Mid Report - A2 assignment

Sensors and Control for Mechatronic Systems (University of Technology Sydney)

41014 Sensors and Control

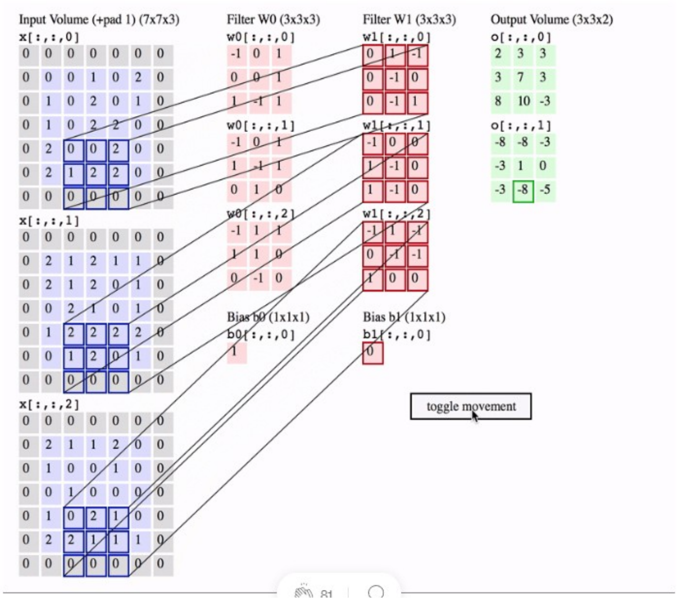
Mid review report

Group Project 8: Visual servoing of a handheld monocular camera

This report will be reviewing the works that have been done up to 17/April/2022 in preparation for the Group Project 8: Visual servoing of a handheld monocular camera. All the group members have been researching and learning about the technologies that is related to visual servoing. In particular, I have been investigating object detection in R-CNN, YOLO (you only look once) algorithms for visual servoing. In this report, I will be exploring these technologies to compare which is better solution for the project so we can put our hands to actually work on the codes for the next weeks left.

For a monocular camera to detect a particular object, object detection technology is essential. Object detection is a technology related to image processing that is used for detecting certain objects in the image or video. By using this, we can detect an object in the image or video or even real time camera view. Before making object detection happen, there are few things that have to be learned. These things are deep learning, neural network, datasets, and image recognition tools.

Convolutional neural networks (CNN)



Input Volume (+pad 1) (7x7x3)

x[:,:,0]

0	0	0	0	0	0	0
0	0	0	1	0	2	0
0	1	0	2	0	1	0
0	1	0	2	2	0	0
0	2	0	0	2	0	0
0	2	1	2	2	0	0
0	0	0	0	0	0	0

x[:,:,1]

0	0	0	0	0	0	0
0	2	1	2	1	1	0
0	2	1	2	0	1	0
0	0	2	1	0	1	0
0	1	2	2	2	2	0
0	0	1	2	0	1	0
0	0	0	0	0	0	0

x[:,:,2]

0	0	0	0	0	0	0
0	2	1	1	2	0	0
0	1	0	0	1	0	0
0	0	1	0	0	0	0
0	1	0	2	1	0	0
0	2	2	1	1	1	0
0	0	0	0	0	0	0

Filter W0 (3x3x3)

w0[:,:,0]

-1	0	1
0	0	1
1	-1	1

w0[:,:,1]

-1	0	1
1	-1	1
0	1	0

w0[:,:,2]

-1	1	1
1	1	0
0	-1	0

Bias b0 (1x1x1)

b0[:,:,0]

1

Filter W1 (3x3x3)

w1[:,:,0]

0	1	-1
0	-1	0
0	-1	1

w1[:,:,1]

-1	0	0
1	-1	0
1	-1	0

w1[:,:,2]

-1	1	-1
0	-1	-1
1	0	0

Bias b1 (1x1x1)

b1[:,:,0]

0

Output Volume (3x3x2)

o[:,:,0]

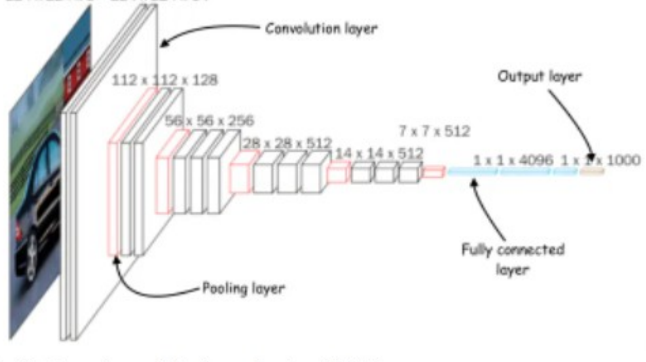
2	3	3
3	7	3
8	10	-3

o[:,:,1]

-8	-8	-3
-3	1	0
-3	-8	-5

Figure : Simran (2019)

Convolutional neural network was invented for capturing patterns in the multidimensional space. This means that CNN can be used to process in capturing patters in the images. As opposed to humans, CNN cannot see through the flat image and predict what is in the image. This has to go through some process in order to detect objects in the image. This process is called convolutional layers. They act like a filter to extract what is only needed from the image. Each filter will have different values and extracts. And each filter will be stacked for detection of visual pattens. For example, the lower layers will produce feature maps for vertical and horizontal edges, corners, and other simple patterns. The next layers can detect more complex patterns such as grids and circles Ben (2021). As these layer go deeper, more complex objects such as cars, house, person, cats can be detected. Although CNN can do image classification, it cannot tell where in the image the object is located. This creates problems as we need something that can tell where the object is located. This problem can be solved by having supervised machine learning, which we can train the models on labeled example.

