Project Summary

Project No. 606 - Bar Ilan University

Software Development

Café recommendation system for coffee enthusiastic

<u>Submitted By:</u>

Ahigad Genish ID: 316228022 email: ahigad.genish@gmail.com

Omer Sela ID: 316539535 email: omersela1010@gmail.com

<u>Project Supervisors</u>:

Professor Gelles Ran

Professor Cohen Izack

<u>Submission Date</u>: 16/09/2024

Abstract

Choosing the right coffee shop can be challenging, especially for those who value quality over convenience.

While existing recommendation systems rely heavily on general user ratings, they often fail to cater to the specific preferences of coffee aficionados.

Our project addresses this problem by developing a website that helps users identify the best coffee shops based on personalized filters such as location (e.g., a city like Tel Aviv), coffee type (e.g. Cappuccino, Espresso or Black Coffe), and cup type (Take away or dine in).

The system rates coffee shops by analyzing features related to coffee quality, such as the presence of crema and the way it is served, providing tailored recommendations for true coffee lovers.

The primary components of the system include user input handling, image detection, machine learning for feature classification, and integration with Google Maps API as we will present here.

Table of Contents

Introduction
.... 4

Methodology
.... 9

Implementation
.... 24

Results
.... 34

Conclusion
.... 60

References
.... 62

Introduction

Background and Motivation

In today's digital age, recommendation systems have become integral to our daily lives, guiding our choices in everything from entertainment to dining. Platforms like Netflix and Google Maps rely heavily on user-generated ratings to suggest movies, restaurants, and other services. However, while these systems are convenient, they often struggle to provide accurate recommendations tailored to specific tastes and preferences. This is particularly true in niche areas, such as coffee culture, where the quality of the product can significantly vary depending on factors that general users may not fully appreciate.

Coffee enthusiasts, for instance, seek more than just a popular spot with good reviews; they value the craftsmanship behind each cup, the type of coffee used, the presence of crema, and the expertise of the barista. Unfortunately, current recommendation systems tend to homogenize user ratings, giving equal weight to all reviews regardless of the reviewer's knowledge or experience with coffee. This approach can lead to biased recommendations that favor popular opinions over quality, leaving true coffee lovers underserved.

Motivated by this gap, our project aims to develop a more refined and specialized recommendation system specifically designed for coffee aficionados. By focusing on key coffee quality indicators—such as the presence of crema, the way the coffee is served, and the visual characteristics of the coffee itself—our system will provide recommendations that better align with the discerning tastes of coffee enthusiasts. This project leverages machine learning and image detection technologies to analyze user-generated photos and reviews, creating a sophisticated tool that offers more reliable and relevant suggestions for those who take their coffee seriously.

Problem Statement

The widespread use of recommendation systems in various domains has revolutionized how users discover and select services. However, existing systems, such as those used by platforms like Google Maps, primarily rely on aggregated user ratings without considering the expertise or specific preferences of individual reviewers. This generalized approach often results in recommendations that cater to the average user, overlooking the needs of niche communities with more refined tastes.

In the context of coffee, this presents a significant problem. True coffee enthusiasts value more than just a café's popularity—they seek establishments that demonstrate a high level of craftsmanship, using professional equipment and precise techniques to produce a superior cup of coffee. Current recommendation systems fail to adequately address these requirements, as they do not account for the nuanced factors that define coffee quality, such as the type of coffee used, the presence of crema, or the skill with which the coffee is served.

This project addresses the critical gap in existing recommendation systems by developing a specialized platform that caters to coffee aficionados. The system will analyze key indicators of coffee quality through machine learning and image detection techniques, offering recommendations based on factors that matter most to serious coffee drinkers. By doing so, the system will provide more accurate and meaningful recommendations, helping users find coffee shops that meet their high standards and preferences.

Project Objectives

The primary objective of this project is to develop a specialized recommendation system that accurately identifies and suggests high-quality coffee shops tailored to the preferences of coffee enthusiasts. To achieve this, the project will focus on the following specific goals:

Develop a User-Friendly Website:

Create an intuitive interface that allows users to easily select a location (e.g., a city like Tel Aviv) and apply specific filters, such as coffee type and order preference (Take away or Dine-in).

Ensure that the website provides a seamless user experience, enabling users to quickly find coffee shops that meet their preferences.

Integrate with Google Maps API:

Utilize the Google Maps API to gather images and data of nearby coffee shops based on the user's selected location.

Ensure that the system can retrieve and process relevant visual data content from Google Maps.

Implement Advanced Image Detection Techniques:

Develop and integrate an image detection module that can analyze photos of coffee from user submissions on Google Maps.

Train the system to identify key quality indicators, such as the type of coffee used, the presence of crema, and the way coffee is served.

Apply Machine Learning Models for Quality Assessment:

Train and fine-tune machine learning models to evaluate and classify coffee quality based on visual features extracted from images.

Combine the results from image analysis with user preferences to generate personalized coffee shop recommendations.

Provide Reliable and Accurate Recommendations:

Enhance the recommendation system to provide high-quality suggestions by applying weighted evaluations to the detected features of coffee images and accurately rating them.

Test and Validate the System:

Conduct thorough testing of the recommendation system in various locations, ensuring its effectiveness in identifying and recommending high-quality coffee shops.

Validate the system's performance by comparing its recommendations with user's feedback.

Document and Illustrate the System:

Create comprehensive documentation that details the system's architecture, functionality, and implementation process.

Include diagrams and visual representations of the system's components and workflow to facilitate understanding and future development.

By fulfilling these objectives, the project will deliver a robust and reliable recommendation system that significantly improves the coffee shop discovery experience for discerning coffee lovers.

Scope and Limitations

The scope of this project is to develop a website that rates coffee shops based on various quality features. The key aspects include:

Geographical Focus:

The project will primarily concentrate on coffee shops located within specific cities, such as Tel Aviv and Netanya. Extending the project to cover additional cities is not included due to the costs associated with using the Google API, which charges for retrieving data and images from different locations.

Quality Features:

The system will evaluate coffee shops based on certain coffee quality features, such as coffee type (Cappuccino, Espresso, Black Coffee), Type of cup (Take away \ Dine – in), crema presence, and serving style. While this provides valuable insights, it may not encompass all potential user preferences or specific attributes that some users might consider important.

Data Sources:

The system relies on data gathered from Google Maps and user-generated content. This approach may lead to limitations in data accuracy and completeness, as the information provided is dependent on the availability and reliability of these sources.

Methodology

System Design

The system consists of several key components that interact to deliver high-quality recommendations. These include a web-based user interface, a machine learning model for image classification, an image detection module, and the Google Maps API. The system's architecture is designed to be modular, allowing for future enhancements and expansions.

System Components

User Interface (UI):

Purpose: The UI allows users to input their location and preferences, then receive personalized coffee shop recommendations.

Technology: The website is built using modern web technologies, ensuring a responsive and user-friendly experience.

Functionality: Users can specify locations, apply filters based on coffee type, and view recommended coffee shops. The system also supports displaying additional details like pricing, crowd levels, and security.

Recommender System:

Core Functions: The recommender system acts as the central hub, processing user inputs, interacting with the Google Maps API, and orchestrating the flow of data through the image detection and ML model components.

Process Flow:

1. The user enters a location.

2. The recommender system queries Google Maps for nearby coffee shops.

3. Images of the coffee shops are retrieved and analyzed.

4. The ML model classifies the images, and the results are processed to generate a recommendation list.

5. The top-rated coffee shops are presented to the user.

Google Maps API Integration:

Purpose: The Google Maps API is used to obtain coffee shop images and geolocation data.

Functionality: The system sends requests to Google Maps to retrieve images of coffee shops in the specified location. These images are then passed to the image detection module for further analysis.

Image Detection Module:

Purpose: This module is responsible for identifying relevant features in the images retrieved from Google Maps.

Technology: Computer vision techniques are applied to detect specific objects that indicate the quality of coffee preparation, such as the presence of a crema on espresso.

Process Flow:

Images are fed into the detection module.

The module identifies coffee cups and other key quality indicators.

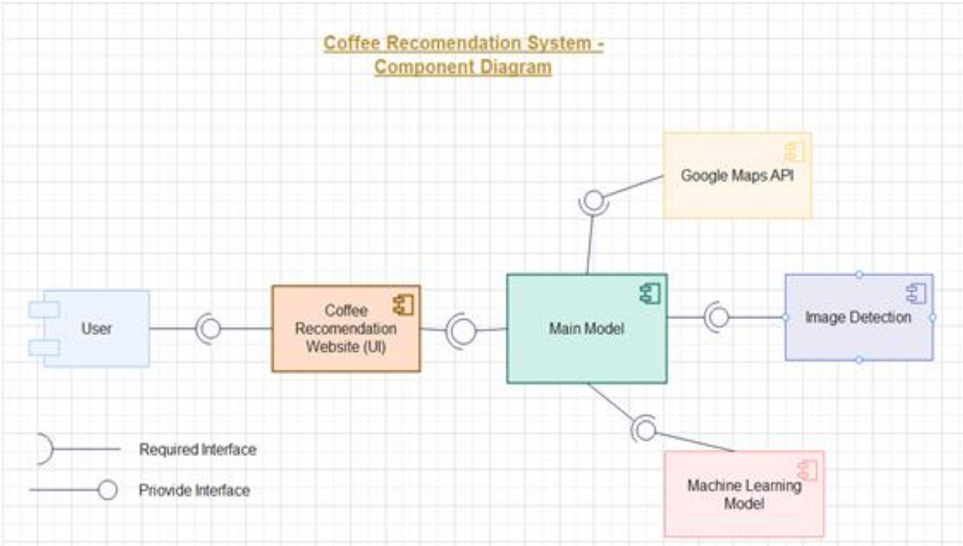
The processed images are passed to the ML model for classification.

