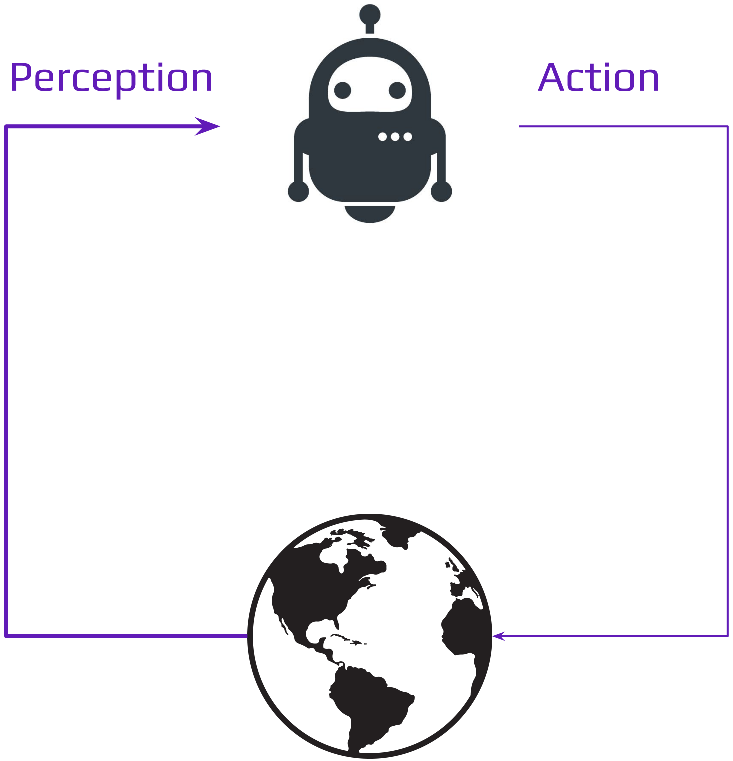
ENGR 3421:Robotics I

Robotic Vision

11/14/2023

A Robot Needs to Make Decisions



Robotic Vision Introduction

- Robotic Vision
- Image Processing
- Digital Image Creation
- Digital Image Representations
- Image Transformations
- ArUco Marker Detection

