

## Article

# Revolutionizing Small-Scale Retail: Introducing an Intelligent IoT-based Scale for Efficient Fruits and Vegetables Shops

Abdullah Zargham <sup>1</sup>, Ihtisham Ul Haq <sup>2</sup>, Tamara Alshloul <sup>3</sup>, Samad Riaz <sup>1</sup>, Ghassan Husnain <sup>4</sup> ,  
Muhammad Assam <sup>5</sup>, Yazeed Yasin Ghadi <sup>6</sup>  and Heba G. Mohamed <sup>7,\*</sup> 

<sup>1</sup> Department of Electrical Engineering, University of Engineering & Technology, 25000 Peshawar, Pakistan

<sup>2</sup> Department of Mechatronics Engineering, University of Engineering & Technology, 25100 Peshawar, Pakistan

<sup>3</sup> Department of General Education, Liwa College of Technology, 15222 Abu Dhabi, United Arab Emirates

<sup>4</sup> Department of Computer Science, Iqra National University, 25100 Peshawar, Pakistan;  
ghassan.husnain@gmail.com

<sup>5</sup> Department of Software Engineering, University of Science and Technology, 28100 Bannu, Pakistan

<sup>6</sup> Department of Computer Science, Al Ain University, 64141 Al Ain, United Arab Emirates

<sup>7</sup> Department of Electrical Engineering, College of Engineering, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia

\* Correspondence: heg Mohamed@pnu.edu.sa

**Abstract:** In the bustling streets of Pakistan, small-scale fruits and vegetables shops stand as vital hubs of daily life. These humble establishments are where people flock to satisfy their everyday needs. However, the traditional methods employed by shopkeepers using manual weighing scales have proven to be time-consuming and limit the shopkeepers' ability to serve multiple customers simultaneously. But in a world rapidly embracing automation and propelled by the wonders of artificial intelligence, a revolution is underway. In this visionary paper, we introduce the concept of an intelligent scale that will revolutionize the retail process. This remarkable scale possesses the power to automate numerous tasks, making the shopping experience seamless and efficient. Imagine a scale that not only determines the weight of the produce but also possesses the ability to detect and identify each fruit and vegetable placed upon it. By harnessing the potential of cutting-edge technology, we fine-tuned pre-trained models, such as YOLOv5n and YOLOv7, on our extensive dataset, consisting of 12 classes and 2622 images. The dataset was collected manually and it closely aligns with real-time scenarios, ensuring that the distribution in our training and validation sets were similar and that it reflected what our models will encounter during testing. As a result, our YOLOv5n and YOLOv7 models have achieved astonishing mean Average Precision (mAP) scores of 0.98 and 0.987, respectively. YOLOv5n demonstrates an impressive processing speed of 20 frames per second (fps) on a CPU, while it reaches an impressive 125 fps on a GPU. Similarly, YOLOv7 achieves a processing speed of 2 fps on a CPU, which escalates to 66.6 fps on a GPU. These extraordinary results testify to the remarkable accuracy and efficacy of our system when subjected to real-world testing scenarios. To ensure accurate weighing, we incorporated a load cell with an hx711 amplifier, providing precise measurements that customers can trust. However, our intelligent scale does not stop there. We understand that determining weight alone is insufficient when it comes to transactions. Hence, a meticulously crafted Python script was developed to map each specific item to its corresponding price based on its weight. With all these incredible features in place, the experience of purchasing from a fruits and vegetables shop is taken to new heights. The intelligent scale is accompanied by a user-friendly graphical user interface (GUI), where customers can conveniently view their order and prices. Once the order is complete, a simple click on the print button generates a neatly printed bill, ensuring a seamless transaction. The implications of this intelligent scale are profound. Shopkeepers can now serve customers faster and more efficiently, effortlessly managing multiple transactions simultaneously. The introduction of automation enhances the overall shopping experience, leaving customers delighted and eager to return. This amalgamation of technology and traditional commerce heralds a new era, where small-scale shops can thrive and adapt to the ever-evolving needs of the modern world.



**Citation:** Zargham, A.; Haq, I.U.; Alshloul, T.; Riaz, S.; Husnain, G.; Assam, M.; Ghadi, Y.Y.; Mohamed, H.G. Revolutionizing Small-Scale Retail: Introducing an Intelligent IoT-based Scale for Efficient Fruits and Vegetables Shops. *Appl. Sci.* **2023**, *13*, 8092. <https://doi.org/10.3390/app13148092>

Academic Editor: M. M. Kamruzzaman

Received: 1 June 2023

Revised: 4 July 2023

Accepted: 7 July 2023

Published: 11 July 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** AIoT; computer vision; deep learning; weighing system; graphical user interface; fruits and vegetable detection

---

## 1. Introduction

In recent years, the rapid transformation driven by digitization and automation has brought about profound changes in various facets of our lives. Significantly, advancements in technology, particularly in the realm of artificial intelligence (AI), have served as the catalyst for this paradigm shift. Digitization involves the conversion of analog information into digital formats, while automation leverages technology to execute tasks with minimal human intervention. The integration of AI applications has played a pivotal role in harnessing the full potential of digitization and automation. By utilizing AI, digitization enables the analysis of unstructured data, encompassing images, audio, and text, thereby extracting valuable insights from these sources. For instance, AI-powered image recognition algorithms have facilitated the automatic tagging and categorization of images. Simultaneously, AI's impact on automation is equally momentous, empowering machines to perform tasks that were traditionally dependent on human intelligence. These intelligent machines execute complex tasks with precision and swiftness, bolstering work efficiency and diminishing labor costs. By leveraging digitization and AI-powered automation, we can bring about a transformative change in the small-scale retail sector of Pakistan.

In Pakistan, where a significant proportion of the population belongs to a low socio-economic status [1], small-scale retail outlets play a pivotal role in fulfilling their daily consumption needs. These outlets offer goods in quantities that are affordable and accessible to this segment of the population, given their limited purchasing power, unlike large-scale supermarkets that primarily cater to bulk purchases. However, the pricing of goods in these small-scale retail outlets relies on conventional weight scales, which involve a manual and time-consuming process. To address the inherent inefficiencies and high labor demands associated with this traditional method, it becomes imperative to embrace digitization and automation. Drawing inspiration from the advancements in artificial intelligence (AI), we propose the adoption of an intelligent weighing scale in small-scale retail outlets across Pakistan. This innovative scale would incorporate a deep learning algorithm capable of real-time detection and recognition of various fruits and vegetables, accurately weigh the items, and display their corresponding prices on a user-friendly graphical interface. Such implementation would significantly enhance operational efficiency and reduce labor requirements, consequently benefiting both the retailers and the customers in these outlets.

Deep learning has seen a significant rise in recent years, which can be primarily attributed to the progress in digital technologies and availability of large data sets [2]. Digital technologies such as high-performance computing and powerful GPUs have facilitated the training of deep neural networks on large data sets [2]. This combination of powerful computing resources and large data sets has enabled deep learning to achieve state-of-the-art performance in a wide range of applications of computer vision [3]. Deep learning models have played a crucial role in the development of real-time object detection systems. Object detection is a computer vision task focused on recognizing and localizing objects present in images or videos. It involves identifying the presence of specific objects and determining their spatial coordinates within the given visual data [4]. The objective of object detection is to enable machines to perceive and understand the content of images or video frames [5]. Traditional object detection algorithms, such as Harris Corner Detector [6], Scale-Invariant Feature Transform (SIFT) [7], and Speeded Up Robust Features (SURF) [8,9], rely on hand-crafted features and heuristics to detect objects. These object detection algorithms are relatively simple and computationally efficient, but they are not as accurate and robust as deep-learning-based object detection algorithms. In contrast to traditional approaches, deep-learning-based models employ convolutional neural networks (CNNs) [10]

to automatically extract features from raw image data. CNNs are designed to capture hierarchical representations of visual information, enabling them to learn complex patterns and structures present in images [11]. By leveraging these learned features, deep learning models exhibit increased robustness to variations in object appearance, such as changes in lighting conditions, viewpoints, and occlusions [12]. The capability of deep-learning-based object detection models to generalize well to new images and improve detection performance compared to hand-crafted feature-based methods has been harnessed effectively for fruits and vegetables detection and recognition. In this context, we have specifically fine-tuned the parameters of two versions of the YOLO algorithm, namely YOLOv5n [13] and YOLOv7 [14], capitalizing on their inherent capacity to adapt and excel in detecting fruits and vegetables across various types of images. YOLO, or You Only Look Once, is a single-stage object detector, as opposed to the commonly used two-stage object detectors such as R-CNN [15], Fast R-CNN [16], and Faster R-CNN [17]. YOLO algorithm, adopts a unique methodology of processing images by making predictions in a single pass [18]. It analyzes the entire input image as a whole and directly predicts the presence and spatial coordinates of objects within it. This approach distinguishes YOLO from traditional two-stage object detection methods, resulting in computational efficiency by eliminating the need for region proposal and subsequent refinement steps [18]. By processing the image only once, YOLO achieves real-time object detection capabilities, making it well-suited for resource-constrained environments [19].

In this paper, we propose a computer vision-based weighing scale system for the real-time detection and measurement of fruits and vegetables. The system comprises a camera, a loadcell, an hx711 module, an Arduino microcontroller, and a laptop. The camera captures real-time video, which is then processed by a deep learning model, such as YOLOv5n or YOLOv7, running on the laptop to make real-time predictions of the object. The Arduino microcontroller interfaces the hx711 module with the loadcell, which is used to measure the weight of the objects. Serial communication is used to transfer the weight measurements to the laptop. A python script has been written in order to create a graphical user interface (GUI). The GUI has everything on it, such as particular item real time video, its name, weight, and price with respect to their weight. Using graphical user interface (GUI) you can add different items to your bill once you weigh them by just clicking the "add" button. At the end, you can receive your bill in printed form when you complete your order by just clicking the "print" button that has been shown on the graphical user interface (GUI). The system is designed to be efficient, reducing labor costs when multiple weighing scales are utilized, as the shopkeeper needs to enter the price of the object only once. This intelligent weighing scale would empower shopkeepers to serve customers more efficiently, enabling multiple customers to be attended to simultaneously. The reduction in labor requirements would also alleviate the burden on the shopkeepers, allowing them to focus on other aspects of their business.

Our proposed system has the potential to improve the efficiency and accuracy of small-scale retail operations. This study is a step towards modernizing small-scale retail operations and future research could be focused on the implementation of this system in real-world scenarios and its impact on the industry.

The contribution of this article can be summarized into following points:

- This article highlights the challenges faced by small-scale retail shops in Pakistan. It discusses issues such as the manual processes, limited resources, and inefficient management systems that hinder the growth and profitability of these businesses.
- A solution to the aforementioned problems is proposed in the article by suggesting the adoption of global trends such as digitalization and automation. The need for small-scale retail shops in Pakistan to leverage technology to streamline their operations, enhance efficiency, and improve customer experience is emphasized.
- A concise overview of the existing literature related to the detection and classification of fruits and vegetables is provided in the article. The techniques and algorithms utilized for achieving accurate identification and classification of different fruits and

vegetables are discussed. Furthermore, the literature related to weighing systems is also explored.

- The process of collecting the necessary dataset for training the model is described in the article. How the dataset should be aligned with the scenario expected during testing or real-time implementation of the model is explained. The collection methods may include the passive capturing of images of fruits and vegetables from different angles, under varying lighting conditions, and with different orientations.
- Step-by-step instructions for training the model using the collected dataset are provided in the article. Preprocessing steps, including data annotation and splitting, are covered. Furthermore, the model validation process is explained.
- Finally, an overview of a user-friendly graphical user interface (GUI) for the proposed prototype was provided. The GUI is designed to simplify the interaction between the users and the automated system. An intuitive interface is provided for tasks such as order initiation, prize editing, item addition, and order completion bill, aiming to enhance user experience and streamline the process.

