

## CLOSED-CIRCUIT TELEVISION DATA AS AN EMERGENT DATA SOURCE IN URBAN RAIL PLATFORM CROWDING ESTIMATION

**Riccardo Fiorista**, Corresponding Author

Ph.D. Student, Transit Lab

Department of Urban Studies and Planning

Massachusetts Institute of Technology, Cambridge, MA 02139, USA

Email: riccardo.fiorista@mit.edu

**Awad Abdelhalim**

Assistant Director of Research, Transit Lab

Department of Urban Studies and Planning

Massachusetts Institute of Technology, Cambridge, MA 02139, USA

Email: awadt@mit.edu

**Anson Stewart**

Deputy Director, Transit Lab

Department of Urban Studies and Planning

Massachusetts Institute of Technology, Cambridge, MA 02139, USA

Email: ansons@mit.edu

**Gabriel L. Pincus**

Manager, Data Science and Engineering

Washington Metropolitan Area Transit Authority

300 7th Street SW, Washington, DC 20024, USA

Email: glpincus@wmata.com

**Ian Thistle**

Manager, Ridership Analysis

Washington Metropolitan Area Transit Authority

300 7th Street SW, Washington, DC 20024, USA

Email: ithistle@wmata.com

**Jinhua Zhao**

Professor of Cities and Transportation

Department of Urban Studies and Planning

Massachusetts Institute of Technology, Cambridge, MA 02139, USA

Email: jinhua@mit.edu

Word Count: 6494 words + 4 table(s) × 250 = 7494 words

Submission Date: August 7, 2025

**ABSTRACT**

Accurately estimating urban rail platform occupancy can enhance transit agencies' ability to make informed operational decisions, thereby improving safety, operational efficiency, and customer experience, particularly in the context of crowding. However, sensing real-time crowding remains challenging and often depends on indirect proxies such as automatic fare collection data or staff observations. Recently, Closed-Circuit Television (CCTV) footage has emerged as a promising data source with the potential to yield accurate, real-time occupancy estimates. The presented study investigates this potential by comparing three state-of-the-art computer vision approaches for extracting crowd-related features from platform CCTV imagery: (a) object detection and counting using YOLOv11, RT-DETRv2, and APGCC; (b) crowd-level classification via a custom-trained Vision Transformer, Crowd-ViT; and (c) semantic segmentation using DeepLabV3. Additionally, we present a novel, highly efficient linear-optimization-based approach to extract counts from the generated segmentation maps while accounting for image object depth and, thus, for passenger dispersion along a platform. Tested on a privacy-preserving dataset created in collaboration with the Washington Metropolitan Area Transit Authority (WMATA) that encompasses more than 600 hours of video material, our results demonstrate that computer vision approaches can provide substantive value for crowd estimation. This work demonstrates that CCTV image data, independent of other data sources available to a transit agency, can enable more precise real-time crowding estimation and, eventually, timely operational responses for platform crowding mitigation.

*Keywords:* Computer Vision, Urban Rail, Crowding

## INTRODUCTION & RELATED WORK

Accurately estimating rail ridership during revenue service is essential for safe, efficient, and customer-focused urban rail operations (1, 2). While most prior work centers on in-vehicle crowding (3), platform-level crowding requires separate attention due to its implications for passenger safety, dwell time, and station-level congestion. Existing methods, including dynamic factor models (4), regression trees (5), and graph neural networks (6), often focus on short-term prediction by reconstructing occupancy from historical data, Automatic Fare Collection (AFC) records, and train movement data. Although these sources are typically available in real-time, true platform occupancies are not, limiting the effectiveness, particularly of auto-regressive time-series models. In this context, Closed-Circuit Television (CCTV) footage presents a promising source for obtaining accurate occupancy estimates, enhancing operational awareness and short-term prediction.

Computer Vision (CV) is needed to extract crowd information from such CCTV imagery. Recent CV advancements offer substantial improvements in automated object detection and scene understanding, achieving unparalleled accuracy and inference speed (7, 8). Notably, architectures such as *You Only Look Once (YOLO)* (9) and *DEtection TRansformer (DETR)* (10) have successfully been employed for people counting in public transit (11, 12). Domain-specific methods, including Auxiliary Point Guidance Crowd Counting (APGCC) (13, 14) and density map estimation (15), further improved crowd counting accuracy in densely populated scenes where traditional detection methods face limitations due to occlusions and image artifacts. Techniques such as the model-agnostic Slicing Aided Hyper Inference (SAHI) (16), which evaluate smaller image patches, have been developed to further address these challenges.

Beyond detection, recent research has demonstrated the effectiveness of image classification and semantic segmentation in various contexts. Vision Transformers (ViT) (17) have shown excellent classification performance in transportation-related tasks, such as predicting bus delays from traffic camera imagery (18). Additionally, semantic segmentation approaches such as DeepLabV3 (19) and U-Net (20) provide pixel-level scene understanding, further enriching the set of available CV tools to identify image regions containing certain objects.

Despite these advancements, the potential of CV-derived features for real-time platform crowding estimation remains underexplored. Existing research has primarily focused on in-station passenger flow (21) or vehicle counts (22), with limited investigation into continuous, real-time CCTV-based occupancy estimation. A notable exception is Wang et al.’s study (23), which applied an enhanced YOLOv8 model to a single day’s CCTV footage at a Beijing metro station, yielding promising yet noisy initial results.

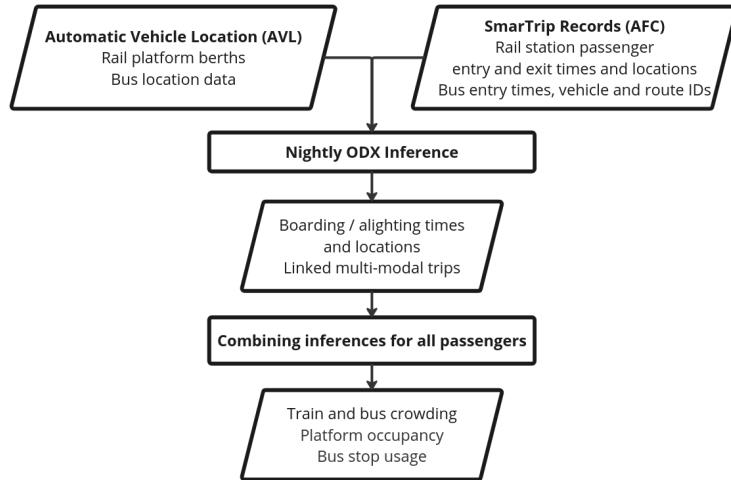
This paper comprehensively evaluates how CCTV-derived features can estimate rail platform occupancy. We use a privacy-preserving, self-collected, labeled CCTV dataset covering multiple platforms and revenue service days within the Washington Metropolitan Area Transit Authority (WMATA) Metrorail network. Specifically, we investigate three CV approaches:

1. **Crowd counting**, employing YOLOv11 (24) and RT-DETRv2 (25) object detection, as well as APGCC (13) head counting, enhanced with Slicing-Aided Hyper Inference (SAHI);
2. **Crowd-level classification** using a custom fine-tuned Vision Transformer model, which we refer to as *Crowd-ViT* (17);
3. **Segmentation-based crowd estimation**, applying the DeepLabV3 model (26) with a novel Mixed-Integer Linear Programming (MILP)-based calibration approach.

We evaluate the performance of these image-based approaches against ground truth occupancy estimates derived from WMATA’s internal implementation of the Origin–Destination Interchange (ODX) algorithm (27). Specifically, we assess model accuracy at three aggregation levels: per-image, per-train-arrival event, and as 15-minute bin mean aggregate.

## METHODOLOGY

This section introduces the ODX-derived ground-truth platform occupancy estimates used for image labeling, followed by a description of the generated CCTV dataset and the CV methods used to estimate platform crowding: object detection, classification, and segmentation. We then present a novel MILP-based calibration method that enables direct occupancy estimation from segmentation maps and conclude with an overview of the metrics used for performance assessment.

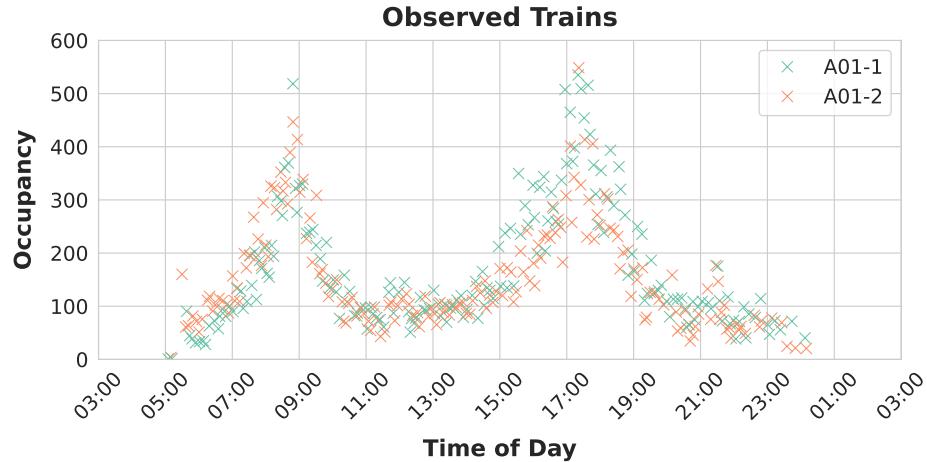


**FIGURE 1:** High-level Origin Destination Interchange (ODX) algorithm diagram. Smartcard data from WMATA’s closed-loop Automatic Fare Collection (AFC) system is combined with Automatic Vehicle Location (AVL) data. After the end of a revenue service day, ODX reconstructs each rider’s journey, additionally providing probability-based estimates of rail platform and bus berth occupancies, vehicle loads, and travel times.

