Simple In-place Data Augmentation for Surveillance Object Detection

Munkh-Erdene Otgonbold[1,3] Ganzorig Batnasan[1] Munkhjargal Gochoo[1,2]

[1] Department of Computer Science and Software Engineering, United Arab Emirates University, UAE
[2] Emirates Center for Mobility Research, United Arab Emirates University, UAE
[3] Department of Electronics, Mongolian University of Science and Technology, Mongolia
omunkuush@uaeu.ac.ae, gbatnasan@uaeu.ac.ae, mgochoo@uaeu.ac.ae,

Abstract

Motivated by the need to improve model performance in traffic monitoring tasks with limited labeled samples, we propose a straightforward augmentation technique tailored for object detection datasets, specifically designed for stationary camera-based applications. Our approach focuses on placing objects in the same positions as the originals to ensure its effectiveness. By applying in-place augmentation on objects from the same camera input image, we address the challenge of overlapping with original and previously selected objects. Through extensive testing on two traffic monitoring datasets, we illustrate the efficacy of our augmentation strategy in improving model performance, particularly in scenarios with limited labeled samples and imbalanced class distributions. Notably, our method achieves comparable performance to models trained on the entire dataset while utilizing only 8.5 percent of the original data. Moreover, we report significant improvements, with mAP@.5 increasing from 0.4798 to 0.5025, and the mAP@.5:.95 rising from 0.29 to 0.3138 on the FishEye8K dataset. These results highlight the potential of our augmentation approach in enhancing object detection models for traffic monitoring applications.

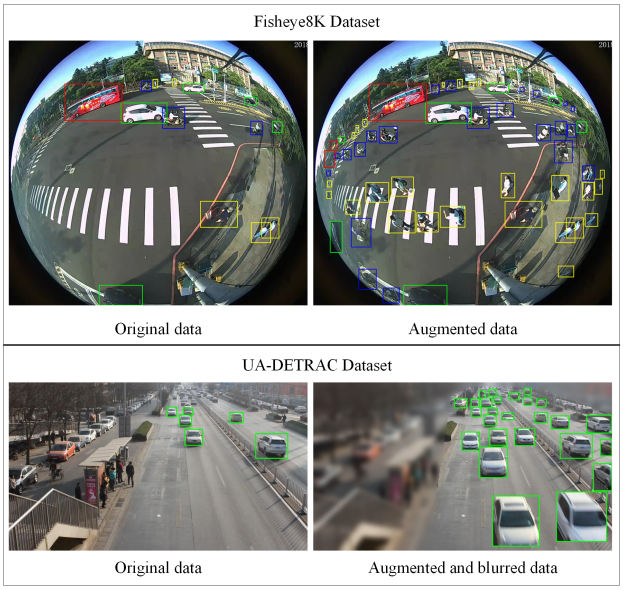

Figure 1. The comparison between the original sample images and augmented images of FishEye8K and UA-DETRAC datasets. Both augmented samples include a comparatively larger number of objects due to the in-place augmentation. In contrast, the UA-DETRAC sample has blurred areas, which are the regions of non-interest determined by subtracting the polygonal area of the bounding boxes of all the objects.

1. Introduction

Traffic cameras are extensively utilized for monitoring traffic conditions, particularly in areas surrounding intersections. Vision algorithms for cameras have been developed to automate a range of tasks, such as detecting and tracking vehicles and pedestrians [1], as well as re-identification [2]. Object detection, a fundamental task of computer vision, plays a crucial role in understanding visual scenes. This process includes pinpointing the location of the objects and outlining precise bounding boxes around them. Essentially, it enables the computer to recognize and categorize different items present in a given image. The field of computer vision has experienced significant advancements across different areas, extending beyond traffic monitoring. Applications like face recognition [3], robotic grasping [4], and human interaction [5] have all benefited from these developments, largely driven by the rapid progress of deep convolutional neural networks (CNNs) [6], [7], [8], [9]. While CNN architectures have shown exceptional performance, their effectiveness heavily relies on the availability of extensive sets of accurately labeled training data. For object detec-

tion models, the need for data augmentation is more crucial as collecting labeled data for detection is more costly and common detection datasets have many fewer examples than image classification datasets. A more advanced method for data augmentation involves utilizing segmentation annotations, which can be acquired either through manual efforts or generated by an automated segmentation system. This technique involves generating new images by positioning objects at different locations within pre-existing scenes [10], [11], [12]. Though it doesn't attain flawless photorealism, the approach of employing random placements has demonstrated unexpected effectiveness for object instance detection [10], which is a fine grained detection. In contrast, object detection concentrates on identifying instances of objects from a specified category. By placing training objects at unrealistic positions, implicitly modeling context becomes difficult and detection accuracy drops substantially.

