

Automation of Logistic Services using Optical Character Recognition (OCR)

M Calvin Isaac

Department of Electronics and
Instrumentation
Ramaiah Institute of Technology

Shiva Prasad R

Department of Electronics and
Instrumentation
Ramaiah Institute of Technology

Prasaanth R

Department of Electronics and
Instrumentation
Ramaiah Institute of Technology

Rahul Holla M

Department of Electronics and
Instrumentation
Ramaiah Institute of Technology

K M Vanitha

Assistant Professor
Department of Electronics and Instrumentation
Ramaiah Institute of Technology

Abstract—Sorting of the packages manually can be a tedious job. This paper aims to automate this process by using the concept of Optical Character Recognition (OCR). In this project we aim to build a system which will automate the process of sorting logistics that is done in courier offices or postal offices. This system consists of a conveyor belt where the parcel is placed and starts to move. When the IR sensor detects a parcel, it would trigger the camera module to capture an image. The camera would capture the image of the postal code on the parcel. This image would be sent for further processing to the Raspberry Pi and Optical Character Recognition would be done on it to identify the numbers, by Tesseract. After the numbers are identified according to the specified constraints as required by the offices, control signals would be given to the servo motors also attached to the side of the conveyor system to sort out the packages as necessary and these packaged would be put in separate containers and would be sent out the respective areas to be delivered.

Keywords— *Optical Character Recognition, Raspberry Pi, Tesseract*

I. INTRODUCTION

Currently in the majority of logistic companies sorting and stamping are done manually which is time consuming and over consumption of human resource, which may lead to errors. Due to large time consumption for the manual processes, there is delay in cargo evacuation which leads to stacking up of undelivered shipments in warehouses.

In India, automation of processes is still in the embryonic stage and non-standardization in the industry due to its fragmentation further slows down the progress. Acceptance and adoption of technological advancements like sorting, stamping, warehouse management system, etc. can resolve the issue. Besides this, there is lack of quality workforce in this sector, and the available skill set needs to be upgraded urgently.

In our solution, we aim at automating this sector, which leads to faster process and increased efficiency. A warehouse automation system can drastically reduce the workforce required to run a facility, with human input required only for a few tasks.

II. ALGORITHMS INCORPORATED

A. Softwares

TensorFlow is a machine learning system that operates at large scale and in heterogeneous environments. TensorFlow uses dataflow graphs to represent computation, shared state, and the operations that mutate that state. It maps the nodes of a dataflow graph across many machines in a cluster, and within a machine across multiple computational devices, including multicore CPUs, general-purpose GPUs, and custom-designed ASICs known as Tensor Processing Units (TPUs). This architecture gives flexibility to the application developer: whereas in previous “parameter server” designs the management of shared state is built into the system, TensorFlow enables developers to experiment with novel optimizations and training algorithms. TensorFlow supports a variety of applications, with a focus on training and inference on deep neural networks. Several Google services use TensorFlow in production, we have released it as an open-source project, and it has become widely used for machine learning research. In this paper, we describe the TensorFlow dataflow model and demonstrate the compelling performance that TensorFlow achieves for several real-world applications.

Keras was developed as a neural network API. It is a library written specifically for Python (I am sure it will be used with R and Javascript eventually.). It works with other libraries and packages such as TensorFlow which makes deep learning easier. Keras was developed to allow for quick experimentation and for fast prototyping.

Keras allows the modules to be constructed and can be integrated with previous code or used as a base for a new project. It runs on the CPU and GPU of any laptop/server/desktop. It supports convolutional and recurrent neural networks.

B. CNN Training Methodologies

The convolutional layer is the core building block of a CNN. The layer's parameters consist of a set of learnable filters, which have a small receptive field, but extend through

the full depth of the input volume. The model is trained with MNIST dataset.^[13]

Max pooling is a sample-based discretization process. The objective is to down-sample an input representation (image, hidden-layer output matrix, etc.), reducing its dimensionality and allowing for assumptions to be made about features contained in the sub-regions binned.

