Intelligent Landmark Detection System

Prajwal Mrithyunjay Hulmani
Dept. of Computer Science and Engineering
University of Texas at Arlington
Arlington, USA

Karan Bhavsar
Dept. of Computer Science and Engineering
University of Texas at Arlington
Arlington, USA

Abstract

This project presents a unified framework for intelligent landmark localization and recognition in natural images. Our system integrates a carefully engineered data pipeline, a modern one-stage detector optimized for fast inference, and a benchmark comparison against a two-stage Faster R-CNN model. All components follow standardized COCO-style formatting to ensure reproducibility and fair evaluation. We assess model outputs using mean Average Precision (mAP), precision, recall, and inference speed. Experimental results show that the optimized one-stage model offers higher mAP at IoU 0.50 and significantly faster runtime, making it suitable for interactive or real-time scenarios. The report details dataset preparation, model architecture, training methodology, and visualization utilities, followed by a discussion of strengths, limitations, and future extensions.

---

1. Introduction

Recognizing man-made landmarks in photographs is central to use cases such as cultural heritage retrieval, tourist assistance, and AR-based guidance. For such systems to be deployable in practice, it is not enough to have a strong detector; they also need a well-structured pipeline that unifies data preparation, training, evaluation, and visualization so that results are consistent, comparable, and easy to interpret. In this project, we build a complete landmark detection framework that pits a custom one-stage detector against a Faster R-CNN baseline under aligned experimental settings. The pipeline applies the same preprocessing and evaluation procedures to both models and offers a lightweight interface that shows their predictions side by side on a given image.

Our contributions are threefold: (i) a reproducible COCO-based dataset and preprocessing workflow; (ii) a one-stage detector optimized for fast inference and accurate landmark localization; and (iii) a unified evaluation and visualization framework that supports both quantitative metrics and qualitative comparison of model predictions.

---

2. Related work

Two-stage detectors such as Faster R-CNN achieve strong localization by first proposing candidate regions and then refining them via classification and box regression. One-stage models (e.g., RetinaNet, YOLO) instead make dense predictions in a single pass, trading some complexity for speed and real-time suitability. The MS COCO benchmark and its metrics provide a common standard for evaluating detection models, while advances like Feature Pyramid Networks and residual backbones have improved multi-scale reasoning and trainability. Within this landscape, our work compares a custom one-stage detector to a Faster R-CNN baseline using COCO-style data and metrics for landmark detection, where targets vary substantially in scale, symmetry, and scene clutter.

---

3. Implementation

3.1 Dataset and preprocessing

The system operates on COCO-style JSON annotations with separate train and validation splits under data/{train,val}/images and matching annotations.json files. All images are resized to 640×640, using aspect-ratio–preserving padding when needed. Data augmentation consists of random horizontal flips, light color jitter, and moderate scale jitter within a controlled range so that bounding boxes remain reliable. Class names are read either directly from the annotation file or from an associated configuration list.

3.2 Models

- *Custom detector: A modular, one-stage, anchor-based detector with multi-scale prediction heads. For each predefined anchor at appropriate feature-map strides, the network outputs class logits and bounding-box regressions. Boxes are represented in normalized form* $[x_1, y_1, x_2, y_2] \in [0,1]$.

- *Faster R-CNN baseline: A conventional two-stage detector with a convolutional backbone (optionally equipped with an FPN). It produces bounding boxes in absolute pixel coordinates* $[x_1, y_1, x_2, y_2]$.

3.3 Training

Models are trained with the AdamW optimizer using an initial learning rate of 5×10^{-4}*, cosine learning-rate decay, and three warmup epochs. We use a batch size of 8 and enable automatic mixed precision to increase training throughput. Each model is trained for 50 epochs, and the checkpoint with the best validation mAP is retained. Non-maximum suppression (NMS) with IoU = 0.5 is applied unless noted otherwise. Random seeds are fixed to keep experiments reproducible.*

3.4 Evaluation and visualization

On the validation split, we report mAP@0.50, mAP@0.50:0.95, precision, recall, and inference frames per second (FPS). A visualization tool renders predictions from both detectors on the same input image. Normalized boxes from the custom model are converted to pixel coordinates using the original image dimensions so they can be directly compared with Faster R-CNN outputs. A lightweight app (e.g., app.py) provides a simple interface to upload an image and display each model's bounding boxes, labels, and confidence scores.

---

4. Results

4.1 Overall metrics

The table below summarizes comparative performance on the validation set.

Model	mAP@0.50	mAP@0.50:0.95	Precision	Recall	Inference FPS	Parameters
Custom Detector	0.71	0.44	0.83	0.78	42	24M
Faster R-CNN	0.67	0.41	0.80	0.74	18	41M

The custom detector yields higher mAP@0.50 and significantly higher throughput, indicating suitability for latency-sensitive scenarios such as interactive demos or real-time assistance.

4.2 Class-wise performance

We report per-class AP at IoU 0.50 for representative classes.

Class	Custom Detector	Faster R-CNN
Taj_Mahal	0.78	0.73
Eiffle Tower	0.70	0.66

4.3 Qualitative analysis

Overlay visualizations show the custom detector typically *produces tighter boxes around central domes and facades, reflecting strong localization on medium-to-large structures. Faster R-CNN can sometimes capture broader context in cluttered scenes, aiding detection of distant or small instances but occasionally reducing box tightness. Simple ensembling via NMS merging provided a modest +0.02 gain in mAP@0.50 during preliminary tests.*

4.4 Example prediction structure

The viewer accepts combined outputs from both models in a simple structure. For a 600×400 image, normalized custom boxes are scaled to pixels for rendering.

- *Custom: boxes = [[0.25, 0.15, 0.75, 0.85]], labels = ["Taj_Mahal"], scores = [0.92] → pixels: [150, 60, 450, 340]*
- *Faster R-CNN: boxes = [[120, 80, 520, 420]], labels = ["Taj_Mahal"], scores = [0.88]*

---

5. Discussion

The superior runtime of the custom detector is largely due to its single-stage, dense-prediction architecture combined with mixed-precision training. Its higher accuracy at IoU 0.50 indicates strong localization performance, especially on the larger landmark structures that dominate the dataset. The Faster R-CNN baseline, however, remains a strong contender, particularly in scenarios where its proposal mechanism can better capture small or partially occluded landmarks. Current limitations of the system include sensitivity to extreme aspect ratios and weaker performance on very small objects. Potential extensions include adopting more powerful backbone networks, exploring test-time augmentation strategies, and applying techniques such as weighted box fusion to further refine predictions.

---

6. Conclusion

This work introduced a reproducible pipeline for landmark detection that trains a custom one-stage detector and benchmarks it against a Faster R-CNN baseline under consistent conditions. By unifying data processing, evaluation metrics, and visualization tools, the framework supports both rigorous quantitative analysis and intuitive qualitative inspection of model outputs. Experimental results indicate that the custom detector achieves higher mAP@0.50 along with substantially better inference speed, making it well-suited for interactive or real-time applications. Future directions include enhancing multi-scale feature representations, exploring semi-supervised fine-tuning for underrepresented landmarks, and optimizing the system for deployment on resource-constrained platforms.

References

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," Advances in Neural Information Processing Systems (NeurIPS), 2015.

[2] T.-Y. Lin, M. Maire, S. Belongie, et al., "Microsoft COCO: Common objects in context," European Conference on Computer Vision (ECCV), 2014.

[3] T.-Y. Lin, P. Dollár, R. Girshick, et al., "Feature pyramid networks for object detection," IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, real-time object detection," IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2016.