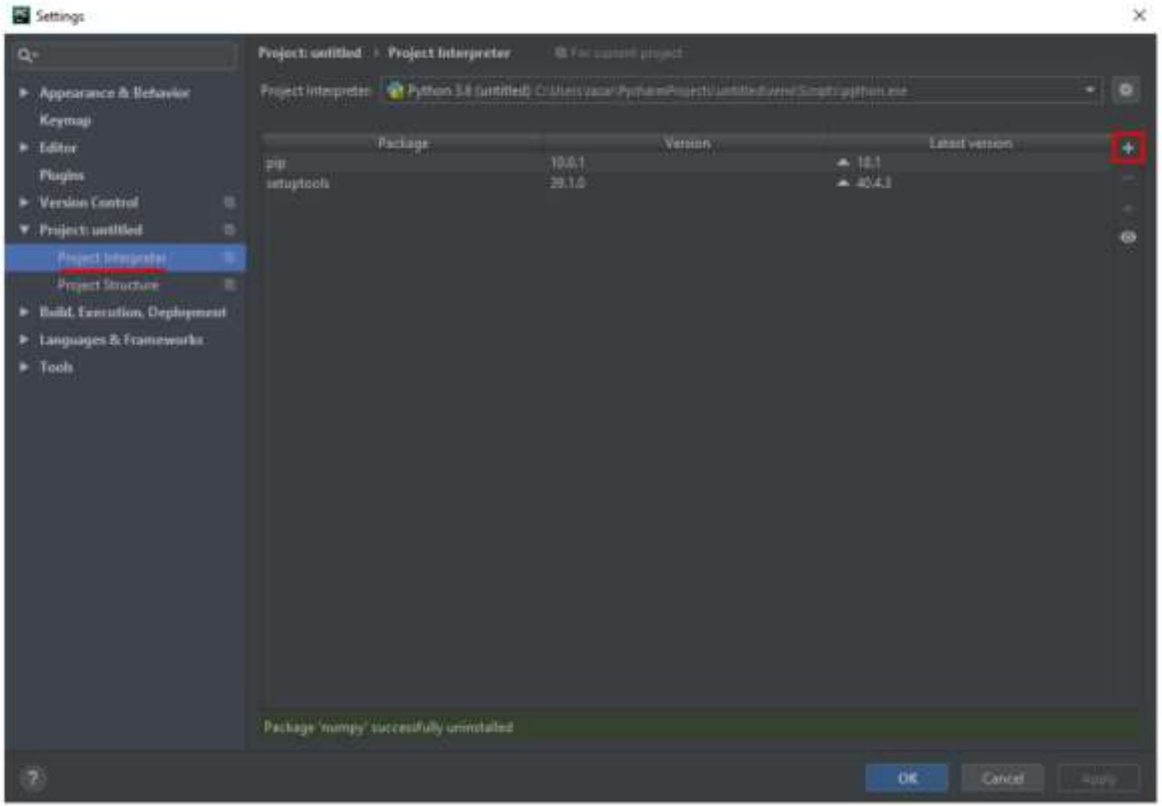
https://docs.opencv.org/3.4/d4/d86/group__imgproc__filter.html