A dense layer is just a regular layer of neurons in a neural network. Each neuron receives input from all the neurons in the previous layer, thus densely connected. The layer has a weight matrix W , a bias vector b , and the activations of previous layer a . The following is the docstring of class Dense from the Keras documentation:

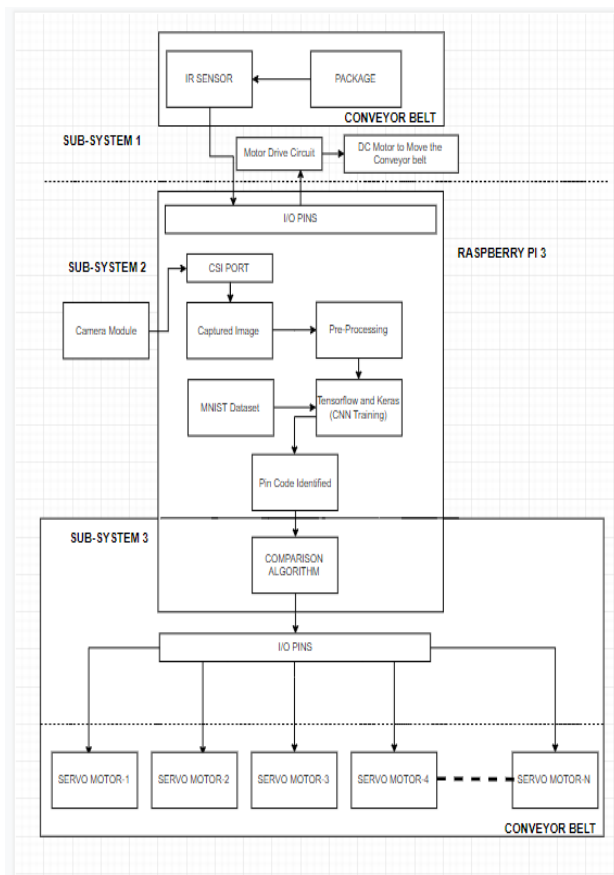
output = activation(dot(input, kernel) + bias)

where activation is the element-wise activation function passed as the activation argument, kernel is a weights matrix created by the layer, and bias is a bias vector created by the layer.

Dropout is a technique used to tackle Overfitting. The Dropout method in keras.layers module takes in a float between 0 and 1, which is the fraction of the neurons to drop.

III. SUBSYSTEMS

This chapter consists of all the subsystems involved in the product. It also contains information about how each of them are implemented.



A. Subsystem 1: Detection of package on the conveyor system

Subsystem 1 consists of a Conveyor belt where the packages are placed, and for the detection and presence of package we make use of infrared sensor which is interfaced to raspberry pi I/O.

The conveyor belt is driven by a DC motor which is controlled by raspberry pi with the help of the motor driver circuit.

Once the packages are placed on the conveyor belt, Infrared sensor is used to detect the presence of the package and sends a signal to raspberry pi for the camera to capture an image.

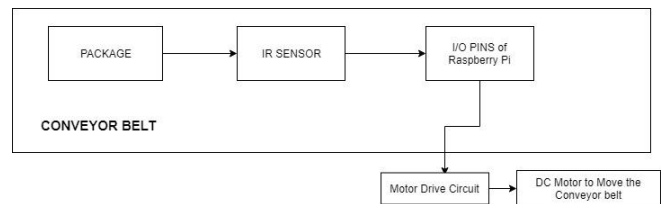


Fig. 2 Sub-System1 Block Diagram

B. Subsystem 2: Image processing and Optical Character Recognition

The subsystem 2 mainly comprises of the image processing^[3] and pin code identification. When the camera module, connected to CSI port, captures the image, the image will be pre-processed. The image will be first re-scaled, mostly enlarging the image will work as to recognize small numbers.

Once the image is rescaled, the process of de-noising the image, to remove the unwanted grains or noise in the image, will be done. De-noising will be done using the Gaussian filter. The Gaussian filter convolutes the image using Gaussian kernel. Here, the dimensions of the kernel and standard deviations in both directions can be determined independently. Gaussian blurring is very useful for removing Gaussian noise from the image.

