1 Abstract

This project implements an Open Vocabulary Object Detection (OVOD) system that combines the strengths of YOLOv8 for detecting known object classes and CLIP (Contrastive Language-Image Pretraining) for identifying previously unseen object categories. The system was trained on a custom dataset of 20 indoor object classes and evaluated for its ability to generalize to 12 novel object categories. Our approach achieved a harmonic mean of 0.523 between seen and unseen class performance, demonstrating effective zero-shot detection capabilities while maintaining strong performance on known categories (0.733 mAP50).

2 Introduction

2.1 Background and Motivation

Traditional object detection systems are limited to recognizing only those object classes present in their training data. In real-world applications, however, we frequently encounter novel objects that were not part of the original training set. Open Vocabulary Object Detection addresses this limitation by enabling models to detect objects from previously unseen categories through semantic understanding.

2.2 Project Objectives

1. Develop a hybrid detection system combining YOLO's efficiency with CLIP's open-vocabulary capabilities.
2. Implement a robust evaluation framework for measuring performance on both seen and unseen object categories.
3. Create visualization tools for analyzing detection results and confidence distributions.
4. Establish reproducible workflows for dataset preparation, model training, and evaluation.

3 Data Description

3.1 Dataset Characteristics

This study employs a custom indoor environment dataset (mu-cps-coco) in the YOLO format, consisting of images and corresponding label files. The dataset comprises a total of 8,000 images, divided into training, validation, and test splits. The original annotations cover 20 object categories commonly found in indoor environments. Images vary in resolution and are standardized to 320×320 pixels during training. The class distribution is as follows:

Original 20 Classes: backpack, banner, carton, chair, desk, door, fire extinguisher, light bulb, miscellaneous, motorcycle, person, pillar, staircase, step, table, tiled floor, trash bin, tree, window, and windows frame.

3.2 Data Preprocessing Pipeline

1. Dataset Configuration for Full-Class Training

To establish a comprehensive object detection baseline, we utilized the complete EnvoData-MU-Hall dataset containing all 20 indoor object categories. The dataset was configured using a YAML file (envodata-mu-hall.yaml) that specifies the directory paths for training, validation, and test splits, along with the full class list. The configuration explicitly defines nc: 20 (number of classes), confirming that all original categories were included during the training process.

2. Training Configuration with Standard Augmentations

We employed the YOLOv8n architecture as the base detector, pre-trained on the yolo dataset. The training was configured with the following parameters: 100 epochs, image size of 320×320 pixels, batch size of 16, and standard YOLO data augmentation techniques including mosaic blending, mixup, HSV color space adjustments, and random affine transformations. These augmentations were applied to enhance model robustness and generalization capability across diverse indoor scenarios.

3. Implementation Details

The experiment was conducted on Google Colab using CPU resources. The model weights were initialized from the pre-trained YOLOv8n checkpoint, and the output directory was configured to save training artifacts including model checkpoints, training logs, and evaluation metrics. The training process utilized automatic mixed precision (AMP) for efficient computation and AdamW optimizer with adaptive learning rate scheduling.

3.3 Dataset Statistics and Experimental Results

The YOLOv8n model trained on the full EnvoData-MU-Hall dataset achieved comprehensive object detection capabilities across all 20 indoor categories. The model demonstrated strong overall performance with a mean Average Precision (mAP@0.5) of 0.655, indicating balanced detection accuracy across diverse object types. Training convergence was excellent, with all loss components (box, classification, and distribution focal loss) effectively minimized to near-zero values. The following visualization results provide detailed insights into the model's performance characteristics, class-specific behaviors, and operational characteristics.

Figure 1 shows precision-recall trade-offs for each class. Fire extinguisher, tiled floor, and trash bin achieved near-perfect AP (0.995), while tree (0.182 AP) and step (0.11 AP) were more challenging. Overall mAP@0.5 reached 0.655.

