Smart City Traffic Intersection: Impact of Video Quality and Scene Complexity on Precision and Inference

Zhuoxu Duan, Zhengye Yang, Richard Samoilenko, Dwiref Snehal Oza, Ashvin Jagadeesan,
Mingfei Sun, Hongzhe Ye, Zihao Xiong, Gil Zussman, Zoran Kostic
Dept. of Electrical Engineering, Columbia University, New York City
{zd2235, zy2318, rs4094, dso2119, ms5898, aj2929, hy2610, zx2273, gz2136, zk2172@}@columbia.edu

***Abstract*—Traffic intersections are prime locations for deployment of infrastructure sensors and edge computing nodes to realize the vision of a smart city. It is expected that the needs of a smart city, in regards to traffic and pedestrian traffic systems monitored by cameras/video, can be met by using state-of-the-art artificial-intelligence (AI) based object detectors and trackers. A critical component in designing an effective real-time object detection/tracking pipeline is the understanding of how object density, i.e., the number of objects in a scene, and image-resolution and frame rate influence the performance metrics. This study explores the accuracy and speed metrics with the goal of supporting pipelines that meet the precision and latency needs of a real-time environment. We examine the impact of varying image-resolution, frame rate and object-density on the object detection performance metrics. The experiments on the COSMOS testbed dataset show that varying the frame width from 416 pixels to 832 pixels, and cropping the images to a square resolution, result in the increase in average precision for all object classes. Decreasing the frame rate from 15 fps to 5 fps preserves more than 90% of the highest F1 score achieved for all object classes. The results inform the choice of video preprocessing stages, modifications to established AI-based object detection/tracking methods, and suggest optimal hyper-parameter values.**

***Index Terms*—Object Detection, Smart City, Video Resolution, Deep Learning Models.**

I. Introduction

Urban environments pose significant challenges towards realization of smart city intersections, owing to the constant and dense flow of vehicular and pedestrian traffic in constrained spaces. Deployment of traffic/pedestrian tracking systems is complicated by strict design requirements, exemplified by the need to maximize detection accuracies and to achieve close to real-time detection. Automation of smart intersections requires the use of artificial intelligence (AI), which needs large quantity of high quality data to train object detection and tracking models. To that effect, either hand-crafting a robust, fast and reliable model, or choosing an off-the-shelf neural network architecture is only one of many decisions in designing smart traffic intersection monitoring system. Video streams from surveillance cameras which monitor traffic intersections are the primary source of data. Hardware specifications and video characteristics of the cameras dictate the design and functioning of a model. Video resolution, aspect ratio and video frame rate, to name a few of the critical variables, are the basic characteristics of the data, and their inherent biases will directly influence the model performance. More abstract characteristics such as the frequency and per-frame-density of objects (henceforth referred to as "object density") will also have an effect on the speed of inference.

Traffic intersections are prime locations for deploying smart-city sensors, communications and edge computing nodes. The study described in this paper relies on video recordings of a smart city intersection which is part of the Cloud Enhanced Open Software Defined Mobile Wireless Testbed for City-Scale Deployment (COSMOS) [1], [2], located in New York City (NYC) as part of the NSF PAWR program. The study uses cameras mounted high above the COSMOS pilot intersection at 120th Street and Amsterdam Avenue, as seen in Fig. 1 and Fig. 2. The bird's-eye cameras are mounted on the 12th floor of the Mudd building at Columbia University, therefore minimizing object occlusion (due to a near-vertical perspective) and avoiding the issues of privacy. The wide-angle cameras also provide a contextual view of the entire intersection. With highly elevated positions of the bird's eye cameras, the performance of object detection on small objects can be notably degraded - in an image with 1920×1080 resolution, the size of a pedestrian is smaller than 30×30 pixels. Higher resolution video input, with aspect ratio matched to the scene under consideration, is intuitively expected to produce notably better object detection accuracy, but also to significantly increase the inference time.

To explore the changes in detection performance as a function of input resolution and aspect ratio, and to understand the changes in inference speed with different object densities, we conducted 3 investigations using YOLOv4 [3] as the base object-detection model:

- The detection performance as a function of different input resolutions and aspect ratios.
- The inference time of the deep-learning based video processing pipeline as a function of varying object density.
- F1 score study to evaluate the quality of object detection under various input resolutions and frame rates.

This work was supported in part by NSF grants CNS-1827923, OAC-2029295, and CNS-2038984, an NSF-BSF grant CNS-1910757, and an AT&T VURI award.

Fig. 1: Four cameras deployed at different heights on the Mudd building at Columbia University, COSMOS pilot site - 120th St. and Amsterdam Ave, NYC.