Achieving balance in the number of instances per class within a traffic monitoring dataset presents a formidable challenge, primarily stemming from the extensive diversity observed across various regions. The dataset encompasses a wide range of geographical locations, each characterized by unique traffic patterns, infrastructure layouts, and behavioral norms. Road objects from specific classes tend to appear more in specific region (e.g., bikes in warm countries, less pedestrians in countries with extreme weather). Consequently, ensuring equitable representation of these diverse elements necessitates meticulous consideration and strategic planning. Balancing the classes becomes particularly intricate as disparities in traffic volume, road conditions, and cultural practices manifest differently across regions, complicating efforts to maintain proportional class distribution. Therefore, meticulous attention to detail and adaptive methodologies are imperative in addressing these inherent complexities and achieving a harmonious balance in the dataset's class distribution.

We propose to tackle the issue by applying in-place augmentation on the same position from the same camera. This increases the diversity in the locations of traffic objects while ensuring that those objects appear from correct angle. In Figure 1, we illustrate sample augmentation generated by our proposed method. The figure contrasts an original image without any augmentation with an augmented image sample, amplified by a factor of 20X, sourced from the Fisheye8K [13] and UA-DETRAC [14] datasets.

To enhance the object's impact on the model, we increased the number of objects within the image rather than augmenting the number of images. We selected and placed objects from the same camera input image that could be placed without overlapping with original and previously selected objects. We show with extensive tests on two traffic monitoring datasets that our augmentation approach can be used for improving model performance when few labeled samples are available.

2. Related Works

Zoph et al. [15] delve into the enhancement of generalization performance for detection models through the exploration of learned, specialized data augmentation policies. With meticulous curation, they assembled subsets of images from the COCO dataset [16], ranging in size from 5000 to 23000 images. The researchers observed a notable improvement in detection accuracy of more than +2.3 mAP across different ResNet backbones, resulting in mAP values ranging from 39.0 to 42.1. Furthermore, when applied to a distinct detection model featuring an AmoebaNet-D backbone [17], the method achieved a remarkable increase of +1.5% mAP, attaining a state-of-the-art accuracy of 50.7 mAP.

Ghiasi et al. [18] explores the efficacy of Copy-Paste augmentation for instance segmentation, revealing that random object pasting yields significant performance improvements over previous methods focusing on contextual modeling. Moreover, they demonstrate that integrating Copy-Paste with semi-supervised techniques that utilize additional data via pseudo-labeling, such as self-training, yields notable enhancements. Their approach achieves a mask AP of 49.1 and a box AP of 57.3 on COCO instance segmentation, surpassing the previous state-of-the-art by +0.6 mask AP and +1.5 box AP.

Dvornik et al. [19] proposed a data augmentation method consisting of two main steps: first, utilizing bounding box annotations to model visual context and train a CNN to predict object presence or absence; second, employing the trained context model to generate new object locations. Their approach, applied to a subset of the Pascal VOC'12 dataset [20], involved training a single multiple-category object detector with significantly more labeled data, resulting in a 1.3% average improvement over baseline across various categories

Cubuk et al. [21] propose a simplified approach to automated augmentation strategies, eliminating the need for a separate search phase. They apply their method across CIFAR-10/100, SVHN, ImageNet, and COCO datasets [16]. Notably, using EfficientNet-B7, they achieve a 1.0% increase in accuracy over baseline augmentation and a 0.6% improvement over AutoAugment on the ImageNet dataset.

Behpour et al. [22] offers a game-theoretic perspective on data augmentation in object detection, seeking optimal adversarial perturbations of ground truth data to enhance test-time performance. They demonstrate significant improvements of approximately 16%, 5%, and 2% respectively on the ImageNet[23], Pascal VOC [24], and MS-COCO [16] object detection tasks compared to leading data augmentation methods.

Kisantal et al. [25] tackle the performance gap in ob-

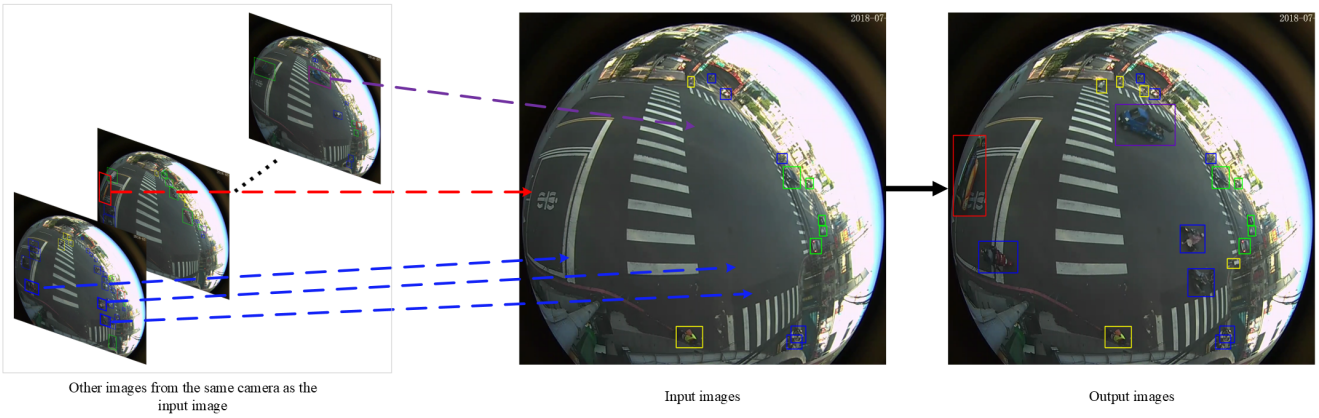

Figure 2. In-place object augmentation method on Fisheye8k dataset. An augmented output sample image has multiple objects that appear on the other frames of the same surveillance camera video.

