Applied Data Science Department

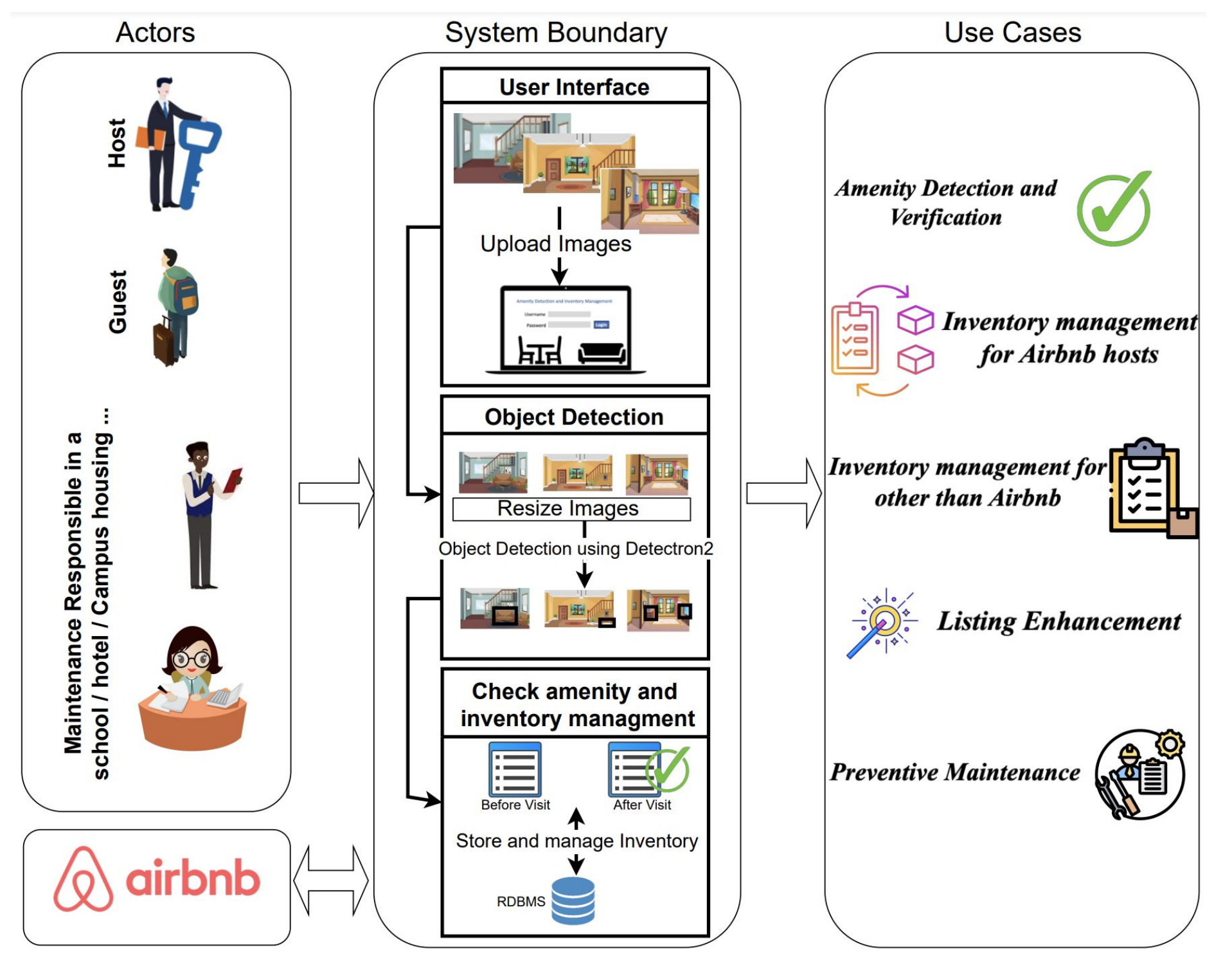
Amenity Detection and Inventory Tracking Using Detectron2

Project Advisor: Dr. Simon Shim

Faiza Ayoun
Nilisha Makam Prashantha
Sangamithra Murugesan
Joshna Devi Vadapalli

Introduction

The project addresses the exploding demand for efficient property management regarding short-term rentals like Airbnb. Motivated by the industry's concerns regarding property condition post-checkout, this project leverages deep learning and computer vision to develop a web application for amenity detection in property images. In addition to image detection, the tool incorporates inventory tracking features, providing property owners with a complete solution for monitoring and managing their amenities. The image below shows the System Boundary, Actors, and Use Cases.