## 2. Literature Review

### 2.1. Fruits and Vegetables Classifications

In the proposed prototype, the main objective is to classify and recognize different types of fruits and vegetables, which can be achieved using image classification or object detection models. Prior research has been conducted by various researchers who have utilized different deep learning models to perform classification, recognition, and detection tasks for different types of fruits and vegetables. Let us begin by examining the earlier works in which researchers conducted the classification or identification of diverse types of fruits and vegetables. There are numerous studies that have employed a range of techniques such as Convolutional Neural Networks (CNNs) and Deep Neural Networks (DNNs). In reference [20], the author conducted a study on classification and recognition of fruits. He utilized a convolutional neural network for the classification of fruits. In his study he used two datasets composed of 3 and 20 classes, respectively. He trained CNN using three different optimization algorithms such as Adam, SGD, and RMSprop. In another paper [21], the author employed a DNN based on CNN for vegetable category recognition. He utilized a dataset composed of 8 classes and 160 images. He performed 3 million iterations for training CNN which seem computationally expensive. In the research in reference [22], caffe and chainer DNN frameworks were utilized for vegetable recognition. The dataset used in this study consisted of 15 classes and 150 images. It transpires that the limited size of the dataset used in studies [21,22] may lead to concerns about the ability of the model to generalize well to new datasets. In this work [23], the author presented the idea of transfer learning. He utilized various convolutional neural network (CNN) architectures on a dataset composed of 15 classes and 21,000 images for image classification. The typical CNN and four state-of-the-art architectures, including VGG16, MobileNet, InceptionV3, and ResNet, were trained and their performances were compared. Results showed that training a CNN from scratch on a small dataset resulted in lower performance compared to training a pre-trained model. The findings of this study suggest that, when working with small datasets, transfer learning is a technique that can be applied to enhance the performance of the algorithm. In this study [24], the author addresses the challenge of fruit classification in the horticulture industry by leveraging advancements in AI and deep learning. The author employs a dataset consisting of 26,149 images encompassing 40 different types of fruits for experimentation. To enhance the classification performance, the author introduces a customized head comprising five different layers into the MobileNetV2 architecture. By replacing the classification layer of MobileNetV2, a modified version called TL-MobileNetV2 is obtained. Transfer learning is employed to leverage the knowledge gained from pre-trained models. By retaining the pre-trained weights, TL-MobileNetV2 achieves an impressive accuracy of 99%, surpassing the performance of MobileNetV2 by 3%. Moreover, TL-MobileNetV2 exhibits a remarkably low

equal error rate of just 1%. Precision, recall, and F1-score measures demonstrate excellent performance, with TL-MobileNetV2 achieving 99% for each metric. In this study [25], the authors propose an autonomous system using a CNN-based approach to classify date fruit, addressing the absence of such a system and the limitations of manual expertise. They create a dataset with eight distinct classes of date fruit for training and employ preprocessing techniques like image augmentation, decayed learning rate, model checkpointing, and hybrid weight adjustment to enhance the model's performance. The proposed model, based on MobileNetV2 architecture, achieves an impressive 99% accuracy rate. Comparing it to other models like AlexNet, VGG16, InceptionV3, ResNet, and MobileNetV2, the proposed model consistently outperforms them in terms of accuracy, validating its effectiveness in date fruit classification.

Overall, the literature demonstrates the application of deep learning models, such as CNNs and DNNs, for fruit and vegetable classification. The use of transfer learning and pre-trained models shows promise in enhancing model performance, particularly when working with small datasets. However, it is important to note that the literature does not directly address the real-time detection and recognition of fruits and vegetables. The studies primarily focus on the classification task, where the goal is to assign a specific category or label to an input image. Real-time detection and recognition involve identifying and localizing fruits and vegetables in a video or live camera feed, which requires additional techniques such as object detection or instance segmentation.

## 2.2. Fruits and Vegetables Detections

The preceding literature focuses on classification and recognition tasks in non-real-time settings. However, our proposed work requires the real-time recognition and classification of fruits and vegetables. Due to this, we had to employ an object detection model for carrying out the recognition and classification tasks in real time. In previous research, several researchers have explored object detection algorithms for fruits and vegetables recognition and classification in real time. In this section, we will provide a summary of these techniques. In one work, the idea of a billing mart was presented [26]. YOLOv2, an object detection algorithm, was utilized by the author for the detection and recognition of vegetables. The model was trained on three classes and an accuracy of 70% was achieved, which is not considered sufficient for real-time object detection. However, no graphical user interface (GUI) was developed for shopping purposes. In another study [27], YOLO algorithm was used for the classification and detection of vegetables. A dataset of 100 images of three different classes of vegetables was collected by the author. Out of these, 60 images were used for training purposes, while 40 images were utilized for testing. However, due to the limited size of the dataset, there was a lack of diversity in the orientation of the vegetables, resulting in poor performance in detecting unknown orientations. This poor performance was attributed to the absence of data augmentation. Consequently, an accuracy of only 61.6% was achieved by the model, which is deemed insufficient for real-time vegetable detection. These results highlight the necessity for larger and more diverse datasets in order to train robust and accurate models for vegetable classification and detection tasks. In another paper [28], the utilization of the deep learning model YOLOv4-tiny for the task of fruits and vegetables detection was described. A dataset was collected from various sources such as Google and Kaggle, and it was labeled using the Roboflow framework. The results obtained from the YOLOv4-tiny model showed a mean average precision (mAP) of 51% with an inference time of 18 milliseconds. However, upon further analysis, several reasons for the suboptimal performance of the model were identified. Firstly, the dataset used in this study was collected from multiple sources, which led to a lack of uniformity in the distribution of the dataset. Additionally, the number of examples per class varied, with some classes having a higher number of examples than others. In the research of reference [29], the author proposes an accurate and real-time image-based multi-class fruit detection system for smart farms. The framework utilizes an improved Faster R-CNN deep learning model, comprising a fruit image library, data augmentation techniques, an

enhanced model, and performance evaluation. The work's notable contributions include a comprehensive outdoor orchard image library with 4000 real-world images and optimized convolutional and pooling layers for higher accuracy and faster detection. The test results demonstrate superior accuracy and processing time compared to traditional detectors. The proposed algorithm achieves over 91% mean average precision (mAP) for detecting apples, mangoes, and oranges, with improved image processing speed, making it suitable for autonomous harvesting and yield mapping systems. In another paper [30], a novel fruit detection algorithm is presented for a plum harvesting robot. The algorithm addresses the challenges of accurately recognizing plums, considering their small size, dense growth, and occlusions in the environment. The proposed approach is a lightweight plum detection procedure based on an improved version of the YOLOv7 algorithm. The methodology involves collecting plum images, establishing train/validation/test sets, and training the detection model with data augmentation. The authors introduce modifications to YOLOv7, including updated anchor box sizes based on observed plum sizes and an SE module for capturing channel interdependencies. The Improved-YOLOv7 model achieves promising results, with Precision, Recall, and mAP scores of 70.2%, 72.1%, and 76.8%, respectively. Comparative analysis demonstrates that the model outperforms other YOLO models in terms of accuracy and generalization in complex environments. Furthermore, in our work, we also included the class of plum, and our YOLOv7 model demonstrates greater precision and recall for this class compared to the research mentioned. In another paper [31], the author presents a fruit detection model called YOLO-Oleifera, specifically designed for oil-seed camellia fruit in orchards. It modifies the YOLOv4-tiny architecture to address challenges such as lighting changes, occlusion, and fruit overlap. The model uses the k-means++ clustering algorithm to improve bounding box priors for accurate fruit detection. Additional convolutional kernels are added to reduce computational complexity while effectively learning fruit features. The model utilizes bounding boxes for region of interest extraction and adaptive stereo matching, enabling precise fruit positioning. Ablation experiments demonstrate the effectiveness of the modifications. Testing shows robust detection performance under varying illumination conditions, with reduced precision and recall for occluded fruit. Compared to other models, YOLO-Oleifera achieves the highest Average Precision with a small data weight of 29 MB. It demonstrates real-time capability and stability in complex orchard environments, serving as a reference for mobile picking robots. In another paper [32], three models for fruit detection and classification were utilized. These models included CNN, YOLOv4, and YOLOv5. The CNN was trained on the fruit360 dataset, which consisted of 131 classes of fruits and vegetables. After fine-tuning, the CNN achieved an accuracy of 98%. However, it was observed that, despite its high accuracy, the CNN did not perform well in real-time scenarios. To address this issue, the author collected two additional datasets with 19 and 12 classes, respectively. Two YOLO models, namely YOLOv4 and YOLOv5, were trained using these datasets. The YOLOv4 model achieved an accuracy of 70%, while the YOLOv5 model achieved an accuracy of 78%. These later models demonstrated improved performance in real-time scenarios compared to the CNN, but their performance did not reach the baseline level. Furthermore, the author developed a Python script for billing purposes. However, no specific device was mentioned for weighing purposes in the research. The detection and classification focused solely on fruits and vegetables, with fixed weights and prices. As a result, these methods may not be suitable for small-scale outlets with low purchasing power, particularly in the context of the majority of the population in Pakistan. In another study [33], the author developed an Internet of Things (IoT)-based system for the sale of fruits and vegetables. The system incorporates a scale that automatically weighs and identifies the fruits and vegetables, providing a corresponding bill to the customer. To enhance the detection capability of the system, the author utilized the fruit360 dataset and supplemented it with additional collected data. The combined dataset consisted of 2100 training images. To train the detection model, the author employed a Single Shot Multi Frame Detector (SSD) implemented in Python. The SSD model takes an image as input and automatically detects and localizes objects within

the image. It achieves this by framing the position of the objects and employing correction techniques to refine the object's location. The model also returns labels indicating the type of object detected. The author reported an impressive accuracy of 94.50% for the trained SSD model, accompanied by a precision of 95% and a recall of 95.09%. Furthermore, the algorithm demonstrates an inference speed of 3.06 Frames Per Second (FPS). It is worth noting that the dataset used in this study was a combination of the fruit360 dataset and a manually collected dataset. Although the achieved accuracy is excellent, it was tested on a set of only 210 images. In comparison, our models achieved a mean Average Precision (mAP) of 0.98 and were validated on 525 images with 1818 instances. Moreover, our work exhibits slightly higher precision and recall. Additionally, the graphical user interface (GUI) we developed does not require a database, unlike the system described in this study.

Overall, a comprehensive analysis of the existing literature indicates that the majority of models focusing on the detection and classification of fruits and vegetables heavily rely on the fruit360 dataset or datasets sourced from online resources. However, these studies often suffer from the limitation of utilizing small datasets, resulting in inadequate performance when it comes to detecting unknown orientations and variations. The reported accuracy levels in most of these studies range from 51% to 78%, which falls short of the requirements for real-time applications that demand high precision and recall [34]. This gap often overlooks the crucial aspect of achieving accurate measurements on validation datasets, as the datasets utilized in the aforementioned literature review tend to differ from what the models would encounter during testing. Additionally, there is a notable gap in the literature regarding the development of weighing systems and graphical user interfaces (GUIs) for practical implementation. To facilitate the accurate measurement of variable weights of different fruits and vegetables, it is imperative to have a weighing scale with a precise weighing system. Moreover, in order to enhance usability, an interface that allows customers to utilize the weighing scale for self-service purposes is essential.

### 2.3. Weighing Scale

Our proposed prototype involves the secondary task of weighing specific fruits and vegetables. Previous research has shown that the hx711, in combination with a load cell, can be utilized for this purpose [24,28]. The load cell generates very small voltages within the range of millivolts and detecting small changes in voltages can be difficult [28]. To address this issue, the hx711 microcontroller has an inbuilt Analog-to-Digital Converter (ADC), which directly converts analogue voltages into digital values. The authors also used Arduino for the interfacing of the hx711 amplifier.

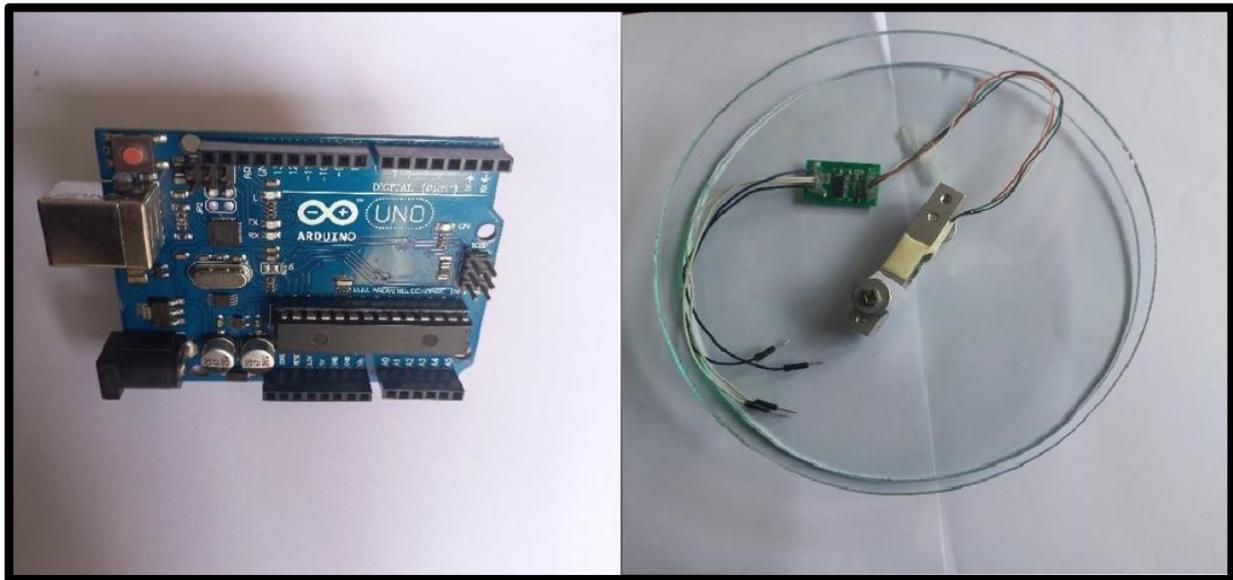
## 3. Methodology

The methodology of this research paper can be divided into three main sections: hardware, deep learning models, and graphical user interface.