ject detection between small and large objects by employing Mask-RCNN on the MS COCO dataset [16]. Their approach involves oversampling images containing small objects and augmenting them through the repeated copy-pasting of small objects. This method leads to a significant 9.7% relative enhancement in instance segmentation and a 7.1% improvement in object detection of small objects compared to the current state-of-the-art performance on MS COCO [16].

Shao et al. [26] address the limitations of SRe2L's "local-match-global" matching method by introducing "generalized matching" through G-VBSM. Their approach surpasses state-of-the-art methods by 3.9%, 6.5%, and 10.1% on CIFAR-100 [27], Tiny-ImageNet [28], and ImageNet-1k [23], respectively, demonstrating superior performance across small and large-scale datasets.

These methods have achieved favorable results through the integration of traditional augmentation methods with other techniques. While effective in enhancing results, the augmentation method poses significant computational demands due to the increased volume of data. However, these augmentation methods often struggle to preserve the realism of the images.

3. Datasets

The Fisheye8K [13] dataset contains 8000 images from 18 different cameras with 157K bounding boxes for five object classes. Of this, 5287 images of 14 cameras in the train set are divided into 2712 images of 4 cameras in the test set. We have selected 450 images, a small subset from the Fisheye8K [13] dataset's train set, which is only 8.5% of the full train set. Figure 3 shows a comparison bar graph of small, medium, and large objects in the Fisheye8K dataset's trainset and selected small set. The proportion of small, medium, and large objects in the 2 datasets is almost the same, and it is slightly different for the last 5th class.

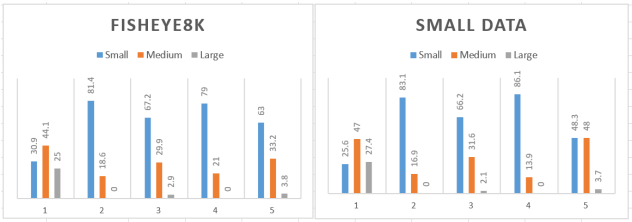

Figure 3. Number of objects of (a) Full dataset and (b) Sampled small dataset that is 8.5 percent of the full dataset.

UA-DETRAC [14] dataset comprises 100 videos which is divided into 83k images from 60 sequences for training set and 56k images from 40 sequences for testing set. We have selected 7140 images small set from UA DETRAC [14] dataset's train set. which is 8.5% of the full train set. As for 8.5%, after selecting a dataset that fully represents the Fisheye8K [13] dataset and achieving good results, the UA DETRAC [14] dataset also took a smaller portion from the larger dataset with the same percentage.

4. Augmentation

We introduce a stationary camera-based object augmentation method, where Figure 2 shows an example of our proposed augmentation process. When performing a data augmentation, to improve the effect of the object on the model, we augmented the number of objects in the image, not the number of images. On the contrary, we tried to reach and outperform the accuracy of the full dataset with as small as possible portion of the full set. In doing so, from all objects captured in the same camera video as the input image, those that can be positioned in the original location in the image without overlapping with others are chosen and placed accordingly. We opted for 3, 10, and 20 objects as values for

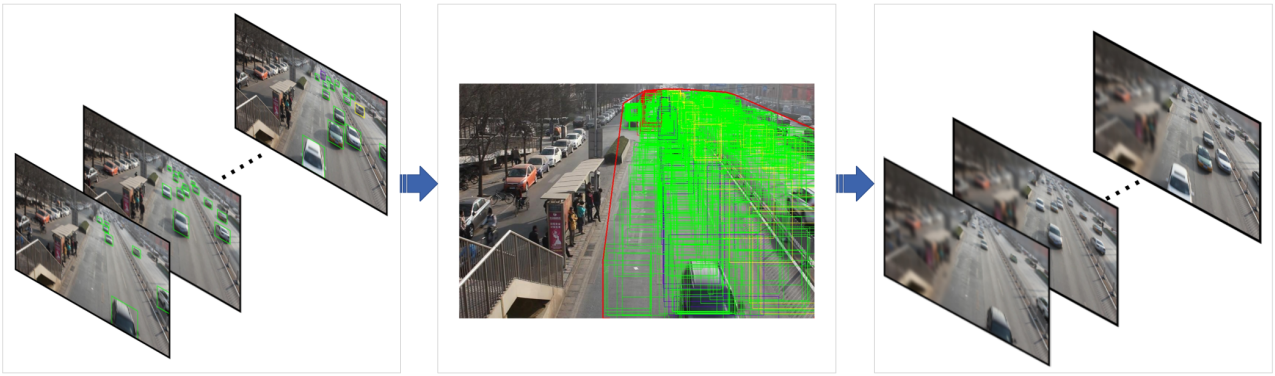

Figure 4. Determining the region of interest, the polygonal area drawn in red, from all object labels in the specific camera video.