## Ground Truth Occupancy Data

The ground truth rail platform occupancy values in this study are derived from WMATA’s ODX algorithm (27), which combines smart card tap data, Automatic Vehicle Location (AVL), and operational context to infer likely rider paths and platform occupancies. A schematic overview of this process is provided in Figure 1. Supporting both tap-in and tap-in/out systems, ODX assigns probabilities to possible trajectories and can thus also provide estimated platform occupancy. While WMATA uses the tap-in/out fare collection system, tapping during transfers is not required, which increases assignment uncertainty at transfer stations.

The occupancy estimates that ODX generates are continuous, i.e.  $y \in \mathbb{R}^+$ , for each train-arrival event, by averaging over plausible rider trajectories as illustrated by the sample shown in Figure 2. These values correspond to the moment of peak platform crowding, internally referred to as the *magic moment*, when a train has arrived, all passengers have alighted, and none have



**FIGURE 2:** Sample of observed trains over a revenue service day on Metro Center platforms A01-1/2 with ODX-estimated platform occupancies attributed to the individual train-arrival events on the y-axis.

yet boarded. Since ODX retroactively computes these estimates overnight, neither real-time occupancy data nor individual rider trajectories are available during revenue service.

We use these per-train-arrival event values to label CCTV images and compute 15-minute mean occupancies. For the CCTV-based occupancy estimation, we do not leverage any further exogenous variables other than the image data from various cameras located on a specific rail platform.

### CCTV Data

This section details our video collection and image sampling approach. Furthermore, we illustrate how continuously-valued ODX estimates are attributed to images for the counting-based approaches (object detection, head counting, and segmentation), and how crowd-level class labels are established, which will be used for the Crowd-ViT classification approach.

#### *Data Acquisition and Sampling*

Platform	#Cameras	#Images	#Train-Arrival Events	#All Captured	#EMPTY	#LOW	#MEDIUM	#HIGH
A01-1	2	4,310	567	408	103	136	288	40
A01-2	3	4,041	517	0	113	135	234	35
A03-1	3	1,036	80	80	16	25	36	3
A03-2	1	352	80	80	18	26	33	3
B01-2	1	88	20	20	5	2	6	7
C01-C	6	33,253	1,284	1,176	338	229	596	122
F05-C	2	152	17	17	0	0	1	16
<b>Totals</b>	<b>18</b>	<b>43,232</b>	<b>2,565</b>	<b>1,781</b>	<b>593</b>	<b>553</b>	<b>1,194</b>	<b>226</b>

**TABLE 1:** Summary of available CCTV data for Metro Center (A01/C01), DuPont Circle (A03), Gallery Place–Chinatown (B01), and Navy Yard–Ballpark (F05). Side platforms are indicated by *I/2*; center platforms by *C*.

The video dataset was collected in collaboration with the WMATA Data Lab. WMATA’s rail network comprises 123 platforms across 97 stations, monitored by over 1,000 cameras at fare gates, platforms, and rail track segments. This study focuses on seven platforms across four central Washington D.C. stations, namely, Metro Center, DuPont Circle, Gallery Place–Chinatown, and Navy Yard, using footage from 18 platform-mounted cameras. Videos were sampled in a privacy-preserving manner between June 3<sup>rd</sup>, 2024, and January 20<sup>th</sup>, 2025, during weekday peak hours (07:00–09:00 and 16:00–19:00) and high-demand events such as July 4<sup>th</sup> and major sporting events. The selected stations and corresponding platform-camera configurations are:

- *Metro Center*: A01-1/2 (Red Line, side platforms); C01-C (Blue/Orange/Silver, center platform).
- *Gallery Place–Chinatown*: B01-1/2 (Red Line, side platforms); F01-C (Green/Yellow, center platform).
- *Dupont Circle*: A03-1/2 (Red Line, side platforms).
- *Navy Yard–Ballpark*: F05-C (Green Line, center platform).

The sampled video streams were then processed to extract three frames per train-arrival event: 5 seconds before, at, and 5 seconds after the magic moment. The ODX occupancy estimates collected for each train-arrival event serve as a proxy ground truth. This approach mitigates timing misalignment and increases visual diversity, improving dataset robustness. Figure 3 shows an example sequence in high crowding, and Table 1 summarizes the dataset. We note that the resulting dataset is imbalanced due to variation in sampling periods. With the most samples, Metro Center receives particular emphasis in our analysis.



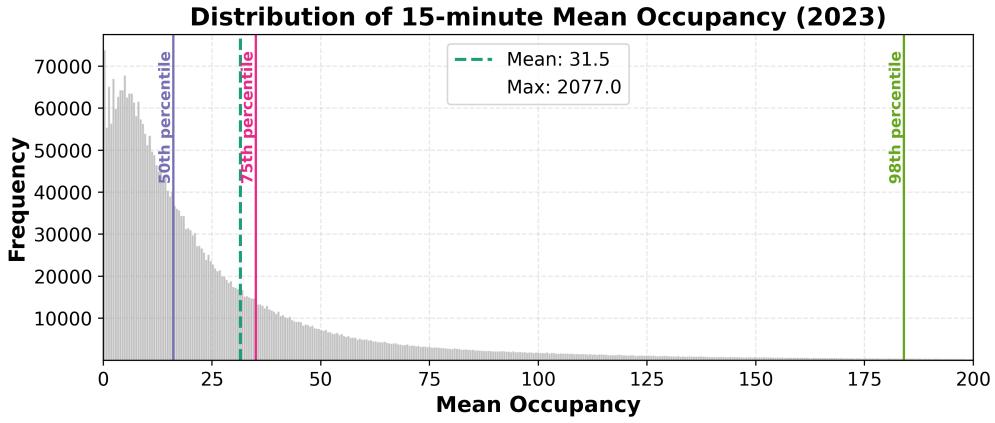
**FIGURE 3:** The three sample frames captured during a train-arrival event from a camera at A01-2 showing the progression around the magic moment (-5s, 0s, +5s) of a scene with crowd level *high*. Note, how +5s features a visually more crowded scene, highlighting the need for temporal variation in CCTV sampling.

#### *Image Pre-Processing*

Despite suboptimal lighting and image quality, the fixed field of view, angle, and consistent camera conditions enable effective pre-processing. We compute and apply per-camera color normalization values across all images for all CV models. The static views further allow for masking specific *areas of interest*, as illustrated in Figure 4, which we use in post-processing to prevent cross-platform inference when multiple platforms appear in a single frame.



**FIGURE 4:** Sample footage of the Metro Center Red line (A01-1/2) platforms with an overlaid area-of-interest mask indicating the region to which CV-based results are limited.



**FIGURE 5:** 15-minute mean occupancy values across the WMATA system for all platforms computed over 2023 with 50<sup>th</sup> (mean), 75<sup>th</sup>, and 98<sup>th</sup> percentiles.

#### Crowd Level Label Generation

As detailed above, we use ODX’s continuous per-train-arrival event occupancy estimates as target labels. For classification, however, discrete labels are required. Without normative thresholds for crowd levels, we define *crowding class* boundaries based on *per-platform* occupancy distributions observed during 2023 weekday service days, excluding zero-valued occupancies. Specifically, we use the 50<sup>th</sup>, 75<sup>th</sup>, and 98<sup>th</sup> percentiles to delineate four classes for each platform. This scheme empirically captures typical peak-hour volumes in the *medium* class and reserves the *high* class for exceptional crowding. Based on these thresholds, we define the following four crowd levels:

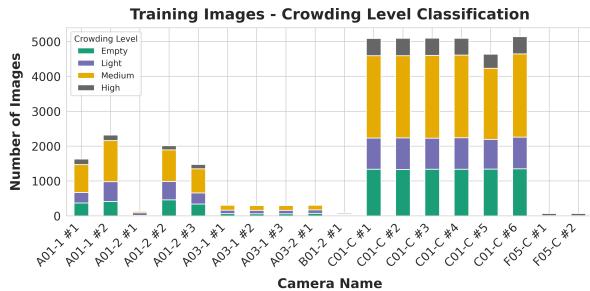
- **Empty:**  $C_{\text{empty}} \in [0^{\text{th}}, 50^{\text{th}}]$
- **Low:**  $C_{\text{low}} \in (50^{\text{th}}, 75^{\text{th}}]$
- **Medium:**  $C_{\text{medium}} \in (75^{\text{th}}, 98^{\text{th}}]$
- **High:**  $C_{\text{high}} \in (98^{\text{th}}, 100^{\text{th}}]$

### *crowd-level classification Data Preparation*

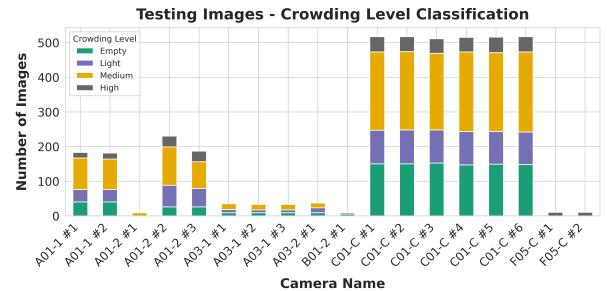
Across all platforms, distributions are heavily skewed toward lower occupancy values, even after excluding zeros. A long right-hand tail is evident, with large gaps between the median and the 98<sup>th</sup> percentile, highlighting the rarity and extremity of high crowding events. These patterns are consistent across the WMATA system, as shown in Figure 5.