Robotic Vision

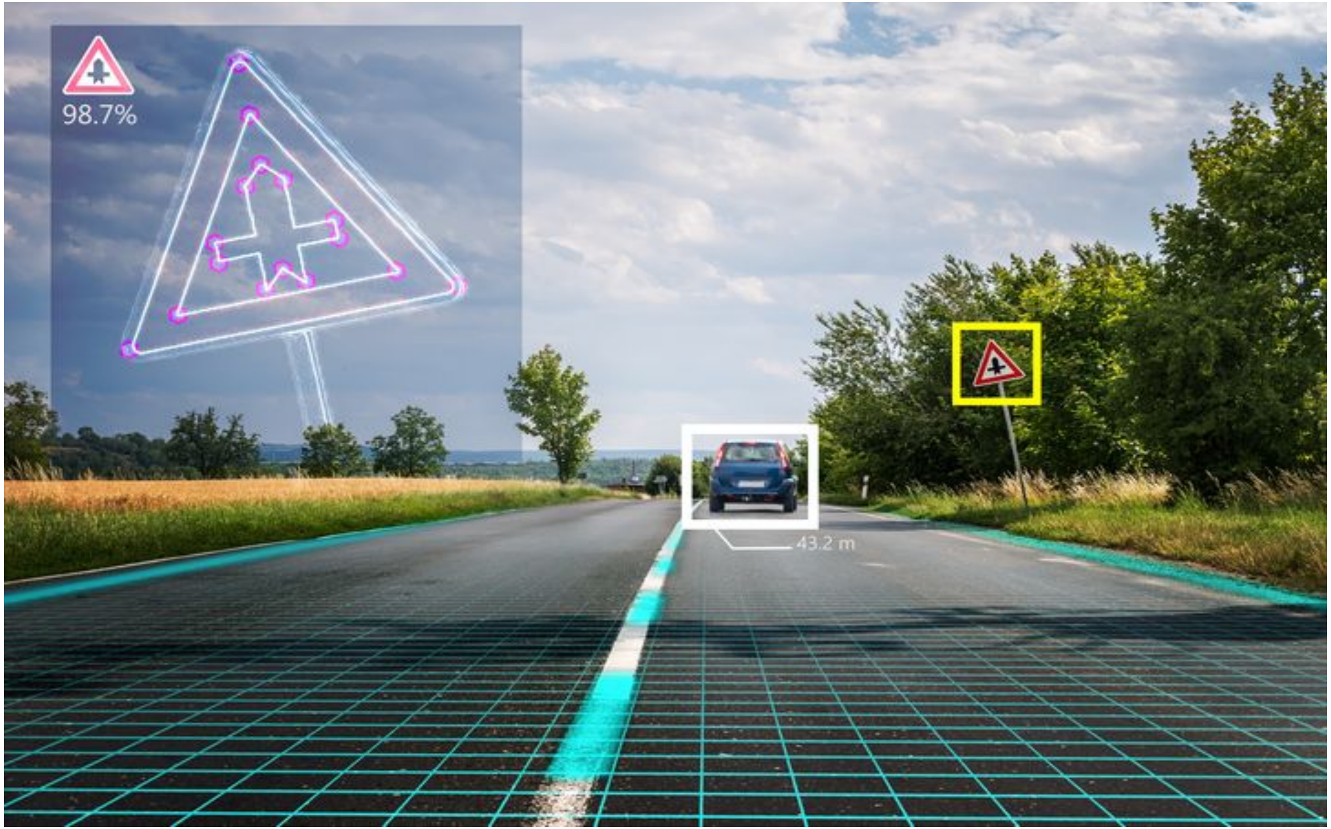
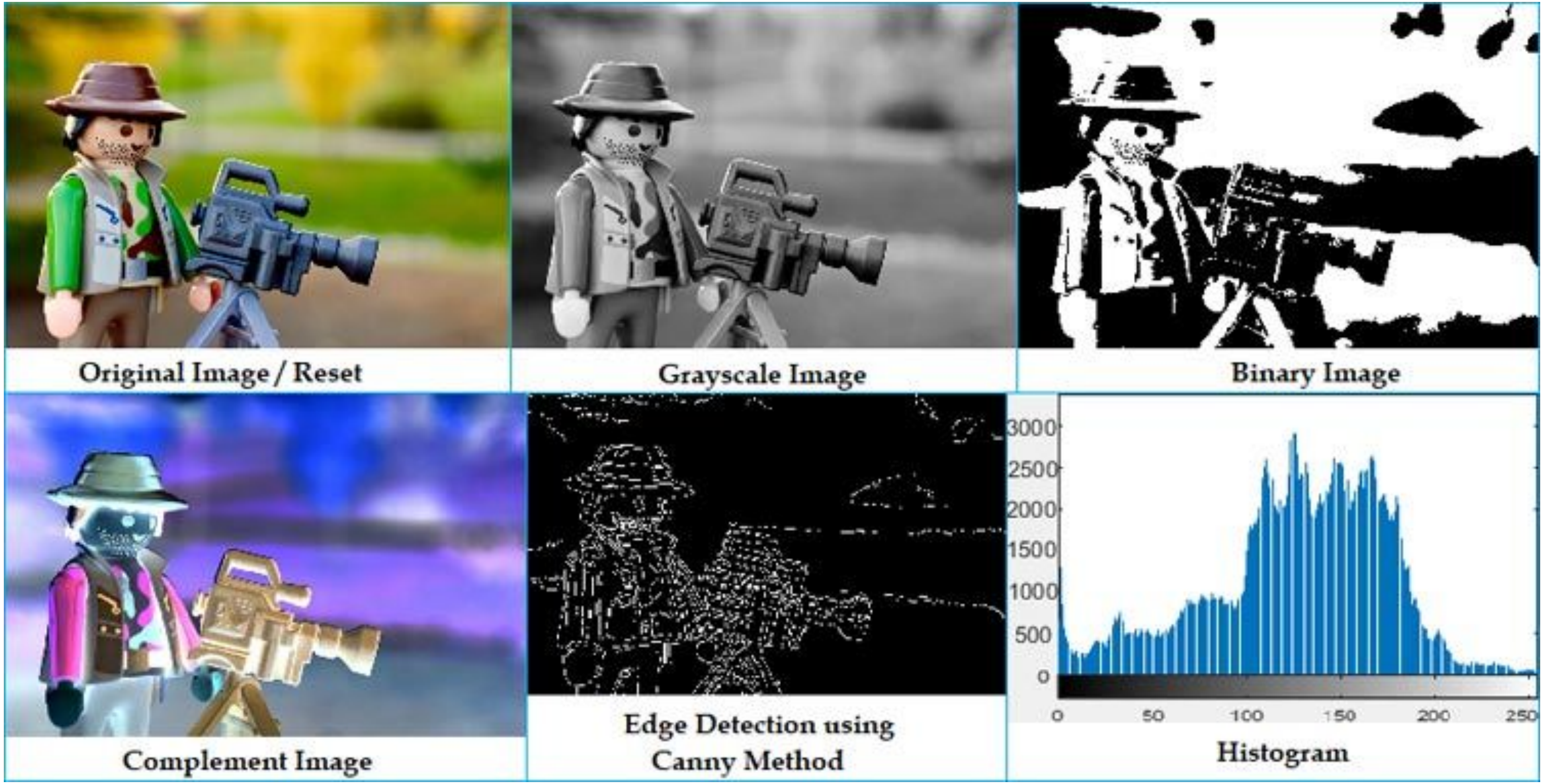