Next, the image is binarized. Adaptive Gaussian thresholding is used as it produces the best results. Rather than setting a one global threshold value, we let the algorithm calculate the threshold for small regions of the image. Thus, we end up having various threshold values for different regions of the image.

Then, the character recognition is done by Tesseract^[9], which is a highly popular OCR^{[4][8][10][11][12]} engine. The engine will be trained with numerous fonts, styles of handwritten scripts. With the help of these data sets, the characters in the pin would be recognized by Tesseract. The resultant number can

be stored in various file formats such as .txt, .pdf, etc. we would like to prefer saving it in .txt file format.

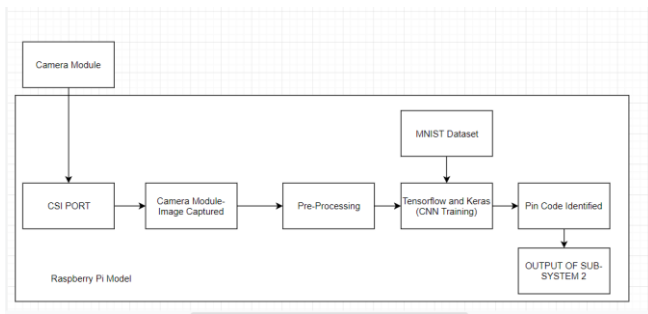


Fig. 3 Sub-System2 Block Diagram

C. Subsystem 3: Sorting^{[2][5]}

The output of the second subsystem which is the image captured and OCR being performed on the pin-code obtained from the image would be compared with the pre-defined pin code ranges using the comparison algorithms which would be implemented on the Raspberry Pi microprocessor by using Python coding. The pre-defined postal codes and ranges would be as required by the postal or courier office where the device would be used.

This would accordingly be given as control inputs to the input-output pins of the Raspberry Pi and based on these inputs the servo motors would be controlled accordingly by the signals given to the motor drivers which would drive the motors.

These motors driven would accordingly sort the packages which need to be sorted which were put on the conveyor belt and would be put into different boxes for delivery.

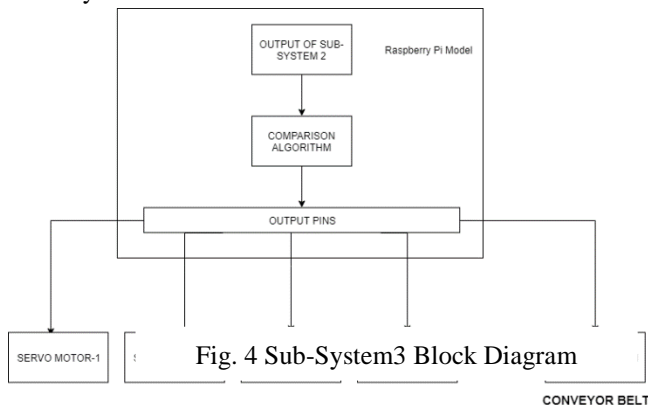


Fig. 4 Sub-System 3 Block Diagram

IV. COMPONENTS AND MODULES USED

A. Raspberry Pi (Raspberry Pi Model B RASP-PI-3 Motherboard)

This motherboard from Raspberry Pi^[1] features 4 USB ports, an HDMI port and a 3.5mm audio jack, among other features. It is built on the latest Broadcom 2837 ARMv8 64bit processor. The motherboard also features a micro SD card slot. The micro SD card slot follows a push-pull technique. Install this motherboard in your computer for a fast and responsive system.

The Raspberry Pi hardware has evolved through several versions that feature variations in memory capacity and peripheral-device support.

The Ethernet adapter is internally connected to an additional USB port. In Model A, A+, and the Pi Zero, the USB port is connected directly to the system on a chip (SoC). On the Pi 1 Model B+ and later models the USB/Ethernet chip contains a five-port USB hub, of which four ports are available, while the Pi 1 Model B only provides two. On the Pi Zero, the USB port is also connected directly to the SoC, but it uses a micro USB (OTG) port.