Machine Learning Model:

Purpose: The ML model classifies coffee shop images based on their quality, such as the presence of high-quality equipment, professional preparation techniques, and overall presentation.

Technology: The model is trained on a diverse dataset of coffee shop images, enabling it to accurately classify and rate coffee quality.

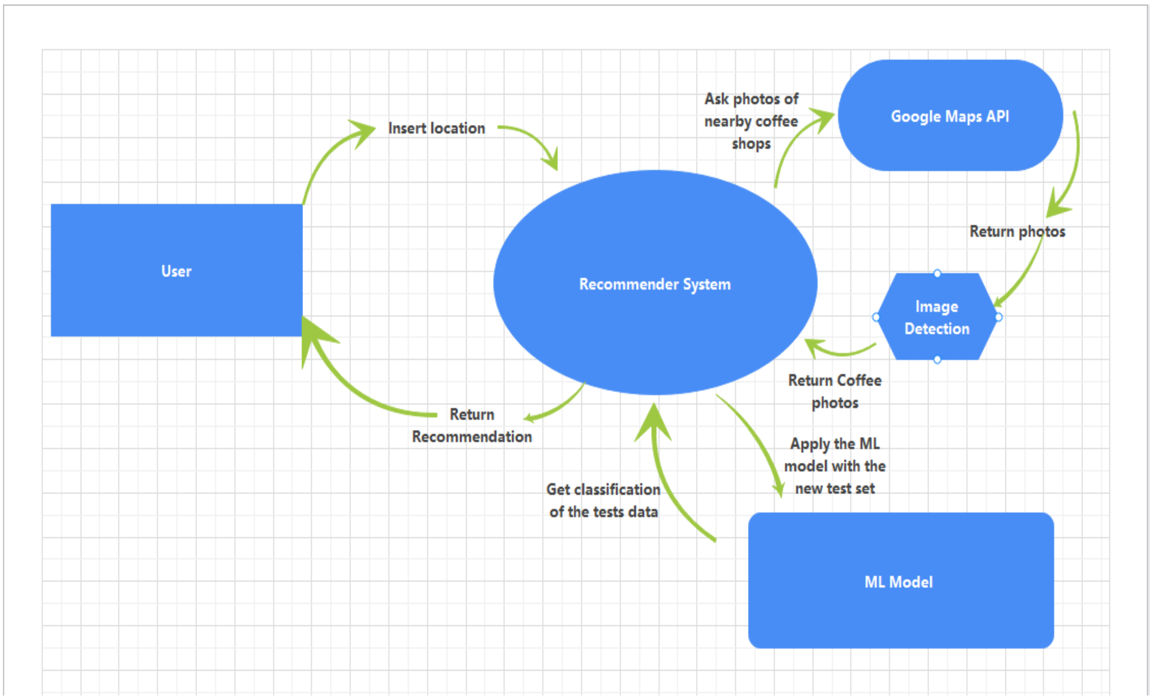
Functionality: The ML model applies a scoring mechanism to the detected features, contributing to the overall recommendation score.



Component Diagram

Data Flow and Interaction

The user initiates the interaction by entering a location into the system. The recommender system queries the Google Maps API to retrieve images of nearby coffee shops. These images are processed by the image detection module, which identifies relevant quality features. The processed data is then analyzed by the ML model to classify and rate the coffee shops. Finally, the recommender system compiles the results and returns a list of top recommendations to the user.



Context Diagram: A context diagram illustrates the interactions between the recommender system and external entities.

Data Collection

Source of Data

The data used to train the coffee shop recommendation system was primarily gathered through manual collection and the Open Images dataset. This approach allowed us to curate a highly specific and relevant dataset, focusing on the visual characteristics of coffee served in various settings.

Image Data Collection:

Manual Collection: We gathered approximately 700 images of coffee cups from various coffee shops and from google photos. These images were collected manually by visiting coffee shops and photographing the coffee, or by sourcing them from publicly available online platforms. This process allowed us to obtain a diverse set of images representing different types of coffee and presentation styles.

Open Images Dataset: In addition to manually collected images, we utilized the Open Images dataset v7 (available at https://storage.googleapis.com/openimages/web/index.html) to source additional coffee cup photos. This dataset provided us with many labeled images, which were essential for training an image detection model. This model was later used to filter the photos retrieved from the Google Maps API, ensuring that only images containing coffee cups were included in our analysis.

System Data Source:

Google Maps API: While we did not use the Google Maps API for initial data collection, it played a crucial role in the system's operation. Once the image detection model was trained using our manually collected data and the Open Images dataset, we used the Google Maps API to retrieve photos from location-based business listings. The trained model was then employed to filter these images, isolating those that contained coffee cups, which were subsequently used to evaluate coffee quality.

Data Labeling

To prepare our dataset for training the machine learning model, we manually labeled each of the 700 images with specific features that characterize the quality and presentation of the coffee.

These labels included:

<u>Type of Coffee</u>**:**

Black: Plain coffee without milk or cream.

Cappuccino: Coffee made with steamed milk foam.

Espresso: Strong black coffee made by forcing steam through ground coffee beans.

<u>Crema Presence</u>**:**

Yes: Image shows the presence of crema (a layer of foam on top of espresso).

No: Image lacks crema.

<u>Take Away or Not</u>**:**

Yes: Coffee is served in a disposable cup, indicating it is for take away.

No: Coffee is served in a ceramic cup or glass, indicating it is for dine-in.

<u>Served way quality</u>**:**

Good: Coffee is well served with additional elements like saucers, spoons, etc.

Poor: Coffee is presented without attention to detail or lacks any supplementary elements.

<u>Coffee Color:</u>

Light, Medium, Dark: Based on the visible coffee color, which can indicate different roast levels and brewing methods.

<u>Presentation Drawings (For Cappuccino):</u>

Yes: The foam on the cappuccino has artistic drawings or patterns.

No: The foam is plain without any artistic presentation.

Data Preprocessing

To ensure that our dataset was suitable for training the machine learning model, we undertook a comprehensive data preprocessing phase, which included data cleaning, filtering, and augmentation.

Data Cleaning:

Image Quality Filtering: Low-quality images, such as those that were blurry or poorly lit, were removed to ensure the dataset consisted only of high-quality images.

Duplicate Removal: Any duplicate images were identified and removed to avoid skewing the model's learning process.

Data Augmentation:

Objective: To expand the dataset from 700 images to approximately 10,257 images, we employed data augmentation techniques. This step was critical to increase the diversity of the dataset and enhance the model's ability to generalize across different visual presentations of coffee.

Techniques Used:

Rotation: We rotated images at various angles to simulate different camera perspectives.

Flipping: Images were flipped horizontally and vertically to introduce more variability in the dataset.

Scaling: Images were slightly scaled up or down to mimic variations in size and proportion.

Final Dataset:

Following these data augmentation techniques, the dataset was expanded to approximately 10,257 images. Each augmented image retained the labels assigned during the initial data collection phase, ensuring consistency and accuracy in training the model.

Image Detection

Feature Extraction

In our project, accurately identifying coffee images is crucial to evaluating coffee quality. To automate the process of detecting coffee cups within images, we employed a Convolutional Neural Network (CNN)-based model, specifically the YOLOv8 model. This model was selected for its efficiency and accuracy in object detection tasks, making it well-suited for identifying coffee cups—a critical first step in our analysis process.

Model Training

The training process involved several key steps:

Data Preparation:

We sourced images of coffee cups from the Open Images Dataset and manually labeled them to indicate the presence of coffee cups. The labeled dataset was divided into training, validation, and test sets to facilitate model training and evaluation.

A configuration YAML file was created to define the paths for training, validation, and test images, as well as the class labels (in this case, a single class: Coffee_cup).

Model Setup and Training:

We used Google Colab to train the YOLOv8 model. The setup involved installing the ultralytics package and mounting Google Drive to store the dataset and model checkpoints.

The model was initialized using a pre-trained YOLOv8 model (yolov8n.pt), which was then fine-tuned on our coffee cup dataset. The training process was configured to run for 10 epochs, with the model optimized for binary classification to distinguish between images containing coffee cups and those without.

After training, the final model weights were saved for future use in detecting coffee cups in images obtained from the Google Maps API.

Model Validation and Adjustment:

The model's performance was evaluated using the validation set, and adjustments were made to improve its accuracy. This included fine-tuning parameters and using performance metrics like precision, recall, and the F1-score to ensure the model's reliability.

Model Deployment:

Once the model was trained and validated, it was deployed to analyze images retrieved from the Google Maps API. The model was able to filter images to identify those containing coffee cups, which were then passed on for further quality assessment.

Final Output:

The trained YOLOv8 model allowed us to efficiently and accurately detect coffee cups in images, serving as the foundation for our system's ability to evaluate coffee quality based on visual features. This model was integral to filtering out irrelevant images and ensuring that only relevant coffee-related photos were analyzed for quality assessment.

Machine Learning Model

After the coffee images were meticulously labeled and augmented, the next step was to set up a machine learning model to evaluate the quality of a given coffee cup based on visual features. This model is designed to classify the coffee images into various categories that represent different quality aspects of the coffee, such as the type, presentation, and additional elements that enhance the coffee experience.

Model Features and Classification

The primary features used by the model are the pixels of the given image, which represent the visual characteristics of the coffee cup. The model is trained to classify these images into relevant categories. For example, the model might classify an image as "Cappuccino with a well-layered crema, served in a ceramic cup with a saucer and spoon." Such classifications are essential for assessing the quality of the coffee based on its appearance.

Training Process

The training process involved feeding the labeled and augmented dataset into a Convolutional Neural Network (CNN), which is particularly effective for image classification tasks. The CNN model learns to recognize patterns and features in the images that correspond to different classes, such as the type of coffee, the presence of crema, the serving method, and the presence of additional items like a spoon or saucer.