Image Processing



Original Image / Reset

Grayscale Image

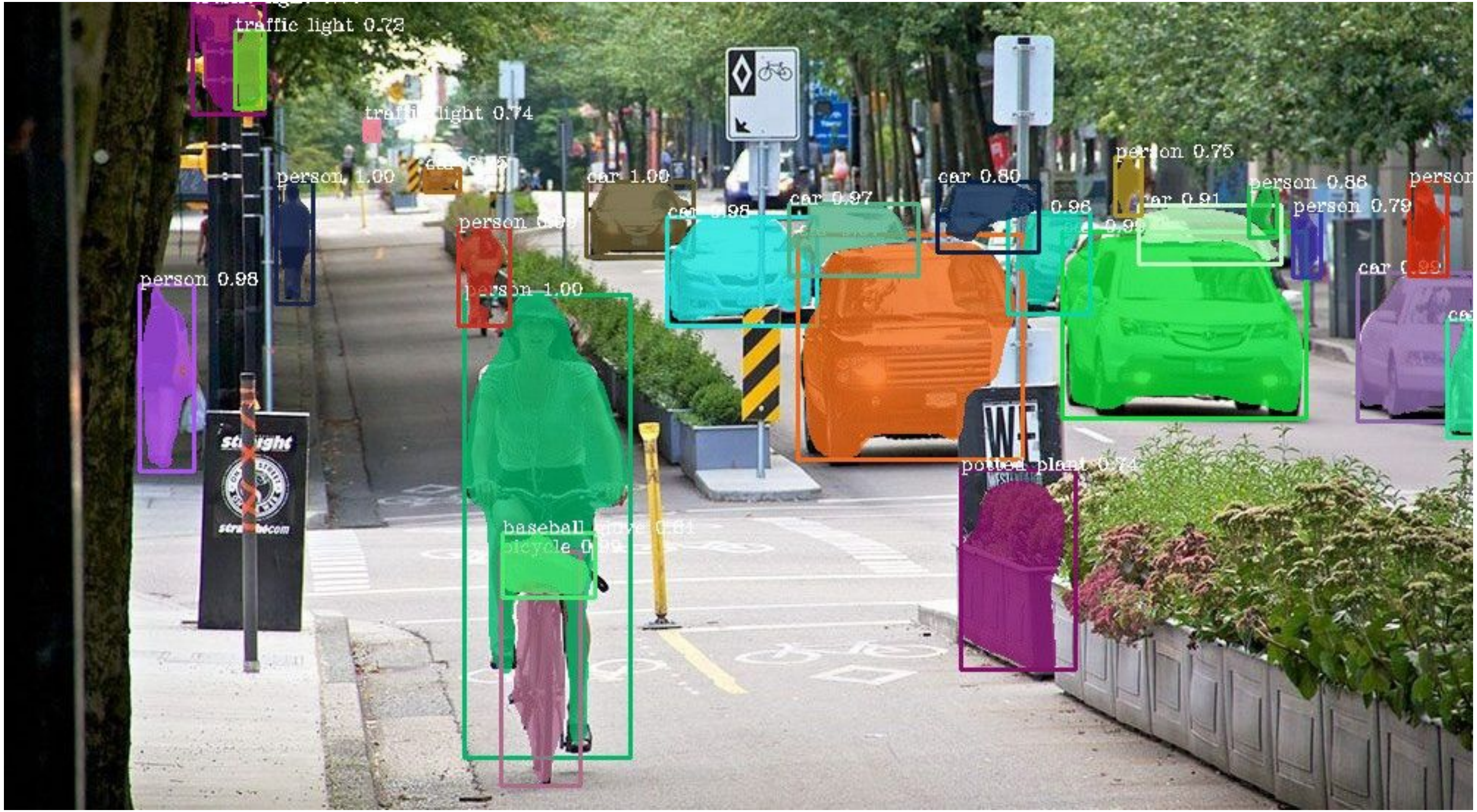
Binary Image

Complement Image

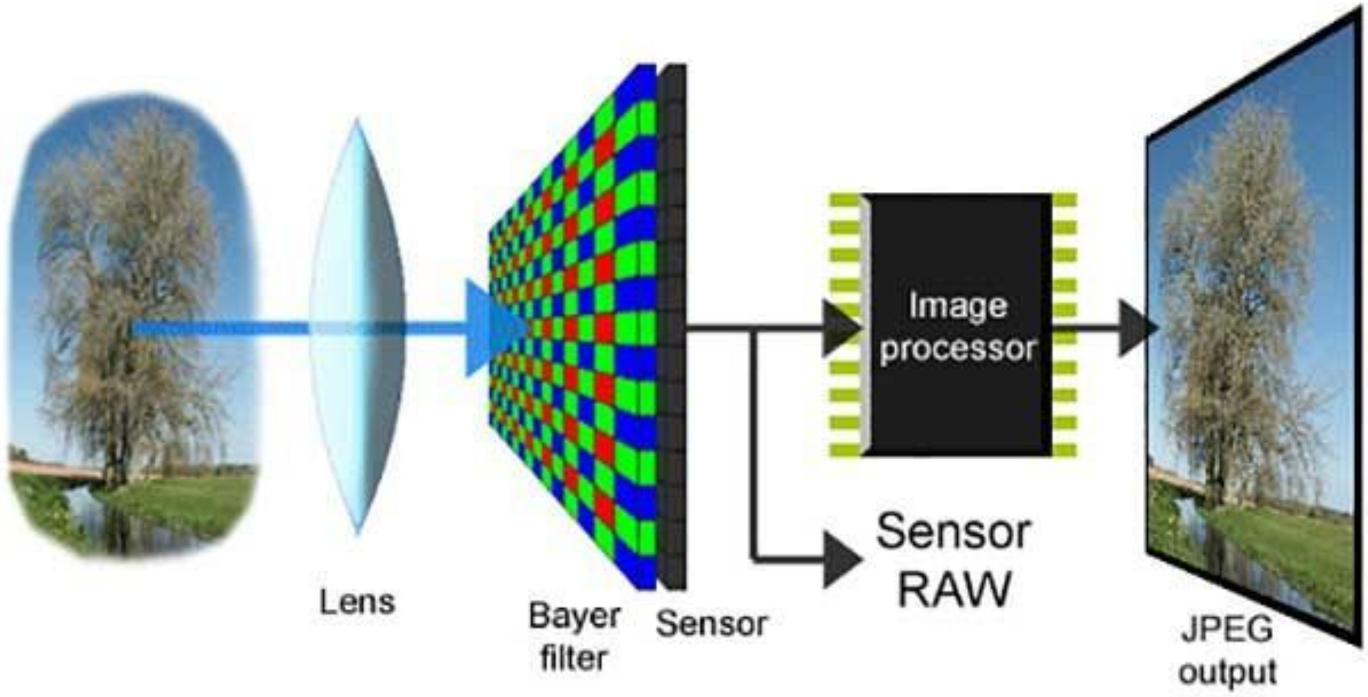
Edge Detection using Canny Method

Histogram

Image Processing



