Street Parking Presence Inference from Street-Level Imagery via Multi-Cue Detection and Geo-Aggregation

Midterm Progress Report — CS 766

Chirag Jain (9087168606) Ritik Singh (9087047321)
University of Wisconsin-Madison

March 25, 2026

Note on writing assistance. This report was prepared with the assistance of Claude/ChatGPT for drafting, restructuring, and language refinement. All the content, results, experimental details, observations were shared with the LLMs to generate a coherent and structured report. All technical content, experiments, results, figures, and final edits were reviewed and verified manually.

1 Project Goal and Mid-Term Summary

Our project studies street parking presence inference from street-level imagery. The original proposal framed the task as a two-stage pipeline: first detect image-level parking-related cues such as parking signs, parking meters, and curb cues; then aggregate evidence across nearby views to infer whether a street segment supports on-street parking. The key motivation is that parking cues are often small, sparse, occluded, and viewpoint-dependent, so single-image predictions can be brittle.

At the mid-term stage, our main completed milestone is a strong **parking sign detection baseline**. We prepared a YOLO-format dataset from MTSD, trained a YOLOv8m detector, built a custom evaluation pipeline for image-level presence inference, and analyzed both quantitative and qualitative failure modes. Our best validation result so far is:

- **mAP@50 = 0.5487**
- **mAP@50–95 = 0.3824**
- **Image-level F1 = 0.6673**
- **AUROC = 0.8310**

In parallel, we also explored a second cue: **parking meter detection**. We analyzed Mapillary Vistas v2.0 to identify parking-meter classes and then ran a preliminary **zero-shot parking-meter evaluation** using a COCO-pretrained YOLO11x model, since it has parking-meter as one of the classes. This experiment showed that parking-meter detection has some transferable signal, but currently suffers from many false positives and should be treated as a **weak auxiliary cue** rather than a reliable standalone detector.

These results indicate that the parking-sign detector is already a useful baseline for the downstream aggregation stage, but also that single-image detection still misses a meaningful fraction of

positives. This is consistent with our original hypothesis and strongly motivates the next phase of segment-level aggregation.

2 Progress Since the Proposal

2.1 Dataset Preparation and Label Mapping

We selected the Mapillary Traffic Sign Dataset (MTSD) as the main dataset for the first cue detector. The released data provides a large number of traffic sign annotations, but it is not directly packaged for our task. We therefore created a binary parking-sign detection dataset by mapping multiple parking-related sign variants (`information-parking-g1, information-parking-g5`) into a single class called `parking_sign`.

The final mapping includes multiple no-parking, no-stopping, parking information, tow-away, and parking-restriction sign variants. We excluded the generic `other-sign` label because it has no consistent visual identity and would introduce substantial noise. We also filtered out annotations marked as ambiguous, occluded, out-of-frame, or dummy.

After preprocessing, our local MTSD setup contained:

- 52,453 images on disk
- 41,909 annotation files
- split files listing 36,589 train, 5,320 validation, and 10,544 test image IDs

In practice, the mid-term experiments rely on the **validation split** because the available labeled annotations correspond to train/validation data. The test split IDs exist in the released split files, but it is not directly usable for our current local evaluation pipeline because of unavailable annotations.

2.2 Exploratory Data Analysis

We ran an exploratory analysis script to better understand class balance and object scale. The most important findings are:

- Total raw labeled objects: 206,386
- Total clean objects after filtering: 63,806
- Positive images after filtering: 4,446 ($\approx 10.6\%$)
- Negative images after filtering: 24,390
- Effective negative-to-positive ratio: $\approx 5.5 : 1$
- Total positive parking-sign instances: 5,983
- Tiny signs ($< 0.1\%$ of image area): 48,317 ($\approx 75.7\%$)
- Median relative object area: 0.031% of the image

These statistics were important for interpreting the detector behavior. The main challenge is not only class imbalance, but also the fact that most signs are **very small objects** in large street-view images.