Fig. 5 Raspberry Pi 3

Raspberry Pi 3 Specifications

SoC: Broadcom BCM2837
 CPU: 4× ARM Cortex-A53, 1.2GHz
 GPU: Broadcom VideoCore IV
 RAM: 1GB LPDDR2 (900 MHz)
 Networking: 10/100 Ethernet, 2.4GHz 802.11n wireless
 Bluetooth: Bluetooth 4.1 Classic, Bluetooth Low Energy
 Storage: microSD
 GPIO: 40-pin header, populated
 Ports: HDMI, 3.5mm analogue audio-video jack, 4× USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)

Fig. 6 Raspberry Pi 3 Specifications

B. Raspberry Pi Camera^[5]

The camera consists of a small (25mm by 20mm by 9mm) circuit board, which connects to the Raspberry Pi's Camera Serial Interface (CSI)^[6] bus connector via a flexible ribbon cable. The camera's image sensor has a native resolution of five megapixels and has a fixed focus lens. The software for the camera supports full resolution still images up to 2592x1944 and video resolutions of 1080p30, 720p60 and 640x480p60/90. Installation involves connecting the ribbon cable to the CSI connector on the Raspberry Pi board.



Fig. 7 Raspberry Pi Camera

C. Infra Red Sensor Module

This Infrared obstacle/object detection sensor is super easy to use. It comes with on board potentiometer to adjust the sensitivity. The output is digital signal so it is easy to interface with any microcontroller. Obstacle detection via infrared reflection, it is non-contact detection. It is based on light reflection; the detection does vary with different surface.

It comes in a pair of Infrared emitter and receiver at the front of module, whenever there is object blocking the infrared source, it reflects the infrared and the receiver get it and the signal go through a comparator circuit on board. And depending on the threshold that being adjusted, it will output logic LOW at output pin and the green LED will light up to indicate the detection. Turning the on-board potentiometer clock wise will increase the sensitivity and further increase the detection range. Compatible with 5V or 3.3V power input.

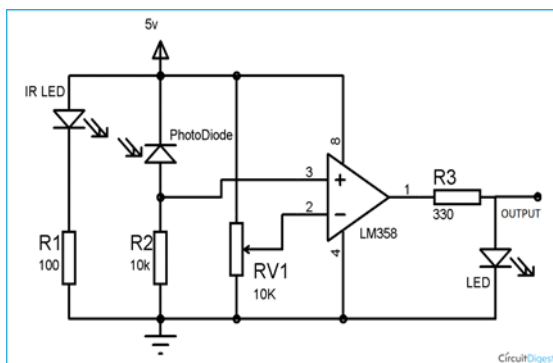
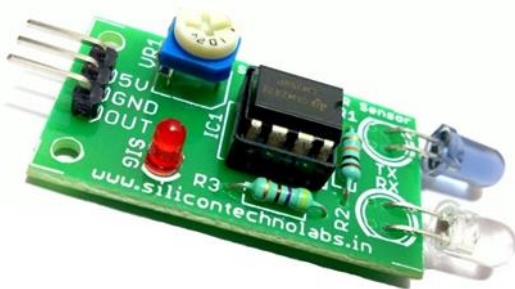


Fig. 8 a) IR Sensor Module b) IR Sensor Module Internal Circuit

D. Servo Motors

A servomotor is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration. It consists of a suitable motor coupled to the raspberry pi module in order for it to act as a sorting arm.



Fig. 9 Servo Motor

E. DC Motor

A DC motor is any of a class of rotary electrical machines that converts direct current electrical energy into mechanical energy. The most common types rely on the forces produced by magnetic fields. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic, to periodically change the direction of current flow in part of the motor. The DC motor is interfaced with the Raspberry PI module to move the conveyor system



Fig. 10 DC Motor

V. EXPERIMENTAL RESULTS

The image capture on raspberry pi camera is processed by resizing, de-noising, thresholding and binarization using OpenCV. The processing was performed and the output of processed image will be as shown below:

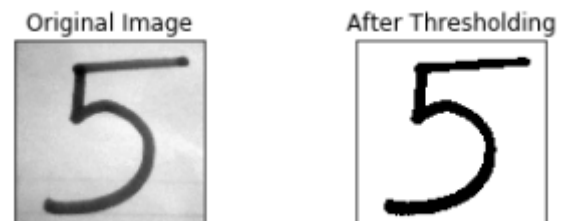


Fig. 11 a)Original Image b) Pre-Processed Image

Once the image is preprocessed, tesseract is applied on the image and the output is stored in a file which is read and sorting is done.

