Object Sorting System using Visual Inspection through Computer Vision

Daksha Sunil Saraf

Abstract: ***Object tracking is one of the major fundamental challenging problems in computer vision applications as difficulties in tracking objects can arise due to intrinsic and extrinsic factors like deformation, camera motion, motion blur and occlusion, this project presents a helpful application with a real-time object detection system that can automatically capture the user-defined important objects and segregate them instantly with the help of a rack and pinion mechanism. The problem discussed in this article will be solved using object detection using deep neural network especially convolutional neural networks. We would also try to present an embedded technique for real-time object detection and recognition that runs at high frames per second (FPS) by applying deep learning for computer vision. To address the real-life problems where, instant accurate decision taking capability of a robot is not yet achieved. For this, object detection is the new lookout. A camera module used to track objects. But object tracking is one of the major fundamental challenging problems in computer vision applications. There are many difficulties in tracking objects which arise due to intrinsic and extrinsic factors like deformation, camera motion, motion blur and occlusion. This project presents a helpful application with a real-time object detection system that can automatically capture the user-defined important objects. By identifying these users defined objects we shall segregate them instantly with the help of a robotic arm. The problem discussed in this article will be solved using object detection using deep neural network especially convolutional neural networks. We would also try to present an embedded technique for real-time object detection and recognition that runs at high frames per second (FPS) by applying deep learning for computer vision. Thus, by integrating computer vision applications with deep learning and robotics, we can achieve a more reliable, accurate and time conserving technology for the industries with heavy machinery.***

Keywords: Object Tracking, Hardware System, Conveyor

1.Introduction

1.1Background

In manufacturing industries, the existing system for raw material handling is done by laborer's. In a sense, raw materials used in concrete are just like the food we eat: They should be of good quality, they must be stored and handled properly, and they must not be contaminated. The old adage "You are what you eat" comes to mind, because the quality of raw materials going into a concrete mix has a direct effect on how the final product will turn out – in both fresh and hardened states. With that in mind, the storage and handling of raw materials will play an important role in determining a plant's layout and equipment needs. When the raw materials arrive at the factory site the first task is to segregate the usable and unusable material for which in existing system, they use the manpower to segregate it. In this system the amount paid to the labourers is too high, the time consumed by them is also more to perform segregation and the accuracy level of segregating the materials by man power is also comparatively low.

2.Problem Statement

To design a material sorting system with visual inspection of material using computer vision-based object detection technique

Aim

- The aim of this project is to build a object sorting system using object detection using computer vision so it could identify the object in frame in the provided to it with maximum expected output also to build an hardware system having conveyor gear mechanism and arm mechanism to make the whole system automated.
- To enhance the speed of recognition of object and if possible, to take instant action to through processors in system
- To take minimum time for this whole process to be successfully done

3.Objectives

Objective of the project is to create the electronic material handling system by visual inspection using object detection which can be used to reduce the efforts of workers as well as to reduce the time spent in inspection of the components (Tomatoes), during their arrival at manufacturing station.

The system can be upgraded by using lasers, ultrasonic sensors to handle multiple tasks at a time. The manufacturing industries always have this need to segregate out the unwanted raw material from a heap, for example- in tomato ketchup manufacturing industries, there is a need to separate out the rotten or unripe tomatoes while they are travelling on the conveyor belt to the washer. Thus, the project targets the manufacturing industries as their audience.

4.Design and Implementation

Hardware Design

Design of Conveyor Belt

Belt speed: 2 m/s
Capacity of conveyor (M) = 100 kg/hr
Diameter of driving and driven pulley = D1 = D2 = D (Assumed). Hence,
Arc of contact = 180 – (D1 – D2)/C * 60 = 180^{0} K1 = 1.25 (For Capron material),
K2 = 40 (180^{0} Arc of contact) Zp = 1 (single led flat belt)
As we know,
Dmin = k1 * k2 * Zp Dmin = 1.25*40*1 Dmin = 50mm
Designing flat belt by using manufacturing data Standard diameter for pulley 50mm

$$N1/N2 = D/d \quad (1)$$

Length of Flat Belt

$$L=2C+\pi/2(D + d) +(D-d)^2\ C \quad (2)$$

C=600
mm=2
feet (assumed) L=2(600) + π (50+50) + 0
L=1357.1 mm ≈ 1360 mm L=1.36
m

$$Tb = 1.37*w*f*L*g \quad (3)$$

Where, Tb is in N
W = weight of object 100gm = 0.1 kg
f = coefficient of friction = 0.35 (Rubber Cotton belt)
L = Conveyor length in meters, conveyor length is half of the total length
g = Acceleration due to gravity =9.81m/s^2
Tb = 1.37*0.1*0.35*2*9.81
= 0.94N ≈1N

$$Pb = Tb * v/1000 \quad (4)$$

Where,
Pb is in KW

Tb = steady state belt tension in N v = speed of belt in m/sec = 2 m/sec
Pb = 1*2/1000 = 0.002 KW = 2 Watt
Belt width: M= ρC(0.9B – 0.05)^2 V As,
M = 100kg/hr
ρ = 480 kg/m^3
C = 0.075 (for horizontal flat belt)= 2m/s
Hence, B ≈ 70mm Hence,
B = 100mm (standard)