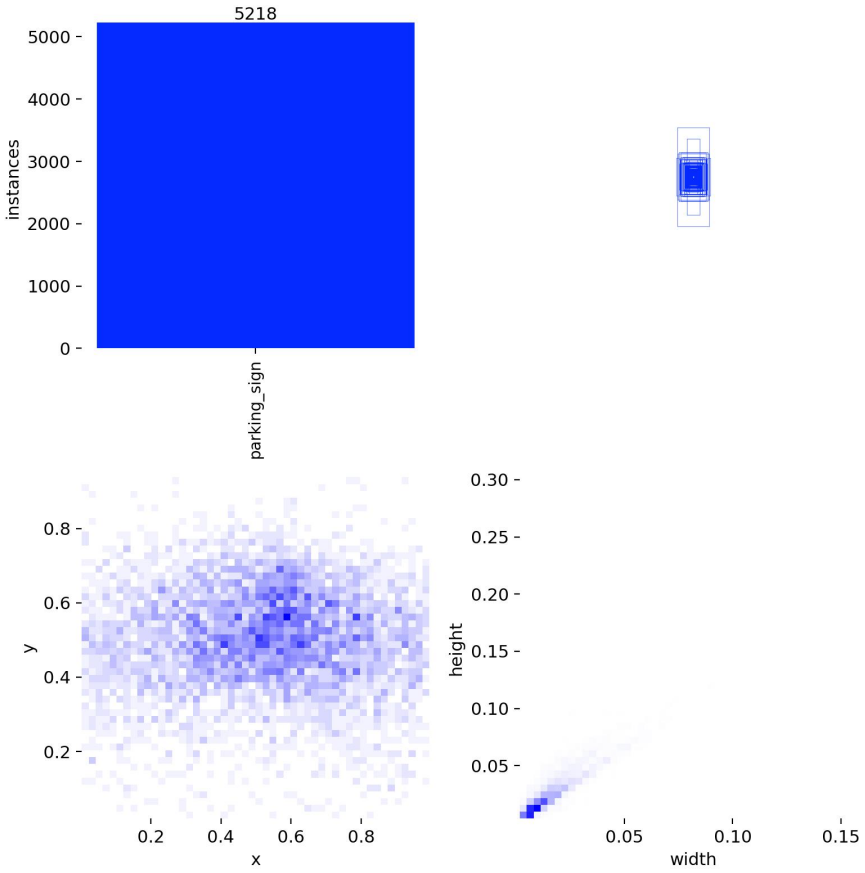


Figure 1: Label distribution and bounding-box size statistics from the processed parking-sign dataset. Most parking signs are very small relative to the image, which makes the task a difficult small-object detection problem.

We also analyzed **Mapillary Vistas v2.0** to understand which parking-related cues are actually available beyond traffic signs. This was important because our final project aims to use multiple cues. We found that Vistas contains 123 unique classes, including explicit parking-related objects such as:

- `object-parking-meter` (839 instances),
- `object-traffic-sign-information-parking` (3,418 instances),
- `construction-flat-parking` (3,355 instances),
- `construction-flat-parking-aisle` (247 instances).

At the same time, the dataset contains much richer curb-related structure:

- `construction-barrier-curb` (59,767 instances),

- `construction-flat-curb-cut` (17,582 instances).

This EDA led to an important project insight: **explicit parking objects in Vistas are relatively sparse, while curb-related scene structure is much more abundant**. This suggests that for the final project, curb cues may be a stronger second signal than parking meters, while parking-meter detection is best viewed as a supplementary cue.

2.3 Training Pipeline

We implemented a full training pipeline using YOLOv8m. The model was trained on Kaggle with the following main configuration:

- Model: YOLOv8m
- Image size: 640
- Epochs: 50
- Batch size: 32
- Class loss weight: 2.0
- Augmentations: mosaic, mixup, copy-paste

Training had to be completed in two phases. The first run covered epochs 1–24, after which the Kaggle session crashed. We then resumed with a second run from epoch 25 onward and combined the results afterward for analysis. Even though the process was interrupted, the final training curves were smooth and consistent, suggesting that the resumed training behaved as expected.

From Figure 2, several trends are clear:

- Training and validation losses decrease steadily.
- Precision stabilizes in the mid-0.6 range.
- Recall improves more gradually, reflecting the difficulty of recovering small distant signs.
- mAP@50 and mAP@50–95 continue improving through the later epochs, although with diminishing returns near the end.

This suggests that the model is learning useful signal and not obviously overfitting, but that the task remains challenging.

