ISTANBUL TECHNICAL UNIVERSITY ★ FACULTY OF MANAGEMENT

IMAGE DETECTION USING DEEP LEARNING FOR PAYMENT SYSTEM IMPROVEMENT

INDUSTRIAL ENGINEERING DESIGN II

Begüm Nur OKUR
Hilmi ÖZTÜRK

Department of Industrial Engineering

Thesis Advisor: Dr. Mehmet Ali ERGÜN

JUNE 2024

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ İŞLETME FAKÜLTESİ

ÖDEME SİSTEMLERİNİN İYİLEŞTİRİLMESİ İÇİN DERİN ÖĞRENMEYİ KULLANARAK GÖRÜNTÜ ALGILAMA UYGULAMASI

ENDÜSTRİ MÜHENDİSLİĞİ TASARIMI II

Begüm Nur OKUR
Hilmi ÖZTÜRK

Endüstri Mühendisliği Bölümü

Danışman: Dr. Mehmet Ali ERGÜN

HAZİRAN 2024

To my family and my best friends,

- Begüm

To my family,

-Hilmi

TABLE OF CONTENTS

Page

ABBREVIATIONS

AI	**:** Artificial Intelligence
CNN	**:** Convolutional Neural Network
CT	**:** Computed Tomography
DBN	**:** Deep Belief Networks
DL	**:** Deep Learning
DNN	**:** Deep Neural Networks
FN	**:** False Negative
FP	**:** False Positive
LSTM	**:** Long Short-Term Memory Network
ML	**:** Machine Learning
QR	**:** Quick Response
RCNN	**:** Region-Based Convolutional Neural Network
RFID	**:** Radio Frequency Identification
RNN	**:** Recurrent Neural Network
TN	**:** True Negative
TP	**:** True Positive
YOLO	**:** You Only Look Once

LIST OF FIGURES

Page

LIST OF TABLES

IMAGE DETECTION USING DEEP LEARNING FOR SYSTEM IMPROVEMENT

SUMMARY

In restaurants operating with the open buffet method, customers sit in a line and place the food, drinks and other products they want on a tray. At the end of the line, a staff member calculates the cost of the customer's purchases and payment is made. In this study, it is planned to automate the payment section in order to reduce labor costs and make the system more effective. At the end of the row, a photo of the tray will be taken with a camera, and the products in this photo are detected by the deep learning model. Datasets containing food and beverage images were used to develop this model. These datasets will be used for training the model after going through various data processing steps and were also used as test data to measure the success of the model. The model was evaluated using a range of performance metrics. Based on the errors in the test data, improvements were made to the dataset. The model that yielded the most optimal results was employed to detect objects on the tray. The trained model was utilized in the function that detects the objects on the tray and outputs total price information using the Python language. Furthermore, the purchased products in the tray and information about this purchase were recorded. The classical system with a cashier and the system that detects images were simulated according to customer density. The most cost-effective and customer-satisfactory system has been proposed according to customer density.

ÖDEME SİSTEMLERİNİN İYİLEŞTİRİLMESİ İÇİN DERİN ÖĞRENMEYİ KULLANARAK GÖRÜNTÜ ALGILAMA UYGULAMASI

ÖZET

Açık büfe yöntemiyle işeyen restoranlarda müşteriler bir sıranın başına geçer ve tepsiye istedikleri yemek, içecek ve diğer ürünleri koyarlar. Sıranın sonunda ise bir personel, müşterinin aldıklarının ücretini hesaplar ve ödeme gerçekleşir. Bu çalışmada iş gücü maliyetini düşürmek ve sistemi daha efektif hale getirmek için ödeme bölümünün otomatik hale gelmesi planlanmıştır. Sıranın sonunda bir kamera ile tepsi fotoğrafı çekilir, bu fotoğrafta bulunan ürünler ise derin öğrenme modeli ile tespit edilir. Bu modeli geliştirmek için yiyecek ve içecek resimlerinin bulunduğu verisetleri kullanılmıştır. Bu verisetleri çeşitli veri işleme adımlarından geçildikten sonra modelin eğitimi için kullanılacak ve modelin başarısını ölçmek için test verisi olarak da kullanılmıştır. Çeşitli performans metrikleri ile model değerlendirilmiştir. Test verisindeki hatalardan yola çıkarak veristerinde iyileştirmeler yapılmıştır. En iyi sonucu veren model ise tepsi üzerinde nesne tespiti yapmak için kullanılmıştır. Eğitilen model, python dili kullanılarak tepsideki nesneleri tespit edip toplam fiyat bilgisi çıktısı veren fonksiyon içinde kullanılmıştır. Ayrıca tepsideki alınan ürünler ve bu satın alma işlemi ile ilgili bilgiler kaydedilir. Kasiyerin olduğu klasik sistem ile görüntü tespiti yapan sistem müşteri yoğunluğuna göre simüle edilmiştir. Yoğunluğa göre maliyet ve müşteri memnuiyeti açısından en etkili sistem önerilmiştir.

1. INTRODUCTION

1.1 Purpose of Thesis

In the aftermath of the pandemic and increase in digitalization, there has been a notable surge in contactless services in our daily routines. The term "untact," coined in Korea in the post-pandemic era, is a fusion of "un" and "contact," encapsulating interactions that do not involve direct face-to-face contact. Many businesses are enthusiastically adopting untact services, utilizing autonomous payment systems, kiosks, and self-service counters. These technologies are being seamlessly integrated into conventional retail settings (An et al., 2020).

During this study, we seek to provide a solution for reducing wait times at self-service restaurants through the use of computer vision technology. We will use Deep Learning (DL), a subset of Artificial Intelligence (AI), which is regarded as one of the most influential applications among emerging technologies. Minimizing the wait time at the checkout counter post-food purchase holds significant value in our fast-paced contemporary society, especially for individuals with tight schedules. The objective is to mitigate potential human errors within the system with cashiers and streamline workplaces by digitizing processes, thereby curbing the escalating labor costs.

1.2 Literature Review

This literature review explores the intersection of deep learning in object detection, particularly in the context of food and beverages, and the automation of meal pricing in self-service restaurants. By examining the current state of research and industry practices, the review aims to identify gaps and opportunities, providing a foundation for the subsequent development and implementation of an automated pricing system using food images. Starting with an overview of artificial intelligence, the review delves into applications of machine learning and deep learning. It then covers computer vision, examining its applications in object detection, image processing, and image analysis, with a specific focus on food recognition. In the final part, we review restaurant payment system automation, highlighting the crucial role of image detection in related works.

1.2.1 Artificial intelligence

Artificial intelligence is the backbone of our system, that’s why it is crucial to define and understand its connections with other concepts. The concept of artificial intelligence traces its roots back to the 20th century when the renowned mathematician Alan Turing pondered the possibility of machines thinking like humans (Turing, 2009). However, it wasn't until the 21st century that artificial intelligence significantly eased our daily lives, expanding its applications. Artificial Intelligence (AI) refers to the development of computer systems with various sub-algorithms that can perform tasks such as learning, causality and problem solving that typically associated with human intelligence. Today, Artificial Intelligence is a universal field because its main function goes beyond learning and perception, and covers very specific subfields such as playing chess, proving mathematical theorems, writing poetry, driving on a crowded street and diagnosing diseases (Russell and Norvig, 2010). Even though these subfields have increased substantially, AI’s potential has still not been fully revealed. Accordingly, AI is still classified as emerging technology because it still meets the requirement that "the definition, development and application areas of emerging technologies are still rapidly expanding, and its potential is still largely unrealized" (Kufeoglu, 2022).

Advancements in AI are closely linked to progress in other technologies such as big data and cloud computing. The interconnected nature of these fields contributes to innovations in various domains. Furthermore, AI will become more understandable by defining the concepts of machine learning, artificial neural networks and deep learning that it encompasses.

In order to make progress in this field, it is important to use concepts and terms correctly. Although machine learning, artificial intelligence and deep learning are terms used interchangeably from time to time, the truth is that Machine Learning (ML) is a subset of Artificial Intelligence (AI); It encompasses approaches that enable machines to learn from data without explicit programming, dynamically adjusting and minimizing errors to make decisions based on processed data, and Deep Learning (DL), a subset of ML, is used to extract patterns from data, mimicking the architecture of biological neural networks and DL includes computational models that use the layers (input, hidden and output) in neural networks and reflect the brain's comparison and categorization process, and it is a subset of machine learning (Jakhar and Kaur, 2020).

1.2.2 Deep learning applications and architectures

Deep learning is a subset of machine learning that includes training multiple-layer artificial neural networks to learn and make predictions or choices from complicated, unstructured data. The most common form of machine learning, whether deep learning or not, is supervised learning, trained on labeled data and aiming to have the machine adjust internal parameters to produce output scores for each category; in a deep learning system with potentially hundreds of millions of tunable weights, the learning algorithm is trained while it calculates a gradient vector to show how the error will change with small adjustments, and the weights are then updated in the opposite direction of the gradient, optimizing the input-output function of the machine through iterative training on a large dataset (LeCun et al., 2015). To summarize, ongoing advances in deep learning methodologies, powered by extensive research and the collaborative efforts of the global scientific community, hold the promise of revolutionizing how machines perceive, process and interpret complex information, paving the way for transformative applications in a wide range of fields.

Deep learning finds application in a wide range of areas, from autonomous vehicles to handwritten character recognition, from signature verification to voice processing (Yapici et al., 2019). When utilized with appropriate architecture and dataset, deep learning outperforms traditional machine learning methods and various architectures have been devised over time to tackle diverse problems. Previously mentioned use cases are increasing over time with the advancements in DL methods. Especially, CNNs (Convolutional Neural Networks), RNNs (Recurrent Neural Networks), and LSTMs (Long Short-Term Memory networks), stand out in processing multimedia content to solve complex problems. As computing technology progresses, CNNs are providing opportunities for image processing, image classification, restoration of images and exhibiting their success in generating outcomes (Jiang et al., 2022). On the other hand, RNNs and LSTMs are tailored for processing sequential data, such as video event recognition, speech recognition, and image-to-text translation (Khan et al., 2019). The development of DL algorithms, CNNs or deep autoencoders, not only impacts typical tasks like object classification but is also efficient in other related tasks such as object detection, localization, tracking, or image segmentation, and these advancements are facilitated by the competitive environment that drives researchers worldwide to develop various technologies for implementing segmentation across various domains (Ghosh et al., 2019).

Other DL architectures include Deep Neural Networks, capable of demonstrating complex non-linear relationships; Deep Belief Networks, employing layered graphical models for hierarchical feature representation; Recurrent Neural Networks, showcasing dynamic temporal behavior with loops; Boltzmann Machines, networks of symmetrically connected units making stochastic decisions; Restricted Boltzmann Machines, with bipartite visible and hidden layers; Deep Autoencoders, neural networks for dimensionality reduction; and Convolutional Neural Networks, specialized in image processing by learning feature extraction filters (Dixit et al., 2018). DL's maturity has also led to encouraging outcomes in Food Image Segmentation, a previously challenging task (Mohanty et al., 2022). DL in the food industry helps businesses and large-scale food factories by optimizing processes, decreasing human errors, lowering waste, reducing costs, boosting consumer satisfaction, and allowing personalized orders (Kumar et al., 2021).

As opposed to conventional food category recognition techniques utilized for many years, Convolutional Neural Networks (CNNs) separate themselves with a significantly increased number of consecutive layers, enabling the discovery of higher-level features and hierarchical connections; however, the escalating depth in neural networks can require extended periods of training and create higher computational costs, but many research studies have shown that CNNs with increased depth can significantly improve image processing in the realm of food category recognition (Zhang et al., 2023).

In conclusion, the continuous advancement and diversification of deep learning architectures not only improve their ability to solve problem capabilities but also pave the way for a transformative era in machine learning, offering solutions that outperform traditional methods across a wide range of challenges and applications.

1.2.3 Computer vision

Computer vision is a branch of deep learning that focuses on instructing robots in interpreting and comprehending visual data from the outside environment. The goal of computer vision is to mimic human vision skills, allowing computers to recognize objects, find patterns, and derive useful insights from visual inputs. Image and video identification, object detection, facial recognition, medical image analysis, autonomous cars and augmented reality are some of the applications of computer vision.

1.2.3.1 Object detection with deep learning

In order to fully comprehend an image, it is essential to accurately estimate the fundamental concepts and positions of things present in each image, in addition to focusing on classifying various images, object detection is the name of this task, which typically consists of several smaller tasks including face, pedestrian, and skeleton detection (Felzenszwalb et al., 2009). These tasks collectively contribute to a more comprehensive understanding of images and are crucial for various applications, including image classification, human behavior analysis, face recognition, and autonomous driving (Zhao et al., 2019). Among the DL models used for object detection, there are models such as YOLO (You Only Look Once), which can evaluate an image in a single process and work quickly, and RCNN (Region-based Convolutional Neural Network), which works with a region recommendation step.

You Only Look Once (YOLO)

You Only Look Once (YOLO) is a popular and commonly used algorithm (Sultana et al., 2020). YOLO is well-known for its object detection feature, which was introduced in the first edition (Zhiqiang & Jun, 2017). The YOLO object detection method differentiates itself with its small model size, rapid computational speed, and straightforward structure, allowing direct output of both bounding box position and category through the neural network; its efficiency is derived from the capability of analysing images directly, allowing real-time video detection; the global image approach of YOLO encodes extensive data, minimizing background detection errors; and its robust generalization ability empowers the transfer of learned features across diverse domains (Jiang et al., 2022). From YOLO-v1 (2015) to the most recent YOLO-v8 (2023), the YOLO family has grown and stayed at the top of object detection over the years, bringing architectural improvements and performance enhancements in computer vision applications (Hussain, 2023). According to Ramla et al. (2022), Yolo algorithm is a deep learning model for object detection based on Convolutional Neural Networks (CNN). It works on multiple boundaries in the image. However, YOLO optimizes object detection performance by using only one CNN. R-CNN, on the other hand, determines the regions in which the target object may occur and then runs a CNN for each possible region. However, YOLO uses a single CNN for these possible regions. That's why YOLO produces much faster results than R-CNN. To find the coordinate and class of the target object, Yolo first divides the main image into n*n grids. Figure 1.1 shows this division. The number n varies depending on the value entered. Each grid first checks whether the target object exists in its area. If there is a target, it checks whether its center is within its own grid. If its center is also within the grid, it is responsible for finding 3 things. These are the height of the object, its length and the class it belongs to.

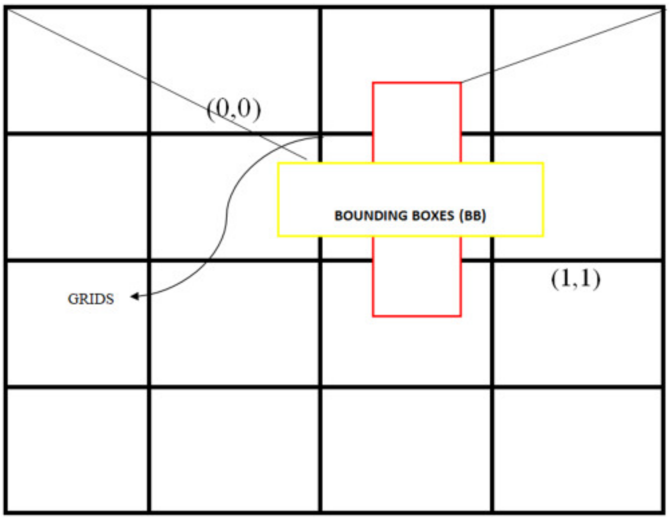


Figure 1.1: Division of 4*4 Grid.

Region-based Convolutional Neural Network (RCNN)

RCNN is an object detection model known by the abbreviation "Region-based Convolutional Neural Network". The RCNN (Region-based Convolutional Neural Network) model initiates the object detection process by identifying candidate object regions through a "region proposal" step, estimating potential object locations, followed by the utilization of a convolutional network to individually processing these regions, involving scaling, classification, and precise localization of objects (Girshick et al., 2014). RCNN employs a selective region recommendation to identify potential object regions in an image, which are subsequently processed individually by a pre-trained convolutional neural network, enabling more accurate and effective object detection (Li et al., 2022). This model marked a significant advance in object detection and served as the foundation for later models such as Fast RCNN and Faster RCNN (Girshick, 2015).