3.1.2 Design of Rack and Pinion Mechanism

Material: nylon 66/ polyacetal
Density: 1.41 g/cm^3 => 1.41x10-6 kg/mm^3 Mass of plate:
Volume of plate = 40x40x5= 8000 mm^3
Mass = density x volume = 1.41x10-6 x 8000 = 11.28 gm
Mass of rack
Volume of rack = 150x20x10 = 30000 mm^3
Mass = density x volume = 1.41x10-6 x 30000= 42.3 gm
Mass of object to be pushed = 500gm
Total mass = mass of object + mass of rack + mass of plate
= 500 + 42.3 + 11.28= 553.58 gm = 600 gm

Hence, Force required by the rack to push the weight Fr = u mg + ma
For servo motor 0 – 60^{0}
☾ 2 sec Hence,
ω = 30^{0} ☾ 0.523 rad/sec
Hence, for 1 sec, α = 0.523 rad/s^2
Fr = 1.176 + 0.3138 r
As we know, Tp
= Fr * r
Where,
Tp = Torque applied by pinion
Fr = Force on rack
r = radius of pinion
Tp=(1.176 + 0.3138r)r r= 24mm Force applied by the rack
Fr= 1.178 + 0.3138r Fr=1.183N

For gear of rack and pinion Pressure angle= 20^{0}
RPM of pinion (Np)= 5 rpm
No of teeth on pinion = 18
No of teeth on gear = 30
Calculation for rack and pinion module Yp = 0.484 – (2.87/Zp) = Yp= 0.305 Yg = 0.484 – (2.87/Zg)= Yg= 0.380
$\sigma bp. Yp < \sigma bg. Yg$
Opinion is weaker in bending than gear. Design pinion for bending Assuming, b =10m Pb = $\sigma bp * b * m * y$= Pb= 92.4 m^2
Cv = 3/3+v
V = 4.6158m m/s
Pt = 2 mt /dp = 0.0217 /m
Peff = Cs * t / Cv Pb = FOS * Peff = 0.19 mm std module is 0.5 mm B = 10 x m = 5mm Rack & length = 15cm

Calculation for Gear Box for DC Motor to Conveyor Shaft

Power at pinion = V.I = 12x5= 60 Watt
Width of gear = b = 10mm RPM of pinion = Np = 100RPM FOS = 1.5

Design of Frame

Self-weight of link
Density of aluminum = 2700 M = density x volume
= 2700 * [0.004b – 1.6 x 10^{-5}] x 3.2
= 8640 [0.004b – 1.6 x 10-5]

Moment of inertia, I = b^4 - $(b – 0.008)^4$ / 12
Distance of top most layer from the neutral point = b/2
b = 20mm Cross-sectional beam Thickness = 4mm
Breadth / width = 20mm Length of major beam = 0.8
Length of minor beam= 0.18

4.1.2 Selection of Bearing

Bearing are required to mount the roller pulley to the frame stand. The use of bearing is to provide the end support to the shaft as well as to provide a relative moveable support. So that pulley can rotate around its axis. We used roller contact bearing of ball type. The shaft diameter of pulley is 20mm so the required diameter of the pulley is to be of 20mm inside diameter. According to bearing designation, 6202 will be appropriate for this assembly. As it has 15mm inside diameter to mount the pulley shaft. Here we have 2 pulleys giving us two shafts which has four ends, so quantity of bearing required is 4.

Radial ball bearing: 6202 Dimensions of bearing:

Inside diameter = 20mm Outside diameter = 35mm Widths= 20mm

Hardware Specification Raspberry Pi 4 Model B

- Broadcom BCM2711, Quad-core cortex- A72 (ARM v8) 64-bit SOC @1.5GHz
- 1 GB, 2 GB or 4 GB LPDDR4 - 3200 SDRAM (depending on model)
- 2.4 Gz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
- Gigabit Ethernet
- 2 USB 3.0 ports, 2 USB 2.0 ports
- Raspberry Pi standard 40 pin GPIO header (fully backwards compatible with previous boards)
- 2× micro-HDMI ports (up to 4kp60 supported)
- 2- lane MIPI DSI display port
- 2- lane MIPI DSI camera port
- 4- pole stereo audio and composite video port
- H.265 (4kp60 decode), H264 (1080 p60 decode, 1080p30 encode)
- Open GL ES 3.0 graphics
- Micro SD card slot for loading operating system and data storage
- 5V DC via USB-C connector (minimum 3A*)
- 5V DC via GPIO header (minimum 3A*)
- Power over Ethernet (PoE) enabled (required separate PoE HAT)
- Operating temperature: 0-50 degree Celsius ambient