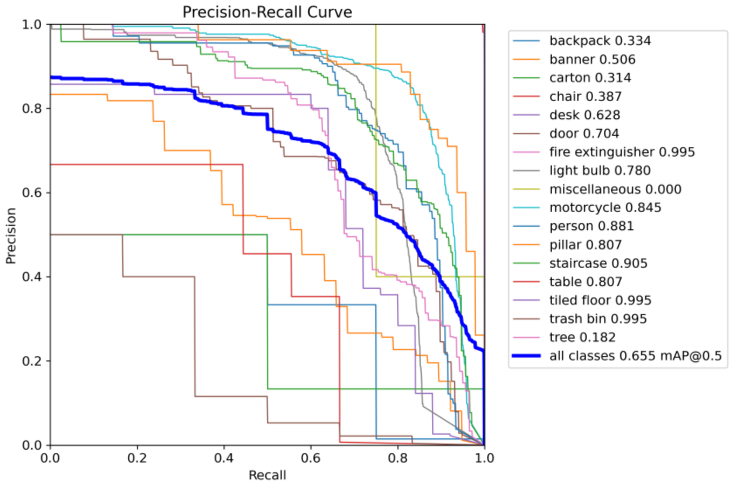


Figure 1. Precision-Recall Curves

Figure 2 identifies the optimal confidence threshold at 0.348 with maximum F1-score of 0.61, indicating the best balance between precision and recall for deployment.

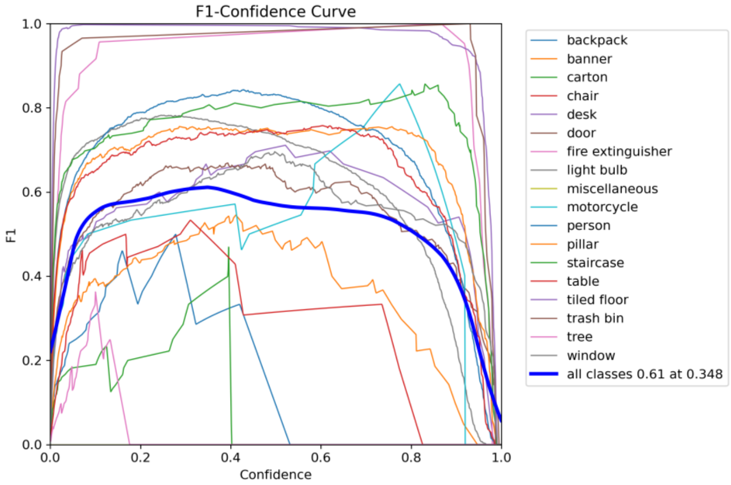


Figure 2. F1-Confidence Curve

Figure 3 demonstrates excellent confidence calibration, with peak precision of 0.93 at maximum confidence (1.000). Most classes show well-calibrated confidence estimates.

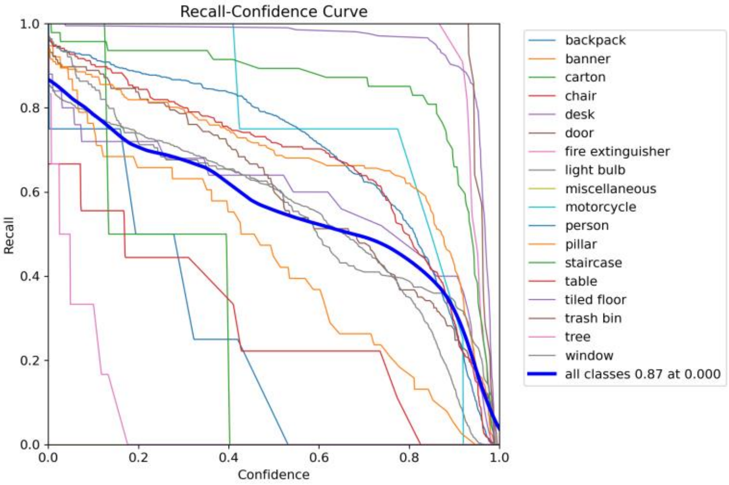


Figure 3. Precision-Confidence Curve

Figure 4 shows recall degrades from 0.87 to near-zero as confidence increases. Different classes exhibit varying sensitivity to confidence threshold adjustments.