This distributional skew is also reflected in the generated dataset, as illustrated in Table 1, displaying substantial class imbalance. This imbalance can bias model training by overfitting to over-represented classes and under-representing rare cases such as *high crowding*. Thus, we apply stratified sampling and data augmentation techniques to partially address this shortcoming. Stratified sampling ensures that each crowding class is proportionally represented in training and testing splits, preserving class distribution. Data augmentation involves generating additional samples through image transformations to increase dataset diversity and reduce underrepresentation of crowd levels such as *high*.

To train our custom Crowd-ViT models, we apply a 90%/10% training/testing split stratified by crowd level (empty/light/medium/high). Where possible, we restrict the testing set to train-arrival events jointly captured by all cameras on a given platform, thus avoiding leakage of visually correlated images. This results in 1,599 jointly captured train-arrival events (32,498 images) for training and 179 (3,624 images) for testing. An additional 6,684 training and 426 testing images come from train-arrival events captured by only one camera. For A01-2, where no jointly captured train-arrival events were available, images were randomly split while maintaining class stratification. Figure 6 shows the resulting class and camera distributions.



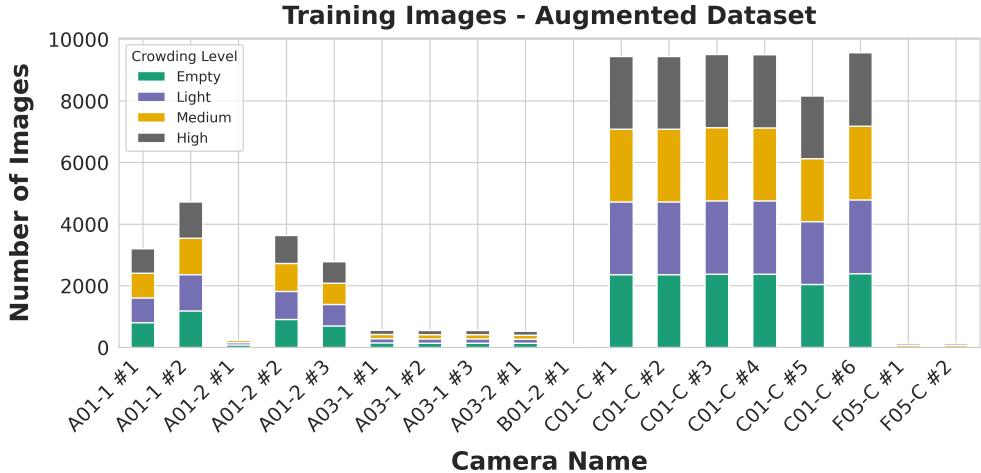
(a) Per-camera training dataset distribution.



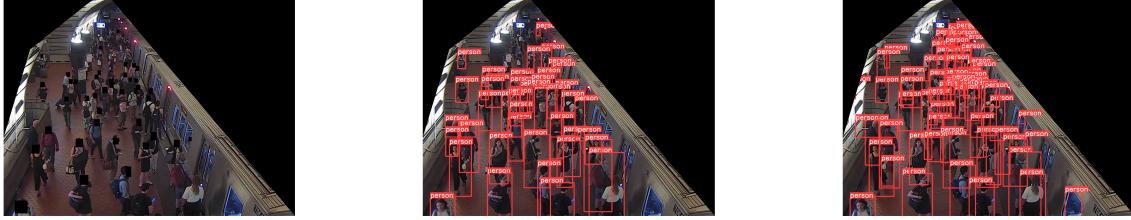
(b) Per-camera testing dataset distribution.

**FIGURE 6:** Distributions of sampled per-crowd level images across cameras. We apply a 90%/10% training-testing split with stratified sampling based on crowd level labels to allow for a similar distribution between the two sets. Images of individual train-arrival events were kept in the same set when possible.

Furthermore, we apply data augmentation to the training dataset. For the *empty* class, where variation between images is minimal, naïve image repetition suffices. For the remaining classes, we apply targeted data augmentation using random brightness and contrast shifts, color jitter, rotations up to 15°, Gaussian noise, and motion blur, each applied with 30% probability. This results in a balanced training set across all crowd levels, as illustrated in Figure 7.



**FIGURE 7:** Crowd-ViT classification training dataset balanced through augmentation.



(a) Original CCTV footage from Metro Center platform A01-1. (b) YOLOv11 with SAHI, limited to COCO class 0 (person). (c) RT-DETRv2 with SAHI, limited to COCO class 0 (person).

**FIGURE 8:** Detection results on Metro Center platform A01-1 from YOLOv11 and RT-DETRv2 (with SAHI), compared to the original masked CCTV frame. Detections use a 30% confidence threshold, and area-of-interest masking is applied.

## Crowd Counting

### Object Detection

To count people on CCTV images, we apply two state-of-the-art object detection models: YOLOv11 (model *YOLO11x*, 56.9M parameters)(28) and RT-DETRv2 (model *r101vd*, 76M parameters)(29), both pre-trained on the Common Objects in Context (COCO) 2017 dataset (30). Due to variability in image quality, ranging from 500×500 to 1920×1080 pixels, alongside poor lighting, streaming artifacts, and image distortions, we enhance detection accuracy using SAHI. SAHI processes fixed-size patches of each image independently, thus improving a models’ capability to detect objects in noisy, occluded, and spatially complex images. We illustrate the effectiveness of SAHI when combined with RT-DETRv2 in Figure 9.

We impose a confidence threshold  $\geq 30\%$  for both models to minimize false positive detections. This threshold was assessed empirically using randomly sampled images from the training dataset. Occupancy at the platform level is then estimated by counting the detection bounding boxes for COCO class 0 (*person*). To reduce temporal noise, we record the maximum person



(a) RT-DETRv2 estimates without SAHI, showing detections of COCO class 0 (person).

(b) RT-DETRv2 with SAHI, applied to the same scene, for COCO class 0 (person).

**FIGURE 9:** Comparison of RT-DETRv2 detection results on Metro Center platform A01-1 without and with SAHI. Detections are considered when above a 30% confidence threshold, and area-of-interest masking is applied. Note how, on the right (RT-DETRv2 *with* SAHI), riders in more distant and distorted parts of the image are successfully detected.

count across the three frames captured (-5s, 0s, +5s) around the magic moment of each train-arrival event. We discuss both results on a per-image, per-train-arrival event, and per 15-minute bin basis. We compute the maximum occupancy count across the three images for the per-train-arrival event approach. For the 15-minute bin, instead, we consider the mean of all maximum values sampled in the respective bin.

#### *Head Counting*



**FIGURE 10:** APGCC model head counting results with associated confidence for C01-C #4.

In addition to object detection, we leverage one of the most salient features in overhead views of crowded scenes: the human head. Specifically, we employ the state-of-the-art APGCC model, pre-trained on the ShanghaiTech A datasets (31). APGCC uses an auxiliary point guidance mechanism that filters candidate head locations before final localization (13). In Figure 10, we show that the APGCC model outputs x/y-coordinates of detected heads for each image (indicated by the "x" markers). Equivalently to the object detection models, we record APGCC's maximum

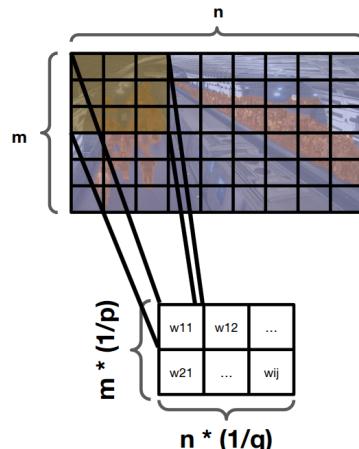
estimate across the three frames for each train-arrival event, and for the 15-minute bin, the mean of these maximum values across the respective bin.

### crowd-level classification

Rather than exact counts, the *magnitude* of platform occupancy often offers a sufficiently informative and potentially more robust signal for estimation and prediction. We explore this type of encoding by developing a crowd-level classification framework using CCTV imagery. Without pre-trained models tailored to this task, we fine-tune a Google Vision Transformer (ViT) model *for each camera*, which captures spatial and contextual features (32). We adapt the final classification layer to output four logits aligned with the aforementioned platform-specific crowd levels.

For evaluation, estimates are generated on the aforementioned, held-out testing dataset. We consider the median estimated crowd level across the three sampled frames for each train-arrival event and, for the 15-minute bin, the *median* estimated crowd level across all images captured in the respective bin.

