Constellation Dataset: Benchmarking High-Altitude Object Detection for an Urban Intersection

Mehmet Kerem Turkcan
Columbia University
New York, NY
mkt2126@columbia.edu

Chengbo Zang
Columbia University
New York, NY
cz2678@columbia.edu

Sanjeev Narasimhan
Columbia University
New York, NY
sn3007@columbia.edu

Gyung Hyun Je
Columbia University
New York, NY
gj2353@columbia.edu

Bo Yu
Columbia University
New York, NY
by2345@columbia.edu

Mahshid Ghasemi
Columbia University
New York, NY
mg4089@columbia.edu

Javad Ghaderi
Columbia University
New York, NY
jg3465@columbia.edu

Gil Zussman
Columbia University
New York, NY
gil.zussman@columbia.edu

Zoran Kostic
Columbia University
New York, NY
zk2172@columbia.edu

Abstract

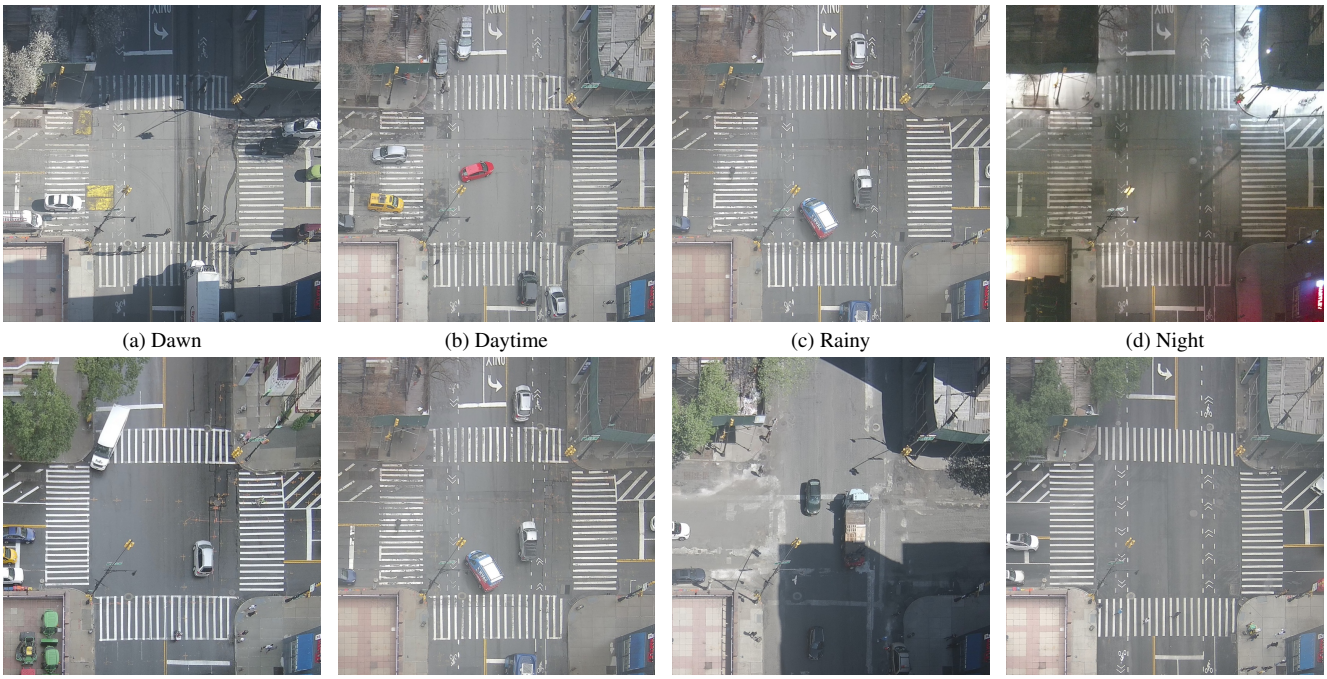
We introduce Constellation, a dataset of $13K$ *images suitable for research on detection of objects in dense urban streetscapes observed from high-elevation cameras, collected for a variety of temporal conditions. The dataset addresses the need for curated data to explore problems in small object detection exemplified by the limited pixel footprint of pedestrians observed tens of meters from above. It enables the testing of object detection models for variations in lighting, building shadows, weather, and scene dynamics. We evaluate contemporary object detection architectures on the dataset, observing that state-of-the-art methods have lower performance in detecting small pedestrians compared to vehicles, corresponding to a 10% difference in average precision (AP). Using structurally similar datasets for pretraining the models results in an increase of 1.8% mean AP (mAP). We further find that incorporating domain-specific data augmentations helps improve model performance. Using pseudo-labeled data, obtained from inference outcomes of the best-performing models, improves the performance of the models. Finally, comparing the models trained using the data collected in two different time intervals, we find a performance drift in models due to the changes in intersection conditions over time. The best-performing model achieves a pedestrian AP of 92.0% with 11.5 ms inference time on NVIDIA A100 GPUs, and an mAP of 95.4%.*

1. Introduction

Urban landscapes are transitioning towards "smart cities" with the underlying aim of catering to societal welfare [2, 31]. Considering deployment challenges and low-latency solution requirements, traffic intersections stand out as optimal locations for embedding intelligence nodes in smart cities. A city's operations can be enhanced through the interlinking of these nodes situated at adjacent intersections.

