IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

Stay Smart: Intelligent Amenity Detection for Hospitality Platforms

[1] Atharva Samir Ghatwal, [2]Aryan Santosh Patil, [3]Sahil Santosh Patil, [4]Sangita Nikumbh

[1]B.E Student, Department of Artificial Intelligence And Data Science Engineering, New Horizon Institute of Technology And Management, Thane, India

[2]B.E Student, Department of Artificial Intelligence And Data Science Engineering, New Horizon Institute of Technology And Management, Thane, India

[3]B.E Student, Department of Artificial Intelligence And Data Science Engineering, New Horizon Institute of Technology And Management, Thane, India

[4]Assistant Professor, Department of Artificial Intelligence And Data Science Engineering, New Horizon Institute of Technology And Management, Thane, India

Abstract: Stay Smart: Intelligent Amenity Detection for Hospitality Platforms

This paper presents an intelligent amenity detection system for hospitality platforms using Detectron2. A fine-tuned RetinaNet model, trained on a curated Open Images dataset, is employed to detect household amenities such as coffee makers and fireplaces. Data preprocessing, annotation formatting, and model training were conducted with GPU acceleration to enhance performance. The model was evaluated using mean Average Precision (mAP), achieving high accuracy and robust generalization to unseen images. Despite challenges like class imbalance and overlapping objects, it outperforms generic detection models. This work establishes a scalable pipeline for real-time deployment, with future enhancements focused on dataset expansion and hyperparameter optimization.

***Index Terms* -** Object detection, Detectron2, deep learning, hospitality platforms, computer vision, RetinaNet.

I. INTRODUCTION

Object detection plays a vital role in computer vision, enabling applications across autonomous systems, surveillance, robotics, and e-commerce. While early methods depended on handcrafted features, deep learning-based models such as Faster R-CNN, YOLO, and RetinaNet have revolutionized accuracy and efficiency. Among these, Detectron2, developed by Facebook AI Research (FAIR), offers a modular and scalable framework, making it highly effective for real-world applications.

Importance of Amenity Detection: Detecting household amenities is essential for industries like hospitality and real estate, where platforms such as Airbnb rely on accurate image based listings. Manual annotation is time-intensive and susceptible to errors, leading to inconsistencies. Automating this process improves property classification, enhances searchability, and ensures reliable listing verification, ultimately benefiting both property owners and users.

II. LITERATURE SURVEY

To automate amenity detection in hospitality platforms, a variety of computer vision techniques have been employed. The identification of household amenities, image-based property classification, and automated tagging for real estate listings have all been the focus of several research studies.

[1] Machine learning-based pattern recognition and image processing techniques (2023): This paper discusses digital image processing and pattern recognition using machine learning algorithms. It explores various techniques such as image segmentation, feature extraction, and noise reduction to improve the accuracy of pattern recognition. The study highlights applications in medical imaging, remote sensing, and surveillance while emphasizing the role of artificial intelligence in automated image processing.

[2] Statistical scaling of urban amenities and their spatial distribution (2022): This study examines how urban amenities scale with population size, using spatial distribution modeling. While your project focuses on indoor amenities, both projects rely on image-based detection and classification of amenities. Insights from this research could be used to enhance dataset diversity by considering regional variations in amenity availability.

[3] Evaluating access to green spaces in smart cities using geospatial analysis (2023): This research presents a geospatial analysis framework for evaluating green space accessibility in smart cities. Using Open Street Maps data and statistical modeling, the study assesses urban parks and recreational areas across Medellin, Milan, Chicago, Singapore, and Mumbai. The findings highlight unequal distribution of green spaces and suggest policy interventions to ensure fair accessibility. The study also emphasizes technology-driven urban planning to enhance community well-being.

[4] TensorMask: A Foundation for Dense Object Segmentation (2019): This paper introduces TensorMask, a framework for dense sliding window instance segmentation. It explores the use of 4D tensors to improve instance segmentation accuracy and compares its performance with Mask R-CNN. The study provides a new approach to object detection and segmentation, which can be useful in amenity detection for hospitality platforms by enhancing detection precision.