### 3.1. Hardware Section

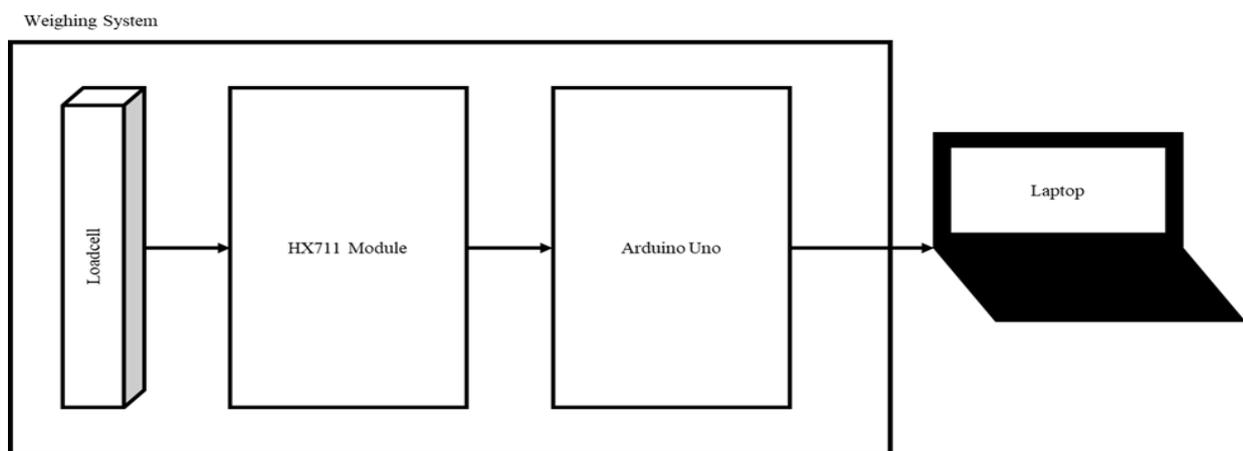
The hardware component of our prototype, as shown in Figure 1, comprises a load cell, an HX711 amplifier, a laptop, and an Arduino board. The primary function of this section is to weigh various fruits and vegetables and transmit their corresponding weight values to the graphical user interface through serial communication between the laptop and Arduino. The load cell we have used is a type of transducer that converts mechanical deformation into electrical signals, such as voltages. Specifically, we have used strain gauges as our load cell type. Strain gauges are made up of a metal bar with attached strain gauges, and they work based on the Wheatstone bridge principle. When an external force is applied to the load cell, such as by placing fruits or vegetables on it, the resistance of the strain gauges varies. This variation in resistance is directly proportional to the applied force, and is reflected in the form of voltages. The voltages generated are typically in the millivolt range and require amplification for further processing. To amplify the analog voltage readings of the load cell, we have integrated an HX711 amplifier. The amplifier

comprises a microcontroller with an analog-to-digital converter that transforms the analog readings into digital values. We have employed the “HX711.h” library to facilitate this conversion process and make the necessary calibrations for accurate results. The laptop is used to program the Arduino board and interface it with the load cell through the HX711 amplifier. It is also responsible for receiving the weight values transmitted by the Arduino through serial communication. The Arduino board obtains the digital weight values from the HX711 amplifier and communicates them to the laptop via serial communication.



**Figure 1.** Hardware components.

Overall, our prototype provides precise measurement of the weight of individual fruits and vegetables, enabling accurate prediction of their market prices. Figure 2 provides a comprehensive depiction of the complete weighing process.



**Figure 2.** Illustration of the entire weighing procedure.

### 3.2. Deep Learning Model

To accurately predict the prices of all fruits and vegetables, it is crucial to begin by identifying and recognizing each individual fruit and vegetable. This process typically entails a series of steps, starting with the collection and annotation of a suitable dataset. Subsequently, a model is trained on this custom dataset. For the task of fruit and vegetable

identification or classification, we have utilized two versions of the YOLO algorithm: YOLOv5 and YOLOv7.

YOLOv5 is the fifth iteration of the YOLO (You Only Look Once) series, renowned for its effectiveness in object detection tasks trained on the COCO dataset. The YOLO5 architecture consists of a backbone, neck, and head, as shown in Figure 3. YOLOv5 employs CSPDarknet53 as its backbone, which is a convolutional neural network based on DarkNet-53. By utilizing the CSPNet [35] strategy, the feature map of the base layer is partitioned into two parts and subsequently merged through a cross-stage hierarchy. This split and merge approach offers notable advantages to YOLOv5, including a reduction in parameters and computational requirements (FLOPs), thereby improving the inference speed crucial for real-time object detection models. Additionally, YOLOv5 incorporates the Path Aggregation Network (PANet) [36] as its neck, enhancing information flow and enabling efficient feature pyramid creation. Feature pyramids aid in the successful generalization across object scales, facilitating the identification of objects in various sizes. PANet further improves the utilization of accurate localization signals in lower layers, resulting in enhanced object location accuracy. The model head consists of a YOLO layer and is primarily responsible for the final detection step. It utilizes anchor boxes to construct output vectors containing class probabilities, objectness scores, and bounding boxes. YOLOv5 is available in various sizes, namely YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. Each size has its own characteristics in terms of depth and width multiples, resulting in different numbers of parameters and inference speeds. YOLOv5n maintains the same depth multiple of 0.33 as YOLOv5s. However, it reduces the width multiple from 0.5 to 0.25 [37]. As a consequence, YOLOv5n utilizes approximately 75% fewer parameters compared to other YOLOv5 sizes [37]. One notable advantage of YOLOv5n is its significantly higher inference speed, as shown in Figure 4. This means that the algorithm can process object detection tasks more quickly compared to other sizes of YOLOv5.

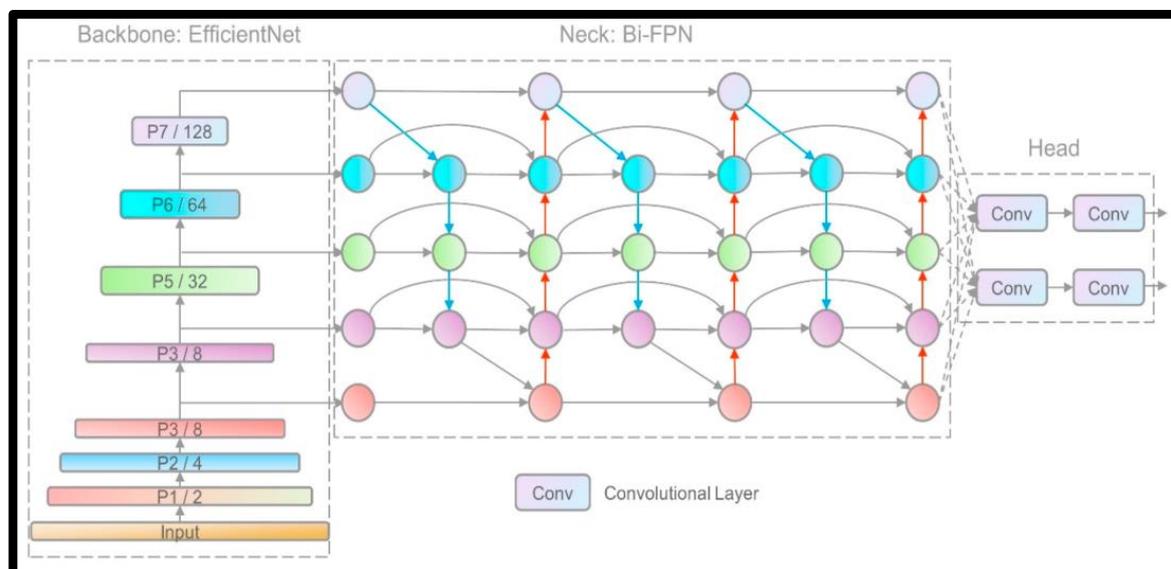
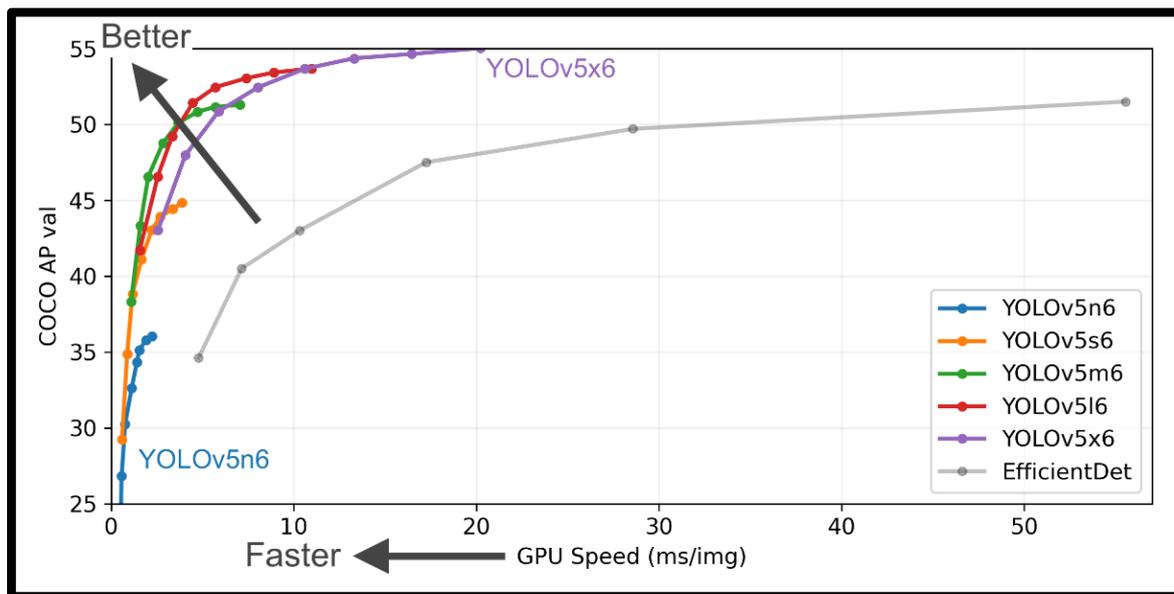


Figure 3. Architecture of YOLOv5 [38].



**Figure 4.** Performance of algorithms [39].

YOLOv7 represents a new iteration of the YOLO (You Only Look Once) model, featuring the distinct architecture showcased in Figure 5. YOLOv7 introduces architectural improvements to enhance detection speed and accuracy. The YOLO architectures typically consist of a backbone, head, and neck. The backbone performs crucial tasks such as extracting important features and passing them to the head through the neck. Unlike its predecessors, YOLOv7 replaces the darknet backbone with an extended efficient layer aggregation network (E-ELAN). E-ELAN is an enhanced version of ELAN that improves the network's learning capabilities while maintaining the stable state achieved by ELAN. It achieves this by introducing expand, shuffle, and merge cardinality operations within the computational blocks, while keeping the transition layer architecture unchanged. Group convolution is employed to expand channels and cardinality. Each computational layer applies the same group parameter and channel multiplier to all its computational blocks. The resulting feature maps are then shuffled into groups based on the set group parameter and concatenated together, preserving the original number of channels in each group. Finally, merge cardinality combines the feature maps from different groups by adding them together. E-ELAN not only retains the original ELAN architecture but also encourages diverse feature learning across different groups of computational blocks [40].

Model scaling is a technique used to modify different characteristics of a model to meet specific requirements for inference speed. When it comes to concatenation-based architectures like YOLOv7, scaling the depth factor can impact the transition layer's in-degree after a computational block. This situation necessitates considering multiple scaling factors simultaneously. For example, increasing the depth can alter the input-output channel ratio of a transition layer, potentially reducing hardware utilization. Therefore, a compound scaling method is necessary for concatenation-based models. This approach entails determining the change in the output channel of a computational block when scaling its depth factor and applying the same change as a width factor scaling to the transition layers. By employing this compound scaling method, the model's original characteristics and optimal structure are preserved [40].

