WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Implementation of Embedded Systems for Object Detection

M. Narendra Kumar[1], V. Leela[2], Y. Arun Venkat [3], P. Vijay Prakash [4], Y. Sai Charan[5], P. Madhu[6]

Senior assistant professor, SIR CR REDDY College of Engineering [1]
UG scholars, SIR CR REDDY College of Engineering [2, 3, 4, 5, 6]

ABSTRACT—

The aim of this project is to implement an embedded system for object detection using a raspberry pi and a camera module. The system utilizes python libraries such as tensor flow, Open CV and a Convolutional Neural Network (CNN) to train the raspberry pi 3 for object detection. Object detection is the process to identify the certain object by capturing the image of an object using camera or giving an image to the data base. The data set used for this project includes electronic components like resistors, capacitors, raspberry pi, LCD display and others. The output of the system is displayed on a screen and a speaker, which can be used to teach deaf and blind people about the objects in their surroundings. The system is self-contained and can be portable in any place with minimal charges.

Keywords— (Object detection; Convolutional Neural Networks; Open CV; Raspberry Pi 3; Tensor Flow).

1. INTRODUCTION

Object detection is a popular computer vision technique that involves identifying objects of interest within an image or video stream. With the help of hardware like the Raspberry Pi and camera modules, object detection can now be easily implemented in embedded systems.

Using a Raspberry Pi and a camera module, object detection can be achieved through a combination of hardware and software components. The camera module captures images or video frames that are then processed by the Raspberry Pi's CPU and GPU. The software component typically involves running a deep learning model that has been trained to detect specific objects in images or videos. The classification and recognition of variety of materials that are present in our surroundings become an important visual competition have been focused by computer vision systems in the recent years.

There are various deep learning frameworks available for implementing object detection on a Raspberry Pi, such as TensorFlow, and OpenCV [5]. These frameworks enable the use of pre-trained models or allow the creation of custom models for specific use cases.

In this paper, we propose an object detection system, a very useful application to make this process easier, which gives better performance than the traditional methods of classifying objects and improves efficiency.

The rest of the paper is as follows: Section 2 summarizes the related work. Section 3 summarizes concepts discussed in this paper. Section 4 summarizes our methodology in utilizing DL and convolutional neural networks (CNN) in the object detection problem. Section 5 presents the results. Section 6 presents conclusion. Section 7 presents references.

2. RELATED WORK

The key efforts on the object detection problem and embedded system implementations that have been published are summarised in this section.

Vandana Mansur & K. Sambavi [1] “highway drivers drowsiness detection system model with raspberry pi and CNN technique”. The model is trained using Keras, an open-source tool, with TensorFlow as the backend. A buzzer, a Raspberry Pi model 4 camera module, and other components make up the hardware setup. Using the buzzer output, the video recording, labelling of open or closed eyes, and the score are obtained as a real-time status to warn the driver when they are falling asleep. The design phase for the Raspberry hardware is deployed concurrently with the training phase. The full AI training software is provided along with the standalone hardware module. The design platform's real-time picture acquisition predicts driver drowsiness with an accuracy of 95% to 96%.

V. M. Kumari & B. Bhushan’s [2] “Real-time Object Detection on Raspberry Pi for Smart Surveillance Systems” This study describes a Raspberry Pi-based real-time object detection system for intelligent surveillance systems. The authors suggest an improved method for real-time object identification on the Raspberry Pi platform. They show off the system's capacity to find things in surveillance footage and assess its performance in terms of speed and accuracy.

Vidya Kamath &A. Renuka.[3] “Performance Analysis of the Pretrained Efficient Det for Real-time Object Detection on Raspberry Pi”. In this study, a raspberry Pi runs Efficient Det to handle the object detection task. They carry out object identification for a particular task using pre-trained tensor flow

models, and they assess the effectiveness of their hardware. In comparison to other common object identification models, Efficient Det is a cutting-edge object detection model that delivers excellent accuracy with fewer parameters. With limited resource devices like the Raspberry Pi, real-time object detection using pre-trained Efficient Det models is possible.

3. THEORY

This section summarizes the main concepts that were implemented in this paper.

3.1 Tensor flow

Both deep learning and conventional machine learning problems can be handled by TensorFlow. It offers an adaptable architecture that makes it simple for programmers to create and implement machine learning models on a variety of hardware, including CPUs, GPUs, and mobile devices.

Predictive analytics, natural language processing, image and audio recognition, and other machine learning applications all frequently employ TensorFlow. It is frequently employed in academia and research, where it is frequently used to create and test fresh machine learning models [5].