In conclusion, object detection plays a pivotal role in understanding images comprehensively, and deep learning models such as YOLO and RCNN have significantly contributed to this field, with YOLO's real-time capabilities and efficient global image approach, and RCNN's region-based convolutional approach that has paved the way for subsequent advancements in object detection models.

1.2.3.2 Image preprocessing

One of the image preprocessing operations is data augmentation. According to Beddiar et al. (2023), 5 data augmentation techniques can be mentioned. These can be said as sharpening, scaling, blurring, color differentiation, and changing the direction of the object in the picture. According to Sen and others (2023), noise in the picture is image distortion caused by inconsistent change in contrasts . While noise in the picture does not cause difficulty for the human brain to understand the whole picture, it can be a factor that reduces the correct prediction score in DL algorithms. According to the study by Beddiar et al., the original images and noise-added images of the images obtained from CT scans were used to detect COVID-19 patients. It has been observed that adding noise to images increases the success of the DL algorithm by 2.96% compared to not adding noise.

1.2.3.3 Image processing and image analysis

In the core of computer vision there are two important concepts which are image processing and image analysis. Image processing and image analysis are two distinct concepts that can be defined as follows: Image processing is a set of image operations that improve an image's quality by removing flaws such as geometric distortion, wrong focus, excessive noise, irregular lighting and camera movement but image analysis, on the other hand, is the process of distinguishing items from the background and producing quantitative information that is used in subsequent control systems for decision making and both of these ideas require a series of steps that can be broadly classified into three levels: low-level processing, middle-level processing, and high-level processing (Brosnan and Sun, 2004).

1.2.3.4 Detecting food image using deep learning based algorithms

Food is an indispensable part of human life, as well as an important part of socializing in our daily lives. Computer vision use-cases are growing and brings innovations in this field of our lives. The work, which creates recipes based on the food pictures, is one of the milestones in this regard and an exemplary use-case. The evolution of object recognition and scene classification, fueled by the emergence of extensive labeled datasets has redefined image processing and this progress extends to specialized areas such as image segmentation, with the introduction of datasets like Recipe1M+ potentially marking a significant advancement in the field (Marín et al., 2021). In another paper, the focus is on the introduction of the IndianFood-7 Dataset, designed to augment the capabilities of computer vision models in discerning popular Indian culinary items, with this dataset encompassing over 800 annotated images representing seven diverse Indian dishes emerging as a pivotal resource for both training and evaluating object detection models, and the paper undertaking a comprehensive comparative analysis of advanced object detection models, specifically YOLOR and YOLOv5, systematically evaluating their efficacy in recognizing and precisely localizing Indian food items (Agarwal et al., 2022). Food trays also an important area for our work since we will focus on their images. However, food tray analysis presents several obstacles, including the inclusion of several foods on the same placemat, different foods served in a single dish, distortions in vision and lighting moves caused by shadowy areas, and the existence of non-food items on the surface of the tray (Aguilar et al., 2018).

1.2.3.5 Model evaluation metrics

Various metrics are used to evaluate the performance of these models. One of them is Intersection over Union. According to Chumachenko and others (2022), Intersection over Union (IoU) measures the success or failure of localisation. The IOU score represents the overlap ratio between two bounding boxes.. In other words, it can be expressed as the ratio of the intersection of two boxes to their union.The acceptable range for IoU is between 0.5 and 1. Figure 1.2 shows a visualization of the IoU calculation for a model that detects sheep as its target object.The green rectangular box indicates the location of the targeted object, which is the actual value. The red box shows the predicted location.

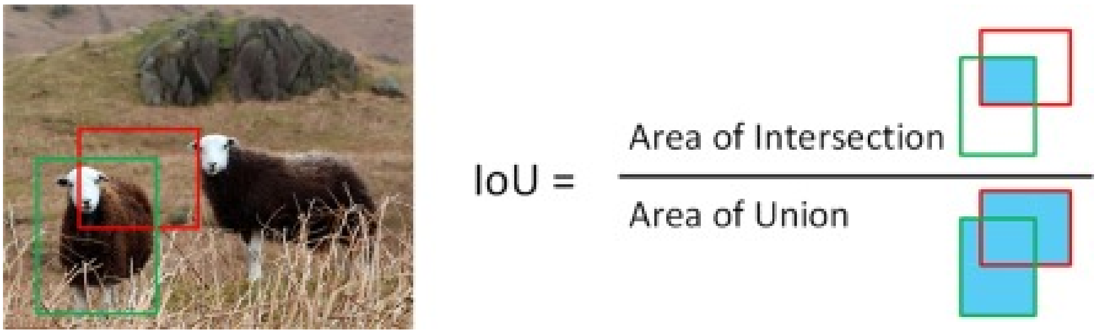


Figure 1.2: Visual representation of IoU calculation.

Reis and Turk explain (2022) that the cross-validation technique is a method for verifying the success of a model and determining whether it is overfit or underfit. The dataset is partitioned into k equal subsets. The model is trained using k-1 subsets, and its performance is evaluated using the remaining subset, referred to as the validation data. This process is repeated k times, with a different subset used for validation each time. The average of the scores obtained represents the model's performance . Figure 1.3 shows a cross validation application with k = 5.

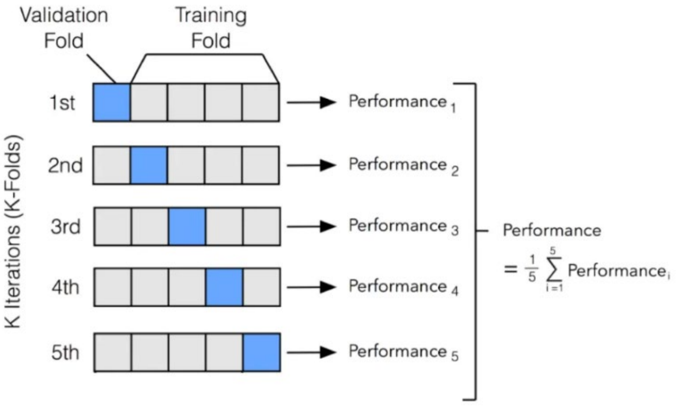


Figure 1.3: 5 Fold Cross Validation.

1.2.4 Restaurant payment systems automation

The adoption of restaurant payment systems automation has the potential to improve efficiency and deliver a smooth dining experience for both customers and staff, making it a key factor in streamlining various work-related processes. In this part we investigated payment system technologies and studies related to our work worldwide.

1.2.4.1 Payment systems technologies

The advancement of technology has made it possible to incorporate artificial intelligence and computer vision into restaurant payment systems automation, this innovation increases accuracy and speeds up the payment process by identifying and processing menu items effortlessly. This technology provides an easy transaction experience by intelligently recognizing the ordered products, facilitating a fast and error-free payment procedure in dining halls, and is very popular in the post-pandemic world. The rapidly changing restaurant technology market is changing the hospitality and tourism industries, with AI, smartphone apps, kiosks, chatbots, and robots playing a crucial part in converting guest experiences and automating operations, despite their early application, prompting managers in restaurants to look for advice on successfully integrating these technologies for service excellence (Blöcher & Alt, 2021).

1.2.4.2 Related study

During the contactless services period, studies were conducted to automate the food industry, utilizing the QR code system in markets and the RFID system in certain establishments; however, bakeries faced challenges in this transition. In a study tested on a donuts dataset, an AI-based checkout system, employing image recognition, is being developed to overcome these difficulties and evolve from a conventional barcode-based semi-automation self-checkout system, aiming to establish a fully automated service flow (Kim et al., 2021). Another study aims to reduce payment delays by shifting the task of fruit and vegetable variety classification from cashiers, the simple structure of CNN and the YOLO V3 method for real-time processing were used, and it was shown that real-time use is practical (Rogozinska & Mielczarek, 2019). Last but not least, a study examines the viability of a prototype system using five deep learning networks for automated and accurate dish recognition in self-serving Chinese restaurants and the system performs well, with accuracy, mean average precision, and recall exceeding 97.0% for three transferred deep neural networks (Lan et al., 2022).

To sum up, these studies highlight the continuous progress in restaurant automation, particularly in employing AI for image recognition, showcasing the evolving landscape of our work in this field.

2. SYSTEM ANALYSIS

2.1 System Definition

Self-service restaurants typically implement a system that allows customers to select their own meals. Diners can either place the chosen dishes on a plate themselves or request staff to serve the food onto the plate. Self-service restaurants afford customers increased options and autonomy in their dining experience. Additionally, these businesses generally appeal to a wide customer base by serving customers with a comfortable atmosphere and more affordable prices.

Customers entering the business move to the front of the food queue. Customer takes a tray from the front of the line. Throughout the entire food pickup and payment line, there is a platform where the customer can slide the tray without carrying it. The tray slide platform can be u-shaped or line-shaped depending on the installation layout of the restaurant. This system ensures that the queue between customers is formed regularly. The pickup order includes soup, main course, salad, appetizers, desserts, side dishes and beverages. It stops the tray from sliding when there is a product the customer wants. The customer tells the staff about the product he/she wants and has it served on a plate, or the customer can buy it himself/herself. At each step, customers place their purchases on the tray. The customer, who places the desired products on his tray, enters the payment queue at the end of the food row. The cashier calculates the cost of the customer's purchased items. It then asks the customer whether the payment method will be cash or card. According to the customer's answer, the payment method is selected and payment is made. The customer finishes the queue and exits the system.

Normally self-service restaurants are operated as mentioned above. The innovation will place in the payment area. The manual payment system will become an automatic system. A picture of the customer's tray will be taken with a camera placed at the end of the food receiving row. The products on the tray are detected by the model created by the DL algorithms. Product-price matching has been done before. The products purchased and the cost of each are visible to the customer. In this way, the customer's trust in the system is ensured. The customer chooses the payment method and after making the payment the customer exits the system.

2.2 System Boundaries

The deep learning algorithm is trained to recognise all the products available in the restaurant. The data used to train the model will consist of ready-made datasets and a dataset consisting of images of the products in the restaurant.
The system does not cover all restaurants. It only begins when the customer picks up a tray and joins the queue to receive food, and the system ends when payment is made. Those outside the scope can be identified as tables where customers eat and spice stands.

2.3 System Stakeholders

Customer: Customers will be affected with reduced waiting times and improved customer experience, the customers of the restaurant are positively affected by implementing this system.

Employees: Employees are affected by this system both positively and negatively. This system, which eliminates the employee at the cash register from the system, is a negative affect as it aims to reduce the number of employees. On the other hand, the benefit of the system to the employees is that other employees work in a more optimal working environment and repetitive work steps are reduced

Restaurant: The stakeholder most affected by this system is the restaurant business. The reduction in the number of employees is a positive effect that reduces personnel costs. However, the high initial installation cost of the system is a negative aspect for the business.

Model Developers: The model used should be continuously improved with the kaizen principle and the success rate of the model should be increased with new data. In addition, developers are responsible for errors in the software system in which the model will be used.

2.4 SWOT Analysis

Strengths

Reduced Queue: With the system to be installed in the payment queue, customer satisfaction can be achieved by reducing the waiting time in the queue.

Collecting Data: People's food preferences can change with the seasons. Some dishes and some side dishes, drinks or desserts can often be eaten together. With this system, the data of the products on the tray that are photographed are recorded in the database. With this data, the company can understand the relationship between customer food preferences and seasonal food needs. This information can be used to create different campaigns.

In addition, the above-mentioned estimation can be used to determine the amount of dishes that are consumed less seasonally. Waste is reduced by varying the amount of food served according to these estimations. In this way, the company can increase its profits.

Weaknesses

Fault in Model: The deep learning model is trained using data. However, insufficient or unsuitable data can lead to incorrect results in the image detection algorithm established by the model. Moreover, the model's overfitting or underfitting can result in incorrect output, leading to a misidentification of products in the customer's tray. This in turn may have negative financial and reputational implications for the enterprise.

Camera Failure: The camera may capture blurred or incorrect images due to external factors and hardware malfunctions. The model could produce undesirable results when unable to accurately identify the dishes in the picture.

Opportunities

It is anticipated that this detection and payment system, which is planned to be established, will be a pioneering development in the food sector. It is a positive development in terms of business prestige.

Image processing technology is continually evolving with the emergence of novel algorithms and advanced methods for processing image data. This enables the enhancement of the image detection model applied within restaurant settings and consequently, boosting model success rates.

Threads

As the image recognition process will be executed on a server, any issues that could arise on the internet may have a negative impact on the exchange of information within the system.

If the customer does not comprehend the food recognition system or if the processing time becomes longer than necessary due to the customer, it could result in delays in the payment queue and consequently lead to customer dissatisfaction.

2.5 Identifying Problems

Firstly, it is necessary to identify the current issues. To this end, the problems encountered in self-service eateries are provided below.

Length of the Payment Queue: Customers concluding their meal selection process subsequently transition to the payment queue. The cashier, in the capacity of their duties, manually computes the transaction total, after which the customer proceeds to effectuate the payment. The manual execution of this procedure may engender an elongated payment queue, thereby precipitating instances of customer dissatisfaction.

Errors caused by the cashier: The occurrence of errors in payment arising from inaccuracies in the cashier's calculation of charges, coupled with potential extensions in transaction processing times due to human-related factors such as fatigue or forgetfulness, may result in customer dissatisfaction.

2.6 Identifying Areas for Improvement

As an innovative system, it may be challenging for customers to comprehend. To achieve a better understanding, it is recommended that regular surveys be conducted to obtain customer feedback and subsequently improve the system. This methodology should be consistently repeated to adhere to the principle of kaizen.

2.7 Constraints and Requirements

2.7.1 Constraints

Server Response Constraint: A certain amount of time must pass for the tray image to arrive at the server, be processed by the model, and display the meal-price match to the user interface.
Customer Constraint : The time customers spend in the queue system depends on the server response time as well as on the customer itself. It is an independent variable. Personal characteristics such as the customer's age, knowledge level, and propensity to use new technology are restrictive.
Model Running Constraint :A certain amount of time must pass for the model to detect which products are in the tray image. Affects server response constraint.

2.7.2 Requirements

Hardware Requirement: The new camera system must be procured and the server on which the model will run must be rented or procured. In addition, a screen is required so that the customer can see the products he bought and the price.

Software Requirement: It is necessary to write an application to create a user interface. In addition, a deep learning algorithm should be developed using software languages to establish an image detection model.

2.8 System Requirement Definition

First, we need a model that can detect images of food, drinks, desserts and other products purchased in our system. This model will be a deep learning algorithm using image processing technology. It is expected that this algorithm will be able to detect situations that are slightly out of the norm, such as too few portions and too many portions.

In order for the model to be able to recognise and classify the dishes in the images taken, it must be trained with a sufficient number and quality of image data. This data can also be obtained from the relevant images on the Internet and from the company where the application is to be used. In addition, each type of food in the dataset must be correctly labeled.

Before training the model, certain procedures must be carried out on the data sets. These processes will convert our data set into a format that permits deep learning methods to more effectively detect images. Some of these procedures include image filtering, labeling, training and testing data separation and data augmentation.

The model has been trained using a variety of deep learning algorithms. The optimal method is chosen through model evaluation techniques.

To make predictions with this model, a remote server must be provided. An application must be developed to display the names, prices and quantities of the products detected by the model.

A camera system should be placed at the end of the food queue to take pictures of the products in the trays. These images are transmitted to the model through the application. The model distinguishes the item illustrated and presents the corresponding price and product information on the screen as output generated by the application. This informative display screen is located adjacent to the camera that takes the tray's pictures.

2.9 Purpose of the Study

The goal of this project is to improve restaurant systems by eliminating the cashier, to improve the process that is prone to human error, to save time and money by automating the payment area, and to contribute to the company's digitization process.

The process to be improved is to reduce the waiting time in the payment queue. Image processing technology was used to reduce the waiting time at checkout, and the quantity and price of the purchased products are displayed on the user interactive screen. The payment process is also automated with this application, without the need for a cashier.

Simultaneously, data on which products were consumed, when and how much were consumed will be collected. Thanks to this data, the business will prevent food and product waste due to not being sold. While some products can be stored for a long time, the shelf life of some products - usually hot dishes- is one day.