Dense urban locales are particularly challenging for autonomous vehicles because of occlusions, high density of car and pedestrian traffic, multitude of vehicles navigating in diverse trajectories at varied velocities, physical obstructions, and the unpredictable movements of pedestrians. The layout of buildings and their electrical power and communications infrastructure present opportunities to install sensors on city buildings, use V2X and related technologies and thereby upgrade autonomous vehicles to cloud-connected vehicles [8, 10]. The ultimate goals are a dramatic increase in pedestrian safety and improvements in traffic flow.

Utilizing cameras to monitor and track pedestrian and vehicular movement is an effective solution for traffic surveillance and increasing safety at smart intersections. A great majority of cameras in cities are positioned no higher than the first floor, which presents complications for object detection associated with occlusions and the intricacies of top-down view transformations. In major metropolises, cameras can be positioned at higher altitudes, which facilitates easier generation of bird's-eye perspective images



(a) Dawn (b) Daytime (c) Rainy (d) Night

(e) Old Pavement (2019) (f) Faded Pavement (g) Unpaved (h) Repaved

Figure 1. Constellation contains different scenes, with changing time-of-day, weather conditions and background elements for the same camera. (a-d) Different weather and time-of-day conditions; (e-h) changes to the scene background.

through low-latency perspective transformations.

The overwhelming majority of the available datasets for object detection in urban settings are collected using low-altitude cameras. This is convenient from the perspective of cameras' installations but frequently results in visual occlusions of objects.

In this study, we introduce Constellation, a dataset containing $13,314$ human-annotated images with bounding box annotations for pedestrians and vehicles captured at an intersection in New York City using high-elevation cameras on the COSMOS testbed [26]. This dataset offers the ability to study pedestrian and vehicle detection in a real-world urban environment, enabling the development of robust algorithms that can handle the challenges posed by diverse weather conditions, lighting variations, and complex scenes. Constellation is the first dataset to feature an actual deployment in an urban metropolis, spanning multiple years, seasons, weather conditions, and times of day.

Using the Constellation dataset, we compared contemporary object detection models and found that single-stage YOLO models [35] are the best-performing in terms of inference time and precision. Transformer-based RT-DETR models and Faster R-CNN-based CFINet obtained poorer results [23, 45]. We then investigated pretraining based on other similar datasets and custom augmentations to improve model performance. We found that using pretraining datasets improves the model performance on our Constellation dataset by 1.8% mAP, providing a visible increase in pedestrian detection performance. We used the best-performing VisDrone-pretrained YOLOv8x model to automatically label data from the intersection and used this newly acquired pseudo-labeled data as a pretraining dataset, showing that this approach is competitive against using external datasets for pretraining. We release the codebase and baseline models trained with Constellation.

2. Related Works

Pedestrian and Vehicle Detection A wealth of datasets from low-altitude on-vehicle or stationary cameras for vehicle and pedestrian detection have been released [11, 25, 33, 34]. Largely, these datasets do not feature scenes of urban metropolises and focus either on vehicles or pedestrians. Many of the publicly available traffic datasets specifically feature footage captured from street-level elevations. There exist datasets for drone-based high-altitude object detection, which largely focus on campus areas or highways rather than urban streetscapes, and some of which lack annotated bounding boxes needed for object detection [14–16, 30, 40, 43]. Images from low-altitude cameras are friendly for object detection, segmentation, and tracking applications since vehicles and pedestrians all occupy a large enough number of pixels (greater than 64x128). Sizeable pixel areas make it easy for convolutional networks to cap-

ture high-quality feature maps and are present heavily in standard benchmark datasets like COCO [20].

Real-Time Object Detection For real-time object detection, single-stage object detectors based on SSD or YOLO have gained popularity in recent years [4, 12, 22, 27–29, 36, 42]. Unlike two-stage detectors that feature region proposal networks which propose potential bounding boxes, single-stage detection models generate a fixed number of bounding box proposals based on integrated low-dimensional feature maps. Recently proposed transformer-based architectures like the detection transformer (DETR) lack the real-time performance required for deployment [6]. Recently, Real-Time Detection Transformer (RT-DETR), which replaces the encoder in DETR with a hybrid encoder, has been proposed as an alternative to YOLO architectures [23].

Small Object Detection Object detection models typically struggle with small object detection. Tiny pedestrians in a bird's eye camera view are represented by inadequately precise feature maps. This results in poor detection and tracking accuracy. Many approaches have been proposed to improve small object detection, especially in the context of aerial object detection captured by VisDrone and DOTA datasets [38, 40, 41, 47]. We note that aerial object detection datasets contain significantly different views than the static high-altitude infrastructure-colocated camera placements which we consider in this study.