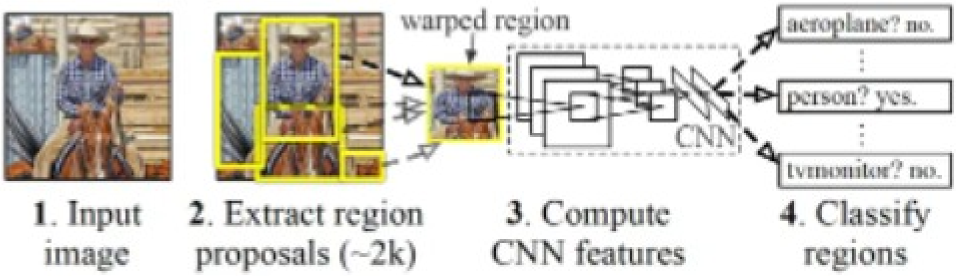


Architecture of convolutional neural network (CNN)

Figure 1: Ben (2021)

Region-based Convolutional Neural Network (R-CNN)

The R-CNN deep learning model



R-CNN architecture

Figure 2: Ben (2021)

R-CNN is composed of three components. Extract region, Compute CNN features, and Classify religions. In extract region, a region selector is used for finding regions of pixels in the image that can be having the objects, called (RoI), region of interest. 2000 RoIs can be generated in image. In compute CNN features, RoI is passed onto CNN for processing every region and then the CNN uses layers to encode the objects into a “single dimensional vector of numerical values” Ben (2021). Finally, in classify regions, classifier machine learning model will process the encoded objects and separate objects which we want from the background. R-CNN can be a good choice of object detection but there are few problems. The main problem is that as the calculation takes few seconds, real-time object detection is not possible. Thus, making us not use this for the group project.

YOLO (you only look once)

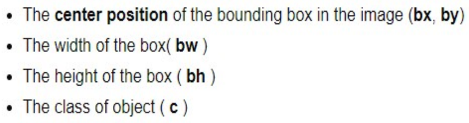
YOLO is a deep learning-based approach for object detection. Object detection algorithm that uses deep learning can be classified into two groups Classification based and Regression based. Lentin(2020)

Classification based

This uses RoI and convolution neural network as stated above topic (R-CNN) this process is computationally expensive and not suitable for real-time object detection.

Regression based

This is an algorithm that doesn’t use RoI but uses detection technique that can predict classes and bounding boxes in the image at one look. This is where the name YOLO (you only look once) came from. First, YOLO algorithm works by dividing the image into various grids. Each grid having dimensions of S x S. after splitting the image the algorithm will detect objects in every grid cell that have been divided. After the detection, the cell will be combined to become a bounding box. Bounding box is consists of following parameters (figure 4). Each bounding box will consist of probability value (pc) which represents the probability of the object that is in the bounding box.



- The **center position** of the bounding box in the image (**bx**, **by**)
- The width of the box(**bw**)
- The height of the box (**bh**)
- The class of object (**c**)

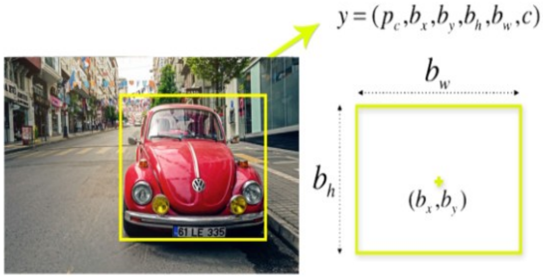


Figure : Lentin(2020)

YOLO is very fast, and relatively accurate compared to other object detection algorithms which means that it is suitable for this group project. We will be having further investigation on this algorithm and see if this can fit to our project requirements

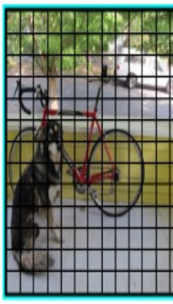


Figure : Hmrish

Neural networks

Neural networks, also known as artificial neural networks (ANNs) or simulated neural networks (SNNs), are a subset of machine learning and are at the heart of deep learning algorithms IBM (2020). The we will be using open-source neural networks such as darknet for object detection.

Figure 3: Lentin (2020)

Darknet is an open-source neural network frame that is written in C language and CUDA-technology this allows it to be really fast and making real-time predictions. This also allows us to have access to objects that have been pretrained datasets easily which means that we don't have to have all sorts of objects pre trained which can be time consuming. For this project, we were required to have simple object that is easy to detect, which means that we will have a ball, or checkerboard pretrained by taking lots of photos of it.

Microsoft Planner

The following figure is Microsoft planner that I have been plotting to track and plan the works for this project. Currently, background research and downloading actual software (YOLO) to play around with it has been done and starting to code is the next step which will be started from this week.

Figure 4: Microsoft planner

Reference

IBM Cloud Education (17/08/2020) Neural Networks

https://www.ibm.com/cloud/learn/neural-networks#:~:text=Neural%20networks%2C%20also%20known%20as,neurons%20signal%20to%20one%20another.

Lentin Joseph, A Gentle Introduction to YOLO v4 for Object detection in Ubuntu 20.04

https://robocademy.com/2020/05/01/a-gentle-introduction-to-yolo-v4-for-object-detection-in-ubuntu-20-04/

Introduction to how CNNs Work, Simran Bansari (13/03/2019)

https://medium.datadriveninvestor.com/introduction-to-how-cnns-work-77e0e4cde99b

An introduction to object detection with deep learning, Ben Dickson (21/06/2021)

https://bdtechtalks.com/2021/06/21/object-detection-deep-learning/

YOLO: Real-Time Object Detection Explained, Hmrishav Bandyopahyay (19/03/2022)

https://www.v7labs.com/blog/yolo-object-detection#how-yolo-works