







Enhancing Small Object Detection in Resource-Constrained ARAS Using Image Cropping and Slicing Techniques

Chinmaya Kaundanya¹^a, Paulo Cesar²^b, Barry Cronin²^c, Andrew Fleury²^d,
Mingming Liu¹^e and Suzanne Little¹^f

¹Research Ireland Insight Centre for Data Analytics, Dublin City University, Ireland

²Luna Systems, Dublin, Ireland

chinmaya.kaundanya3@mail.dcu.ie, {paulo.cesar, barry.cronin, andrew.fleury}@luna.systems,


Keywords: Small Object Detection, ARAS, Micromobility, Image Pre-Processing.


Abstract: Powered two-wheelers, such as motorcycles, e-bikes, and e-scooters, exhibit disproportionately high fatality rates in road traffic incidents worldwide. Advanced Rider Assistance Systems (ARAS) have the potential to enhance rider safety by providing real-time hazard alerts. However, implementing effective ARAS on the resource-constrained hardware typical of micromobility vehicles presents significant challenges, particularly in detecting small or distant objects using monocular cameras and lightweight convolutional neural networks (CNNs). This study evaluates two computationally efficient image preprocessing techniques aimed at improving small and distant object detection in ARAS applications: image center region-of-interest (ROI) cropping and image slicing and re-slicing. Utilizing the YOLOv8-nano object detection model at relatively low input resolutions of 160×160, 320×320, and 640×640 pixels, we conducted experiments on the VisDrone and KITTI datasets, which represent scenarios where small and distant objects are prevalent. Our results indicate that the image center ROI cropping technique improved the detection of small objects, particularly at a 320×320 resolution, achieving enhancements of 6.67× and 1.27× in mean Average Precision (mAP) on the VisDrone and KITTI datasets, respectively. However, excessive cropping negatively impacted the detection of medium and large objects due to the loss of peripheral contextual information and the exclusion of objects outside the cropped region. Image slicing and re-slicing demonstrated impressive improvements in detecting small objects, especially using the grid-based slicing strategy on the VisDrone dataset, with an mAP increase of 2.24× over the baseline. Conversely, on the KITTI dataset, although a performance gain of 1.66× over the baseline was observed for small objects at a 320×320 resolution, image slicing adversely affected the detection of medium and large objects. The fragmentation of objects at image slice borders caused partial visibility, which reduced detection accuracy. These findings contribute to the development of more effective and efficient ARAS technologies, ultimately enhancing the safety of powered two-wheeler riders. Our evaluation code scripts are publicly accessible at: https://github.com/Luna-Scooters/SOD_using_image_preprocessing.


1 INTRODUCTION


Micromobility refers to lightweight, usually electric, vehicles such as e-scooters and e-bikes, designed for short-distance travel in urban areas. The rise of these new transportation modes, particularly e-scooters, has introduced significant safety challenges that compli-


cate their integration into urban environments. The growing popularity of e-scooters has been accompanied by a sharp increase in related injuries and fatalities. In the United States, between 2017 and 2021, injuries associated with micromobility vehicles surged by 127%, reaching 77,200 incidents (Chen et al., 2024). Similarly, in 2023, police in Germany reported 9,425 e-scooter accidents, a 14.1% increase from the previous year's 8,260, with the number of fatalities from micromobility-related road accidents doubling compared to 2022. In Paris, e-scooters were used for around 20 million trips on 15,000 rental scooters in 2022, yet the city recorded 459 accidents involving


^a <https://orcid.org/0009-0007-4046-5936>

^b <https://orcid.org/0009-0000-7171-499X>

^c <https://orcid.org/0009-0008-5720-8941>

^d <https://orcid.org/0009-0003-6916-6770>

^e <https://orcid.org/0000-0002-8988-2104>

^f <https://orcid.org/0000-0003-3281-3471>

these vehicles or similar micromobility devices, including three fatal incidents (dwG, 2024).

Advanced Rider Assistance Systems (ARAS) are designed to enhance the safety of powered two-wheelers, such as motorcycles, scooters, and micromobility vehicles. These systems integrate a combination of sensors, advanced algorithms, and connectivity features to assist riders in diverse traffic situations, aiming to reduce accidents and improve the overall riding experience (Ait-Moula et al., 2024). The majority of ARAS systems offer applications such as vehicle collision warning, blind spot detection, Active Cruise Control et cetera (Ait-Moula et al., 2024). Unlike ARAS, Advanced Driver Assistance Systems (ADAS) are developed for four-wheeled vehicles, which have the capacity and computational resources to accommodate and power sophisticated hardware optimized for complex artificial intelligence (AI) models.