augmentation. The model performances trained on the augmented data are presented in the results section under the names: "augmented 3X," "augmented 10X," "augmented 20X," and "Assembled". For Augmented 3X, 10X, and 20X, a selection of 3, 10, and 20 objects from each class, taken by the same camera as the input image, were placed onto the input image. These objects were then positioned onto the input image, ensuring they did not overlap with any previously placed objects. During the object selection process, images had to belong to either the day or night category, consistent with the input image. For Assembled, from the results of the above datasets: small data, augmented 3X, augmented 10X, augmented 20X, the objects of each class's objects with the best results were combined to form the data that gives the best results. Table 1 shows the increase in objects for each class in a small dataset from the Fisheye8K [13] dataset.

	Small Data	3X	10X	20X	Assembled
Bus	219	531	441	428	459
Bike	5060	6375	9296	12618	9296
Car	3720	5022	7298	8038	3720
Pedestrian	812	1765	3434	4298	812
Truck	325	851	773	692	325

Table 1. Number of original and augmented objects in a small dataset from the Fisheye8K dataset.

For the UA DETRAC [14] dataset, only objects in traffic are labeled and objects belonging to the same class standing on the side of the road are not labeled, which makes it difficult for any model to learn the data in that dataset. To solve this problem, we detect the active traffic areas of each camera and blurred the rest of the areas, it makes the data easier to learn for object detection. Figure 4 shows how to blur the image as an example. The augmentation in object counts for each class within a small set of the UA-DETRAC dataset [14] is presented in Table 2.

	Small Data	3X	10X	20X	Assembled
Others	321	3489	2450	2235	2449
Car	44256	65556	106479	129140	44256
Van	4756	20291	28117	25727	25071
Bus	3221	8656	7257	6814	6570

Table 2. Number of original and augmented objects in a small dataset from the UA-DETRAC dataset.

5. Results

We trained the YOLOv7-E6E [29] model by augmenting two extensive traffic datasets, Fisheye8K [13] and UA DETRAC [14], using our custom augmentation technique, resulting in multiple outcomes. Depending on the primary image size, we set the input size to 1280x1280 for the Fisheye8K [13] dataset and 640x640 for the UA-DETRAC [14] dataset. During all training procedures, a pre-trained model trained on the COCO dataset [16] was utilized, with both the IoU threshold and confidence threshold set to 0.5 during evaluation. For Fisheye8K [13] dataset's result, All evaluations were made on the original validation set. For UA-DETRAC [14] dataset's results, the model trained on small dataset was evaluated on the original validation set and other models trained on blurred datasets were evaluated on the blurred validation set.

5.1. Fisheye8K

Figure 5 shows the original and augmented samples in Fisheye8K dataset [13]. The results depicted in Table 3 illustrate the performance of the YOLOv7-E6E [29] model trained on a modest dataset without any augmentation. Notably, the model attained its peak mAP@.5 of 0.4267 in detecting cars, while registering its lowest mAP@.5 of 0.1449 in pedestrian detection. Notably, it showed great performance in detecting buses, with a mAP@.5 of 0.4137. Overall, the model demonstrated a mAP@.5 of 0.4798 and an F1-score of 0.62.

Small Data					
	Precision	Recall	mAP@.5	mAP@.5:.95	f1-score
Bus	0.9221	0.5602	0.5542	0.4137	0.697
Bike	0.7734	0.5142	0.4829	0.2344	0.6177
Car	0.8238	0.6756	0.6589	0.4267	0.7424
Pedestrian	0.7592	0.2993	0.2764	0.1449	0.4293
Truck	0.7872	0.5029	0.4265	0.2303	0.6137
All	0.8132	0.5104	0.4798	0.29	0.62

Table 3. Result of YOLOv7-e6e model on the small dataset.

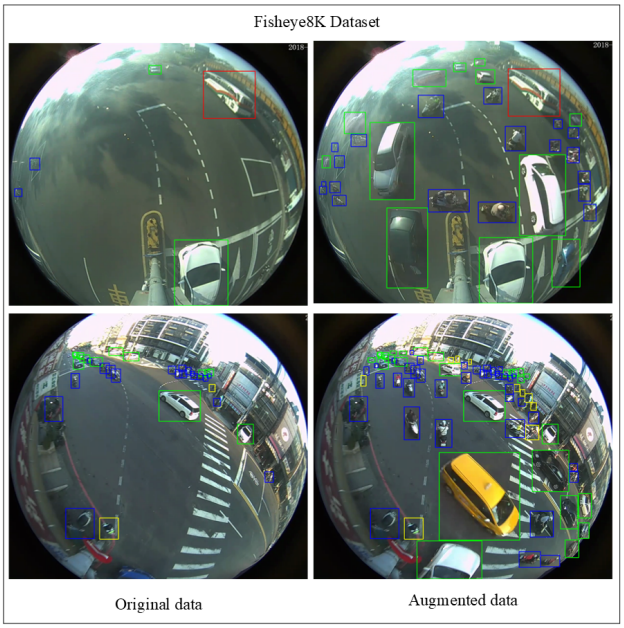

Figure 5. The comparison between original image and augmented image in Fisheye8K dataset.