Quality Evaluation

Once the model was trained, it was used to evaluate the quality of new coffee images by classifying them into the appropriate categories. Each classification corresponds to a set of features that define the quality of the coffee. For instance, a cappuccino with a well-presented layer of crema served in a ceramic cup with a saucer and spoon would likely receive a higher quality rating than a poorly presented coffee.

Final Quality Rating

The final step in the process involves aggregating the classifications to rate the overall quality of the coffee. The model's classifications are used to determine what is most desirable for clients. This rating system allows for a comprehensive evaluation of the coffee's quality, ensuring that the best coffee shops are recommended to users based on the visual appeal and presentation of their coffee.

By combining image classification with quality evaluation, this machine learning model plays a crucial role in the system's ability to assess and rate coffee quality, providing users with reliable recommendations based on visual features.

Website Development

The website of our coffee shop recommendation platform is designed using a multiple client-server model. This approach ensures that each client, whether accessing the website from a desktop, tablet, or mobile device, can seamlessly interact with the system and receive a personalized list of coffee shop recommendations based on their preferences.

Client-Server Interaction

Client-Side Interface:

Responsive Design: The UI is developed using responsive web design principles, ensuring that it adapts to various screen sizes and resolutions. This provides an optimal user experience regardless of the device used.

Interactive Filters: Each client is presented with a set of filters that they can apply to refine their search.

The filters include:

1. **Location:** Users can select their city or area (e.g., Tel Aviv, Netanya) to find coffee shops near them.

2. **Dine-In / Takeaway:** Users can choose between dine-in or takeaway options depending on their preference.

3. **Type of Coffee:** Users can filter coffee shops based on the type of coffee they prefer, such as espresso, cappuccino, or black coffee.

Real-Time Filtering: As users select different filter options, the client-side interface dynamically updates the list of coffee shops in real-time. This immediate feedback is achieved through asynchronous requests to the server, ensuring a smooth and responsive user experience.

Key UI Components

Welcome Page:

The platform greets users with a welcoming dashboard that prominently features a "Get Started" button. This dashboard acts as the central navigation hub, guiding users smoothly to other sections of the website. The welcome page is thoughtfully designed to include essential elements such as copyright information and direct links to various subpages, ensuring users can easily access all necessary information and functionality.

Home Page:

The Home page serves as the primary interactive space where users can input their preferences through a set of filters. These filters allow users to specify criteria such as location, dine-in/takeaway options, and coffee type. Once the filters are applied, users can immediately view a sorted list of recommended coffee shops that best match their selections. This page is the core of the user experience, offering a seamless interface for discovering top coffee spots.

About Page:

The About page provides a comprehensive overview of the website, detailing its purpose, the features it offers, and the benefits it provides to users. This section is designed to clearly communicate the platform's value proposition, helping users understand how the website can enhance their coffee shop selection process. It serves as an informative resource, encouraging users to explore the website's features in more depth.

<u>Server-Side Processing:</u>

Request Handling: When a client applies filters, the UI sends a request to the server containing the selected criteria. The server then processes these criteria to generate a list of recommended coffee shops.

Database Queries: The server queries the database to retrieve coffee shops that match the client's preferences. The data retrieved includes coffee shop images and additional information about the coffee shop.

Sorting and Ranking: The server sorts the coffee shops based on a combination of factors such as rating, proximity, and user-selected filters. The sorting algorithm prioritizes coffee shops that best match the user's preferences and get the highest score after the machine learning model evaluation on this coffee shop photos.

User Experience and Feedback:

Sorted Recommendations: The server returns a sorted list of coffee shops to the client, which is then displayed in the UI. The list includes key details like the coffee shop name, address, rating, and images.

Visual Indicators: The UI highlights the key features of each recommended coffee shop, such as ratings and whether it is best suited for dine-in or takeaway.

Conclusion

By adopting a multiple client-server model, the UI of our coffee shop recommendation platform offers a personalized and efficient way for users to discover the best coffee shops in their area. The use of interactive filters and real-time data processing enhances the overall user experience, making it simple and enjoyable for clients to find their perfect cup of coffee.

System Integration

Integration of Components

The integration of the image detection module, machine learning model, and Google Maps API forms the backbone of the recommendation system, enabling it to deliver personalized and high-quality coffee shop recommendations to users. The process involves several critical steps:

1. User Input and Initial Query:

User Inputs: The user begins by specifying their preferences, such as location (e.g., Tel Aviv, Netanya), coffee type (e.g., espresso, cappuccino), and whether they prefer dine-in or takeaway.

Google Maps API Query: Once the user submits their preferences, the system queries the Google Maps API to search for coffee shops within the specified location that match the user's criteria.

2. Data Retrieval and Image Processing:

Photo Collection: The system retrieves a set of photos from each identified coffee shop using the Google Maps API. These photos are crucial for evaluating the coffee quality at each location.

Image Detection Module: The photos are then processed through the image detection module trained specifically to identify coffee-related images. This step filters out non-coffee images, ensuring that only relevant photos are considered for quality evaluation.

3. Coffee Quality Evaluation:

Machine Learning Model Application: The filtered coffee images are passed through a machine learning model designed to evaluate the quality of the coffee. This model assesses various features, such as the presence of crema (a sign of a well-prepared coffee), how the coffee is served (e.g., with a saucer and spoon), and the overall presentation.

Scoring: Based on the evaluation, each coffee shop is assigned a score. This score reflects how well the coffee from the shop matches the user's preferences and the quality criteria. For instance, a coffee shop that serves a well-presented espresso with a thick layer of crema will receive a higher score.

4. Final Sorting and Presentation:

List Sorting: After all the coffee shops have been evaluated and scored, the system sorts them based on their scores. Coffee shops that provide the best match to the user's preferences and highest quality coffee are ranked higher.

User Interface Presentation: The sorted list is then presented to the user through an engaging and dynamic user interface. Each coffee shop is displayed with:

A photo (or multiple photos) of the coffee served at the shop.

A star rating (ranging from 2-5 stars) representing the score.

The coffee shop's address.

Additional relevant details, such as whether the shop offers dine-in or takeaway options.

Final Presentation:

The final output is a visually appealing list of coffee shops, complete with interactive elements such as hover effects on images, smooth animations when scrolling through the list, and clickable elements that allow users to explore more details about each coffee shop.

This comprehensive system integration ensures that the website not only provides accurate recommendations but also delivers them in a user-friendly and aesthetically pleasing manner, making the coffee shop selection process both enjoyable and efficient for users.

Implementation

Image Detection Model

To accurately identify coffee images, we needed an object detection model that could filter out irrelevant photos returned by the Google API for various coffee shops. Initially, we explored the Grounding Dino object detection model, which showed good accuracy in identifying coffee cups within images. However, its performance was a significant drawback—it took approximately 20 seconds per image to detect whether a coffee cup was present. This delay was unacceptable, especially given the need for scalability, where we need to process at least 10 photos per coffee shop in a user's specified city.

After further research, we identified the YOLOv8 model, which we had previously explored in our Deep Learning course. YOLOv8 was particularly attractive due to its speed and efficiency. By configuring YOLOv8 for binary detection (i.e., whether a coffee cup exists in an image or not), we achieved a significant performance improvement, reducing the prediction time to just 190 milliseconds per image.

To train YOLOv8 for this task, we sourced our data from the Open Images Dataset v7, which includes labeled images with coffee cups. We divided the dataset into training, validation, and test sets to ensure robust model training and evaluation. A config.yaml configuration file was created to manage the training process, and the data was downloaded to Google Drive for easy access and storage.

The YOLOv8 model, once trained, delivered quick predictions with high accuracy, making it an ideal choice for our coffee cup detection needs. See Image Detection Results

Coffee Quality Evaluation model

To evaluate the quality of a coffee cup, we required a machine learning model capable of classifying a given image based on specific coffee-related attributes. These attributes, as mentioned earlier, include:

Type of Coffee:

Black: Plain coffee without milk or cream.

Cappuccino: Coffee made with steamed milk foam.

Espresso: Strong black coffee made by forcing steam through ground coffee beans.

Crema Presence:

Yes: The image shows the presence of crema (a layer of foam on top of espresso).

No: The image lacks crema.

Take Away or Not:

Yes: Coffee is served in a disposable cup, indicating it is for takeaway.

No: Coffee is served in a ceramic cup or glass, indicating it is for dine-in.

Served Way Quality:

Good: Coffee is well-served with additional elements like saucers, spoons, etc.

Poor: Coffee is presented without attention to detail or lacks supplementary elements.

Presentation Drawings (For Cappuccino):

Yes: The foam on the cappuccino has artistic drawings or patterns.

No: The foam is plain without any artistic presentation.

Data Collection and Labeling

We collected approximately 700 images of coffee from the internet, manually labeling them according to the attributes listed above. For example, an image of a cappuccino served in a disposable cup would receive labels such as:

Type of Coffee: Cappuccino

Crema Presence: Yes

Take Away: Yes

Served Way: Poor

Presentation: Yes

These labels were stored in a structured format, making it easier to manage and process the data. Here is an example of how the data was organized:

image	Type of coffee	Type of cup	Crema	Served	Presentation
coffe_image(27).jpg	2	0	1	0	1

Where - Type of coffee:

0 = Black coffee

1 = Espresso

2 = Cappuccino

Data Augmentation

To enhance the model's ability to generalize across different visual presentations of coffee, we expanded the dataset from 700 images to approximately 10,257 images using data augmentation techniques:

Rotation: Rotating images at various angles to simulate different camera perspectives.

Flipping: Horizontally and vertically flipping images to introduce more variability.

Scaling: Slightly scaling images up or down to mimic variations in size and proportion.

Data Normalization

To ensure consistent data representation and optimize model training, images must be standardized to a specific size. Resizing all images to 224x224x3 ensures that the model processes data uniformly, preventing potential biases caused by variations in image dimensions. This standardization is crucial for achieving accurate and reliable results.