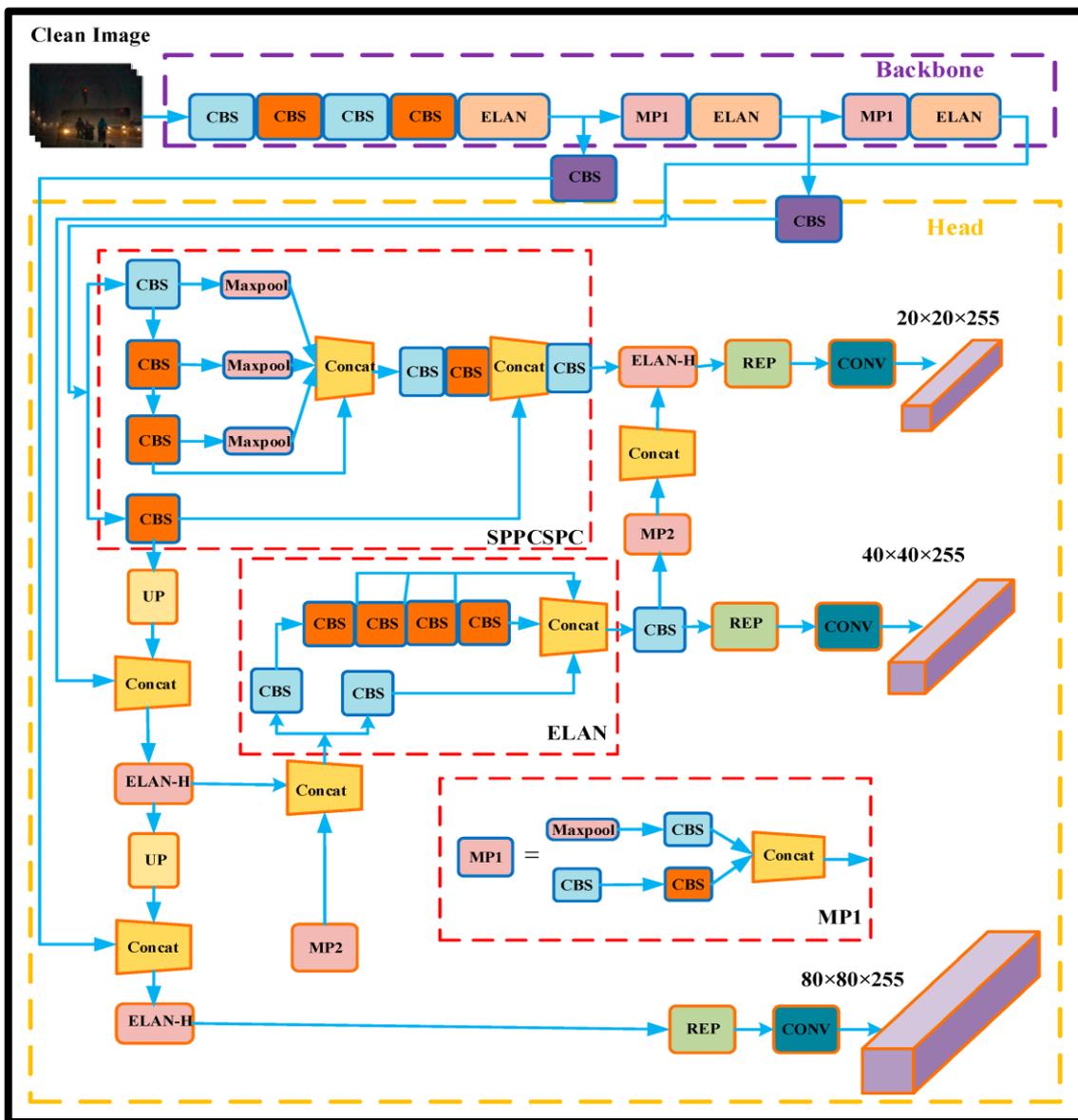


Figure 5. Architecture diagram of YOLOv7 [41].

YOLOv7 incorporates a technique called re-parameterization planning (RP), which focuses on combining multiple models to create a final model that exhibits strong and consistent performance. In RP, specific parts of the model undergo individualized re-parameterization strategies, leading to more effective overall model adaptation. YOLOv7 utilizes gradient flow propagation paths to identify the specific segments or modules within the model that benefit from re-parameterization. This approach helps enhance the model's robustness and performance by leveraging the strengths of different modules within the architecture [42].

In YOLOv7, the lead head serves as the main component responsible for detecting and localizing objects, generating the final output of the network. An auxiliary head is introduced as an additional component in the middle layers, providing extra supervision signals to improve overall model performance. Deep supervision is employed, training both the auxiliary head and lead head using soft labels derived from the lead head's predictions. This approach allows the model to focus on both the representation of data distribution and residual information that still needs to be learned. Two label assignment strategies are introduced: the lead head guided label assigner generates soft labels for both

heads based on ground truth and lead head predictions, while the coarse-to-fine lead head guided label assigner generates coarse and fine labels, optimizing the recall of the auxiliary head in object detection [40].

### 3.3. Collection and Annotation of Dataset

The first step is to collect a comprehensive dataset of fruit and vegetable images, which will serve as the basis for our model training. To achieve this, we have collected 2622 images of 12 different classes of vegetables and fruits. The summary of our dataset including images and instances is shown in Table 1. The images were captured using the camera of a Samsung phone, with varying angles, orientations, and lighting conditions, to ensure the model could generalize well to different scenarios. Furthermore, the images were taken against backgrounds that are similar to our intended usage scenario, which involves the use of a scale to determine the weight of each fruit or vegetable. The preview of our dataset is shown in Figure 6.

**Table 1.** List of datasets with number and name.

Serial Number	Class Name	Number of Images	Number of instances
1	Potato	368	882
2	Tomato	338	1027
3	Onion	250	868
4	Turnip	246	635
5	Chili	314	885
6	Garlic	148	710
7	Carrot	181	301
8	Cucumber	310	467
9	Apricot	155	823
10	Yam	126	731
11	Lemon	68	826
12	Plum	118	947



**Figure 6.** The glimpse of some of the images from dataset.

After collecting the dataset of fruit and vegetable images, the next step is to manually annotate each image with appropriate labels for each fruit and vegetable. To accomplish this, we used the labelling framework [29], which provides a user-friendly interface for manual annotation. Using this framework, we manually drew precise bounding boxes around each fruit and vegetable image and assigned the corresponding labels to each bounding box as shown in Figure 7. The dataset was then split into a training set and a validation set in a ratio of 80:20, respectively, resulting in 2225 training set images along with 128 negative images, and a validation set consisting of 525 images. This process ensured that our dataset was accurately annotated and ready for use in training our supervised deep learning model.



**Figure 7.** A glimpse of some of the annotated images.

### 3.4. Training

Once our dataset had been annotated and split, the next step was to train a deep learning model to recognize and predict the prices of different fruits and vegetables. There are two primary approaches to training a deep learning model, namely training the model from scratch or fine-tuning an already-trained model using transfer learning. In our case, since our dataset contained 2225 training images, which was not considered large enough for training a model from scratch, we decided to employ transfer learning [12] to fine-tune pre-trained deep learning models such as YOLOv5n and YOLOv7. To fine-tune these pre-trained models, we made necessary changes in the Yaml files of each model to adapt them to our specific dataset and task. These changes included adjusting the number of classes to match the number of fruits and vegetables in our dataset and modifying the names of classes according to our custom. Through this process, we were able to leverage the pre-trained weights, such as YOLOv5n.pt and YOLOv7.pt from YOLOv5 and YOLOv7 models, respectively, and learn new features from our dataset to achieve high accuracy in recognizing different fruits and vegetables. The training process of both YOLOv5n and YOLOv7 models is shown in Figure 8. The training process of both the algorithms was the same except the batch size. The batch size for YOLOv5n was chosen as 16 while for YOLOv7 it was selected as 8. At the end of every epoch, the validation of both the models was performed on the validation dataset which was same for both the models. The validation process is shown in Figure 9.

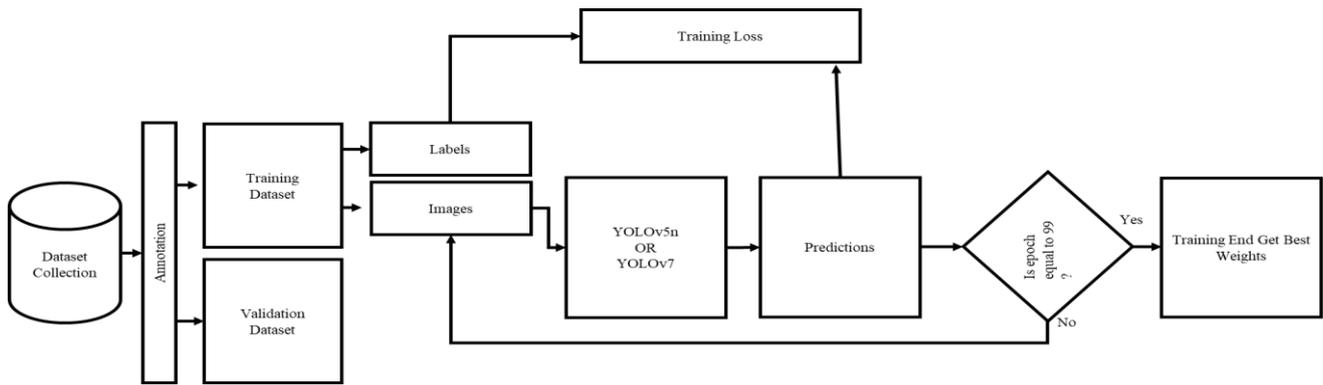


Figure 8. Training process of both YOLOv5n and YOLOv7.

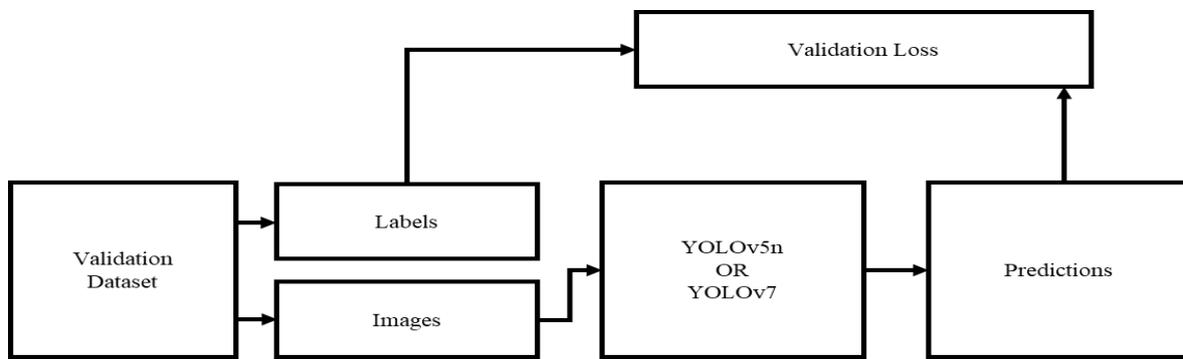


Figure 9. Validation process of both YOLOv5n and YOLOv7.

As the training proceeds, the various type of losses such as class loss, object loss, and bounding box loss, for both training dataset as well as validation dataset of both the models, were decreased. The training and validation losses of both the models was plotted vs number of epochs in Figures 10 and 11, respectively. An analysis of Figures 10 and 11 reveals that neither model exhibits a high bias problem, as indicated by the small training loss, nor a high variance problem, as evidenced by the small validation loss. These remarkable results can be attributed to the meticulous collection of clean and precise data, where uncertainties such as blur images, incorrect labeling, and inaccurate bounding boxes during annotation were effectively minimized.

The decreasing trend in these losses indicates that the model is progressively improving its performance over time. These losses have been computed for both the training dataset and the validation dataset. The formal formulas used to calculate these losses are as follows:

$$l_{class} = \lambda_{class} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{i,j}^{obj} \sum_{C \in class} p_i(c) \log(\hat{p}_l(c))$$

$$l_{obj} = \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{i,j}^{noobj} (c_i - c \wedge_l)^2 + \lambda_{obj} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{i,j}^{obj} (c_i - c \wedge_l)^2$$

$$l_{obj} = l_{obj} = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{i,j}^{obj} b_j (2 - w_i \times h_i) \left[ (x_i - x \wedge_i^j)^2 + (y_i - y \wedge_i^j)^2 + (w_i - w \wedge_i^j)^2 + (h_i - h \wedge_i^j)^2 \right]$$

Both the models were trained for 100 epochs. At the end of training, the best weights of both the models were obtained which were then utilized for inference or real-time fruits and vegetables detections.

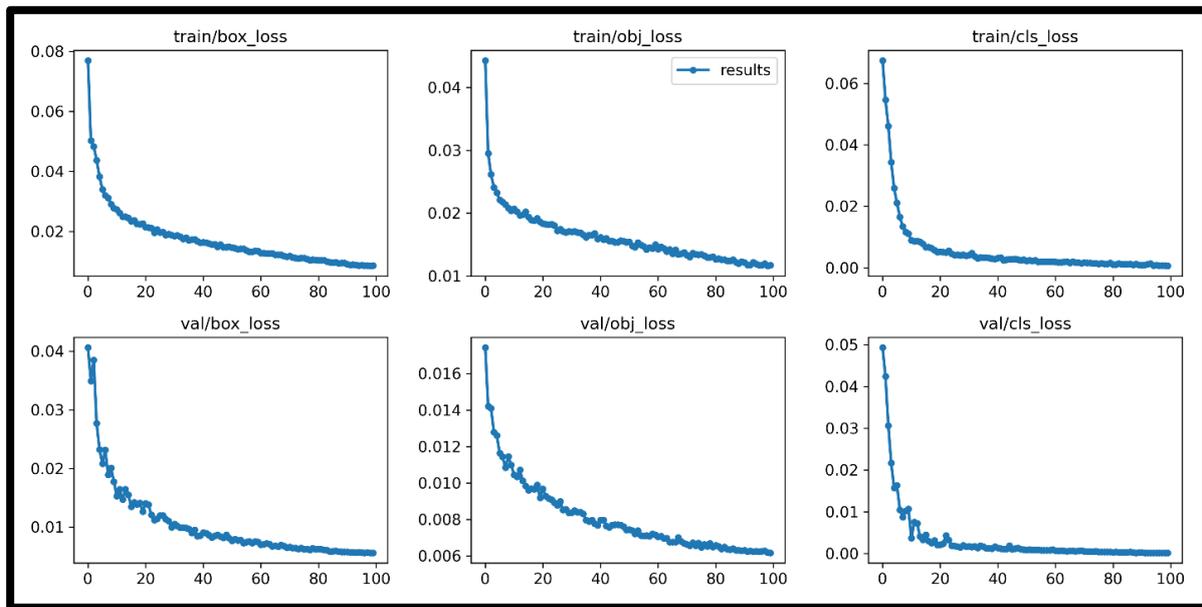


Figure 10. Losses of YOLOv5n during training and validation.