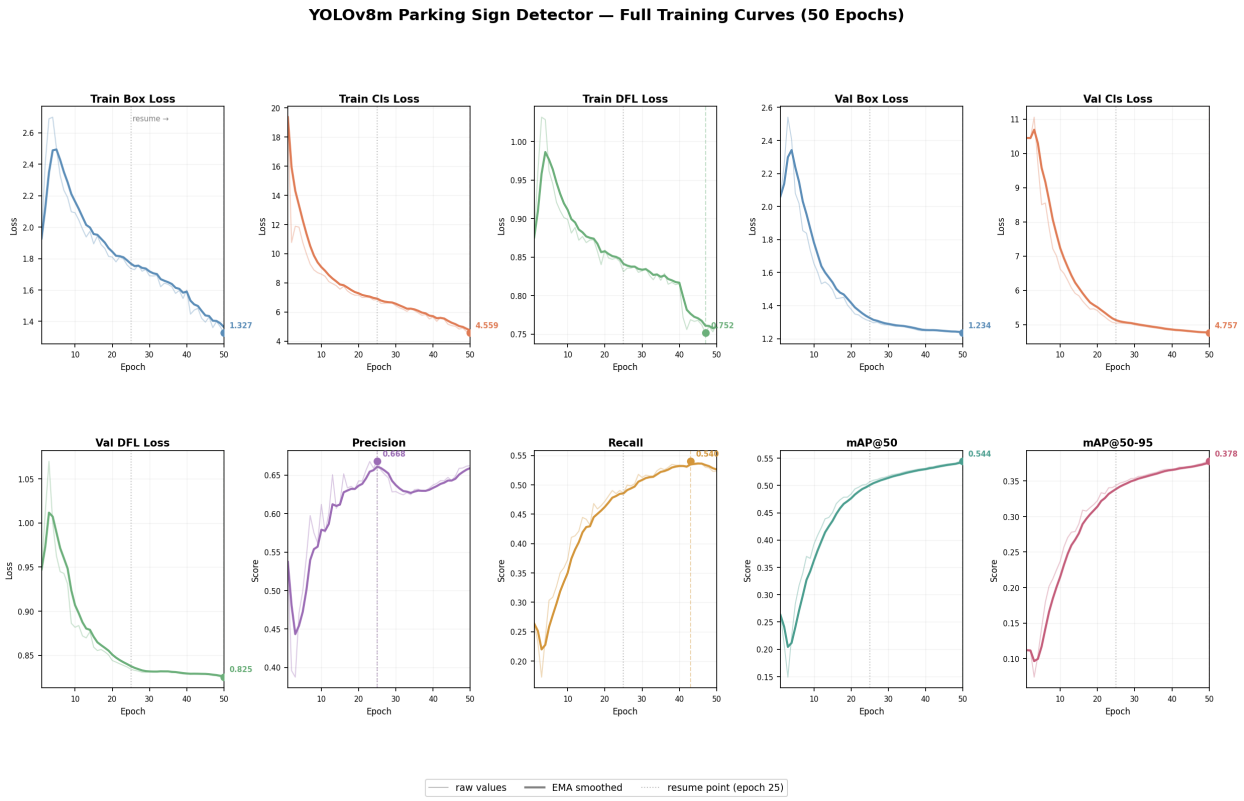


Figure 2: Combined training and validation curves over 50 epochs. The run was completed in two stages due to infrastructure interruptions, but the losses and validation metrics show stable convergence.

3 Evaluation Setup

A major change from the initial plan is that we built a custom **image-level evaluation script** in addition to standard YOLO validation. This was necessary because our downstream task is not only bounding-box localization; we ultimately care about whether an image contains parking-related evidence that can later be aggregated over a street segment.

Our evaluation therefore has two parts:

1. **Box-level evaluation:** standard YOLO detection metrics such as mAP, precision, and recall.
2. **Image-level evaluation:** for each image, we take the maximum detection confidence across all predicted boxes and convert it into a binary parking-presence score. This allows us to compute precision, recall, F1, and AUROC at the image level.

This second evaluation is especially important because it better matches the next stage of our project: segment-level aggregation.

For parking-meter evaluation, we used a similar idea in a cross-dataset zero-shot setting. We took a COCO-pretrained detector, restricted it to the parking-meter class, and evaluated it against Mapillary Vistas parking-meter annotations. Ground-truth parking-meter polygons were converted into bounding boxes so that we could compute both image-level and box-level metrics.

4 Current Quantitative Results

4.1 Final Validation Results (Epoch 50 Best Checkpoint)

Our final validation results from the best checkpoint after 50 epochs are shown in Table 1.

Table 1: Main validation results for the parking sign detector.

Metric	Value
mAP@50	0.5487
mAP@50–95	0.3824
Precision (box-level)	0.6616
Recall (box-level)	0.5373
Best image-level threshold	0.15
Best image-level F1	0.6673
Image-level AUROC	0.8310

The results look good based on the data and resources we had. The box-level metrics show that the model is learning to localize parking signs reasonably well, while the image-level F1 and AUROC show that it can separate positive and negative images with useful reliability.

4.2 Threshold Analysis

We also swept the confidence threshold to study the precision-recall trade-off at the image level. Selected results are shown in Table 2.

Table 2: Image-level threshold sweep on the validation set.

Threshold	Precision	Recall	F1	Accuracy
0.15	0.6913	0.6448	**0.6673**	0.9299
0.20	0.7328	0.6052	0.6629	0.9329
0.30	0.7565	0.5517	0.6381	0.9318
0.50	0.8262	0.4672	0.5969	0.9312
0.70	0.9150	0.3155	0.4692	0.9222

The best operating threshold is around 0.15. This is relatively low, which indicates that many useful detections are not extremely high-confidence. In other words, if we use a very strict confidence threshold, recall drops sharply and many parking cues are lost. This is exactly the kind of behavior that makes aggregation valuable: several weak or partial detections across nearby images may still provide strong segment-level evidence.