[5] Detection, Instance Segmentation, and Classification for Astronomical Surveys with Deep Learning (DeepDISC) (2023): This paper discusses Detectron2-based deep learning techniques for object detection, deblending, and classification in astronomical surveys. It highlights the use of neural networks for object segmentation in space imaging. While focused on astronomy, the methodology can be applied to amenity detection in hospitality, particularly in handling object occlusion and complex image segmentation tasks.

II. PROPOSED APPROACH

This study leverages Detectron2’s RetinaNet model to develop a structured amenity detection pipeline, ensuring accurate and efficient identification of household amenities. The dataset, sourced from Open Images, undergoes rigorous preprocessing, including annotation formatting, data cleaning, and augmentation to enhance model robustness. RetinaNet, chosen for its balance of accuracy and efficiency, utilizes a ResNet-50 backbone with Feature Pyramid Networks (FPN) and focal loss to address class imbalance. The training process is GPU-accelerated, with the dataset split into training (80%), validation (10%), and testing (10%) sets. The methodology includes dataset curation, fine-tuning of a pre-trained model, and performance evaluation using Mean Average Precision (mAP). The trained model demonstrates high accuracy and strong generalization, effectively localizing household amenities across varied image conditions, ensuring reliable detection in diverse environments.

Contributions and Future Scope: The key contributions of this work include a specialized object detection pipeline, improved automation for property listings, and a scalable approach for amenity recognition. Future efforts will focus on expanding the dataset, refining detection models, and deploying the system in real-time applications for enhanced commercial usability.

III. REQUIREMENTS

I. Hardware Requirements

- The model training and inference were conducted on Google Colab, leveraging GPU acceleration for deep learning computations. The minimum hardware specifications are:
- Processor: Intel Xeon CPU (Google Colab VM)
- GPU: NVIDIA Tesla K80 / P100 / T4 (session dependent)
- Memory: 12–16 GB RAM
- Storage: Google Drive integration for dataset and model storage CUDA-enabled GPUs were utilized to accelerate training and inference.

II. Software Requirements

- The project was implemented in a Linux-based environment with the following dependencies:
- Operating System: Ubuntu (Google Colab)
- Programming Language: Python 3.8
- Deep Learning Framework: PyTorch 1.4+ with CUDA 10.0 support
- Object Detection Framework: Detectron2
- Additional Libraries:
- torch, torchvision (deep learning models)
- numpy, pandas, opencv-python (data processing)
- matplotlib, tqdm (visualization and tracking)

IIII. ALGORITHMS AND FLOWCHART

- The data preprocessing algorithm involves downloading images and annotations from Open Images, filtering relevant objects, formatting bounding boxes for Detectron2, and cleaning data by removing duplicates and missing annotations. The dataset is split into training (80%), validation (10%), and testing (10%) sets.

- The model training algorithm starts with loading a COCO pretrained RetinaNet model. Training parameters such as batch size, learning rate, and max iterations are configured. Data augmentation techniques, including horizontal flipping, color jittering, and scaling, improve generalization. The model is trained using stochastic gradient descent (SGD), and its performance is validated using mAP scores before saving the best-performing weights.

- For inference and object detection, the trained model loads its final weights and applies preprocessing to input images. The image is resized and converted into a Detectron2 compatible format. RetinaNet processes the image, filtering low-confidence detections. The output is post processed by drawing bounding boxes with class labels and confidence scores, providing clear visualizations of detected amenities.