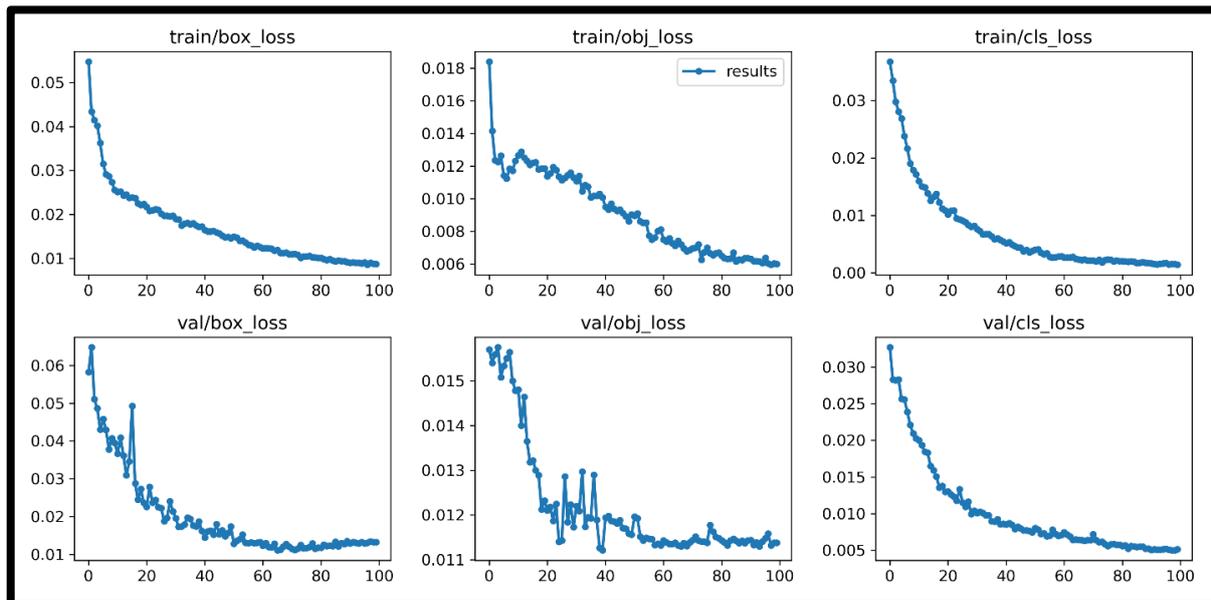
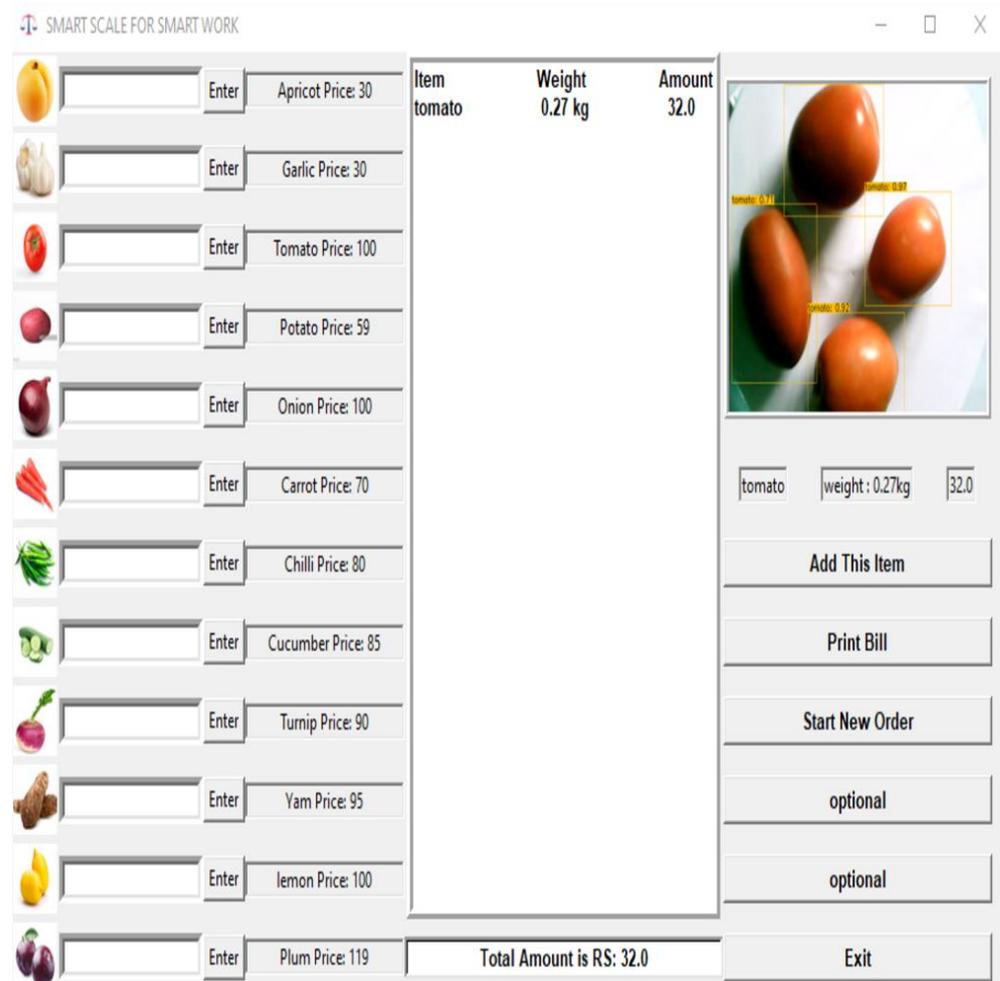


Figure 11. Losses of YOLOv7 during training and validation.

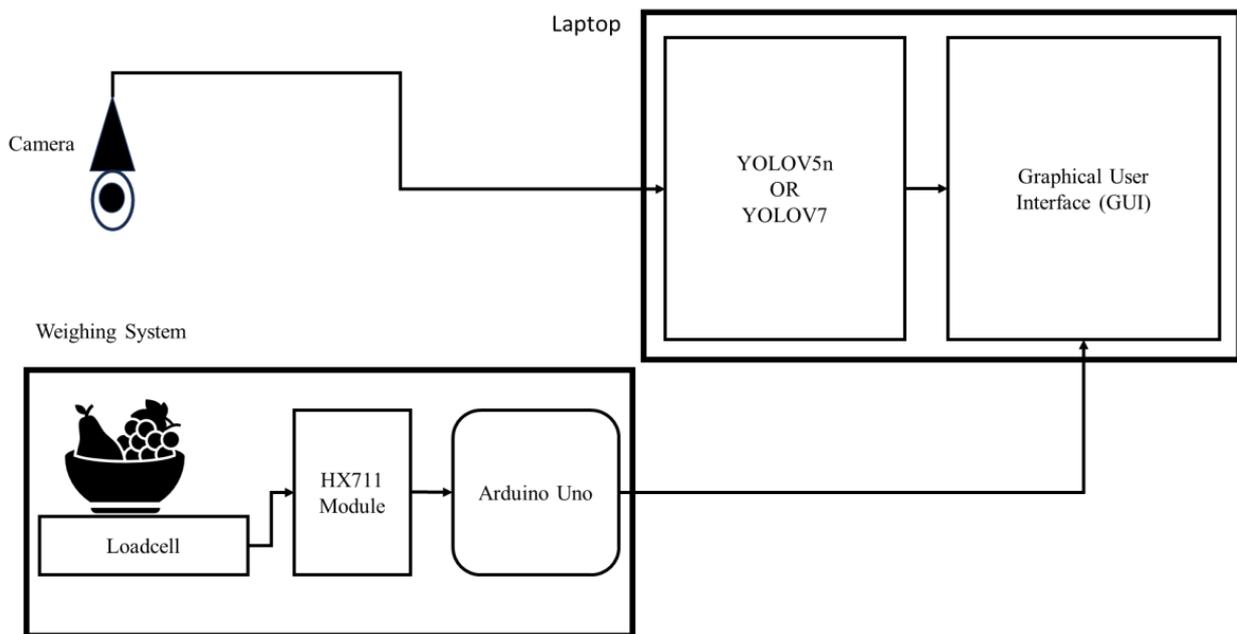
### 3.5. Graphical User Interface

To enhance the user-friendliness of our weighing scale, we developed a Python script that incorporates a graphical user interface (GUI). This GUI runs both the inference code for real-time fruit and vegetables detection and the serial communication code that retrieves weight values. The best weights or parameters of both fine-tuned models, such as YOLOv5n and YOLOv7, can be used for real-time object detection, and the weights of various fruits and vegetables are obtained from the load cell via an Arduino. The GUI displays the real-time object detection video, along with item prices, item weights, and various buttons that allow users to initiate and complete their orders. Additionally, there are buttons available to the shopkeeper, which allow them to enter updated prices for items. Our system was designed to provide a seamless and intuitive user experience, with features that allow for easy ordering and updating of pricing information. The GUI is shown in Figure 12.



**Figure 12.** Preview of graphical user interface (GUI).

The complete architecture of our prototype is depicted in Figure 13. Overall, prototype introduces a novel and innovative approach to selling fruits and vegetables. We have developed a comprehensive architecture that combines a weighing system, a graphical user interface (GUI), and deep learning models for real-time inference. By placing the fruit or vegetable on a load cell, our weighing system accurately measures its weight, which is then transmitted to the GUI via Arduino and laptop communication. Simultaneously, a camera fixed above the load cell captures real-time videos, which are processed by our advanced deep learning model, YOLOv5n or YOLOv7, for precise fruit or vegetable classification and recognition. What sets our system apart is its outstanding performance compared to existing research papers. We have achieved higher mean Average Precision (mAP), precision, recall, F1 score, and inference speed than previous research [26,32,33]. Moreover, our prototype incorporates an attractive GUI that offers user-friendly interactions. It includes features like price editing options, real-time inference display, and a list of items added by the buyer. Notably, our system does not require a database, distinguishing it from other retail systems mentioned in the literature. Overall, our work represents a significant advancement in the field, combining cutting-edge technology, superior performance, and an appealing user interface.



**Figure 13.** The entire process of weighing, detection, and price prediction.

#### 4. Results and Discussions

In this section, we present an analysis of the outcomes achieved by our trained models, namely YOLOv5n and YOLOv7, as assessed on a validation dataset comprising 525 images containing a total of 1818 instances. Our evaluation process encompasses various performance metrics, including Precision, Recall, F1 score, mean Average Precision at IoU threshold 0.5 (mAP50), and mean Average Precision within the IoU range of 0.5 to 0.95 (mAP50-95). By employing these metrics, we thoroughly assess the effectiveness and accuracy of our models in object detection tasks, providing a comprehensive evaluation of their performance.