4.3 Training Progress Over Time

To show intermediate progress clearly, Table 3 summarizes key checkpoints during training.

Table 3: Progression of validation performance across checkpoints.

Checkpoint	mAP@50	Image-level F1	AUROC
Epoch 20	0.4835	0.6430	0.8088
Epoch 25	0.5065	0.6535	0.8226
Epoch 50	**0.5487**	**0.6673**	**0.8310**

The model improved steadily throughout training. However, the gains from epoch 25 to epoch 50 were smaller than the gains earlier in training, suggesting that the detector is approaching a plateau. This is useful from a project-planning perspective: it suggests that future improvements are now more likely to come from **aggregation and additional cues** than from extensive further tuning of the sign detector alone.

4.4 Preliminary Zero-Shot Parking-Meter Results

In addition to the trained parking-sign baseline, we performed a preliminary zero-shot experiment for parking-meter detection. The idea was to test whether a strong COCO-pretrained detector already transfers well enough to street-view parking-meter imagery to serve as a useful second cue.

We used a COCO-pretrained YOLO11x model and evaluated it on **Mapillary Vistas validation images**, using the class `object-parking-meter` as ground truth. The full validation set contained:

- 2,000 validation images,
- 50 positive images containing at least one parking meter,
- 1,950 negative images.

Before evaluating on the full validation set, we first ran a **positives-only sweep** to understand the effect of inference resolution and confidence threshold. That sweep showed that larger input resolution substantially improves recall, which is consistent with our qualitative observation that parking meters are often very small in street-view imagery. Based on this analysis, we selected `imgsz=1280` for the full run.

The final full-validation results are shown in Table 4.

Table 4: Zero-shot parking-meter evaluation on Mapillary Vistas validation set using COCO-pretrained YOLO11x.

imgsz	conf	Img P	Img R	Img F1	Box P	Box R	Box F1
1280	0.05	0.109	0.520	0.181	0.0469	0.168	0.0734
1280	0.10	**0.158**	0.480	**0.238**	**0.0758**	0.158	**0.1020**

For the better setting, `imgsz=1280, conf=0.10`, the detailed counts were:

- image-level: TP = 24, FP = 128, TN = 1822, FN = 26
- box-level: TP = 15, FP = 183, FN = 80

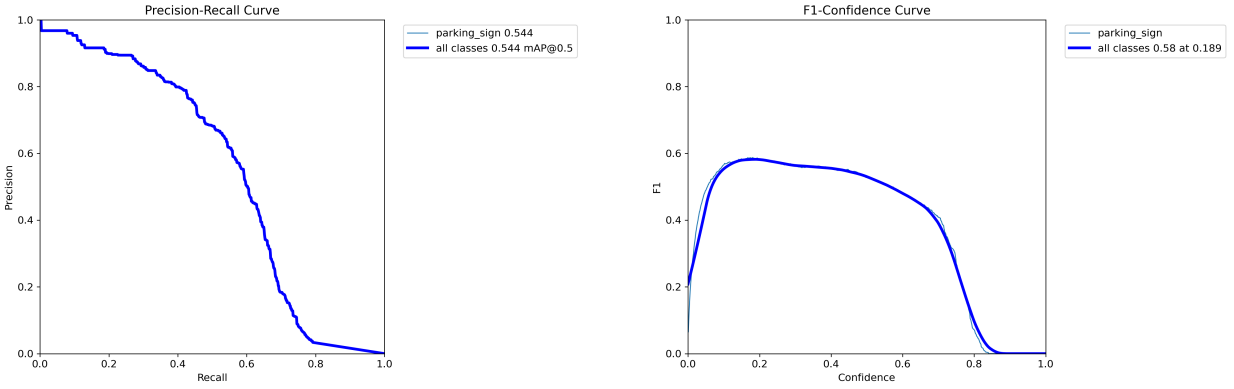
These results lead to three main conclusions:

- **Zero-shot transfer is real but limited.** The detector recovers nearly half of positive parking-meter images at the image level, so it is not behaving randomly.
- **Precision is poor.** The detector produces many false positives, especially on pole-like objects and other narrow vertical street furniture.
- **Parking meters are therefore a weak auxiliary cue.** They may still contribute useful evidence in a multi-cue system, but they are not reliable enough to serve as a primary cue by themselves.

Between the two settings, `conf=0.10` gives the better overall trade-off because it substantially reduces false positives while only slightly reducing recall. This makes it the more reasonable operating point if parking-meter scores are later incorporated into cue fusion or aggregation.