For YOLO models, adding extra levels of the Feature Pyramid Network (FPN) has been suggested to improve the size of the feature maps, and thus the capability to increase small object detection accuracy [17]. Recently, RT-MDet proposed modifying the YOLO architecture by integrating large-kernel depth-wise convolutions and soft labels for aerial data [24]. PP-YOLOE models use several architectural choices to improve upon YOLO [42]. LSKNet proposes large and selective kernel modules to give models the contextual information needed [19]. Spatial Transform Decoupling improves upon ViT-based object detectors by estimating spatial transform parameters through separate network branches [44]. Changing the resolution of the input through rescaling, slicing, super-resolution, or changing the model resolution, has been explored extensively for improving the performance of object detectors [1, 7, 32].

In this paper, we bring forward a new dataset aimed at high-altitude object detection in urban environments. We seek to motivate models that work in this real-world scenario that differs from existing datasets. Aerial drone-based imagery fundamentally differs from our use case due to the difference in capture height and stationarity of the camera; in many cities, urban laws restrict the collection of drone imagery, making it unsuitable for real-world deployment. High-altitude infrastructure-colocated object detection uses significantly different angles than aerial high-altitude views and requires building models robust to changes in conditions. The data split of the Constellation dataset accounts for a variety of environmental and time-of-day conditions, across several years of both training and testing, making it fundamentally different from other datasets. We evaluate Constellation on state-of-the-art models for aerial object detection.

Table 1. Training and testing dataset sizes, grouped by the different weather conditions.

Weather	Split	# of Images
Overcast	Train	1,205
	Test	602
Sunny	Train	7,355
	Test	1,282
Night	Train	1,600
	Test	450
Sun/Sharp Shadows	Train	70
	Test	150
Foggy	Train	-
	Test	600

3. Dataset Creation and Analysis

Constellation is a dataset of images of a typical street intersection in an urban metropolis, acquired by a high-altitude camera installed on a side of a building. The dataset contains images collected in two different periods: years 2019/2020 and 2023. There are a total of 28 different time intervals (in length between 1 minute and 1 day) in the dataset, with different times of day, time of year, weather, and background qualities.

The dataset contains $13,314$ human-annotated images with bounding box annotations for pedestrians and vehicles. An overview of the dataset grouped by weather condition, with the suggested training/test set splits we use for evaluation is provided in Table 1 with the number of images per split. The dataset is split into 5 conditions: (i) overcast, (ii) sunny, (iii) night, (iv) sunny with sharp shadows, and (v) foggy. The foggy images in the dataset correspond to a rare day with wildfire smoke.

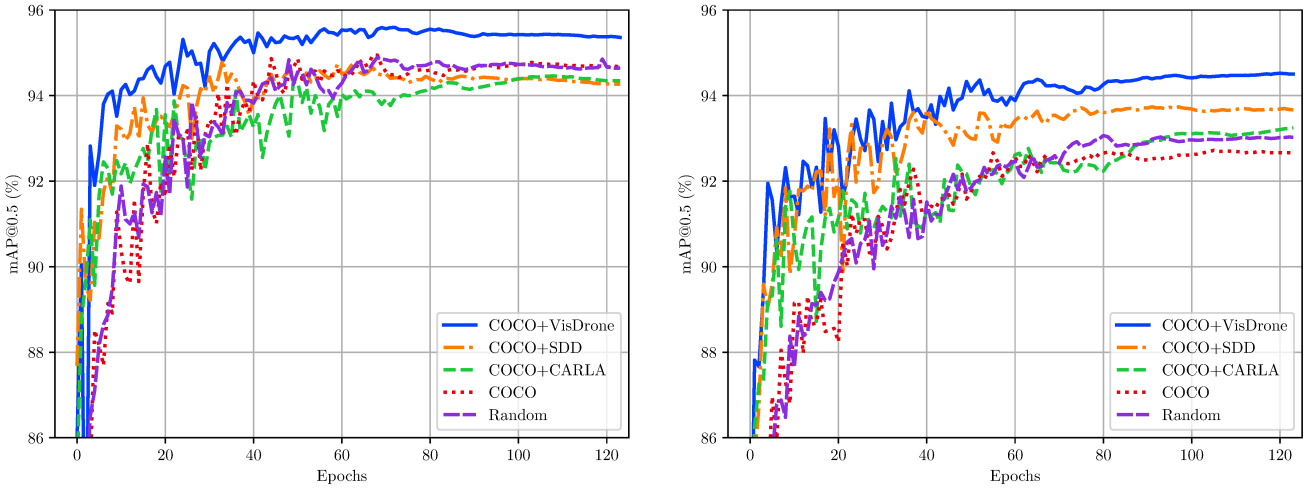
Dataset Description An overview of the dataset is shown in Figure 1. The dataset contains data collected at all times of day (a-d), which results in a large variability in lighting conditions as sharp shadows are cast by buildings over the intersection depending on the time of day. These shadows result in large brightness changes that object detection models need to take into account (for example scenes shown in (a) and (g)). The background is highly variable due to the changes in shadows and lighting, frequent changes to the scene background, birds flying over the intersection, parked

(a) (b)

Figure 2. Examples of two complex crowded frames from Constellation.