The precision metrics of two models, YOLOv5n and YOLOv7, were evaluated and demonstrated in Figures 14 and 15. These metrics provide important insights into the models' performance in accurately predicting object classes and minimizing false positives. The optimal weights for both models resulted in a remarkable precision score of 0.972 across all classes. Precision is a measure of how accurately the model predicts the positive classes, and a score of 0.972 indicates that the models accurately predicted over 97% of the true positive classes. In addition to high precision, the models also exhibited an extremely low number of false positives. Furthermore, the models' performance was compared to previous research [26,32,33]. The models outperformed previous work by achieving higher precision. This indicates that the models have improved upon existing methods and are more effective in accurately detecting and classifying objects in the weighing scales context. To provide a more detailed breakdown of the precision values for individual classes in YOLOv5n and YOLOv7, Tables 2 and 3 were presented, respectively. These tables provide a comprehensive view of how well each class is predicted by the models. By examining the precision values for individual classes, it is possible to identify any specific classes that may pose challenges for the models or classes that the models excel at detecting. We calculated precision metrics by employing standard equations given below:

$$P = \frac{TP}{TP + FP}$$

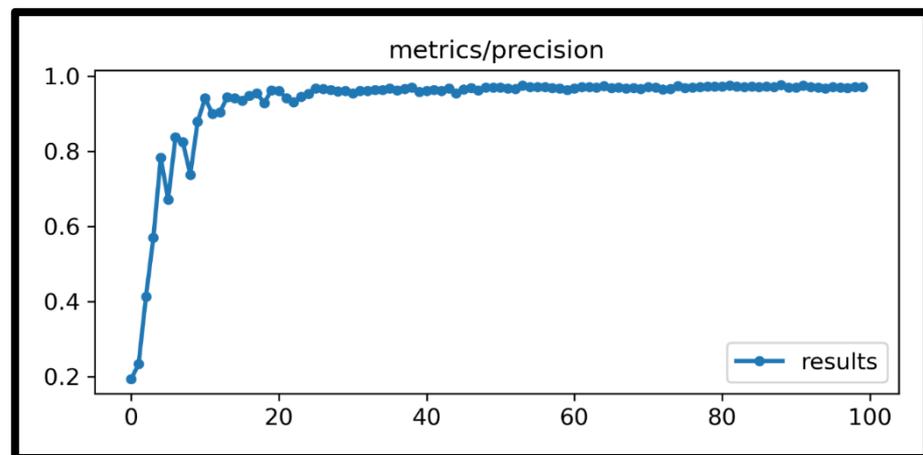


Figure 14. Precision metric graph of YOLOv5n.

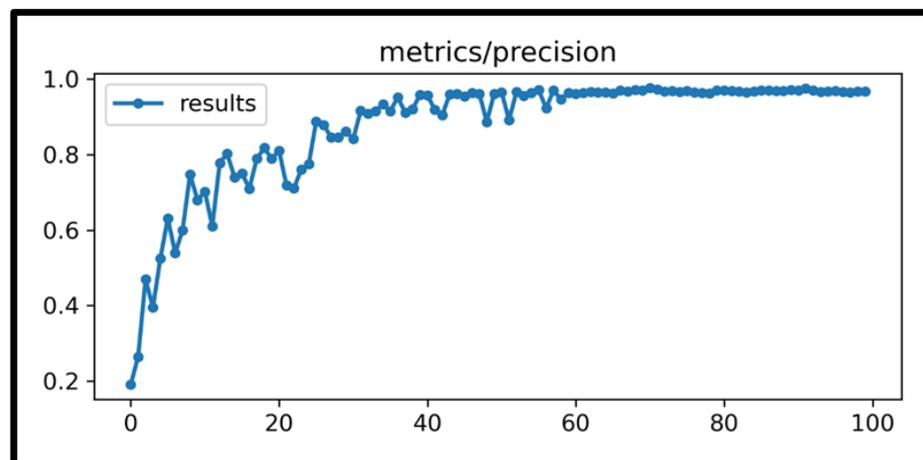


Figure 15. Precision metric graph of YOLOv7.

The recall metrics for two models, YOLOv5n and YOLOv7, were evaluated and compared. The results are depicted in Figures 16 and 17, respectively. Both models achieved high recall values, with YOLOv5n achieving a recall of 0.988 and YOLOv7 achieving a recall of 0.986 across all classes. Recall is a metric used to evaluate the performance of object detection models. It measures the ability of a model to correctly identify positive instances out of all the actual positive instances. Based on the provided recall values, it can be inferred that both versions of YOLO (YOLOv5n and YOLOv7) were able to identify over 98% of the true positive instances, indicating a high level of performance. Additionally, the results indicate that there were very few false negatives, meaning that the models successfully minimized the instances where positive objects were missed. Comparing the results with previous research [26,32,33], it is stated that the models achieved higher recall values. Furthermore, a detailed breakdown of the recall values for individual classes is provided in Table 2 for YOLOv5n and Table 3 for YOLOv7. These tables likely contain a list of different classes and their corresponding recall values, allowing for a more fine-grained analysis of the model's performance across specific object categories. The recall metrics were calculated using the given equations:

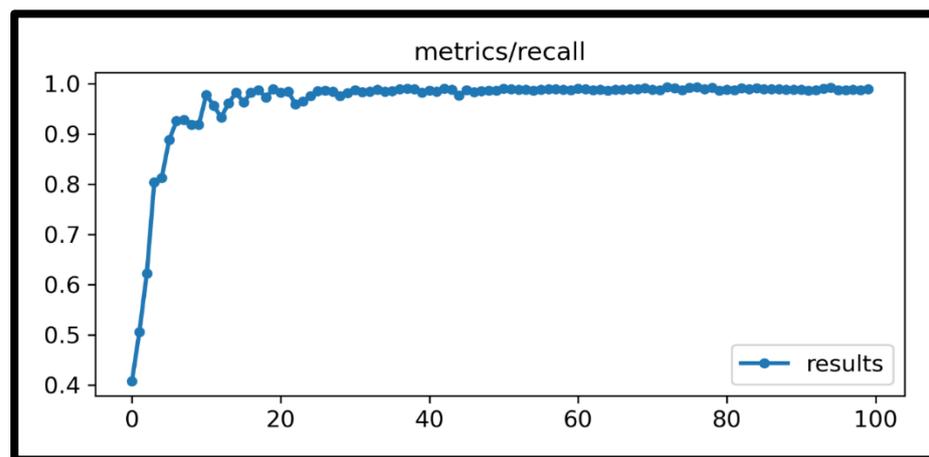
$$R = \frac{TP}{TP + FN}$$

**Table 2.** The metrics values of YOLOv5n.

Class	Images	Instances	Precision	Recall	F1 Score	mAP50	mAP50-95
All	525	1818	0.972	0.988	0.979	0.988	0.932
Potato	525	130	0.973	1	0.986	0.992	0.983
Tomato	525	254	0.964	0.984	0.973	0.992	0.988
Onion	525	151	0.977	1	0.988	0.995	0.973
Turnip	525	126	0.947	0.986	0.966	0.988	0.942
Chili	525	192	0.897	0.891	0.893	0.945	0.691
Garlic	525	155	0.975	1	0.987	0.991	0.921
Carrot	525	66	0.967	1	0.983	0.98	0.821
Cucumber	525	106	0.984	1	0.991	0.993	0.935
Apricot	525	171	0.998	1	0.998	0.995	0.995
Yam	525	136	0.997	1	0.998	0.995	0.946
Lemon	525	203	0.997	1	0.998	0.995	0.995
Plum	525	128	0.989	1	0.994	0.995	0.991

**Table 3.** The metrics values of YOLOv7.

Class	Images	Instances	Precision	Recall	F1 Score	mAP50	mAP50-95
All	525	1818	0.972	0.986	0.978	0.987	0.951
Potato	525	130	0.974	1	0.986	0.991	0.989
Tomato	525	254	0.951	1	0.974	0.993	0.988
Onion	525	151	0.998	1	0.998	0.995	0.984
Turnip	525	126	0.952	0.984	0.967	0.988	0.919
Chili	525	192	0.921	0.854	0.886	0.929	0.741
Garlic	525	155	0.975	0.998	0.986	0.994	0.923
Carrot	525	66	0.949	1	0.97	0.984	0.948
Cucumber	525	106	0.966	1	0.982	0.992	0.977
Apricot	525	171	0.996	1	0.997	0.995	0.995
Yam	525	136	0.999	1	0.999	0.995	0.962
Lemon	525	203	0.994	1	0.996	0.996	0.994
Plum	525	128	0.99	1	0.994	0.995	0.995



**Figure 16.** Recall metric graph of YOLOv5n.

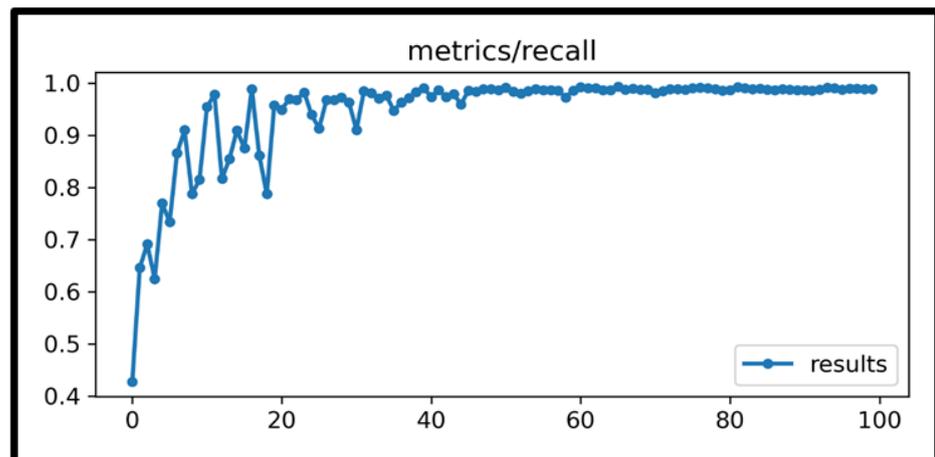


Figure 17. Recall metric graph of YOLOv7.

Both models attained F1 scores of 0.979 and 0.978, respectively, across all classes. This indicates that both versions of YOLO achieved a balanced combination of precision and recall, capturing a high proportion of true positive instances while minimizing false positives and false negatives. Furthermore, we present the F1 scores for individual classes in Tables 2 and 3 for YOLOv5n and YOLOv7, respectively. The F1 scores were calculated using the provided equations:

$$F1\ Score = \frac{2 \times (P \times R)}{P + R}$$

The evaluation of object detection models commonly employs two metrics: mean Average Precision at an IoU threshold of 0.5 (mAP50) and mean Average Precision within the IoU range of 0.5 to 0.95 (mAP50:95). These metrics provide an indication of the model's ability to accurately localize and classify objects with varying levels of IoU thresholds. Figures 18 and 19 depict the performance of two models, YOLOv5n and YOLOv7, in terms of mAP50 and mAP50:95. For YOLOv5n, the model achieved an mAP50 of 0.988 and an mAP50:95 of 0.932. On the other hand, YOLOv7 attained an mAP50 of 0.987 and an mAP50:95 of 0.951. These metrics were calculated using a validation dataset, which is separate from the training dataset. Typically, models tend to perform well on the training dataset due to the optimization of their parameters based on that specific data. However, in this case, both YOLOv5n and YOLOv7 demonstrate mAP values exceeding 0.98 on the validation dataset. This indicates that the models were able to generalize effectively to unseen data, which is a positive sign. When a model overfits the training data, it tends to perform exceptionally well on the training set but struggles to generalize to new, unseen data. However, since both models achieved high mAP values on the validation dataset, it suggests that they did not overfit the training data. Additionally, it is worth noting that the achieved mAP values of these models are higher compared to previous works [26,32,33], indicating an improvement in performance. Furthermore, the performance of the models for individual classes can be seen in Table 2 for YOLOv5n and Table 3 for YOLOv7. These tables provide the mAP50 and mAP50:95 values specifically for each class, allowing a more detailed analysis of the model's performance across different object categories. The calculations of mAP for different IOU threshold were conducted using the provided equations:

$$mAP = \frac{1}{n} \sum_{k=1}^{k=n} AP_k$$

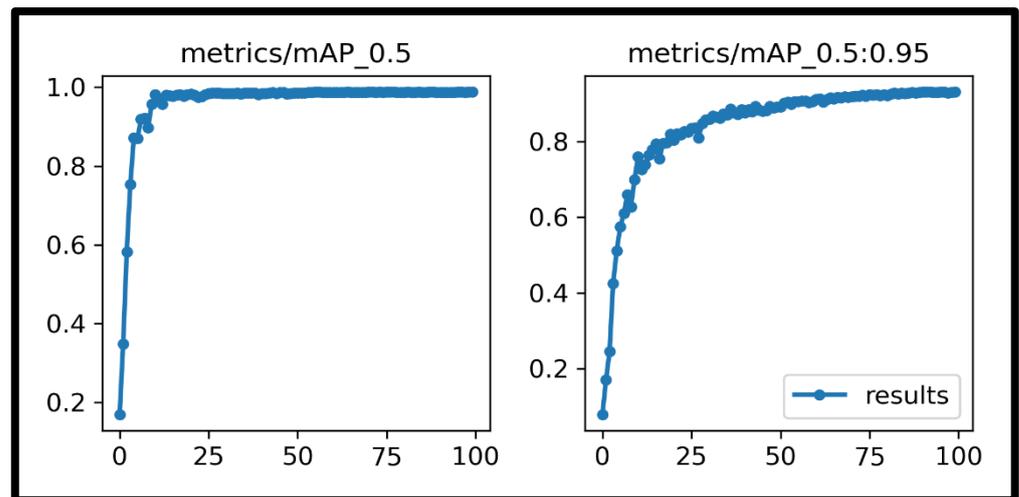


Figure 18. mAP of YOLOv5n.