Dataset Statistics

The augmented dataset includes the following distribution of attributes:

Attribute Counts:

crema: 7696

presentation: 1885

type_of_cup: Dine-in 9204

cappuccino: 3042

espresso: 2977

black: 4238

served_way: 5525

Attribute Percentages:

crema: 75.03%

presentation: 18.38%

type_of_cup: 89.73%

cappuccino: 29.66%

espresso: 29.02%

black: 41.32%

served_way: 53.87%

Initial Model Attempts

We divided our dataset into training, validation, and test sets using an 80/20 split. Initially, we experimented with VGG16 and ResNet50 architectures for classification. Unfortunately, these models did not perform well, even when we trained them on individual attributes like coffee type. Despite trying different hyperparameters and strategies, the results remained unsatisfactory.

We consulted with our Deep Learning course professor and other experts, exploring multiple approaches, but none yielded significant improvements.

Switching to YOLOv8

After several attempts, we reconsidered our approach and decided to leverage the YOLOv8 model, which had proven effective for object detection. We hypothesized that using YOLOv8 to detect specific coffee-related attributes could improve accuracy and reduce noise in the system.

To train YOLOv8, we created a new dataset where each coffee cup was annotated with bounding boxes and labeled according to the attributes. We first tested this approach on the coffee type attribute, and it worked successfully. Encouraged by this, we extended the method to all attributes, allowing YOLOv8 to handle the multi-class detection task.

This approach eliminated dependency on certain classes that might not coexist (e.g., Black Coffee with Crema), and it allowed us to detect coffee types accurately before applying the relevant model to classify the remaining attributes. This not only improved accuracy but also enhanced system performance by reducing the likelihood of predicting non-existent classes.

Final Model Deployment

After completing the training, we ended up with four distinct models, each specialized for one of the primary attributes:

Type of Coffee

Type of Cup

Crema Presence

Served Way Quality

We chose not to apply the model to the presentation feature since the presence of crema and a good serving method in a cappuccino is already considered a strong indicator of good quality.

Conclusion

This multi-model approach allows us to evaluate the quality of coffee in each image (filtered by the object detection model) and provide our server with an API that sorts and ranks the images returned from the Google API based on coffee quality attributes. See Coffee Quality Model Results

Website Development

The implementation of our coffee shop recommendation system involved creating a robust client-server architecture, with the server built using Flask - a lightweight WSGI web application framework in Python. This choice allowed for seamless integration of our machine learning models and efficient handling of client requests.

Server-Side Implementation (Flask - Python)

1. Flask Server Setup:

We initialized a Flask application to handle incoming requests from the client-side.

The server was configured to handle CORS (Cross-Origin Resource Sharing) to allow requests from the client-side application.

2. Integration with Trained Models:

The server incorporates two key machine learning models: a. Object Detection Model (YOLOv8): Used to identify coffee cups in images. b. Quality Evaluation Model: Assesses various attributes of the detected coffee.

These models were loaded at server startup to ensure quick response times.

3. API Endpoints:

We developed several API endpoints to handle different ○ functionalities:

Accepts user preferences and returns sorted coffee shop recommendations.

Processes individual images through our ML pipeline.

4. Google Maps API Integration:

The server uses the Google Maps API to fetch coffee shop data based on user location.

This integration allows us to retrieve shop information and associated images.

5. Image Processing Pipeline:

When images are received (from Google Maps), they are processed through the following pipeline:

a. Object Detection: The YOLOv8 model filters images to identify those containing coffee cups.

b. Quality Evaluation: Filtered images are then assessed by the quality evaluation model.

6. Recommendation Algorithm:

We implemented a custom algorithm that considers:

User preferences (location, coffee type, dine-in/takeaway)

Quality scores from our ML models

This algorithm sorts and ranks coffee shops to provide personalized recommendations.

7. Error Handling and Logging:

Robust error handling was implemented to manage potential issues with API calls, model predictions, or data processing.

We set up comprehensive logging to track server operations and facilitate debugging.

Client-Side Implementation

1. React.js Frontend:

The client-side was developed using React.js, providing a responsive and interactive user interface.

2. Chakra UI Integration:

We incorporated Chakra UI, a modular and accessible component library, to enhance the visual appeal and user experience of our application.

Chakra UI provided pre-built, customizable components that accelerated our development process and ensured a consistent design language throughout the application.

State Management:

We used React hooks to manage application state and side effects.

API Integration:

Axios was utilized to make asynchronous API calls to our Flask server.

User Interface Components:

We created reusable components for:

Search filters (location, coffee type, dine-in/takeaway)

Coffee shop display cards

Loading indicators

These components were styled and enhanced using Chakra UI's extensive component library.

Animations and Interactivity:

To create a more engaging user experience, we implemented various animations:

Smooth transitions between pages and components

Hover effects on interactive elements

Loading animations for asynchronous operations

We used libraries like Framer Motion in conjunction with Chakra UI to achieve fluid and performant animations.

Image Optimization and Display:

High-quality photos of coffee shops and coffee were integrated into the UI to provide visual context and enhance the overall aesthetic.

We implemented lazy loading for images to improve performance, especially on slower connections.

Image carousels and galleries were added to showcase multiple photos for each coffee shop.

Responsive Design:

The UI was designed to be fully responsive, ensuring a consistent experience across desktop, tablet, and mobile devices.

We utilized Chakra UI's responsive style props and custom media queries to create layouts that adapt seamlessly to different screen sizes.

The responsive design includes:

Flexible grid layouts for coffee shop listings

Collapsible navigation menus for mobile views

Adjustable font sizes and spacing for readability across devices

Accessibility:

Leveraging Chakra UI's built-in accessibility features, we ensured that the application is usable by people with various disabilities.

This includes proper heading structures, ARIA labels, and keyboard navigation support.

Performance Optimization:

We implemented code splitting and lazy loading of components to reduce initial load times.

Assets were optimized to balance quality and performance, especially for mobile users.

This implementation allows for efficient processing of user requests, leveraging our trained models to provide accurate and personalized coffee shop recommendations. The separation of client and server responsibilities ensures a scalable and maintainable architecture for future enhancements.

Results

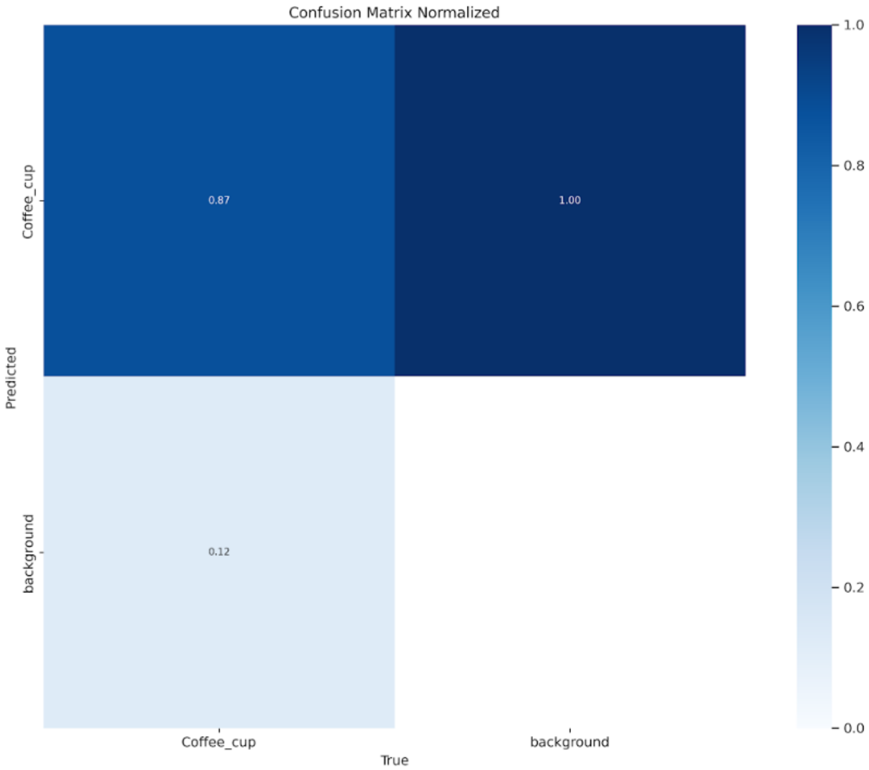
This section presents the findings and outcomes of our project, highlighting key data and insights obtained from our methodology. We provide a comprehensive overview of the results, including quantitative metrics, qualitative observations, and any significant patterns or trends identified. Through detailed analysis and visualization, we aim to illustrate the effectiveness of our approach and how it meets the objectives set forth. This section serves as a bridge between the methods employed and the conclusions drawn, offering a clear and objective account of the project’s achievements and discoveries.

Image Detection Model Results

The YOLOv8 model, after training, delivered quick predictions with high accuracy, making it an ideal choice for our coffee cup detection needs.

Screenshots and metrics, including prediction times, confusion matrix and various graphs, will be provided to illustrate the model’s performance.

Normalized Confusion Matrix



This image shows a normalized confusion matrix for a binary classification task, where the model is trying to distinguish between "Coffee_cup" and "background" classes.

The x-axis represents the true labels, while the y-axis represents the predicted labels.

The color intensity corresponds to the values in the matrix, with darker blue indicating higher values and lighter blue indicating lower values.

The matrix values:

Top-left (0.87): When the true class is "Coffee_cup", the model correctly predicts it 87% of the time.

Top-right (1.00): When the true class is "background", the model correctly predicts it 100% of the time.

Bottom-left (0.13): When the true class is "Coffee_cup", the model incorrectly predicts it as "background" 13% of the time.

Bottom-right (0.00): When the true class is "background", the model never incorrectly predicts it as "Coffee_cup".

This confusion matrix indicates that the model performs very well in identifying background instances (100% accuracy) but has some difficulty with coffee cup instances, misclassifying them as background in 13% of cases.

