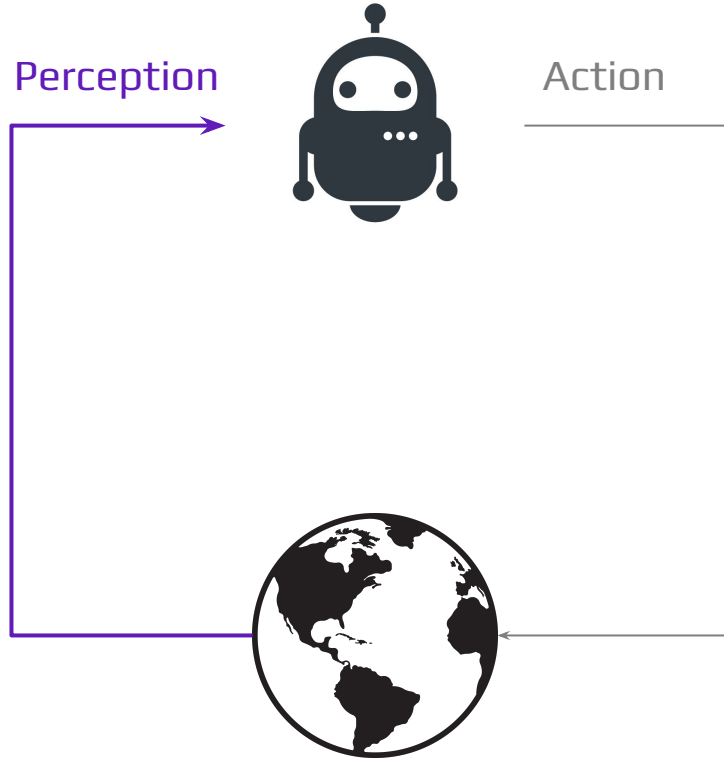


ENGR 4421: Robotics II

Robotic Vision

11/01/2022

Robot Perception



Robotic Vision Introduction

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