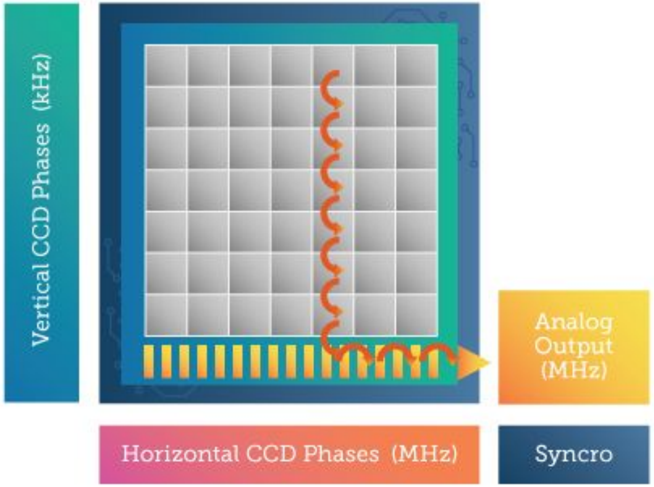
Digital Image Creation



Digital Image Color Channels

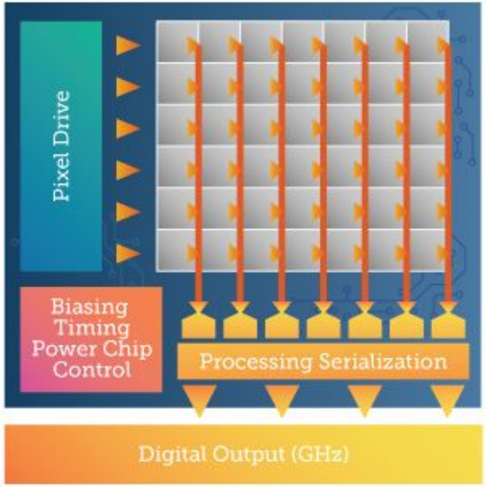
CCD

Photon to Electron
Conversion (Analog)

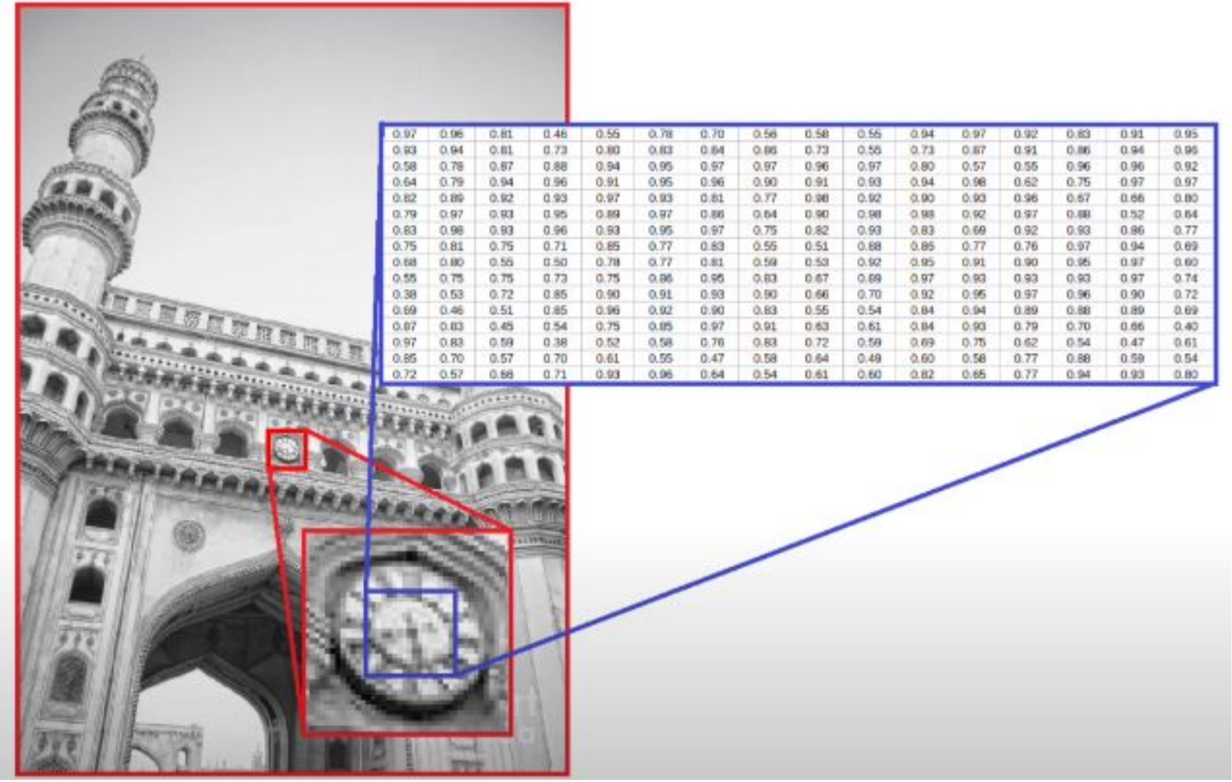


CIS

Photon to Voltage
Conversion (Digital)