2.10 Success Criteria

Model Success Rate: Two rates can be distinguished. Firstly, evaluation methods can be employed to measure the success rate of the model. This value denotes the rate of correct recognition of images in the test data and can be described as the theoretical success rate. Secondly, it refers to the rate of accurately recognizing what the products are on the tray, how many they are, and the portion size. It can be described as the actual success rate.

Database Usage: The effect of predictions on a company's income using the collected data from the database, which outlines the dates and products purchased by customers. This data serves as a secondary measure of success.

Time Spent in Payment Queue: It starts when the customer leaves the food queue and enters the payment queue. It ends when the customer exits the system. The variables that affect this time are the model cycle time, the customer's transaction speed, the time from the camera to the server, from the server to the user interface, and the customer's payment time. It is a general criterion. The variables affecting this criterion can also be evaluated individually as success criteria.

Cycle Time of the Model: The process commences with the tray image as input to the model, and concludes when the items on the tray have been detected. It indicates how long it takes the model to detect the products on the tray. This measure of success is a dependent variable that affects the time spent in the payment queue.

Ease of Use: Incorporating new technologies into preexisting systems offers users medium and long-term advantages. However, users may encounter several challenges when adapting to the updated system. The swifter the customer adjusts to the new system, the less time the customer waits in the payment queue.

3. METHODOLOGY

3.1 Data Collecting

The restaurant plans to implement object detection to determine the classes of food and other products sold. Two classes will be created for each meal: half portion and full portion. For instance, rice pilaf will have a class for half portion and another for full portion. Data will be collected based on these classes. Price information of products belonging to these classes will be saved in Excel.

In order for deep learning algorithms to detect target objects on an image, they must be trained with images similar to that object. Within the scope of this project, a data set containing food images is required.

3.1.1 Datasets

They are ready-made datasets uploaded to data sharing platforms by other users. The data sets within the scope of the project contain images in json, png, jpg formats. Pictures are folded according to type. Additionally, there may be formats containing all image data in ".h5" format for special data reading functions such as "Keras HDF5Matrix". This special format allows data to be read to the Python interface in a shorter format than other methods. The advantage of this method is that a regular data set allows us to obtain data in a shorter time than other methods during the data collection phase. In this method, the dataset is obtained from images found on the internet or taken with a camera. Images that differ from each other in resolution are an element that increases the effort in editing the dataset. Additionally, obtaining the dataset takes longer than other methods. As previously stated, in our study, we will use both ready-made datasets and an expanded dataset created from the beginning customized to the menu of the restaurant where the project will be implemented.

3.1.2 Data collection method

First of all, ready-made data sets available on the internet will be used. Secondly, if there are not enough images for some classes in the datasets, the method of manually creating the dataset will be used. The data set will also include images showing foods served in half portions. This will allow the algorithm to distinguish between half and full portions. Data with dishes on the tray will be used to test the model. In this way, our model will be tested with images similar to the tray images in the restaurant system.

3.2 Image Labeling

Deep learning algorithms work by establishing a relationship between image data belonging to the same class and checking whether there is an object belonging to that class in the image given for testing. This check is performed for all classes and determines which classes an object belongs to. When labeling data, two different image-based labeling can be mentioned.

Some images contain only the target object. For this image data, one type of classification is sufficient. For example, if there is only a plate of rice in the picture and there is no object other than rice in that picture, it can be defined as a picture with only the target object. What is meant by different objects is that there is no object belonging to another class. For example, in a data set containing dishes, it is undesirable to include yogurt in the images of the rice class. However, the plate under the rice is not a different object. In order to enrich the data set, adding a picture of rice served on different plates in a picture of a class can increase the success of the model. The purpose here is to show that the basic object belonging to the rice class is rice, and to teach the model that changing the plate underneath does not change the class of the targeted object. Some images may have more than one target object on them. If there are objects belonging to the targeted class on the image data, these objects are classified by labeling with various methods. The data labeling method to be used in this study is to enclose the targeted object in a rectangle and assign a class.

3.2.1 Image Labeling Methods

Basically, two labeling methods can be mentioned. These can be said as manual labeling and automatic labeling. Manual labeling is when the user classifies the targeted object on the image by drawing a bounding area. For datasets containing a large number of data, this process may take a long time. However, it is the most reliable method in terms of accurately labeling images. Automatic image labeling is created using pre-trained deep learning algorithms. The task of these algorithms is to enclose objects belonging to the targeted class within bounding lines. It is faster than the manual method in large data sets. However, the labels and classes created by the algorithm must be checked by a user.

3.2.2 Image labeling applications

Special softwares is used for image labeling. Roboflow software and LabelImg software will be used in this study. Roboflow, which serves as a website, is one of these software. According to Shandilya et al. (2023), Roboflow allows labeling of image and video data by using bounding rectangular boxes. Images and videos in various formats can be easily loaded into this software, and the data set required for frequently used image processing algorithms can be easily exported. Additionally, the tools required for data augmentation can be easily used with Roboflow. With these features, the prominent benefits of Roboflow are that it saves time and effort. LabelImg software is a PC application that performs object detection and labeling on images. However, unlike Roboflow, it does not support data augmentation. In the study by Guo et al. (2023), LabelImg software was used to detect 4 defects on the radiographic image, enclosing each defect in a rectangular box on each image and defining its label.

3.3 Data Augmentation

Data augmentation is used to prevent underfitting or overfitting during model training. The model may encounter challenges in identifying certain food images. so enlarging the dataset is necessary to increase accuracy. New synthetic data is obtained using the techniques mentioned above. The number of data in the dataset increases. Descriptions and examples of some data augmentation methods are shown below.

Rotation: New images are obtained by rotating the image at such angles.

Mirroring: New image data is obtained by reflecting the image on the x-axis or y-axis.

Scaling: It is achieved by zooming in or out on the image so that the targeted object remains in the newly created image.

Adding Noise: Noise is unwanted and distorted pixels in the image. The performance of the model can be improved by adding noise to the image.

3.4 Data Preprocessing

Resizing

Although higher resolution images contain more detail, the time it takes to train the model increases. In addition, the detail contained in high-resolution data may cause the model to memorize, that is, overfitting. To prevent these situations, image data can be resized to a lower resolution such as 640 * 640 pixel resolution. The model can detect the fundamental features in the picture, leading to more successful learning. The dataset may also include images of varying resolutions. To standardize the data, all images must have the same resolution.

Data splitting

To determine which machine learning and deep learning algorithms are more successful, various success metrics are compared. To make this comparison, the dataset must be split into training and test data. The dataset's images are typically divided into 'train data' for algorithm training and 'test data' for evaluation. The standard practice is to split 80% of the data for training and the remaining 20% for testing. This allocation is done randomly to ensure objectivity.

K-fold cross validation

As the data set is randomly split into training and testing data, the model's score may vary. To reduce the error caused by these changes and obtain more accurate results, the cross-validation method can be used.

3.5 Deep Learning Methods

Because the R-CNN model is slower than YOLO, its use in the restaurant payment system causes the queue to get longer. This situation contradicts the main purpose of the system. It would be more logical to use YOLO in the system.

3.5.1 Evaluation of the model

The trained model is tested using the test data, generating various scores. While high scores are preferred, they may also indicate overfitting, which is undesirable.The IOU, mAP, Precision, Recall and F1 Score metrics mentioned below are the metrics used to measure the success of our object detection model.

Mean average precision (mAP) is a technique used to evaluate the performance of object detection algorithms. The confusion matrix components used to make this evaluation compare the actual and predicted values, as shown below. The actual value refers to the position of the bounded rectangular box made in the "Image Labeling" stage where the targeted object is located.

True Positives (TP): It is the number of times the model predicts a positive value when the actual value is positive.

True Negatives (TN): It is the number of times the model predicts a negative value when the actual value is negative.

False Positives (FP): It is the number of times the model's prediction is positive even though the actual value is negative. It is also called Type 1 error.

False Negatives (FN): This is the number of times the model predicts a negative value when the actual value is positive. It is also known as Type 2 error.

According to Chumachenko et al. (2022), precision value is the ratio of TP value to the sum of TP and FP values. Precision score is a metric that shows the model's ability to avoid FP. To qualify as TP, the IoU value must be equal to or greater than a specific threshold. A threshold of 0.5 corresponds to the 'mAP0.5' evaluation method, while a threshold of 0.75 corresponds to the 'mAP0.75' evaluation method.

The recall value is calculated as the ratio of TP to the sum of TP and FN values. This metric measures whether it can detect all instances belonging to a class. The “F1 Score“ is the harmonic mean of the Precision and Recall values, and is the value that shows the effect of these two metrics when evaluating the performance of the model.

3.5.2 Deploying the model

The model is ready for use after the training and testing phases. It is designed to detect predefined target objects on real image data, specifically tray images obtained from the restaurant. Tray images taken from the camera are estimated with the YOLO model. The detected meals, their quantity, price information, total amount and total working time of the application are output. In addition, information on how many products were purchased, process time and time information are kept for the business to use in the future. In this way, the restaurant obtains information about which products are sold, in what quantities and when. By saving the images taken for object detection, the data set is provided to develop the model. In order to implement the mentioned tasks, functions were written using the Python language.

3.6 Simulation & Evaluation Sytsem

The objective of this study was to ascertain the differences between the conventional cashier payment system and the payment system that employs a model capable of detecting objects. The Arena simulation program was employed as the simulation application. It should be noted that the number of customers queuing is distributed according to the varying customer density observed throughout the day. Two distinct systems were subjected to simulation, with varying densities.

The duration of the payment period in the classical system was observed in the Selfiş restaurant and a time study was conducted. In the system that employed image processing, the average time taken for tray detection was calculated on the basis of the processing time of the model.

A survey was conducted on individuals who had previously purchased food from open buffet restaurants to ascertain the number of people in the queue who would abandon their purchases. The survey results enabled the calculation of potential customer losses that may occur in both systems.

In light of the aforementioned potential customer loss, it was determined which system was the more cost-effective option in accordance with the varying customer density.

4. APPLICATION

4.1 Decided Restaurant

Deciding on the restaurant to work with for this design project is an important step of the application part. Since Turkish dishes and beverages are very diverse, we took care to make it a sufficiently inclusive restaurant. Selfiş restaurant in the campus is a suitable restaurant to demo this application because it is a system where the project can be implemented both in terms of self-service and the variety of Turkish dishes they offer. In addition, it is advantageous when collecting data for the simulation study due to its location.

4.2 Data Collection

We agreed with the restaurant on the condition that we would be there at certain time periods for the data collection process. We planned to take tray photographs and collect data during the restaurant's relatively quiet hours, rather than during busy hours.

4.3 Dataset Preparation

We followed a two-stage process while creating the dataset. In order not to spend extra effort and save time for standard classes such as boxed drinks, we used ready-made drink images on the internet. We created a data set by manually taking tray photographs for the part specific to the restaurant we work in, namely soups, main courses, desserts and salads, as it should be specific to the restaurant we work in.

4.3.1 Ready dataset

Below there are two sample pictures of our dataset. These are took from open-source internet resources. The advantage of using these is that they are well-positioned and ready to label and can be easily found with basic search. In the first figure, it can be seen a global beverage brand (Figure 4.1). In the second, a local known beverage is shown (Figure 4.2).

Figure 4.1 : Internet based global brand beverage example.

Figure 4.2 : Internet based local brand beverage example.

4.3.2 Manually prepared dataset

In the second part of creating a dataset for our work, we took pictures of selected trays in the restaurant that we worked together. These tray photos includes several types of main courses, soups, desserts, salads and beverages. In Figure 4.3 you may see a traditional Turkish dish meatballs and potatoes, rice pilav and bread. In Figure 4.4, you may see one of the most preferred menu on the restaurant. The third picture shows a dessert and tea combination (Figure 4.5). Finally, the last picture taken from us to create a dataset is showing a piece of cake which is sold by the restaurant that we worked with. The purpose of showing these pictures is helping the reader to better understand our dataset.

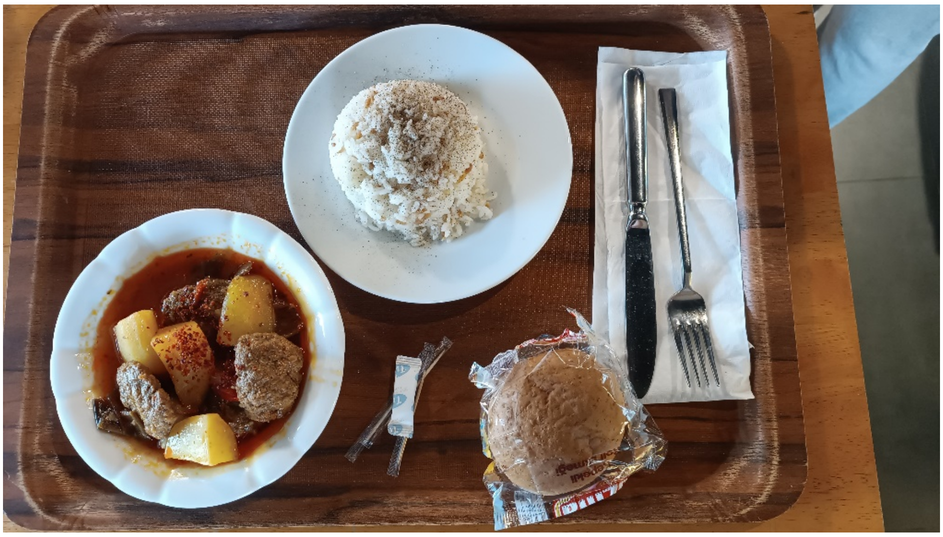


Figure 4.3 : An example of traditional Turkish food tray.

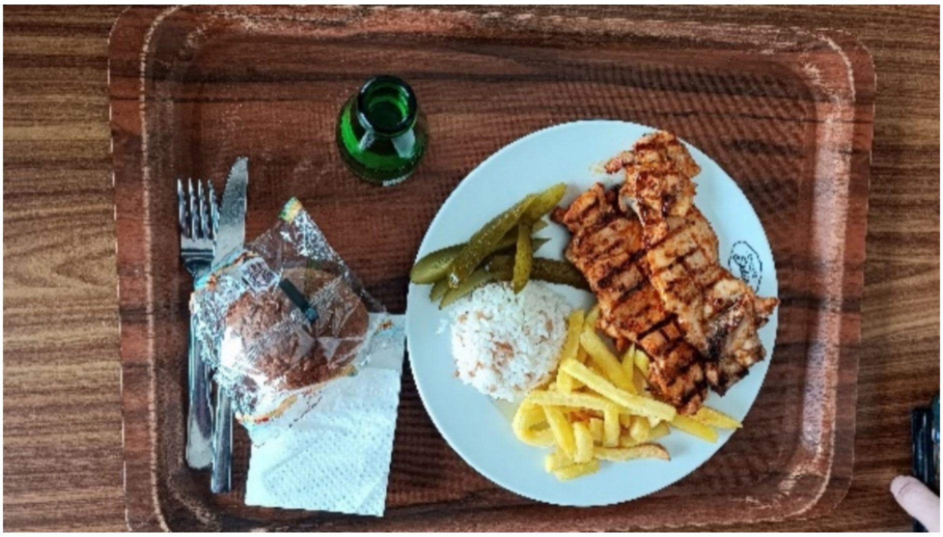


Figure 4.4 : Chicken menu tray example.

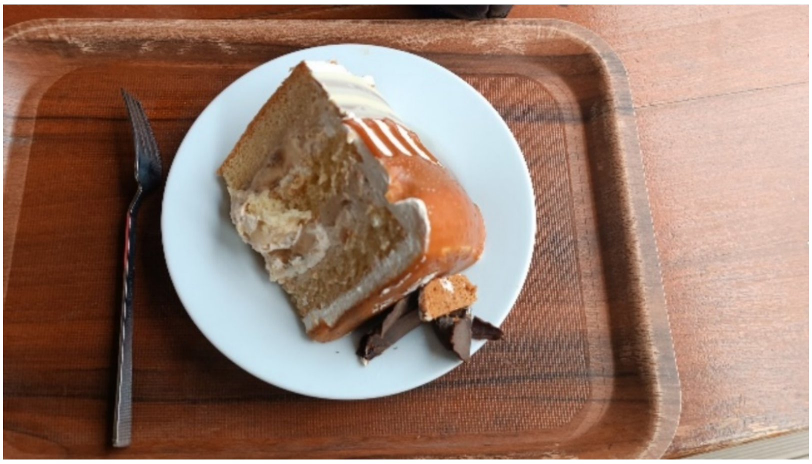


Figure 4.6 : A piece of cake tray example.