### Crowd Segmentation



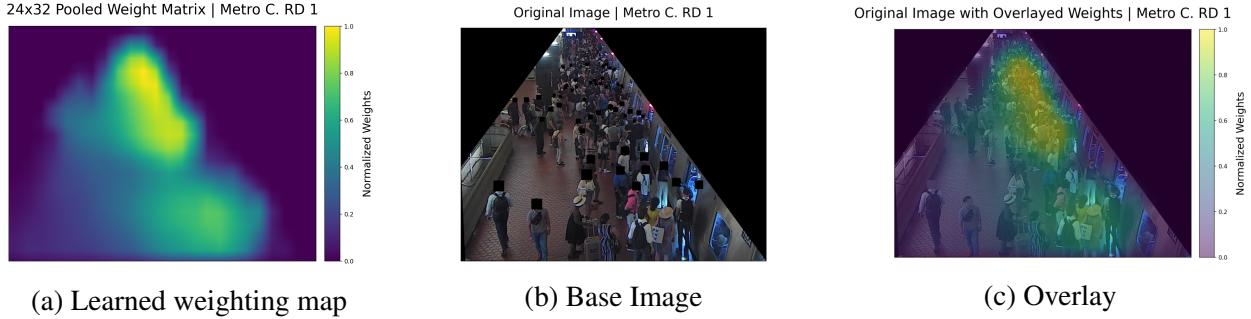
**FIGURE 11:** Visualization of segmentation map, maximum pooling, and multiplication with weight matrix. Underlying is the DeepLabV3-generated binary segmentation map (person-class/background pixels) of size  $m \times n$ . The learned weight matrix is then applied to each pooled region.

Finally, we apply semantic image segmentation to estimate the image area riders occupy in a given CCTV frame, serving as a proxy for occupancy estimation. We employ the DeepLabV3 model (model *ResNet101*, 58.6 million parameters)(19, 33) pre-trained on the PASCAL VOC 2012 dataset (34). The model produces segmentation maps identical in size to the input images, assigning binary values *one* to pixels classified as *class 15 (person)* in the PASCAL VOC taxonomy and *zero* otherwise.

From these maps, we extract three representations. First is the raw pixel count of detected person-class regions. Second, the ratio of human-class pixels to total image area. Third, an estimated platform occupancy count, obtained by the product of the segmentation maps with a learned, camera-specific calibration weight map. For the latter, we formulate a novel MILP ridge regression model producing pixel-weighting matrices, enabling efficient and interpretable mappings from

segmentation outputs to crowd estimates.

### Segmentation Map Weighting



**FIGURE 12:** Sample of how a learned segmentation map calibration relates to the main crowding areas in a CCTV image. This calibration map is used to compute platform occupancy estimates from segmentation maps provided by a DeepLabV3 semantic segmentation model.

Our dataset benefits from fixed camera viewpoints and consistent platform view angles. The underlying passenger behavior, entering through designated access points and queuing near train doors, generates a stable visual structure. Since all images are sampled during train-arrival events, the spatial distribution of queuing serves as a reliable proxy for occupancy levels.

Object detection and classification models yield directly interpretable outputs such as counts or labels, but segmentation methods produce dense, pixel-level maps requiring post-hoc calibration. To bridge the gap between these pixel activations and true platform occupancies, we introduce a novel, MILP-based ridge regression formulation, given as:

$$\begin{aligned}
 & \min_{\{\mathbf{w}_c\}_{c \in C}} \quad \sum_{c \in C} \sum_{a \in A^c} \sum_{i \in I_c^a} \left( y_a^i - \sum_{j=1}^{\frac{m}{p}} \sum_{k=1}^{\frac{n}{q}} \mathbf{P}_{a,c}^i[j,k] \mathbf{w}_c[j,k] \right)^2 + \lambda \sum_{c \in C} \sum_{j=1}^{\frac{m}{p}} \sum_{k=1}^{\frac{n}{q}} \mathbf{w}_c[j,k]^2, \\
 & \text{s.t.: } \mathbf{P}_{a,c}^i[j,k] = \frac{1}{pq} \sum_{u=0}^{p-1} \sum_{v=0}^{q-1} \mathbf{X}_i^c[jp+u, kq+v], \quad \forall c \in C, a \in A^c, i \in I_c^a, j, k, \\
 & \quad \mathbf{P}_{a,c}^i \in \mathbb{R}^{\frac{m}{p} \times \frac{n}{q}}, \quad \mathbf{w}_c \in \mathbb{R}^{\frac{m}{p} \times \frac{n}{q}}, \\
 & \quad p, q \text{ divide } m, n, \quad \lambda > 0, \quad y \in \mathbb{N}^{|I|}, \quad X_i^c \in \{0, 1\}^{m \times n}, \\
 & \quad c \in C \text{ (cameras), } \quad a \in A^c \text{ (train-arrival events captured by } c), \\
 & \quad i \in I_c^a \text{ (images of event } a \text{ captured by } c).
 \end{aligned}$$

Here, for each camera  $c \in C$ , we index train-arrival events by  $a \in A^c$  and the corresponding images by  $i \in I_c^a$ . The term  $y_a^i$  denotes the ground-truth ODX-based platform occupancy estimate for the  $i$ -th image of event  $a$ , as captured by camera  $c$ . The binary segmentation map  $\mathbf{X}_i^c \in \{0, 1\}^{m \times n}$  is produced by the DeepLabV3 model to indicate regions of class *person* in the original image. In contrast,  $\mathbf{P}_{a,c}^i \in \mathbb{R}^{\frac{m}{p} \times \frac{n}{q}}$  is obtained by applying a **maximum**-pooling operation over non-overlapping  $p \times q$  blocks of  $\mathbf{X}_i^c$ , illustrated in Figure 11, thereby reducing its size. The

camera-specific weight matrix  $\mathbf{w}_c \in \mathbb{R}^{\frac{m}{p} \times \frac{n}{q}}$  shares the same shape as this pooled segmentation map  $\mathbf{P}_{a,c}^i$ . The objective in the formula above is then to minimize the squared residuals between the ground-truth occupancy  $y_a^i$  and the weighted sum:

$$\sum_{j=1}^{\frac{m}{p}} \sum_{k=1}^{\frac{n}{q}} \mathbf{P}_{a,c}^i[j,k] \mathbf{w}_c[j,k], \quad (1)$$

with an additional L2 regularization term  $\lambda \sum_{c \in C} \sum_{j,k} \mathbf{w}_c[j,k]^2$  to discourage large weights. We solve this convex program using the ECOS solver (35) via the Python CVXPY interface (36).

Once the optimization yields each camera’s weight matrix  $\mathbf{w}_c$ , we compute a real-valued platform occupancy estimate by element-wise multiplying the learned weight matrix with the pooled DeepLabV3 segmentation maps  $\mathbf{P}_{a,c}^i$  in the testing dataset. Visually,  $\mathbf{w}_c$  assigns higher weights to regions corresponding to areas closer to the train and farther from camera  $c$ , as demonstrated in Figure 12. Furthermore, Figure 13 illustrates how different choices of pooling granularity ( $p, q$ ) affect the resolution of  $\mathbf{w}_c$ , preserving the overall spatial weighting structure while varying detail.



**FIGURE 13:** Learned weight maps with increasing resolution as pixel-pooling size is reduced (from left to right). Smaller pooling sizes reveal more platform features, with higher weights near train entrances and farther from the camera.

### Gradient Boosting Decision Trees: LGBM

CV-derived counts often suffer from differences in scale. Thus, to map CV-derived features to platform occupancy estimates, we use Light Gradient Boosted Machines (LGBM) (37), a non-linear ensemble method that builds decision trees sequentially, each correcting residuals from the previous iteration. LGBM’s histogram-based training, leaf-wise growth, and native support for regularization, missing values, and parallelization make it well-suited for high-dimensional input data (38).

We train a separate Light Gradient Boosted Machine (LGBM) model for each CV method to predict platform occupancy at the per-train-arrival event level. For each arrival, we collect three images per available camera, apply the respective CV method, and aggregate the resulting estimates per camera using either the mean (for detection, counting, and segmentation) or the median (for classification). These per-camera summary statistics are then used as input features to the LGBM model, which produces a single platform-level occupancy estimate for the corresponding train-arrival event. LGBM’s robustness to missing data allows the inclusion of events with incomplete camera coverage. Finally, we average the resulting LGBM estimates across all train arrivals within each 15-minute bin to assess each CV method’s suitability for aggregate-level crowding estimation.

## Performance Evaluation

We aim to quantify the effectiveness of various CV approaches for estimating platform occupancy at different granularities: per-image, per-train-arrival event, and per-15-minute bin. To formalize this tasks, we assume a model  $f$  which maps a feature vector  $x$ , derived from CCTV imagery, to an estimated occupancy  $\hat{y}$ , i.e.  $\hat{y} = f(x)$ . To evaluate the performance of  $f$ , we use several metrics. For the occupancy estimation task, we distinguish:

- Per-train-arrival event estimates  $\hat{y}_a$  for each train-arrival event  $a \in A$
- Per-image estimates  $\hat{y}_a^i$  for each image  $i \in I_a$  associated with train  $a \in A^c$  captured by camera  $c \in C$ .
- Per-interval estimates  $\hat{y}_t$  for each 15-minute bin  $t \in T$ .

Since ODX provides per train-arrival event occupancy estimates  $y_a$ , these serve as targets for  $\hat{y}_a$  and  $\hat{y}_a^i$ . For aggregated estimates  $\hat{y}_t$ , the target is the 15-minute mean occupancy:

$$\bar{y}_t = \frac{1}{|A_t^c|} \sum_{a \in A_t^c} y_a \quad \text{where } A_t^c = \{a \in A^c \mid \text{train } a \text{ arrived during time bin } t\}. \quad (2)$$

First, we report the coefficient of determination ( $R^2$ ) to assess model fit. To account for differences in scale between vision-based estimates and true occupancy values, we also report a normalized  $\tilde{R}^2$ , defined by first standardizing estimates and ground truth using a z-score transformation and then computing:

$$\tilde{R}^2 = R^2(z(y), z(\hat{y})), \quad (3)$$

Where the z-transformation is computed using:

$$z(x) = \frac{x - \mu_x}{\sigma_x}, \quad (4)$$

with  $\mu_x$  and  $\sigma_x$  denoting the mean and standard deviation of variable  $x$ , i.e. our CV-based estimates, respectively.