Actors
System Boundary
Use Cases
Host
Guest
Maintenance Responsible in a school / hotel / Campus housing ...
airbnb
User Interface
Upload Images
Object Detection
Resize Images
Object Detection using Detectron2
Check amenity and inventory managment
Before Visit
After Visit
Store and manage Inventory
RDBMS
Amenity Detection and Verification
Inventory management for Airbnb hosts
Inventory management for other than Airbnb
Listing Enhancement
Preventive Maintenance

Methodology

Embracing the CRISP-DM methodology, the project spans the following stages.

- **Project Understanding:** Defining the main agenda of building a web application for amenity detection and inventory tracking.
- **Data Understanding:** Building taxonomy and Collecting many images of 32 different classes using Roboflow and open-images.
- **Data Preparation:** Removing irrelevant images, labeling them using LabelMe, splitting them into training, validation, and testing, and then performing data augmentation using Roboflow. The resulting Dataset had 6000+ images.
- **Model Development:** Selecting models based on a literature review and feeding the customized image dataset into models like Mask R-CNN, YOLO V7, EfficientDet, RetinaNet, and Detection Transformer.
- **Model Evaluation:** Assessing and comparing model performances using metrics like Mean Average Precision (mAP) and Intersection over Union (IoU). YOLO V7 was the best performing with mAP 0.852
- **Deployment:** Building a stand-alone web application using Streamlit and libraries like Folium, integrating the best-performing model, and creating a MySQL database hosted in Google Cloud Platform for inventory tracking.