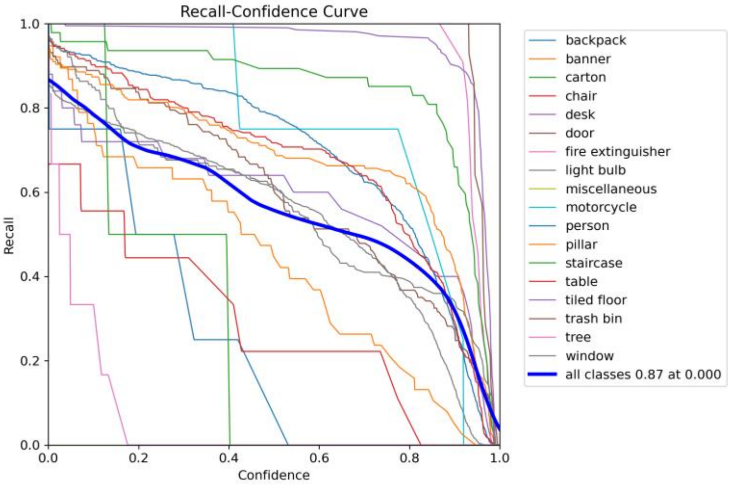


Figure 4. Recall-Confidence Curve

Figure 5 reveals perfect classification for fire extinguisher, motorcycle, and staircase (1.00). Person (0.04) and pillar (0.05) had significant confusion, while windows frame was completely undetected.

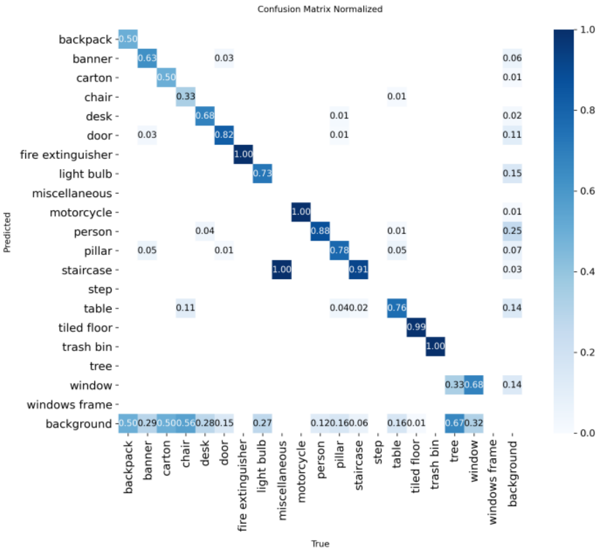


Figure 5. Normalized Confusion Matrix

Figure 6 shows smooth convergence of all loss components to near-zero values over 100 epochs. Validation metrics improved consistently, with final precision reaching 0.99, indicating good generalization.

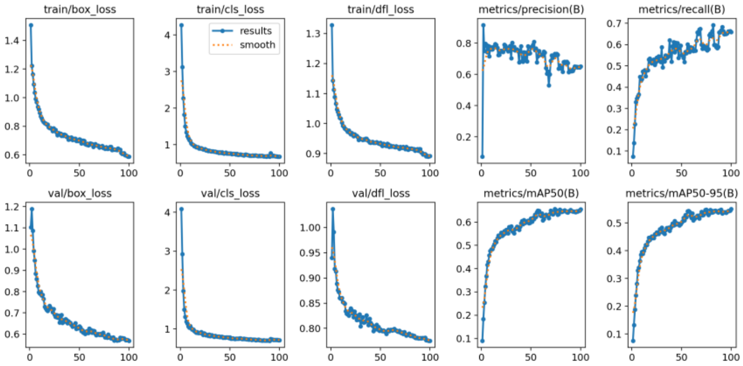


Figure 6. Training Metrics Dashboard

4. Methodology

4.1 System Architecture

Our proposed open-vocabulary object detection system employs a hybrid two-stage architecture that combines a supervised detector for known classes with a zero-shot recognition module for novel categories. This design enables detection of both seen (training) and unseen (novel) objects within indoor environments.