Second, to evaluate the error between the ground truth  $y$  and our estimates  $\hat{y}$ , we employ the 95<sup>th</sup> percentile Absolute Error (95<sup>th</sup> AE). Here, 95<sup>th</sup> AE captures the highest values in terms of absolute deviation between the target platform occupancy and the model's estimation. The formulation, where  $\eta$  denotes the percentile, can be written as:

$$\text{AE 95}^{\text{th}} = \eta_{.95}(|y - \hat{y}|) \quad (5)$$

Third, we define a custom weighted Mean Absolute Error (wMAE), which more severely penalizes estimation errors during periods of atypical platform occupancy by incorporating temporal context through a z-score-based weight measure.

Let  $t \in \{1, \dots, T\}$  index each 15-minute bin in the testing set, and let  $y_t$  and  $\hat{y}_t$  denote the observed and estimated occupancy values at time  $t$ , respectively. Each time-step  $t$  is assigned to one of  $B = 4 \times 24 \times 7 = 672$  15-minute bins in the weekly cycle, indexed by  $b(t) \in \{1, \dots, B\}$ .

Let  $\mu_b$  and  $\sigma_b$  denote the historical mean and standard deviation of occupancy for bin  $b$ , computed across all weeks in 2023. For each  $t$ , we define a z-score-based weight as follows, where the term inside the norm represents the traditional z-score computation:

$$w_t = 1 + \left| \frac{y_t - \mu_{b(t)}}{\sigma_{b(t)}} \right|. \quad (6)$$

The weighted Mean Absolute Error is then defined as:

$$\text{wMAE} = \frac{1}{\sum_{t=1}^T w_t} \sum_{t=1}^T w_t \cdot |y_t - \hat{y}_t|. \quad (7)$$

This formulation increases the penalty for estimation errors during high-deviation periods such as unusual morning or afternoon peaks, operational disruptions, or special events. The presented formulation for  $w_t$  ensures that  $w_t \geq 1 \forall t \in T$ , preserving numerical stability.

## EXPERIMENTS & RESULTS

This section evaluates CV models for estimating ground-truth platform occupancy from CCTV imagery at per-image, per-train-arrival, and per-15-minute bin levels. We first present individual model performance, followed by an analysis of LGBM-based aggregation across cameras per train-arrival event. Lastly, we assess 15-minute bin accuracy using LGBM outputs. A special focus is laid on Metro Center platforms A01-1/2 and C01-C due to data availability. Performance differences between side and center platforms are also examined.

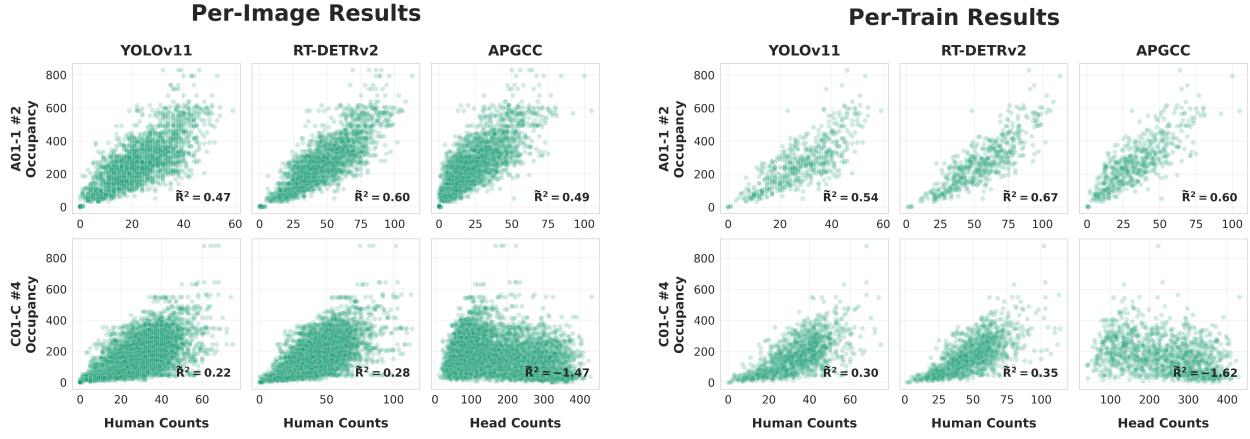
All experiments were run on an MIT SuperCloud instance with one Nvidia V100 GPU, 20 Intel Xeon-G6 cores, and 170GB RAM.

### Object Detection

Applied to the full CCTV dataset, we evaluate three, pre-trained, detection-based models: YOLOv11, RT-DETRv2, and the head-counting model APGCC. Figure 14 shows results from one representative camera on a side and one on a center platform.

Per-image estimates in Figure 14a show positive correlation with ODX-derived ground truth across all models. Side platforms yield higher  $\tilde{R}^2$  scores, likely due to elevated camera angles and reduced directional passenger mixing. RT-DETRv2 with SAHI performs best, capturing a broad range of crowd levels. APGCC performs reasonably on side platforms but fails on center platforms due to systematic false positives, where floor tiles and tracks are often misclassified as heads, resulting in negative  $\tilde{R}^2$  values (see Figure 10). All models undercount in high-density scenes, with maximum observed counts of 96 (YOLOv11 + SAHI), 199 (RT-DETRv2 + SAHI), and 455 (APGCC).

Aggregating detections per train-arrival event improves estimation accuracy as shown in Figure 14b. This temporal aggregation also reduces noise, narrowing estimation dispersion. RT-DETRv2 remains most accurate, while YOLOv11 continues to greatly undercount, with an overestimation following a pseudo-exponential trend at mid-range occupancies. APGCC overestimates in lower occupancy conditions on side platforms and fails on center platforms, resulting in negative  $\tilde{R}^2$  values.



(a) *Per-image*: Each data point corresponds to a single CCTV frame captured. This produces a higher-resolution but noisy mapping between object detection counts and true occupancy.

(b) *Per-train-arrival event*: For each train-arrival event, the maximum detection count across frames is used. This approach aggregates the signal from Figure 14a and produces a cleaner relationship between observed and true occupancies.

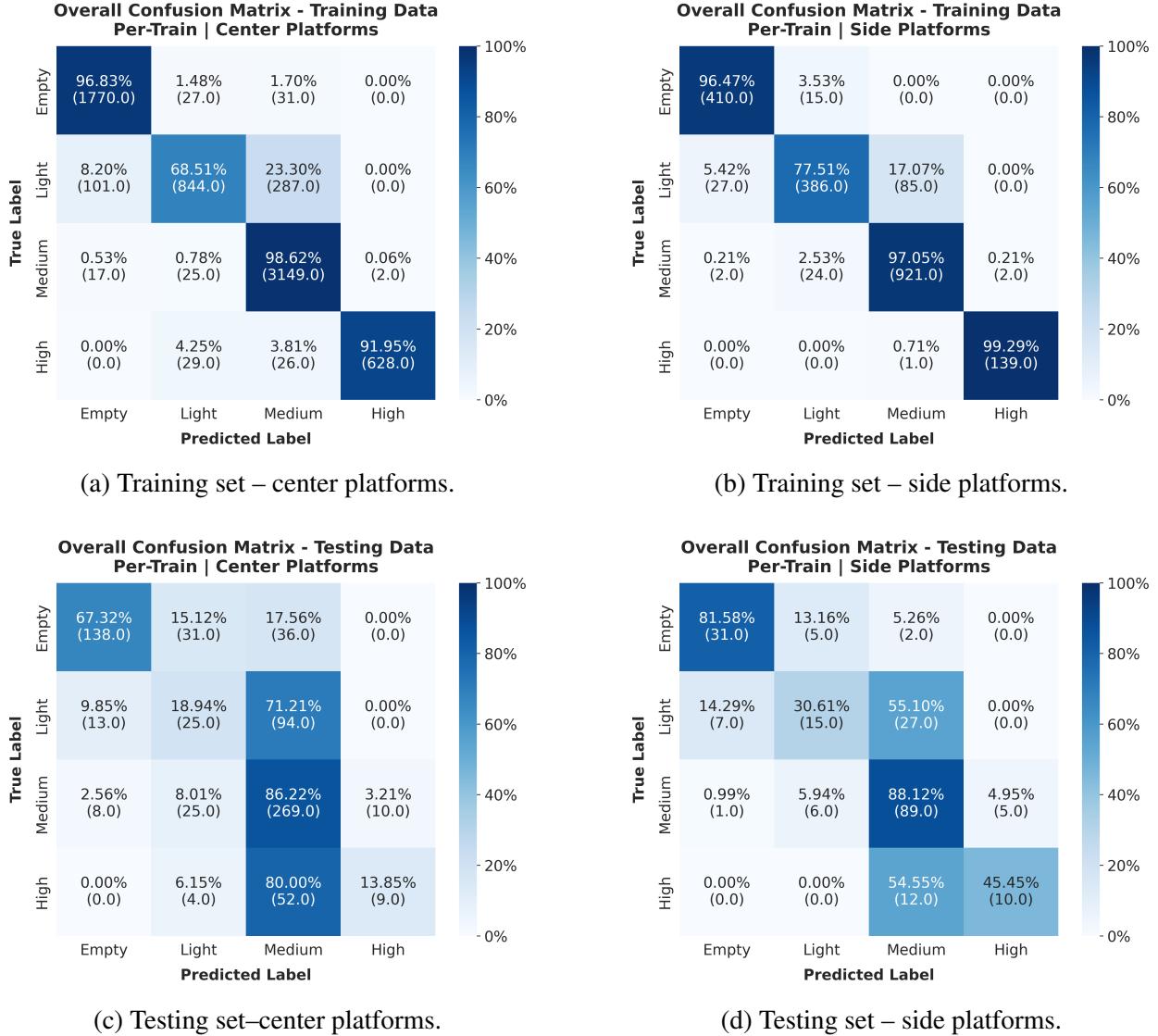
**FIGURE 14:** Object-detection (YOLOv11/RT-DETRv2) and head counting (APGCC) results plotted against true platform occupancy estimates from ODX. In both figures, *top*: Metro Center Red line platform A01-2 (**side platform**); *bottom*: Gallery Place Chinatown Yellow/Green line C01-1 (*center platform*).