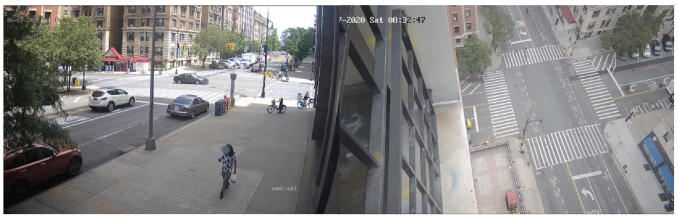

Fig. 2: Left: camera view from the 1st floor of the Mudd building; Right: bird's eye view from the 12th floor of the Mudd building.

An example of an application that can take advantage of the results in this paper is the investigation of social distancing behavior of pedestrians during pandemics (e.g., [4]).

II. Related Work

This section provides a review of the existing research in object detection involving multiple input resolution models, the impact of the density of objects in video frames, and other relevant video analytics research.

Speed/Accuracy Trade-offs: Huang et al. explored speed/accuracy trade-off for "real-time" object detectors [5], focusing on Faster R-CNN, R-FCN, and SSD systems. YOLOv4, which has a significant edge in inference speed, was not evaluated.

Multiple Input Resolutions: Peng et al. explored varying model performance metrics due to decreasing input resolution as a result of down-sampling [6]. The authors found that the best balance between detection accuracy, detection speed, and file size (upon image compression) was at 8 times downsampling, and that file compression does not necessarily have an adverse effect on the overall accuracy. These conclusions do not generalize to unavoidable problems related to small object detection.

Density of Objects: In busy urban environments, the density (number) of objects in a scene is highly variable, and is a function of the time of the day and the type of object (pedestrian or vehicle). This variability results in the imbalance problems as defined by Oksuz et al. [7]. Class imbalance is the most commonly encountered problem in object detection, where disparity in the numbers of input bounding boxes pertaining to different classes occurs. Scale imbalance arises when objects present in different scales along with different number of examples for each respective scale. This imbalance imposes a limitation on how effectively the model learns to identify examples of each class when their relative scale varies throughout the data. While the implications of class and scale imbalance are easily experienced through training, the degree of impact on inference time are of particular interest in our analysis. Similar explorations utilizing CUDA profiling are not found in recent literature.

Video Analytics for Object Detection: Video analytics are often combined with classical digital image processing methods as well as new deep learning based techniques. In order to achieve comparable performances while reducing the cost of resources, the choice of video configuration (e.g., resolution and frame rate) should be set up properly. Configurations that meet the accuracy threshold can often vary by many orders of magnitude in their resource demands [5]. For example, the higher the resolution, the more pixels are involved in the representation of individual objects, and the need for computation scales up in neural network models. The best configuration can also vary over time. Many video analytics systems choose their overall optimal configurations by conducting profiling at the beginning of a video, while Chameleon system [8] keeps up with the intrinsic dynamics in videos by periodically profiling to find an optimal resource-accuracy trade-off. Chameleon leverages domain-specific insights on the temporal and spatial correlations of these configurations to realize its frequent changes. This work inspired us to evaluate our object detectors at different video settings. Due to potential communication bandwidth restriction in city deployments, the results of our study can be used for model selection and configuration optimization in the traffic intersection applications.

III. Methodology

This section describes the methodology undertaken for multiple resolution, object density, and video analytics studies. We discuss the custom COSMOS smart-intersection dataset, and the modified YOLOv4 object detector used to classify vehicles and pedestrians in the intersection.

A. System Architecture

Testbed: The study has been performed using video recordings captured by the 12th floor bird's-eye cameras (Fig. 1) integrated into the pilot node of the COSMOS testbed [1], located at the intersection of Amsterdam Avenue and 120th street in Manhattan, NYC. The beyond-5G COSMOS testbed integrates (i) low-latency high-bandwidth wireless technologies such as millimeter-wave (mmWave) [9]; (ii) optical x-haul communications [10]; (iii) AI-enabled edge cloud computing servers; and (iv) sensors such as cameras [11].

Data Acquisition: Highly elevated cameras under consideration in this study are used with the goal of privacy preservation, which would be compromised by cameras located at lower

floors that can be used to recognize pedestrian faces and vehicle licence plates. Bird's eye cameras also reduce the problem of object occlusions while providing a wide-angle contextual view into the whole intersection, as shown in Fig. 2. The videos are recorded in 1920×1080 resolution at 15 frames per second and processed into a calibrated video with a 90 degree vertical bird's-eye view. Whereas privacy preservation is guaranteed, the small size of objects is a serious challenge for object detection algorithms which have to provide sufficient detection accuracy for safety-critical traffic applications.

Data Preprocessing: To get true "bird's eye" videos that are perpendicular to the ground, video calibration is applied. It transforms the distorted traffic intersection scene into a rectangle with a uniform size. This is achieved by calculating the homography matrix which maps the coordinates of raw videos into the real-world coordinates. Image cropping is implemented in the next step - the removal of irrelevant image parts increases the per-pixel size of information-carrying features, and improves the detection performance.