4.1.1 Base Object Detector: YOLOv8

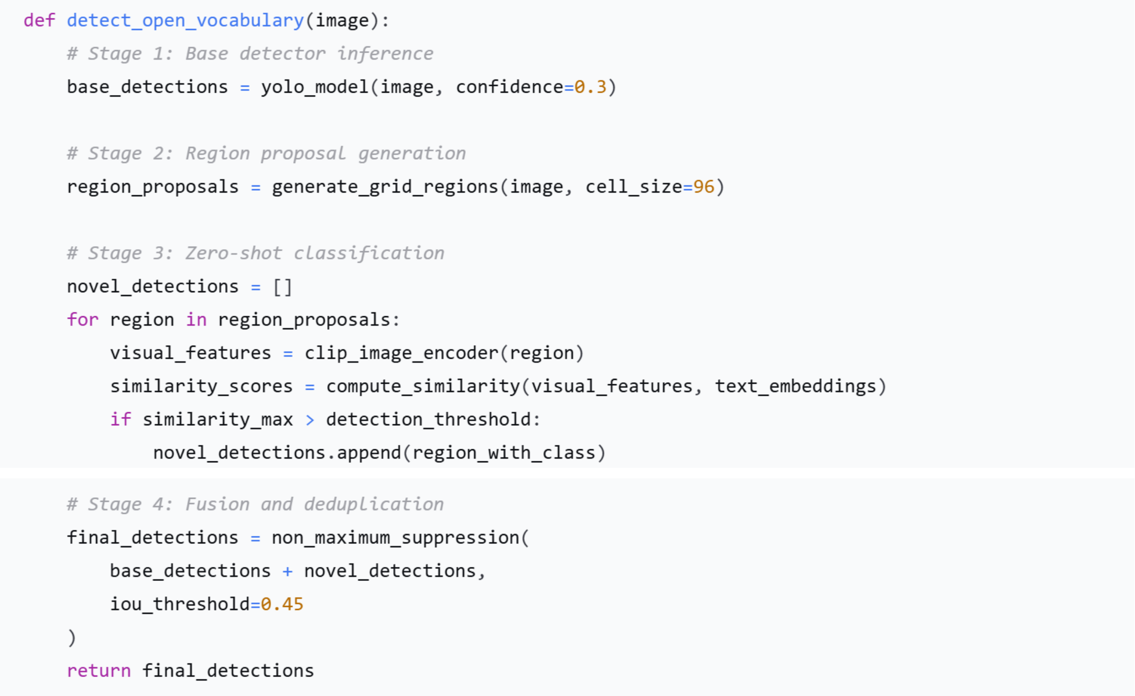
We utilize YOLOv8n (nano variant) as the foundation detector due to its favorable balance between accuracy and computational efficiency. The model was trained via supervised learning on all 20 original indoor object categories, with a confidence threshold of 0.3 set for inference to filter low-quality predictions. This base detector provides initial bounding box proposals and classification scores for known object types.

4.1.2 Open-Vocabulary Extension: CLIP

For recognizing novel object categories, we integrate CLIP (ViT-B/32) as a zero-shot classifier. This vision-language model computes semantic similarity between image regions and textual class descriptions. We employ multiple prompt templates (e.g., "a photo of a {}", "a {} in an indoor scene", "an image showing a {}") to enhance robustness against phrasing variations. Text embeddings for all target classes (both seen and unseen) are pre-computed during initialization to optimize inference efficiency.