### crowd-level classification

Fine-tuning per-camera Crowd-ViT models on our augmented, class-balanced training dataset yields strong training performance but limited generalization. On the training set, per-image accuracy reaches 89.15%, with high F1 scores across all classes, including *empty* (0.93), *medium* (0.92), and *high* (0.94). Aggregating estimates to the per-train-arrival event level further improves accuracy to 92.19%, with F1 scores exceeding 0.94 for all but the *light* class (0.73).

In contrast, performance on the testing dataset drops substantially, with 60.96% per-image accuracy and low F1 scores for the under-represented *light* (0.24) and *high* (0.32) classes. The model performs best on the more frequent *empty* (0.74) and *medium* (0.70) classes. Per-train-arrival aggregation improves test accuracy to 63.42%, with slight gains in F1 for *empty* (0.77) and *medium* (0.72). Overall, accuracy is significantly higher on side platforms (69.05%) than on center platforms (61.76%) across all studied platforms.

The performance gap reflects overfitting, likely due to the ViT’s 86 million parameters and limited training data. Still, confusion matrices (Figure 15) show that misclassifications largely occur between adjacent classes, suggesting the model captures ordinal structure. No extreme misclassifications (e.g., *empty* as *high*) were observed. Here, limitations arising from our relatively limited image dataset (relative to what is typically needed for training a vision transformer model), noise from our automatic label-generating process, and the crowd level bands chosen based on occupancy percentiles are all areas for improvement to reduce the observed misclassifications. Despite modest accuracy, the model still provides useful categorical signals, allowing to clearly distinguish empty from crowded scenes.

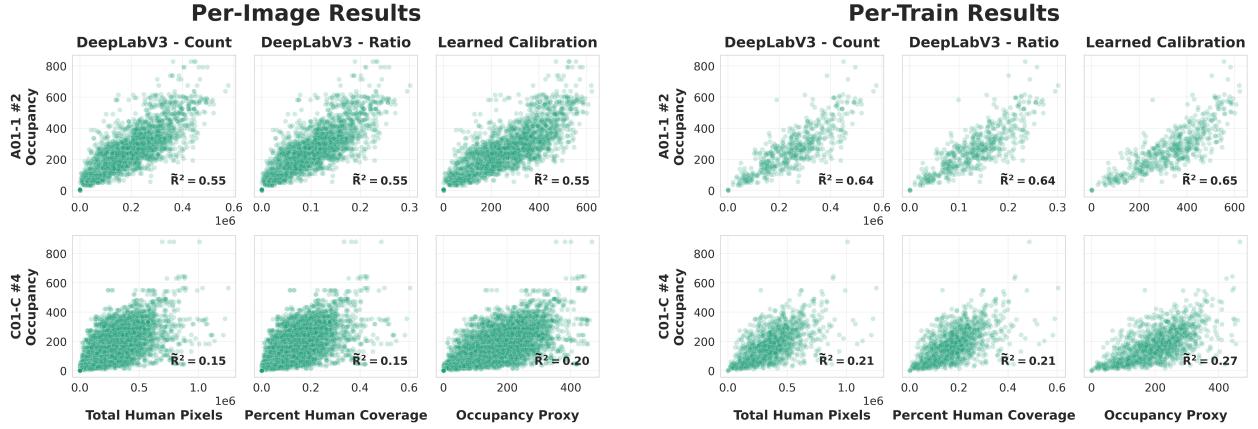


**FIGURE 15:** Confusion matrices for per-platform Crowd-ViT models, evaluated on classifying *per-train-arrival event* platform crowd levels, shown by platform type. Left: center platforms. Right: side platforms. Top row: training dataset results. Bottom row: testing dataset results. Values are normalized by row, with diagonals indicating correct classifications.

## Segmentation

We evaluate segmentation-derived estimates of platform occupancy at per-image and per-train-arrival event levels using the DeepLabV3 semantic segmentation model outputs. As outlined earlier, three features are extracted: (i) raw binary (person/no-person) pixel counts, (ii) the ratio of these pixels within platform-specific regions of interest (see Figure 4), and (iii) calibrated estimates derived via our MILP-based ridge regression approach, applied only to cameras with at least 600 annotated samples in the training set (platforms A01-1, A01-2, and C01-C).

Figure 16 shows strong correlations among these segmentation-based representations, with calibrated outputs aligning most closely with the ODX ground truth. Normalized  $\tilde{R}^2$  scores remain



(a) *Per-image*: Each data point corresponds to a single CCTV frame. The higher temporal resolution introduces noise but captures frame-level variation in crowding.

(b) *Per-train-arrival event*: Maximum segmentation value across all frames per arrival. Temporal aggregation smooths frame-level fluctuations and sharpens the relationship to true occupancy.

**FIGURE 16:** Semantic segmentation (DeepLabV3) features plotted against true platform occupancy estimates from ODX. In both figures, *top*: Metro Center Red line platform A01-2 (side platform); *bottom*: Gallery Place Chinatown Yellow/Green line C01-1 (center platform).

moderate, with errors increasing at mid-to-high occupancy levels, similar to detection-based methods. Side platforms, such as A01-1, demonstrate particularly strong performance due to favorable camera angles, where calibration significantly improves model accuracy. However, pixel-based methods tend to over-count relative to true occupancy, notably for camera #2 on platform A01-1.

Certain cameras exhibit non-linear, saturating response patterns, suggesting that fixed-pixel mappings may inadequately capture spatial crowd complexities in dense or distorted scenes, even with calibration.

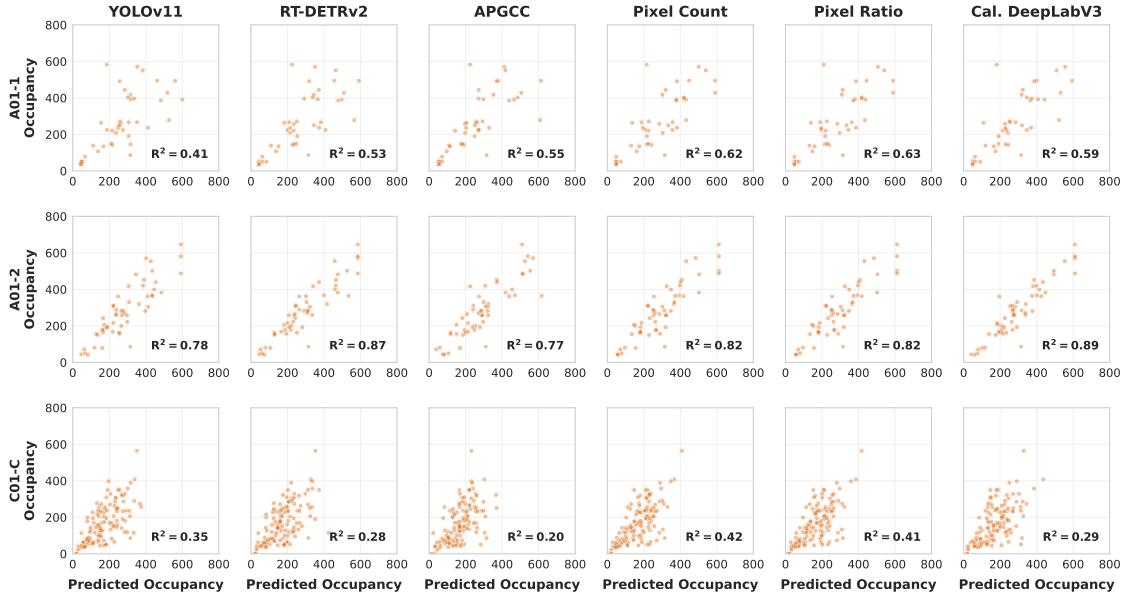
As observed with the object detection models, aggregating segmentation features per train-arrival event, selecting maximum values from the three frames around each train-arrival event’s magic moment, reduces variability and enhances estimation stability (see Figure 16b), consistent with findings from object detection methods.