Robotic Vision

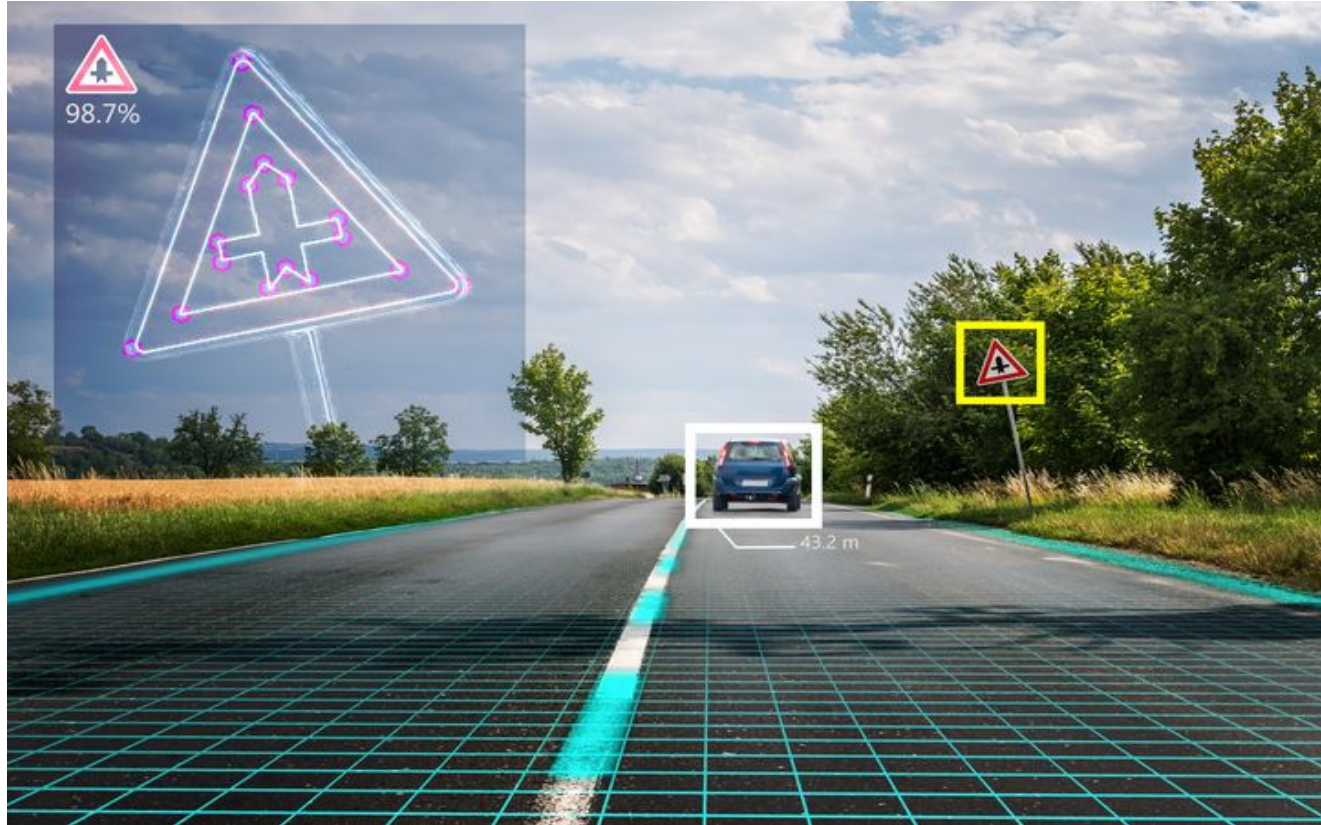


Image Processing



Original Image / Reset



Grayscale Image



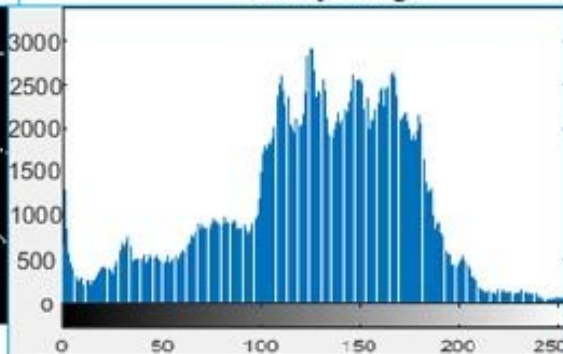
Binary Image