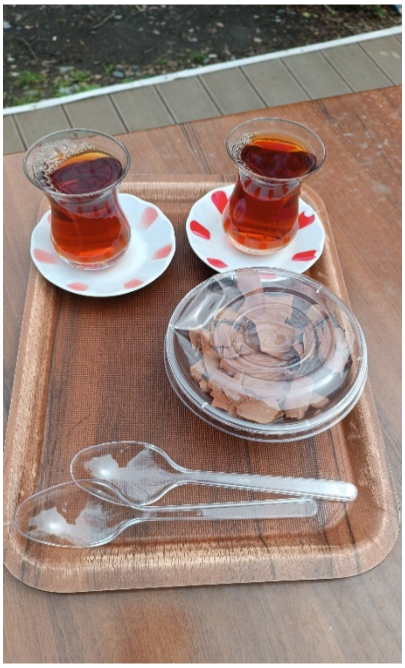
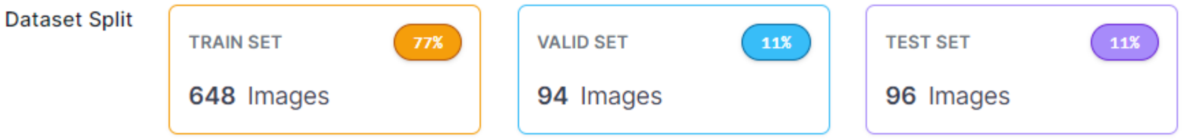


Figure 4.5 : Tea and dessert tray example.

4.4 Dataset Properties

We currently use nearly a thousand photos in our dataset. There are 838 photos in the first version of our dataset. In the basic model, after the train-test-validation distinction, 648 photographs can be allocated to train, 96 photographs to be used for testing, and 94 photographs to be used for validation (Figure 4.7). These numbers may increase or decrease as the model develops. Additionally, numbers may change after data augmentation. This issue is the reason for the observed dataset width differences in the model differences used. The image below shows how many food/beverage/dessert classes there are in the dataset. The class balance in the current dataset is as follows (Figure 4.8). This chart will be used for tracking purposes while enriching the dataset. Additions will be made for classes with a small number of data.



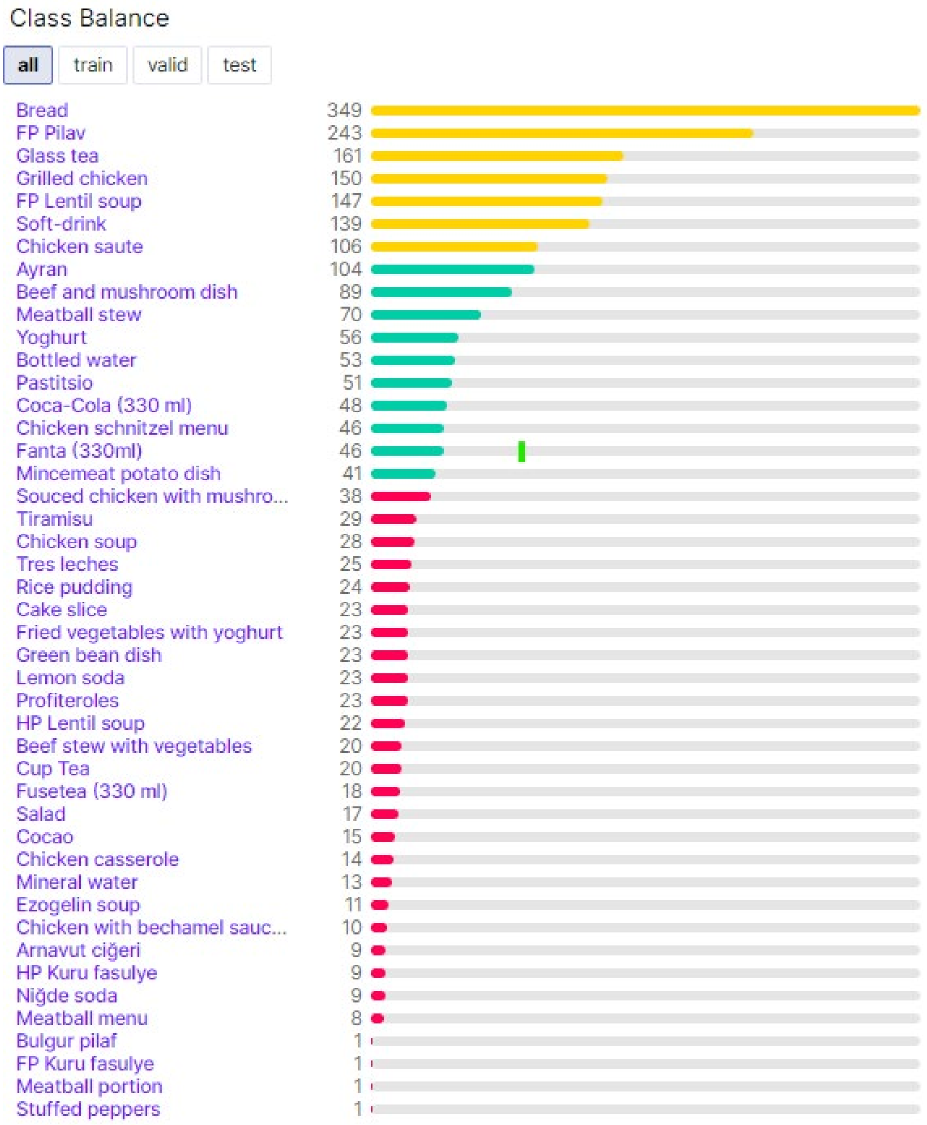
Dataset Split

TRAIN SET 77%
648 Images

VALID SET 11%
94 Images

TEST SET 11%
96 Images

Figure 4.7 : Dataset split.


Class Balance

Bread	349
FP Pilav	243
Glass tea	161
Grilled chicken	150
FP Lentil soup	147
Soft-drink	139
Chicken saute	106
Ayran	104
Beef and mushroom dish	89
Meatball stew	70
Yoghurt	56
Bottled water	53
Pastitsio	51
Coca-Cola (330 ml)	48
Chicken schnitzel menu	46
Fanta (330ml)	46
Mincemeat potato dish	41
Souced chicken with mushro...	38
Tiramisu	29
Chicken soup	28
Tres leches	25
Rice pudding	24
Cake slice	23
Fried vegetables with yoghurt	23
Green bean dish	23
Lemon soda	23
Profiteroles	23
HP Lentil soup	22
Beef stew with vegetables	20
Cup Tea	20
Fusetea (330 ml)	18
Salad	17
Cocao	15
Chicken casserole	14
Mineral water	13
Ezogelin soup	11
Chicken with bechamel sauc...	10
Arnavut ciğeri	9
HP Kuru fasulye	9
Niğde soda	9
Meatball menu	8
Bulgur pilaf	1
FP Kuru fasulye	1
Meatball portion	1
Stuffed peppers	1

Figure 4.8 : Class balance.

4.5 Image Labelling

In order for deep learning algorithms to function, images from the same class must be correlated, and each image that is provided for testing must have an object from the same class present. To make things simpler, we prefered using Roboflow to perform image labelling. Below you may see the examples of different kinds of foods in different classes (Figure 4.9, Figure 4.10).

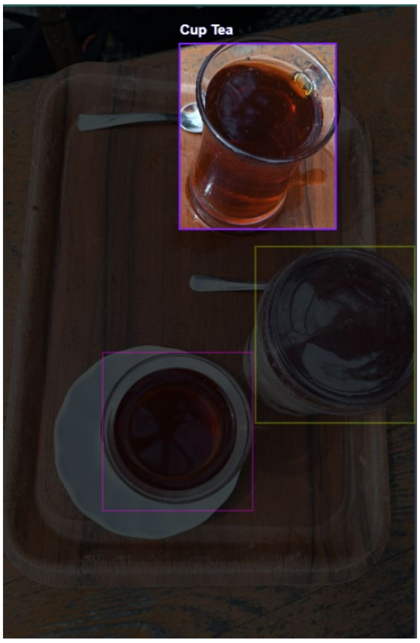


Figure 4.9 : Cup tea class example.

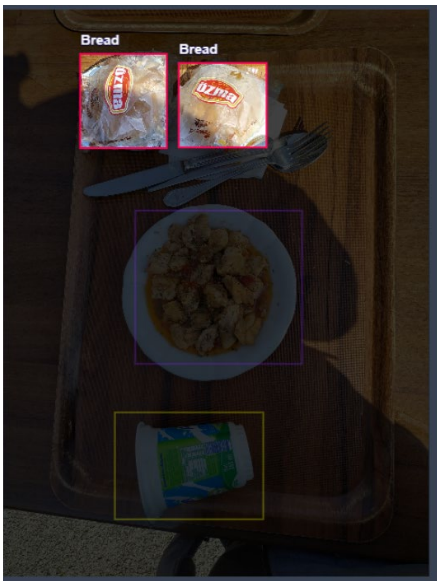


Figure 4.10 : Bread class example.

Another issue to focus on is half portion and full portion classes. Customers may not always choose full portions of the food they buy. It has been observed that this situation occurs frequently, especially in the lentil soup class. You can observe this difference in the figure below. The image of a full portion of lentil soup is below (Figure 4.11). There is also an image of the half portion lentil soup class below (Figure 4.12). Full portions are labeled as “FP” and half portions are labeled as ”HP”.

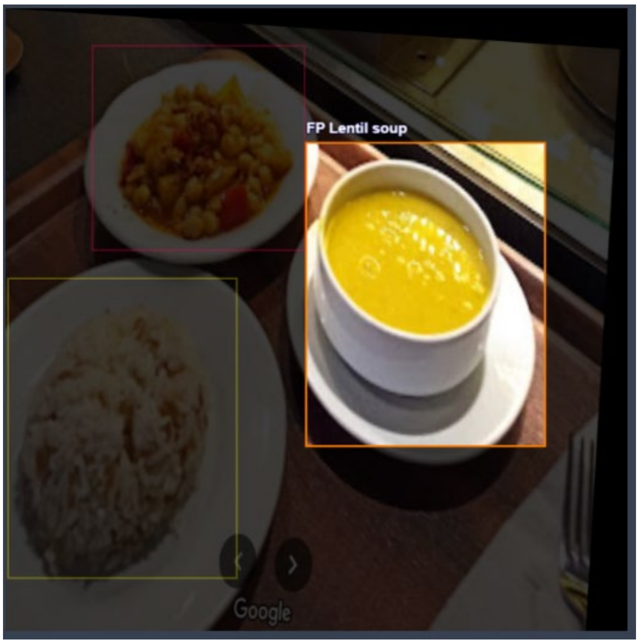


Figure 4.11 : Full portion of lentil soup.

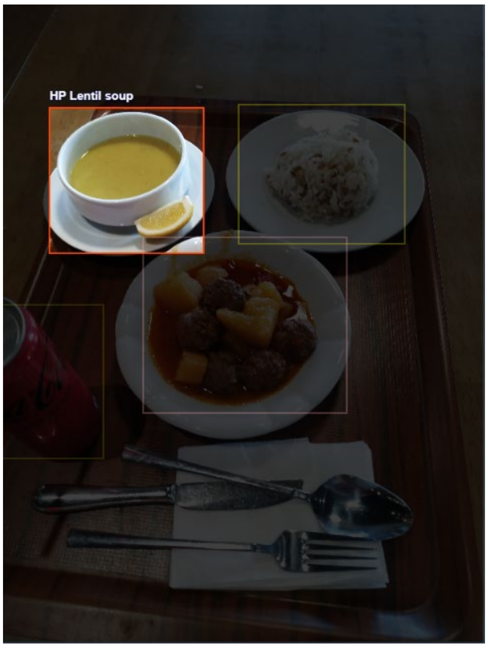


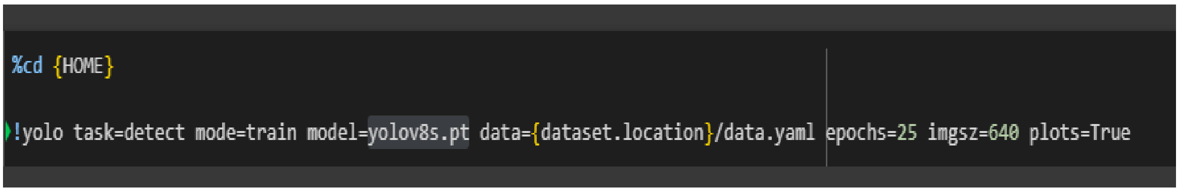
Figure 4.12 : Half portion of lentil soup.

4.6 Model Selection

YOLO V8 will be utilized for training with 25 epochs. The data is directly sourced from Roboflow and converted into ".yaml" format as "data.yaml". During this model training, the YOLO's native model "yolov8s.pt" will be employed to obtain the weights file. With this weights file, the best-performing epoch will be saved as "best.pt". Below are the results of mAP, precision, and recall based on the preprocessing and data augmentation processes conducted on the dataset. From version 1 (V1) to version 5 (V1) you may see the table below (Table 4.1). The scores for 25 epochs are provided in the images. The recall, precision, F1 score, and mAP values corresponding to the epoch row with the highest mAP score for that version are listed below. Additionally, the total number of data points in the dataset is specified (Figure 4.13). From version 6 (V6) to version 10 (V10) you may see the table on the next page (Table 4.2).

	V1	V2	V3	V4	V5
Dataset Size	838	838	3301	3114	3240
Preprocessing	-	Auto orient, Resizing (640*640 pixel)	Auto orient, Resizing (640*640 pixel)	Auto orient, Resizing (640*640 pixel)	Auto orient, Resizing (640*640 pixel)
Data Augmentation	-	-	Outputs per training example: 5 Rotation: Between -15° and +15°	Outputs per training example: 5 Brightness: Between -15% and +15%	Outputs per training example: 5 Noise: Up to 1.01% of pixels
Precision	0,7336	0,8900	0,9619	0,9679	0,9588
Recall	0,6139	0,9782	0,9815	0,9786	0,9793
F1 Score	0,6684	0,9320	0,9716	0,9732	0,9689
mAP50	0,7118	0,9861	0,9856	0,9816	0,9868
mAP50-95	0.47421	0,7766	0,7470	0,7702	0,7766

Table 4.1 : Scores and information from v1 to v5.



```
%cd {HOME}

!yolo task=detect mode=train model=yolov8s.pt data={dataset.location}/data.yaml epochs=25 imgsz=640 plots=True
```

Figure 4.13 : YOLO model.

	V6	V7	V8	V9	V10
Dataset Size	1964	856	3225	4726	3240
Preprocessing	Auto orient, Resizing (640*640 pixel)	Auto orient, Resizing (640*640 pixel)	Auto orient, Resizing (640*640 pixel)	Auto orient, Resizing (640*640 pixel)	Auto orient, Resizing (640*640 pixel)
Data Augmentation	Outputs per training example: 5 Flip: Horizontal, Vertical	Outputs per training example: 5 Grayscale: Apply to 10% of images	Outputs per training example: 5 Shear: ±10° Horizontal, ±10° Vertical	Outputs per training example: 7 Flip: Horizontal, Vertical Rotation: Between -15° and +15° Shear: ±10° Horizontal, ±10° Vertical Grayscale: Apply to 10% of images Brightness: Between -15% and +15% Noise: Up to 1.01% of pixels	Outputs per training example: 5 Flip: Horizontal, Vertical Rotation: Between -15° and +15° Brightness: Between -15% and +15% Noise: Up to 1.01% of pixels
Precision	0,9518	0,9518	0,9354	0,9647	0,9607
Recall	0,9733	0,9733	0,9708	0,9684	0,9812
F1 Score	0,9625	0,9625	0,9528	0,9665	0,9709
mAP50	0,9799	0,9799	0,9843	0,9863	0,9709
mAP50-95	0,7723	0,7723	0,7686	0,7528	0,7706

Table 4.2: Scores and information from v6 to v10.