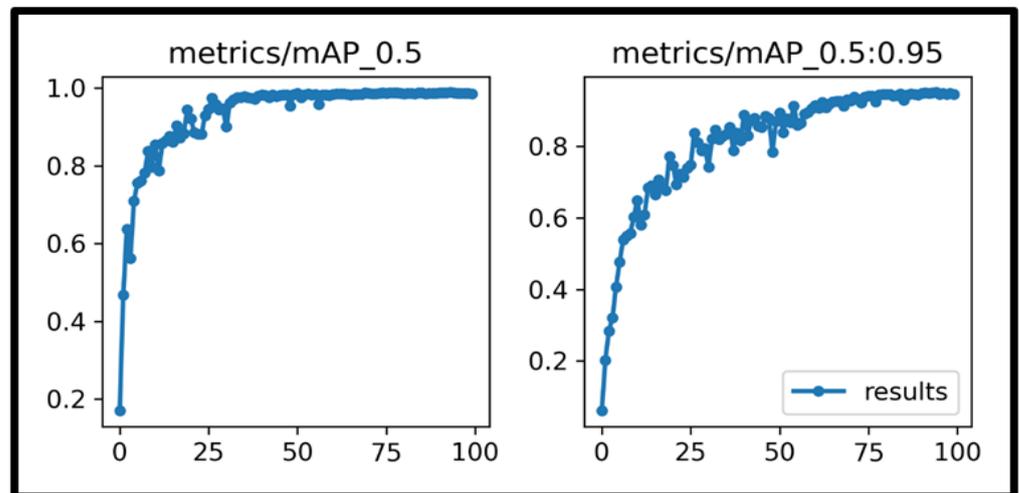


Figure 19. mAP of YOLOv7.

The performance metrics for the YOLOv5n and YOLOv7 models reveals their high accuracy and effectiveness in object detection tasks. Both models consistently achieved a precision of 0.972 for all classes, indicating their ability to accurately predict over 97% of true positive instances. The recall values were also impressive, with YOLOv5n achieving a recall of 0.988 and YOLOv7 achieving a recall of 0.986 for all classes, demonstrating their capability to identify a high proportion of true positive instances. Furthermore, the F1 scores for both models were equally impressive, with YOLOv5n and YOLOv7 achieving scores of 0.979 and 0.978, respectively, highlighting the model's balanced performance in combining precision and recall. Additionally, the mean Average Precision at IoU threshold 0.5 (mAP50) was 0.988 for YOLOv5n and 0.987 for YOLOv7, indicating their ability to accurately localize and classify objects as shown in Figures 20 and 21. The mean Average Precision within the IoU range of 0.5 to 0.95 (mAP50:95) was also remarkable, with YOLOv5n achieving a value of 0.932 and YOLOv7 achieving 0.951, further validating the model's robust performance across different IoU thresholds.

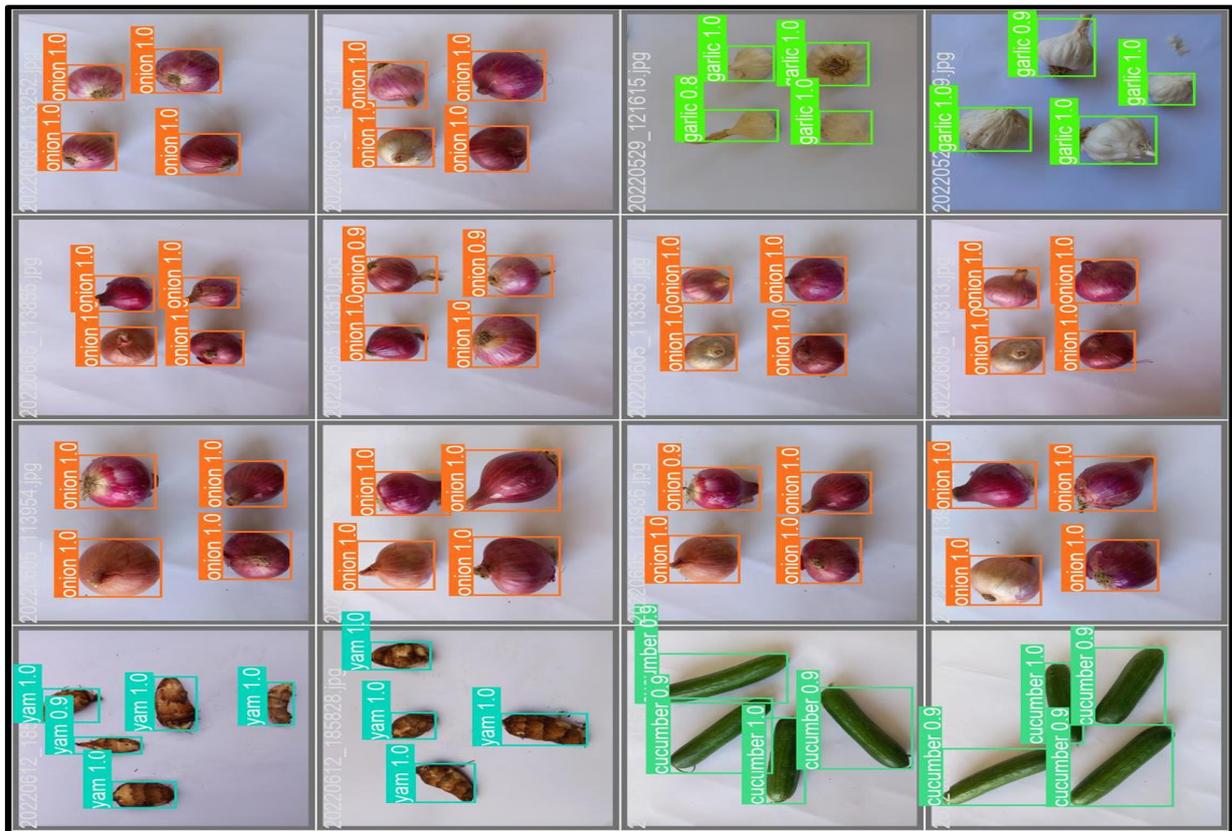


Figure 20. Results of YOLOv5n on validation batch.

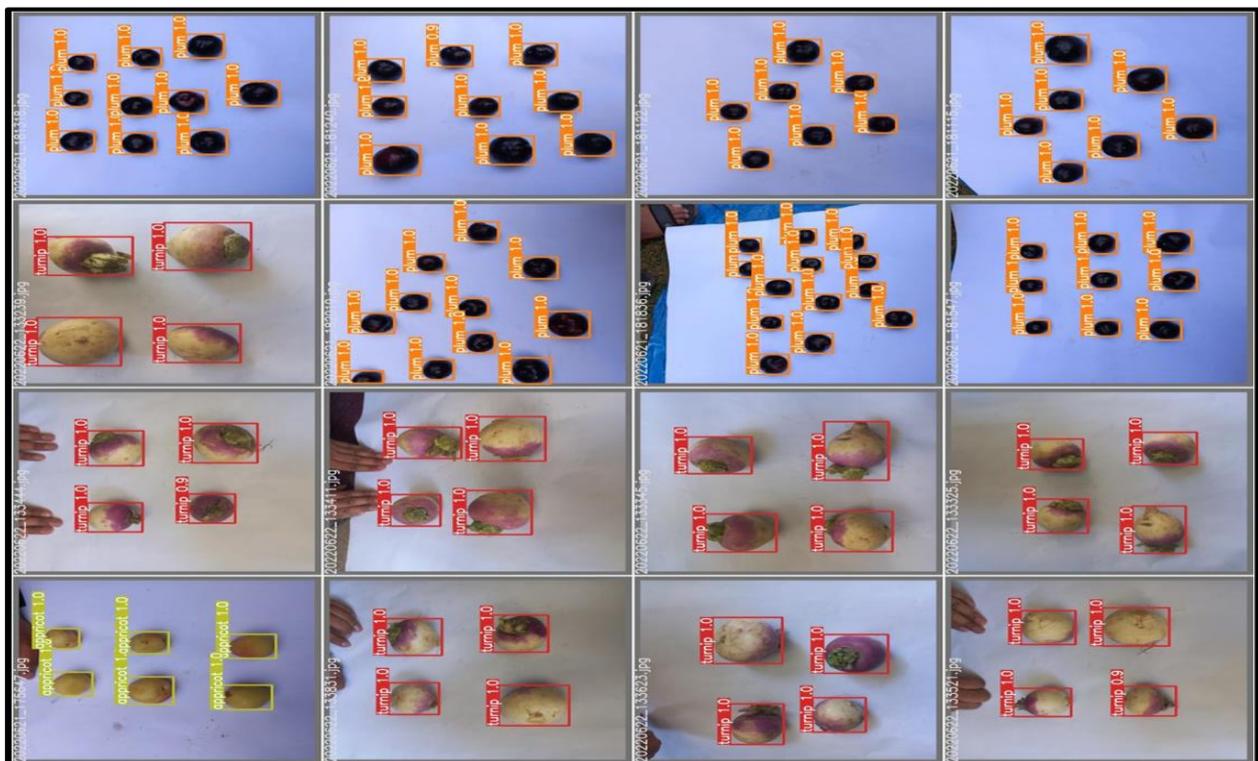
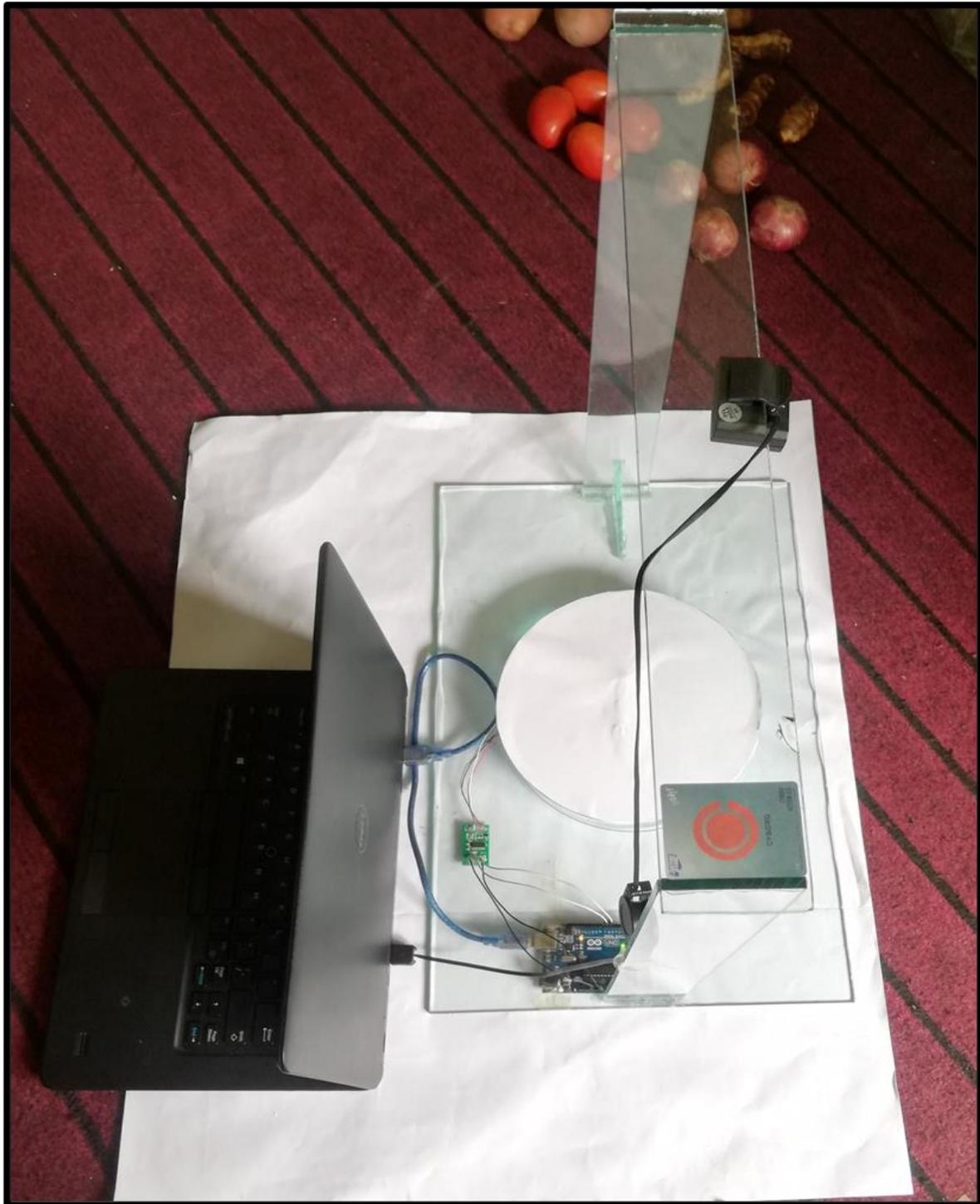


Figure 21. Results of YOLOv7 on validation batch.

The prototype presented in Figure 22 comprises a laptop, a camera, a glass frame designed for secure placement of the camera at its highest position, a load cell, an hx711 module for data acquisition from the load cell, and an Arduino Uno for controlling and processing the acquired data. This setup was constructed with the purpose of securely accommodating the camera within the glass frame, allowing for stable and reliable positioning.