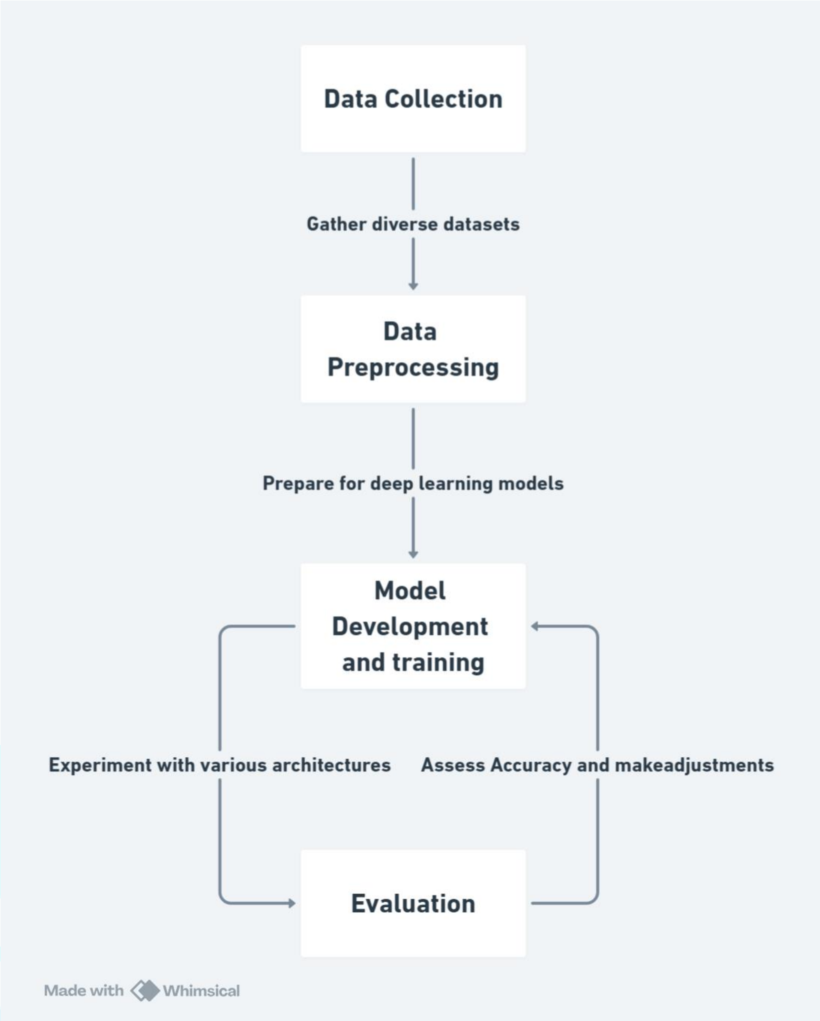

Fig 4.1 Flowchart

The flowchart outlines the process of developing a deep learning model for object detection. It begins with Data Collection, where diverse datasets are gathered. Next, Data Preprocessing is performed to clean and prepare the data for training. The Model Development and Training phase follows, where various architectures are experimented with, and hyperparameters are adjusted. After training, the model undergoes Evaluation, where accuracy is assessed. If needed, adjustments are made, and alternative architectures are tested to improve performance, creating a feedback loop between training and evaluation until optimal results are achieved.