The data provided in Table 4 indicates that the model excelled in detecting buses, achieving a mAP@.5 of 0.4811 and F1-score of 0.7249. Conversely, its performance was least effective in detecting pedestrians, with a mAP@.5 of 0.1224 and an F1-score of 0.3667. Additionally, the model demonstrated great performance in detecting cars, achieving a mAP@.5 of 0.4069 and an F1-score of 0.7212. Overall, the model yielded a mAP@.5 of 0.3054 and an F1-score of 0.5859.

Small Data 3X					
	Precision	Recall	mAP@.5	mAP@.5:.95	f1-score
Bus	0.8037	0.6602	0.641	0.4811	0.7249
Bike	0.7705	0.4909	0.4625	0.2243	0.5997
Car	0.8338	0.6354	0.6237	0.4069	0.7212
Pedestrian	0.7882	0.2389	0.2252	0.1224	0.3667
Truck	0.8177	0.3778	0.3732	0.2924	0.5168
All	0.8028	0.4806	0.4651	0.3054	0.5859

Table 4. Result of YOLOv7-e6e model on the small dataset with a 3X augmentation.

In Table 5, the model demonstrated notable performance in detecting cars, achieving a mAP@.5 of 0.6403 and an F1-score of 0.727. However, its performance was comparatively lower in detecting pedestrians, with a mAP@.5 of 0.262 and an F1-score of 0.3972. Additionally, the model showed satisfactory performance in detecting buses, achieving a mAP@.5 of 0.5727 and an F1-score of 0.7089. Overall, the model yielded a mAP@.5 of 0.4931 and an F1-score of 0.6141.

Small Data 10X					
	Precision	Recall	mAP@.5	mAP@.5:.95	f1-score
Bus	0.902	0.5839	0.5727	0.4029	0.7089
Bike	0.6893	0.5697	0.5152	0.2425	0.6238
Car	0.8115	0.6584	0.6403	0.4094	0.727
Pedestrian	0.5962	0.2978	0.262	0.1376	0.3972
Truck	0.7765	0.5071	0.4753	0.2906	0.6135
All	0.7551	0.5234	0.4931	0.2966	0.6141

Table 5. Result of YOLOv7-e6e model on the small dataset with a 10X augmentation.

Table 6 illustrates the model's better performance in car detection, achieving a mAP@.5 of 0.6623 and an associated F1-score of 0.7288. However, its performance was less satisfactory in pedestrian detection, recording its lowest mAP@.5 at 0.2113, with an F1-score of 0.3486. Furthermore, the model exhibited commendable performance in detecting buses, attaining a mAP@.5 of 0.5224 and an F1-score of 0.6693. Overall, the model achieved a mAP@.5 of 0.4339 and an F1-score of 0.5502.

Small Data 20X					
	Precision	Recall	mAP@.5	mAP@.5:.95	f1-score
Bus	0.9044	0.5312	0.5224	0.3879	0.6693
Bike	0.7079	0.5193	0.4778	0.2232	0.5991
Car	0.7719	0.6903	0.6623	0.40949	0.7288
Pedestrian	0.6705	0.2355	0.2113	0.1041	0.3486
Truck	0.4638	0.3599	0.2956	0.2319	0.4053
All	0.7037	0.4672	0.4339	0.27131	0.5502

Table 6. Result of YOLOv7-e6e model on the small dataset with a 20X augmentation.

Table 7 demonstrates the model's proficiency in car detection, achieving a mAP@.5 of 0.6957 and an associated F1-score of 0.7404. However, its performance was less than pedestrian detection, recording its lowest mAP@.5 at 0.176, with an F1-score of 0.3009. Furthermore, the model showed a great performance in detecting buses, attaining a mAP@.5 of 0.5692 and an F1-score of 0.6625. Overall, the model achieved a mAP@.5 of 0.5025 and an F1-score of 0.6005.

Assembled					
	Precision	Recall	mAP@.5	mAP@.5:.95	f1-score
Bus	0.7274	0.6082	0.5692	0.409	0.6625
Bike	0.6821	0.5246	0.4783	0.225	0.593
Car	0.7631	0.719	0.6957	0.4393	0.7404
Pedestrian	0.8883	0.1812	0.176	0.0916	0.3009
Truck	0.8138	0.623	0.5932	0.4038	0.7054
All	0.7749	0.5312	0.5025	0.3138	0.6005

Table 7. Result of YOLOv7-e6e model on the assembled dataset.