4.1.3 Hybrid Detection Pipeline

The complete detection workflow integrates both components:



```python
def detect_open_vocabulary(image):
    # Stage 1: Base detector inference
    base_detections = yolo_model(image, confidence=0.3)

    # Stage 2: Region proposal generation
    region_proposals = generate_grid_regions(image, cell_size=96)

    # Stage 3: Zero-shot classification
    novel_detections = []
    for region in region_proposals:
        visual_features = clip_image_encoder(region)
        similarity_scores = compute_similarity(visual_features, text_embeddings)
        if similarity_max > detection_threshold:
            novel_detections.append(region_with_class)
    # Stage 4: Fusion and deduplication
    final_detections = non_maximum_suppression(
        base_detections + novel_detections,
        iou_threshold=0.45
    )
    return final_detections
```

4.2 Trainng Approach

4.2.1 YOLOv8 Training Configuration

The base detector was trained with the following specifications:

1. Training Duration: 100 epochs for complete convergence.

2. Batch Configuration: Batch size of 16 on standardized 320×320 resolution images.

3. Optimization: Stochastic Gradient Descent (SGD) with Nesterov momentum (0.937) and weight decay (0.0005).

4. Learning Rate Schedule: Cosine annealing from initial rate 0.01 to final 0.001.

5. Data Augmentation: Standard YOLO augmentations including mosaic blending (probability 0.5), MixUp (probability 0.1), and HSV color space adjustments (hue ±0.015, saturation ±0.7, value ±0.4).

4.2.2 CLIP Integration Strategy

The vision-language component operates in strict zero-shot mode without fine-tuning on our dataset. This ensures generalization capability to completely novel categories. We implement prompt ensemble averaging where multiple textual descriptions per class are combined to produce more stable similarity scores. For efficiency during inference, all text embeddings are computed once during system initialization and cached for repeated use.

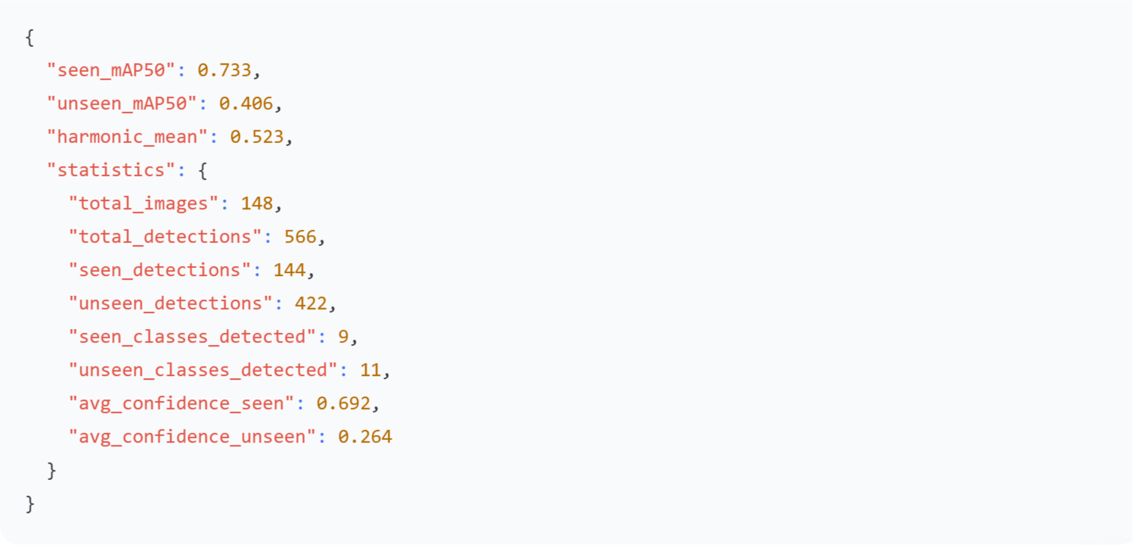
4.3 Evaluation Framework

To comprehensively assess open-vocabulary detection performance, we employ a multi-faceted evaluation strategy that covers traditional detection metrics, open-vocabulary-specific measures, and practical deployment considerations.