5 Additional Validation Plots

The standard YOLO validation plots are also informative and will be included in the final report appendix or supplementary material.



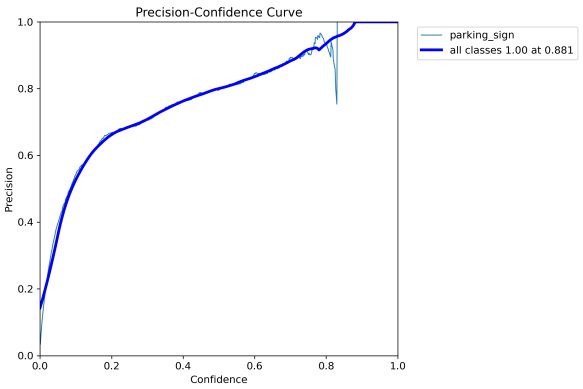
(a) Precision–recall curve

(b) F1–confidence curve

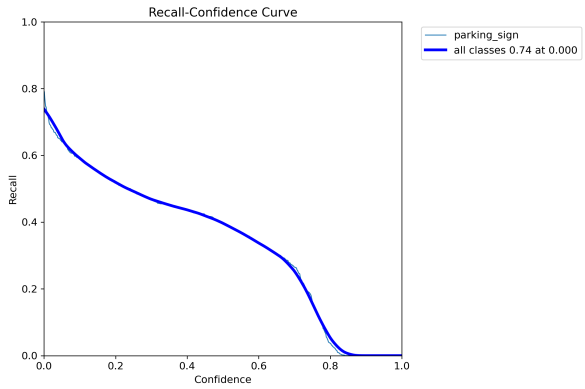
Figure 3: Detection performance curves from YOLO validation.

Figure 3 supports several observations:

- The precision–recall curve confirms moderate but useful detection performance, with mAP@50 around 0.54.
- The F1–confidence curve shows that the detector works best at a relatively low confidence threshold, again indicating that retaining weaker detections is important.



(a) Precision–confidence curve



(b) Recall–confidence curve

Figure 4: Precision and recall behavior as the confidence threshold changes.

Figure 4 shows that precision increases as the threshold becomes stricter, but recall drops rapidly. This is another sign that strict single-image decision rules are not ideal for our final problem setting.

6 Qualitative Findings and Error Analysis

A useful part of our mid-term progress was not just the final numbers, but also the qualitative analysis of *why* the model succeeds or fails.

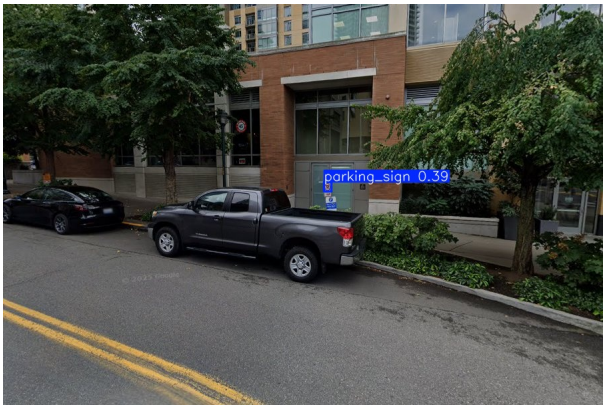
6.1 Resolution and Scale Sensitivity

One of the clearest findings from our experiments is that the detector is strongly limited by object scale. We tested the same scene at different effective scales and observed the following behavior:

- Figure 5a shows that at **640** resolution on the zoomed-out image, the model missed both parking signs.
- Figure 5b shows that at **960** resolution on the same zoomed-out image, the model detected one of the signs.
- Figure 5c shows that at **1280** resolution on the same zoomed-out image, the model detected both signs.
- Figure 5d shows that on a manually zoomed-in crop, the model detected the signs more reliably even at lower inference size (imgsz=640).

This is a strong confirmation of the dataset analysis: the detector is **resolution-limited rather than concept-limited**. In other words, the model has learned what parking signs look like, but in wide street scenes the signs are often too small after image resizing for reliable detection.

(a) Original Image at imgsz=640 (miss)



(b) Original Image at imgsz=960 (partial detection)

(c) Original Image at imgsz=1280 (both signs detected)

(d) Original Image Manually Zoomed at imgsz=640 (Full detection)

Figure 5: Detection results for Seattle image (near Cornish College of the Arts) showing the same scene under different effective scales. This figure illustrates that detection success improves as the parking signs occupy more pixels in the model input.

This finding is important because it directly supports our project motivation. If a single view is zoomed out, the cue may be missed; if another nearby view captures the sign more closely, the cue may become detectable. This is exactly why multi-view aggregation should help.