**Figure 22.** Our Prototype.

Although the process of selling fruits and vegetables is automated by our intelligent IoT-based weighing scale, there are a few challenges associated with the utilization of

our intelligent IoT-based weighing scale, particularly in small-scale retail shops. Firstly, the current model of the smart scale is trained on a predefined dataset comprising only 12 classes, thus unrecognized fruits and vegetables will not be recognized by the scale. Gathering additional data and fine-tuning the model to encompass new classes is necessary to address this issue. Secondly, the scale requires specific hardware components, such as a Raspberry Pi or Jetson Nano, and an LCD screen for proper functioning, which may incur additional costs for small-scale retail shops. Lastly, the dynamic nature of product pricing requires manual input of updated prices into the GUI of each scale, which can be time-consuming for shopkeepers managing multiple scales. Addressing these challenges would enhance the scale's effectiveness and reliability.

## 5. Conclusions

In conclusion, the proposed intelligent weighing scale system for small-scale fruits and vegetables shops in Pakistan holds great potential to revolutionize the retail process and improve efficiency in this sector. By incorporating deep learning algorithms, and weighing sensors, the system automates the detection, recognition, weighing, and pricing of various fruits and vegetables. The use of deep learning algorithms, such as YOLOv5n and YOLOv7, enables real-time and accurate object detection and recognition, eliminating the need for manual entry of prices for each item. This not only saves time but also reduces labor requirements, allowing shopkeepers to serve multiple customers simultaneously. The graphical user interface (GUI) provides a user-friendly platform for adding items to the bill and displaying their prices based on their weights. By modernizing small-scale retail operations, especially in low-income segments of the population, this system addresses the challenges of labor-intensive and time-consuming pricing methods. It empowers shopkeepers to serve customers faster, improves accuracy in weighing and pricing, and reduces human error. However, further research and real-world implementation are necessary to evaluate the system's performance, scalability, and impact on the industry. Additionally, continuous refinement and updates to the deep learning models and algorithms can enhance the system's accuracy and adaptability to different fruits and vegetables. Overall, the intelligent weighing scale system has the potential to transform the small-scale retail sector in Pakistan, providing a more efficient, automated, and customer-friendly shopping experience while benefiting both shopkeepers and customers alike.

**Author Contributions:** Conceptualization, A.Z. and I.U.H.; methodology, T.A., G.H., S.R., M.A. and H.G.M.; software, A.Z. and I.U.H.; validation, T.A. and M.A.; formal analysis, A.Z., G.H. and Y.Y.G.; investigation, A.Z., I.U.H., T.A., G.H. and S.R.; resources, T.A. and M.A.; data curation, G.H. and Y.Y.G.; writing—original draft, A.Z., I.U.H., T.A. and S.R.; writing—review and editing, G.H., M.A., Y.Y.G. and H.G.M.; visualization, S.R. and I.U.H.; supervision, G.H. and H.G.M.; project administration, Y.Y.G. and H.G.M.; funding acquisition, H.G.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023TR140), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Chaudhry, I.S.; Malik, S. The Impact of Socioeconomic and Demographic Variables on Poverty: A Village Study. *Lahore J. Econ.* **2009**, *14*, 39–68. [[CrossRef](#)]
2. Zhao, Z.Q.; Zheng, P.; Xu, S.T.; Wu, X. Object Detection with Deep Learning: A Review. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3212–3232. [[CrossRef](#)]

3. Murthy, C.B.; Hashmi, M.F.; Bokde, N.D.; Geem, Z.W. Investigations of object detection in images/videos using various deep learning techniques and embedded platforms—A comprehensive review. *Appl. Sci.* **2020**, *10*, 3280. [CrossRef]
4. Forsyth, D. Object detection with discriminatively trained part-based models. *Computer* **2014**, *47*, 6–7. [CrossRef]
5. Hayat, S.; Kun, S.; Tengtao, Z.; Yu, Y.; Tu, T.; Du, Y. A Deep Learning Framework Using Convolutional Neural Network for Multi-Class Object Recognition. In Proceedings of the 2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC), Chongqing, China, 27–29 June 2018; pp. 194–198. [CrossRef]
6. Derpanis, K. The Harris Corner Detector. York Univ., No. March, pp. 2–3, 2004. Available online: [http://windage.googlecode.com/svn/trunk/Mindmap/Tracking/Papers/\[2004\]](http://windage.googlecode.com/svn/trunk/Mindmap/Tracking/Papers/[2004]) (accessed on 20 March 2023).
7. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
8. Bay, H.; Tuytelaars, T.; Van Gool, L. LNCS 3951—SURF: Speeded Up Robust Features. *Comput. Vision—ECCV 2006*, pp. 404–417. Available online: [http://link.springer.com/chapter/10.1007/11744023\\_32](http://link.springer.com/chapter/10.1007/11744023_32) (accessed on 20 March 2023).
9. Bay, H.; Ess, A.; Tuytelaars, T.; Van Gool, L. Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.* **2008**, *110*, 346–359. [CrossRef]
10. Saxena, A. An Introduction to Convolutional Neural Networks. *Int. J. Res. Appl. Sci. Eng. Technol.* **2022**, *10*, 943–947. [CrossRef]
11. Li, Z.; Liu, F.; Yang, W.; Peng, S.; Zhou, J. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 6999–7019. [CrossRef]
12. Alzubaidi, L.; Zhang, J.; Humaidi, A.J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaria, J.; Fadhel, M.A.; Al-Amidie, M.; Farhan, L. *Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions*; Springer International Publishing: Berlin/Heidelberg, Germany, 2021; Volume 8. [CrossRef]
13. Ghimire, A.; Werghi, N.; Javed, S.; Dias, J. Real-Time Face Recognition System. *arXiv* **2022**, arXiv:2204.08978.
14. Rangari, A.P.; Chouthmol, A.R.; Kadadas, C.; Pal, P.; Singh, S.K. Deep Learning based smart traffic light system using Image Processing with YOLO v7. In Proceedings of the 2022 4th International Conference on Circuits, Control, Communication and Computing (I4C), Bangalore, India, 21–23 December 2022; pp. 129–132. [CrossRef]
15. Sangeetha, V.; Prasad, K.J.R. Syntheses of novel derivatives of 2-acetylfuro[2,3-a]carbazoles, benzo[1,2-b]-1,4-thiazepino[2,3-a]carbazoles and 1-acetyloxycarbazole-2- carbaldehydes. *Indian J. Chem. Sect. B Org. Med. Chem.* **2006**, *45*, 1951–1954. [CrossRef]
16. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448. [CrossRef]
17. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef]
18. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
19. Jiang, P.; Ergu, D.; Liu, F.; Cai, Y.; Ma, B. A Review of Yolo Algorithm Developments. *Procedia Comput. Sci.* **2022**, *199*, 1066–1073. [CrossRef]
20. Unal, H.B.; Vural, E.; Savas, B.K.; Becerikli, Y. Fruit Recognition and Classification with Deep Learning Support on Embedded System (fruitnet). In Proceedings of the 2020 Innovations in Intelligent Systems and Applications Conference (ASYU), Istanbul, Turkey, 15–17 October 2020. [CrossRef]
21. Sakai, Y.; Oda, T.; Ikeda, M.; Barolli, L. A vegetable category recognition system using deep neural network. In Proceedings of the 2016 10th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), Fukuoka, Japan, 6–8 July 2016; pp. 189–192. [CrossRef]
22. Ikeda, M.; Oda, T.; Barolli, L. A vegetable category recognition system: A comparison study for caffe and Chainer DNN frameworks. *Soft Comput.* **2019**, *23*, 3129–3136. [CrossRef]
23. Ahmed, M.I.; Mahmud Mamun, S.; Zaman Asif, A.U. DCNN-Based Vegetable Image Classification Using Transfer Learning: A Comparative Study. In Proceedings of the 2021 5th International Conference on Computer, Communication and Signal Processing (ICCCSP), Chennai, India, 24–25 May 2021; pp. 235–243. [CrossRef]
24. Gulzar, Y. Fruit Image Classification Model Based on MobileNetV2 with Deep Transfer Learning Technique. *Sustainability* **2023**, *15*, 1906. [CrossRef]
25. Albarrak, K.; Gulzar, Y.; Hamid, Y.; Mehmood, A.; Soomro, A.B. A Deep Learning-Based Model for Date Fruit Classification. *Sustainability* **2022**, *14*, 6339. [CrossRef]
26. Ragesh, N.; Giridhar, B.; Lingeswaran, D.; Siddharth, P.; Peeyush, K.P. Deep learning based automated billing cart. In Proceedings of the 2019 International Conference on Communication and Signal Processing (ICCCSP), Chennai, India, 4–6 April 2019; pp. 779–782. [CrossRef]
27. Sachin, C.; Manasa, N.; Sharma, V.; Kumar, N.A.A. Vegetable Classification Using You Only Look Once Algorithm. In Proceedings of the 2019 International Conference on Cutting-Edge Technologies in Engineering (ICoN-CuTE), Uttar Pradesh, India, 14–16 November 2019; pp. 101–107. [CrossRef]
28. Latha, R.; Sreekanth, G.; Rajadevi, R.; Nivetha, S.; Kumar, K.; Akash, V.; Bhuvanesh, S.; Anbarasu, P. Fruits and Vegetables Recognition using YOLO. In Proceedings of the 2022 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 25–27 January 2022; pp. 2–7. [CrossRef]

29. Wan, S.; Goudos, S. Faster R-CNN for multi-class fruit detection using a robotic vision system. *Comput. Netw.* **2019**, *168*, 107036. [[CrossRef](#)]
30. Šumarac, J.; Kljajić, J.; Rodić, A. A Fruit Detection Algorithm for a Plum Harvesting Robot Based on Improved YOLOv7. In *International Conference on Robotics in Alpe-Adria Danube Region 2023 May 27*; Springer Nature: Cham, Switzerland, 2023; pp. 442–450. [[CrossRef](#)]
31. Tang, Y.; Zhou, H.; Wang, H.; Zhang, Y. Fruit detection and positioning technology for a Camellia oleifera C. Abel orchard based on improved YOLOv4-tiny model and binocular stereo vision. *Expert Syst. Appl.* **2023**, *211*, 118573. [[CrossRef](#)]
32. Chidella, N.; Reddy, N.K.; Reddy, N.S.D.; Mohan, M.; Sengupta, J. Intelligent Billing system using Object Detection. In *Proceedings of the 2022 1st International Conference on the Paradigm Shifts in Communication, Embedded Systems, Machine Learning and Signal Processing (PEMS)*, Nagpur, India, 6–7 May 2022; pp. 11–15. [[CrossRef](#)]
33. Wang, B.; Xie, Y.; Duan, X. An IoT Based Fruit and Vegetable Sales System. In *Proceedings of the 2021 5th International Conference on Cloud and Big Data Computing (ICCBDC)*, Liverpool, UK, 13–15 August 2021; pp. 109–115. [[CrossRef](#)]
34. Nandanwar, V.G.; Kashif, M.; Ankushe, R.S. Portable Weight Measuring Instrument. In *Proceedings of the 2017 International Conference on Recent Trends in Electrical, Electronics and Computing Technologies (ICRTEECT)*, Warangal, India, 30–31 July 2017; pp. 44–48. [[CrossRef](#)]
35. Wang, C.Y.; Liao, H.Y.M.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A new backbone that can enhance learning capability of CNN. In *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Seattle, WA, USA, 14–19 June 2020; pp. 1571–1580. [[CrossRef](#)]
36. Wang, K.; Liew, J.H.; Zou, Y.; Zhou, D.; Feng, J. PANet: Few-shot image semantic segmentation with prototype alignment. In *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9196–9205. [[CrossRef](#)]
37. Zhai, N. Detection using Yolov5n and Yolov5s with small balls. *Comput. Sci.* **2022**, *12168*, 428–432. [[CrossRef](#)]
38. Xu, R.; Lin, H.; Lu, K.; Cao, L.; Liu, Y. A forest fire detection system based on ensemble learning. *Forests* **2021**, *12*, 217. [[CrossRef](#)]
39. Available online: <https://github.com/ultralytics/yolov5/releases> (accessed on 20 March 2023).
40. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. *arXiv* **2022**, arXiv:2207.02696.
41. Qiu, Y.; Lu, Y.; Wang, Y.; Jiang, H. IDOD-YOLOV7: Image-Dehazing YOLOV7 for Object Detection in Low-Light Foggy Traffic Environments. *Sensors* **2023**, *23*, 1347. [[CrossRef](#)] [[PubMed](#)]
42. Hussain, M.; Al-Aqrabi, H.; Munawar, M.; Hill, R.; Alsboui, T. Domain Feature Mapping with YOLOv7 for Automated Edge-Based Pallet Racking Inspections. *Sensors* **2022**, *22*, 6927. [[CrossRef](#)] [[PubMed](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.