The results of our experiments across five variations of

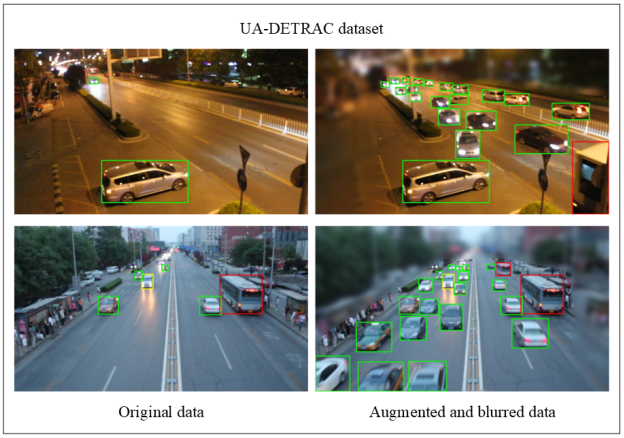

Figure 6. The comparison between original image and augmented image in UA-DETRAC dataset.

small data are summarized in Table 8. The highest recall, mAP@.5, and mAP@.5:.95 are 0.5312, 0.5025, and 0.3138, respectively, achieved in the assembled dataset. The highest precision and F1-score are 0.8132 and 0.62, respectively.

	Precision	Recall	mAP@.5	mAP@.5:.95	f1-score
Small Data (SD)	**0.8132**	0.5104	0.4798	0.29	**0.62**
SD 3X	0.8028	0.4806	0.4651	0.3054	0.5859
SD 10X	0.7551	0.5234	0.4931	0.2966	0.6141
SD 20X	0.7037	0.4672	0.4339	0.27131	0.5502
Assembled	0.7749	**0.5312**	**0.5025**	**0.3138**	0.6005

Table 8. Results of the YOLOv7-E6E model on multiple small datasets.

We observe significant improvements, with the mAP@.5 increasing from 0.4798 to 0.5025, and the mAP@.5:.95 rising from 0.29 to 0.3138 using the Assembled dataset.

5.2. UA-DETRAC

Small Data					
	Precision	Recall	mAP@.5	mAP@.5:.95	f1-score
Others	0	0	0	0	0
Car	0.7135	0.7565	0.7171	0.5225	0.7344
Van	0.6155	0.4754	0.4179	0.3319	0.5365
Bus	0.6155	0.7771	0.7359	0.5607	0.7574
All	0.7387	0.4677	0.4677	0.3538	0.5071

Table 9. Results of YOLOv7-E6E model on the small data.

Figure 6 shows the original and augmented samples in UA-DETRAC dataset [14]. The results presented in Table 9 illustrate the performance of the YOLOv7-E6E model trained on a modest dataset without augmentation. Notably, the model achieved its highest mAP@.5 of 0.7359 for detecting Buses, while recording its lowest mAP@.5 of 0 for Others detection. It showed impressive performance in detecting Cars, with a mAP@.5 of 0.7171. Overall, the model exhibited a mAP@.5 of 0.4677 and an F1-score of 0.5071.

Blurred Small Data					
	Precision	Recall	mAP@.5	mAP@.5:.95	f1-score
Others	0.9326	0.6032	0.6062	0.4496	0.7326
Car	0.8458	0.7973	0.7714	0.5681	0.8208
Van	0.6712	0.6474	0.584	0.451	0.6591
Bus	0.8807	0.8811	0.8487	0.6503	0.8809
All	0.8326	0.7323	0.7026	0.5298	0.7734

Table 10. Results of YOLOv7-E6E model on the blurred small data.

As shown in Table 10, the YOLOv7-E6E object detection model demonstrates its performance on a small dataset containing blurred regions. The model achieved a strong mean Average Precision (mAP@.5) of 0.7026, indicating a good overall ability to detect objects. However, its performance varied across different classes. Notably, it excelled at identifying Buses, achieving a top mAP@.5 of 0.8487. This suggests the model effectively learned the distinctive features of buses even when partially obscured. Conversely, Van detection were the most challenging, with a lowest mAP@.5 of 0.584. The F1-score of 0.7734 further supports this notion, suggesting a good balance between precision and recall for most classes, but potentially needing improvement for van detection.

Blurred Small Data 3X					
	Precision	Recall	mAP@.5	mAP@.5:.95	f1-score
Others	0.9071	0.6177	0.6243	0.4617	0.7349
Car	0.8449	0.7734	0.7717	0.5657	0.8076
Van	0.581	0.6639	0.5939	0.4578	0.6197
Bus	0.8707	0.8861	0.8664	0.6257	0.8783
All	0.8009	0.7353	0.7141	0.5277	0.7601

Table 11. Results of YOLOv7-E6E model on the blurred small data with 3X augmentation.