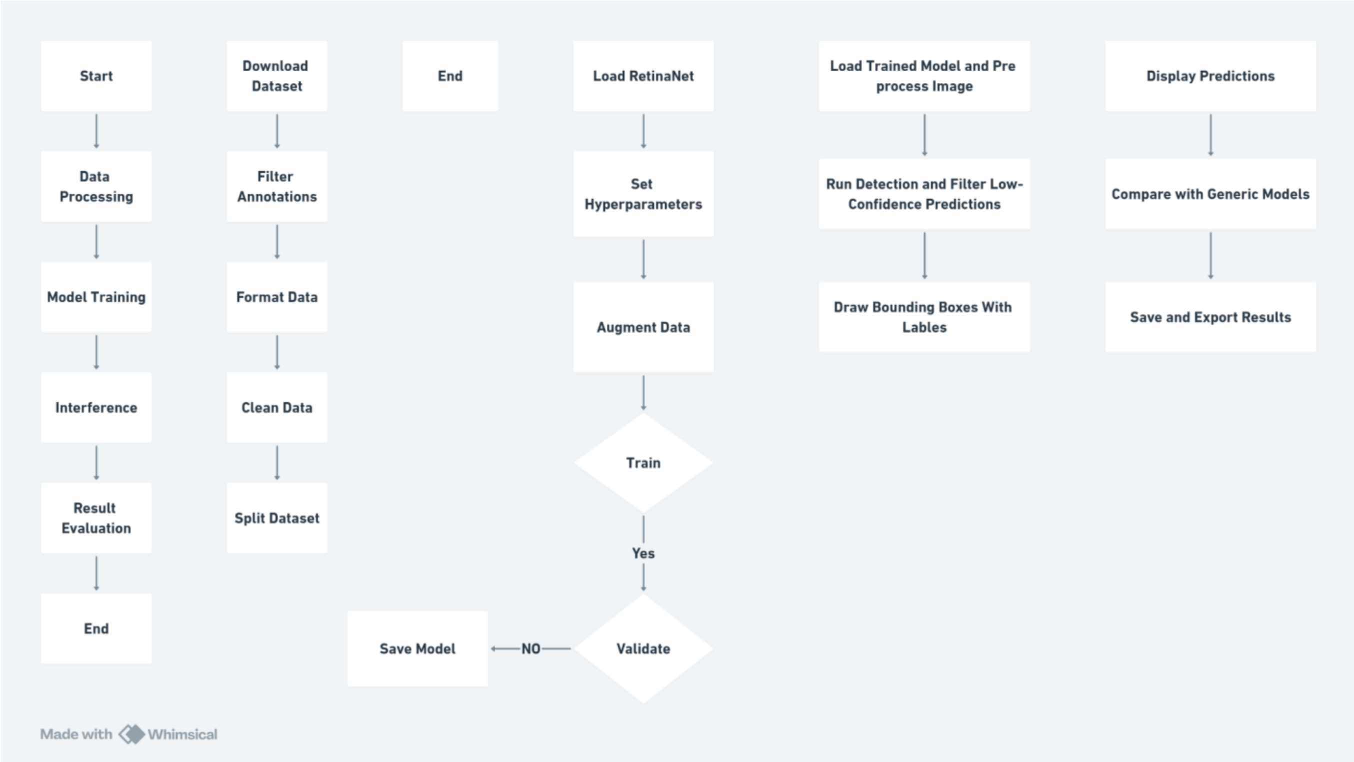

Fig 4.2 Activity Diagram

This activity diagram represents the process of training and evaluating an object detection model using RetinaNet. It begins with the Start node, initiating the workflow. The first step is Data Processing, where raw data is prepared by cleaning, resizing, and transforming it for model training. Next, the Model Training phase begins, allowing the model to learn object detection patterns. Once trained, the model enters the Inference stage, where it makes predictions on unseen images. The Result Evaluation step assesses performance using metrics like accuracy and precision. If satisfactory, the process Ends; otherwise, adjustments are made for improvement. Simultaneously, the dataset is obtained through the Download Dataset step. This data undergoes Filter Annotations to refine labels, followed by Format Data, where the dataset is structured into a usable format. The Clean Data phase removes inconsistencies, ensuring quality input. The dataset is then Split into training, validation, and test sets to optimize model performance.

The next phase involves Loading RetinaNet, the object detection framework. Hyperparameters such as learning rate and batch size are then set. To improve robustness, Data Augmentation techniques like flipping and scaling are applied. Training is then conducted, and if successful, it proceeds to Validation. If the model meets performance criteria, it is Saved for future inference. During the prediction phase, the Trained Model is Loaded, and images are Preprocessed. The model Runs Detection, filtering out low-confidence predictions. Detected objects are visualized by Drawing Bounding Boxes with Labels. These predictions are Displayed, and the results are Compared with Generic Models to measure improvements. Finally, the results are Saved and Exported, concluding the workflow with the End node. This structured process ensures the efficient development, evaluation, and deployment of an object detection model while maintaining data integrity and accuracy.