(a) Constellation (b) CARLA (c) Stanford Drone Dataset (d) VisDrone

Figure 3. Different data sources used for experiments: (a) Constellation; (b) CARLA; (c) Stanford Drone Dataset; (d) VisDrone.



(a) YOLOv8x Models (b) YOLOv8n Models

Figure 4. Comparison of the model performance for (a) YOLOv8x, and (b) YOLOv8n (b) models for different pretraining schemes. Performance for the two model architectures follows similar trends with VisDrone pretraining achieving better results for both architectures.

cars, and waiting pedestrians. Moreover, the foreground is difficult to estimate given the small size of the shadows and their similarity to pedestrians wearing dark colors. Frames cover all seasons of the year, and weather conditions that include (i) sunny, (ii) overcast, and (iii) rain.

The dataset comprises of $13,314$ annotated frames, divided into $10,230$ training and $3,084$ test images collected at 28 different time intervals. The raw frames are cap-

tured from the side of a building with a viewing angle that is slightly less than perpendicular to the street plane. The frames are transformed using a perspective transformation such that the viewing angle to the street is exactly 90 degrees, to represent the bird's-eye view perspective. The images in the dataset have a resolution of 832x832, with the intersection centered in the frame. The original images are captured in 1920x1080 resolution and are subsequently transformed to obtain a top-down view and cropped to square images. Following the input size scaling rules for YOLO models and to enable low processing times without relying on sliced inference techniques [1], the input image size is chosen as 832x832. For all frames, axis-aligned bounding boxes are provided for pedestrian and vehicle classes. All annotations were done in two rounds where the work of an initial annotator was reviewed by a second annotator.

Eighteen of the time intervals are obtained from videos annotated at 10 frames per second (FPS) in years 2019&2020. Five of these contain 300 frames, and the other 13 contain 450 frames - for a total of 7,350 annotated frames. In addition to these, the dataset contains 5,964 frames captured and annotated at 20 s intervals in 10 different scenes during 2023 to increase variability in the subject, lighting, weather, time-of-year, and background conditions.

Constellation features many challenging frames with large crowds where picking individuals apart is difficult; we show some examples of such scenes in Figure 2, where annotation requires thinking and care by human annotators.

Training Set The training set consists of 10,230 frames divided into 19 subsets collected on different days with different weather and time-of-day conditions. 15 of the subsets are from 2019 and 2020 (corresponding to 6,160 frames), and 4 are from 2023 (corresponding to 4,070 frames). All frames are collected from an intersection in Manhattan.

Test Set The test set consists of 3,084 frames divided into 9subsets collected on different days with different weather and time-of-day conditions. This test set contains 12,399 vehicles and 14,229 pedestrian bounding boxes. 3 of the subsets are from years 2019 and 2020 (corresponding to 1,201 frames), and 6 are from 2023. 5 of the subsets from 2023 feature the intersection before repavement. Each subset contains 70 to 2,000 frames. As part of the dataset, we provide splits of data depending on the acquisition year for studies involving the transfer of knowledge from old frames to new frames.

Pseudo-labeled Data Collection After the initial training of object detection models based on the data above, we used the YOLOv8x model in inference mode and collected 50,557 more images following a tracking-by-detection paradigm, using the tracked objects to provide additional annotations [18]. To deal with scenarios where an object is not detected, we mask the image to keep only the pixels where an object was detected (with 8-pixel padding) and use the median of the past 120 seconds (captured at 1FPS) as the background pixels for the rest of the image. For generating this dataset, during data collection, we use the ByteTrack tracking algorithm [46].

4. Experiments

We explored the performance of object detection models in terms of accuracy in detection and inference time on our dataset over five experiments: (i) model architecture, (ii) pretraining dataset and image augmentation, (iii) model scaling, (iv) semi-supervised training, and (v) performance drift. For computation, we use eight A100 NVIDIA GPUs with 40 GB VRAM and a batch size of 16. For YOLOv8x models trained at 2x resolution, we use a batch size of 8 during training due to the higher memory requirements. Since we focus on small object detection and the sensitivity of IoU to small bounding boxes, we report AP and mAP for Intersection over Union (IoU) value of of 0.6.

4.1. Evaluating Model Architectures

Different models offer different trade-offs between detection accuracy and inference time. For object detection experiments, we explored YOLOv8, RT-DETR, and CFINet models [12, 23, 36]. An alternative to the one-stage object detectors considered, CFINet integrates Faster R-CNN with a coarse-to-fine region proposal network and feature imitation learning [45]. As an alternative to models that were trained at the input resolution (832x832), we also explore models trained at 2x resolution. To do so, we naively upscale our input images to 1664x1664 resolution using bilinear interpolation or by using Real-ESRGAN [39]. This allows for the CNN layers to process small objects with better precision while avoiding the large inference time cost of applying super-resolution. Table 2 shows the results of trying these different architectures. All models are trained for 125 epochs. 2x refers to models trained using bilinear interpolation, while 2x+SR refers to models trained using Real-ESRGAN. As the use of Real-ESRGAN adds a sizable inference time latency, we give results for two separate models. P2, P6 and P2-P6 refer to YOLO architectures using different detection layers; default models use layers P3-P5, P2 models use layers P2-P5, P6 models use layers P3-P6. We propose a modified architecture that uses layers P2-P6 [12, 21].

