Design of Low-Cost Object Identification Module for Culinary Applications

B Pranav Vijay Chakilam, Revanth, Vyshnavi Muppirala, A Anilet Bala* Vivek Maik
Department of Electronics and Communication Engineering, College of Engineering and Technology, SRM Institute of Science and Technology, Chennai, Tamilnadu, India
Email: aniletba@srmist.edu.in*

Abstract. Using up one's fridge's contents often requires touch creativity, which is typically hard to come by in a very busy world. This aim is to style a model that may scan the things in one's fridge and recommend recipes that supported what one has got, even taking one's preferences and dietary restrictions under consideration. Computers are commencing to process globe objects without the requirement for codes or human intervention in their language. This project aims to create a module that uses object recognition to detect vegetables and fruits and display recipes that include those foods. This concept uses a camera that is installed in a refrigerator to show it into a sensible one. The system identifies objects that are placed inside using complex object recognition algorithms using the camera. If a vegetable or fruit is placed inside, the system will identify it and displays its name on the screen. It creates an indication of the chances of object recognition. The platform enables users to put any number of various vegetables into the refrigerator. The system then checks a list of recipes that contain the ingredients inside the refrigerator and filters those that feature the vegetables available. Together with this, we also aim to watch the refrigerator's contents and find the freshness and age using various sensors and provide inventory and warnings when they are on the verge of completion.

Keywords: Object Identification; Machine learning; Deep Learning; Python; Image processing; prediction accuracy

1. Introduction
When we are shown a picture, the objects that are present inside the image are immediately recognized. Simultaneously, it requires heaps of time and preparatory training data for enabling machines with the same recognition capabilities [1], [2]. With recent advances in Deep Learning (DL), hardware in terms of processing power, and Big Data's advent, the Computer Vision domain has become significantly simpler and increasingly instinctive. Object Detection and its applications have become pervasive across a gamut of fields. From assisting self-driving cars to drive in the presence of traffic to spotting unruly incidents and rough conduct in packed places, analyzing and constructing scouting reports to help sports teams to make sure that optimum internal control of manufacturing parts is guaranteed, its applications have become ubiquitous. Object Detection can be realized in the most efficacious manner with the help of DL [3].

Pair of methodologies by which deep learning can surpass existing object detection techniques are listed next. Rather than acquiring patches from the underlying image, the image is skilled in a neural network to scale back the size [4]. The neural network is accustomed to suggest selective patches of the image [5], [6]. Now, rather than preparing multiple neural networks to tackle the different aspects
of object detection, a single, deep neural network will be attempted to understand and provide a solution to the entire problem. The upside of taking this approach is that each of the discrete modules of a neural network will optimize the contrary parts of the identical neural network [7], facilitating the combined training of the entire deep model [8]. Training a model from scratch is often cumbersome and consumes a lot of time and data to ameliorate a situation like this. Transfer learning can be employed to complete the training process with fewer data points and lesser time. In transfer learning, the understanding, patterns, and learning attained from one task are utilized and applied in another task. The idea is based on the logic the two similar tasks that appear completely unrelated might have identical underlying patterns based on which the model detects the target, so we make use of this training in our task directly, bypassing the repetition of the extensive training which was initially performed. The improvement in accuracy while decreasing training time and data by employing transfer learning has been repeatedly, widely, and reliably demonstrated now.

2. Literature Survey

a) Object Identification techniques
The method referred to understanding and analyzing object detection frameworks dependent on DL that tackle various aspects of the problem, like occlusion, clutter, and low-resolution R-CNN [9]. The method was used to assess the impact of deep learning, especially in object detection utilizing Faster R-CNN [10]. It was surmised that state-of-the-art object detection networks relied on region proposal algorithms to estimate the objects’ positions. The detection networks’ running time has been substantially decreased in advanced networks, like R-CNN and SPPNet, revealing insights about the computation power consumed by region proposal. An RPN can be described as follows, a fully convolutional network that estimates the boundaries of the objects and the scores reflecting the certainty that an object is contained, at every position, simultaneously, the training process of the RPN IS end-to-end in nature to realize accurate region proposals. The fast R-CNN further utilizes these for detection. Earlier work regarding object detection frame as a classification problem, thereby utilizing classifiers to perform detection. Using YOLO, we look at object detection as a regression problem, where the target variables are bounding boxes isolated in space and the corresponding class probabilities. A single neural achieves this directly from full images by running it through just once. As the entire pipeline, just one network, end-to-end, direct, optimization of the performance is made possible. It performs other detection methods like R-CNN and DPM when generalizing natural images to other domains like artwork. Unlike the RCNN models, which use RPN, which requires multiple passes on the image, the BBOX predictor divides the image into grids and generates the boxes in one go, thus the name, ‘You only look once.’ YOLO works with images only once it is substantially faster than all models of RCNN.

