Object Mapping and Room Outlining Robot

Koushik J[1], L Vishwesh S Arya[2], Shamita J[3], Sunidhi V Chate[4], Vanishree Abhay[5]

[1]Student, Dept. of Information Science, Dr Ambedkar Institute of Technology, Karnataka, India.
[2]Student, Dept. of Information Science, Dr Ambedkar Institute of Technology, Karnataka, India.
[3]Student, Dept. of Information Science, Dr Ambedkar Institute of Technology, Karnataka, India.
[4]Student, Dept. of Information Science, Dr Ambedkar Institute of Technology, Karnataka, India.
[5]Professor, Dept. of Information Science, Dr Ambedkar Institute of Technology, Karnataka, India.

---***---

Abstract-*A key area for development in Robotics and Automation is modularity and computational efficiency with respect to limited available computing power. This project makes use of a general-purposeminicomputer such as the Raspberry Pi to perform autonomous navigation around the room while running Object Detection simultaneously in conjunction with three ultrasonic sensors for controlling movements of the robot.*

1. INTRODUCTION

The object Mapping and Room Outlining Robot is a unique robot designed to map the layout and trace the edges of the room while detecting objects that appear in front of it. The robot uses three Ultrasonic sensors(two anterior sensors and one right peripheral sensor) to navigate itself autonomously, governed by the prewritten logic. The two anterior sensors work in unison to evaluate whether there is an object in the robot's path. If an object is detected, the robot automatically captures an image of the object and passes the image through a neural network to classify the image. This classification and identification of the image is done using TensorFlowLite and OpenCV. The robot outputs a layout diagram after it completes its movements which contains the exact layout of the room and the labels of the objects detected in the respective positions of the layout.

2. HARDWARE DESIGN

2.1 List of Hardware

- On Board Computer: Raspberry Pi Model 3B
- Power Supply(24V) and power bank
- Webcam
- Ultrasonic Sensors HC SR04
- Motor Driver L298N
- High watt motors
- Jumper wires
- Breadboad
- 1k Ω Resistors
- Robot chassis

2.2 Hardware and Software Description

a) Raspberry Pi

Raspberry Pi is a single-board computer, which means that the microprocessor, memory, wireless connectivity, and all ports are all on one circuit board. The OS of Raspberry Pi is also Linux based so this is extremely helpful in practical usage and programming.

The Raspberry Pi 3 Model B is the third generation Raspberry Pi. This powerful credit-card sized single board computer can be used for many applications and supersedes the original Raspberry Pi Model B+ and Raspberry Pi 2 Model B. Whilst maintaining the popular board format the Raspberry Pi 3 Model B has powerful processer. Additionally, it adds wireless LAN & Bluetooth connectivity making it the ideal solution for powerful connected designs. [1][2]

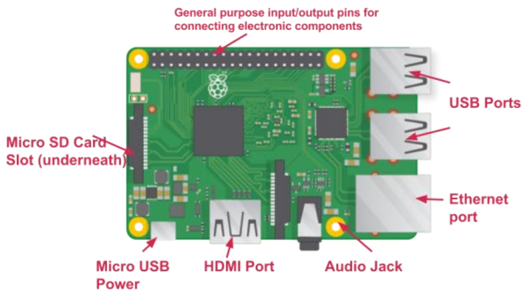

Fig-1 : Components of Raspberry Pi Model 3B

b) Ultrasonic Sensors HC SR04

The HC-SR04 ultrasonic sensor provides 2cm to 400cmof non-contactmeasurement functionality with a ranging accuracy that can reach up to 3mm. Each HC-SR04 module includes an ultrasonic transmitter, a receiver and a control circuit.