Complement Image

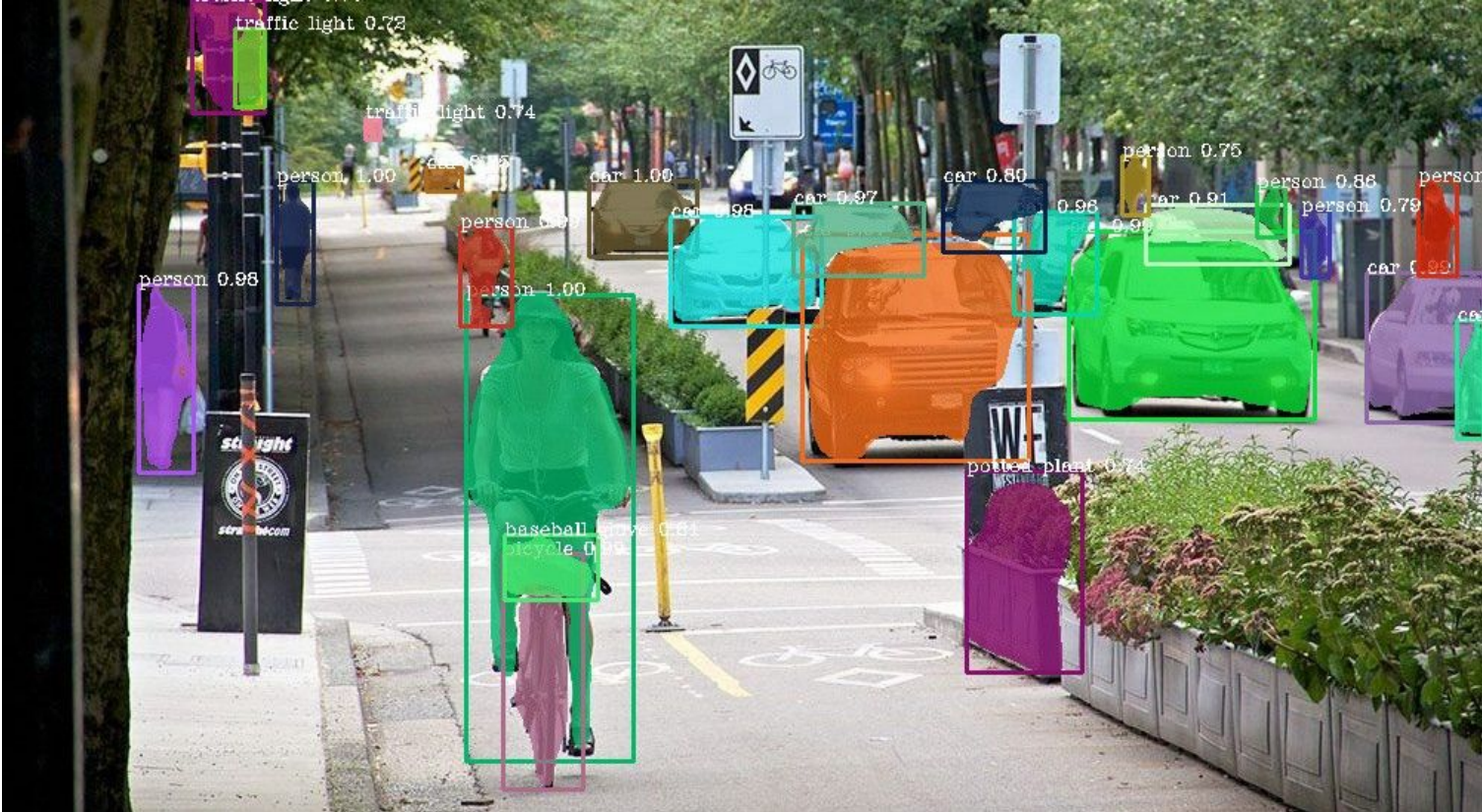


Edge Detection using
Canny Method

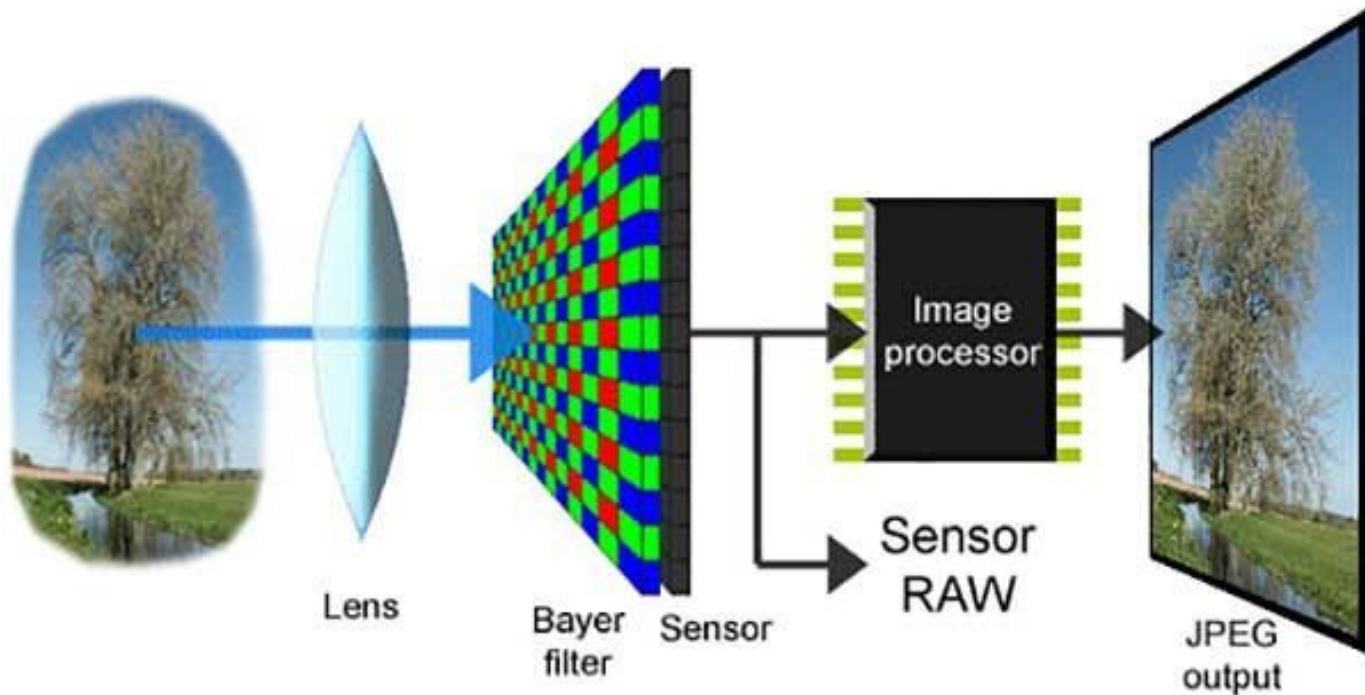


Histogram

Image Processing

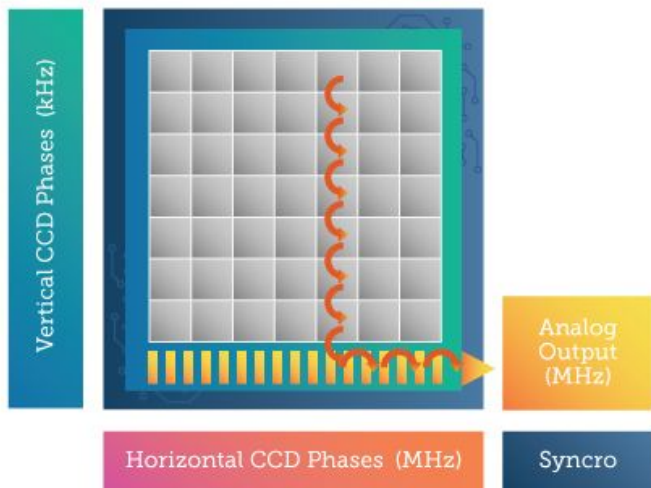