In this project, F1 score measures the accuracy of predicting the correct class, while mAP evaluates the ability to locate the correct position. Considering the project requirements, a selection of precision and augmentation that emphasizes F1 score would be more appropriate. However, it's essential not to completely disregard the mAP values. According to the table below, applying the model in version 4 to our system would yield the most accurate results. The weighted score in the table is calculated by assigning a weight of 5x to the F1 score, 2x to mAP50-95, and 1x to mAP50. Precision, recall, mAP 50 and mAP50-90 values can be found in figure 4.14.

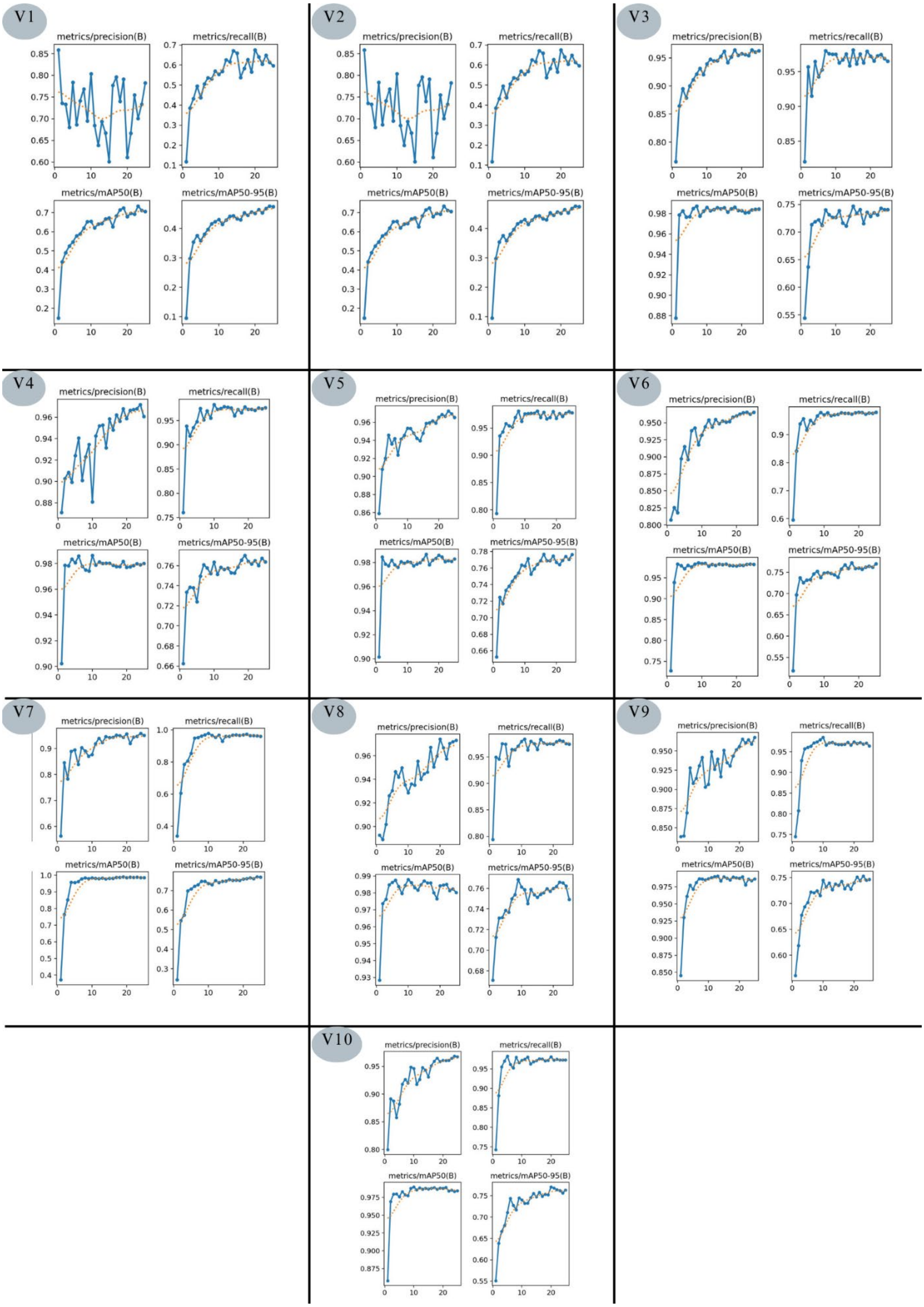


Figure 4.14 : Performance graphs from version 1 to 10.

4.7 Model Evaluation

We looked at the test results to see if there was a problem. In the pictures below, you can see where the model made mistakes and where it outperformed. In figure 4.15, you may realize that the algorithm detects a bread from side table. Although the ability to recognize bread even though only a small part of it is visible shows that the model works well, in real life, marking the products in such side trays may cause customers to pay more than they need to pay. To eliminate this potential problem, it will be sufficient to carefully select the location where the tray will be photographed and define it with lines while the system is being installed.

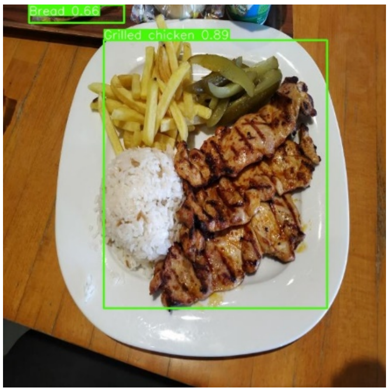


Figure 4.15 : An example of a detection from the side tray.

Another mistake we noticed while reviewing the test results was that the model incorrectly labeled the product. Shown in figure 4.16, a main course is labeled as bottled water. This is a predictable error, but since it is not very frequent, it does not pose a serious problem. However, if it is determined that this error increases in a certain class, it would be a good solution to increase the data set for that product as a preventive action.

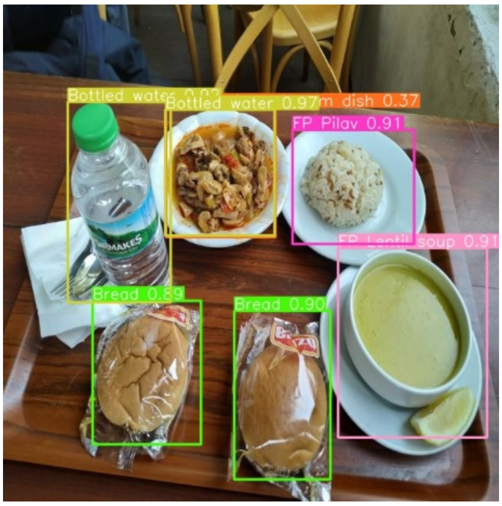


Figure 4.16 : An example of a wrong labeling.

We realized that another incorrect labeling problem was caused by the angle in which the photo was taken. As seen in Figure 4.17, the algorithm failed to read the rice pilaf left behind. The conclusion we draw from this can be eliminated by taking the photographs with a camera placed on top of the trays, as planned.

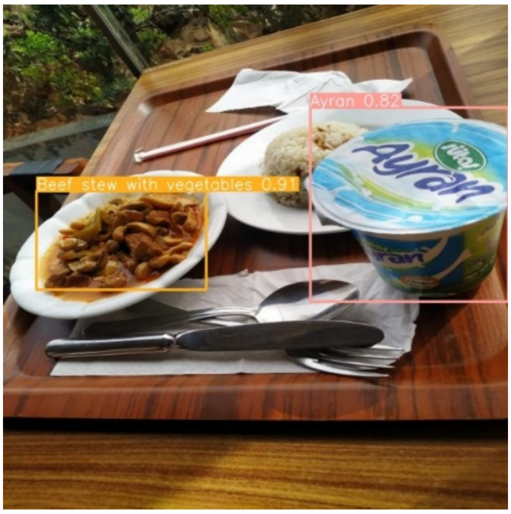


Figure 4.17 : An example of not being able to detect the product left behind.

When we examine the test results of the model, it works exactly as expected in terms of not assigning a class to personal belongings such as a phone, wallet, etc. We made sure that personal items placed on the tray were not perceived as food or anything else. A successful example of this situation can be seen in figure 4.18 below. The model, which correctly identified the food, bread and drink on the tray, did not assign the wallet and the customer's hand to a class.

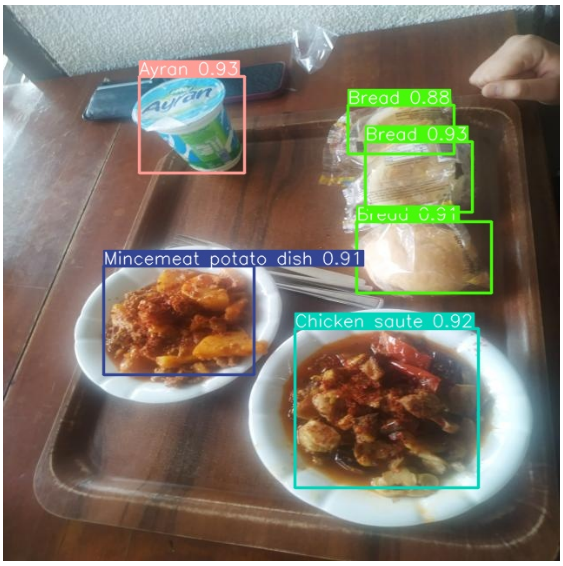


Figure 4.18 : A successful example of not labeling personal items in the tray.

It is possible that incorrect labeling may sometimes be made while labeling the dataset. You can see an example of a mislabeled main course in Figure 4.18 below. The main course is labeled as soft-drink. Despite this, when we examined the test data outputs of the model, we saw that the model labeled this food correctly. This is an indication that the model was trained well as a result of other correct labeling in that class. In Figure 4.19 you can see that the model classifies the food correctly.

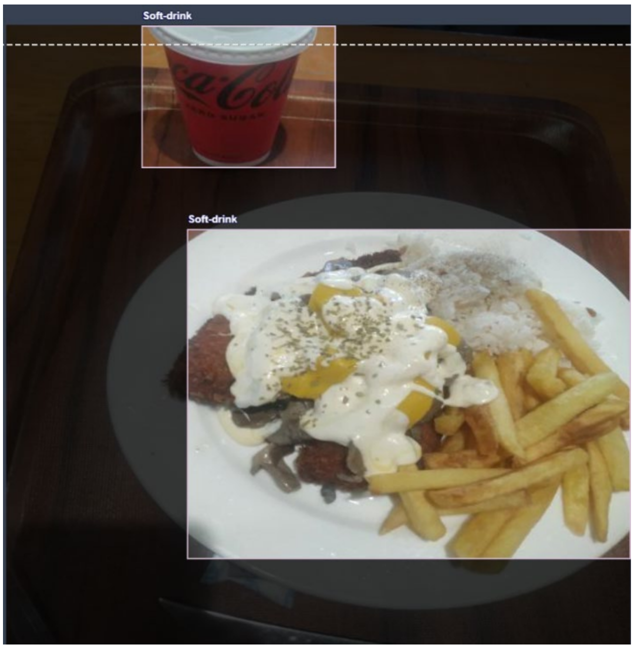


Figure 4.19 : Mislabeled data example.

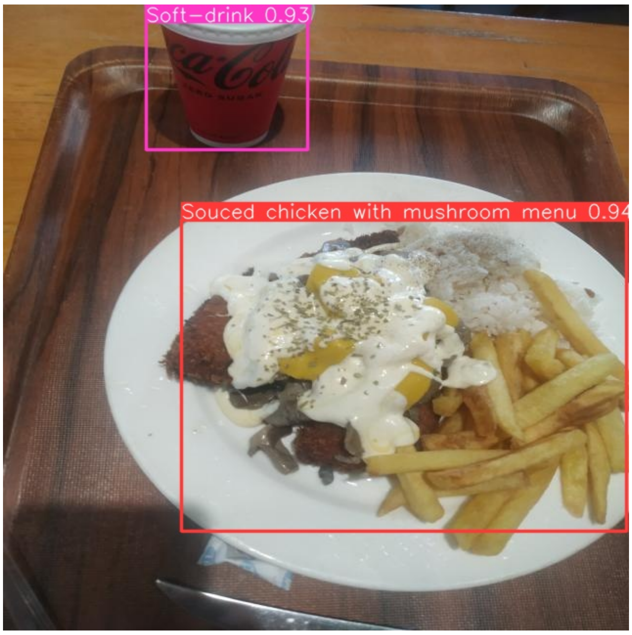


Figure 4.20 : A successful example of the test result.

4.8 Dataset Enrichment

We used the first dataset we created when building the first models. We built different models and choosed the one that gave the best results. Later, while evaluating its performance on the test data, we identified points where we needed to improve the dataset.

The first of these was the balancing of classes, which was mentioned in the previous sections. While creating the dataset, collecting more samples from some classes and less from others resulted in the class balance of the dataset being unbalanced. There were many examples, especially of the bread and rice found on most of the trays. In addition, there were fewer examples of main course dishes that were not popular. We identified these and took more photos and tagged them. We randomly divided these new additions into train, validation and test sets.

Another step we took to enhance our model was enriching the dataset by adding more photos to specific classes that had high error rates in our test results. By identifying these problematic classes, we aimed to provide the model with more examples, ensuring it could learn and generalize these classes better.

Since it is more challenging for the model to distinguish between full and half portions compared to identifying the type of food, we augmented the dataset with additional examples focused on portion sizes. For instance, the bean dish has both full portion and half portion classes. To improve the model's performance in this area, we added more labeled photos to both classes.

Since the number of mineral water images in the dataset was relatively small, additional images of trays containing mineral water were included in the dataset. The model was trained again by combining the old and new images. However, as shown in Figure 4.21, our model incorrectly predicted the presence of mineral water in images containing a fork and spoon. To solve this problem, images containing mineral water and a fork were also included in the dataset, as well as images containing only fork and spoon, as shown in Figure 4.22. Adding these images enabled the model to distinguish cutlery, mineral water, and other items.

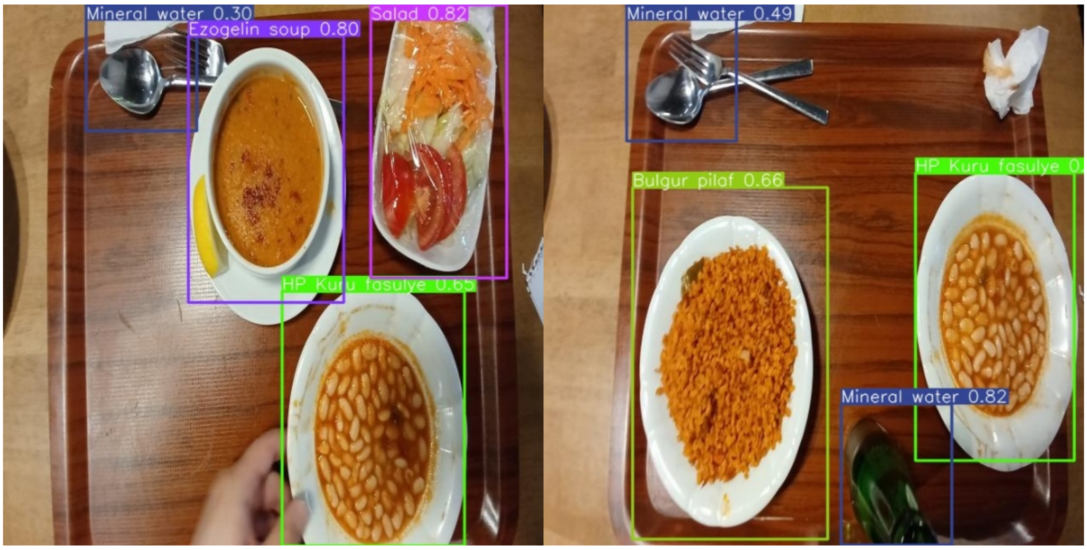


Figure 4.21 : Model predicts incorrectly.