6.2 Partial Localization of Composite Signs

Another recurring observation is that the detector often boxes only the most salient sub-part of a parking sign assembly, such as the blue "P" symbol or a no-parking icon, rather than the full stacked signboard including time restrictions.

This behavior is not surprising for two reasons:

1. MTSD is a traffic-sign dataset, so annotations are naturally sign-centric rather than designed for full sign-assembly understanding.
2. Text-heavy restriction plates are smaller and more variable than the main symbol panel, making them much harder to learn.

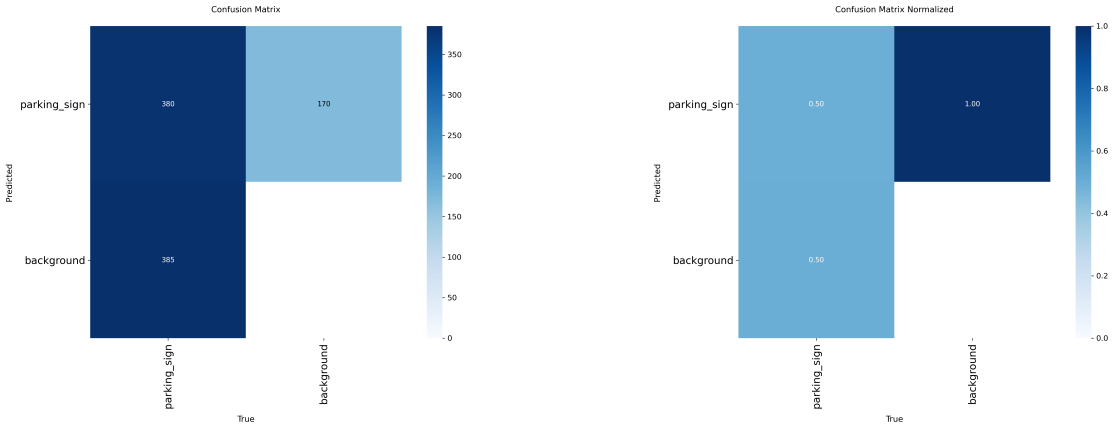
For our current task, this is acceptable because we mainly need **parking-cue presence detection**, not full OCR-based rule parsing. However, it is an important limitation to state clearly: our

present detector is suitable for cue detection, but not yet for complete parking-rule understanding.

Figure 6: Qualitative examples from manually collected images. These illustrate successful detections and also the tendency to localize the most visually salient sub-sign rather than the full sign assembly (like image on right which only detects the 'P').

6.3 Confusion Matrix Interpretation

We also inspected the validation confusion matrices produced by YOLO. They show that the detector still produces a meaningful number of false negatives, which is consistent with the moderate recall values reported earlier. This again suggests that future gains are unlikely to come only from tightening the detector threshold; instead, combining evidence across views is a more promising direction.



(a) Raw confusion matrix

(b) Normalized confusion matrix

Figure 7: Optional confusion matrices from YOLO validation.

6.4 Resolution Sensitivity and Object Scale Analysis

During qualitative evaluation, we observed a non-intuitive behavior related to input image resolution. In certain cases, reducing the input resolution (e.g., from 640 to 160) improved the detection of distant parking signs. This section analyzes this phenomenon and provides insights into the scale sensitivity of the model.

Observation. We observed that some parking signs that were not detected at higher resolutions (e.g., 640 or 960) became detectable when the input resolution was reduced (e.g., 160). An example of this behavior is shown in Figure 8, where the same image produces different detection results depending on the input resolution.

Explanation. Object detection models such as YOLO operate on fixed-size inputs and rely on hierarchical feature maps to detect objects at different scales. However, detection performance is strongly dependent on the relative size of the object within the resized image.

At higher resolutions, distant parking signs occupy only a small number of pixels relative to the full image. After multiple downsampling operations in the network, these objects become extremely small in deeper feature maps, making them difficult to detect. In contrast, when the input image is resized to a smaller resolution, the background is compressed and the object occupies a larger relative portion of the image. This effectively increases the prominence of the object in the feature maps, making detection easier.

Dataset-Induced Bias. This behavior is consistent with our dataset analysis. As shown in Figure 9, many parking signs in the training data are small and low-resolution, particularly in wide street-view images. Consequently, the model learns to detect parking signs at specific scales that are most common in the dataset.

Implications. These observations highlight that the model is not inherently learning "small images," but rather learning to detect objects at specific relative scales. Changing the input resolution shifts objects into or out of the scale range that the model is most sensitive to.

This has important implications:

Figure 8: Effect of input resolution on detection. Left: Higher resolution (imgsz=1280) input where the parking sign is missed. Right: Lower resolution (imgsz=160) input where the same sign is detected.