There are only four pins on the HC-SR04: VCC (Power), Trig (Trigger), Echo (Receive), and GND (Ground). The working of the ultrasonic sensors is discussed in detail in section IV b. [5]

c) L298N Motor Driver

The L298N is a dual H-Bridge motor driver which allows speed and direction control of two DC motors at the same time. The module can drive DC motors that have voltages between 5 and 35V, with a peak current up to 2A. It can individually drive up to two motors making it ideal for building two-wheel robot platforms. The driver controls the motors' rotating direction by changing the polarity of its input voltage. [8]

d) Raspbian OS

For this project, the OS used is the Raspbian Buster OS. Raspbian is the official Operator Systems released by Raspberry Pi foundation, the Raspbian Buster OS was released as the successor of Raspbian Stretch and is compatible with all the models of Raspberry Pi. Buster is based on the latest version of Debian Linux 10. Raspbian Buster provides a user-friendly platform for code development as well as for building this project. [1]

e) Python

Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. With Raspberry Pi, Python lets us connect our project to the real world. Python syntax is very clean, with an emphasis on readability. Python also supports various key libraries that are crucial for the functionality of this project.

f) RPi GPIO

To use Raspberry Pi GPIO pins in Python, we need to import RPi GPIO package which has classes to control GPIO. This RPi GPIO Python package is already installed on Raspbian OS. A key feature of the Raspberry Pi is the row of GPIO (general-purpose input/output) pins along the top edge of the board. These GPIO pins enable us to connect the ultrasonic sensors and the motor driver to the Raspberry Pi. Adding additional components to the robot is also made easier due to the presence of the GPIO pins and the RPi GPIO library. [1]

g) TensorFlow Lite and OpenCV

TensorFlow Lite is an open-source, product ready, cross-platform deep learning framework that converts a pre-trained model in TensorFlow to a special format that can be optimized for speed or storage. [9]

OpenCV (Open-Source Computer Vision Library) is an open-source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. [7]

TensorFlow Lite has several algorithms to classify images based on pretrained datasets, such as the Google COCO dataset. The CV2 library of OpenCV is used to perform some basic pre-processing on the image which is then classified.

h) Paramiko

Paramiko is a Python implementation of the SSH protocol, providing both client and server functionality. Paramiko itself is a pure Python interface around SSH networking concepts. The SSH protocol made available by Paramiko helps to set up a connection with the PC and the Raspberry Pi on the robot. [10]

i) Turtle

Turtle is a pre-installed Python library that enables users to create pictures and shapes by providing a virtual canvas. The turtle library is used to draw the final layout of the room as recorded by the robot. [11]

2.3 Hardware Components Connections

The chassis is a four-wheel aluminum bodies to which are connected two motors. The motor driver L298N is used to control the operations of the motors and it is housed on the upper flat body of the metal chassis along with the Raspberry Pi, the three ultrasonic sensors and the web camera.

The Raspberry Pi is glued onto the top of the chassis along with the breadboard which helps set up the ultrasonic sensors. The breadboard is used to introduce a resistor to the circuit of connections between the Raspberry Pi and the ultrasonic sensors. The upper anterior sensor is housed slightly elevated using a wooden apparatus. All the connections between the Raspberry Pi, Ultrasonic sensors and the Motor driver is made using jumper wires. The connections of the Motor Driver are illustrated in Fig. 2, and all the Pin connections of the Raspberry Pi are illustrated in Fig. 3.

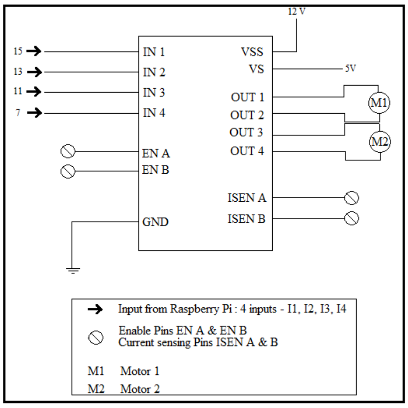

Fig-2 : Pin connections of the Motor Driver(L298N)

```
Raspberry Pi Pins  |  Motor Driver(L298N)
        2          |     5v pin
        6          |     GND pin
        7          |     IN4
       11          |     IN3
       13          |     IN2
       15          |     IN1

Raspberry Pi Pins  | Bottom anterior sensor(HC-SR04)
       18          |     ECHO
       16          |     TRIG

Raspberry Pi Pins  | Top anterior sensor(HC-SR04)
       22          |     ECHO
       37          |     TRIG

Raspberry Pi Pins  | Right peripherl Sensor(HC-SR04)
       29          |     TRIG
       31          |     ECHO
```

Fig-3 : Pin connections of the Raspberry Pi

The power supply for the Raspberry Pi is provided using a powerbank which is also placed on the chassis, an additional power supply for the motors is provided external from the power socket via an adapter.