ACKNOWLEDGMENT

We take this golden opportunity to sincerely thank our head, Dr. M. Jyothirmayi, HOD, Electronics and Instrumentation, RIT and project coordinator, Dr. H.S.Niranjana Murthy, Associate Professor, Electronics and Instrumentation, RIT for letting us proceed with the idea of practically implementing Automation of Logistic Services using Optical Character Recognition.

We wish to express our heartfelt gratitude to the project panel for their constant support and benevolence.

We would also like to thank our project guide Ms K M Vanitha, Asst. Professor, Electronics and Instrumentation, RIT for her endless effort in guiding us through different phases of the project with ease.

REFERENCES

- [1] Mr. Suhas M Patil, Sumeet Sanjay Walam, Siddhi Prakash Teli, Bilal Sikandar Thakur, Ravindra Ramchandra Nevarekar, "Object Detection and separation Using Raspberry PI", Proceedings on second international conference on inventive communication and computational Technologies, 2018.
- [2] Md. Mostafizur Rahman Komol, Amit Kumer Podder, "Design and Construction of Product Separating Conveyor based on Colour", 3rd International Conference on Electrical Information and Communication Technology, 2017.
- [3] Mr. S. S. Kulkarni, Mr. A. D. Harale, Mr. A. V. Thakur, "Image Processing for Driver's Safety and Vehicle Control using Raspberry Pi and Webcam", IEEE International Conference on Power Control, signals and Instrumentation Engineering, 2017
- [4] Shalini Sonth, Jagadish S Kallimani, "OCR Based Facilitator for the Visually Challenged", International Conference on Electrical, Electronics, Communications, computer and optimization Techniques, 2017.
- [5] Rushali Sahu, Manoj Kumar Swain, Kunja Bihari Swain, Rakesh Kumar Patnaik, "Productive and Economical Sorting of Objects by Low Resolution Camera", IEEE International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials, 2017.
- [6] Tasnim Sorwar, Sabbir Bin Azad, Sayed Rizban Hussain, Azfar Isa Mahmood, "Real-time Vehicle monitoring for traffic surveillance and adaptive change detection using Raspberry Pi Camera Module", IEEE Region 10 Humanitarian Technology Conference, 2017.
- [7] Mr. Rajesh M, Ms. Bindhu K. Rajan, "Text recognition and face detection aid for visually impaired person using raspberry pi", International Conference on Circuits Power and Computing Technologies, 2017.
- [8] Rithika.H, B. Nithya santhoshi, "Image Text To Speech Conversion in the Desired Language by Translating with Raspberry Pi", International Journal of Computer Science Trends and Technologies, 2017.
- [9] Qi Li, Weihua An, Anmi Zhou, Lehui Ma, "Recognition of Offline Handwritten Chinese Characters Using the Tesseract Open Source OCR Engine" International Conference on Intelligent Human Machine systems and Cybernetics, 2016.
- [10] Chamila Liyanage, Thilini Nadungodage, Ruwan Weerasinghe, "Developing a commercial grade Tamil OCR for recognizing font and size independent text" International Journal of Computer Applications, 2015.
- [11] Sandip Rakshit, Amitava Kundu, Mrinmoy Maity, Subhajit Mandal, Satwika Sarkar, Subhadip Basu, "Recognition of Handwritten Roman Numerals using Tesseract Open Source OCR engine", International Conference on Advances in Computer Vision and Information Technology, 2009.
- [12] Sirvan Khalighi, Parisa Tirdad, Hamid R. Rabiee, Mehdi Parviz, "A Novel OCR System for Calculating Handwritten Persian Arithmetic Expressions", International Conference on Machine Learning and Applications, 2009.
- [13] Modified National Institute of Standards and Technology (MNIST) database- <http://yann.lecun.com/exdb/mnist/>