The results presented in Table 11 illustrate the performance of the YOLOv7-E6E model trained on a small dataset with 3X augmentation. The model excelled at identifying Buses, achieving a top mAP@.5 of 0.8664. Conversely, Van detection proved the most challenging, with the lowest mAP@.5 of 0.5939. This difference of 0.2725 in mAP@.5 highlights a significant performance gap. Vans may require more diverse details for accurate identification, details that might have been limited even with augmentation. The F1-score of 0.7601 further supports this notion, suggesting a good balance between precision and recall for most classes, but potentially needing improvement for van detection.

As depicted in Table 12, the YOLOv7-E6E object detection model showcases its potential on a limited dataset significantly enriched with various transformations (10X augmentation). This extensive augmentation approach likely furnished the model with a wider spectrum of object appearances to learn from, potentially enhancing its generalizability. The model attained a commendable overall mean Average Precision (mAP@.5) of 0.7149, indicating its pro-

Blurred Small Data 10X					
	Precision	Recall	mAP@.5	mAP@.5:.95	f1-score
Others	0.8873	0.6311	0.6262	0.4522	0.7376
Car	0.7358	0.8754	0.8108	0.6006	0.7995
Van	0.6702	0.6518	0.5771	0.4491	0.6609
Bus	0.8615	0.9104	0.8453	0.6359	0.8853
All	0.7887	0.7672	0.7149	0.5345	0.7708

Table 12. Results of YOLOv7-E6E model on the blurred small data with 10X augmentation.

ficient ability to detect objects despite the constraints of a modest dataset.

Delving deeper into the findings, we notice some fluctuations in performance across different object categories. The model excelled in identifying Buses, achieving a top mAP@.5 of 0.8453. This implies that the model effectively grasped the distinctive features of buses even with the significant dataset augmentation. Conversely, Van detection posed as the most challenging task, with the lowest mAP@.5 of 0.5771. This considerable difference of 0.2682 in mAP@.5 underscores a substantial performance gap. It's plausible that vans exhibit a more varied range of visual characteristics crucial for precise identification, details that might still be limited even with a 10X augmentation. The F1-score of 0.7708 aligns with this notion, indicating a good balance between precision and recall for most categories, albeit potentially warranting enhancement for van detection.

Blurred Small Data 20X					
	Precision	Recall	mAP@.5	mAP@.5:.95	f1-score
Others	0.7984	0.2233	0.2158	0.1781	0.349
Car	0.7748	0.8472	0.8065	0.6071	0.8084
Van	0.6847	0.6461	0.5736	0.4507	0.6648
Bus	0.8684	0.877	0.8492	0.6734	0.8727
All	0.7816	0.6484	0.6113	0.4773	0.674

Table 13. Results of YOLOv7-E6E model on the blurred small data with 20X augmentation.

The outcomes depicted in Table 13 showcase the performance of the YOLOv7-E6E model trained on a modest dataset with 20X augmentation.The model excelled at detecting Buses, achieving a top mAP@.5 of 0.8492. Conversely, the model struggled with the 'Others' category, achieving a much lower mAP@.5 of 0.2158. This substantial difference of 0.6334 in mAP@.5 highlights a significant performance gap. It's possible that the 'Others' category encompasses a highly diverse set of objects, making it more challenging for the model to learn comprehensive detection patterns, even with a large number of augmented images. The F1-score of 0.674 partially reflects this notion, suggesting an overall trade-off between precision and recall that might be improved for specific classes like 'Others'.

The results presented in Table 14 illustrate the performance of the YOLOv7-E6E model trained on assembled data. The model achieved mean Average Precision (mAP@.5) of 0.7036, indicating its effectiveness in detecting objects within this varied dataset. The model demonstrated exceptional proficiency in identifying buses, attaining a peak mAP@.5 of 0.8581. Conversely, detecting vans posed the most formidable challenge, yielding the lowest mAP@.5 of 0.5089. This notable discrepancy of 0.3492 in mAP@.5 underscores a substantial performance contrast.

Assembled					
	Precision	Recall	mAP@.5	mAP@.5:.95	f1-score
Others	0.8996	0.6609	0.6529	0.4976	0.762
Car	0.8511	0.7945	0.7945	0.5769	0.8218
Van	0.564	0.6211	0.5089	0.4056	0.5912
Bus	0.8919	0.8857	0.8581	0.644	0.8888
All	0.8017	0.7406	0.7036	0.531	0.766

Table 14. Results of YOLOv7-E6E model on the assembled data.

	Precision	Recall	mAP@.5	mAP@.5:.95	f1-score
Small Data (SD)	0.7387	0.4677	0.4677	0.3538	0.5071
Blurred SD	**0.8326**	0.7323	0.7026	0.5298	**0.7734**
Blurred SD 3X	0.8009	0.7353	0.7141	0.5277	0.7601
Blurred SD 10X	0.7887	**0.7672**	**0.7149**	**0.5345**	0.7708
Blurred SD 10X	0.7816	0.6484	0.6113	0.4773	0.674
Assembled	0.8017	0.7406	0.7036	0.531	0.766

Table 15. Results of the YOLOv7-E6E model on multiple small datasets.