ADAS hardware platforms are varied, ranging from complex multiprocessor system-on-chip (MPSoC) CPUs to traditional microcontroller units (MCUs), digital signal processors (DSPs), and specialized hardware like field-programmable gate arrays (FPGAs), application-specific integrated circuits (ASICs), or dedicated GPU platforms such as NVIDIA's Tegra and Jetson families. Due to the resource-constrained environment of micromobility vehicles, implementing AI-based ARAS necessitates the use of low-compute hardware platforms. As a result, there is a trade-off between the complexity of the models implemented and the hardware platform, which impacts performance metrics including functional accuracy, energy consumption, and processing speed (latency and throughput) (Borrego-Carazo et al., 2020).

ARAS typically consist of monocular cameras, multi-camera setups, or multi-sensor fusion systems, and can operate in either active or passive modes. In this study, we consider a monocular camera-based, passive ARAS that utilizes a low-specification hardware platform running a two-dimensional (2D) object detection convolutional neural network (CNN) model. This system alerts riders to potential headway-monitoring events involving objects approaching from the front, where a headway monitoring warning is triggered when the distance to the vehicle ahead becomes unsafe and more distancing is required. The standard procedure for deploying such an ARAS involves mounting the camera at a fixed position on the vehicle and calibrating it to map the three-dimensional (3D) world coordinates to the two-dimensional (2D) image coordinate system. A crucial component of this is the object detection model,

which is then used by an object tracking algorithm that monitors objects over time to determine their distance and velocity based on predefined rules. Since the ARAS system relies on the initial detection of objects by the model, early detection accuracy is vital to the overall performance of the system.

Considering huge advancements in the object detection field, CNN models exhibit impressive performance with affordable computational requirements, making them suitable for resource-constrained devices. However, detecting small or distant objects remains a major challenge compared to objects of conventional scale. Small objects occupy fewer pixels and contain less information, resulting in substantially lower detection performance. The common challenges associated with small object detection include: (1) insufficient feature representation from individual layers in basic CNNs for small objects; (2) a lack of contextual information necessary for accurate detection; (3) an imbalance between foreground and background training examples that complicates classification; and (4) a scarcity of positive training examples for small objects (Liu et al., 2021). In ARAS applications on resource-constrained platforms, significant computational limitations make it crucial to balance the trade-off between required latency and the accuracy of object detection models. Low-specification platforms, such as microcontrollers with limited memory, are unable to support advanced object detection algorithms or process high-resolution images, which exacerbates the difficulty of detecting small or distant objects.

To address this challenge, we evaluate two image preprocessing techniques: (1) image center region-of-interest (ROI) cropping and (2) image slicing and reslicing, using the KITTI (Geiger et al., 2013) and VisDrone (Du et al., 2019) datasets. The motivation for employing these techniques is to enhance the model's performance in detecting small or distant objects by introducing computationally inexpensive operations as an image preprocessing step during model inference. Experiments are conducted using the YOLOv8-nano (Jocher et al., 2023) model with input image resolutions of 160×160, 320×320, and 640×640. In this study, we adopt the definition of small objects as specified in the MS COCO (Lin et al., 2014) evaluation metrics: objects with bounding boxes occupying areas less than or equal to 32×32 pixels are classified as "small," those up to 96×96 pixels are considered "medium," and large objects exceed these dimensions. These size thresholds are widely recognized within the community for datasets involving common objects. The initial experiments are performed on the VisDrone dataset to validate the effectiveness of the

two techniques before evaluating them on the KITTI dataset.

Given that the VisDrone dataset comprises images captured by drone-mounted cameras, it provides a rich source of small or distant objects, which are particularly challenging for object detection models. Moreover, drones themselves exemplify resource-constrained environments. Due to the limited availability of labelled ARAS datasets specifically for two-wheelers, we selected the KITTI dataset for our evaluation. The KITTI dataset contains front-facing camera images collected from cameras mounted on cars, serving as an appropriate benchmark for assessing the aforementioned image preprocessing techniques.

The image centre region-of-interest (ROI) cropping experiments on both the VisDrone and KITTI dataset demonstrate a positive trend in the model's performance, particularly in detecting small objects as the cropping factor increases. However, aggressive cropping causes the model to miss a significant number of large objects that majorly lie outside the cropped ROI, thereby reducing detection performance for such object sizes. Excessive cropping leads to a narrow field of view, which excludes many medium and large objects, underscoring the importance of balancing zoom levels to optimize detection across different object scales.