b) Sensor Array and Integration Methods
In [11] suggest that a module can be attached to a traditional refrigerator to equip it with the functionalities of a smart refrigerator. In [12] experiments with the idea of using RFID and barcode scanners for detecting and adding food items to the inventory, proposes a system that can notify and alert users about the status of the contents inside their refrigerator using a Wi-Fi module. These papers were used to gain insights into the design of an integrated sensor array that can monitor and notify the status of a refrigerator’s contents.

3. Statement of Problem
There has been massive research in deep learning in recent years, along with efforts to use it in culinary applications, but there has been no practical implementation in refrigerators. A low-cost module is required that can convert existing normal refrigerators into smart ones, and an autonomous sensor array is required that can constantly monitor and notify users about the state of the refrigerators’
contents. A light, accurate, and low-latency model is required that can perform object-identification and recommend recipes.

4. System Design

4.1 Software Specifications:
I. Object Identification Module:
a) We use Convolutional Neural Network for object identification by implementing YOLO architecture. Python code is written on Google Collab Notebook.
b) Libraries required: pandas, NumPy, os, pickle, Matplotlib, and dark flow.
II. Deploying the model on Cloud:
a) Google AI platform - To train the object identification model.
b) Google Compute Engine - To deploy the trained model on the cloud.
c) Libraries and packages required: Google Cloud SDK, notif-tools. Google CloudVM (Virtual Machine) Instance details- Debian Operating System, 3.75GB RAM, 10GB Storage Memory. The system's fundamental representation is shown in the block diagram in Figure 1, and the flow chart is shown in Figure 2.

![Figure 1: Block Diagram of Object Identification](image1)

![Figure 2: Flowchart of Object Identification](image2)
5. Yolo Architecture
This variant of our model is an object detector because the name suggests it detects objects within images. It is different from the mobile net model. In this, it is ready to identify multiple objects within identical images.

Our network uses features from the whole image to predict each bounding box. The bounding box is an imaginary, 2-dimensional, rectangular box, drawn around an object as in Figure 3; it encompasses it. Here, every bounding box's prediction in the picture, covering all the classes, is performed simultaneously, suggesting that the network's reasoning is done globally regarding the complete image and each object that the image comprises. The YOLO algorithm allows real-time speeds and end-to-end training and does this without affecting the average precision and ensuring that it is kept high, as in Figure 4. The system dissects the input image in an $s \times s$ grid as in Figure 5[13]. In case the middle of an object lies when the model thinks that there is no object inside the bounding region. Else, as in Figure 3 the intersection should ideally calculate YOLO architecture output score over union (IOU) of the ground truth with the predicted box cell, irrespective of the amount of boxes O. Every bounding region output five predictions. They are $(x, y)$, which are the coordinates representing the bounding region’s center’s center concerning the grid cell's borders, $(w, h)$ which stand for the width and height are anticipated concerning the whole image. Also, every cell of the grid estimates $p$ probabilities, representing the conditional probability, i.e., given that an object is contained in the bounding box. Simultaneously, this reflects the certainty with which the model thinks that the bounding box contains an object and that the accuracy, which it attributes to the prediction that this object corresponds to a particular class [14].