The model shows perfect precision for the background class but has room for improvement in recall for the coffee cup class. Overall, it demonstrates good performance, especially for background detection, but there's potential to enhance its ability to correctly identify coffee cups.

Performance & Predictions

True Positive Case:

Prediction time: 40 – 190 milliseconds per image.

Apply the model on this photo:

```
# Load the trained YOLOv8 model

model = YOLO(".\YOLO_V8\GoogleColab\runs-20240406T160452Z-001\runs\detect\train3\weights\best.pt")

# Perform object detection on the image

results = model.predict(images_path)
```

Output

image 1/1 C: \Users\USER\git\Coffee-Recommendation-System-606\Coffee_photos\images\1-50\coffe_image(20).jpg:

480x640 2 Coffee_cups, 98.8ms

Speed: 3.0ms preprocess, 98.8ms inference, 1.0ms postprocess per image at shape (1, 3, 480, 640)

True Negative Case:

Apply the model on this photo:

```
# Load the trained YOLOv8 model

model = YOLO(“.\YOLO_V8\GoogleColab\runs-20240406T160452Z-001\runs\detect\train3\weights\best.pt")

results = model.predict(images_path)
```

Output

---

image 1/1 C:\Users\USER\Downloads\coca.jpg:

640x384 (no detections), 120.9ms

Speed: 2.9ms preprocess, 120.9ms inference, 1.0ms postprocess per image at shape (1, 3, 640, 384)

For more information and visual results -

https://github.com/AhigadGenish/Coffee-Recommendation-System-606/tree/main/YOLO_V8/GoogleColab/runs-20240406T160452Z-001/runs/detect/train3

Coffee Quality Model Results

The Coffee Quality Model comprises four distinct sub-models, each dedicated to predicting a specific attribute of a given coffee cup. In this section, we will present the results for each model:

1. Type of Coffee Classification Model
This model predicts the type of coffee in each cup. It classifies the coffee as Black Coffee, Cappuccino, Espresso, or None of the Above.

2. Crema Detection Model
This model predicts whether the given coffee cup contains crema, an indicator of coffee quality and freshness.

3. Type of Cup Classification Model
This model identifies whether the coffee cup is intended for takeaway (disposable) or if it is served in a ceramic or glass cup, which can impact the drinking experience.

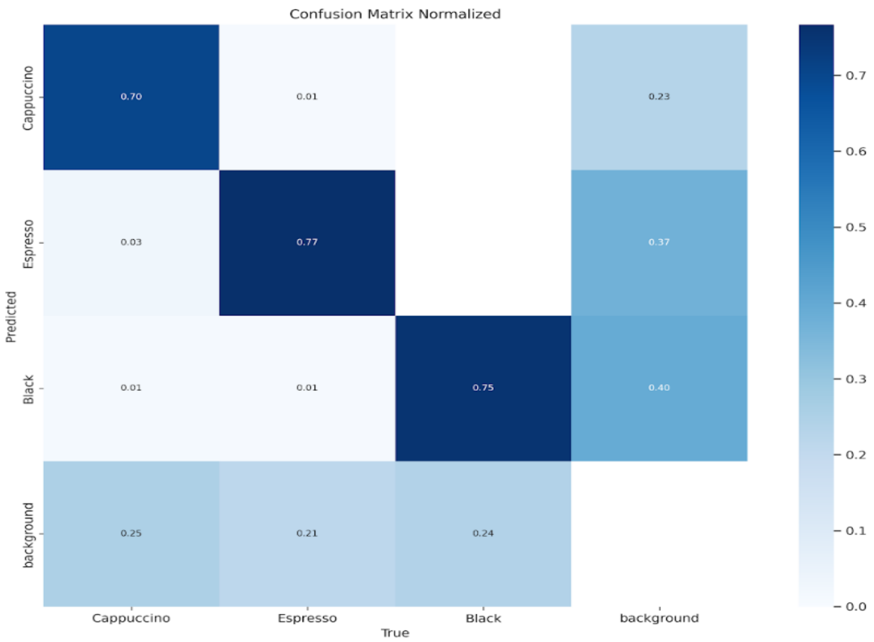
4. Serving Style Model
This model predicts whether the coffee cup is served with a saucer or spoon, elements that enhance the customer's experience.

To illustrate the performance of these models, we will provide screenshots and detailed metrics, including prediction times, confusion matrices, and various graphs.

Type of Coffee Classification Model

The Type of Coffee Classification Model is designed to categorize coffee images into one of three classes: Cappuccino, Espresso, or Black Coffee. Additionally, it includes a background class to identify non-coffee images, which are filtered out by a preliminary model before being passed to the classifier.

The provided normalized confusion matrix highlights the model's performance across these classes:



Coffee Type - Normalized Confusion Matrix

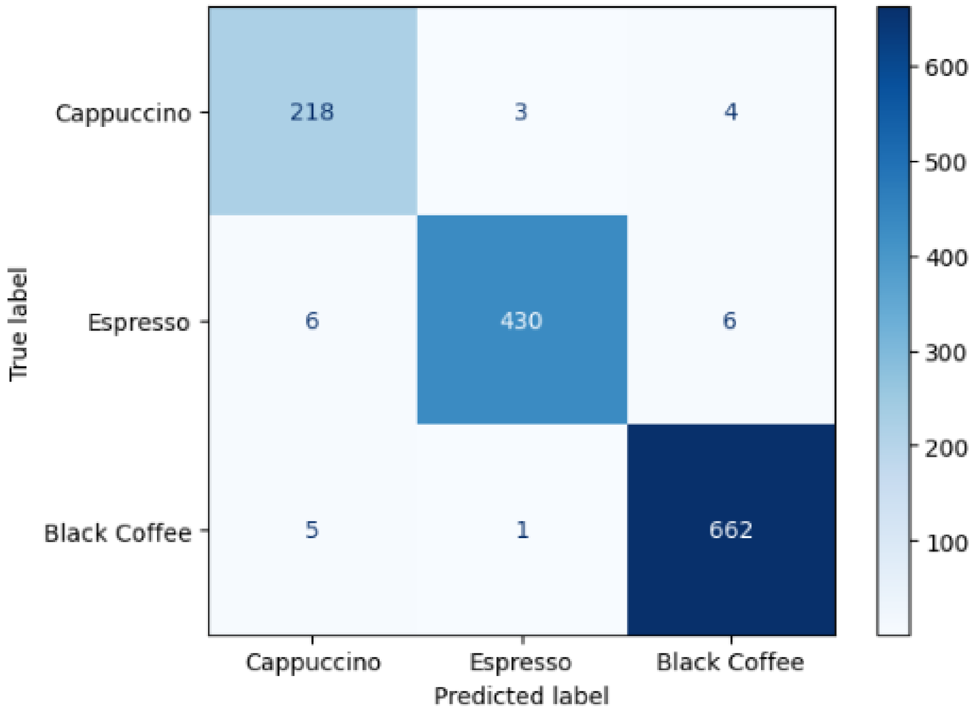
Cappuccino: The model correctly classifies 70% of Cappuccino images. However, there is some confusion with the background class, where 23% of the images are incorrectly identified as non-coffee images.

Espresso: The model achieves a 77% accuracy rate in correctly identifying Espresso images. There is a minor mix-up with other classes, but the primary misclassification is again with the background, which is expected as it attempts to filter out non-coffee images.

Black Coffee: This class shows a 75% accuracy rate. The confusion primarily occurs with the background class, where 40% of Black Coffee images are mistakenly classified as background.

Background: As this model is designed to primarily filter out non-coffee images, the background class sees higher misclassification rates, especially with Cappuccino and Black Coffee.

While the normalized confusion matrix represents the model's accuracy when dealing with a broader range of images (including non-coffee), it is important to note that when the input consists solely of coffee cups, the model's accuracy significantly improves. In a test conducted on 1,335 coffee images, the accuracy was about 98%.



[Train and test of coffee type classification]

Predictions

Detect Cappuccino

```
# Path to your image

images_path = r"C:\Users\USER\git\Coffee-Recommendation-System-606\Coffee_photos\images\1-50\coffe_image(16).jpg"



# Load the trained YOLOv8 model

model = YOLO(r"C:\Users\USER\git\ Coffee-Recommendation-System-606\App\server\resources\MLmodels\coffee_type_classification\coffee_type_classification_v1.pt")



# Perform object detection on the image

results = model.predict(images_path)
```

Output

```
image 1/1 C:\Users\USER\git\Coffee-Recommendation-System-606\Coffee_photos\images\1-50\coffe_image(16).jpg: 224x224 1 Cappuccino, 41.2ms

Speed: 2.0ms preprocess, 41.2ms inference, 0.0ms postprocess per image at shape (1, 3, 224, 224)
```

Detect Espresso

Output

image 1/1 C:\Users\USER\git\Coffee-Recommendation-System-606\Coffee_photos\images\1-50\coffe_image(36).jpg: 224x128 1 Espresso, 32.9ms

Speed: 1.0ms preprocess, 32.9ms inference, 1.0ms postprocess per image at shape (1, 3, 224, 128)

Detect Black

Output

image 1/1 C:\Users\USER\git\Coffee-Recommendation-System-606\Coffee_photos\images\1-50\coffe_image(1).jpg: 224x192 1 Black, 36.0ms

Speed: 1.0ms preprocess, 36.0ms inference, 1.0ms postprocess per image at shape (1, 3, 224, 192)

Summary of Performance:

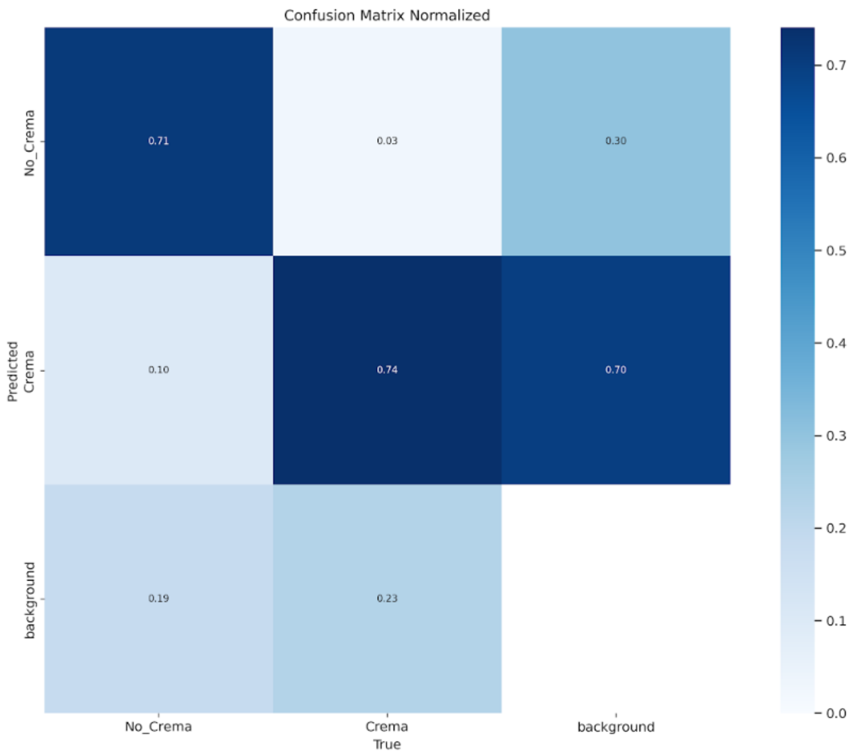
The Type of Coffee Classification Model performs well when identifying actual coffee types, particularly when dealing with clear, distinct coffee images. The YOLOv8 model, which was used in the training process, offers quick and accurate predictions when the input images are straightforward, containing only coffee. However, as seen in the confusion matrix, when non-coffee images slip through or the image contains additional elements, the model tends to misclassify them as background, leading to some performance degradation.

To illustrate these findings, additional screenshots and example detections were provided. These examples demonstrate how the YOLOv8 model handles coffee-only images effectively but struggles with mixed or non-coffee images, a scenario we've aimed to minimize by filtering inputs with a preliminary layer.

Crema Classification Model

The Crema Classification Model is designed to detect the presence of crema in coffee images. Additionally, it includes a background class to identify non-coffee images, which are filtered out by a preliminary model before being passed to the classifier.

The provided normalized confusion matrix highlights the model's performance across these classes:



Crema Class - Normalized Confusion Matrix

This confusion matrix illustrates the performance of a classification model that predicts whether a coffee has crema (the frothy layer on top) or not. The matrix is normalized, meaning the values represent the percentage of predictions relative to the total predictions for each class.

Classes:

No_Crema: Images of coffee without crema.

Crema: Images of coffee with crema.

Background: Non-coffee images or images that the model has classified as background.

Analysis:

No_Crema Class:

True Positive (0.71): 71% of the No_Crema images were correctly classified as No_Crema by the model.

False Negative (0.03): 3% of No_Crema images were incorrectly classified as having crema.

False Positive (0.30): 30% of No_Crema images were misclassified as background. This indicates some confusion between coffee without crema and non-coffee images.

Crema Class:

True Positive (0.74): 74% of the images with crema were correctly classified as Crema by the model.

False Negative (0.10): 10% of crema images were incorrectly classified as No_Crema.

False Positive (0.70): 70% of crema images were misclassified as background. This is a significant amount, suggesting the model struggles to distinguish between coffee with crema and non-coffee images.

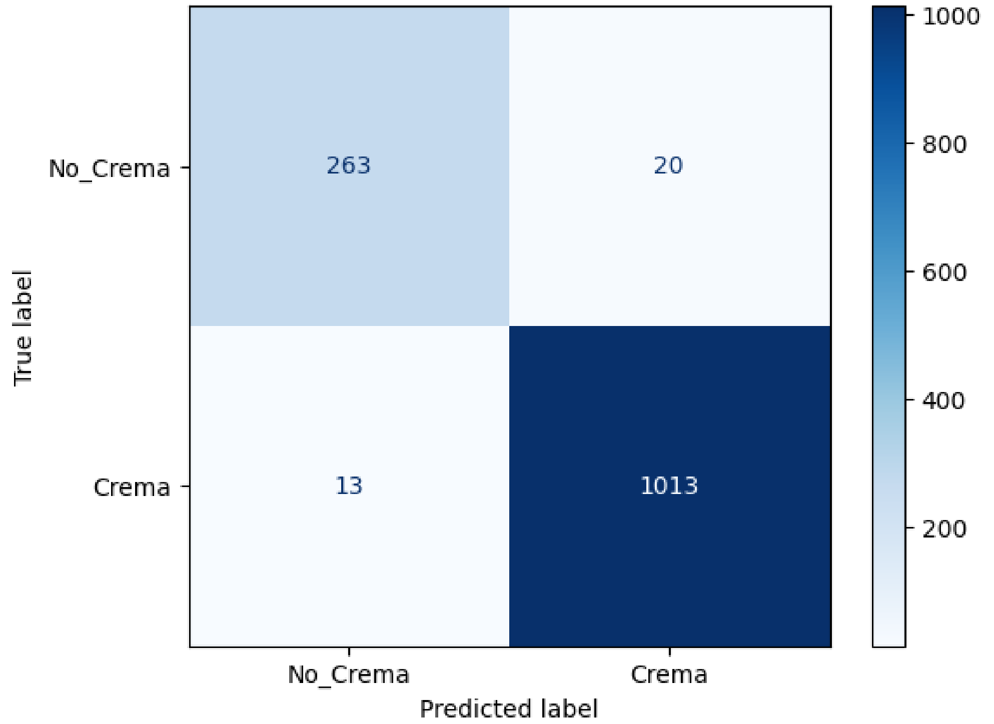
Background Class:

True Positive (0.70): 70% of background images were correctly identified as background by the model.

False Negative (0.19): 19% of background images were incorrectly classified as No_Crema.

False Positive (0.23): 23% of background images were misclassified as having crema.

While the normalized confusion matrix represents the model's accuracy when dealing with a broader range of images (including non-coffee), it is important to note that when the input consists solely of coffee cups, the model's accuracy significantly improves. In a test conducted on 1,335 coffee images, the accuracy was about 95%.



Train and test of crema classification

Predictions

Detect Crema

```
# Path to your image

images_path = r"C:\Users\USER\git\Coffee-Recommendation-System-606\Coffee_photos\images\1-50\coffe_image(16).jpg"



# Load the trained YOLOv8 model

model = YOLO(r"C:\Users\USER\git\ Coffee-Recommendation-System-606\App\server\resources\MLmodels\crema_classification\crema_classification_v1.pt")



# Perform object detection on the image

results = model.predict(images_path)
```

Output

```
image 1/1 C:\Users\USER\git\Coffee-Recommendation-System-606\Coffee_photos\images\1-50\coffe_image(16).jpg: 224x128  1 Crema, 29.0ms

Speed: 1.0ms preprocess, 29.0ms inference, 1.0ms postprocess per image at shape (1, 3, 224, 128)
```

Detect No Crema

Output

image 1/1 C:\Users\USER\git\Coffee-Recommendation-System-606\Coffee_photos\images\1-50\coffe_image(1).jpg: 224x192 1 No_Crema, 37.5ms

Speed: 1.0ms preprocess, 37.5ms inference, 1.0ms postprocess per image at shape (1, 3, 224, 192)

Summary of Performance:

The normalized confusion matrix reveals that the model performs adequately in distinguishing between coffee images with and without crema, but it faces challenges when it comes to differentiating these images from non-coffee content. However, when the input consists solely of coffee cup images, the model's accuracy significantly improves. In a test conducted on 1,335 coffee images, the accuracy was approximately 95%.

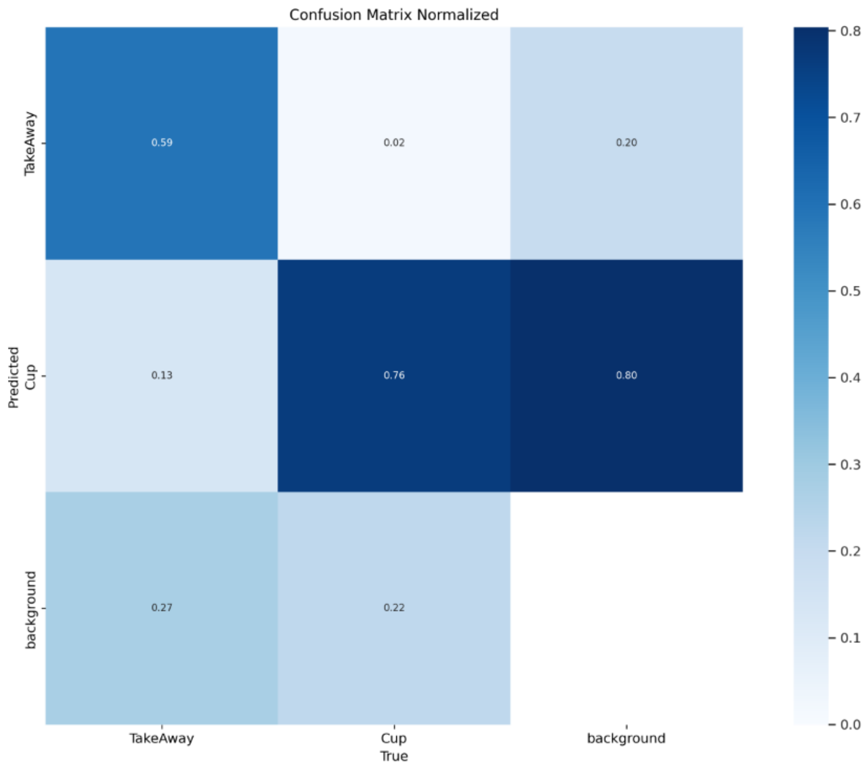
The Crema Classification Model, trained using the YOLOv8 model, is highly effective in quickly and accurately identifying coffee types, particularly in straightforward coffee-only images. However, as the confusion matrix illustrates, the model tends to misclassify images containing additional elements or non-coffee content as background. This misclassification can degrade performance, especially in scenarios with mixed or non-coffee images.