3.2 Convolutional Neural Network

A specific type of neural network known as a CNN is utilised in a variety of industries and has proven to be highly successful in resolving a number of problems, including face and object recognition and OCR. The convolution mechanism used by CNN's layers gave the system its name. Convolution is the process of multiplying each pixel of the image by each value in the kernel, which is a different matrix. The formation of multiple images from the original image thanks to the convolution operation improves the features extracted from the original image and boosts the efficiency of the classification procedure. With CNN, each layer is in charge of a specific duty [6].

3.3 Open CV

Just a few of the several programming languages that OpenCV is compatible with include C++, Python, and Java. It provides a wide range of functions and modules to do various image processing tasks, such as image filtering, feature detection, object recognition, and tracking. Moreover, OpenCV has modules for deep learning, machine learning, and computational photography. Many images and video file types, including well-known ones like JPEG, PNG, and MPEG, are supported.

Many applications, such as robotics, autonomous cars, augmented reality, and surveillance systems, make extensive use of OpenCV. It is also widely employed in academic settings and research to create and evaluate fresh computer vision algorithms [5].

4. METHODOLOGY

The system's design consists of three main components: image acquisition, object detection, and output display. For image acquisition, we used a Raspberry Pi camera module to capture images of the electronic components and we prepare a data set. In that data set the components are divided into folders, in each folder there are nearly 100-150 images which we can see in fig 1. The captured images were then pre-processed using OpenCV [5] to improve their quality and reduce noise.

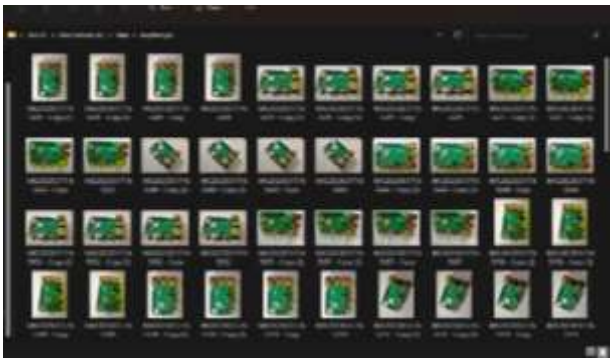


Fig 1: data set

For object detection, we trained the Raspberry Pi using Python, including libraries such as TensorFlow and CNN. We trained the system using a data set of electronic components, including resistors, capacitors, LCD displays, and other similar objects. We used a Convolutional Neural Network (CNN) to train the system to recognize the objects in the data set.

Finally, for the output display, we used a screen and a speaker to display the results of the object detection. When an object is detected, the system announces the name of the object through the speaker and displays the name of the object on the screen.

Fig Block diagram

Explanation of block diagram

The block diagram explains about the overview of the project. Here speaker, LCD and web cam is connected to the raspberry pi. Power bank provides supply to the raspberry pi. The input image resistor is captured using web cam and that image is transferred to the raspberry pi, it compares the input image with the reference data set. Based on that comparison if the image is matched with the reference data set the output will be displayed on an LCD display and the audio will speak out through speaker.

Raspberrypi 3B+: The Raspberry Pi 3B+ is a single-board computer that provides the processing power and connectivity for our embedded system. It has a 1.4GHz 64-bit quad-core ARM Cortex-A53 CPU, 1GB of RAM, and a Broadcom Video Core IV GPU. It also has built-in Bluetooth and Wi-Fi connectivity, making it easy to connect to other devices.

LCD: The object detecting system's output is displayed on the LCD (Liquid Crystal Display). It is a flat-panel display that shows images using the ability of liquid crystals to modulate light. This project makes use of a common 16x2 character LCD. LCD The Raspberry Pi's GPIO (General Purpose Input/Output) pins are used to connect it to the computer.

Webcam: The webcam is used to capture images of the electronic components that are being detected. It is connected to the Raspberry Pi through a USB port and is used in conjunction with the OpenCV library to pre-process the images.

Speaker: The speaker is used to announce the name of the electronic component detected by the system. It converts electrical energy into sound waves, which can be heard by the user.

Power bank: The power bank is used to provide power to the Raspberry Pi. It is a portable battery pack that can be used to charge electronic devices such as smartphones, tablets, and laptops. The power bank used in this project provides a 5V, 2A output, which is sufficient to power the Raspberry Pi 3B+.

Explanation of project code

Initially we trained the data set using python idle were we use tensor flow and open cv [5]. In this training we need to load the images into a training folder. When we run that code, it displays like this

Fig 2: IDLE shell

In figure 2 we can clearly see that the training is going on. The system reads the data in the data set in alphabetical order and it takes around 20% of the data as a validation set. This validation set increases the accuracy of the system. After completion of training, we can detect our objects.