Image slicing and re-slicing show impressive improvements in detecting small objects, especially with the grid-based slicing strategy on the VisDrone dataset, where remarkable performance is observed over the baseline. However, on the KITTI dataset, while a notable performance increase is seen for small objects at lower resolutions, image slicing negatively impacts the detection of medium and large objects. Qualitative analysis in Figure 3 and Figure 4 indicates that slicing the images often leads to abrupt truncation of medium and small objects, causing them to appear only partially within the image slices. This partial visibility adversely affects the model's performance due to incomplete object representations. Figure 5 and Figure 6 demonstrate the trend in the mAP scores for all the scales and input resolutions on both VisDrone and KITTI test sets.

The structure of the paper is organized as follows: Section 2 reviews related techniques in the area of small object detection. In Section 3, we explain the two image preprocessing methods of interest. Section 4 provides details about the datasets used in the experiments, while Section 5 describes the experimental setup. The results of the experiments are presented and analyzed in Section 6, and finally, our conclusions are drawn in Section 7.

2 RELATED WORK ON SMALL OBJECT DETECTION

Existing techniques for improving small object detection often involve modifications to the architectures of object detection models specifically tailored for enhanced detection of small objects. However, these modifications typically increase the number of parameters, which is suboptimal for resource-constrained devices with limited memory. Another common approach involves image augmentation techniques, such as copy-pasting small objects into various positions, dividing and resizing images, or using generative adversarial networks (GANs) to generate synthetic samples of small objects. While effective, these methods are primarily training-time strategies and do not address the challenges of optimizing inference performance on resource-limited hardware. Although recent studies have proposed lightweight object detection architectures, they remain too large for ultra-low-specification devices such as microcontroller units (MCUs).

2.1 Small Object Detection Using Architectural Adjustments

Small object detection (SOD) remains a significant challenge in computer vision due to limited pixel representation, which makes feature extraction difficult. A common approach to tackle these challenges involves architectural modifications aimed at preserving crucial spatial information and enhancing feature representations at multiple scales. Feature Pyramid Networks (FPN) (Lin et al., 2017) have been extensively used to address the loss of spatial details for small objects. FPN-based models integrate low-level and high-level feature maps to enhance small object detection accuracy. Despite their success, recent research indicates that simple feature fusion may introduce noise, potentially overwhelming the signal from small objects. Thus, many enhancements have been proposed, such as PANet (Liu et al., 2018), which improves the information transmission between feature maps by introducing a bottom-up path augmentation. Similarly, NAS-FPN (Ghiasi et al., 2019) uses neural architecture search to optimize feature fusion schemes across different layers, improving the representation for small targets.

Another notable architectural adjustment is the attention mechanism, which has been used to amplify relevant spatial information. For instance, SSP-Net (Hong et al., 2021) incorporates context attention modules that emphasize features at specific scales, addressing gradient inconsistencies and improving

small object detection. More recent approaches, based on new versions of YOLO (Zhao et al., 2023; Li et al., 2023; Wang et al., 2023; Tang et al., 2024), include additional prediction heads specifically for detecting extremely small objects, although at the cost of increased computational demands.

2.2 Small Object Detection Using Image Augmentation Techniques

In addition to architectural modifications, image pre-processing and augmentation techniques play a vital role in enhancing small object detection in resource-constrained environments. Data augmentation approaches such as oversampling and copy-paste methods have been proposed to address the insufficient representation of small objects in training datasets. Kisanal et al. (Kisanal, 2019) introduced a copy-paste technique to increase the diversity of small object instances by duplicating and pasting small objects into different parts of the image. While effective, these methods sometimes introduce unrealistic context, which can degrade detection performance.

To overcome the limitations of traditional augmentation, contextual-aware augmentation strategies have been explored. For instance, Chen et al. (Chen et al., 2019) proposed RRNet, which leverages a semantic segmentation network to ensure that augmented objects are placed in semantically consistent regions of the image, leading to better detection performance. Similarly, Zhao et al. (Zhao et al., 2019) used context-preserving transformations, such as modifying brightness and blending objects into suitable backgrounds, to enhance the detection of small objects.

Another line of research involves super-resolution techniques to enhance the quality of small objects before detection. GAN-based super-resolution models, such as MTGAN (Bai et al., 2018), aim to improve the visibility of small targets by generating higher-resolution representations. These methods have been effective in enhancing the detection of small objects, especially in scenarios like UAV-based and remote sensing applications, where objects of interest are typically far from the camera (Li et al., 2017). However, GAN-based methods can be computationally expensive, making them significantly challenging to deploy in resource-constrained environments.