To better understand the model's strengths and weaknesses, additional examples and screenshots were provided. These demonstrate how the YOLOv8 model excels in detecting crema in coffee-only images but struggles with more complex or mixed content—a scenario mitigated by a preliminary filtering layer.

Type of cup Classification Model

The Type of Cup Classification Model is designed to distinguish between disposable takeaway cups and glass/ceramic cups used for dine-in, based on coffee images. Additionally, it includes a background class to identify and filter out non-coffee images using a preliminary model before passing them to the classifier.

The provided normalized confusion matrix highlights the model's performance across these classes:



Type of cup Class - Normalized Confusion Matrix

This confusion matrix illustrates the performance of a classification model that predicts different types of cups or cup-related images. The matrix is normalized, meaning the values represent the percentage of predictions relative to the total predictions for each class.

Classes:

Takeaway: Images of takeaway cups.

Cup: Images of regular cups.

Background: Non-cup images or images that the model has classified as background.

Analysis:

Takeaway Class:

True Positive (0.59): 59% of the Takeaway cup images were correctly classified as Takeaway by the model.

False Negative (0.02): 2% of Takeaway cup images were incorrectly classified as regular cups.

False Positive (0.39): 39% of Takeaway cup images were misclassified as background. This indicates significant confusion between takeaway cups and non-cup images.

Cup Class:

True Positive (0.76): 76% of the regular cup images were correctly classified as Cup by the model.

False Negative (0.13): 13% of regular cup images were incorrectly classified as Takeaway cups.

False Positive (0.22): 22% of regular cup images were misclassified as background. This suggests some difficulty in distinguishing between regular cups and non-cup images.

Background Class:

True Positive (0.80): 80% of background images were correctly identified as background by the model.

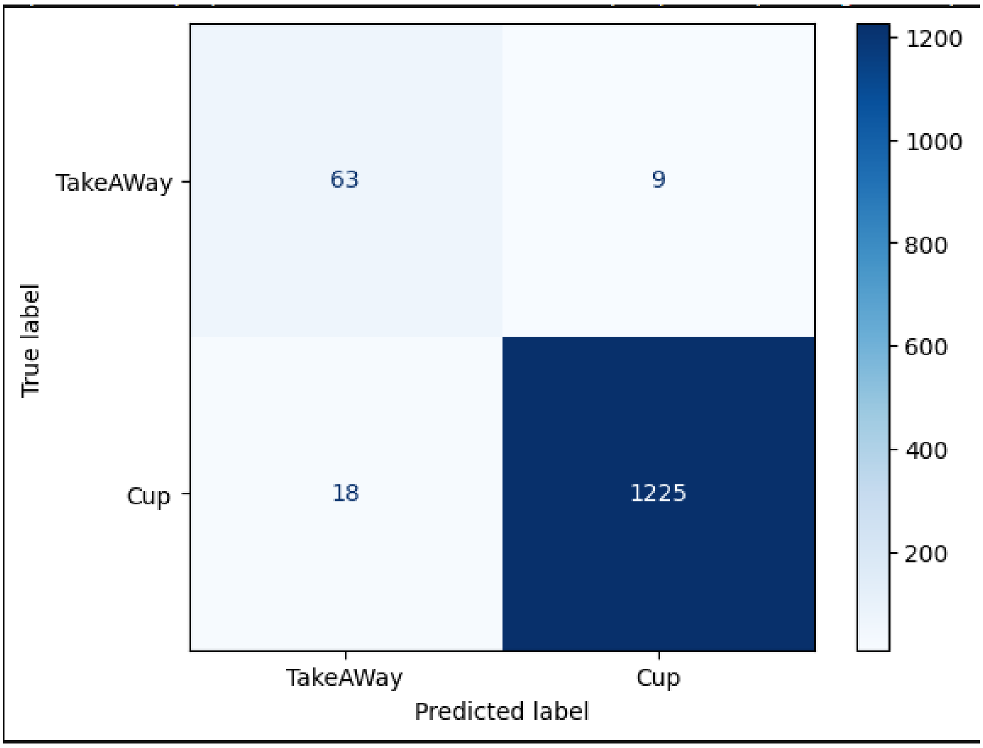
False Negative (0.27): 27% of background images were incorrectly classified as Takeaway cups.

False Positive (0.20): 20% of background images were misclassified as regular cups.

Overall Performance:

The model performs best at identifying regular cups and background images, with accuracies of 76% and 80% respectively. However, it struggles more with takeaway cups, correctly identifying them only 59% of the time. There's a notable tendency to misclassify takeaway cups as background (39%), which suggests the model might have difficulty with certain features of takeaway cups that make them appear more like non-cup objects.

While the normalized confusion matrix represents the model's accuracy when dealing with a broader range of images (including non-coffee), it is important to note that when the input consists solely of coffee cups, the model's accuracy significantly improves. In a test conducted on 1,335 coffee images, the accuracy was about 96%.



Train and test of type of cup classification

Predictions

Detect Take Away

```
# Path to your image

images_path = r"C:\Users\USER\git\Coffee-Recommendation-System-606\Coffee_photos\images\1-50\coffe_image(16).jpg"

# Load the trained YOLOv8 model

model =

YOLO(r"C:\Users\USER\git\ Coffee-Recommendation-System-606 \App\server\resources\MLmodels\type_of_cup_classification\type_of_cup_classification_v1.pt" )

results = model.predict(images_path)
```

Output

```
image 1/1 C:\Users\USER\git\Coffee-Recommendation-System-606\Coffee_photos\images\1-50\coffe_image(16).jpg: 224x224 1 TakeAway, 35.0ms

Speed: 2.0ms preprocess, 35.0ms inference, 1.4ms postprocess per image at shape (1, 3, 224, 224)
```

Detect Cup

Output

image 1/1 C:\Users\USER\git\Coffee-Recommendation-System-606\Coffee_photos\images\1-50\coffe_image(1).jpg: 224x192 1 Cup, 34.3ms

Speed: 1.0ms preprocess, 34.3ms inference, 0.2ms postprocess per image at shape (1, 3, 224, 192)

Summary of Performance:

The Coffee Cup Classification Model, trained using the YOLOv8 model, is highly effective in quickly and accurately identifying coffee cup types, particularly in straightforward coffee-only images. However, as the confusion matrix illustrates, the model tends to misclassify images containing additional elements or non-coffee content as background. This misclassification can degrade performance, especially in scenarios with mixed or non-coffee images.

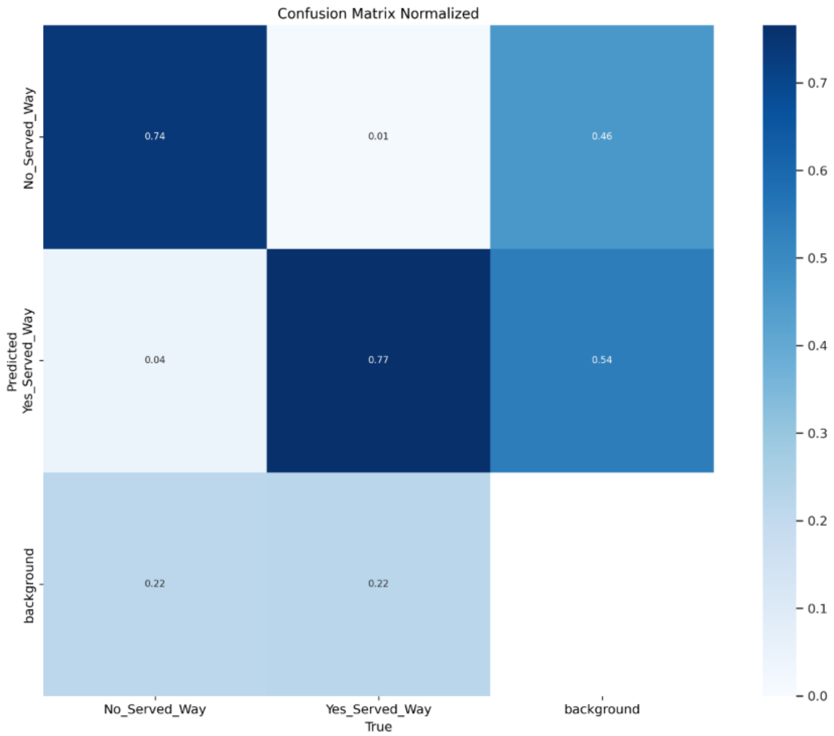
The model performs best at identifying regular cups (76% accuracy) and background images (80% accuracy). However, it struggles more with takeaway cups, correctly identifying them only 59% of the time. There's a notable tendency to misclassify takeaway cups as background (39%), which suggests the model might have difficulty with certain features of takeaway cups that make them appear more like non-cup objects.

To mitigate these issues and improve overall performance, a preliminary filtering layer could be implemented to focus the model's attention on coffee cup-only images, where it demonstrates high accuracy (96%). This approach would help leverage the model's strengths while minimizing its weaknesses in handling mixed or complex image content.

Served Way Classification Model

The Served Way Classification Model is designed to distinguish between different ways coffee is served (e.g., on a saucer with a spoon or without) based on coffee images. Additionally, it includes a background class to identify and filter out non-coffee images using a preliminary model before passing them to the classifier.

The provided normalized confusion matrix highlights the model's performance across these classes:



Served Way Class - Normalized Confusion Matrix

This confusion matrix illustrates the performance of a classification model that predicts whether a coffee is served in a specific way or not. The matrix is normalized, meaning the values represent the percentage of predictions relative to the total predictions for each class.