The two motors operate simultaneously to move forward or backward, the robot turns to the left when the left motor moves forward and the right motor turns in the opposite direction, the right turn works in a similar method, but the rotations opposite to that of the left turn.

To control the motors, the four inputs of the motor driver receive signals from the raspberry pi and the motor driver then sends the appropriate current supply to move the robot in one of the required four directions. The truth table regarding the four input pins and the direction of movement is given in Table a.

Input pin 1	Input pin 2	Input pin 3	Input pin 4	Function
Low	High	High	Low	Robot moves forward
High	Low	Low	High	Robot moves backward
High	Low	High	Low	Robot turns left
Low	High	Low	High	Robot turns right

Table a : Truth table of Motor Movements

3. METHODOLOGY

3.1 Architectural Design

The architectural design of the whole system contains two computers, the first is the client PC which contains the program to set up the connection with the robot, sends commands and receives outputs, and the second is the RaspberryPi on the robot which contains the main program that governs the robot's logic and calls upon other programs as well.

Step 1 : As seen in Fig 4 the first step is to set up the connection with the PC and the robot, it is done via the SSH protocol. The framework acts as a Client-Server model where the PC sends commands and receives some response from the robot.

Step 2 : After the initial connection is setup, the program executed in the PC sends a command to the robot to execute the main program of the robot, i.e the program that accomplishes the main objective of the robot.

Step 3 : Based on the algorithm/logic of the code, the robot begins its movements and records all its movements in the form of turtle commands into a text file within the Raspberry Pi.

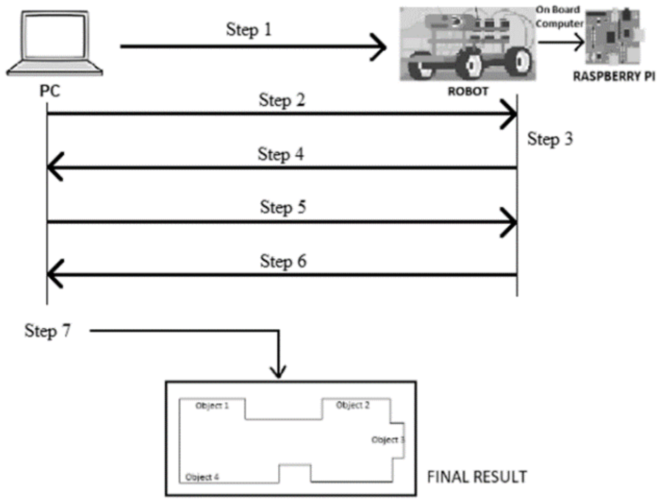

Fig-4 : Design of the system's operational steps

Step 4 : Once the main program of the robot is completed, the program within the computer understands the completion via the standard streams.

Step 5 : Next the program within the computer sends a command to read the contents of the text file stored within the Raspberry Pi.

Step 6 : Using the output stream the computer receives all the contents of the text file.

Step 7 : Each line of the text file contains a record of the robot's movements in the form of turtle commands, hence each line is iteratively executed and the final layout diagram is drawn on the screen, which contains the layout of the room as well as the label of the objects in the room.

3.2 Governing Logic Of The Robot's Movements

The main program that governs the robot's movements and it's autonomous navigation is designed around the three ultrasonic sensors. There are three methods pertaining to each of the sensors, and the main method is based on the front lower anterior sensor. Based on the readings received from the lower anterior sensors, the other two methods or the external Image Classifying method is called.

The Image Classifying task uses the SSD MobileNet V2 Quantized COCO model as the Image classifying model and it is trained using transfer learning from the Google COCO dataset. [4] [6]

In the working of the main program of the robot, if the distance calculated by the lower anterior sensor is below a certain threshold, then the upper anterior sensor and its respective method are called, the distance is calculated using the upper anterior sensor as well. If there is a significant difference between the distances calculated by both the sensors,then the robot concludes that there is an object in front of the robot and not a wall. It takes an image of the object, calls the image classifier and proceeds by going around the object.

If the distance calculated by the lower anterior sensor and the upper anterior sensor are same, then the robot concludes that there is a wall present in front of the object and hence it takes a left turn at that point.

During all these operations, the right peripheral sensor and its corresponding method are called as well, it is called upon in each cycle or each operation in which some movement happens. The right peripheral sensor repeatedly checks whether the distance to the right of the robot is above a certain threshold. If the distance is above a certain threshold, then the robot proceeds by taking the right turn and continues with a new cycle of operation.