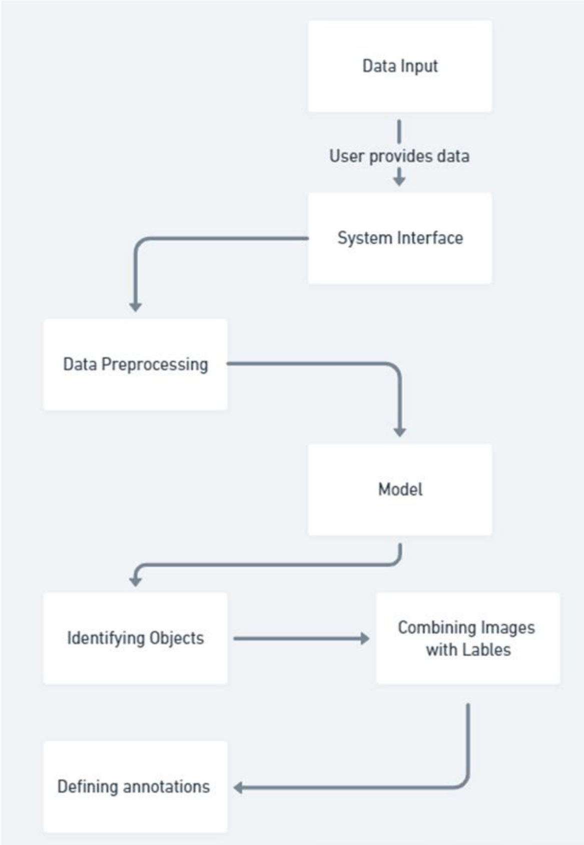

Fig. 4.3 Sequence Diagram

This sequence diagram represents the workflow of an object detection system, outlining the key stages from data input to annotation definition. The process begins with Data Input, where a user provides relevant data, such as images, through a System Interface. This interface acts as a medium between the user and the system, ensuring the data is collected in a structured manner for further processing. Once the data is received, it moves to the Data Preprocessing stage. Here, various transformations take place, such as image resizing, normalization, noise reduction, and other refinements to ensure the data is clean and ready for model training. Preprocessing plays a crucial role in improving the efficiency and accuracy of object detection models. Next, the refined data is passed to the Model, which is responsible for analyzing and identifying patterns. The model processes the input images and begins detecting potential objects based on its learned parameters.

At this stage, the system begins Identifying Objects, where it recognizes objects within the image by extracting key features and assigning categories to detected entities. Once objects are identified, they are Combined with Labels, meaning the detected objects are paired with their respective class names, confidence scores, and bounding box coordinates. This step is crucial for making the results interpretable and usable for further applications. Finally, the labeled objects go through the Defining Annotations stage, where annotations are finalized and structured for use in datasets, visualization, or further machine learning tasks. These annotations help in refining the model's performance by providing accurately labeled training data, which can be used for iterative improvements. Overall, this diagram represents a structured pipeline for object detection, starting from raw data input, processing, model execution, object recognition, and annotation definition, ensuring a seamless workflow for training and improving object detection systems.

V. EXPERIMENTAL SETUP DESIGN

The experimental setup for Stay Smart: Intelligent Amenity Detection for Hospitality Platforms involves a structured approach to data processing, model training, and evaluation. The system is developed using Google Colab with an NVIDIA Tesla GPU (K80/P100/T4) for high-performance computing, while the software stack includes Python 3.8, PyTorch, Detectron2, OpenCV, and Matplotlib. The dataset consists of diverse hospitality images, which undergo preprocessing steps like resizing, normalization, augmentation, and annotation with bounding boxes for accurate amenity detection. The RetinaNet model (Detectron2) is trained using COCO-pretrained weights with optimized hyperparameters, including a learning rate of 0.00125 and a batch size of 2, running for 300 iterations on the registered dataset. Evaluation is conducted using Mean

Average Precision (mAP) and validated through COCOEvaluator, ensuring the model's accuracy. The trained model is then tested on sample images, with detected amenities visualized through bounding boxes and batch processing applied for large-scale validation. This setup guarantees an optimized, scalable, and efficient amenity detection system for hospitality platforms.