Digital Image Creation

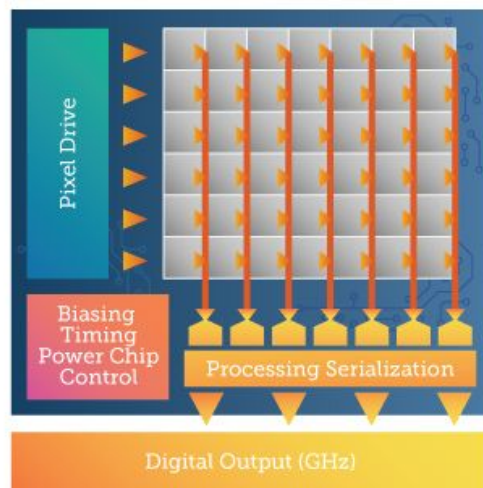


Digital Image Color Channels

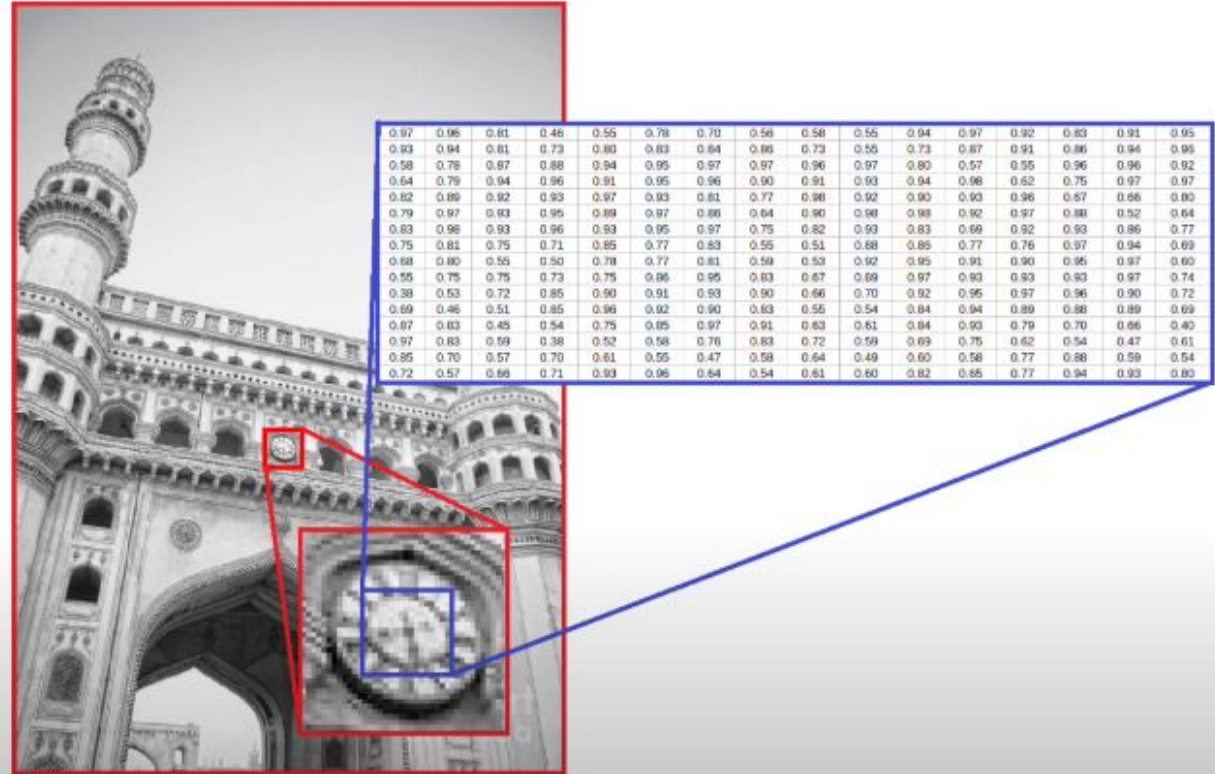
CCD
Photon to Electron
Conversion (Analog)