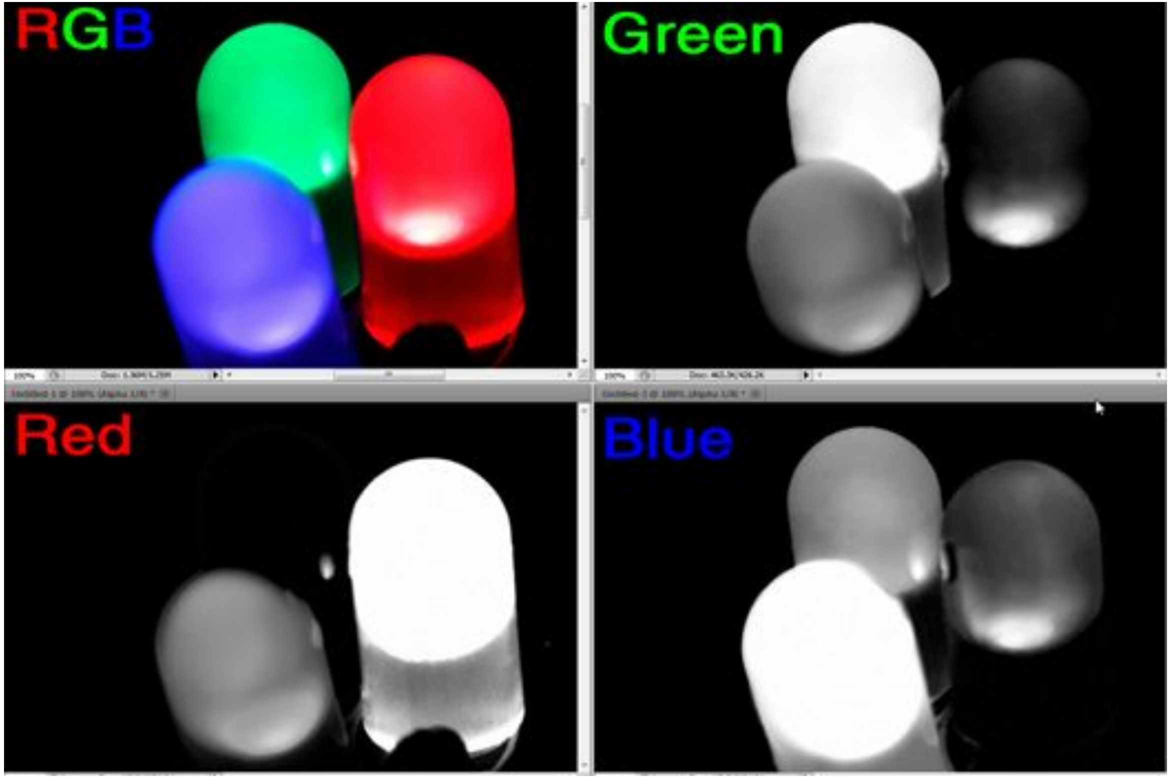


Digital Image Representations



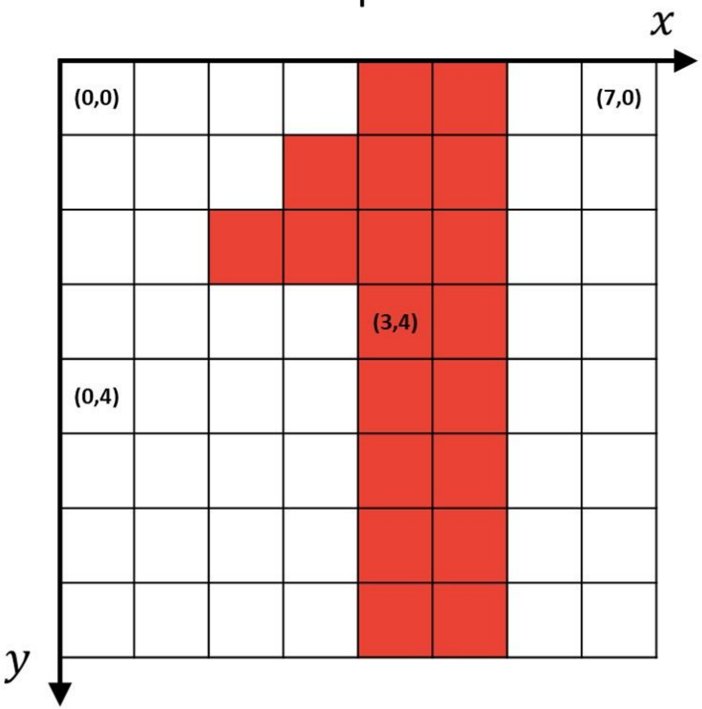
0.97	0.96	0.81	0.46	0.55	0.78	0.70	0.56	0.58	0.55	0.94	0.97	0.92	0.83	0.91	0.95
0.93	0.94	0.81	0.73	0.80	0.83	0.84	0.86	0.73	0.55	0.73	0.87	0.91	0.86	0.94	0.96
0.58	0.78	0.87	0.88	0.94	0.95	0.97	0.97	0.96	0.97	0.80	0.57	0.55	0.96	0.96	0.92
0.64	0.79	0.94	0.96	0.91	0.95	0.96	0.90	0.91	0.93	0.94	0.98	0.62	0.75	0.97	0.97
0.82	0.89	0.92	0.93	0.97	0.93	0.81	0.77	0.98	0.92	0.90	0.93	0.96	0.67	0.66	0.80
0.79	0.97	0.93	0.95	0.89	0.97	0.86	0.64	0.90	0.98	0.98	0.92	0.97	0.88	0.52	0.64
0.83	0.98	0.93	0.96	0.93	0.95	0.97	0.75	0.82	0.93	0.83	0.69	0.92	0.93	0.86	0.77
0.75	0.81	0.75	0.71	0.85	0.77	0.83	0.55	0.51	0.88	0.86	0.77	0.76	0.97	0.94	0.69
0.68	0.80	0.55	0.50	0.78	0.77	0.81	0.59	0.53	0.92	0.95	0.91	0.90	0.95	0.97	0.60
0.55	0.75	0.75	0.73	0.75	0.86	0.95	0.83	0.67	0.89	0.97	0.93	0.93	0.93	0.97	0.74
0.38	0.53	0.72	0.85	0.90	0.91	0.93	0.90	0.66	0.70	0.92	0.95	0.97	0.96	0.90	0.72
0.69	0.46	0.51	0.85	0.96	0.92	0.90	0.83	0.55	0.54	0.84	0.94	0.89	0.88	0.89	0.69
0.87	0.83	0.45	0.54	0.75	0.85	0.97	0.91	0.63	0.61	0.84	0.93	0.79	0.70	0.66	0.40
0.97	0.83	0.59	0.38	0.52	0.58	0.76	0.83	0.72	0.59	0.69	0.75	0.62	0.54	0.47	0.61
0.85	0.70	0.57	0.70	0.61	0.55	0.47	0.58	0.64	0.49	0.60	0.58	0.77	0.88	0.59	0.54
0.72	0.57	0.68	0.71	0.93	0.96	0.64	0.54	0.61	0.60	0.82	0.65	0.77	0.94	0.93	0.80