Object Detection: YOLOv4 [3] is a state-of-the-art one-stage object detector, which simultaneously resolves region proposals and executes the classification. To improve the detection of pedestrians and vehicles, we customized YOLOv4 in the following ways: (i) we extracted shallower feature maps with small receptive fields to detect pedestrians of small sizes; (ii) we implemented anchor box re-clustering to select proper scale factors of anchor boxes during the training; and (iii) we applied transfer learning using the following relevant datasets: ImageNet dataset [12], VisDrone2019 dataset [13], and our COSMOS traffic dataset.

B. Varying Input Resolution

Video resolution is an important parameter to consider when trying to optimize the performance of the object detection model. A significant problem that presents itself, both during training and inference, is the inability of the model to detect pedestrians with a precision as good as the precision achieved in the detection of vehicles. The intuitive explanation is that pedestrians are much smaller than vehicles, especially when considering that bird's-eye videos are taken from very high elevation. By using larger input resolutions, the size of the pedestrians can be up-scaled in absolute pixel size, leading to better results in the detection.

The backbone of the YOLOv4 architecture takes square images as input to the model, and therefore YOLO model contains a built-in preprocessing block to scale arbitrary image inputs to a resolution whose width and height is a multiple of 32. Given that the cameras used in our experiments record in 1920×1080 native resolution, this study seeks to find the optimal preprocessing to down-sample and crop the input video frames while maximizing the average precision among the vehicle and pedestrian class. The dataset of COSMOS intersection videos was processed and partitioned into : (i) the original dataset of 1920×1080 videos with $16 : 9$ aspect ratio ($16 : 9$ native frames); and (ii) cropped 832×832 dataset (square-cropped frames) which focuses on the center of the intersection. From these two datasets, we are able to specify the desired resolution as a parameter in the YOLOv4 model for resizing.

TABLE I: Models for Input Resolution Study

Model ID	Dataset	Width	Height
a1	1920×1080	832	480
a2	1920×1080	608	352
a3	1920×1080	416	256
b1	1920×1080	832	832
b2	1920×1080	608	608
b3	1920×1080	416	416
c1	832×832	832	832
c2	832×832	608	608
c3	832×832	416	416

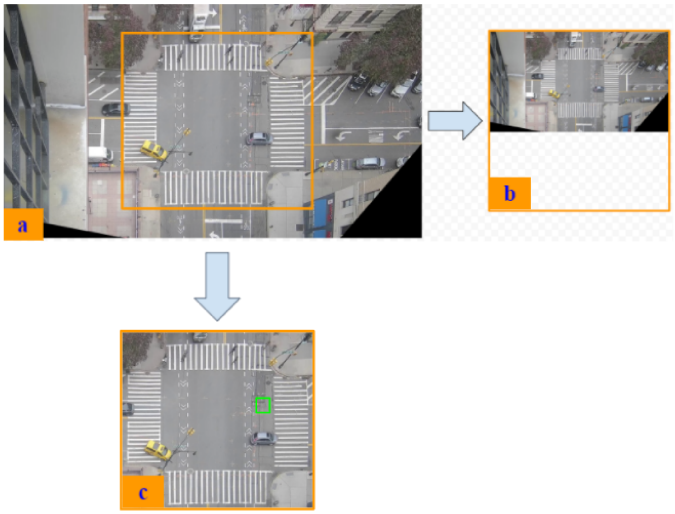

Fig. 3: (a) Preprocessed $16 : 9$ native frame; (b) $16 : 9$ frame squared by zero-padding; (c) Square-cropped frame of the relevant part of traffic intersection only

The study experiments with three models for each resolution of each unique cropping that can be done from both datasets, as summarized in Table I. The three considered frame widths for the input frames are $832, 608$, and 416 pixels. The first three models are created from the original 1920×1080 frames without any preprocessing. The second set of three models uses a downsampled version of the original 1920×1080 frames keeping the original aspect ratio while applying zero padding to generate the square frames. The last set of three models uses the input frames which were downsampled directly from the 832×832 dataset, which is the cropped dataset generated by preprocessing the original $16 : 9$ frames without any (excess) information other than the activity within the traffic intersection. The illustration of the three types of frames is presented in Fig. 3.

C. Object Density Study

By varying the video resolution, we seek to identify trends from the best-performing to the worst-performing models, in terms of the amount of raw pixels that the model can learn

TABLE II: Sum of Objects Over All Frames in Videos

Camera/Video	Total Number of Objects
GoPro_1	26786
GoPro_2	21311
GoPro_3	19628
GoPro_4	19002
Hikvision_1	13935
GoPro_5	11999
Hikvision_2	8755
Hikvision_3	8688
Hikvision_4	6354
Hikvision_5	4452

from and use to infer vehicles and pedestrians. Unlike for multiple resolution studies, the effect of scale and class imbalance in object detection on the inference time and hardware resource use is not intuitively obvious.