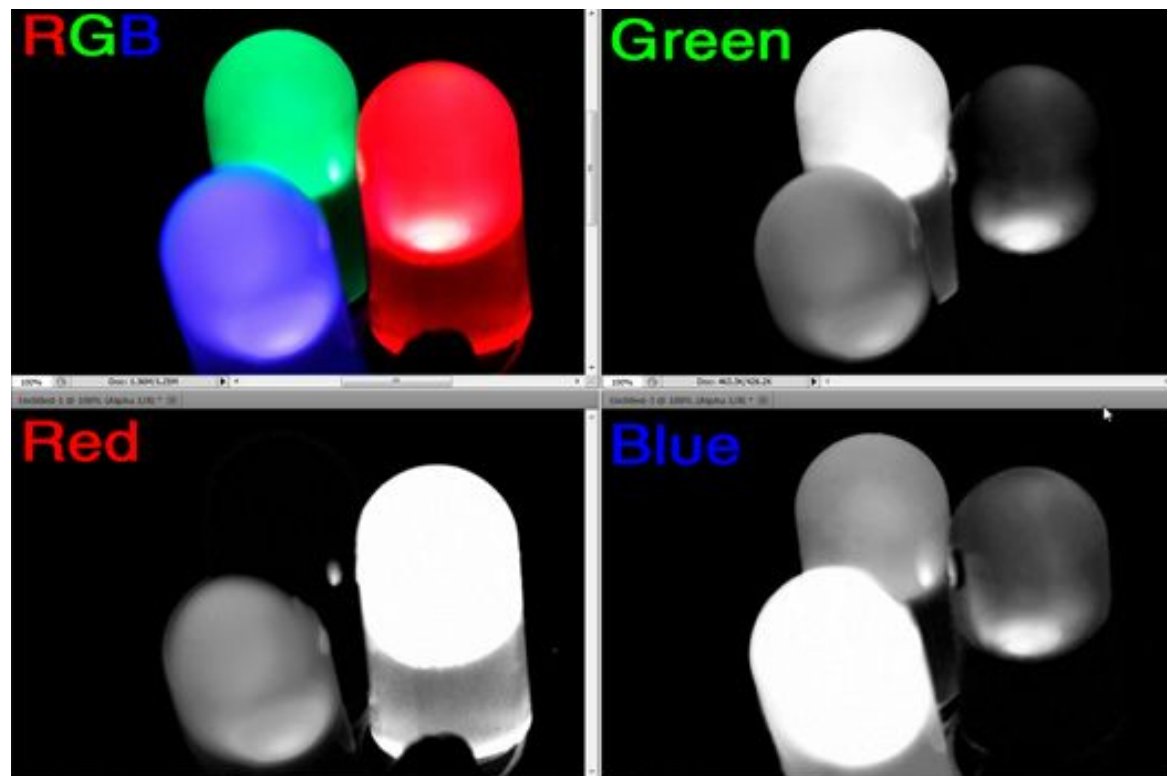
CIS
Photon to Voltage
Conversion (Digital)