Camera Module

- 8-megapixel camera capable of taking photographs 3280×2464 pixels
- Capture video at 1080p30, 720p60 and 640×480p90 resolutions
- All software is supported within the latest version of the Raspbian Operating System
- Supports Raspberry Pi 1,2 and 3
- Package includes-1×8MP Raspberry Pi camera v2 (original)
- Dimensions: size: 25mm×23mm×9mm, weight: 3g
- Applications: CCTV security camera, motion detection time lapse photography

Servo Motor

- Operating Voltage is +5V typically
- Torque: 2.5kg/cm
- Operating speed is 0.1s/60°
- Gear Type: Plastic
- Rotation: 0°-180°
- Weight of motor: 9gm

DC Motor

• Gear Material	Plastic
• Rated RPM	100
• Operating Voltage (VDC)	12
• Rated Torque(kg-cm)	2.9
• Stall Torque (Kg-Cm)	11.4
• Load Current(A)	0.3
• No-Load Current	60mA
• Gearbox Diameter (mm)	37
• Motor Diameter(mm)	32
• Motor Length(mm)	82
• Shaft Diameter (mm)	6
• Shaft Length (mm)	22
• Weight (gm)	90
• Shipment Weight	0.195 kg

Software Implementation

Technology Required

- Python (as main programming language)
- C (programming for Arduino)
- OpenCV (for Image Processing)
- PyTorch (main Deep Learning framework)
- Tensor flow lite (Deep Learning framework for raspberry pi)

Software Specification

Design of Object Detection Algorithm

For object detection algorithm, after going through literature for the same. We finalize YOLOv3 as a main architecture for the object detection algorithm.

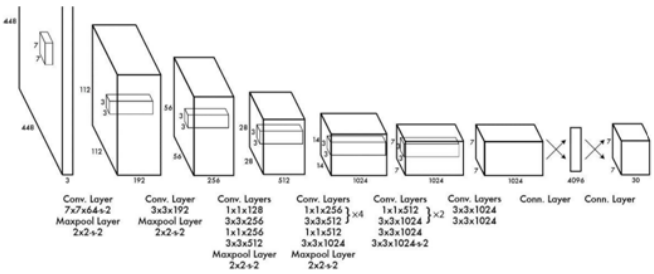

Figure 3.2.1: Architecture of YOLO

Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layer reduces the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection.

For Object detection model, Input size is 416x416 pixels. Image is divided into grid of 13x13, 26x26 and 52x52 i.e. prediction is done at 3 scales. To increase stability of the bounding box, we defined anchor boxes. Each grid has 3 anchor boxes with variable dimensions. The network predicts 5 bounding boxes at each cell in the output feature map. The network predicts 5 coordinates for each bounding box, tx, ty, tw, th, and to. If the cell is offset from the top left corner of the image by (cx ,cy) and the bounding box prior has width and height pw, ph, then the predictions correspond to:

Where, bx = x co-ordinate of the bounding box by = y co-ordinate of the bounding box bw = width of the bounding box

bh = height of the bounding box

Since we constrain the location prediction the parametrization is easier to learn, making the network more stable. Using dimension clusters along with directly predicting the bounding box center location improves YOLO by almost 5% over the version with anchor boxes.

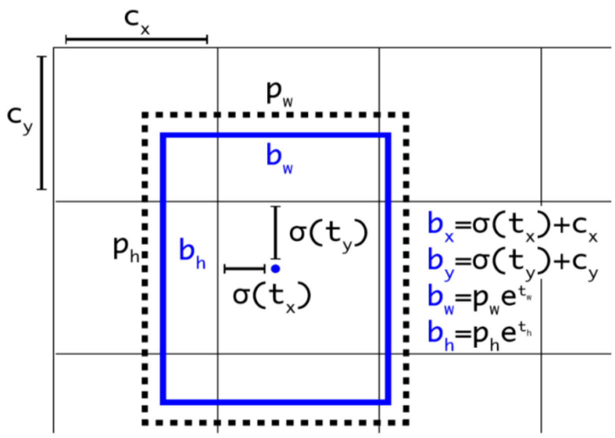

Figure 3.2.2: Anchor box

$$b_x = \sigma(t_x) + c_x$$
$$b_y = \sigma(t_y) + c_y$$
$$b_w = p_w e^{t_w}$$
$$b_h = p_h e^{t_h}$$
$$Pr(\text{object}) * IOU(b, \text{object}) = \sigma(t_o)$$