We continued the analysis by examining if there is consistent relationship between the number of objects in an input video (object density) and the speed/computational efficiency of the model inference. Six models were chosen from the original nine models explored in the resolution study. These models consist of the set of three models trained on the square resolution frames, and the set of three models trained on the $16:9$ native frames. The names of the videos for this study can be seen in Table II. Half of the videos were recorded using a GoPro camera, while the other half were recorded with a Hikvision camera. The videos are arranged in order of decreasing total number of objects (density) across all frames. By doing so, the trends in the profiling metrics, such as total session time and kernel time, can be observed as we vary the object density in a linear fashion.

Code profiling is used to understand resource utilization of a given program, hardware resource consumption in terms of time and memory and, ultimately, to resolve performance bottlenecks and optimize models to execute faster. Deep learning frameworks like TensorFlow and PyTorch support profiling either through built-in libraries, or through an external profiler such as NVIDIA Visual Profiler (NVVP). Our YOLOV4 model is built using the CUDA/C-based Darknet framework. Darknet framework does not have a built-in profiling tool, and since it relies on CUDA, NVVP is used.

D. Video Analytics for Object Detection

High-quality videos are hard to obtain especially in real-time systems due to possibly limited transmission bandwidth and time delay caused by hardware. Therefore, examining how the detection performance changes as the video quality decreases is important for configuring the underlying communication systems/protocols.

To obtain videos of reduced quality, we down-sampled the original videos to get lower resolutions and lower effective frame rates. The lower resolution was achieved by YOLO's default resolution adjustment scheme. For our experiments we included videos of resolution 416×416, 608×608 and 832×832. The lower effective frame rates were achieved by freezing a frame (duplicating a frame) over the span of the consecutive frames. For purposes of illustration, let us define a video, such that it is a tuple whose entries are defined through an indexing multi-set. Then, the original video A can be defined as

$$A_{i \in I}, I = \{1, 2, 3, 4, 5, 6, 7, ...\}, \tag{1}$$

where each $i \in I$ is a unique entry, despite I being a multi-set.

A new video, called B, with half of the effective frame rate as the original, would have its frames defined as

$$B_{j \in J}, J = \{j^2 : j \in I, j \pmod 2 = 1\}. \tag{2}$$

In the given example, B would be defined as

$$B_{j \in J}, J = \{1, 1, 3, 3, 5, 5, ...15, 15\}. \tag{3}$$

A third video C, with one third of the effective frame rate as the original, would have its frames defined as

$$C_{k \in K}, K = \{k^3 : k \in I, j \pmod 3 = 1\}. \tag{4}$$

In the given example, C would be defined as

$$C_{k \in K}, K = \{1, 1, 1, 4, 4, 4, 7, 7, 7, ...13, 13\}. \tag{5}$$

We experimented with effective frame rates of 15, 7.5 and 5 frames per second.

The performance is evaluated in the following ways:

- We consider performance not only based on the evaluation against ground truth annotations, but also against the highest accuracy results. This allows us to monitor the performance drop-off for low-quality videos compared to the highest-quality videos.
- F1-score is used as the key metric as it is the harmonic mean of precision and recall, which are both important indicators of model performance.

IV. Results

A. Resolution Study

The goal of the resolution study was to understand the empirical impact of varying resolutions and aspect ratios on the performance. In the study, nine different models were evaluated. The three resolutions for each set of aspect ratios were 832, 608, and 416. The performance measure used to evaluate the models was the mean Average Precision (mAP) over three classes considered in each video: pedestrians, vehicles, and background.

As intuition might suggest, the models which were trained on higher resolution input frames resulted in higher mAP values compared to the other two classes. This finding is consistent for all three types of frames, and can be seen in Fig. 4. Considering the performance of models trained on 3 distinct types of frame cropping, it can be seen that the square-cropped frames outperform, in mAP measure, both $16:9$ native frames and $16:9$ squared with zero-padding frames, over all resolutions. Since the square-cropped frames focus only on the activity within the intersection, there is a smaller amount of irrelevant information being considered in the object detection. The other two types of input frame cropping still contain useless (and ultimately harmful) information such as

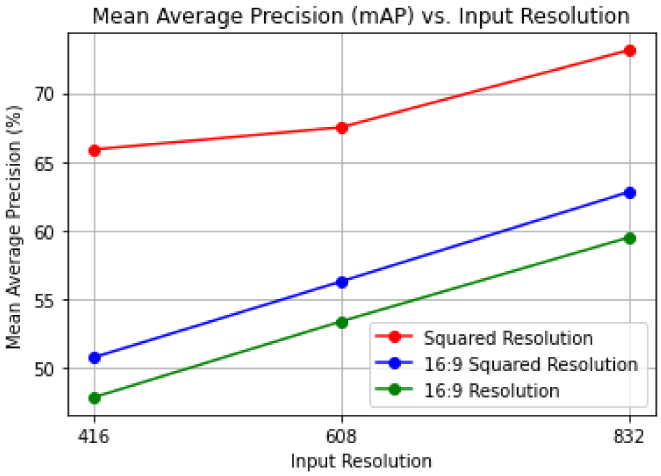

Fig. 4: Average Precision for both pedestrian and vehicle class: (red) Square cropped frame, (blue) 16 : 9 frame squared with zero-padding, (green) 16 : 9 native frame