Overall, calibrated segmentation estimates consistently outperform raw pixel-based methods. Nevertheless, uncalibrated ratios and pixel counts remain strong predictors, highlighting the utility of segmentation-based features, particularly in scenarios where calibration data is limited.

### Cross-Camera Per Train-Arrival Event Estimation

Most platforms are monitored by multiple CCTV cameras, allowing various views on the same train-arrival event to estimate platform crowding. Thus, we leverage this multi-view setup using an LGBM model to integrate per-camera CV-derived features into a unified platform-level occupancy estimate for each train-arrival event. Due to LGBM’s robustness to missing values, the model can accommodate partial input, i.e., when not all camera views are available for a given train-arrival event. No additional features are included in the LGBM feature space to show the value of per-arrival and per-platform CV-based estimates for crowding state estimation. The analysis is restricted to platforms A01-1/2 and C01-C, with sufficient data coverage across all model types.

Figure 17 presents the results of such a multi-camera LGBM model, evaluated on the previously stratified testing dataset. While the model successfully learns patterns, substantial variation remains, particularly at high occupancy levels. The performance varies across platforms and CV methods, with no one universally dominant CV method in terms of  $R^2$  values.



(a) LGBM predictions on the testing dataset using per-train-arrival event aggregated CV features across all camera views for platforms A01-1/2 and C01-C.

**FIGURE 17:** Cross-camera per-train-arrival event platform occupancy estimates using CV-derived features aggregated across multiple cameras.

Table 2 summarizes model performance metrics across the three platforms on the testing dataset. The calibrated DeepLabV3 model achieves the best *overall* results, with the lowest MAE (56.42), 95<sup>th</sup> percentile AE (164.51), and wMAE (68.66), as well as the highest  $R^2$  (0.64), supporting its operational viability for high-fidelity per-train-arrival event estimation. The wMAE specifically represents a 7.37% improvement over the next-best model, namely RT-DETRv2.

Model	MAE	95 <sup>th</sup> AE	wMAE $\downarrow$	$R^2$
YOLOv11	67.87	176.41	80.18	0.57
Crowd-ViT	68.58	181.02	78.11	0.62
APGCC	64.72	201.22	75.50	0.57
RT-DETRv2	63.06	169.00	74.12	0.61
<b>Cal. DeepLabV3</b>	<b>56.42</b>	<b>164.51</b>	<b>68.66</b>	<b>0.64</b>

**TABLE 2:** Performance of CV-based per-train-arrival event platform occupancy estimates. Results are averaged over platforms A01-1/2 and C01-C. The calibrated DeepLabV3 segmentation approach outperforms other models across all key metrics.

### Joint Per-15-minute Bin Estimation

We now target the mean 15-minute bin platform occupancy. Here, we leverage the previously LGBM-computed per-train-arrival event estimates and compute their mean across each 15-minute bin. The results presented in Table 3 show that the calibrated DeepLabV3 approach again achieves the lowest MAE and wMAE values, although it does not yield the best performance in terms of 95<sup>th</sup> AE or  $R^2$ . Regarding wMAE, an approximately 2.15% improvement over the next best model, APGCC, is achieved.

These findings align with our assessment of the segmentation results, where we concluded that the variance of the segmentation-derived features increases significantly in high occupancy regimes. Furthermore, these results are computed from simple mean aggregation, amplifying the previously identified over- and under-estimation trends.

Model	MAE	95 <sup>th</sup> AE	↓wMAE	$R^2$
YOLOv11	61.74	157.35	67.96	0.51
Crowd-ViT	61.21	153.02	64.17	<b>0.61</b>
RT-DETRv2	55.64	<b>144.13</b>	61.44	0.58
APGCC	58.59	156.27	61.02	0.56
<b>Cal. DeepLabV3</b>	<b>54.10</b>	152.42	<b>59.71</b>	0.57

**TABLE 3:** Results for current 15-minute bin mean occupancy estimation computed from multi-camera per-train-arrival event LGBM estimates.

### OPERATIONAL INTEGRATION

Integrating vision-based methods, such as object detection, classification, and segmentation, involves computational and operational costs. Empirical estimates of training and inference times, as well as resource requirements from our experiments are summarized in Table 4.

	YOLOv11 SAHI	RT-DETRv2 SAHI	APGCC	Crowd-ViT	DeepLabV3	DeepLabV3 Calibration
Training [s]	N/A	N/A	N/A	1.190±0.068	N/A	0.009±0.007
Inference [s]	3.453±0.296	3.273±0.852	0.25±0.023	0.073±0.003	0.625±0.049	0.003±0.001
Requirements	CPU/GPU	CPU/GPU	CPU/GPU	GPU/Data/Occupancy	CPU/GPU	CPU/Seg. Maps/Occupancy

**TABLE 4:** Per-image resource requirements and timing for single-frame inference, tested on a node with one Nvidia V100 GPU, 20 Intel Xeon-G6 cores, and 170GB RAM.

SAHI-enhanced detection models (YOLOv11, RT-DETRv2) require over three seconds per image due to SAHI’s multi-patch evaluations (approximately 0.073–0.076 seconds per patch), whereas APGCC, Crowd-ViT, and DeepLabV3 achieve sub-second inference times, suitable for near real-time applications. In practice, however, inference speed is less restrictive than access to labeled data and platform occupancy counts.

For agencies lacking labeled data, pre-trained detection models such as RT-DETRv2 and APGCC offer readily implementable solutions, balancing accuracy and cost-efficiency. RT-DETRv2 is especially advantageous due to its superior performance and ability to detect other objects such as left-behind luggage and bicycles. If threshold-based monitoring is sufficient, segmentation approaches such as DeepLabV3 appeal for their simplicity, speed, and stable outputs. When occu-

pancy data is available, calibration through the MILP-learned weight maps allows a direct mapping to actual occupancy values, without notably increasing inference times.

We recommend two deployment strategies: agencies with occupancy labels and sufficient computing resources should consider our calibrated segmentation pipeline. In contrast, those without labeled data can effectively use pre-trained detection models and keep the option to fine-tune these later through manual labeling.

Large transformer models (e.g., Crowd-ViT) require substantial labeled data to generalize effectively for classification tasks. While well-defined operational crowd level thresholds can improve their accuracy, high data demands limit the advisability of training and employing these models.

Finally, aggregating multiple frames and camera views reduces noise significantly, thus improving predictive accuracy. Hence, we recommend capturing multiple frames when seeking a crowd estimation to enhance robustness. Agencies aiming to manage high-crowding events should strategically reduce sampling frequencies during off-peak times, which optimizes resource usage without compromising operational insight.

## LIMITATIONS & DISCUSSION

This study demonstrated that CCTV-derived features offer valuable, platform occupancy estimation insights in urban rail environments, such as the WMATA Metrorail system. We evaluated several CV methods, including object detection (YOLOv11 and RT-DETRv2 with SAHI), head counting (APGCC), crowd-level classification (Crowd-ViT), and segmentation (DeepLabV3) combined with a novel MILP-based calibration approach. Among these, the calibrated semantic segmentation model proved the most accurate at the per-train-arrival event, and slightly more accurate at the per-15-minute bin levels. Detection and head-counting methods typically undercounted but significantly benefited from multi-camera aggregation and non-linear rescaling via LGBM.

Performance varied notably between side and center platforms. Side platforms generally exhibited better accuracy, attributed to advantageous camera positions and fewer structural constraints. Conversely, center platforms, such as Metro Center C01-C, feature lower ceilings, limited camera coverage, and increased inter-camera variability, reducing overall performance. While aggregation methods partially mitigated these issues, further improvements may require more sophisticated modeling approaches, improved ground truth collection to account for rider mixing, or optimized camera placements, if possible. Although elevated center platforms (e.g., Gallery Place–Chinatown, Navy Yard–Ballpark) hinted at better performance, limited data prevented definitive conclusions.

Our first limitation concerns the spatiotemporal coverage of the CCTV dataset, which was limited, particularly during peak crowding events. This may affect the studied CV-and LGBM-based models' robustness under extreme conditions. Expanding data collection to capture a broader range of operational scenarios could further strengthen generalizability.

Following, another limitation stems from our ground truth data. As a trip assignment algorithm, ODX is not designed to provide highly accurate estimates of rail platform occupancy, and likely introduces unquantifiable noise and potential misestimation in our study, particularly at busy transfer stations where fare gate data does not reflect intra-station movement. Future work should consider supplementing ODX with manual counts or applying the insights developed here to calibrate and refine ODX estimates, ultimately enabling their reuse as a more accurate ground truth.

Additionally, relying on pre-trained, off-the-shelf models for object detection, head counting, and segmentation without domain-specific fine-tuning limited accuracy. Systematic undercounting was observed in high-density scenes, driven by poor lighting, compression artifacts, occlusions, and variable image quality. Future work should investigate how fine-tuning these models to the transit environment could improve robustness and accuracy.

The moderate classification accuracy (63.42%) of the Crowd-ViT model reflected inherent challenges, including class imbalance and visual similarity among neighboring crowd-level classes. While our findings show, that such models can learn to distinguish crowded from non-crowded scenes, enhancing their effectiveness would require significantly larger and more balanced datasets, as well as refined and operationally-informed crowd level definitions.

Finally, structural constraints inherent in our modeling approach posed challenges. While LGBM effectively handled missing data and provided interpretability, it assumed independent samples, neglecting temporal dependencies and recurrent patterns. Integrating real-time visual inputs into advanced temporal models such as transformers, LSTMs, or RNNs, combined with traditional transit data sources, could more effectively capture the dynamic nature of crowd movements.