Our experiments exploring the YOLOv7-E6E model's performance on various small datasets with augmentation techniques are summarized in Table 15. While 10X augmentation achieved the highest overall mAP@.5 of 0.7149 indicating balanced performance, small data with blurred images yielded the highest precision of 0.8326 and F1-score of 0.7734, suggesting improved detection accuracy.

6. Conclusion

In this work, we proposed a simple augmentation technique for the object detection dataset, which is suitable for stationary camera-based datasets. Through extensive testing on two traffic monitoring datasets, we demonstrate that our augmentation approach can enhance model performance, particularly in scenarios where only a limited number of labeled samples are available and a number of samples per class is imbalanced. It achieved the performance level of the model trained on the entire dataset while utilizing only 8.5 percent of the original dataset. However, our suggested augmentation method is only tailored for stationary camera surveillance scenarios, restricting its scalability for general object detection data augmentations. Moreover, in-place augmentation is done by copy-pasting the region of object bounding boxes, which might also include irrelevant image parts from the original sample. Therefore, segmentation labels could be more appropriate, and further investigation is needed.

7. Acknowledgement

This research has been supported by the Emirates Center for Mobility Research (ECMR) through Grant 12R012, United Arab Emirates University (UAEU), United Arab Emirates.

References

[1] Aleksandr Fedorov, Kseniia Nikolskaia, Sergey Ivanov, Vladimir Shepelev, and Alexey Minbaleev. Traffic flow estimation with data from a video surveillance camera. *Journal of Big Data*, 6(1), August 2019. 1

[2] Sultan Daud Khan and Habib Ullah. A survey of advances in vision-based vehicle re-identification. *Computer Vision and Image Understanding*, 182:50–63, May 2019. 1

[3] Rajeev Ranjan, Vishal M. Patel, and Rama Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1):121–135, January 2019. 1

[4] Sulabh Kumra and Christopher Kanan. Robotic grasp detection using deep convolutional neural networks. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, September 2017. 1

[5] Georgia Gkioxari, Ross Girshick, Piotr Dollar, and Kaiming He. Detecting and recognizing human-object interactions. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, June 2018. 1

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2016. 1

[7] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, July 2017. 1

[8] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 1

[9] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1), February 2017. 1

[10] Debidatta Dwibedi, Ishan Misra, and Martial Hebert. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, October 2017. 2

[11] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2016. 2

[12] Georgios Georgakis, Arsalan Mousavian, Alexander Berg, and Jana Kosecka. Synthesizing training data for object detection in indoor scenes. In *Robotics: Science and Systems XIII*, RSS2017. Robotics: Science and Systems Foundation, July 2017. 2

[13] Munkhjargal Gochoo, Munkh-Erdene Otgonbold, Erkhembayar Ganbold, Jun-Wei Hsieh, Ming-Ching Chang, Ping-Yang Chen, Byambaa Dorj, Hamad Al Jassmi, Ganzorig Batnasan, Fady Alnajjar, Mohammed Abduljabbar, and Fang-Pang Lin. Fisheye8k: A benchmark and dataset for fisheye camera object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 5304–5312, June 2023. 2, 3, 4

[14] Longyin Wen, Dawei Du, Zhaowei Cai, Zhen Lei, Ming-Ching Chang, Honggang Qi, Jongwoo Lim, Ming-Hsuan Yang, and Siwei Lyu. Ua-detrac: A new benchmark and protocol for multi-object detection and tracking, 2015. 2, 3, 4, 6

[15] Barret Zoph, Ekin D. Cubuk, Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens, and Quoc V. Le. Learning data augmentation strategies for object detection, 2019. 2

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2014. 2, 3, 4

[17] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search, 2018. 2

[18] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D. Cubuk, Quoc V. Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation, 2021. 2

[19] Nikita Dvornik, Julien Mairal, and Cordelia Schmid. Modeling visual context is key to augmenting object detection datasets. 2018. 2

[20] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, September 2009. 2

[21] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space, 2019. 2

[22] Sima Behpour, Kris M. Kitani, and Brian D. Ziebart. Ada: Adversarial data augmentation for object detection. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, January 2019. 2

[23] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge, 2014. 2, 3

[24] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html. 2

[25] Mate Kisantal, Zbigniew Wojna, Jakub Murawski, Jacek Naruniec, and Kyunghyun Cho. Augmentation for small object detection, 2019. 2

[26] Shitong Shao, Zeyuan Yin, Muxin Zhou, Xindong Zhang, and Zhiqiang Shen. Generalized large-scale data condensation via various backbone and statistical matching, 2023. 3

[27] Geoffrey Hinton Alex Krizhevsky. Learning multiple layers of features from tiny images, 2009. 3

[28] Amirhossein Tavanaei. Embedded encoder-decoder in convolutional networks towards explainable ai, 2020. 3

[29] 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 (CVPR)*, 2023. 4