Empty Seat Detection in Campus Restaurants for IISER - B Community Using Computer Vision

Written by Siddhi Pravin Lipare and Aryan Jain
3rd year, Data Science and Engineering Major, IISER Bhopal

Final Report

Motivation

As the intake in IISER - B is increasing day by day, the crowd in the campus restaurants such as Sudarshan Restaurant, Indian Coffee House (ICH) is increasing as well. A significant number of students, often commuting from distant hostel premises, encounter challenges in securing seating arrangements at Sudarshan restaurant due to high occupancy. Furthermore, during exam days when maximum students study in the Library, they prefer having lunch from Sudarshan or ICH to save their time. If there is no space, they have to either go back or wait till they find a seat. This consumes a lot of time in their already busy schedules. To solve this issue, it would be helpful to know ahead of time if there are empty tables or seats available.

Problem Statement

- Our objective is to develop a Computer Vision-based System that captures footage from a camera within Campus Restaurants and detects vacant seats.
- **Step 1**: The first stage involves assessing the presence of a person and chair within defined bounding boxes.
- **Step 2**: In the final step, if the bounding boxes corresponding to a person and a chair intersect, the seat is considered occupied (indicated in red); otherwise, it remains unoccupied (indicated in green).

YOLO

- **Algorithm Finalisation**: We have decided to use YOLOv8, an object detection model, trained on the COCO Dataset, consisting of 80 labels and over 330k images. YOLO stands for ”You Only Look Once” because it has the ability to predict all objects in one forward pass. It was developed by Ultralytics and released on January 10th, 2023.

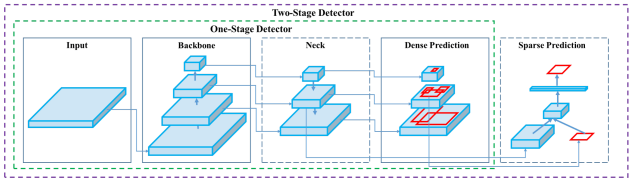

Figure 1: YOLOv8 Architecture

- **How YOLO Works**: YOLO uses a grid based approach i.e it divides the image into grid predicting bounding boxes, and class probabilities within each cell. By simultaneously estimating bounding box coordinates, object classes, and confidence scores for these predictions, YOLO performs rapid and accurate object detection in a single pass of a Convolutional Neural Network, making it suited for real time object detection. It uses Non- Maximum Suppression which helps in eliminating duplicates and highly overlapping predictions and keeping only the most confident accurate bounding boxes for each

object. YOLOv8 uses a new architecture known as SPP-Net and PANet (Path Aggregation Network) that enhances its ability to detect objects of different sizes and shapes. YOLOv8 also uses a new activation function, Mish, which provides improved accuracy and faster convergence compared to traditional activation functions like ReLU.

$$Mish(x) = x \cdot \tanh(\log(1 + e^x)) \quad (1)$$

Pre Processing Steps

1. **Object Detection**: Implementation of YOLO on an image taken in Sudarshan Restaurant and Indian Coffee House (ICH). This detects all the objects present in the image which are present in the COCO Labels.
2. **Detecting Overlap**: Defining an IoU (Intersection over Union) function to calculate the area of overlap between "person" and "chair". If the IoU is greater than 0.2, then the chair is marked by a Red bounding box (Occupied), else it would be marked by a Green bounding box (Empty).

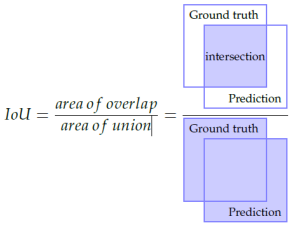

Figure 2: Intersection over Union

3. **Video Implementation**: After successful implementation on an image, we implemented it on a video by reading 258 frames of a 15 second Sudarshan video and continuously displaying the number of vacant seats over all subsequent frames.
4. **Generating a CSV File**: We generated a CSV file displaying the number of vacant seats corresponding to each timestamp. There seems to be jitter in the generated output video since the processing time is slightly slow especially on CPU. Some of the chairs were not being detected since the positioning of the camera was not accurate enough. We tried capturing videos from various views but could not find a place where all chairs could get perfectly detected.