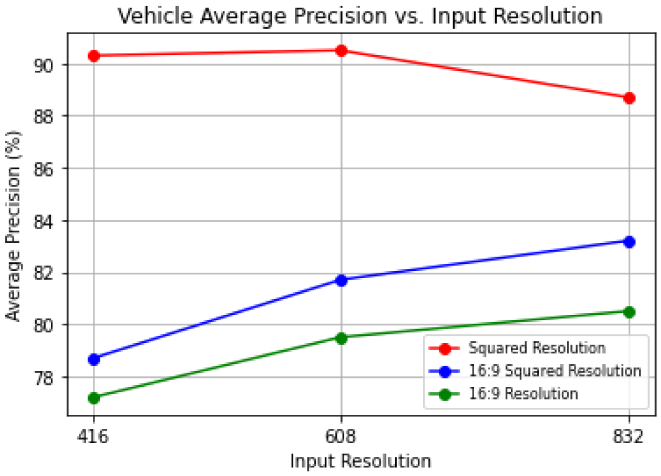

Fig. 5: Average Precision for the vehicle class: (red) Square cropped frame, (blue) 16 : 9 frame squared with zero-padding, (green) 16 : 9 native frame

the sides of streets and even irrelevant padding, as seen in Fig. 3 (a) and (b).

Considering the average precision (AP) of each object class separately, there is a profound improvement in performance when using the square-cropped input frames, for vehicle and pedestrian classes, illustrated by Figs. 5 and 6, respectively. The important observation is that vehicle AP is notably higher than pedestrian AP. By training the models on all three resolutions using the square-cropping, the AP achieved for the pedestrian class is at least 15% higher for all three resolutions, with a maximum increase of approximately 20% for the 416×416 frames. The other two types of input frames, the 16 : 9 native frames and the 16 : 9 squared with zero padding frames, serve as the baseline methods, show the improvement which the square-cropping method has on the detection of pedestrians, with the 16 : 9 squared with zero-padding input frames outperforming the native 16 : 9 frames. YOLOv4 takes input frames of a square size, meaning that there is built-in preprocessing into the model architecture. This may explain why there is worse performance on the native 16 : 9 frames given that the 16 : 9 squared frames have already been resized.

Given that the mAP is higher for the square input frames, it is also the case that the vehicle class benefited from the square input frame preprocessing. One thing to note is the marginal increase in precision over the input resolution. This is to be expected because of the native size of the vehicles when compared to the native size of pedestrians in a bird's eye viewpoint. An incremental increase in the size of a pedestrian will have a more profound impact on detection compared to the same change made on a vehicle.

B. Object Density Study

Four core metrics are used in the object density study:

1) Average Kernel Time - AKT
2) Average Total Session Time - ATST

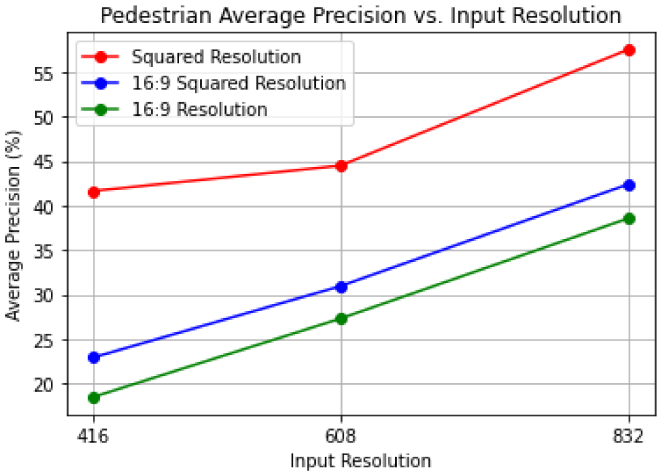

Fig. 6: Average Precision for the pedestrian class: (red) Square cropped frame, (blue) 16 : 9 frame squared with zero-padding, (green) 16 : 9 native frame

3) Average Utilisation - AU
4) Average Number of Kernel Invocations - ANKI

Three analyses were conducted in the object density study.

The first analysis involves obtaining all four metrics for each model. Plots of the metrics as a function of input resolution can be seen in Fig. 7, where red points denote square-cropped resolution models, and blue points denote 16 : 9 native resolution models. The analysis shows the linear relationship between AKT, ATST, AU and the input resolution. On average, AKT, ATST, and AU increase as a function of resolution for all models trained on both datasets. However, ANKI does not show a discernible trend as a function of resolution.