Βιβλιοθήκη Opencv

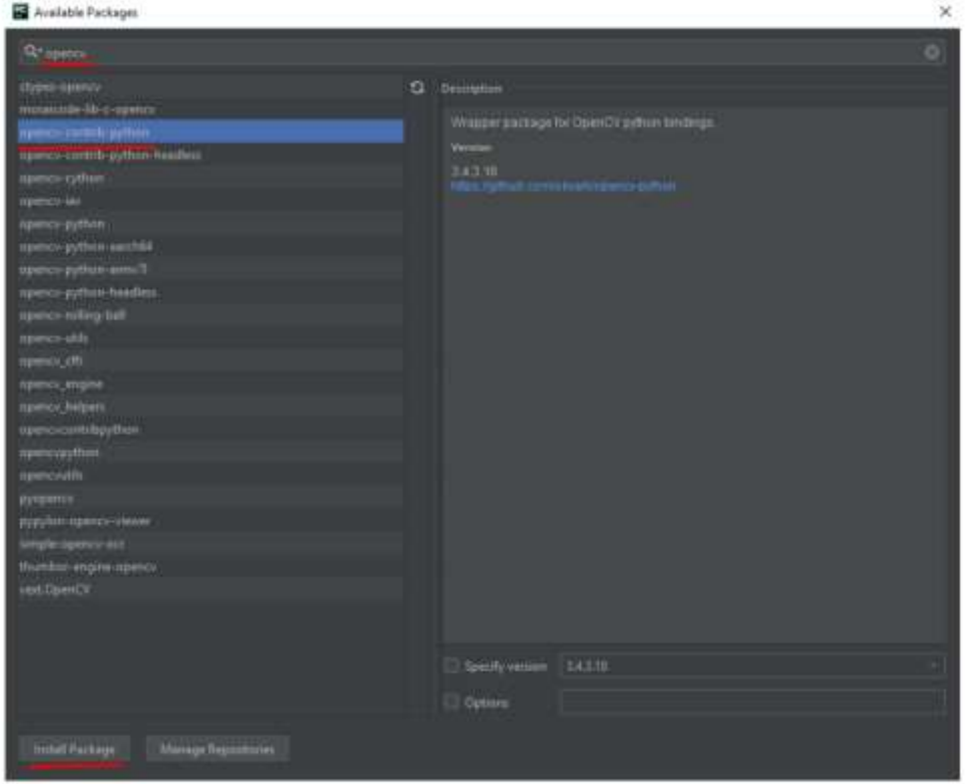
Εγκατάσταση βιβλιοθήκης στο PyCharm

1. Πηγάινουμε στο menu: File -> Settings
2. Στο δεξιο tree επιλέγουμε “Project: <name>” -> Project Interpreter
3. Επιλέγουμε προσθήκη (+)
4. Αναζητούμε “opencv”
5. Επιλέγουμε “opencv-contrib-python”
6. Install Package…

Βιβλιοθήκη Opencv



Βιβλιοθήκη Opencv

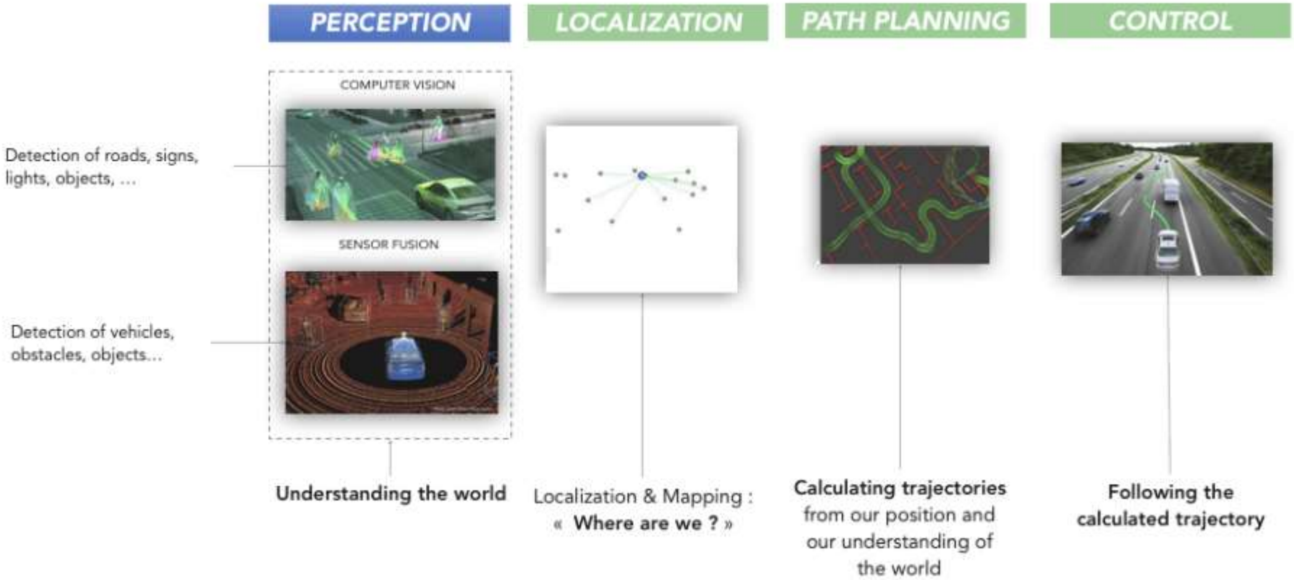


Βιβλιοθήκη Opencv

Με τον παρακάτω κώδικα μπορούμε να δούμε αν η βιβλιοθήκη εγκαταστάθηκε σωστά. Επίσης μπορούμε να δούμε ποιο version της βιβλιοθήκης χρησιμοποιούμε.

```
import cv2
print(cv2.__version__)
```