Despite these limitations, our results demonstrate that even univariate, image-based inputs can provide accurate, near real-time estimation of rail platform occupancy. For several platforms with favorable camera configurations, post-processed predictions achieved  $R^2 \geq 0.8$ , underscoring the practical viability of CV-based approaches. These findings highlight the promise of CCTV as an emergent data source for operational decision-making and call for greater attention to its integration in future transit analytics systems.

## CONCLUSION

This work explored CCTV imagery as a novel real-time data source for estimating platform-level crowding within the WMATA Metrorail system, leveraging a dataset of 43,232 images collected from seven platforms and 18 cameras. We automatically labeled these images using occupancy estimates from the WMATA-internal integration of the Origin Destination Interchange (ODX) algorithm, aligned with train-arrival events.

We evaluated various CV approaches, namely, object detection (YOLOv11, RT-DETRv2), head counting (APGCC), crowd-level classification (Crowd-ViT), and calibrated semantic segmentation (DeepLabV3), for their capability to provide platform-level crowding estimates at various temporal resolutions. The DeepLabV3 semantic segmentation model, combined with our novel MILP-based calibration, allows direct crowd count extraction from segmentation maps in near real-time, outperforming other approaches.

This research advances platform crowding estimation by demonstrating that CCTV-derived features provide interpretable and operationally relevant insights. Future studies can extend this work by integrating these insights into predictive models for real-time operational decision-making and as feature input to time-series forecasting approaches.

## ACKNOWLEDGMENTS

The authors would like to thank WMATA for supporting this project through its academic research partnership with MIT, and the MIT SuperCloud and Lincoln Laboratory Supercomputing Center for providing high-performance computing resources that have contributed to generating the results reported in this paper. Generative AI provided by Grammarly and OpenAI ChatGPT 4o has been used to improve the presented text's grammar, spelling, and wording. No content has been

generated.

**AUTHOR CONTRIBUTION STATEMENT**

The authors confirm their contribution to the paper: Study conception and design: RF, AA, AS, JZ; data collection: RF; analysis and interpretation of results: RF, AA; draft manuscript preparation: RF, AA, AS, JZ, GP, IT. All authors reviewed the results and approved the final version of the manuscript. The authors do not have any conflicts of interest to declare.

## REFERENCES

1. Miller, E., G. E. Sánchez-Martínez, and N. Nassir, Estimation of Passengers Left Behind by Trains in High-Frequency Transit Service Operating Near Capacity. *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 2672, No. 8, 2018, pp. 497–504.
2. Tirachini, A., D. A. Hensher, and J. M. Rose, Crowding in public transport systems: Effects on users, operation and implications for the estimation of demand. *Transportation Research Part A: Policy and Practice*, Vol. 53, 2013, pp. 36–52.
3. Li, Z. and D. A. Hensher, Crowding and public transport: A review of willingness to pay evidence and its relevance in project appraisal. *Transport Policy*, Vol. 18, No. 6, 2011, pp. 880–887.
4. Noursalehi, P., H. N. Koutsopoulos, and J. Zhao, Predictive decision support platform and its application in crowding prediction and passenger information generation. *Transportation Research Part C: Emerging Technologies*, Vol. 129, 2021, p. 103139.
5. Kopsidas, A., A. Douvaras, and K. Kepaptsoglou, Extracting Metro Passenger Flow Predictors from Network's Complex Characteristics. In *Complex Networks and Their Applications XI* (H. Cherifi, R. N. Mantegna, L. M. Rocha, C. Cherifi, and S. Miccichè, eds.), Springer International Publishing, Cham, 2023, pp. 529–540.
6. Wang, Q., S. Wang, D. Zhuang, H. Koutsopoulos, and J. Zhao, Uncertainty quantification of spatiotemporal travel demand with probabilistic graph neural networks. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 25, No. 8, 2024, pp. 8770–8781.
7. Saleh, S. A. M., S. A. Suandi, and H. Ibrahim, Recent survey on crowd density estimation and counting for visual surveillance. *Engineering Applications of Artificial Intelligence*, Vol. 41, 2015, pp. 103–114.
8. LeCun, Y., K. Kavukcuoglu, and C. Farabet, Convolutional networks and applications in vision. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, IEEE, Paris, France, 2010, pp. 253–256.
9. Redmon, J., S. Divvala, R. Girshick, and A. Farhadi, You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016.
10. Zhao, Y., W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, Y. Liu, and J. Chen, DETRs beat yolos on real-time object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2024, pp. 16965–16974.
11. Florez, S. L., A. G. Briones, J. Pavón, R. Fuentes-Fernández, and J. M. Corchado, Monitoring System for Detecting Non-inclusive Situations in Smart Cities. In *Trends in Sustainable Smart Cities and Territories* (L. F. Castillo Ossa, G. Isaza, O. Cardona, O. D. Castrillón, J. M. Corchado Rodriguez, and F. De la Prieta Pintado, eds.), Springer Nature Switzerland, Cham, 2023, pp. 405–416.
12. Goh, C. G., W. H. Lim, J. Chua, and I. Atmosukarto, Image Analytics for Train Crowd Estimation. In *2018 Digital Image Computing: Techniques and Applications (DICTA)*, 2018, pp. 1–6.
13. Chen, I.-H., W.-T. Chen, Y.-W. Liu, M.-H. Yang, and S.-Y. Kuo, *Improving Point-based Crowd Counting and Localization Based on Auxiliary Point Guidance*, 2024, arXiv:2405.10589 [cs].

14. Cheng, Z.-Q., Q. Dai, H. Li, J. Song, X. Wu, and A. G. Hauptmann, Rethinking Spatial Invariance of Convolutional Networks for Object Counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 19638–19648.
15. Khan, K., W. Albattah, R. U. Khan, A. M. Qamar, and D. Nayab, Advances and Trends in Real Time Visual Crowd Analysis. *Sensors*, Vol. 20, No. 18, 2020.
16. Akyon, F. C., S. O. Altinuc, and A. Temizel, Slicing Aided Hyper Inference and Fine-tuning for Small Object Detection. *2022 IEEE International Conference on Image Processing (ICIP)*, 2022, pp. 966–970.
17. Dosovitskiy, A., L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*, 2021.
18. Abdelhalim, A. and J. Zhao, Computer vision for transit travel time prediction: an end-to-end framework using roadside urban imagery. *Public Transport*, 2024.
19. Chen, L.-C., G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, Vol. 40, No. 4, 2017, pp. 834–848, publisher: IEEE.
20. Ronneberger, O., P. Fischer, and T. Brox, U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, part III* 18, Springer, 2015, pp. 234–241.
21. Wong, V. W. H. and K. H. Law, Fusion of CCTV Video and Spatial Information for Automated Crowd Congestion Monitoring in Public Urban Spaces. *Algorithms*, Vol. 16, No. 3, 2023, p. 154.
22. Peppa, M. V., D. Bell, T. Komar, and W. Xiao, Urban Traffic Flow Analysis Based on Deep Learning Car Detection from CCTV Image Series. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XLII-4, 2018, pp. 499–506.
23. Wang, Y., C. Li, F. Dou, J. Dang, and Y. Wang, *Assessment of passenger flow on metro platform based on video images*, 2025, type: Conference Poster.
24. Jocher, G., A. Chaurasia, and J. Qiu, *Ultralytics YOLO*, 2023.
25. Lv, W., Y. Zhao, Q. Chang, K. Huang, G. Wang, and Y. Liu, *RT-DETRv2: Improved Baseline with Bag-of-Freebies for Real-Time Detection Transformer*, 2024, arXiv:2407.17140 [cs].
26. Chen, L.-C., G. Papandreou, F. Schroff, and H. Adam, *Rethinking Atrous Convolution for Semantic Image Segmentation*, 2017, arXiv:1706.05587 [cs].
27. Gordon, J. B., H. N. Koutsopoulos, and N. H. Wilson, Estimation of population origin–interchange–destination flows on multimodal transit networks. *Transportation Research Part C: Emerging Technologies*, Vol. 90, 2018, pp. 350–365.
28. Ultralytics, *Ultralytics GitHub repository*, 2025.
29. University, P., *RT-DETR v2 R101VD model*, 2025.
30. Lin, T.-Y., M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, *Microsoft COCO: Common Objects in Context*, 2014, version Number: 3.

31. Zhang, Y., D. Zhou, S. Chen, S. Gao, and Y. Ma, Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016.
32. Parmar, N. J., A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran, Image Transformer. In *International Conference on Machine Learning (ICML)*, 2018.
33. VainF, *DeepLabV3Plus-PyTorch*, 2025.
34. Everingham, M., S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, The Pascal Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision*, Vol. 111, No. 1, 2015, pp. 98–136.
35. Domahidi, A., E. Chu, and S. Boyd, ECOS: An SOCP solver for embedded systems. In *European control conference (ECC)*, 2013, pp. 3071–3076.
36. Diamond, S. and S. Boyd, CVXPY: a Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 2016.
37. Ke, G., Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, LightGBM: a highly efficient gradient boosting decision tree. In *Advances in neural information processing systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), Curran Associates, Inc., 2017, Vol. 30.
38. Lundberg, S. M. and S.-I. Lee, A unified approach to interpreting model predictions. In *Advances in neural information processing systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), Curran Associates, Inc., 2017, Vol. 30.