Digital Image Color Channels



Digital Image Representations

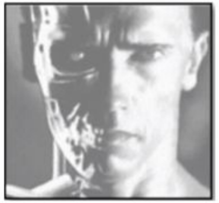
8 × 8 pixels



Pixel-level Image Processing

$f(x,y)$

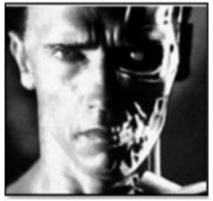
→



$f(x,y) + 20$

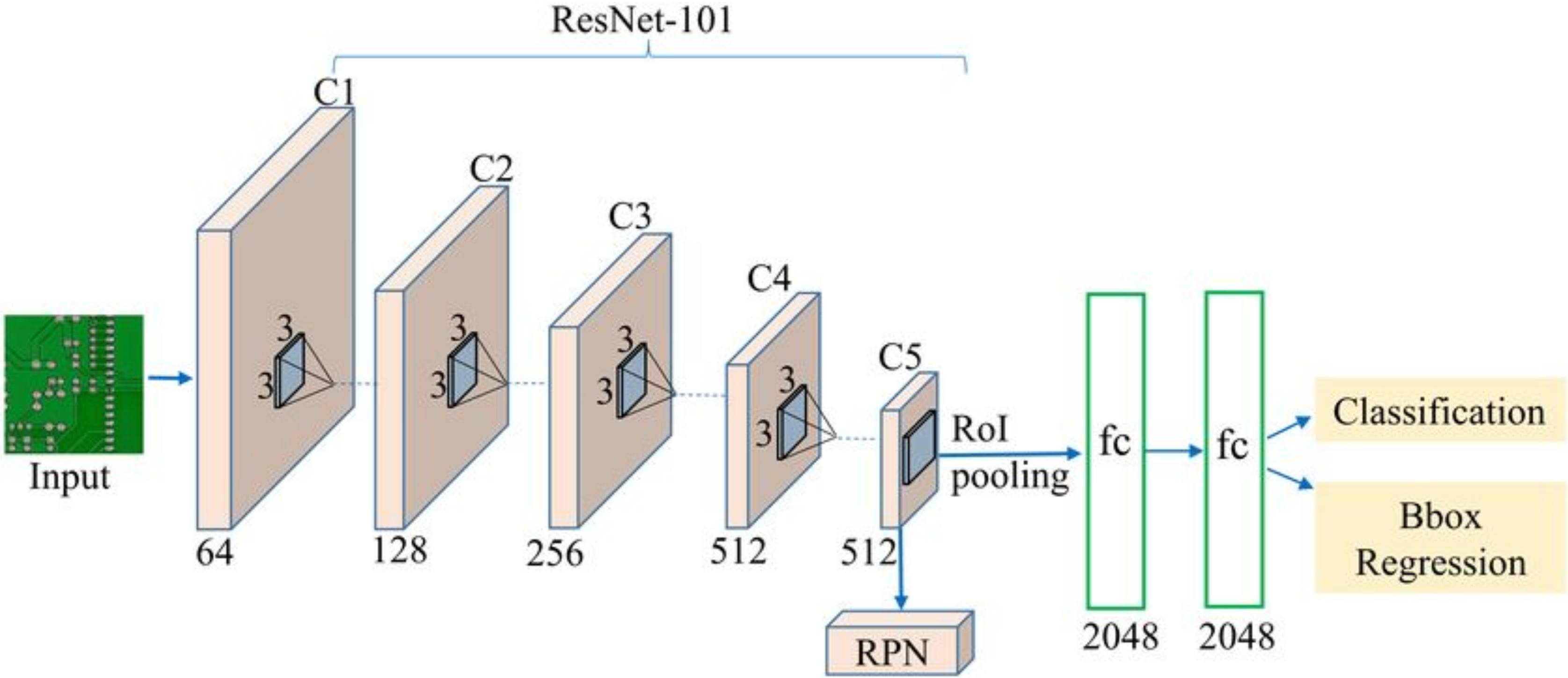
$f(x,y)$

→

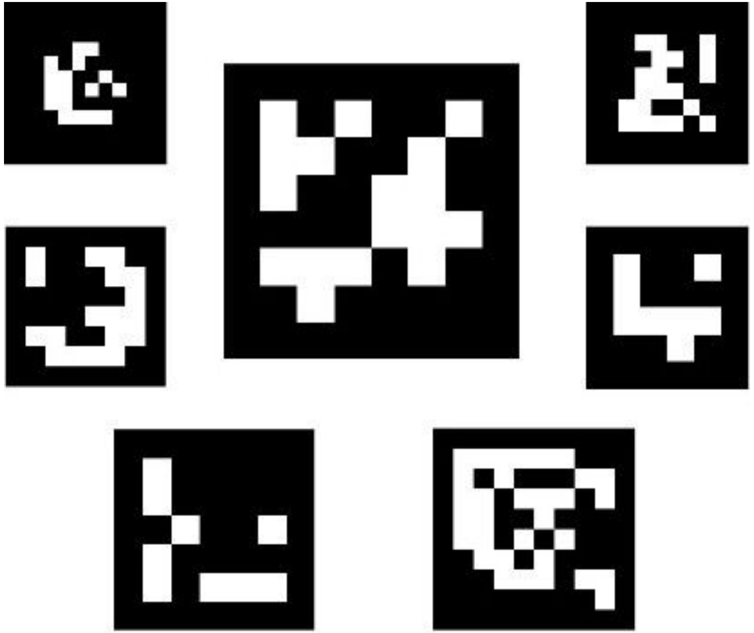


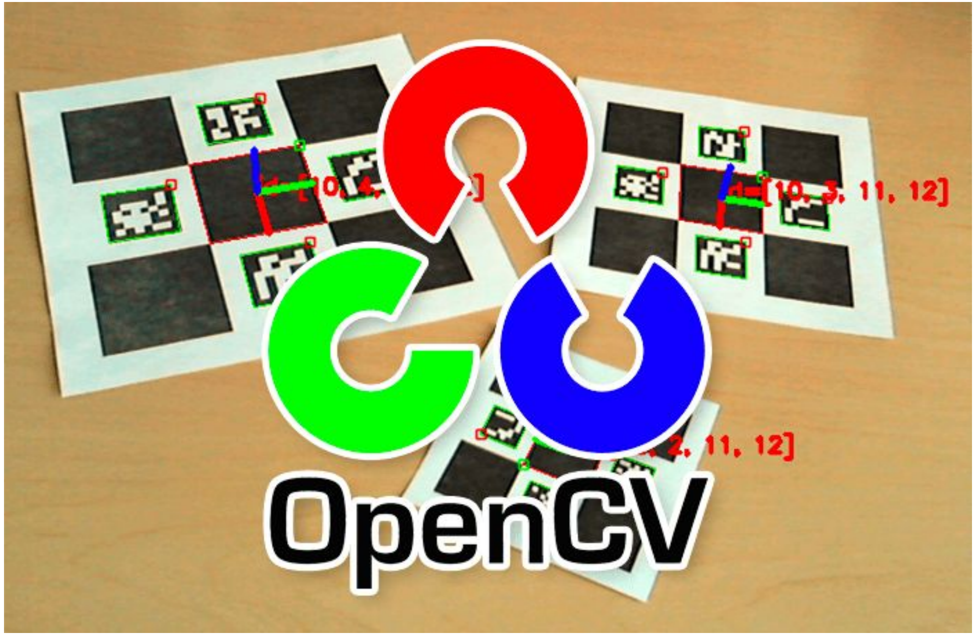
$f(-x,y)$

High-level Image Processing



ArUco Marker Detection





OpenCV ArUco Resources

- Official Tutorial (C++): https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html
- Pyimagesearch Tutorial: https://pyimagesearch.com/2020/12/21/detecting-aruco-markers-with-opencv-and-python/
- Video Tutorial: https://youtu.be/cIVZRuVdv1o