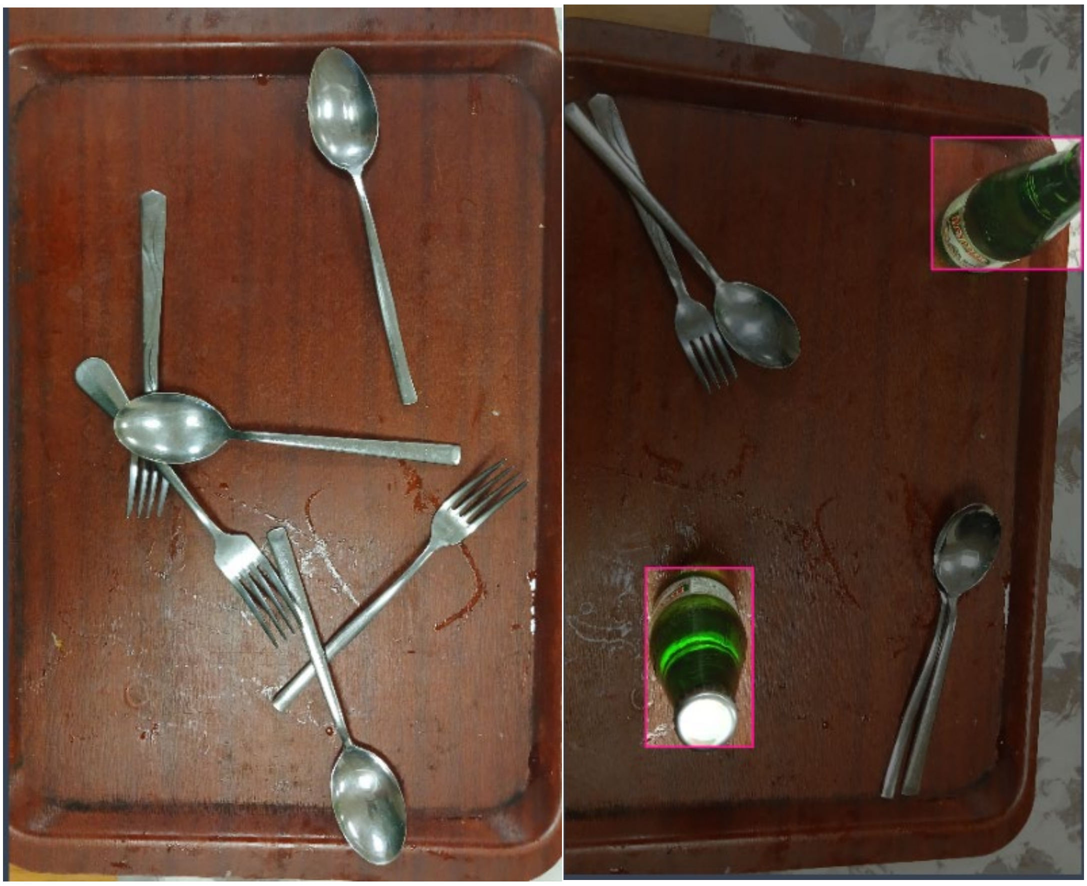


Figure 4.22 : Some of the images added to the dataset.

The distribution of classes in our data set before model training is shown in figure 4.23. There are a total of 1067 images in the dataset. While 807 of these are in the train set, there are 138 images in the validation set and 122 images in the test set.

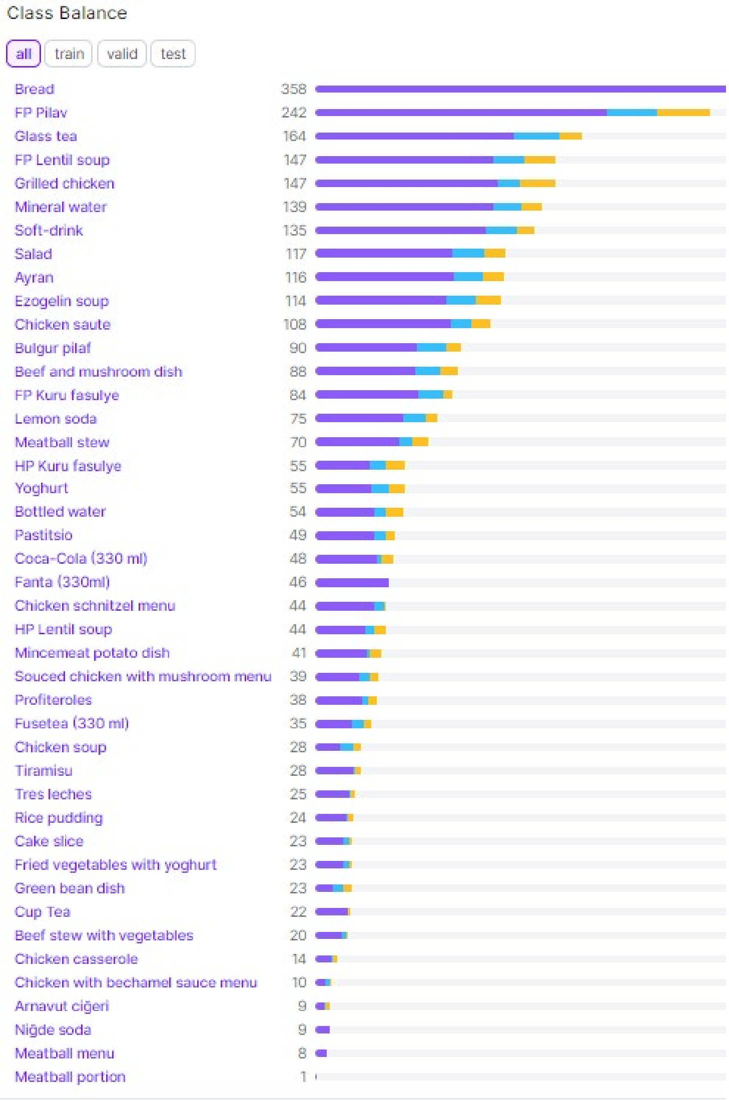


Figure 4.23 : Final class balance.

In Section 4.6, our model was trained using the augmentation method and preprocessing method, which gave the best results. This corresponds to using the augmentation and preprocessing steps in v4. The YOLO model trained with this method has been downloaded.

Figure 4.24 shows an example of prediction of images in the test set after model training. It has been seen that the error in figure 4.21 has been corrected. However, this time a duplication error occurred in figure 4.25 . There are 3 other errors similar to this situation. There are no other incorrect predictions in the 122-image dataset. This situation shows that the model is overfitting in the data of the "HP Kuru Fasulye" class.

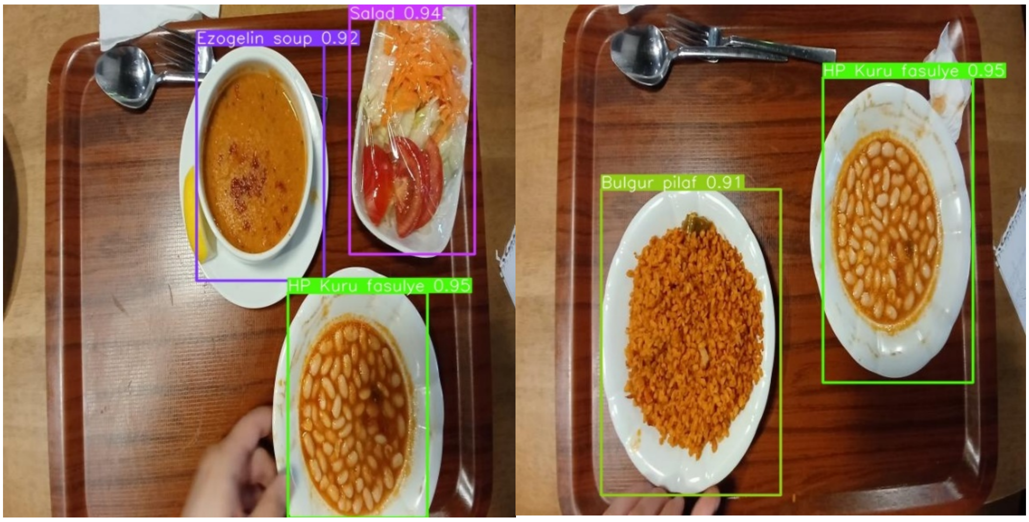


Figure 4.24 : Correct prediction after an error.

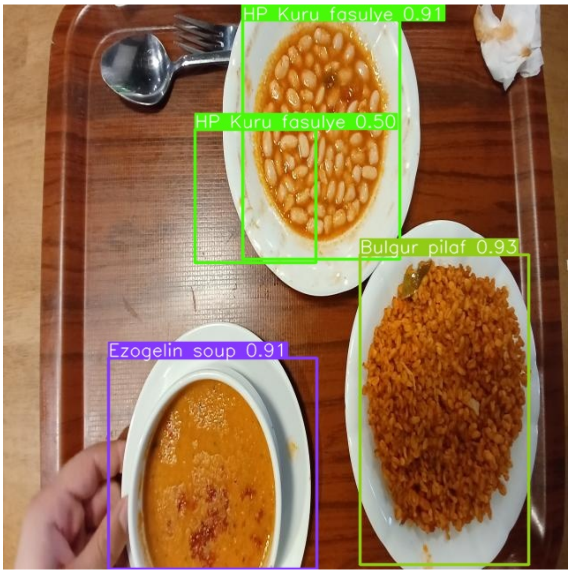


Figure 4.25 : Duplicate error.

4.9 Application of the Model with Python Code

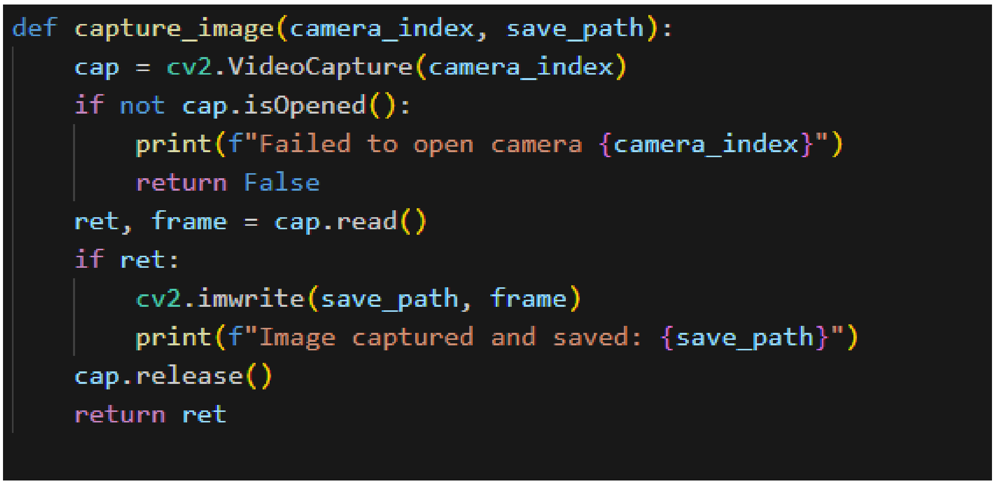
This section presents the Python code developed for the trained "Yolo" model, which is capable of identifying food items within the payment system and subsequently retrieving the total price information as an output.

4.9.1 Installing the necessary library and data

YOLO, CV2, PANDAS, NUMPY, Time, DateTime and OS functions are the Functions belonging to various libraries required for the Python code to run succesfully. The most recent YOLO model discussed in the preceding section will be employed as the model. In order to calculate the total amount of information, the price data for each class included in the data set was created. Furthermore, each transaction is recorded in a process_log data frame, which contains information on the products purchased, the total paid amounts, and the time taken for the products to be received.

4.9.2 Capturing image

As shown in the figure 4.26, the image is caught by the CV2 function of the OpenCV library from the computer camera. Path information is then obtained to save a picture.

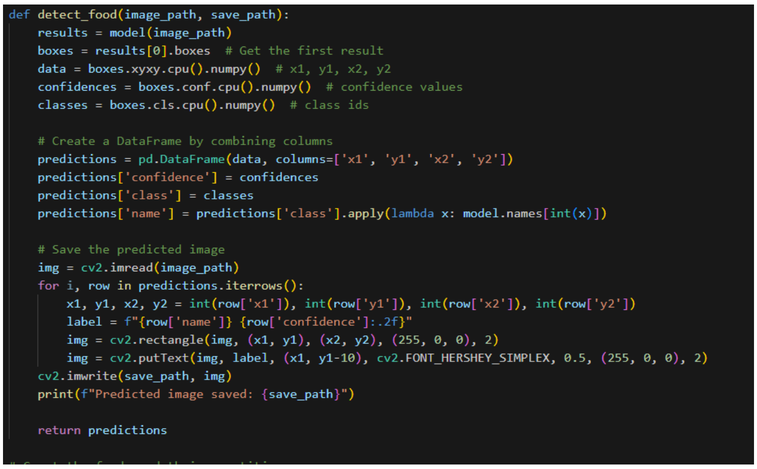


```
def capture_image(camera_index, save_path):
    cap = cv2.VideoCapture(camera_index)
    if not cap.isOpened():
        print(f"Failed to open camera {camera_index}")
        return False
    ret, frame = cap.read()
    if ret:
        cv2.imwrite(save_path, frame)
        print(f"Image captured and saved: {save_path}")
    cap.release()
    return ret
```

Figure 4.26 : Capture_image function

4.9.3 Detection of tray images and recording the estimated image

The "detect_food" function employs a model based on the image of the tray captured using the Capture_images software to estimate the products purchased. In the Predictions Dataframe, the coordinates of the detected objects, confidence scores and the class to which they belong are retained. Subsequently, the food, drinks and desserts identified in the image are delineated by rectangles and accompanying labels. The predicted images serve to record the estimates made by the algorithm. Furthermore, these images can be employed as a data source for the development of the model. Python code of these functions is given in the figure 4.27



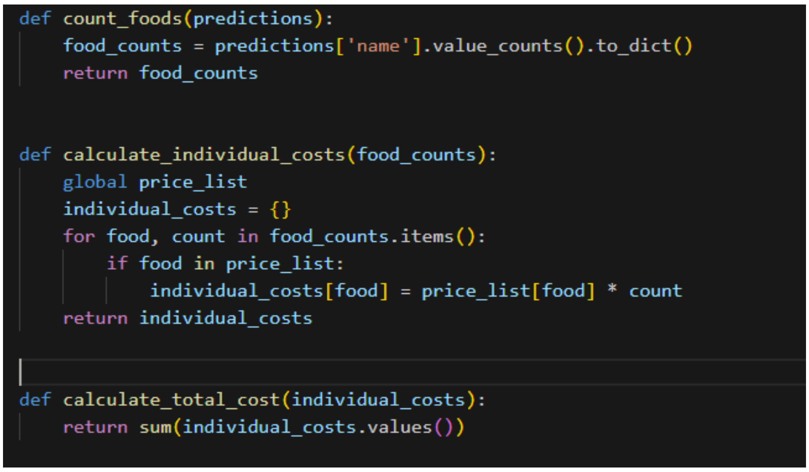
```python
def detect_food(image_path, save_path):
    results = model(image_path)
    boxes = results[0].boxes  # Get the first result
    data = boxes.xyxy.cpu().numpy()  # x1, y1, x2, y2
    confidences = boxes.conf.cpu().numpy()  # confidence values
    classes = boxes.cls.cpu().numpy()  # class ids

    # Create a DataFrame by combining columns
    predictions = pd.DataFrame(data, columns=['x1', 'y1', 'x2', 'y2'])
    predictions['confidence'] = confidences
    predictions['class'] = classes
    predictions['name'] = predictions['class'].apply(lambda x: model.names[int(x)])

    # Save the predicted image
    img = cv2.imread(image_path)
    for i, row in predictions.iterrows():
        x1, y1, x2, y2 = int(row['x1']), int(row['y1']), int(row['x2']), int(row['y2'])
        label = f"{row['name']} {row['confidence']:.2f}"
        img = cv2.rectangle(img, (x1, y1), (x2, y2), (255, 0, 0), 2)
        img = cv2.putText(img, label, (x1, y1-10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 0, 0), 2)
    cv2.imwrite(save_path, img)
    print(f"Predicted image saved: {save_path}")

    return predictions
```

Figure 4.27 : Detect_food function

4.9.4 Cost accounting

As illustrated in Figure 4.28, the "count_foods", "calculate_individual_costs" and "Calculate_total_Cost" functions employ the use of functions to ascertain the number of meals on the tray, the cost of each unit and the total cost. These functions, respectively, operate as follows:

The initial function computes the products and total traditions in the tray using the Dataframe designated as the predic Dataframe, the output of the "detect_food" function. The second function accepts the "food_counts" dictionary, which was created in the preceding function. In the "individual_costs" list, the information generated during the calculation process is stored, taking into account the number of each food item. This list is the output of the function. The third function calculates the total price information of all the foods received in this list and returns this value in numerical form.