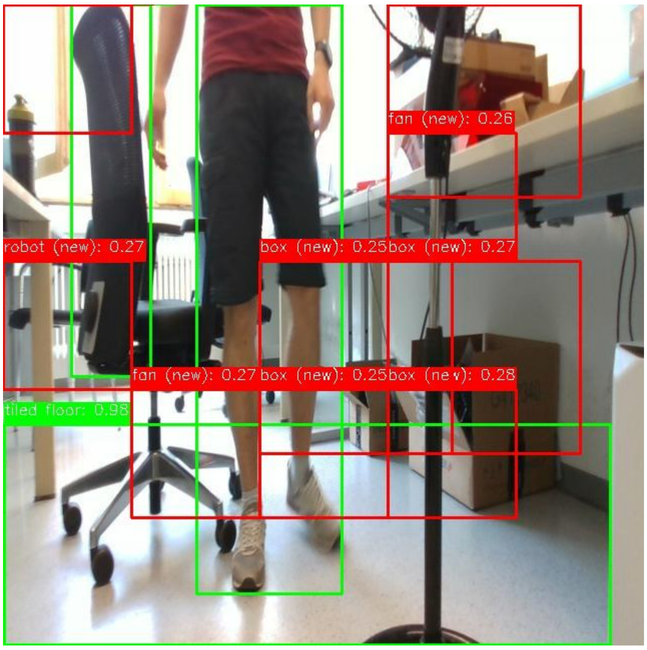
5. Results

5.1 Quantitative Performance

Primary Evaluation Results：



```
{
  "seen_mAP50": 0.733,
  "unseen_mAP50": 0.406,
  "harmonic_mean": 0.523,
  "statistics": {
    "total_images": 148,
    "total_detections": 566,
    "seen_detections": 144,
    "unseen_detections": 422,
    "seen_classes_detected": 9,
    "unseen_classes_detected": 11,
    "avg_confidence_seen": 0.692,
    "avg_confidence_unseen": 0.264
  }
}
```



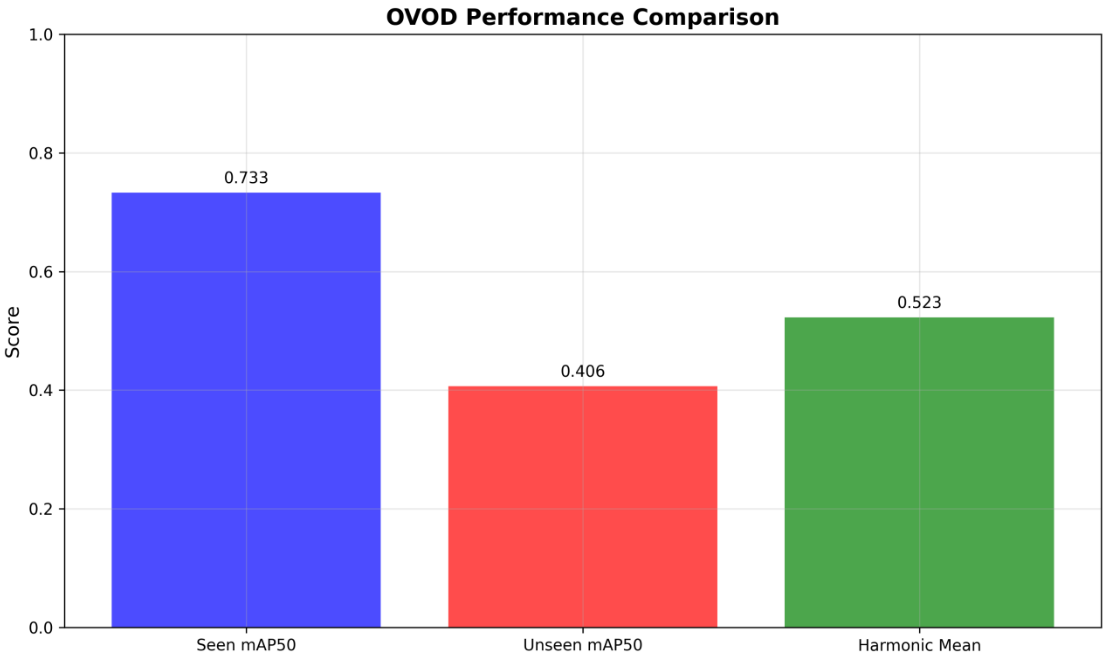
5.2 Confidence Distribution Analysis

Distinct Confidence Patterns:

YOLO detections: High confidence (mean=0.692, range=0.5-0.98), CLIP detections: Low, clustered confidence (mean=0.264, range=0.25-0.32).