The second analysis is relevant to the density of objects within the videos used for inference for our six models. The videos filmed on GoPro cameras have a larger number of

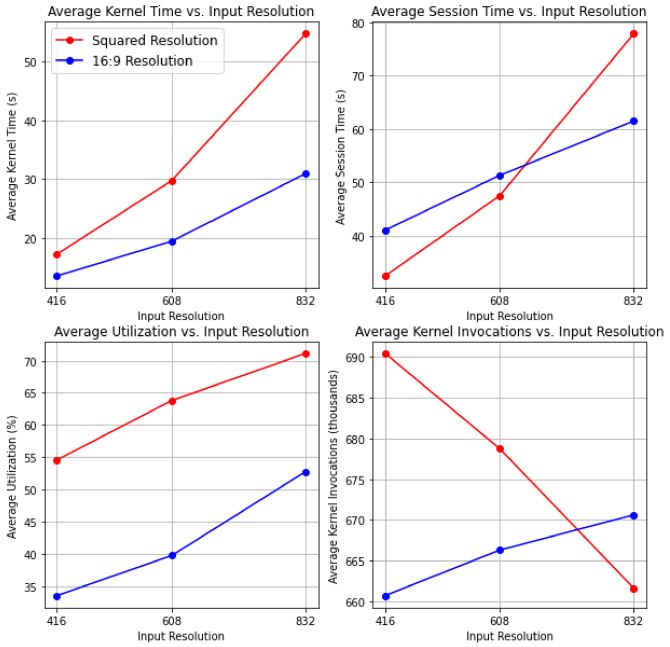

Fig. 7: Average kernel time, session time, utilization, and kernel invocations for models trained on square-cropped videos (red) and $16:9$ native videos (blue)

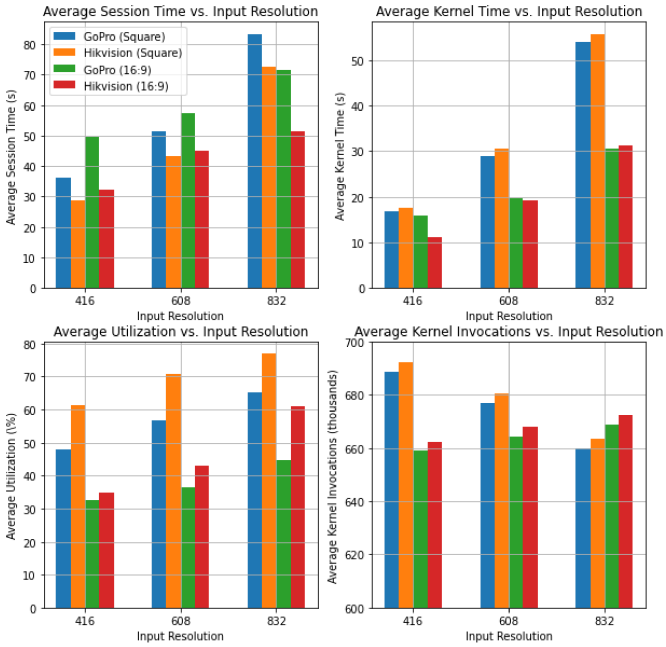

Fig. 8: Session time, kernel time, and utilization for each video (by its total number of frames) for models trained on (green and red) $16:9$ native, and (blue and orange) square-cropped frame datasets

total objects across all frames when compared to those of Hikvision cameras. Due to this fact, we were able to use the type of camera to observe how the density of objects influences the profiling metrics. Through pairwise comparison of bars in Fig. 8, we can determine the difference in metric values between the two cameras for square-cropped and $16:9$ native frame models. The pairwise results of ATST agree with the hypothesis of the density study; this may be due to the fact that session time is greater for cameras recording scenes with larger number of objects (GoPro) when compared to those recording fewer objects (Hikvision). The pairwise results for AU, however, show an increase in average utilization for the camera which is recording fewer objects (Hikvision) when compared to the camera recording more objects (GoPro). Pairwise comparison of the results for ATST, AKT and ANKI shows no obvious trend.

The final analysis considers each video used for inference individually, and categorizes them by the total number of objects found in all frames of the video. Plots were generated for both square-cropped and $16:9$ native frame models for ATST, AKT and AU in Fig. 9. ANKI was disregarded due to the lack of an obvious trend from the second analysis. In terms of trends for each metric, we see that ATST decreases as the total number of objects decreases. AKT, the kernel time, remains relatively unchanged over the total number of objects, except with some odd behavior for the 416×416 model with $16:9$ native frame inputs. Once again, AU proves to be a point of contention, since it increases as the number of objects decrease.