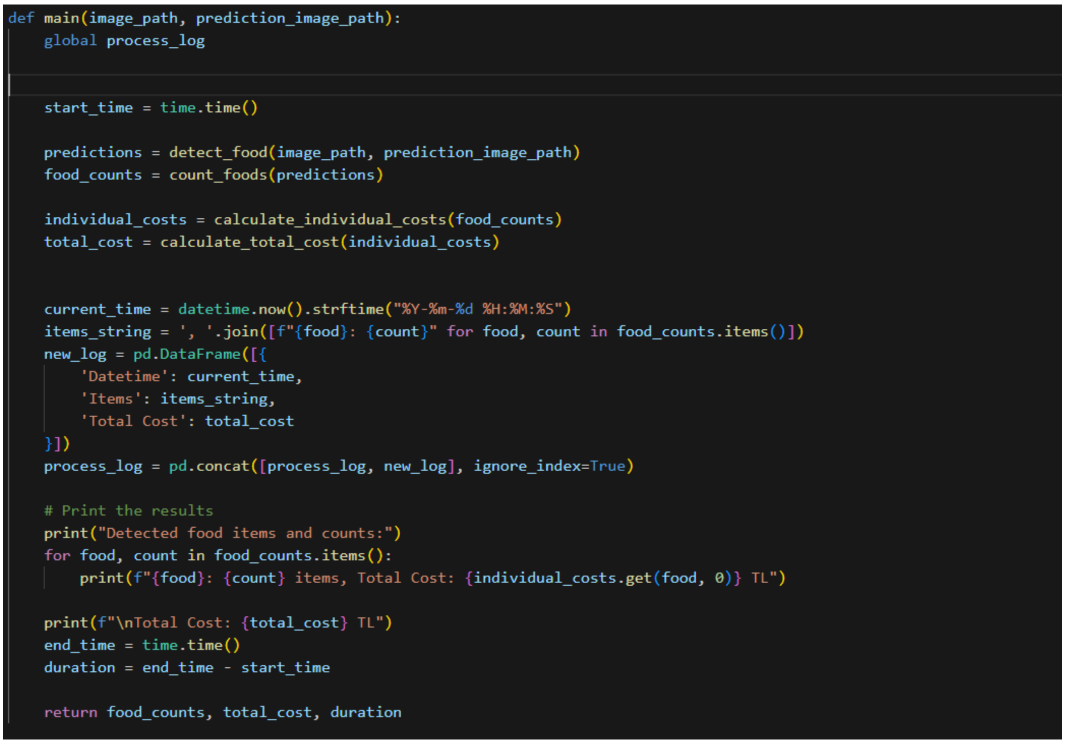


```python
def count_foods(predictions):
    food_counts = predictions['name'].value_counts().to_dict()
    return food_counts


def calculate_individual_costs(food_counts):
    global price_list
    individual_costs = {}
    for food, count in food_counts.items():
        if food in price_list:
            individual_costs[food] = price_list[food] * count
    return individual_costs


def calculate_total_cost(individual_costs):
    return sum(individual_costs.values())
```

Figure 4.28 : Cost functions.

4.9.5 The main function

This function is the primary function in which the sub-functions mentioned in the preceding sections are executed. Before the sub-functions start to work, the current time is recorded as “start_time”. As illustrated in Figure 4.29, the sub-functions are then performed in sequence. Then, "process_log" dataframe is added to the date, the products received and the total price are added. Python cell outputs are created by writing print functions. The current time is recorded with the time function. The process time is calculated by end_time - start_time. This calculates the time spent on the food recognition process.



```python
def main(image_path, prediction_image_path):
    global process_log

    start_time = time.time()

    predictions = detect_food(image_path, prediction_image_path)
    food_counts = count_foods(predictions)

    individual_costs = calculate_individual_costs(food_counts)
    total_cost = calculate_total_cost(individual_costs)


    current_time = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    items_string = ', '.join([f"{food}: {count}" for food, count in food_counts.items()])
    new_log = pd.DataFrame([{
        'Datetime': current_time,
        'Items': items_string,
        'Total Cost': total_cost
    }])
    process_log = pd.concat([process_log, new_log], ignore_index=True)

    # Print the results
    print("Detected food items and counts:")
    for food, count in food_counts.items():
        print(f"{food}: {count} items, Total Cost: {individual_costs.get(food, 0)} TL")

    print(f"\nTotal Cost: {total_cost} TL")
    end_time = time.time()
    duration = end_time - start_time

    return food_counts, total_cost, duration
```

Figure 4.29 : Main function.

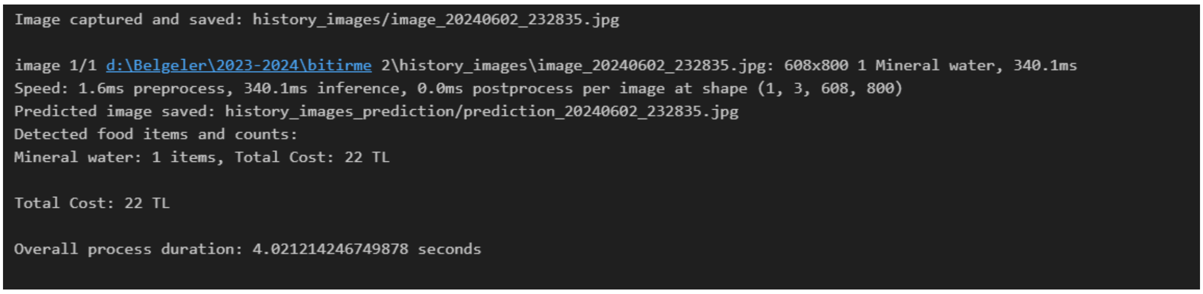
4.9.6 Operation and output of the application

In this section, the detection of food items will be carried out utilising the aforementioned functions. Figure 4.30 illustrates the output of the recognition process. In the figure 4.31, there is a labelized version of the estimated image of the above mentioned output.

Furthermore, the original image and the file paths for the estimated versions created by the model are specified and created. The processing time is appended to the process_log data frame. Subsequently, the data from the data frame is saved to the

Purchase History file. The process_log data frame is then reset and prepared for the next recognition process.

These operations permit the continuous detection of food on the tray. Finally, the Purchase History data frame is converted to an Excel file. With this data in this excel file, the restaurant is able to ascertain which meals are more popular based on historical data and which times of day. Consequently, a descriptive analysis can be conducted on the dishes received. Furthermore, the predictive analysis employed by the restaurant enables the estimation of the optimal quantity of dishes to be consumed at a given time, thereby preventing food wastage. Consequently, the restaurant's expenditure on waste is reduced.



```
Image captured and saved: history_images/image_20240602_232835.jpg

image 1/1 d:\Belgeler\2023-2024\bitirme 2\history_images\image_20240602_232835.jpg: 608x800 1 Mineral water, 340.1ms
Speed: 1.6ms preprocess, 340.1ms inference, 0.0ms postprocess per image at shape (1, 3, 608, 800)
Predicted image saved: history_images_prediction/prediction_20240602_232835.jpg
Detected food items and counts:
Mineral water: 1 items, Total Cost: 22 TL

Total Cost: 22 TL

Overall process duration: 4.021214246749878 seconds
```

Figure 4.30 : Output of process.

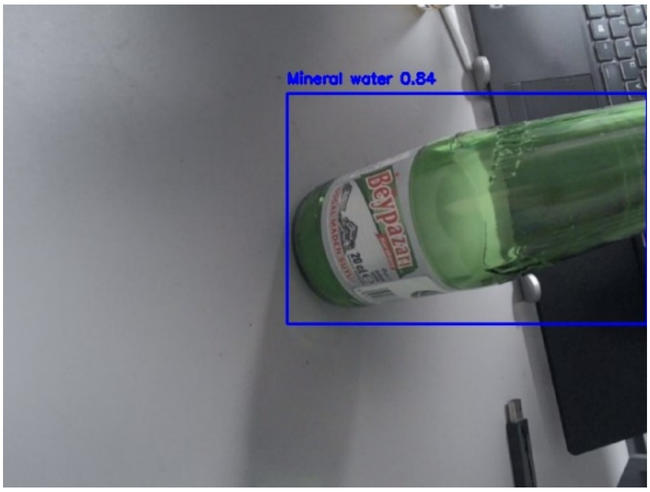


Figure 4.31 : Model prediction.

5. SYSTEM SIMULATION

We can easily see the benefits of the system we have developed by analyzing its effects on the restaurant we work with. For this purpose, a simulation study of the old system and the new system was carried out. While creating the simulation, information obtained from observations in the restaurant, work studies, customer surveys and interviews with the workplace were used.

The first scenario in figure 6.1 shows the current state of the system. Customers log in to the system and decide to stay or leave the restaurant based on the queue length. The remaining customers decide on their meals from the menu, select the dishes, request them to be placed on their tray, and proceed to the payment section. Then it logs out of the system.

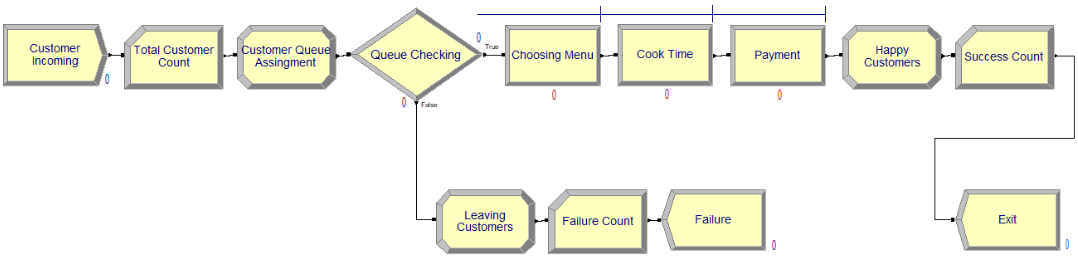


Figure 5.1 : Current state of the system.

When we transform the manual payment system into a new one using image processing and artificial intelligence model, the change occurs in the payment part of the simulation shown in the figure 6.2.

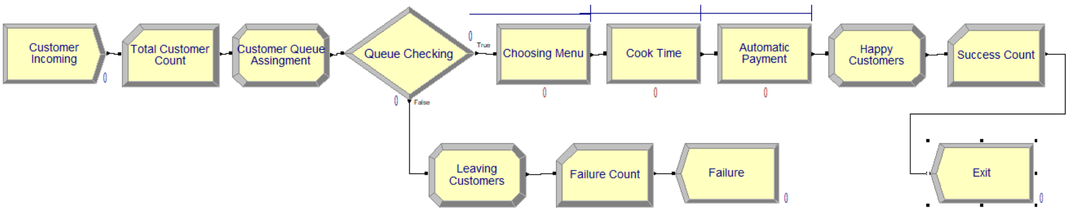


Figure 5.2 : New system in simulation.

The frequency of customer visits was calculated by taking into account the average daily hourly density of the restaurant we work with. As seen in the image below, busy times and less busy times affect customer arrival times.

In the figure 6.3 below, you can see the customer distribution we used when creating the simulation. This distribution, which is directly proportional to restaurant density, can accurately reflect real-life insights because it contains randomness. The working hours of the restaurant, 8 am and 8 pm, were also preserved when setting up the simulation.

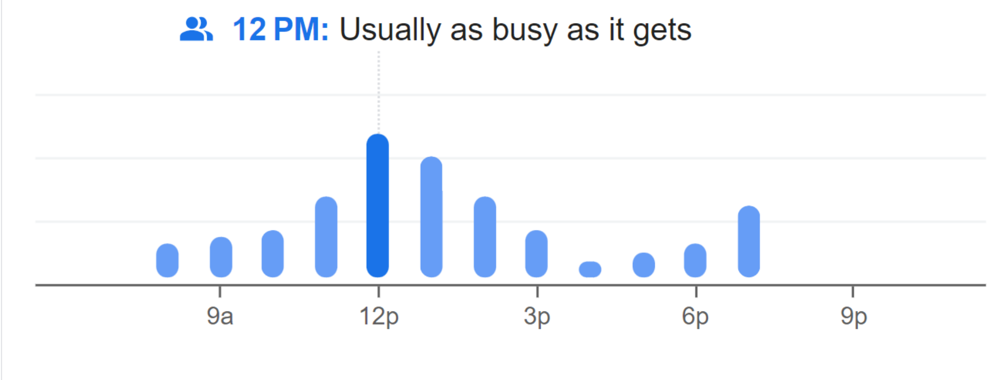
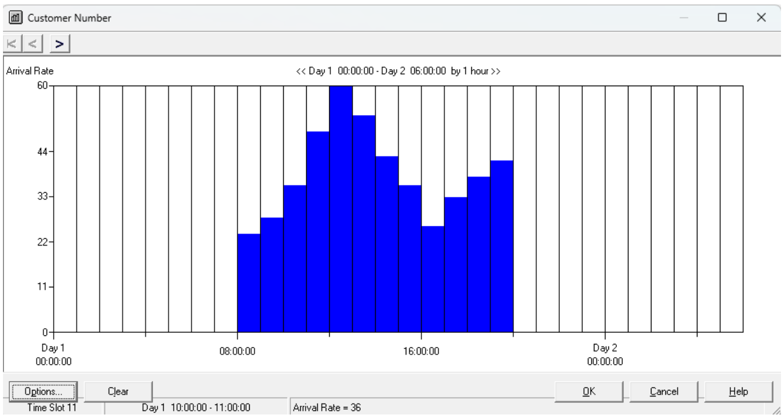


Figure 5.3 : Customer arrival time distribution over the day.

In the image below, you can see the customer distribution we used when creating the simulation. This distribution, which is directly proportional to restaurant density, can accurately reflect real-life insights because it contains randomness. The working hours of the restaurant, 8 am and 8 pm, were also preserved when setting up the simulation.



Time Slot 11 | Day 1 10:00:00 - 11:00:00 | Arrival Rate = 36

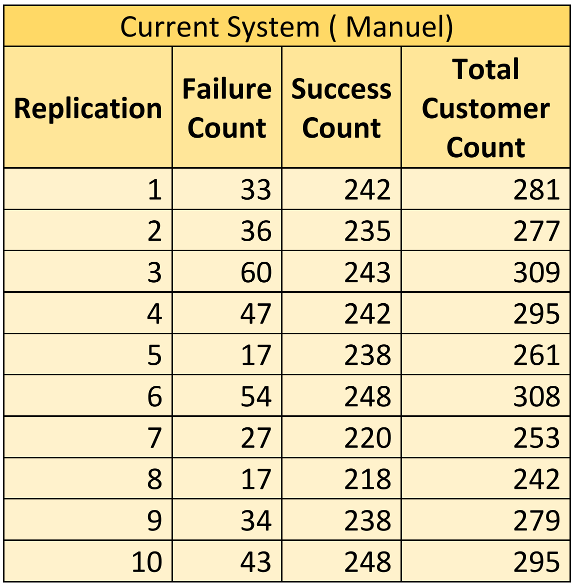
Figure 5.4 : Customer arrival time distribution in the simulation software.

According to the survey we conducted with 300 customers of the restaurant, we analyzed the customers' decision whether to stay in the restaurant or not based on the queue length. After obtaining the customer tolerances, we calculated their expected values and it came out to be 8.1. This means that the customer entering the restaurant at that time prefers to enter the queue and buy food if there are less than 8 people, and if there are more than 8 people, he prefers to leave the queue and leave the system.

The times for choosing a meal from the menu and receiving the meal were also calculated through time studies conducted in the restaurant and added to the simulation. The expected durations of the payment system in its old state and its new state were also calculated and added to the model.