![Figure 3: YOLO](image)

![Figure 4: Architecture](image)
6. Hardware Description

The major concern about food wastage is food hygiene and safety. The food standard must be checked and kept safe from rotting and spoiling by the climatic components like light, temperature, moistness. In this manner, it is valuable to introduce quality checking gadgets at food stores and homes. This Gadget keeps a watch on the natural factor that causes rot of the nourishment. Refrigeration and vacuum storage control many other natural factors. The hardware kit (to be added to the refrigerator) consists of a camera, screen, a Raspberry Pi, and a sensors array consisting of a DHT 11 sensor, a Humidity and temperature sensor. This MQ3 sensor is used to identify and detect the presence of ethanol and alcohol, and a Node MCU, which is a firmware that is used for open-source prototyping board design. In this project, the quality checking device keeps a watch on natural factors like alcohol, Spoilage, temperature, light, and humidity [15]. The main processor board used is Node MCU- 8266. It is interfaced with sensors like MQ3 to detect food quality and spoilage. It is an IoT device that sends the sensor values to a database that is connected to a dashboard. The IoT dashboard is used for logging and monitoring the sensor’s data. The dashboard used here is Cayenne. The data can be logged from anywhere, anytime, and from any gadget because of the internet of things. The position of these sensors is shown in Figure 6.
7. Interpretation
Our system has Raspberry Pi (local system), camera(s), and sensors array inside the refrigerator. The screen is outside the fridge to have access to the data. Once it is powered through, it connects to the strongest Wi-Fi based on user input. The camera is placed to achieve a bird’s-eye view of the vegetable cabinet in our model. However, in real-time refrigerators, we will have cameras in multiple positions to cover all possible locations inside the refrigerator. When the refrigerator door is opened, the lights turn on, and LDR detects the light that triggers the camera (connected to Raspberry Pi). The camera captures the cabinet’s picture and sends it to the Pi, as shown in Figure 9. Our object recognition model (YOLO v2) is deployed on the cloud (Google Cloud Platform) for fast and efficient processing. Hence, the Pi uploads the captured image to the Cloud Compute virtual machine, which contains our object detection model. Here, the computing machine performs complex computations on the uploaded picture by passing it through the model for object identification, and it takes approximately eight seconds to recognize the items. Once the recognition is done, the local system receives the output (Raspberry Pi) in the form of a list, as shown in Figure 10. This output is then used in the recipe retrieval process (on the Pi using an online browser), and the recipes for the objects detected are displayed on the screen, as shown in Figure 10. This uploading and downloading of the image and the recipe retrieval display consume approximately 20 seconds, bringing the total time of the complete process to nearly 30 seconds. The IoT device is installed inside the fridge. The sensors read the data continuously and transfer them via Wi-Fi. The DHT 11 is a temperature and humidity sensor, as mentioned before. It is a digital sensor that detects temperature and humidity every two seconds. The sensor with a voltage supply of 3.5 to 5.0 v, and the temperature ranges from 00 c to 500 c. The sensor operates on a 1 – wire protocol implemented on the firmware, because of which the sensor cannot interface with digital pins. The 40 bytes of data read by the sensor consists of temperature and humidity. The mq3 sensor detects gases like ethanol and alcohol. It has to be placed where conservative foods (fruits and vegetables) are kept. The sensor keeps on detecting the concentration of ethanol. Once the concentration reaches the threshold level, the sensor signals to the analog pin. The prototype board has an inbuilt ADC. All the sensor values keep sending to the cayenne server. The cayenne server stores the data and displays it on the dashboard. The workflow is shown in Figure 7 below:

![Figure 7: Workflow](image-url)
8. Results and Discussion
The model’s efficiency was empirically observed to be 93.77%. Multiple objects were placed in the same picture and tested to test our model's real-time performance. The model was successfully able to identify all the images. The metric used is the confidence score as output by the model. The objects were kept at a distance of 1 m. As observed, the confidence score is independent of the number of classes present, as shown in Figure 8.

| IMAGE | No. of Objects | Classes of Objects | confidence score |
|-------|----------------|--------------------|------------------|
| 1     | 1              | Banana             | 0.45             |
| 2     | 1              | Apple              | 0.63             |
| 3     | 1              | Orange             | 0.7              |
| 4     | 2              | Apple, Orange      | 0.5, 0.6         |
| 5     | 2              | Apple              | 0.49             |
| 6     | 3              | Banana, Orange     | 0.69, 0.57       |

**Figure 8**: Confidence score

This is because the model attempts to draw bounding boxes around objects that it identifies and predicts that box's confidence alone. It was observed that even on occlusion (Figure 9 - apple is occluded), the confidence score of the occluded object decreased considerably; However, the model was still able to make a fairly confident prediction shown in Figure 10.