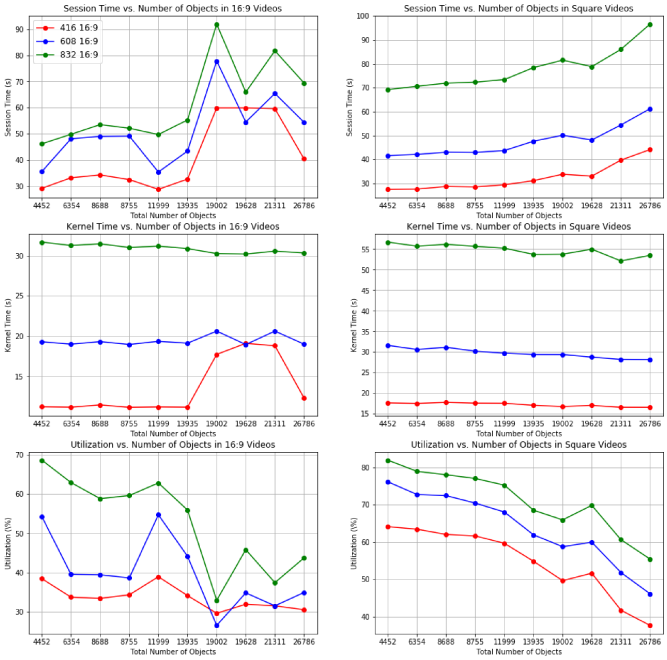

Fig. 9: Session time, kernel time, and utilization for each video (by its total number of objects) for models trained on $16:9$ native (left column), and square-cropped frame (right column) datasets

C. *Impact of Resolution and Frame Rate on F1 Score*

We use different combinations of input video resolutions and frame rates to evaluate the model. After running the

inference, F1 scores are calculated based on the ground truth annotations. We show the results in Fig. 10 for three different resolutions and three sampling frame rates. As expected, the F1 score decreases with reduction in resolution as well as frame rates. We note that there is a dramatic performance drop for pedestrians when resolution decreases from 832×832 to 608×608, while lower frame rates have smaller impact on the decreasing F1 score. This is reasonable since the objects (pedestrians and vehicles) in this traffic scenario (12th floor COSMOS dataset) are relatively small compared to other public traffic datasets, due to the elevation of camera. Lower resolution leads to higher loss of visual information which represents features of objects, whereas lower frame rates keep most of the information as objects move at a slow pace from the vantage point of the bird's eye camera. The impact of the resolution change onto F1 scores for vehicles is limited below 5%, while the accuracy increase from 5 fps to 7.5 fps looks more notable. Cars still preserve a reasonable size in the calibrated birds' eye view so that most of their features remain even in low-resolution videos such as 416×416 and 608×608.

The evaluation results in Fig. 10 show how F1 scores change when unfavorable communication conditions occur compared to the optimal performance. They inform how one should adjust the settings of video streams to minimize the decrease in model performance as much as possible, based on the current communication conditions such as bandwidth, delay, and jitter.

V. Conclusion

We presented several studies investigating the quality of detection of pedestrians and vehicles from videos acquired at a smart city traffic intersection, using modified YOLOv4 object detection architecture. We examined how the variation in image-resolution and object-density influence the mean average precision, average precision, average kernel execution time, average total session time, average utilisation, and average number of kernel invocations. YOLOv4 was used as the object detector due to its state-of-the-art performance, and the ability to modify the model to suit specific datasets. The videos were acquired from multiple bird's eye cameras located at the COSMOS smart-city intersection in NYC, and preprocessed to ensure maximal feature extraction and detection performance.

By varying the cropping method applied to the original videos and the input resolution used to train each model, we aimed to improve the limited accuracy in detection of pedestrians within the videos (as a consequence of their small size when viewed from a bird's eye camera). By varying the frame width from 416 pixels to 832 pixels, we observe the conclusive increase in average precision of the pedestrian class, as well as in the mean average precision among all classes, for all cropping methods used. The best results are observed for the square-cropped dataset, which focuses on the activity within the center of the intersection while cropping out the inactive portions of video frames.

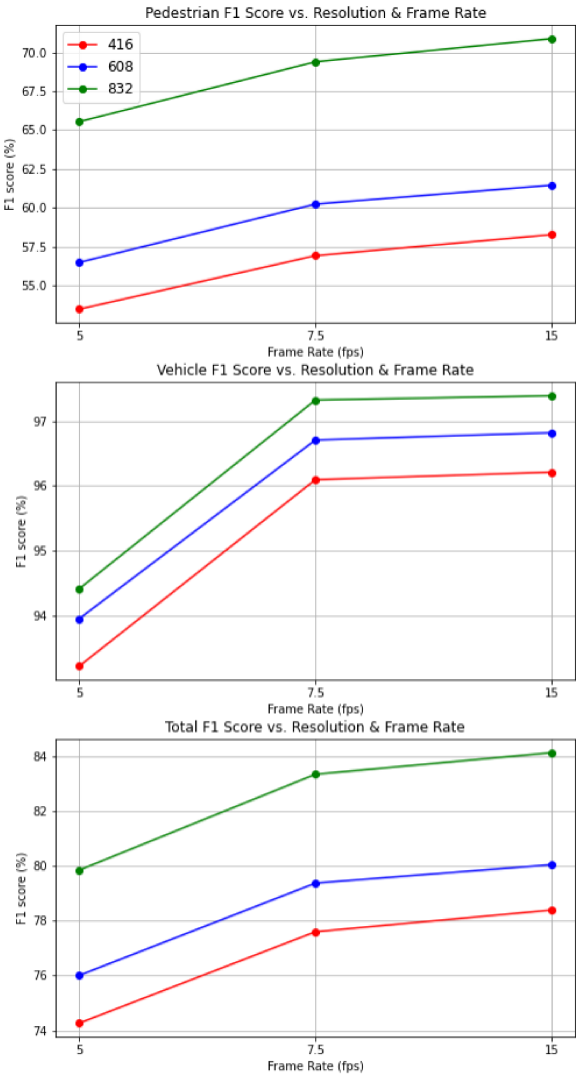

Fig. 10: F1 scores as function of frame rate and resolution, for pedestrians, vehicles and average for both classes, respectively.