Although we achieve the best results when training the models with high-resolution inputs, we see competitive results at real-time constraints with the modified YOLOv8x architectures.

4.2. Evaluating Pretraining Datasets

Pedestrians and vehicles are classes with high amounts of variance. To help generalize our models better to novel sit-

Table 2. Per-class average precision, mean average precision, and inference time for each model for different architectures, using models pretrained on the COCO dataset.

Model Name	Pedestrian AP@0.5	Vehicle AP@0.5	mAP@0.5	Inference Time (ms)
YOLOv8x [12]	87.4	98.6	93.0	11.5
YOLOv8n (2x)	91.2	98.5	94.8	7.2
YOLOv8x (2x)	91.2	98.4	94.8	43.6
YOLOv8n (2x+SR)	90.1	98.4	94.2	7.2
YOLOv8x (2x+SR)	**91.3**	**98.7**	**95.0**	43.6
YOLOv8x (P2)	89.5	98.6	94.0	15.1
YOLOv8x (P6)	89.4	**98.7**	94.0	7.6
YOLOv8x (P2-P6)	89.9	**98.7**	94.3	24.5
DETR-l [23]	86.5	98.1	92.6	9.8
DETR-x [23]	87.3	97.8	92.3	14.5
YOLOv7 [37]	78.8	98.2	92.7	15.4
CFINet [45]	82.8	95.8	89.3	31.4
STD [44]	79.9	90.8	85.3	14.2
LSKNet-S [19]	64.8	90.2	77.5	31.1
PPYOLOE+ [42]	88.5	97.7	93.1	21.6
RTMDet [24]	87.5	98.3	92.9	24.5

Table 3. Comparison between the default augmentations in YOLOv8 and the new set of augmentations proposed for Constellation.

YOLOv8	Constellation
Blur	Blur
MedianBlur	-
ToGray	ToGray
CLAHE	-
RandomBrightnessContrast	RandomBrightnessContrast
RandomGamma	-
ImageCompression	-
	MotionBlur
	RandomShadow
	RandomRain

uations, we explored the integration of Constellation with previously released datasets in literature to improve model performance. To this end, we considered the drone-view VisDrone and Stanford Drone datasets and simulated 3D data generated using the CARLA simulator. We give an overview of the datasets we consider for pretraining in Figure 3.

VisDrone The VisDrone object detection challenge dataset contains 7,019 images [47]. VisDrone images are a mix of top-down and ground-level data with different perspectives and zoom levels. The dataset contains many different object classes, which we convert to our 2-class setup. The dataset contains 175,551 vehicles and 88,261 pedestrians.

Stanford Drone Dataset Stanford Drone Dataset consists of 60 videos collected from 8 different scenes [30]. We use 15 of the videos with a high pedestrian, biker, and vehicle concentration. By moving a sliding window with 20% overlap, we convert each input frame into a set of 832x832 frames. Post-processing, the dataset has 42,727 images containing 222,456 pedestrians and 35,587 vehicles.

CARLA We use the CARLA simulator to generate additional synthetic frames with ground truth annotations [9]. We replicated the placement of the camera artificially in CARLA and used the same perspective transformation as shown in Figure 3(b), and collected 26,984 frames with the same perspective transformation. The dataset contains 91,848 pedestrians and 67,815 vehicles.

Results For pretraining experiments, we initialize and train YOLOv8x models for 125 epochs on the chosen pretraining dataset: (i) VisDrone, (ii) SDD, and (iii) CARLA. After obtaining these pretrained models, we fine-tuned the model on Constellation for 125 epochs.

Experiments for different pretraining schemes are presented in Table 4. Due to the similar performance among all pretraining schemes, we choose to use the COCO-trained models. Surprisingly, the CARLA-generated dataset does not obtain competitive performance despite the compatibility of the setup for the overlapping pedestrian detection problem, indicating a big gap between simulation and real-world performance for this specific problem [13]. Comparing the visual features between CARLA renders and the real-world intersection, we believe this is due to the inadequate realism of CARLA renders when used for training.

The detection results are promising. However, for the application scenarios of high-altitude pedestrian and vehi-

Table 4. Per-class average precision and mean average precision for different pretraining dataset choices for the YOLOv8x architecture.

Pretraining Type (Without Augmentation)	Pedestrian AP@0.5	Vehicle AP@0.5	mAP@0.5
COCO	87.4	**98.6**	93.0
COCO+CARLA	88.6	98.2	93.4
COCO+VisDrone	**90.9**	**98.7**	**94.8**
COCO+SDD	87.4	**98.6**	93.0
Random Initialization	92.5	98.3	92.5

Table 5. Effect of augmentation on pretraining with a YOLOv8x model after 125 epochs. Mean average precision and per-class average precision are shown.