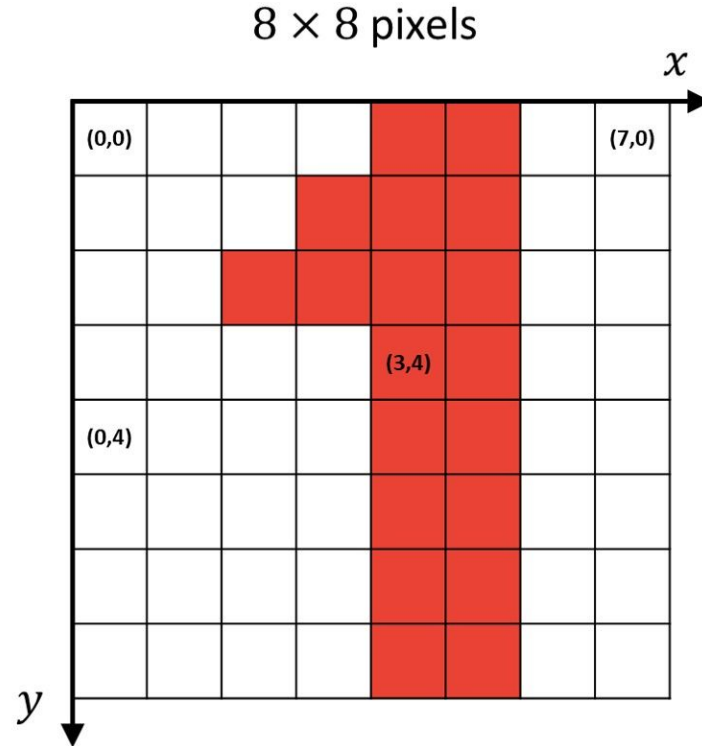
Digital Image Representations



Digital Image Color Channels



Digital Image Representations



Pixel-level Image Processing