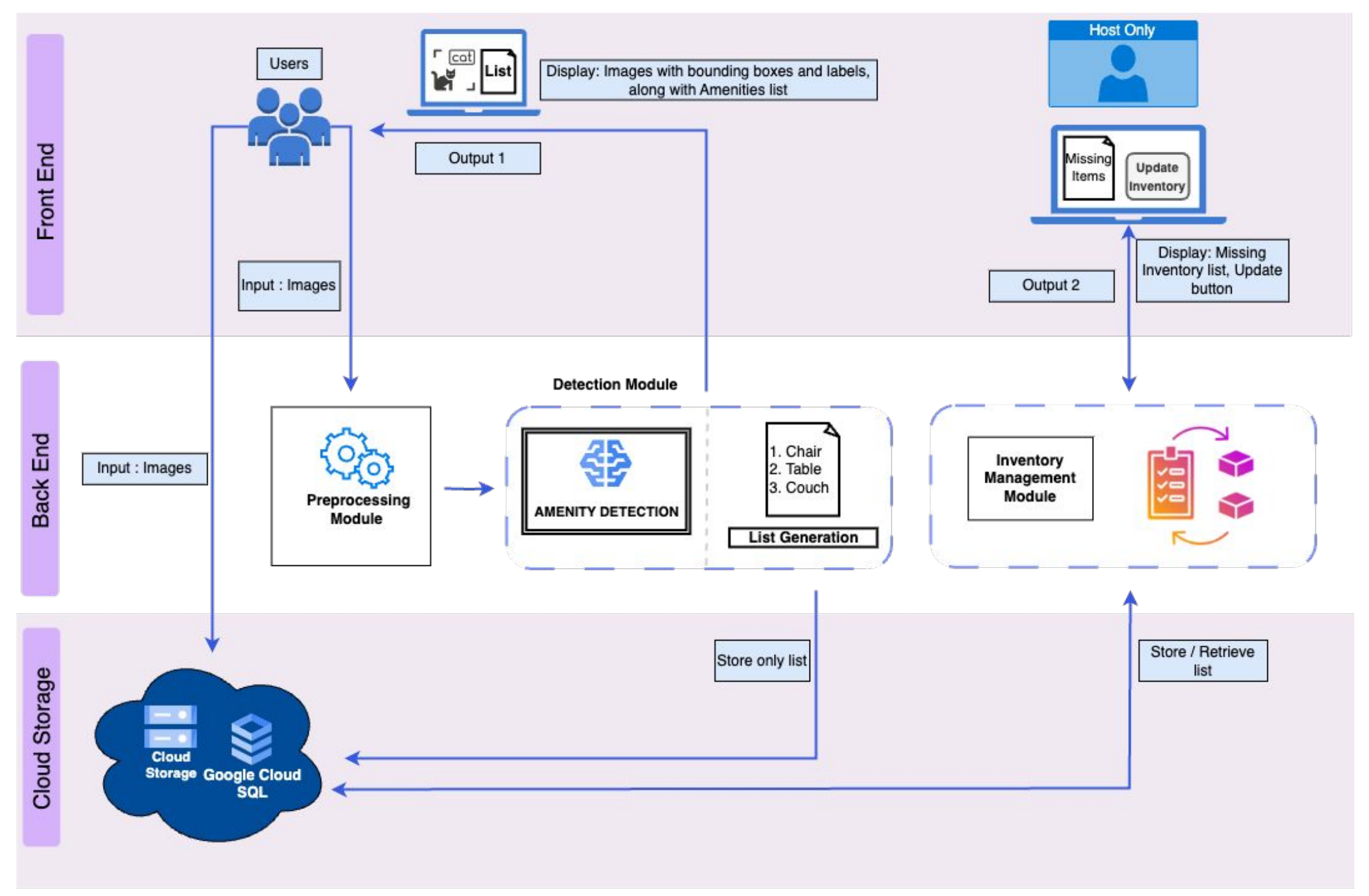
Analysis and Results

Data

Custom Dataset comprises 32 different kinds of amenities present inside and outside a property.
E.g., Swimming pool, stove, or hair dryer.
Data was collected from different sources like Open Images v7 and Google Images, and the bounding boxes of all the 2772 images were manually annotated using the LabelMe tool.
Preprocessed and augmented using Roboflow.

System Design, Web Development & Resource Requirements

Using the preprocessed data, each model was trained in a Google Colab environment using A100 NVIDIA GPU. The dataset is partitioned into 70% training, 20% validation, and 10% testing utilizing Roboflow.
Streamlit was used to build the web platform and folium API for visualizations. The image below is the system design of the project.



Evaluation Criteria

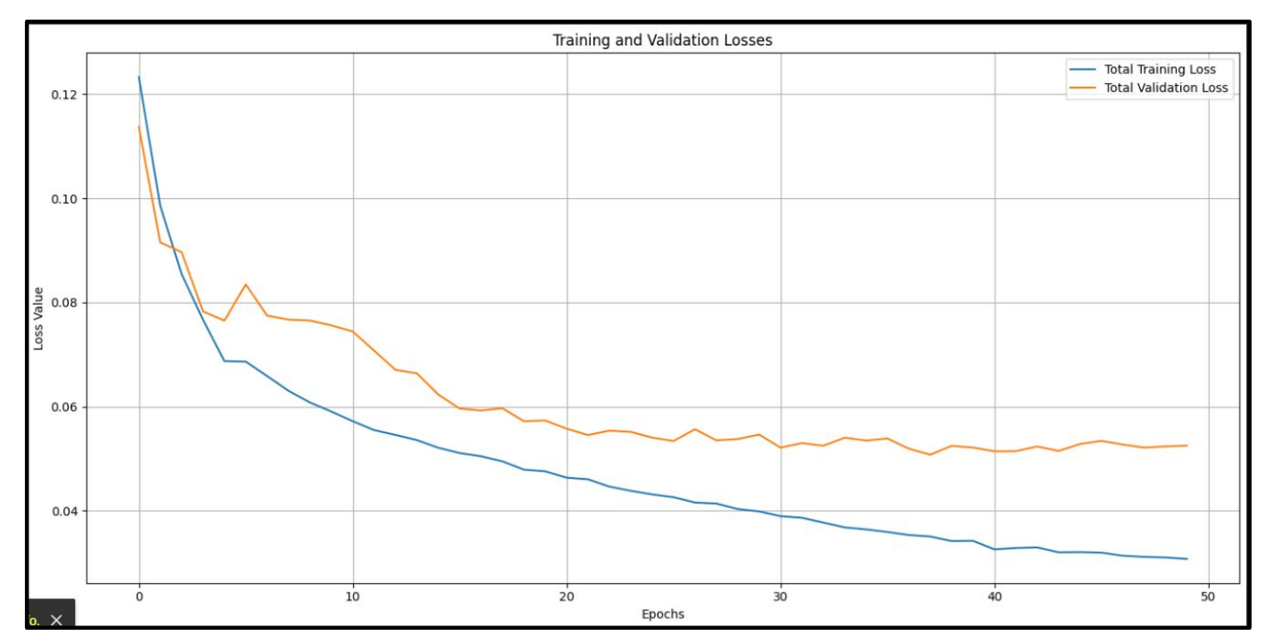
We employ **IoU** to assess the model's bounding box accuracy against ground truth.
Measure the class prediction accuracy using Mean Average Precision (**mAP**)
Precision Recall (**PR**) and learning curves help understand the model's performance changes over training and testing times.