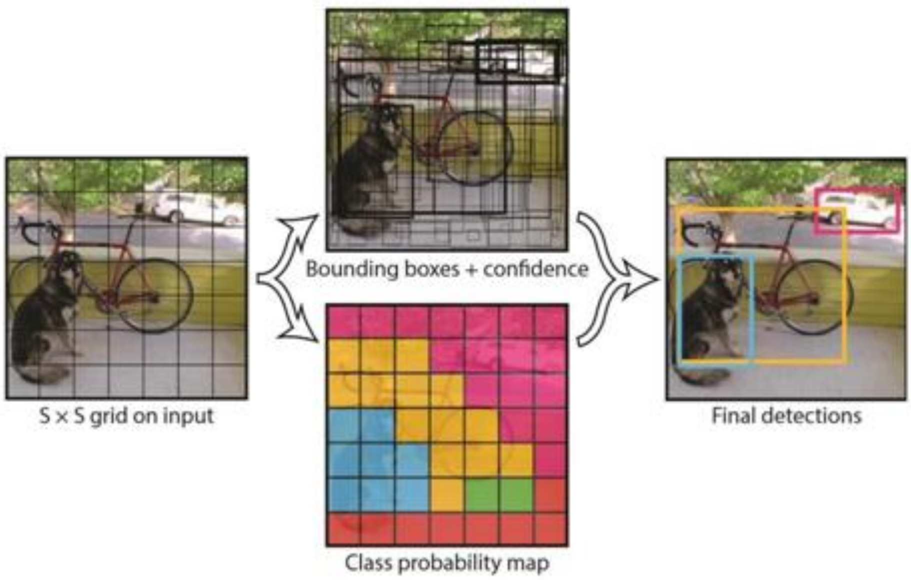

Figure 3.2.3: Sample for object detection

On every image, proposed object detection model predicts 10,576 bounding boxes. Then we sort bounding boxes based on the confidence score. We set threshold value for confidence as 0.5.

Implementation

We implemented state of the art object detection algorithm in PyTorch, A Open Source Deep learning frame work in python. To make our system work real time we used server client configuration, where server is used to do all computationally heavy task and raspberry pi will control all hardware parts. We streamed video from raspberry pi to the server using WebSocket's protocol, which is efficient way to send and receive data in real time. When raspberry is turned on, camera will start to send video frame to the server, server will process this data using state of the art object detection algorithm.

Paper ID: SR221031201117 DOI: 10.21275/SR221031201117

Sends prediction back to the raspberry pi, based on the prediction raspberry pi decides whether to activate rack and pinion mechanism or not. For the object detection algorithm part, algorithm first compresses the image to 480x640x3:3 aspect ratio of the frame captured by the raspberry pi. This compressed image is the passed to convolutional neural network to localize tomato in the image. The object detection algorithm outputs prediction probability, label and bounding box coordinates. Algorithm the perform sorting of the all these based on the prediction probability and only keep prediction with more than 80% prediction probability. Algorithm then use bounding box coordinates to crop ROI (Region of interest) from the given frame. Algorithm the convert image from RGB colour frame to HSV colour frame. Algorithm then perform pixel level calculation of each frame, if the green hsv valued pixels are more in given ROI then it is predicted as unripen tomato and vice versa

Costing

Sr. No	Item	Quantity	Cost per Item	Total Cost
1.	Raspberry Pi	1	3500	3500
2.	Servo motor	1	800	800
3.	Dc Motor	1	500	500
4.	5MP camera module	1	800	800
5	Bearing	4	1000	4000
6	Conveyor Belt	1	3000	3000
7	3D printing	6	200	1200
8	Square Metal Tube	8	200	1600
9	Other Costs		4000	4000
	Total cost			19400

5.Conclusion and Future Scope

In this project we have created an object sorting system using object detection through computer vision and identified the object provided to the system at a specific rate. Our system includes a includes a object been transferred on a conveyer belt one at a time then been detected by object detection algorithm and finally been placed in their respective place using rack and pinion mechanism. By this module we have achieved 87% of detection and localization accuracy of the object of binary classification. Even when the object is in moving condition, this algorithm is capable enough to detect and identify the class of object and if to place in place. The system can be enhanced by increasing the speed of detecting the object. Instead of using rack and pinion more advanced mechanism would help the system to identify the object in less time. The algorithm can be made such that at a time more than one object can be placed in conveyor. Even it can be expected of using less tools and still this system be implemented with less equipment'

References

[1] Journal /Article /Paper - 1. Kuei chen, Chung-Yu Chi, "A real-time system for object detection and location reminding with RGB-D camera," 2014 IEEE International Conference on Consumer Electronics (ICCE) 10, Jan 2014

[2] Book - Ian Goodfellow, Deep Learning – Adaptive Computation and Machine Learning series. MIT Press, 2017.

[3] Website - Jonathan Hui. (2018, Mar 18). *Real Time Object Detection*. Retrieved from http://medium.com