Several unique hybrid techniques have been proposed to enhance small object detection. For instance, EdgeDuet (Yang et al., 2022) utilizes tiling, where video frames are partitioned into smaller tiles, and only the tiles containing potential small objects are offloaded to the cloud for detection. This approach re-

duces the data transmitted to the cloud, thereby accelerating small object detection by focusing processing power on relevant areas. However, this tiling method relies on cloud processing, which can introduce delays if network conditions are poor.

Some studies propose lightweight object detection networks such as the Lightweight Multi-Scale Attention YOLOv8 (Ma et al., 2024), which is a multi-scale fusion attention-based architecture. Similarly, RC-YOLO (Guo et al., 2024) uses predefined anchor boxes to predict target box sizes, improving object detection speed and accuracy. Nonetheless, even though these models are designed to be lightweight, they are still too large for low-spec devices such as microcontroller units.

3 IMAGE PRE-PROCESSING FOR SOD

This section outlines the methodology used in this study, focusing on image preprocessing techniques to evaluate their impact on improving the detection of small or distant objects in Advanced Rider Assistance Systems (ARAS) within resource-constrained environments. Two key image preprocessing techniques were utilized: (1) image center region-of-interest (ROI) cropping and (2) image slicing and re-slicing. The rationale and implementation logic behind both techniques are detailed, along with their respective motivations. To illustrate these methods, visual simulations are presented using sample images.

3.1 Image Centre ROI Cropping

We implement an iterative image centre Region of Interest (ROI) cropping technique to evaluate its impact on object detection performance, especially for detecting small or distant objects relative to the camera. Image cropping, a fundamental technique in spatial domain processing, involves selecting a specific subset of pixels from the original image, thereby reducing its dimensions while maintaining the resolution of the retained area. Our approach systematically increases the crop percentage from all sides of the input images. This methodology is specifically tailored for ARAS use cases, where the upper portion of images – predominantly consisting of the sky – is deemed less critical. Conversely, the lower portion is crucial for detecting objects in proximity to the rider; therefore, we apply minimal cropping to the bottom edge to preserve these important details.

We select four progressive cropping factors. Cropping begins with a 10% reduction from the sides and

top and 1% from the bottom. With each iteration, we increase the cropping by an additional 10% from the sides and top and 1% from the bottom, culminating in a total of 40% cropping from the sides and top and 4% from the bottom. This technique effectively zooms into a designated region of each image across multiple iterations. Initially, the cropping is applied to the original high-resolution images, which are subsequently resized during the YOLOv8 model inference. The detailed technical explanation can be found in the section 5

During each iteration, ground truth bounding boxes are adjusted to ensure a fair comparison with the model's predictions. Boxes that are entirely outside the ROI are excluded, and the remaining annotations are meticulously aligned with the new cropped dimensions. The model's predictions are then compared against the adjusted ground truth annotations, and performance metrics – specifically mean Average Precision (mAP) at a threshold of 0.5 – are computed for various object size categories, small, medium, and large.

3.2 Image Slicing and Re-Slicing

Image slicing and re-slicing is a technique in which high-resolution images are partitioned into multiple low-resolution slices or patches, which are then processed sequentially by an object detection model. Subsequently, the detections from each slice are mapped back onto the original high-resolution image by combining the bounding box predictions. In this work, we consider two types of slicing strategies – grid slicing and vertical slicing – with the number of slices fixed at four. To limit computational overhead, we ensure that there is no overlap among the slices. We utilized the SAHI (Slicing Aided Hyper Inference) (Akyon et al., 2022) implementation for image slicing during model inference and for re-slicing the bounding box predictions. Details about SAHI are provided in Section 3.2.1.

The motivation behind this technique is to leverage the higher inference-per-second capacity of target platforms equipped with dedicated machine learning accelerators, such as GPUs or DSPs, thereby enhancing the ability of compact detection models to detect smaller or more distant objects in real time. Additionally, this technique is advantageous for low-specification microcontrollers with limited image buffer capacity, as high-resolution images can be sliced into multiple patches and processed sequentially, improving detection performance at the expense of additional processing time.