4. ROBOT DEVELOPMENT PHASE

4.1 Connecting Raspberry pi, Motor Driver And Motors

The first part in the progress of the robot after setting up the OS on the Raspberry Pi is to set up the connections between the two motors, motor driver(L298N) and the GPIO pins of the Raspeberry Pi. The connections are illustrated in Fig. 2 and Fig. 3. After the connections have been made, the pin connections are declared in the code and the set up is connected using the RPi GPIO library.

The four methods – forward, reverse, left, right are then written in Python for the movements of the robot. The truth table as shown in Table.a illustrates how the movements work.

Based on the logic written and the robot's state, a time delay parameter is passed to the function to execute one of the movement functions for the specified period of time.

4.2 Setting Up The Ultrasonic Sensors

The key for the autonomous navigation of the robot is the readings received from the three ultrasonic sensors(two anterior sensors and a right peripheral sensor). The ultrasonic sensors detect how close an object or a wall is in front of it, it does so by transmitting an ultrasonic sound wave and then receiving it, the time interval between sending the sound wave and receiving it is used to find the distance of the object or wall in front of the sensors. [5] This is illustrated in Fig. 5.

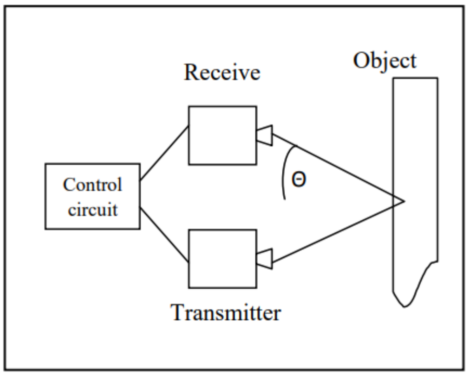

Fig-5 : Working of the Ultrasonic Sensors

The formula used to calculate the distance is the speed, distance and time formula :

Distance = Speed x Time

But since the sound wave travels to and fro the Ultrasonic sensors, the standard above mentioned formula calculates twice the distance between the ultrasonic sensors and the object or the wall, hence we find half the distance :

Distance = (Speed x Time)/2
Taking the speed of sound as 34300cm/s, we get:
Distance = 34300/2 x Time = 17500m/s x Time

From the time taken for the ultrasonic sound wave to be detected by the ultrasonic sensor, the distance of the object or the wall is calculated.

4.3 Controlling The Motors Based On Readings Of Ultrasonic Sensors

Based on the readings of the distance received from the ultrasonic sensors, the relationship between the ultrasonic sensors and the motors are established.

Using the distance calculated by the ultrasonic sensors, the four movement functions can be called based on the requirement; this is explained in the next sub heading.

4.4 Room Outlining Robot Phase

Now that the ultrasonic sensors are set up, as well as the motors, and with the establishment of the relation between the ultrasonic sensors and the motors, the logic for the Room Outlining phase can be developed. For this phase of the robot's development the upper anterior sensor is not used, only the lower anterior sensor and the right peripheral sensors are used.

In this phase of the robot, the robot can autonomously navigate itself from the right hand side of the room and move along the edge of the room, taking the required left turns when a wall is encountered or a right turn when a sufficiently large space to the right of the robot is detected.

The final output of this phase is a layout diagram which shows the exact layout of the room, however this phase does not differentiate between objects or walls in front of the wall.

4.4 Image Capturing AndClassification

The second key phase in the robot's development is the ability to differentiate whether an object is present in front of the robot, or if a wall is present in front of the robot. For this task, the lower anterior sensor and the upper anterior sensor work together. The method that calls for the upper anterior sensor to take the reading is only called if the lower anterior sensor gets a reading below some distance threshold.

Once the upper anterior sensor records the distance, this distance is compared with the reading of the lower anterior sensor. If both the distance are sufficiently similar, then the robot concludes that it has encountered and wall and so turns left and begins a new cycle of operations.