- Detection performance is sensitive to object scale and image resolution.
- Text-heavy signs are particularly affected due to their dependence on fine-grained visual details.
- A single fixed input resolution may not be sufficient for robust detection across all scenarios.

Practical Considerations. To address these limitations, several improvements can be considered:

- Using higher-resolution inference (e.g., 960 or 1280) to better capture small objects.
- Applying multi-scale inference to improve robustness across object sizes.
- Using tiled or patch-based inference for better small object detection.
- Incorporating OCR-based methods in future work for better handling of text-heavy parking signs.

Overall, this experiment highlights the importance of object scale in detection performance and provides valuable insight into the limitations of purely visual detection approaches for parking sign understanding.

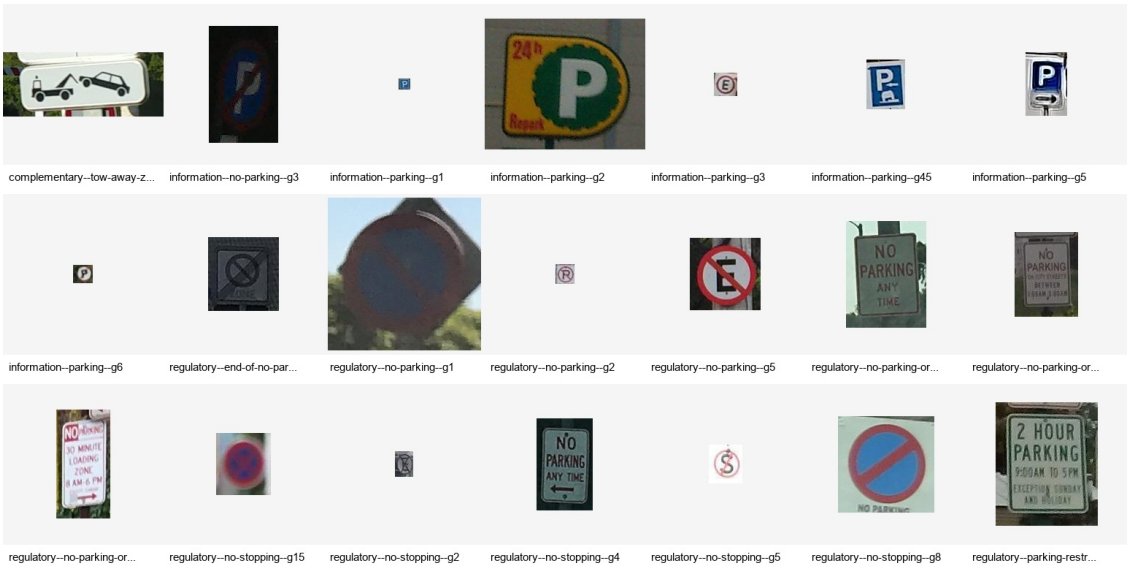


Figure 9: Sample parking sign crops from different categories in the training dataset. The large variation in size, appearance, and text content contributes to scale sensitivity and detection bias.

6.5 Qualitative Parking-Meter Findings

The zero-shot parking-meter experiment also produced useful qualitative insight. On close, clear images, the COCO-pretrained detector can correctly identify parking meters. However, the model struggles much more in realistic street-view scenes where parking meters are small relative to the full image, partially occluded, or visually similar to signposts and other narrow vertical street furniture.

This behavior matches the quantitative results in Table 4. The image-level recall shows that some transferable signal is present, but the low precision indicates that the model frequently mistakes pole-like objects for parking meters. In other words, the detector is not useless, but it is not yet reliable enough to stand on its own.

Taken together, the qualitative meter results match the quantitative findings: parking-meter detection is useful as a **partial cue**, but currently too noisy to be relied on by itself.

Figure 10: Example qualitative parking-meter detection from a manually collected image. The COCO-pretrained YOLO11x model successfully detects the parking meter in a close-view nighttime scene.

7 Difficulties Encountered

Several practical difficulties influenced our progress and also led to some changes in execution strategy.

7.1 Infrastructure and Storage Constraints

The dataset and preprocessing artifacts are relatively large. Working with tens of gigabytes of images and annotations created repeated issues across local storage, Google Colab, and remote execution environments. MTSD was 36 GBs and Vistas was 32.6 GB in size. In particular:

- Google Colab storage limits made it difficult to keep the full dataset and processed outputs in one place.
- We initially explored multiple environments, including local setup and cloud-based options.
- We tried to setup the training environment on Cloudlab but acquiring GPU instances was very difficult. We were somehow able to get a d7525 on Cloudlab Wisc with A30 GPU for 7 hours but

data transfer (36 GBs) took most of the time and then the instance had missing GPU drivers and after installing them it wouldn't boot. So we decided to move to Kaggle.