$f(x,y)$



$f(x,y) + 20$

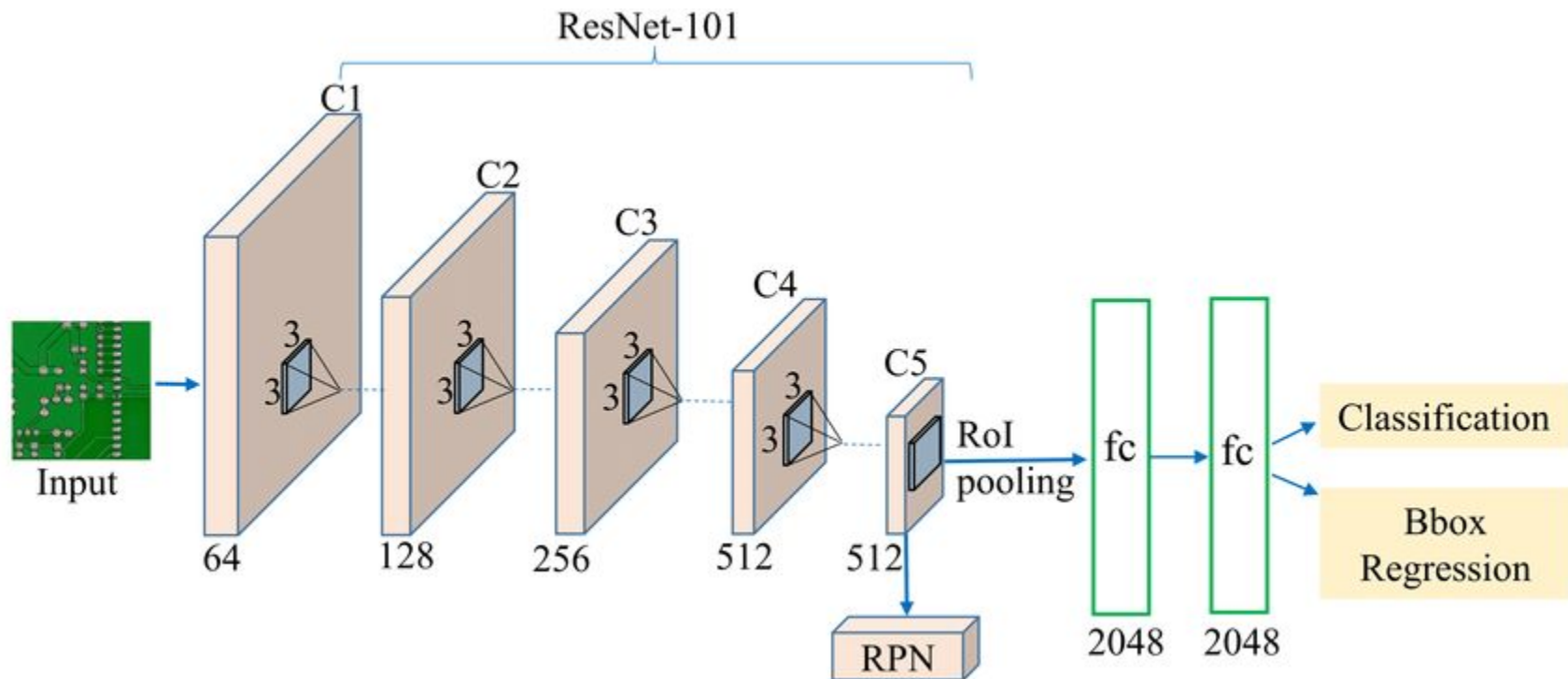


$f(x,y)$

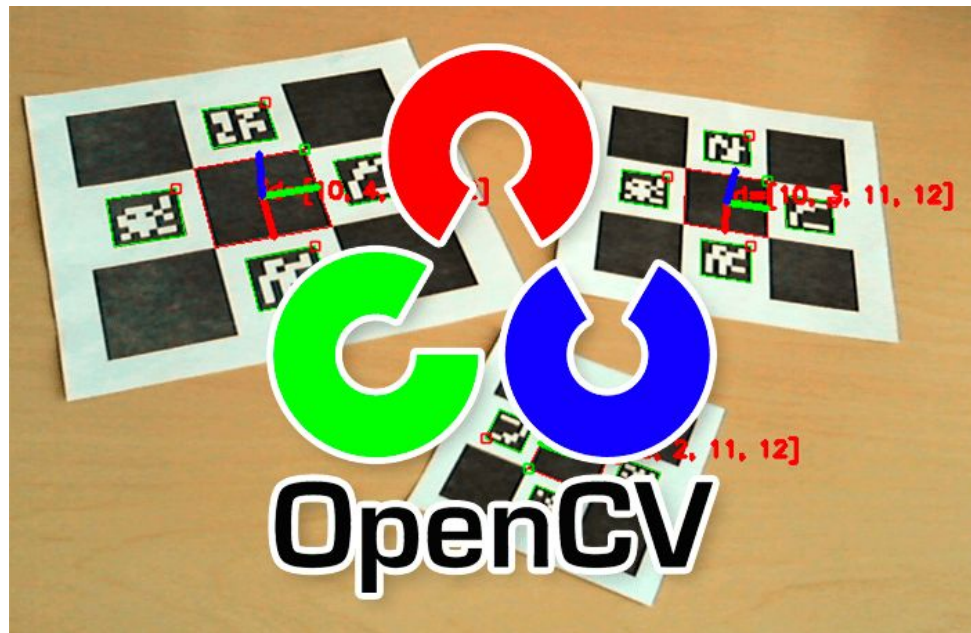
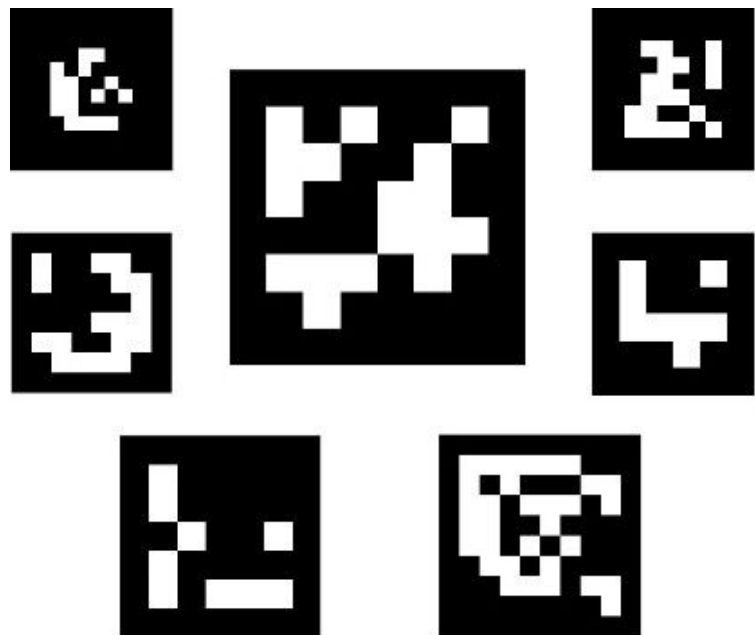