Pretraining Type (With Augmentation)	Pedestrian AP@0.5	Vehicle AP@0.5	mAP@0.5
COCO	90.7	98.6	94.7
COCO+CARLA	90.1	98.7	94.4
COCO+VisDrone	**92.0**	**98.8**	**95.4**
COCO+SDD	90.0	98.6	94.3
Random Initialization	90.6	98.7	94.7

Table 6. Per-class average precision and mean average precision for the final training with the pseudo-labeled data (BoxMask) used for pretraining.

Pretraining Type	Pedestrian AP@0.5	Vehicle AP@0.5	mAP@0.5
BoxMask+Constellation	90.1	98.4	94.3

Table 7. Per-class average precision and mean average precision for two models trained using the two different subsets of the training set: using (i) only data from 2019/2020, and (ii) only data from 2023, and similarly analyze their performance on the test set.

Training Dataset	Test Split	Pedestrian AP@0.5	Vehicle AP@0.5	mAP@0.5
2019/2020	All	76.7	95.8	86.3
	2019/2020	**74.4**	**99.0**	**86.7**
	2023	77.8	93.5	85.6
2023	All	**82.3**	**97.2**	**89.7**
	2019/2020	59.7	97.2	78.5
	2023	**88.1**	**97.5**	**92.8**

cle detection where a single detection error could be catastrophic (for example, safety warnings and navigation for disabled individuals), the AP values of the baseline models for pedestrians are insufficient for real-world deployment.

4.3. Image Augmentation

We assumed that scenes with varying background conditions would benefit from augmentation approaches which are different compared to classical object detection augmentations, which have to consider varying image quality and choose augmentations correspondingly. We explored custom image augmentation approaches tuned for the use case to improve model robustness on other out-of-distribution images.

The default set of augmentations for YOLOv8 and the set of augmentations that we propose are listed in Table 3. We use the Albumentations library to add the new augmentations [5]. RandomShadow creates random transparent patches of darkness in the input simulating shadow, which we postulate should help the model anticipate the sudden changes in brightness levels due to crisp shadows cast by buildings. MotionBlur adds random motion blur and helps simulate the camera blur for fast-moving entities or the possible effects of raindrops on the camera lens. Random rain adds an artificial rain effect on top of the scene and is intended to make the model robust to frame-specific corruption during such weather. We use these augmentations with the models trained for the different pretraining datasets to explore whether the addition of these augmentations could improve model performance.

Table 4 shows the results of various models on the Constellation dataset and Table 5 shows the effect of adding augmentations. We find a consistent minor improvement in AP scores for all models with the added augmentations,

which suggests that simulation of different weather conditions might enable the building of more accurate street-level object detectors. Our best-performing models use the VisDrone dataset for pretraining, resulting in 4.6% improvement in pedestrian AP. Importantly, for vehicles that are larger objects, the pretraining has very little effect, corresponding to only a improvement of 0.1% AP over the COCO-pretrained model. CARLA and SDD datasets underperform compared to VisDrone as pretraining datasets.

4.3.1 Model Scaling

To examine the trends in model scaling, we plot mAP@0.5 per epoch for training with YOLOv8x vs. YOLOv8n models in Figure 4. We observe the same trends and consistent relative ordering of the performance curves in both plots, implying that model comparison can be conducted with smaller, faster-to-train networks. Moreover, we find only a small gap in mAP between the two model architectures for the best-performing models, corresponding to a difference in mAP of 1.1%. As intersection monitoring will benefit from edge computing devices deployed next to cameras to reduce bandwidth and preserve anonymity, this small difference shows the suitability of object detection models in edge deployments.

4.3.2 Semi-Supervised Object Detection

To improve the performance of the models, we deployed the YOLOv8x model trained on Constellation in inference mode for real-time data tracking and bounding box collection as an augmentation approach. Using the tracking-by-detection paradigm allowed the model predictions to avoid false positives and false negatives [3]. Another significant challenge when using such a model for automatic data collection is dealing with objects that have not been detected. To reduce this false negative rate in data collected in this manner, we masked all areas of the image beyond the bounding box bounds with a median image calculated over the last 90 seconds of the video stream.

The unlabeled data for semi-supervision was collected over 4 days, with one frame being saved every 20 seconds. We refer to this dataset as BoxMask. We use this dataset for a 125-epoch pretraining step, followed by 125 epochs of training on human-annotated Constellation images. The results are presented in Table 6. We find that this dataset achieves a similar performance to VisDrone, suggesting a pretraining stage with more diverse pseudo-labeled data could be used as a replacement for standard pretraining datasets.

4.4. Performance Drift

We explored the effect of removing portions of the training data depending on the year of acquisition from the training set, to look at how changes in street-level conditions affect models over time. We ran two experiments (i) using the data from 2019/2020 only, (ii) using the data from 2023 only. The same split is used for the test set. YOLOv8x models are trained for 125 epochs on both splits and test each model on both splits. We summarize the experiments in Table 7. The models trained with split (ii) perform significantly better in the corresponding test set split. We thus postulate that for deploying object detection models, regularly updated data collection and refinement need to be a standard procedure. Future research could focus on minimizing the performance difference between splits (i) and (ii) through novel training or augmentation strategies.