If both the distances are considerably different, then the robot concludes that it has encountered an object and not a wall. Hence, using the webcam is takes a picture of the object, passes the picture to the Image Classification program which classifies the object and records its label in the text file. Then, the robot begins its movements to go around the object and return to the original direction it was pointing towards. [3]

4.5 Incorporating Image Capturing And Classification Into Room Outlining Robot

Once both the key phases – The room outlining phase and the Image classification phase are completed, both the phases are incorporated together into the main program.

Now, all three sensors are in use and can help the robot autonomously navigate along the room while detecting and classifying objects as well as avoiding walls and taking the required changes of movements. The final result of this phase is the layout of the room which contains the actual layout of the room with the labels of the object it detected at the respective positions.

5. RESULT

All the experiments done were done in a room with similar surface with objects placed along the edges of the room. Different tests were done in rooms with different layouts with different objects placed along the edges.

Fig. 6, Fig. 7 and Fig. 8 depict the final result that was displayed on the Client PC's screen, the results are turtle diagrams drawn using the turtle library.

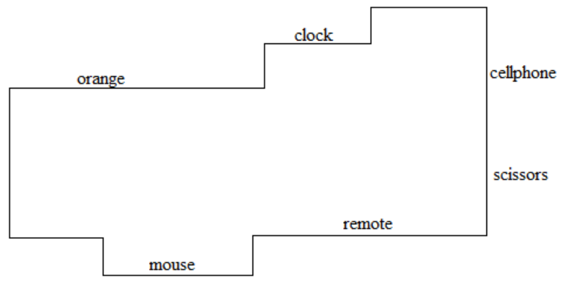

Fig. 6

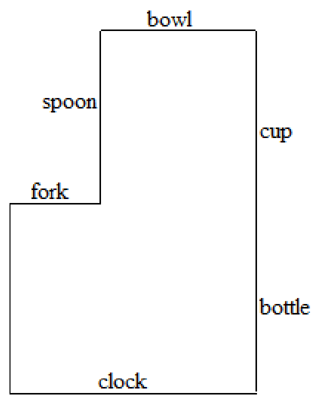

Fig. 7

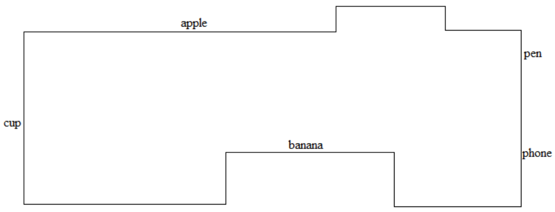

Fig. 8

Fig. 9, Fig. 10 and Fig. 11 are the images of the Robot after the final development was completed. It shows the various components of the robot such as the three Ultrasonic Sensors(HC SR04), Motor Driver(L298N), Raspberry Pi(Model 3B), two motors, power bank, external power supply, web camera and the various connections made using the breadboard, resistors and jumper wires.