We gained insight into the computational complexity of each of the nine models by profiling the inferences performed on ten test videos. By selecting videos which have the same length but a variable total number of objects across all frames, we explored the impact of the object density on the inference time, and other metrics generated by profiling. The results indicate that the session time decreases as the total number of objects decreases, which is to be expected. The kernel time remains relatively unchanged over the total number of objects, except for the outlier behavior for 416×256 resolution. The behavior of the utilization metric is a point of contention since it increases as the number of objects decreases.

According to the F1 score analysis, the best performance is achieved with the highest resolution and frame rate settings. In our case, that is 832×832 pixels and 15 frames per second. When the network conditions worsen, the video stream's first

choice is to reduce the frame rate to some value above 7.5 fps, as its impact on overall performance is limited. Further quality reductions would be determined by the trade-off between tracking performance and image quality preservation.

The results of the study can be used to inform the choice of system configuration, video preprocessing parameters, and YOLO hyper-parameters for video-based monitoring of smart city traffic intersections.

ACKNOWLEDGMENT

The authors thank Ujwal Dinesha, Emily Bailey, and Mahshid Ghasemi. The team is grateful for the support of Columbia School of Engineering, Columbia Data Sciences Institute, and Rutgers/WINLAB.

REFERENCES

[1] D. Raychaudhuri, I. Seskar, G. Zussman, T. Korakis, D. Kilper, T. Chen, J. Kolodziejski, M. Sherman, Z. Kostic, X. Gu, H. Krishnaswamy, S. Maheshwari, P. Skrimponis, and C. Gutterman, "Challenge: COSMOS: A city-scale programmable testbed for experimentation with advanced wireless," in *Proc. ACM MobiCom'20*, 2020.

[2] "Cloud enhanced open software defined mobile wireless testbed for city-scale deployment (COSMOS)." https://cosmos-lab.org/, 2021.

[3] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv e-prints*, p. arXiv:2004.10934, Apr. 2020.

[4] M. Ghasemi, Z. Kostic, G. Zussman, and J. Ghaderi, "Auto-sda: Automated video-based social distancing analyzer," in *3rd ACM Workshop on Hot Topics in Video Analytics and Intelligent Edges (HotEdgeVideo 2021)*, pp. 7–12, Oct 2021.

[5] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7310–7311, 2017.

[6] C. Peng, K. Zhao, A. Wiliem, T. Zhang, P. Hobson, A. Jennings, and B. C. Lovell, "To what extent does downsampling, compression, and data scarcity impact renal image analysis?," *CoRR*, vol. abs/1909.09945, 2019.

[7] K. Oksuz, B. C. Cam, S. Kalkan, and E. Akbas, "Imbalance problems in object detection: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, pp. 3388–3415, Oct 2021.

[8] J. Jiang, G. Ananthanarayanan, P. Bodík, S. Sen, and I. Stoica, "Chameleon: Scalable adaptation of video analytics," in *Proc. ACM SIGCOMM'18*, Aug 2018.

[9] X. Gu, A. Paidimarri, B. Sadhu, C. Baks, S. Lukashov, M. Yeck, Y. Kwark, T. Chen, G. Zussman, I. Seskar, and A. Valdes-Garcia, "Development of a compact 28-GHz software-defined phased array for a city-scale wireless research testbed," in *Proc. IEEE IMS'21*, 2021.

[10] J. Yu, C. Gutterman, A. Minakhmetov, M. Sherman, T. Chen, S. Zhu, G. Zussman, I. Seskar, and D. Kilper, "Dual use SDN controller for management and experimentation in a field deployed testbed," in *Proc. IEEE/OSA OFC'20, T3J.3*, 2020.

[11] S. Yang, E. Bailey, Z. Yang, J. Ostrometzky, G. Zussman, I. Seskar, and Z. Kostic, "Cosmos smart intersection: Edge compute and communications for bird's eye object tracking," in *Proc. 4th International Workshop on Smart Edge Computing and Networking (SmartEdge'20)*, 2020.

[12] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *arXiv e-prints*, p. arXiv:1409.0575, Sept. 2014.

[13] P. Zhu, L. Wen, D. Du, X. Bian, Q. Hu, and H. Ling, "Vision Meets Drones: Past, Present and Future," *arXiv e-prints*, p. arXiv:2001.06303, Jan. 2020.