Model Development

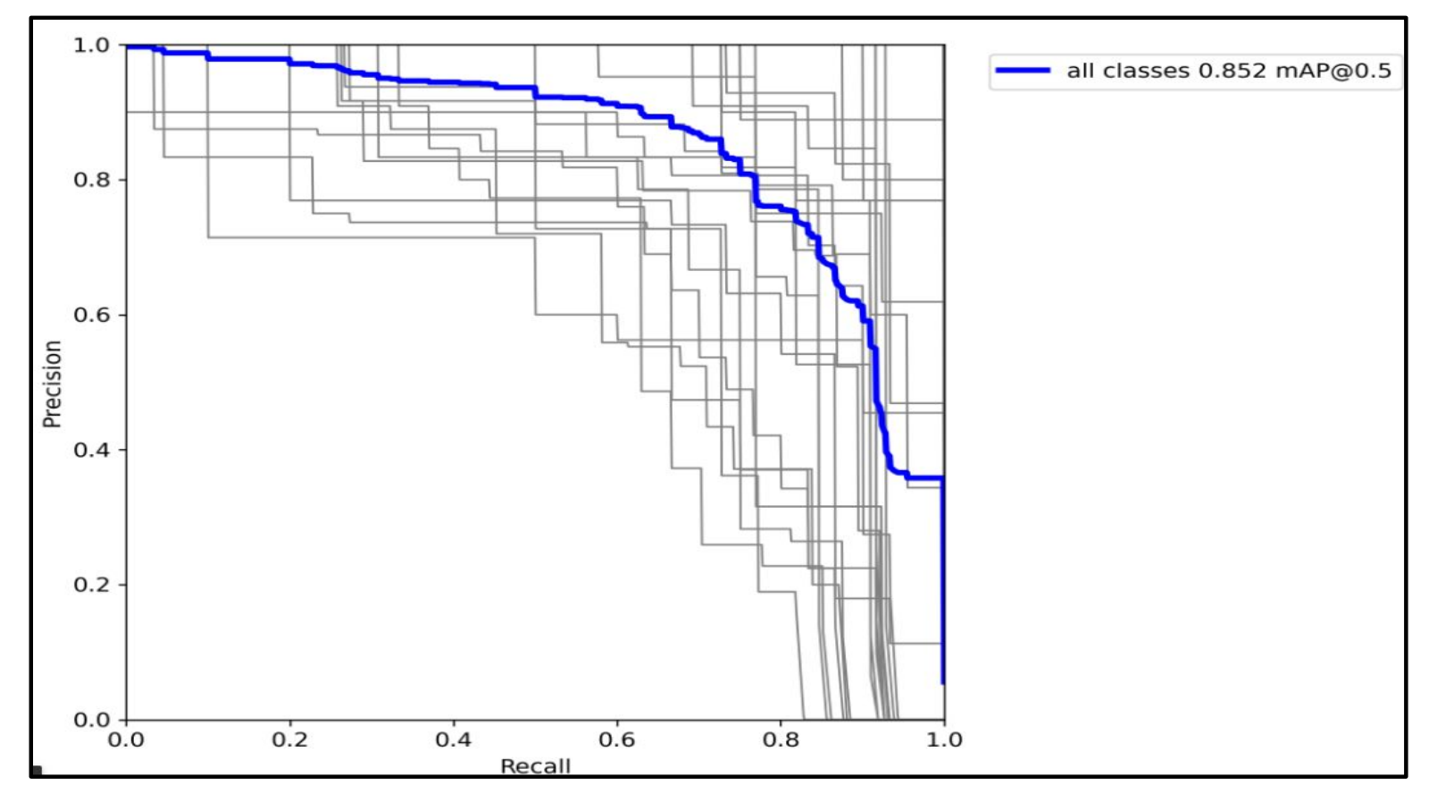
The models of this project aim to answer 'What amenity is depicted in the uploaded image?'
Through transfer learning, pre-trained models such as **Detectron2 Mask RCNN, YOLO V7, EfficientDet D0, RetinaNet, and Detection Transformers (DETR)** were retrained on the custom dataset.