3.2.1 Slicing Aided Hyper Inference (SAHI)

SAHI is an open-source framework designed to enhance the detection of small and distant objects that often struggle to be accurately identified by standard object detection models due to their limited pixel representation in high-resolution images. SAHI addresses this issue by dividing an image into smaller overlapping slices during both training (fine-tuning) and inference stages, resulting in larger relative pixel coverage for small objects, which aids detection without needing extensive modifications to existing object detection models. During inference, an image is sliced into overlapping patches, resized while preserving the aspect ratio, and each patch is processed independently by an object detection model. The final predictions are merged using Non-Maximum Suppression (NMS), with parameters such as the Intersection Over Smaller (IoS) area used to fine-tune detection in cases of overlapping predictions. Additionally, GREEDYNMM (Greedy Non-Maximum Merging) and NMS help in ensuring the best possible bounding box proposals are retained. The pipeline can optionally add predictions from a full-image inference pass to detect larger objects, thereby combining the benefits of detailed small object detection and complete scene analysis. This slicing-based approach is particularly suitable for applications in ARAS for micromobility vehicles, as it can improve the detection of small or far-away objects such as pedestrians, bicycles, or other vehicles, using computationally inexpensive techniques – essential for resource-constrained environments.

4 DATASETS

In this study, we utilize two popular datasets, KITTI and VisDrone, to evaluate the effectiveness of the image preprocessing techniques discussed in the previous section. These datasets were chosen due to their relevance to the target applications and their diverse data characteristics. This section provides an overview of the key attributes of each dataset and highlights their significance for the experiments conducted in this work.

4.1 VisDrone

The VisDrone dataset is a comprehensive benchmark in computer vision, consisting of 10,000 high-resolution images captured by drone-mounted cameras in 14 different cities across China. The images were collected under various weather and lighting

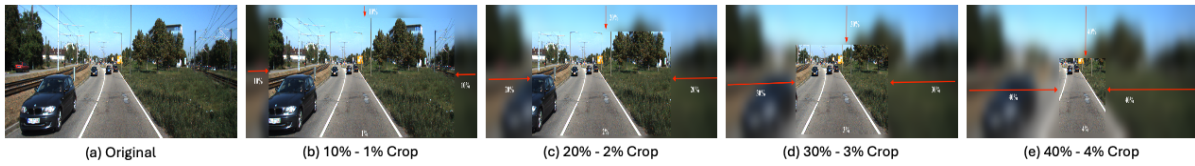


Figure 1: A cropping simulation on a sample image from KITTI dataset. (a) represents the original image followed by the regions-of-interest as per the different cropping factors captioned in the format: left/right/top crop % – bottom crop %. This figure also demonstrates selecting a specific region-of-interest to include salient parts of the image for ARAS use-cases.

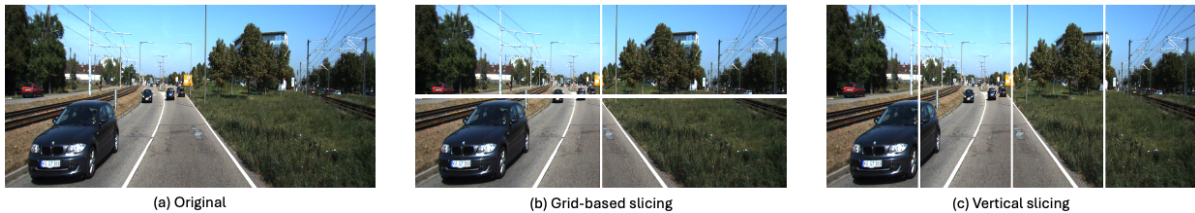


Figure 2: An Image-slicing simulation on a sample image from KITTI dataset. (a) represents the original image, (b) shows the grid-based slicing style and (c) is the vertical slicing style. This figure also demonstrates how different image slices capture different parts of the image. The figure also demonstrates the effect of different slicing styles on the original image.

conditions, encompassing diverse urban and suburban environments. Each image is annotated with detailed information for a wide range of object categories including pedestrians, bicycles, cars, and tricycles. Due to the aerial perspective of drones, the dataset contains a significant number of small or distant objects, which are particularly challenging for object detection models.

Drones operate in resource-constrained environments similar to micromobility vehicles, often having limited computational capabilities and energy resources. This resemblance underscores the relevance of using the VisDrone dataset to evaluate methods that improve object detection performance without imposing significant additional computational burdens.

Table 1: Number of ground truth objects per class and size in VisDrone-Test set.