MODEL DESIGN

The Stay Smart model uses RetinaNet with Detectron2, fine-tuned on a hospitality dataset. It employs ResNet-50 FPN for feature extraction and Focal Loss for class imbalance. Preprocessing includes resizing, normalization, and augmentation. Training is optimized with SGD (lr = 0.00125, batch size = 2, 300 iterations) and evaluated with COCOEvaluator (mAP metric). The final model enables real-time amenity detection for hospitality platforms, enhancing guest experience and automation.

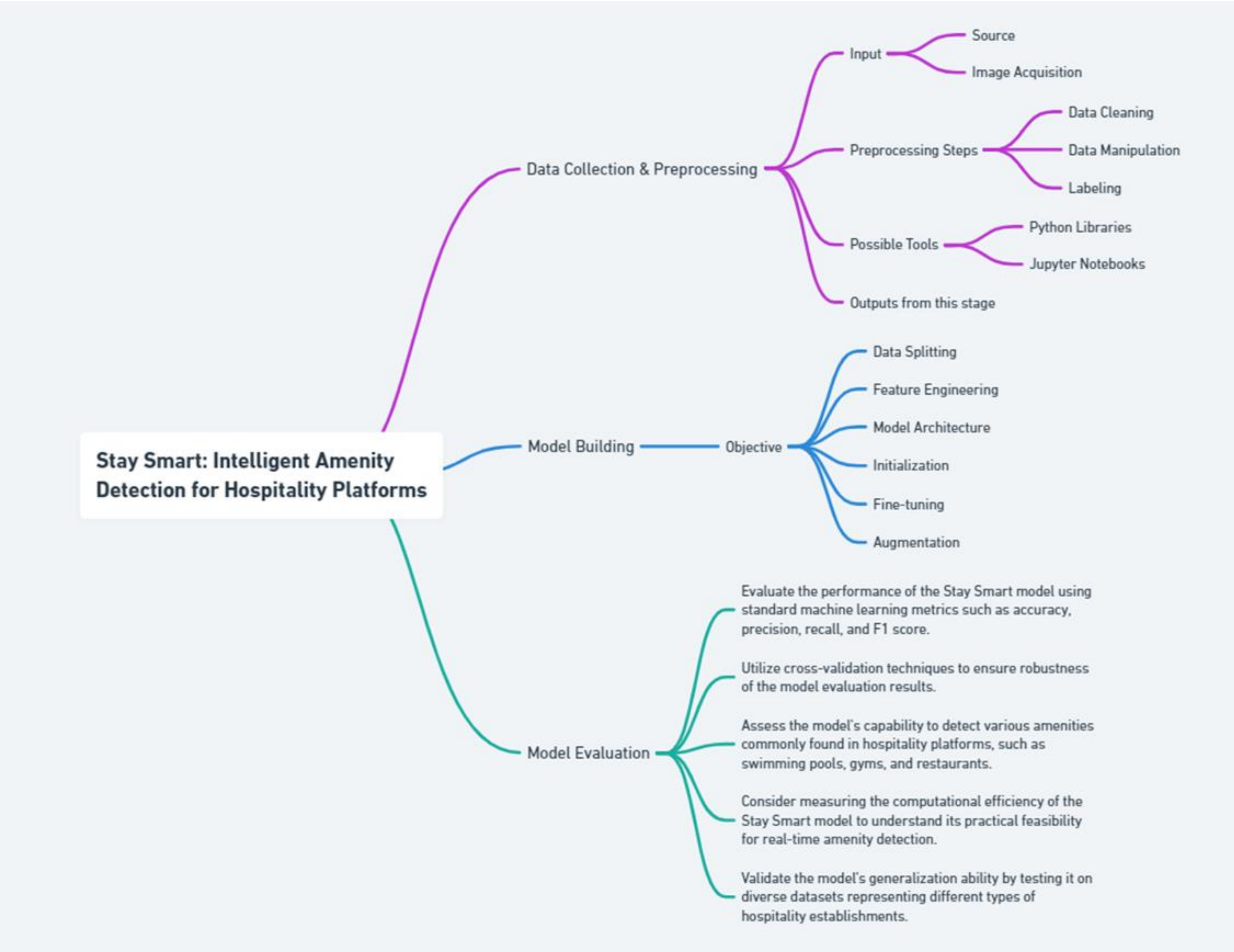

Fig. 5.1 Proposed Model Design

IMPLEMENTATION

The implementation of Stay Smart: Intelligent Amenity Detection for Hospitality Platforms follows a structured approach utilizing Detectron2 and RetinaNet for object detection. The system is built on Google Colab to leverage GPU acceleration, enabling efficient deep learning computations. The training process begins with data preprocessing, where images are resized, normalized, and augmented to improve model generalization. Annotations are formatted in COCO-style datasets for compatibility with Detectron2. The model is configured with pre-trained RetinaNet weights for transfer learning, ensuring a strong baseline for detection. Hyperparameters, including learning rate, batch size, and number of iterations, are fine-tuned for optimal performance. The SGD optimizer and cross-entropy loss function are employed to improve detection accuracy. COCOEvaluator is used to assess model performance, measuring metrics such as mean Average Precision (mAP) to evaluate detection efficiency. For inference, the trained model processes uploaded images and applies bounding boxes around detected amenities. The pipeline filters low-confidence predictions and overlays labels to ensure clarity in results. The final predictions are displayed with visualized annotations, aiding hospitality platforms in automated amenity detection, improving customer experiences and operational efficiency.