Computer Vision applications in Autonomous Vehicles



Democritus University of Thrace
IEEE Student Branch

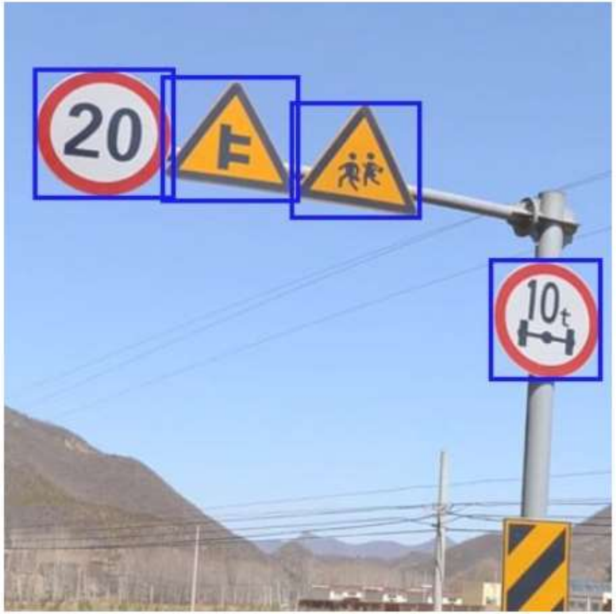
Lane Tracking

Lane Tracking is a vital component of an autonomous vehicle to decide which lane it should stay in, and not move randomly on the road.

Lane detection

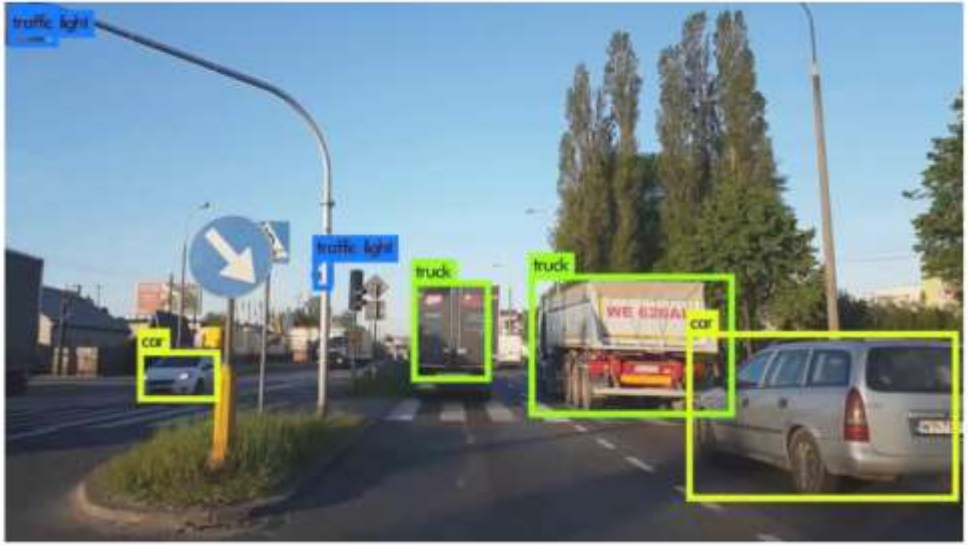
Traffic Signs Detection

Detection of traffic signs is an important task that is achieved using Computer Vision and Deep Learning. Imagine an autonomous vehicle not stopping at a red light, or overspeeding at a school zone. Hence it is vital to detect these signs and act accordingly.



Vehicle Detection

Vehicle detection is also another important part of an autonomous vehicle that can be achieved using Computer Vision or Sonar. There are many object detection algorithms that use deep learning or machine learning to help in detecting that specific object. Some of the most famous algorithms include YOLO, RCN, and SSD.



Pedestrian Detection

Imagine a self-driving car or bike hits a pedestrian. This would raise a big question mark on the road safety of autonomous vehicles. In order to protect pedestrians, and to make wise moves, autonomous vehicles need to detect humans first. This is where computer vision comes into play and detects pedestrians and helps autonomous vehicles make wise decisions, thus increasing road safety.