Class	GT Count		
	Small	Medium	Large
Pedestrian	18,848	2,066	92
People	6,025	348	3
Bicycle	1,067	230	5
Car	15,121	11,843	1,110
Van	2,912	2,669	190
Truck	777	1,445	437
Tricycle	263	249	18
Awning-tricycle	274	300	25
Bus	703	1,714	523
Motor	4,847	992	6

4.2 KITTI

The KITTI dataset is a widely recognized benchmark in computer vision, particularly in the domains of autonomous driving and advanced driver assis-

tance systems. It offers a comprehensive collection of high-quality data captured from vehicles equipped with multiple sensors, including high-resolution RGB cameras, grayscale stereo cameras, and 3D laser scanners (LiDAR). The dataset comprises over 200,000 images with detailed annotations for various tasks such as object detection, tracking, semantic segmentation, and optical flow. Annotated classes include cars, pedestrians, cyclists, and other road users, encompassing a diverse range of urban, rural, and highway environments under different weather and lighting conditions.

Although originally designed for autonomous vehicles, the KITTI dataset is highly pertinent to ARAS for micromobility vehicles. The front-facing camera images simulate the perspective of a rider, capturing dynamic traffic scenarios that are critical for rider safety. This alignment makes KITTI an appropriate and valuable resource for evaluating object detection models in contexts relevant to ARAS applications.

The dataset's rich diversity in object scales, distances, and occlusion levels makes it particularly useful for assessing techniques aimed at enhancing the detection of small or distant objects – challenges that are especially pronounced in ARAS due to limited computational resources and the necessity for timely hazard recognition. By providing a realistic and complex visual environment, the KITTI dataset enables rigorous evaluation of object detection models under conditions that closely mirror real-world riding situations. This facilitates the testing and refinement of image preprocessing strategies intended to improve model performance on small or far-away objects without imposing significant additional computational burden. Consequently, the KITTI dataset

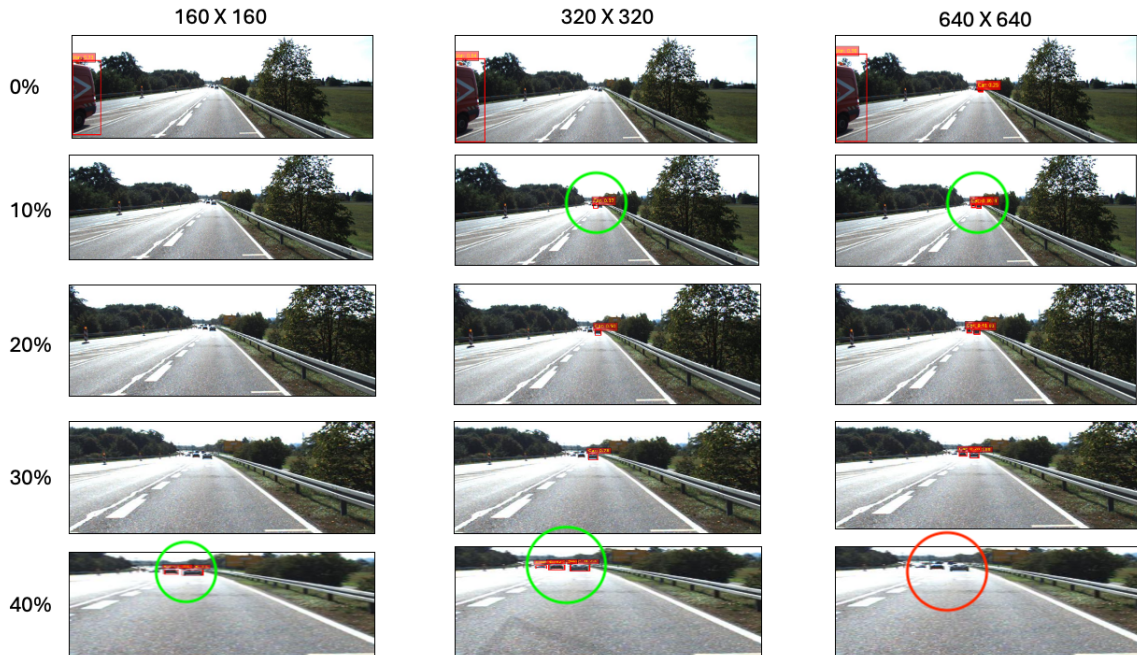


Figure 3: A comparative qualitative analysis of Image-centre-ROI cropping technique on KITTI test set with different resolutions and cropping factors. The rows represent the centre ROI cropping factors (left/right/top crop percentage) and columns denote model input image resolutions. The Green circles highlight the small or distant objects detected due to ROI cropping. The red circle shows the missed objects due to excessive cropping.

serves as an ideal benchmark for advancing the development of efficient ARAS systems in real-world traffic conditions. In this study, we use small and distant terms interchangeably as, particularly in KITTI dataset, there is no class that represents a small object that might appear close to the camera.