5. Ethical Considerations

The Constellation dataset is collected using a camera deliberately installed at an altitude that makes it impossible to discern the pedestrian faces or to read car license plates. The academic institution at whose facilities the camera is installed provided an IRB waiver for sharing this "high elevation data" with the general public since the data inherently preserves privacy.

6. Conclusion

We created and described the Constellation dataset which consists of $13K$ images of traffic intersection scenes acquired by a high-elevation camera deployed in a metropolis. The purpose of the dataset is to support research on high-altitude object detection in dense urban environments, which is a key component for AI-based smart city applications. The analysis of the performance of object detection models using this dataset highlighted the difficulties in detecting small pedestrians in traffic intersections with changing backgrounds, lighting, building shadows, and weather conditions. The experiments with pretraining on similar datasets show that strategic data augmentation combined with pretraining can improve detection accuracy. The best models yield promising results, with the top-performing model achieving a pedestrian detection AP of 92.0% and an overall mean average precision (mAP) of 95.4%. We note that there remains a large gap between pedestrian and vehicle detection performance, which needs to be closed for the deployment of safety-critical intersection monitoring models for analytics and tracking applications. We anticipate that Constellation will help advance object detection methods for high-altitude pedestrian and vehicle monitoring for applications such as safety warning generation and collection of traffic analytics. By releasing the dataset along with the trained baseline models and the corresponding codebase we aim to facilitate future research and applications in urban safety applications.

Dataset Availability

We make the code and datasets available at https://github.com/zk2172-columbia/constellation-dataset.

Acknowledgements

This work was supported in part by NSF grants CNS-1827923, OAC-2029295, CNS-2148128, EEC-2133516, NSF grant CNS-2038984 and corresponding support from the Federal Highway Administration (FHA), ARO grants W911NF2210031 and W911NF1910379.

References

[1] Fatih Cagatay Akyon, Sinan Onur Altinuc, and Alptekin Temizel. Slicing aided hyper inference and fine-tuning for small object detection. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 966–970. IEEE, 2022. 3, 5

[2] Laura Belli, Antonio Cilfone, Luca Davoli, Gianluigi Ferrari, Paolo Adorni, Francesco Di Nocera, Alessandro Dall'Olio, Cristina Pellegrini, Marco Mordacci, and Enzo Bertolotti. IoT-enabled smart sustainable cities: Challenges and approaches. *Smart Cities*, 3(3):1039–1071, 2020. 1

[3] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468. IEEE, 2016. 8

[4] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. YOLOv4: Optimal speed and accuracy of object detection. *arXiv preprint:2004.10934*, 2020. 3

[5] Alexander Buslaev, Vladimir I Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A Kalinin. Albumentations: fast and flexible image augmentations. *Information*, 11(2):125, 2020. 7

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 3

[7] Guang Chen, Haitao Wang, Kai Chen, Zhijun Li, Zida Song, Yinlong Liu, Wenkai Chen, and Alois Knoll. A survey of the four pillars for small object detection: Multiscale representation, contextual information, super-resolution, and region proposal. *IEEE Transactions on systems, man, and cybernetics: systems*, 52(2):936–953, 2020. 3

[8] Shanzhi Chen, Jinling Hu, Yan Shi, Ying Peng, Jiayi Fang, Rui Zhao, and Li Zhao. Vehicle-to-Everything (v2x) services supported by LTE-based systems and 5G. *IEEE Communications Standards Magazine*, 1(2):70–76, 2017. 1

[9] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017. 6

[10] Steve Galgano, Mohamad Talas, Keir Opie, Michael Marsico, Andrew Weeks, Yixin Wang, David Benevelli, Robert Rausch, Kaan Ozbay, Satya Muthuswamy, et al. Connected vehicle pilot deployment program phase 1: Performance measurement and evaluation support plan: New york city. Technical report, United States Department of Transportation, 2021. 1

[11] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 2

[12] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics YOLOv8, 2023. 3, 5, 6

[13] Abhishek Kadian, Joanne Truong, Aaron Gokaslan, Alexander Clegg, Erik Wijmans, Stefan Lee, Manolis Savva, Sonia Chernova, and Dhruv Batra. Sim2real predictivity: Does evaluation in simulation predict real-world performance? *IEEE Robotics and Automation Letters*, 5(4):6670–6677, 2020. 6

[14] Vijay Gopal Kovvali, Vassili Alexiadis, and Lin Zhang PE. Video-based vehicle trajectory data collection. Technical report, 2007. 2

[15] Robert Krajewski, Julian Bock, Laurent Kloeker, and Lutz Eckstein. The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems. In *2018 21st international conference on intelligent transportation systems (ITSC)*, pages 2118–2125. IEEE, 2018.