Classes:

No_Served_Way: Images of coffee not served in the specific way.

Yes_Served_Way: Images of coffee served in the specific way.

Background: Non-coffee images or images that the model has classified as background.

Analysis:

No_Served_Way Class:

True Positive (0.74): 74% of the No_Served_Way images were correctly classified as No_Served_Way by the model.

False Negative (0.01): 1% of No_Served_Way images were incorrectly classified as Yes_Served_Way.

False Positive (0.46): 46% of No_Served_Way images were misclassified as background. This indicates significant confusion between coffee not served in the specific way and non-coffee images.

Yes_Served_Way Class:

True Positive (0.77): 77% of the images with coffee served in the specific way were correctly classified as Yes_Served_Way by the model.

False Negative (0.04): 4% of Yes_Served_Way images were incorrectly classified as No_Served_Way.

False Positive (0.54): 54% of Yes_Served_Way images were misclassified as background. This is a significant amount, suggesting the model struggles to distinguish between coffee served in the specific way and non-coffee images.

Background Class:

True Positive (0.00): 0% of background images were correctly identified as background by the model. This cell is empty in the matrix, indicating a potential issue with background classification.

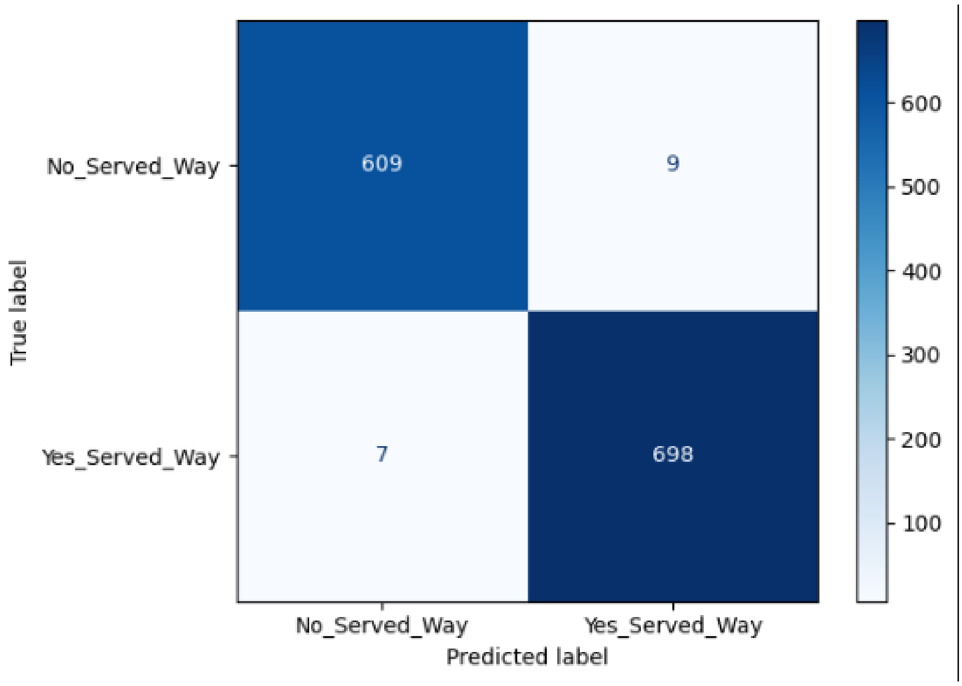
False Negative (0.22): 22% of background images were incorrectly classified as No_Served_Way.

False Positive (0.22): 22% of background images were misclassified as Yes_Served_Way.

Overall Performance:

The model performs reasonably well in distinguishing between coffee served in the specific way and not served in that way, with accuracies of 77% and 74% respectively. However, it faces significant challenges in differentiating coffee images from background or non-coffee content.

While the normalized confusion matrix represents the model's accuracy when dealing with a broader range of images (including non-coffee), it is important to note that when the input consists solely of coffee cups, the model's accuracy significantly improves. In a test conducted on 1,335 coffee images, the accuracy was about 97%.



[Train and test of served way classification]

Predictions

Detect Poor served way

```
# Path to your image

images_path = r"C:\Users\USER\git\Coffee-Recommendation-System-606\Coffee_photos\images\1-50\coffe_image(1).jpg"



# Load the trained YOLOv8 model

model = YOLO(r"C:\Users\USER\git\ Coffee-Recommendation-System-606\App\server\resources\MLmodels\served_way_classification\served_way_classification_v1.pt")



# Perform object detection on the image

results = model.predict(images_path)
```

Output

```
image 1/1 C:\Users\USER\git\Coffee-Recommendation-System-606\Coffee_photos\images\1-50\coffe_image(1).jpg: 224x192 1 No_Served_Way, 31.5ms

Speed: 2.0ms preprocess, 31.5ms inference, 1.0ms postprocess per image at shape (1, 3, 224, 192)
```

Detect Good served way

Output

image 1/1 C:\Users\USER\git\Coffee-Recommendation-System-606\Coffee_photos\images\501-550\coffe_image(506).jpg: 224x192 1 Yes_Served_Way, 46.1ms

Speed: 1.0ms preprocess, 46.1ms inference, 0.8ms postprocess per image at shape (1, 3, 224, 192)

Summary of Performance:

The model performs reasonably well in distinguishing between coffee served in the specific way and not served in that way, with accuracies of 77% and 74% respectively. However, it faces significant challenges in differentiating coffee images from background or non-coffee content.

The model shows a strong tendency to misclassify both types of coffee images (Yes_Served_Way and No_Served_Way) as background, with high false positive rates of 54% and 46% respectively. This suggests that the model might be overly sensitive in categorizing images as background, potentially due to complex or ambiguous features in the coffee images.

Notably, the model appears to have particular difficulty with background images, failing to correctly identify any of them (0% true positive rate for background). Instead, it tends to misclassify background images equally as No_Served_Way or Yes_Served_Way (22% each).

To mitigate these issues and improve overall performance, a preliminary filtering layer could be implemented to focus the model's attention on coffee cup-only images, where it demonstrates high accuracy (97%). This approach would help leverage the model's strengths while minimizing its weaknesses in handling mixed or complex image content.

Conclusion

This project has effectively addressed a significant gap in the current landscape of recommendation systems, particularly in the niche domain of coffee shop recommendations tailored for coffee enthusiasts. By developing a website that prioritizes coffee quality over general popularity, we have created a tool that provides more meaningful recommendations to users who value the intricacies of coffee preparation and presentation.

The system's success lies in its innovative use of machine learning and image detection technologies, particularly the YOLOv8 model, which was chosen for its superior speed and accuracy in detecting coffee cups within images. The initial exploration of the Grounding Dino model revealed its limitations in performance, which led to the adoption of YOLOv8, resulting in a significant reduction in prediction time from 20 seconds per image to just 190 milliseconds. This efficiency is crucial for scaling the system to process large datasets, particularly when evaluating multiple coffee shops in a given location.

The project's multi-model approach has proven to be an effective strategy for evaluating various aspects of coffee quality. By segmenting the classification tasks into distinct models—such as coffee type classification, crema detection, cup type classification, and serving style classification—we ensured that each attribute was analyzed with a high degree of accuracy. This modular design not only improves the system's overall performance but also allows for future enhancements and the potential integration of additional attributes that coffee lovers may consider important.

The user experience was a key focus during the development of the website, resulting in a responsive, intuitive interface built using Flask on the server-side and React.js on the client-side. The integration of Chakra UI further enhanced the visual appeal and usability of the application, ensuring that users could easily navigate and interact with the system, whether they were accessing it from a desktop, tablet, or mobile device. The website’s real-time filtering and personalized recommendation features provide immediate feedback, making the process of discovering new coffee shops both efficient and enjoyable.

Looking ahead, the project lays a strong foundation for further development. Future enhancements could include expanding the geographical scope, incorporating additional coffee quality features, and refining the machine learning models to handle even more complex scenarios. The system's architecture is designed to accommodate such growth, making it a scalable solution that can adapt to the evolving preferences of coffee enthusiasts.

In conclusion, this project has successfully created a robust and reliable recommendation system that elevates the coffee shop discovery experience for discerning coffee lovers. By combining advanced machine learning techniques with a user-friendly interface, we have delivered a tool that not only meets but exceeds the expectations of its target audience. This project highlights the potential of specialized recommendation systems to provide more personalized, accurate, and valuable insights in niche markets, paving the way for future innovations in this space.

References

Source Code:

The code for the project is available on GitHub: GitHub Repository

This repository contains the application, scripts, models, and configuration files used for developing the coffee recommendation system.

Data:

Google Maps API:

Used to retrieve images and geolocation data for coffee shops. API documentation can be found at Google Maps API documentation.

Open Images Dataset v7:

Source: Google. Used for training the YOLOv8 model. Dataset can be accessed at Open Images Dataset.

Manually Collected Data:

Approximately 700 images of coffee cups were manually collected from various coffee shops and online sources at Coffee images.

Results:

YOLOv8 Model Training and Performance:

Details and metrics, including confusion matrices and prediction times, are documented in the project repository.

Coffee Quality Model:

The results from the coffee quality evaluation models, including accuracy and confusion matrices, are available in the project repository.

Website Development:

Implementation details, including server setup and client-side integration, are available in the source code documentation.

ReadMe .md file: Application README

Additional Resources:

YOLOv8 Documentation:

Ultralytics, the creators of YOLOv8, provide detailed documentation and usage instructions at Ultralytics YOLOv8.

Flask Framework Documentation:

Flask documentation can be accessed at Flask's documentation .

React.js Documentation:

Comprehensive React.js documentation is available at React.js Documentation.

Chakra UI Documentation:

Chakra UI's component library and customization options are detailed at Chakra UI Documentation.