After classification, the category is displayed on an LCD screen using the Raspberry Pi GPIO pins. Additionally, the text-to-speech module is used to speak the category out loud.

Fig 3: LCD display showing welcome

The script initializes the GPIO pins for the LCD screen and defines functions for initializing the LCD screen, sending bytes to the screen, toggling the enable pin, and displaying a string on the screen. It then initializes the LCD screen and displays a "WELCOME" message that is shown in the figure 3.

Fig 4: input tab

In figure 4 an input tab is opened when the code is successfully executed. Then we need to choose an image in our system to identify. It then uses a pre-trained machine learning model to classify the images into different categories.

Fig 5: choosing the image

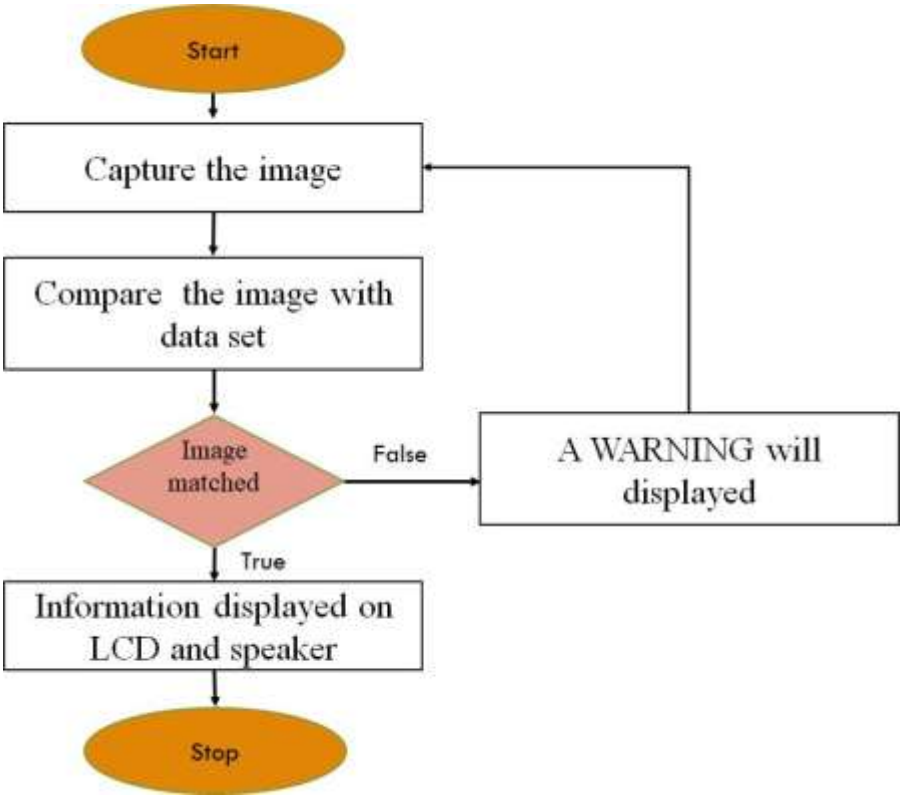
In figure 5 we choose a image from our system which we need to identify after choosing the image press the sumbit button.

The script uses OpenCV [5] to capture video from the webcam and save an image. It then loads the pre-trained model and uses it to classify the image.

Fig 6: LCD showing object name

In figure 6, it displays the category on the LCD screen and speaks it out loud using the text-to-speech module.

FLOWCHART:



5. RESULT

Before we can develop an object recognition system, we must first prepare the electronic components dataset for training. To construct a collection of electronic components, we employed the object identification method, which can identify things in live camera footage. The collected photos are saved into a dataset folder to be used in feature extraction and training methods.

We prepare our own data set by capturing the images of different electronic components like Arduino, raspberry pi, resistors, led, etc. each components photographs are placed separately in each folder. Each folder contains around 100-150 images.

We consider that data set as a reference data set and we start training our system during that training we consider 20% of the data set as a validation set there, we acquire a training accuracy of 71.9% which we can see in the figure 7.

Fig 7: training accuracy is 71.9%

If we increase the test ratio percentage the accuracy is decreasing which is shown in the table 1.