VI. RESULT

Fig 6.1 Amenity Detection

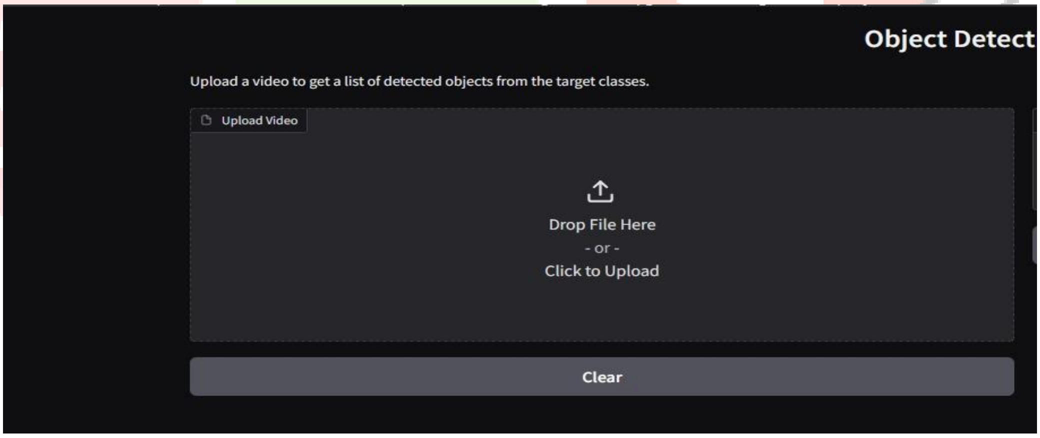

Fig 6.2 Input

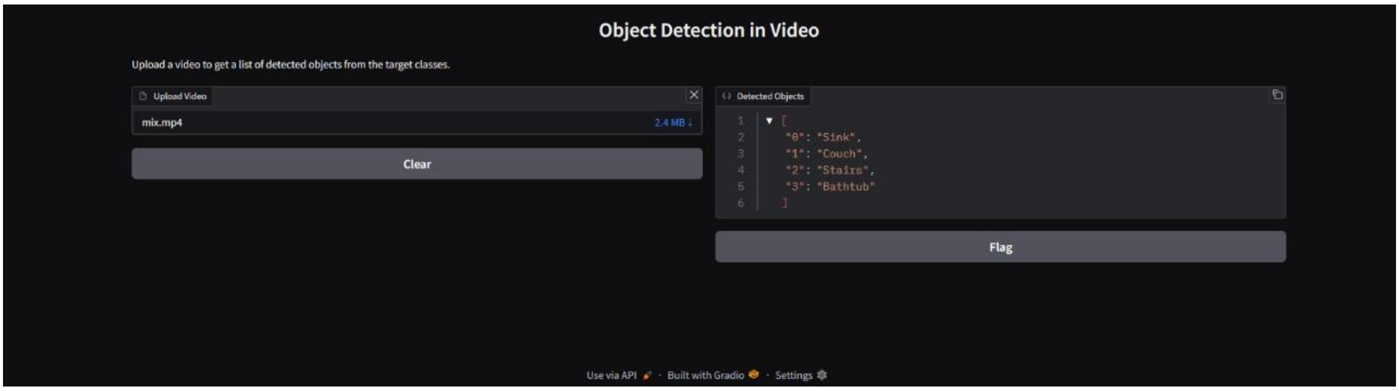

Fig 6.3 UI and Output

VII. CONCLUSION

The Stay Smart: Intelligent Amenity Detection for Hospitality Platforms project successfully leverages deep learning-based object detection to automate amenity identification in hospitality environments. By utilizing Detectron2 and RetinaNet, the system achieves efficient and accurate detection of amenities, reducing the need for manual inspection and enhancing service quality. The integration of COCO-formatted datasets, hyperparameter tuning, and model evaluation ensures robust performance across diverse hospitality settings.

The model's ability to process real-world images, identify amenities with bounding boxes, and filter low-confidence detections makes it a valuable tool for businesses aiming to streamline operations. By deploying this system, hospitality platforms can enhance guest experiences, optimize room management, and improve overall service efficiency through data-driven decision-making.

Future improvements could involve expanding the dataset, integrating real-time detection capabilities, and refining the model for edge deployment on mobile devices. This would further increase accessibility and responsiveness, making amenity detection smarter, faster, and more adaptable to industry needs.