- Kaggle itself introduced interruptions: the training run crashed after epoch 24 and had to be resumed separately.

7.2 Evaluation on Local Hardware

When we first tried full-image inference for the validation split on a local Apple Silicon machine, the process was killed during prediction because the script attempted to pass the entire validation set at once. We fixed this by changing the evaluation code to use chunked prediction. This was an important engineering improvement because it made local validation feasible and reproducible.

Parking-meter evaluation introduced additional engineering difficulties. Full zero-shot evaluation with YOLO11x at high inference resolution was slow locally, and moving the evaluation to Kaggle required extra work because of GPU/runtime compatibility issues (had to move from P100 on Kaggle to T4 to resolve issue). We eventually ran the full validation experiment by restructuring the evaluation code into chunked batches on Kaggle.

7.3 Dataset and Task Difficulty

The dataset contains a large number of negatives and, more importantly, extremely small positive objects. This made it clear that the hardest part of the problem is not simply class imbalance, but small-object detection under varied viewpoint and scale. The qualitative experiments with zoom and inference resolution strongly confirmed this.

The same issue reappears in the parking-meter experiments. Many zero-shot failures were caused by the meters being too small, too far away, or visually confusable with generic poles and other narrow street furniture.

8 Changes from the Original Proposal

The overall project direction remains the same, but the order of implementation has become more focused.

8.1 What Stayed the Same

The core goal remains unchanged:

- detect parking-related cues from street-level imagery,
- aggregate these cues across nearby images,
- infer street parking presence at the segment level.

8.2 What Changed

The biggest practical change is that we decided to build a **strong parking sign baseline first** before integrating other cues. This was a useful revision because it gave us:

- a complete training and evaluation pipeline,
- a reliable first cue detector,

- a better understanding of failure modes,
- and a clean baseline against which later cue fusion can be compared.

At the proposal stage, we had planned to progress more quickly into aggregation and multi-cue integration by mid-term. In practice, more time was needed for dataset preparation, infrastructure debugging, careful evaluation design, and qualitative validation of failure modes. This was worthwhile, because the result is now a credible and well-understood baseline instead of a weaker partially tested system.

8.3 Parking Meter and Curb Cues

Parking meter detection and curb cues remain part of the planned final system, but our understanding of them has become more precise.

The preliminary zero-shot parking-meter experiment was useful because it moved this part of the project from speculation to measurement. Instead of simply assuming that parking meters would be a strong second cue, we now know that the signal is present but noisy. In its current form, parking-meter detection is best viewed as a **weak auxiliary cue**.

At the same time, the Mapillary Vistas class analysis suggests that curb-related classes are much more abundant than parking-meter instances. This makes curb modeling more attractive than we had initially appreciated. As a result, curb cues remain in scope for the second half of the project and may ultimately become the more important non-sign cue.

9 Updated Timeline and Next Steps

Based on current progress, our revised plan for the remainder of the project is:

1. **Segment-level aggregation:** use the per-image confidence outputs from the evaluation pipeline and group nearby images into segments. This is mostly a **Systems problem rather than a ML/Computer Vision problem**. We plan to test simple aggregation rules such as max pooling, mean pooling, and noisy-OR.
2. **Cue fusion:** integrate parking meter cues and, if feasible, curb-related cues.
3. **Ablation studies:** compare single-image predictions versus aggregated predictions, and compare sign-only versus multi-cue variants.
4. **Qualitative error analysis:** study where aggregation helps most, especially under small-object and viewpoint-limited scenarios.

The main goal for the next phase is to show that **aggregation improves over single-image inference**. Given the current detector's image-level F1 of 0.6673 and AUROC of 0.8310, we now have a solid baseline from which such gains can be measured.

10 Conclusion

By the mid-term stage, we have completed the most important foundational milestone of the project: a working and quantitatively validated parking sign detector. We also built a custom image-level evaluation pipeline that better matches the downstream segment-inference objective than standard detection metrics alone.

Our experiments show that the detector is useful but still limited by small-object scale and viewpoint. The model performs noticeably better when the same sign occupies more pixels, whether through zooming or higher inference resolution. This directly supports the original motivation for our project: single-image evidence is fragile, and parking presence should be inferred by aggregating evidence across views.

We also carried out a preliminary zero-shot parking-meter study. That experiment helped us refine the overall project direction: parking-meter detection has partial transfer to street-view imagery, but currently suffers from low precision and should be treated as a weak auxiliary cue rather than a standalone detector.

Overall, the mid-term results suggest that the project is on the right track. The remaining work is now more clearly defined: use this sign detector as a strong baseline, add other cues where feasible, and demonstrate that multi-view aggregation produces more robust street-segment parking inference than single-image detection alone.