Percentage of test data	20%	40%	50%
Number of test data	476	762	953
accuracy	71.9%	70.5%	69.4%

Table 1: Effect of changing the test ratio

In figure 7 we can clearly see that there are 56 training epochs where the training epoch increases training accuracy also increases. As the number of epochs increase, the validation accuracy increases as shown in Figure 8 below:

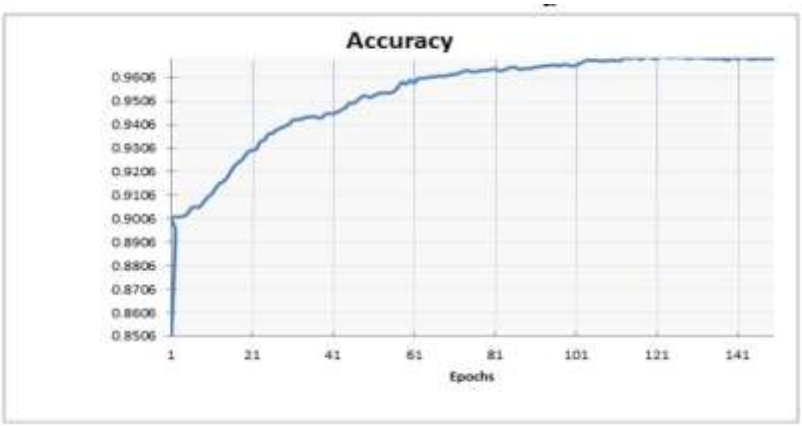


Fig. 8. Validation Accuracy

We tested the system using a data set of electronic components, and the results were promising. The system was able to detect the different components accurately, and the output was displayed on the screen and announced through the speaker.

6. CONCLUSION

In conclusion, our object recognition system has demonstrated encouraging results after being trained on our reference dataset of electronic components. Our system has successfully identified several electronic components, including Arduino, Raspberry Pi, resistors, LEDs, and more, with a training accuracy of 71.9% and in real-time camera footage.

The output displayed on the LED screen and stated through the speaker demonstrates the practical applicability of our system, and the usage of a validation set during training has helped ensure the accuracy of our model. For real-world electronic component recognition jobs, our system's accuracy and resilience can be improved with additional adjustments and optimisations.

The potential of applying object recognition techniques to detect electronic components is generally highlighted by our results. We looked at how altering the test data ratio affected accuracy. The best accuracy is 71.9%, which we obtained using 20% ratio as test data. First, we utilised 20% of the data as test data; later, we increased it to 40% and then to 50%. This system can be used to teach deaf and blind people about the different components, and it can also be used in various fields such as robotics and automation.

7. References

[1] Vandana Mansur & K. Sambavi “highway drivers drowsiness detection system model with raspberry pi and CNN technique”2021 in IEEE.

[2] V. M. Kumari & B. Bhushan's "Real-time Object Detection on Raspberry Pi for Smart Surveillance Systems",2020. In IRJET volume 9.

[3] Vidya Kamath &A. Renuka. "Performance Analysis of the Pretrained Efficient Det for Real-time Object Detection on Raspberry Pi".

[4] M. Nielsen. (2022, June 21). Neural Networks and Deep Learning and it is received from Neural networks and deep learning

[5] Tensor flow library is taken from https://www.tensorflow.org/versions/r2.0/api_docs/python/tf.

[6] Thomas, Elsken; Jan Hendrik, Metzen; Frank, Hutter (2021-11-13). "Simple and Efficient Architecture Search for Convolutional Neural Networks"

[7] V. M. Kumari & B. Bhushan's "Real-time Object Detection on Raspberry Pi for Smart Surveillance Systems",2020. In IRJET volume 9.

[8] https://www.researchgate.net/publication/322514596_Embedded_system_implementation_for_material_recognition_using_deep_learning

[9] (2017, June 21). Keras: The Python Deep Learning Library. Retrieved from https://keras.io

[10]https://www.researchgate.net/publication/337082426_Object_Recognition_Using_Deep_Learning

[11] B. Caputo, E. Hayman, and P. Mallikarjuna. "Class-specific material categorisation." Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on. Vol. 2. IEEE, 2005.

[12] S. Bell., P. Upchurch., N. Snavely., & K. Bala. (2015). Material recognition in the wild with the materials in context database. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3479-3487).

[13] V. Pereira, V. A. Fernandes and J. Sequeira, "Low cost object sorting robotic arm using Raspberry Pi", 2014 IEEE Global Humanitarian Technology Conference - South Asia Satellite (GHTC-SAS), Trivandrum, 2014, pp. 1-6.

[14] S. Sheela, K. R. Shivaram, S. Meghashree, L. Monica., A. Prathima and S. M. Kumar, "Low-Cost Automation for Sorting of Objects on Conveyor Belt", International Journal of Innovative Research in Science, Engineering and Technology. 10, May 2016.