$f(-x,y)$

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>

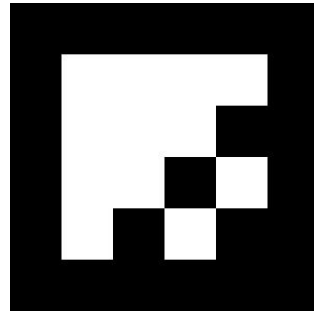
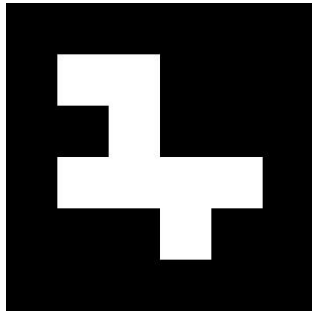
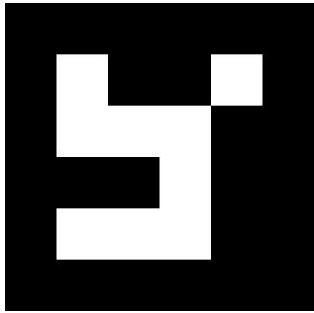
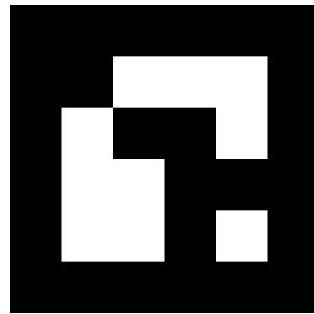
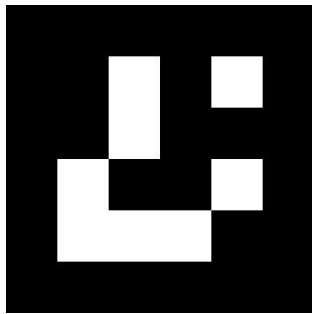
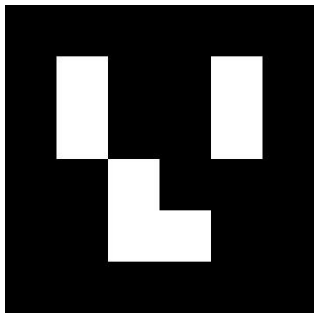
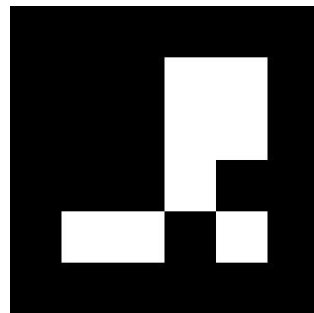
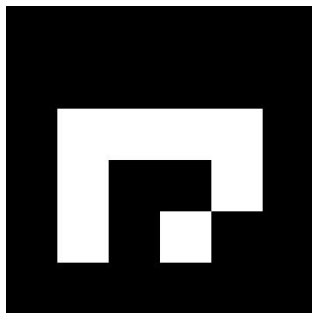
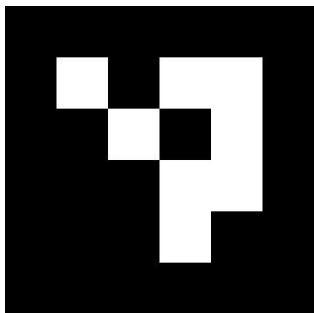
opencv-python Installation

- <https://pypi.org/project/opencv-python/>
- <https://pypi.org/project/opencv-contrib-python/>

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

```
import numpy as np
import cv2 as cv
cap = cv.VideoCapture(0)
if not cap.isOpened():
    print("Cannot open camera")
    exit()
while True:
    # Capture frame-by-frame
    ret, frame = cap.read()
    # if frame is read correctly ret is True
    if not ret:
        print("Can't receive frame (stream end?). Exiting ...")
        break
    # Our operations on the frame come here
    gray = cv.cvtColor(frame, cv.COLOR_BGR2GRAY)
    # Display the resulting frame
    cv.imshow('frame', gray)
    if cv.waitKey(1) == ord('q'):
        break
# When everything done, release the capture
cap.release()
cv.destroyAllWindows()
```

Detect ArUco Markers

```
# Load ArUco dictionary
aruco_dict = cv.aruco.Dictionary_get(cv.aruco.DICT_4X4_50)

# Initialize detector parameters
aruco_params = cv.aruco.DetectorParameters_create()

# To detect ArUco markers
corners, ids, reject_candidates = cv.aruco.detectMarkers(
    frame,
    aruco_dict,
    parameters=aruco_params,
)
```