Results & Discussion

The Detectron2 models, Mask R-CNN and RetinaNet, underwent training for 20,000 iterations, with respective training times of 45 and 52 minutes, respectively. DETR and Efficient Det were trained for 182 minutes and 420 minutes, respectively, taking more computational units.
YOLO V7, on the other hand, underwent 50 epochs of training in 108 minutes, showing a sharp and steady decline in training loss with converging training and validation loss curves, signifying enhanced performance and steady accuracy across the training and validation data.



A PR curve near the upper right corner indicates a model's performance is optimized, marked by high precision, which results in more relevant predictions over irrelevant ones. Additionally, a high recall demonstrates the model's effectiveness in identifying the most relevant instances. YOLO V7 achieves a recall of 81% at the IoU of 0.50 to 0.95



In summary, in single-stage detectors, YOLO V7 topped the performance charts with an mAP of 0.852. The single-stage RetinaNet also showed slightly better performance, outdoing the two-stage Mask R-CNN with an mAP of 0.684 and a quicker inference time of 8 seconds versus the Mask RCNN's 13 seconds. DETR and EfficientDet trailed behind, with DETR attaining an mAP of 0.44 in a 10-second inference time and EfficientDet logging the lowest mAP of 0.225 with the longest inference time at 20.3 seconds.

Model	Recall @0.50-0.95	mAP @ 0.5	Inference Time (sec)
Detectron2 Mask R CNN	0.493	0.666	13
Detectron2 Retinanet	0.64	0.684	8
YOLOV7	0.81	0.852	8.08
EfficientDet	0.358	0.225	20.3
DETR	0.412	0.44	10

Challenges and Solutions

1. Diversity in the custom dataset - A dataset with different angles of amenities needs to be captured for better prediction.
2. Computational limitations - Optimization techniques were implemented to address our resource constraints while training models.
3. Privacy & Data Protection - Challenges were faced in balancing the need for data insights with the ethical responsibility to protect user privacy. The solution is to implement robust protocols with data security.

Summary/Conclusions

This project developed a web application for detecting and tracking amenities in Airbnb properties, using deep learning for object detection. YOLO V7, the top-performing model among the five evaluated, achieved an mAP score of 0.86.
The Streamlit-built app lets hosts and guests upload images for instant amenity recognition and inventory updates, improving guest experience and competitiveness. However, its performance is restricted by the image quality and the detectable range of amenities.

Key References

[1] EHL Insights. (n.d.). Why do people stay in airbnb? https://hospitalityinsights.ehl.edu/travelers-airbnb-study

[2] Precision-Recall. (n.d.). Scikit-learn. https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html#:~:text=The%20precision%2Drecall%20curve%20shows,a%20low%20false%20negative%20rate.

[3] Rath, S., & Rath, S. (2022). Fine Tuning YOLOv7 - Custom Object Detection Training. LearnOpenCV – Learn OpenCV, PyTorch, Keras, Tensorflow With Examples and Tutorials.

[4] Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2022). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv preprint arXiv:2207.02696.

Acknowledgements

The team would like to express their sincere gratitude to Dr. Simon Shim for his invaluable support and guidance throughout the project. Dr. Shims' initial inputs and knowledge in the data analytics domain played a crucial role in the success of this project.

We express our sincere gratitude to San Jose State University and the Department of Applied Data Science for their invaluable support.