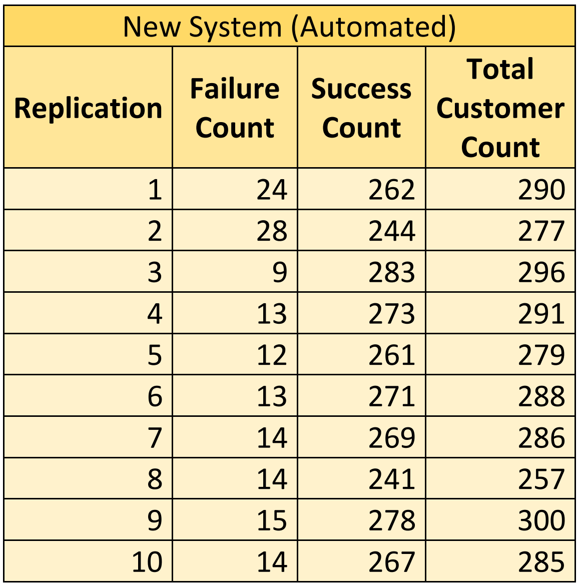
As it will be noticed in the simulation outputs better, the density in the internal flow causes customer losses. We timed the food payment to analyze in more detail the waiting in the payment system, which is the important root cause of these cash register queues. We divided these processes into two stages in the manual system and the automatic system. In the manual system, when the time for the cashier to recognize the food and enter it into the system is called t1, and the time for the person to pay is called t2, t1+t2 is defined as the total time spent at the cash register. Measurements were made in the restaurant to determine what kind of distribution these times fit into and to use them in the simulation. Likewise, in the new system, the camera's recognition of the food on the tray and transferring it to the automatic payment system are named as t1' and t2', respectively. The model was run many times and an average time was determined.

The results obtained after running the simulation 10 times are as in the table below. Table 6.1 shows the total number of customers, failure count and success count in the current manual system. Table 6.2 shows the outputs from the newly installed system.



Current System (Manuel)			
Replication	**Failure Count**	**Success Count**	**Total Customer Count**
1	33	242	281
2	36	235	277
3	60	243	309
4	47	242	295
5	17	238	261
6	54	248	308
7	27	220	253
8	17	218	242
9	34	238	279
10	43	248	295

Table 5.1 : Results after running first simulation scenerio



New System (Automated)			
Replication	Failure Count	Success Count	Total Customer Count
1	24	262	290
2	28	244	277
3	9	283	296
4	13	273	291
5	12	261	279
6	13	271	288
7	14	269	286
8	14	241	257
9	15	278	300
10	14	267	285

Table 5.2 : Results after running second simulation scenerio

5.1 Analysis of Results

The analysis of customer flow and payment systems reveals significant differences between the manual cashier system and the automated payment system. Under the manual cashier system, the average number of customers processed per replication was 280.0, with a standard deviation (SD) of 22.27. The system experienced a notable failure count, with an average of 36.8 customers abandoning the queue (SD = 13.59). Consequently, the average number of successful transactions was 241.2, with an SD of 8.66, indicating considerable variability in performance and customer satisfaction.

In contrast, the automated payment system demonstrated improved results. The average customer throughput per replication increased slightly to 283.9 (SD = 14.12). Importantly, the failure count saw a significant reduction, with an average of only 15.6 customers abandoning the queue (SD = 5.93). This enhancement in queue management translated to a higher success count, averaging 264.9 successful transactions (SD = 13.24). The data suggests that the automated payment system is more efficient and effective, with lower variability and improved customer retention compared to the manual cashier system.

5.2 Future Predictions

Based on the simulation data, the manual cashier system results in an average of 36.8 customers abandoning the queue each day. Over the course of a year, this amounts to a significant loss:

36.8 customers/day × 365 days/year = 13,432 customers/year

This projected yearly loss highlights the critical impact of queue abandonment on customer retention and underscores the importance of implementing an automated payment system to enhance operational efficiency and customer satisfaction.

It should not be forgotten that this system was prepared specifically for the restaurant we work with and with their data. The system that operates in this way can be customized for each restaurant by differentiating variables such as 2 food employee resources, a cashier, customer arrival times and waiting times that comply with the above-mentioned distributions.

6. CONCLUSION AND SUGGESTIONS

6.1 Conclusion

In recent years, there has been a significant shift in daily habits and business operations, especially with the increasing prevalence of contactless services globally. Our research focuses on solving the problem of long wait times in self-service restaurant checkout lines through the innovative use of computer vision technology.

The development process began with an extensive literature review on artificial intelligence, deep learning architectures, and computer vision; It formed the basis for understanding the technological situation and determining appropriate methodologies for our project. System analysis was then conducted to define system boundaries, identify the project's stakeholders, and identify areas for improvement.

The methodology section detailed the data collection, image labeling, data augmentation, and preprocessing steps. We used the You Only Look Once (YOLO) algorithm for object detection due to its efficiency and real-time processing capabilities. The model was then run into simulations to test its effectiveness and efficiency.

The implementation of a deep learning-based image detection system has shown significant improvements in operational efficiency and customer satisfaction. The automatic checkout system performed better than the manual cashier system; This was further proven by simulation results showing increased customer volume and reduced queuing. This system not only simplifies the payment process, but also improves overall service quality by minimizing human error. From a managerial perspective, the system enables data-based decision-making by providing valuable information on customer behavior and operational bottlenecks. In addition, the scalability and feasibility of the system allows it to be adapted to different restaurant installations, which makes the system usable in many restaurants.

Implementing automatic checkout in self-service restaurants has significant social and environmental impacts. From a social perspective, the system improves the dining experience by reducing wait times and increasing customer satisfaction. This improvement is aligned with United Nations Sustainable Development Goal (SDG) 9: Industry, Innovation and Infrastructure, which aims to build resilient infrastructure, promote inclusive and sustainable industrialization and foster innovation. From an environmental perspective, the automated system contributes to reducing food waste. Precise detection and pricing of items on customers' trays can lead to more accurate inventory management and less food overproduction, supporting SDG 12: Responsible Consumption and Production. This goal emphasizes the need to ensure sustainable consumption and production models, which are vital in reducing the environmental impacts of the food industry. Additionally, as automation reduces the need for intensive human labor and associated resources, the reduction in manual processes also leads to reductions in energy consumption and carbon footprints, thus helping to reduce greenhouse gas emissions more efficiently, indirectly meeting Sustainable Development Goals 13: Climate Action. supports. In conclusion, a deep learning-based image detection system for payment optimization in restaurants is a robust, scalable and ethical solution that offers significant managerial, social and economic benefits. Its implementation not only streamlines the payment process, but also aligns with ethical principles such as virtue, justice, duty and utilitarianism, ensuring a fair and efficient service delivery model.

6.2 Suggestions

The dataset generated during the training phase of the model primarily comprises photographic images of meals purchased by customers at the target restaurant. It is possible that this model may not be an effective model for detecting dishes in different restaurants. Should the system proposed in this study be implemented in different restaurants, it would be necessary to create a dataset comprising the products of that restaurant. It is recommended that photographs be taken by combining the food, drinks, or desserts on the tray, with the objective of creating a data set. One type of food should be placed in different positions on the tray. Otherwise, there is a possibility of reducing the success of the model as the food belonging to that species in the dataset is always found in the same location, which may cause overfitting.

Furthermore, in order to circumvent the error encountered in this study, whereby objects that are not intended to be detected on the tray (such as forks and spoons) are classified as a separate entity (mineral water example in this study), the following should be done: Images of trays on which the price information is not required should be incorporated into the dataset without any form of labelling. This approach serves to minimise the likelihood of the model generating erroneous test results.

The model employed in this study demonstrated the capacity to successfully identify products comprising half portions. Nevertheless, there is a possibility that the model may incorrectly identify both full-portion and half-portion versions of the same dish. In restaurants where this system can be used, there is a significant difference between the plates used for half portions and full portions, which can prevent the model from making incorrect determinations. For instance, light-coloured plates can be employed for full portions, while dark-coloured plates can be utilised for half portions.

In this study, minor portions of dishes from an additional tray were discernible on the periphery of the images utilized for the assessment of the model. The model was also able to detect these products. Although this demonstrates that the model is functioning as intended, it is capable of identifying products that do not belong in a specific tray. This results in the product that the customer did not purchase being recorded in the payment system. To circumvent this issue, it is essential to ascertain the angles that the camera will observe when capturing the tray image. It is recommended that customers awaiting payment position their trays outside the aforementioned angle. To guarantee this behaviour, it is necessary to place a warning strip at the limit of the angles visible to the camera.

A potential risk to security and customer interaction with the system is that customers might attempt to obscure the camera's view with their hands or conceal certain food items to pay less. To mitigate such issues, it is feasible to train the model specifically to recognize and prevent these behaviors. By enhancing the system's capabilities in this manner, we can ensure more accurate and secure transactions, maintaining the integrity of the automated payment process.

REFERENCES

Agarwal, R., Bansal, N., Choudhury, T., Sarkar, T., & Ahuja, N. J. (2022, September). IndianFood-7: Detecting Indian Food Items Using Deep Learning-Based Computer Vision. In *International Conference on Advances and Applications of Artificial Intelligence and Machine Learning* (pp. 9-22). Singapore: Springer Nature Singapore.

Aguilar, E., Remeseiro, B., Bolaños, M., & Radeva, P. (2018). Grab, pay, and eat: Semantic food detection for smart restaurants. *IEEE Transactions on Multimedia*, *20*(12), 3266-3275.

An, J. Y., Lee, C. C., Bae, D. E., & Lee, S. H. (2020). A Study on Use of Untact Service: Based on Kiosk Case. *The Journal of Internet Electronic Commerce Research*, *20*(4), 49-73.

Beddiar, D. R., Oussalah, M., Muhammad, U., & Seppänen, T. (2023). A deep learning based data augmentation method to improve COVID-19 detection from medical imaging. *Knowledge-Based Systems*, *280*, 110985. https://doi.org/10.1016/j.knosys.2023.110985

Blöcher, K., & Alt, R. (2021). AI and robotics in the European restaurant sector: Assessing potentials for process innovation in a high-contact service industry. *Electronic Markets*, *31*, 529-551.

Brosnan, T., & Sun, D. W. (2004). Improving quality inspection of food products by computer vision—a review. *Journal of food engineering*, *61*(1), 3-16.

Chumachenko, K., Gabbouj, M., & Iosifidis, A. (2022). Object detection and tracking. *Deep Learning for Robot Perception and Cognition*, 243–278.

Dixit, M., Tiwari, A., Pathak, H., & Astya, R. (2018). An overview of deep learning architectures, libraries and its applications areas. In *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)* (pp. 293-297). IEEE.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2009). Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, *32*(9), 1627-1645.

Ghosh, S., Das, N., Das, I., & Maulik, U. (2019). Understanding deep learning techniques for image segmentation. *ACM computing surveys (CSUR)*, *52*(4), 1-35.

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 1440-1448).

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In

Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).

Guo, W., Qiao, S., Zhao, C., & Zhang, T. (2023). Defect detection for Industrial Neutron radiographic images based on modified Yolo Network. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, *1056*, 168694. https://doi.org/10.1016/j.nima.2023.168694

Hussain, M. (2023). YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection. *Machines*, *11*(7), 677.

Jakhar, D., & Kaur, I. (2020). Artificial intelligence, machine learning and deep learning: definitions and differences. *Clinical and experimental dermatology*, *45*(1), 131-132.

Jiang, H., Diao, Z., & Yao, Y. D. (2022). Deep learning techniques for tumor segmentation: a review. The Journal of Supercomputing, 78(2), 1807-1851.

Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2022). A Review of Yolo algorithm developments. *Procedia Computer Science*, *199*, 1066-1073.

Katarzyna, R., & Paweł, M. (2019). A vision-based method utilizing deep convolutional neural networks for fruit variety classification in uncertainty conditions of retail sales. *Applied Sciences*, *9*(19), 3971.

Khan, M., Jan, B., Farman, H., Ahmad, J., Farman, H., & Jan, Z. (2019). Deep learning methods and applications. *Deep learning: convergence to big data analytics*, 31-42.

Kim, H., Hong, H., Ryu, G., & Kim, D. (2021). A study on the automated payment system for artificial intelligence-based product recognition in the age of contactless services. International Journal of Advanced Culture Technology, 9(2), 100-105.

Kumar, I., Rawat, J., Mohd, N., & Husain, S. (2021). Opportunities of artificial intelligence and machine learning in the food industry. *Journal of Food Quality*, *2021*, 1-10.

Küfeoğlu, S. (2022). Emerging Technologies. In *Emerging Technologies: Value Creation for Sustainable Development* (pp. 41-190). Cham: Springer International Publishing.

Lan, S., Wan, C., Chen, L., Jin, M., & Yu, S. (2022, October). Deep learning-based recognition of Chinese dishes in a waiterless restaurant. In *2022 16th IEEE International Conference on Signal Processing (ICSP)* (Vol. 1, pp. 390-394). IEEE.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436-444.

Li, Z., Wang, Y., Zhang, N., Zhang, Y., Zhao, Z., Xu, D., ... & Gao, Y. (2022). Deep learning-based object detection techniques for remote sensing images: A survey. *Remote Sensing*, *14*(10), 2385.

Marın, J., Biswas, A., Ofli, F., Hynes, N., Salvador, A., Aytar, Y., ... & Torralba, A. (2021). Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(1), 187-203.

Mohanty, S. P., Singhal, G., Scuccimarra, E. A., Kebaili, D., Héritier, H., Boulanger, V., & Salathé, M. (2022). The food recognition benchmark: Using deep learning to recognize food in images. *Frontiers in Nutrition*, *9*, 875143.

Ramla, M., Sangeetha, S., & Nickolas, S. (2022). Towards building an efficient deep neural network based on YOLO detector for fetal head localization from ultrasound images. *Edge-of-Things in Personalized Healthcare Support Systems*, 137–156.

Reis, H. C., & Turk, V. (2022). Covid-DSNet: A novel deep convolutional neural network for detection of coronavirus (SARS-COV-2) cases from CT and chest X-ray images. *Artificial Intelligence in Medicine*, *134*, 102427.

Russell, S. J., & Norvig, P. (2010). *Artificial intelligence a modern approach*. London.

Shandilya, S. K., Srivastav, A., Yemets, K., Datta, A., & Nagar, A. K. (2023). Yolo-based segmented dataset for drone vs. Bird Detection for deep and machine learning algorithms. *Data in Brief*, *50*, 109355. https://doi.org/10.1016/j.dib.2023.109355

Sultana, F., Sufian, A., & Dutta, P. (2020). A review of object detection models based on convolutional neural network. *Intelligent computing: image processing based applications*, 1-16.

Turing, A. M. (2009). *Computing machinery and intelligence* (pp. 23-65). Springer Netherlands.

Yapici, M. M., Tekerek, A., & TOPALOĞLU, N. (2019). Literature review of deep learning research areas. *Gazi Mühendislik Bilimleri Dergisi*, *5*(3), 188-215.

Zhang, Y., Deng, L., Zhu, H., Wang, W., Ren, Z., Zhou, Q., ... & Wang, S. (2023). Deep Learning in Food Category Recognition. *Information Fusion*, 101859.

Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, *30*(11), 3212-3232.

Zhiqiang, W., & Jun, L. (2017, July). A review of object detection based on convolutional neural network. In *2017 36th Chinese control conference (CCC)* (pp. 11104-11109). IEEE.

Curriculum Vitae

Name Surname : Hilmi Öztürk

Date of Birth and Place : Yuregir/Adana 09.08.2000

E-mail : ozturkhilmi43@gmail.com

EDUCATION STATUS:

- **Bachelor's degre** : Currently, Istanbul Technical University, Faculty of Business Administration, Industrial Engineering

PROFESSIONAL EXPERIENCE:

- He worked at TAI in the production planning and control department for 20 days in 2022.
- He did an internship in the ERP consultancy department at IFS Türkiye for 25 working days in 2022.
- He has been working part time as an intern in the business development department at QNB Finansbank since March.

Curriculum Vitae

Name Surname	: Begüm Nur Okur
Date of Birth and Place	: Fatih/İstanbul 05.07.2001
E-mail	: begumnurokur45@gmail.com

EDUCATION STATUS:

- **Bachelor's degree** : Currently, Istanbul Technical University, Faculty of Business Administration, Industrial Engineering

PROFESSIONAL EXPERIENCE:

- She worked at Schneider Electric in the supply chain department for one year from 2022 to 2023.
- She did an internship in the Martur Fompak International Romania plant in 2023.