Table 2: Number of ground truth objects per class and size in KITTI-Test set.

Class	GT Count		
	Small	Medium	Large
Car	478	1580	865
Cyclist	41	83	17
Misc	20	56	17
Pedestrian	89	233	105
Person_sitting	0	10	9
Tram	2	21	15
Truck	20	62	25
Van	35	160	90

5 EXPERIMENT SETUP

For the experiments, we used the YOLOv8-nano model architecture, utilizing *Ultralytics*' (Jocher et al., 2023) API for model training. The standard training configuration provided by *Ultralytics* was used to train YOLOv8n on the KITTI and VisDrone datasets. The model was trained on three input resolu-

tions – 160×160, 320×320, and 640×640 – over a total of 100 epochs. During training, we applied translation (translate=0.1) and scaling (scale=0.5) transformations, along with mosaic augmentation (mosaic=1.0, close_mosaic=10), to simulate the image preprocessing techniques described in Section 3, ensuring consistency with the strategies evaluated during inference.

We conducted two sets of experiments for the image slicing and re-slicing technique, implementing two distinct slicing strategies: grid-based slicing and vertical slicing. For each strategy, we performed experiments at three resolutions – 160×160, 320×320, and 640×640 – both with and without slicing, the latter serving as the baseline for comparison. The mean Average Precision (mAP) scores for small, medium, and large objects, categorized according to the MS COCO guidelines as described in section 1, were evaluated with an Intersection Over Union (IoU) threshold set to 0.5. The primary objective of these experiments was to determine whether real-time image slicing improves the model's ability to focus on small or distant objects – details that might otherwise be missed in the original image – and thereby enhance detection performance for small objects.

We chose non-overlapping grid-based slicing not only because it maintains the aspect ratio of the original image and allows the model to focus on all parts of

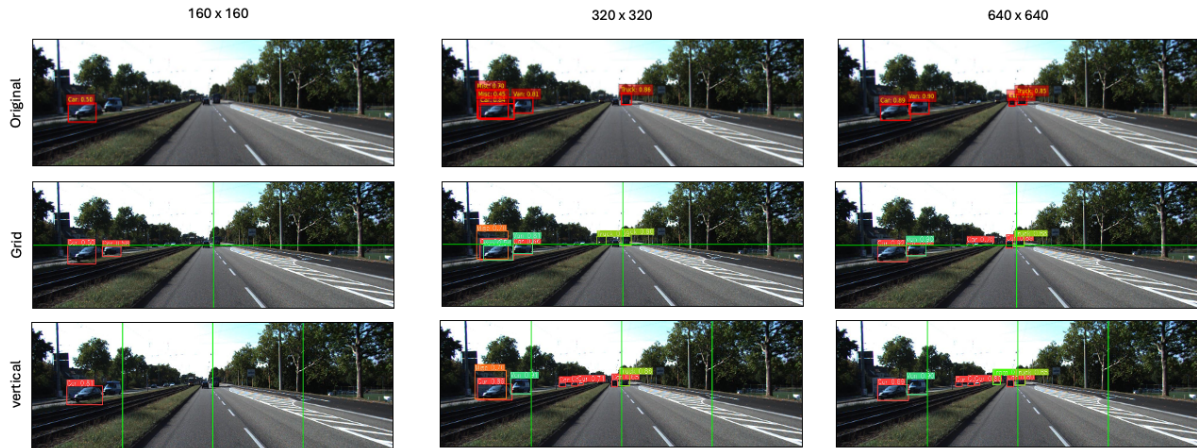


Figure 4: A comparative qualitative analysis of Image slicing and re-slicing technique on KITTI test set with different resolutions and slicing strategies. The rows represent the slicing strategy and columns denote model input image resolutions. The Green circles highlight the small or distant objects detected due to image slicing. The red circle shows the missed objects due to slicing style.