- **Results and Findings**: YOLOv8 performs well by identifying approximately 60% of the empty seats when analyzing a single image, and this was achieved with a set IoU threshold of 0.2 and a confidence threshold of 0.5. However, when dealing with video inputs, the model encounters increased processing demands, leading to a minor fluctuation in predicting seat occupancy. This real-time computational load introduces a slight jitter in the model's output.

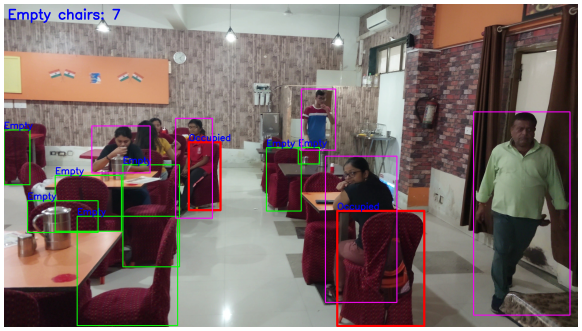

Figure 3: Detection of empty seats in Sudarshan

- We also tried implementing the model in Indian Coffee House (ICH). We got better results in this location since the chairs and persons were distinctly visible and uniformly spaced as compared to Sudarshan Restaurant. Also, due to the different colors of the chairs and tables in ICH, there were almost no errors in the detection of chairs.

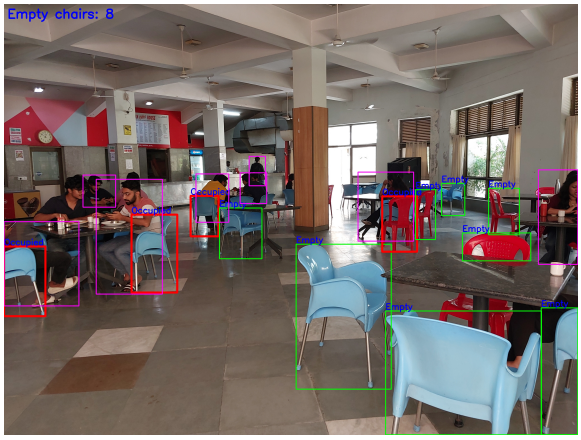

Figure 4: Detection of empty seats in ICH

Future Scope

- Using the Flask library to host our website to display only the number of vacant seats and not the live videos so that no one's privacy is harmed.
- Making a custom dataset for each restaurant and training the YOLO model on them will make our predictions more accurate and possibly faster, but definitely less robust.
- Our task of detecting only empty tables requires a large custom dataset and manual annotating for "Empty" and "Occupied" tables. We will have to create such a custom dataset and manually annotate for every location we choose in future due to which the model will lose its robustness.

Individual Roles

Member 1: Siddhi Lipare

- Analysed various YOLO versions to choose the most optimised one.
- Implemented IoU Algorithm to classify seats as "Empty and "Occupied" on the basis of their area of overlap.
- Implemented YOLO model taking an image as input and analysed how the predictions can be improved.
- Collected data from various locations in both restaurants to capture maximum number of seats.

Member 2: Aryan Jain

- Explored other pre trained models apart from YOLO as well to analyse why YOLO works better than other multi-pass detectors.
- Understood the working of Convolutional Neural Network and how YOLO works in a slightly different manner.
- Implemented YOLO model taking video as an input and analysed how the predictions can be improved.

References

[1] Seat Status Detection in Library by Chia-Hung Lin: GitHub

[2] YOLOv8 Library: GitHub

[3] YOLOv8 Architecture: RoboFlow

[4] Intelligent System for Monitoring the Availability of Bus Passenger Seats using the YOLO Method : IEEE

[5] Deep Learning-Based Object Detection Algorithms on Image and Video: IEEE

[6] YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors : YOLOv7

[7] UAV-YOLOv8: A Small-Object-Detection Model Based on Improved YOLOv8 for UAV Aerial Photography Scenarios: YOLOv8