[16] Robert Krajewski, Tobias Moers, Julian Bock, Lennart Vater, and Lutz Eckstein. The round dataset: A drone dataset of road user trajectories at roundabouts in germany. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6. IEEE, 2020. 2

[17] Huaqing Lai, Liangyan Chen, Weihua Liu, Zi Yan, and Sheng Ye. STC-YOLO: small object detection network for traffic signs in complex environments. *Sensors*, 23(11):5307, 2023. 3

[18] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, page 896. Atlanta, 2013. 5

[19] Yuxuan Li, Qibin Hou, Zhaohui Zheng, Ming-Ming Cheng, Jian Yang, and Xiang Li. Large selective kernel network for remote sensing object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16794–16805, 2023. 3, 6

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 3

[21] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 5

[22] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *Proc. ECCV*, 2016. 3

[23] Wenyu Lv, Shangliang Xu, Yian Zhao, Guanzhong Wang, Jinman Wei, Cheng Cui, Yuning Du, Qingqing Dang, and Yi Liu. Detrs beat yolos on real-time object detection, 2023. 2, 3, 5, 6

[24] Chengqi Lyu, Wenwei Zhang, Haian Huang, Yue Zhou, Yudong Wang, Yanyi Liu, Shilong Zhang, and Kai Chen. Rtmdet: An empirical study of designing real-time object detectors. *arXiv preprint arXiv:2212.07784*, 2022. 3, 6

[25] Sangmin Oh, Anthony Hoogs, Amitha Perera, Naresh Cuntoor, Chia-Chih Chen, Jong Taek Lee, Saurajit Mukherjee, JK Aggarwal, Hyungtae Lee, Larry Davis, et al. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR 2011*, pages 3153–3160. IEEE, 2011. 2

[26] Dipankar Raychaudhuri, Ivan Seskar, Gil Zussman, Thanasis Korakis, Dan Kilper, Tingjun Chen, Jakub Kolodziejski, Michael Sherman, Zoran Kostic, Xiaoxiong Gu, et al. Challenge: Cosmos: A city-scale programmable testbed for experimentation with advanced wireless. In *Proc. ACM MobiCom*, 2020. 2

[27] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 3

[28] Joseph Redmon and Ali Farhadi. YOLOv3: An incremental improvement. *arXiv preprint:1804.02767*, 2018.

[29] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proc. CVPR*, 2016. 3

[30] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory prediction in crowded scenes. In *European Conference on Computer Vision (ECCV)*, page 5, 2016. 2, 6

[31] Luis Sanchez, Luis Muñoz, Jose Antonio Galache, Pablo Sotres, Juan R Santana, Veronica Gutierrez, Rajiv Ramdhany, Alex Gluhak, Srdjan Krco, Evangelos Theodoridis, et al. Smartsantander: IoT experimentation over a smart city testbed. *Computer networks*, 61:217–238, 2014. 1

[32] Jacob Shermeyer and Adam Van Etten. The effects of super-resolution on object detection performance in satellite imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 3

[33] Bing Shuai, Alessandro Bergamo, Uta Buechler, Andrew Berneshawi, Alyssa Boden, and Joe Tighe. Large scale real-world multi person tracking. In *European Conference on Computer Vision*. Springer, 2022. 2

[34] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 2

[35] Juan Terven and Diana Cordova-Esparza. A comprehensive review of yolo: From yolov1 to yolov8 and beyond. *arXiv preprint arXiv:2304.00501*, 2023. 2

[36] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696*, 2022. 3, 5

[37] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7464–7475, 2023. 6

[38] Jinwang Wang, Wen Yang, Haowen Guo, Ruixiang Zhang, and Gui-Song Xia. Tiny object detection in aerial images. In *2020 25th international conference on pattern recognition (ICPR)*, pages 3791–3798. IEEE, 2021. 3

[39] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-ESRGAN: Training real-world blind super-resolution with pure synthetic data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1905–1914, 2021. 5

[40] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3974–3983, 2018. 2, 3

[41] Chang Xu, Jinwang Wang, Wen Yang, and Lei Yu. Dot distance for tiny object detection in aerial images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1192–1201, 2021. 3

[42] Shangliang Xu, Xinxin Wang, Wenyu Lv, Qinyao Chang, Cheng Cui, Kaipeng Deng, Guanzhong Wang, Qingqing Dang, Shengyu Wei, Yuning Du, et al. Pp-yoloe: An evolved version of yolo. *arXiv preprint arXiv:2203.16250*, 2022. 3, 6

[43] Dongfang Yang, Linhui Li, Keith Redmill, and Ümit Özgüner. Top-view trajectories: A pedestrian dataset of vehicle-crowd interaction from controlled experiments and crowded campus. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 899–904. IEEE, 2019. 2

[44] Hongtian Yu, Yunjie Tian, Qixiang Ye, and Yunfan Liu. Spatial transform decoupling for oriented object detection. *arXiv preprint arXiv:2308.10561*, 2023. 3, 6

[45] Xiang Yuan, Gong Cheng, Kebing Yan, Qinghua Zeng, and Junwei Han. Small object detection via coarse-to-fine proposal generation and imitation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6317–6327, 2023. 2, 5, 6

[46] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. In *European Conference on Computer Vision*, pages 1–21. Springer, 2022. 5

[47] Pengfei Zhu, Longyin Wen, Dawei Du, Xiao Bian, Haibin Ling, Qinghua Hu, Qinqin Nie, Hao Cheng, Chenfeng Liu, Xiaoyu Liu, et al. Visdrone-det2018: The vision meets drone object detection in image challenge results. In *Proc. ECCV Workshops*, 2018. 3, 6