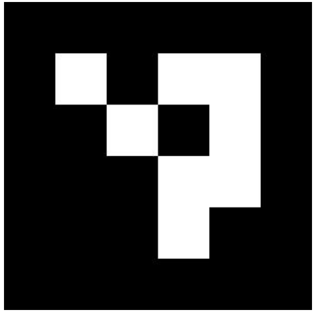
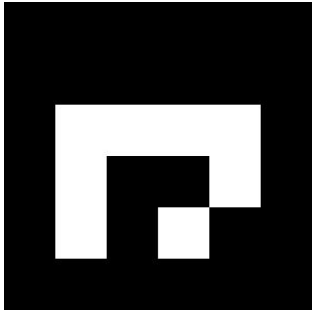
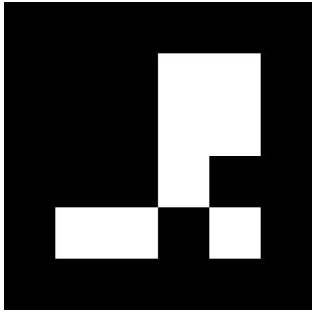
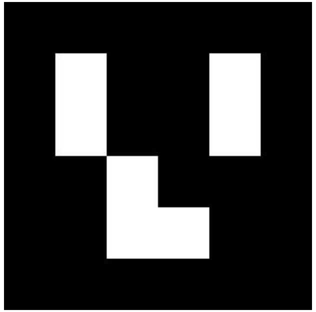
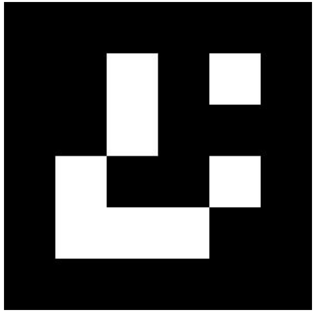
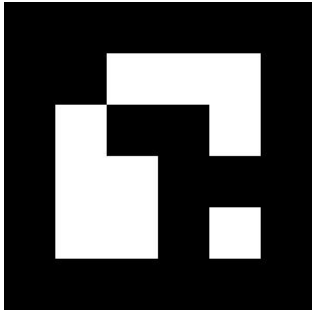
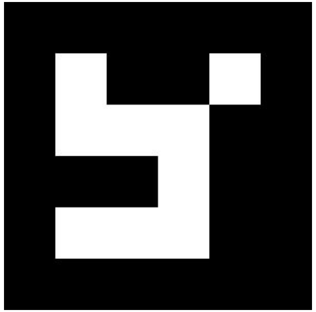
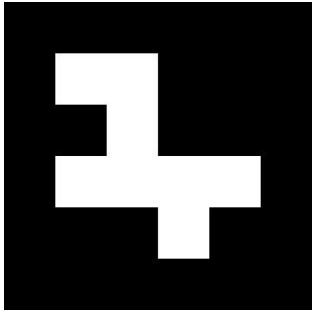
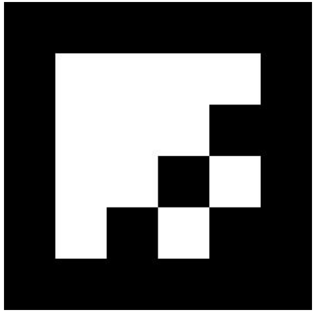
opencv-python Installation

```
sudo apt install python3-opencv
```

Generate ArUco Markers

```
import numpy as np
import cv2


aruco_dict = cv2.aruco.Dictionary_get(cv2.aruco.DICT_4X4_50)  # get ArUco dictionary
aruco_arr = np.zeros((300, 300, 1), dtype=np.uint8)  # create an array to save marker
for i in range(9):  # generate 9 markers
    cv2.aruco.drawMarker(aruco_dict, i, 300, aruco_arr, 1)
    cv2.imwrite('aruco/4x4_' + str(i) + '.jpg', aruco_arr)
```

OpenCV Video Capture

```python
import cv2 as cv
from picamera2 import Picamera2

# SETUP
cam = Picamera2()
config = cam.create_still_configuration()
cam.configure(config)
cam.start()

# LOOP
while True:
    im = cam.capture_array()
    im_rgb = cv.cvtColor(im, cv.COLOR_BGR2RGB)
    im_resize = cv.resize(im_rgb, (800, 600))
    cv.imshow("Camera", im_resize)
    if cv.waitKey(1) == ord('q'):
        break
```

Detect ArUco Markers

```python
import cv2 as cv
from picamera2 import Picamera2
import numpy as np

# SETUP
cam = Picamera2()
config = cam.create_still_configuration()
cam.configure(config)
cam.start()
aruco_dict = cv.aruco.Dictionary_get(cv.aruco.DICT_4X4_50)  # aruco dictionary
aruco_params = cv.aruco.DetectorParameters_create()

# LOOP
while True:
    if cv.waitKey(1) == ord('q'):
        break
    im = cam.capture_array()
    im_rgb = cv.cvtColor(im, cv.COLOR_BGR2RGB)
    im_resize = cv.resize(im_rgb, (400, 300))
    corners, ids, reject_candidates = cv.aruco.detectMarkers(
        im_resize,
        aruco_dict,
        parameters=aruco_params,
    )
    top_left_coords = corners[0][0][0].astype(int)
    bot_right_coords = corners[0][0][2].astype(int)
    print(corners, ids)
    image = cv.rectangle(im_resize, top_left_coords, bot_right_coords, (0, 255, 0), 2)
    cv.imshow("Camera", image)
```