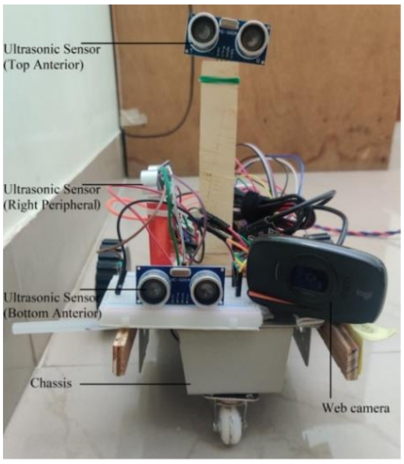

Fig. 9

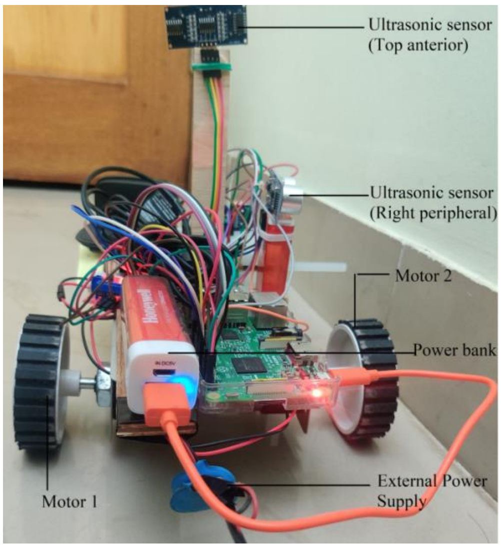

Fig.10

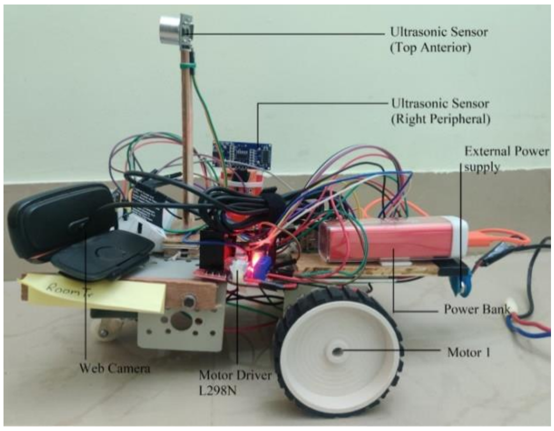

Fig. 11

6. CONCLUSION

In this paper, the method and process of designing and developing a robot capable of autonomously navigating and representing the layout of the room while detecting and classifying objects in its path is presented. It presents a novel approach to design and implement a robot using generic robotic components available, which also enables to further develop and modify the robot as per requirements. The methodology and design mentioned in this paper has been implemented on the illustrated robot.

7. FUTURE WORK

The efficiency of the processes taking place during the operation of the robot as well as the lower cost of developing
it gives way to numerous future developments and improvements.

The project can be used as foundation for further developments in the fields of surveillance and monitoring, human assistance, autonomous working robots, terrestrial drone and mapping robots etc.

This type of a robot opens up many areas of development, research and inventions into automation robots, applying Machine Learning and Deep Learning into portable devices and helps in opening a wide variety of operations and uses. As one key feature of out project is the ease with which new components can be included, the project can be improved by making use of additional ultrasonic sensors to design a better version of the robot which does not adhere to the right side of the room to draw the layout.

8. REFERENCES

[1] Matt Richardson, Shawn Wallace "Getting Started with Raspberry Pi, 2nd Edition" Published by Maker Media, Inc., USA, 2014. Book ISBN: 978-1-457-18612-7

[2] Deepak.S.Kumbhar, H.C. Chaudhari, Shubhangi M.Taur, Shubhangi S.Bhatambrekar"IoT Based Home Security System Using Raspberry Pi-3" INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS January 2019, Volume 6, Issue 1

[3] Sampa Jana , Shubhangi Borkar "Autonomous Object Detection and Tracking using Raspberry Pi "*International Journal of Engineering Science and Computing* 2017 IJESC, Volume 7 Issue No.7

[4] Acep Ansor, Ritzkal Ritzkal, Yuggo Afrianto "Mask Detection Using Framework Tensorflow and Pre-Trained CNN .Model .Based on Raspberry Pi" *Jurnal Mantik*, doi: 10.35335/mantik.Vol4 2020.946 pp1539-1545

[5] Johann Borenstein & Yoram Koren, "Obstacle Avoidance with Ultrasonic Sensors" IEEE journal of Robotics and automation, VOL. 4, NO. 2, APRIL I988, pp. 213-218

[6] Rupa Gurram SweathaSuresh.B. Sneha.B.R. Sushmitha ''Object Tracking Robot on Raspberry Pi using Opencv'' International Journal of Engineering Trends and Technology (IJETT) – Volume 35 Number 4- May 2016 ISSN: 2231-5381

[7] Gary Bradski, Adrian Kaehler, "Learning OpenCV: Computer Vision with the OpenCV Library, O'Reilly Media, Inc.". Copyright.September 2008, 1st Edition, Book ISBN: 978-0-596-51613-0

[8] Dickson Neoh Tze How, Mohd Zafri Baharuddin, Syed Sulaiman Kaja Mohideen, Khairul Salleh Mohamed Sahari, Adzly Anuar "Modular Motor Driver with Torque Control for Gripping Mechanism " 2012 International Symposium on Robotics and Intelligent Sensors

[9] Leigh Johnson "TensorFlow v TensorFlow Lite on Raspberry Pi" available at : https://towardsdatascience.com/3-ways-to-install-tensorflow-2-on-raspberry-pi-fe1fa2da9104

[10] Zadka, Moshe. "DevOps in Python" Book ISBN : 9781484244333

[11] John Hunt, "Advanced Guide to Python 3 Programming", Book ISBN : 9783030259426