VIII. REFERENCES

[1] Lavanya Dalavai, Khader Basha Sk, Venkateswara Reddy B., M. Venkateswara Rao, April 2023, Machine learning-based pattern recognition and image processing techniques.

[2] Talia Kaufmann, Laura Radaelli, Luis M.A. Bettencourt, Erez Shmueli, 2022, Statistical scaling of urban amenities and their spatial distribution, EPJ Data Science.

[3] Jenny Martinez, Sachit Mahajan, August 2023, Evaluating access to green spaces in smart cities using geospatial analysis, IEEE ACCESS.

[4] Rakesh Goyal, Punam, 2022, Overview of segmentation techniques in image processing, including thresholding, edge detection, and neural networks, IEEE International Conference on Current Development in Engineering and Technology (CCET).

[5] Xinlei Chen, Ross Girshick, Kaiming He, Piotr Dollár, 2019, Dense sliding-window instance segmentation using 4D tensors.

[6] Grant Merz, Yichen Liu, Colin J. Burke, Patrick D. Aleo, Xin Liu, Matias Carrasco Kind, Volodymyr Kindratenko, Yufeng Liu, 2023, Detectron2-based deep learning for object detection and segmentation in astronomical surveys, Monthly Notices of the Royal Astronomical Society (MNRAS).

[7] Allena Venkata Sai Abhishek, Sonali Kotni, August 2021, Object detection using Detectron2 and image manipulation with cartoonization, International Journal of Engineering Research & Technology (IJERT).