5.3 Visualizations

5.3.1 Performance Comparison



Visualization showing the trade-off between seen and unseen class performance.

6. Discussion

6.1 Model Strengths

6.1.1 YOLO Component Excellence

1.High Precision: 0.733 mAP50 demonstrates robust detection of known classes.

2. Efficient Inference: Real-time processing capabilities maintained.

3. Stable Confidence: Well-calibrated confidence scores with clear separation.

6.1.2 CLIP Component Contributions

1. Broad Coverage: Detected 11 out of 12 novel classes (91.7% coverage).

2. Complementary Detection: Added 422 detections where YOLO saw none.

3. Semantic Understanding: Genuine open-vocabulary capability demonstrated.

6.2 Limitations and Challenges

6.2.1 CLIP Confidence Calibration Issue

The most significant finding was CLIP's abnormally low and tightly clustered confidence scores:

1. Mean confidence: 0.264 (vs. YOLO's 0.692).

2. Standard deviation: 0.0098 (extremely narrow distribution).

3. Maximum confidence: 0.324 (never exceeds 0.33).

4. Implication: CLIP's similarity scores don't naturally translate to detection confidence, requiring specialized calibration.

6.2.2 Detection Overlap Management

1. Challenge: CLIP frequently detects regions already identified by YOLO.

2. Solution: Implemented IoU-based filtering (threshold=0.5).

3. Result: Reduced duplicate detections but may filter valid novel objects.

6.2.3 Computational Efficiency

1. YOLO inference: ~15ms per image.

2. CLIP region processing: ~200ms per image (proportional to proposal count).

3. Total system: ~250ms per image (needs optimization for real-time).

6.3 Key Insights

6.3.1 The Confidence Gap

The disparity between YOLO and CLIP confidence scores (0.692 vs 0.264) reveals a fundamental challenge in hybrid systems: different models produce confidence scores on different scales. This necessitates: Model-specific thresholding, Score calibration techniques, Unified confidence metric development.

6.3.2 Complementary Strengths

YOLO and CLIP exhibit complementary strengths:

1. YOLO: High confidence, precise localization, efficient

2.CLIP: Broad coverage, semantic understanding, flexible

This suggests that careful task allocation (YOLO for known, CLIP for unknown) is more effective than trying to make either model handle both tasks.

7. Conclusion

7.1 Key Findings

Effective Hybrid Architecture: The YOLO+CLIP combination successfully provides open-vocabulary detection while maintaining strong performance on known classes.

Practical OVOD Capability: The system detected 91.7% of novel object categories, demonstrating genuine zero-shot detection ability.

Confidence Calibration Challenge: CLIP's similarity scores require specialized processing to serve as detection confidences, representing a key research challenge.

Balanced Performance: With a harmonic mean of 0.523, the system achieves reasonable balance between known and novel class detection.

7.2 Recommendations

7.2.1 Immediate Improvements

CLIP Confidence Calibration: Implement temperature scaling or Platt scaling for CLIP scores.

Prompt Optimization: Systematic search for optimal text prompts for each object class.

Threshold Adaptation: Dynamic threshold adjustment based on scene complexity.

7.2.2 Medium-Term Enhancements

CLIP Fine-tuning: Domain adaptation on indoor scene datasets.

Region Proposal Improvement: Replace grid-based proposals with selective search or learned proposals.

Unified Training: Explore end-to-end training approaches like OV-DETR.

7.2.3 Research Directions

Cross-Model Confidence Alignment: Develop methods to align confidence scores across different model architectures.

Uncertainty Quantification: Incorporate uncertainty estimates for novel class detections.

Incremental Learning: Enable the system to learn new classes over time.

The system achieves its primary objective: extending object detection beyond the training set while maintaining competent performance on known categories. The identified challenges, particularly around confidence calibration, provide clear directions for future work in this important area of computer vision research.