**Figure 9**: Image captured

**Figure 10**: Result obtained from the given input
The recipes that feature the objects detected are retrieved and displayed on the screen, as shown in Figure 11.

![Figure 11: Recipes displayed](image)

The status of the refrigerator contents is also displayed on the screen, as shown in Figure 12.

![Figure 12: Dashboard on the screen](image)

![Figure 13: Notification email](image)
Here, Figure 13 shows that Notification email will go to users

9. Conclusion
The model uses YOLO for object recognition to detect vegetables and fruits and display recipes that include those foods. It also monitors several factors, like freshness and age of the components. It is a user-interactive intelligent refrigerator; a customized recipe recommendation system can be developed as per the user's choice, and a voice assistant can be integrated with the module. Automation of inventory restocking can be added. A mobile application can be developed for the user to access from anywhere.

References
[1]. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, And L. Fei-Fei, "Imagenet Large Scale Visual Recognition Challenge," In International Journal Of Computer Vision (Ijcv).2015.
[2]. A. Krizhevsky, I. Sutskever, And G. Hinton, “Imagenet Classification With Deep Convolutional Neural Networks,” In Neural Information Processing Systems (Nips), 2012.
[3]. Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun," Faster R-Cnn: Towards Real-Time Object Detection with Region Proposal Networks," Doi 10.1109/Tpami.2016.2577031, IEEE Transactions on Pattern Analysis and Machine Intelligence.
[4]. Peng Bian1, Wanwan Li2,3, Yi Jin2,3* And Ruicong Zhi4,5 “Ensemble Feature Learning For Material Recognition With Convolutional Neural Networks” Bian Et Al. Eurasip Journal On Image and Video Processing (2018) 2018:64
[5]. S. Ren, K. He, R. Girshick, And J. Sun, “Faster R-CNN: Towards Real-Time Object Detection With Region Proposal Networks,” In Neural Information Processing Systems (Nips), 2015.
[6]. R. Girshick, “Fast R-CNN,” In IEEE International Conference on Computer Vision (ICCV), 2015.
[7]. Ajeet Ram Pathaka,*, Manjusha Pandeya, Siddharth Rautaraya “Application of Deep Learning for Object Detection “International Conference on Computational Intelligence and Data Science (ICCIDIS 2018).
[8]. Parisa Pouladzadeh, Shervin Shirmohammadi, “Mobile Multi-Food Recognition Using Deep Learning Acm Trans. Multimedia Comput. Commun. Appl., Vol. 13, No. 3s, Article 36, Publication Date: August 2017.
[9]. Zhong-Qiu Zhao, Peng Zheng, "Object Detection with Deep Learning: A Review” IEEE 2019.
[10]. Xinyi Zhou, Wei Gong, WenLong Fu3," Application of Deep Learning in Object Detection,” IEEE 2017.
[11]. Hsin-Han Wu, Yung-Ting Chuang, Low-Cost Smart Refrigerator, 2017 IEEE 1st International Conference on Edge Computing
[12]. S. Qiao, H. Zhu, L. Zheng, and J. Ding, "Intelligent Refrigerator Based on Internet of Things,” 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (Euc), Guangzhou, 2017, Pp. 406-409, Doi: 10.1109/Cse-Euc.2017.262.
[13]. Ganjali, M., Qashqaei, M. and Hosseini, S. E., 2020. Microwave photonic frequency multiplication based on Sagnac interferometer with the capability of phase shifting. Optics Communications, 474, p.126059.
[14]. You Only Look Once: Unified, Real-Time Object Detection Joseph Redmon, Santosh Divvala†, Ross Girshick, Ali Farhadi University of Washington, Allen Institute For Ai, Facebook Ai Research
[15]. Haidawati Nasir, Wan Basyar Wan Aziz, Fuead Ali, Kushsairy Kadir “The Implementation of IoT based Smart Refrigerator System,” 2018 2nd International Conference on Smart Sensors and Application (ICSSA).