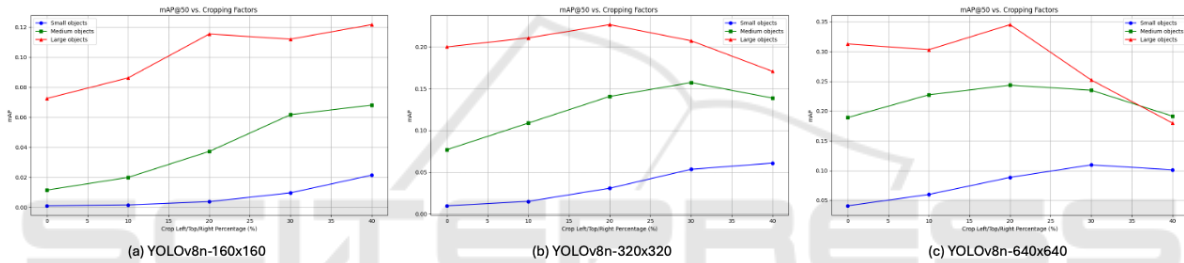


Figure 5: A comparative quantitative analysis of Image-centre-ROI cropping technique on VisDrone test set with different resolutions and cropping factors.

the image individually, but also because it is arguably a challenging slicing style. It is challenging because, it often results in objects near the centre region being awkwardly split across slices, adding complexity to the detection process. Naturally, this makes it the perfect challenge for evaluating the model’s performance especially the small or in this case, distant objects lie near central region. On the other hand, the vertical slicing strategy is less prone to awkwardly splitting objects as it only divides the image along the vertical axis. This method is particularly advantageous for monocular camera-based ARAS applications, as it captures detailed lane-wise information: the outer slices focus on areas such as parked vehicles and sidewalks, while the central slices cover the main roadway, which is crucial for detecting vehicles in these specific regions.

In the image centre ROI cropping experiment, we selected four iterations with different cropping factors. The reference point for the ROI in each iteration is fixed at the centre, and we progressively crop from the horizontal and vertical edges of the image. The first iteration begins by cropping 10% of the overall

pixel width from the left and right sides, 10% from the top, and 1% from the bottom. The remaining portion of the image serves as the ROI for that iteration and is then passed to the YOLOv8n model for inference.

Ground truth boxes of objects that appear completely outside the ROI are eliminated, and those partially appearing inside the ROI are adjusted to include only the portion within the ROI. The mean Average Precision (mAP) scores for small, medium, and large objects – categorized according to the MS COCO guidelines as described in Section 1 – were evaluated using an Intersection Over Union (IoU) threshold set to 0.5. These steps are repeated for all iterations, each time applying additional cropping to emulate a zooming effect, thereby making the ROI progressively smaller. The mAP scores are then recorded for all iterations and compared with the baseline, which is the image without any cropping of the ROI. We perform these evaluations on host machines so different performance and inference latency is to be expected when deployed on low-spec hardware platforms.

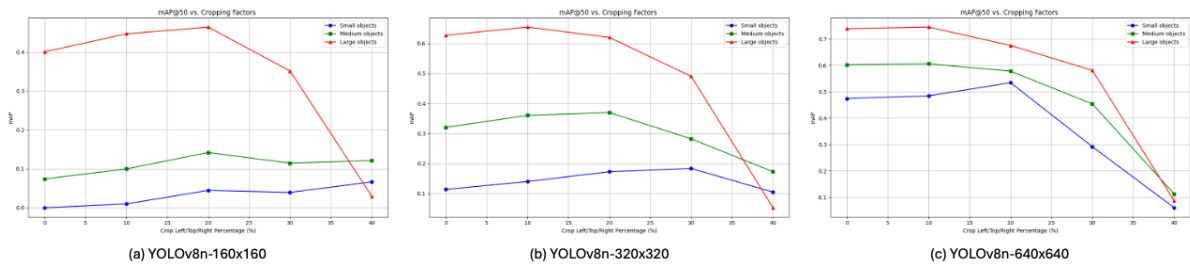


Figure 6: A comparative quantitative analysis of Image-centre-ROI cropping technique on KITTI test set with different resolutions and cropping factors.

6 RESULTS AND DISCUSSIONS

In the following sections, we present a comprehensive evaluation of the YOLOv8n models on the VisDrone and KITTI test sets, focusing on techniques aimed at improving the detection of small and distant objects. Table 1 and Table 2 shows the number of ground truth objects per class and size in VisDrone and KITTI-Test sets respectively. We explore the performance of models at various resolutions (160×160, 320×320, and 640×640) when image slicing is applied, as well as when image centre ROI cropping with different cropping factors is applied to the baseline models. By analysing the mean Average Precision (mAP@50) scores across small, medium, and large object categories, we aim to understand how image slicing and cropping strategies influence object detection performance. Our findings highlight that image slicing poses challenges due to partial object visibility. However, centre ROI cropping, when applied in a controlled manner, can significantly improve the detection of small or distant objects – even at relatively low image resolutions.

6.1 Small/Distant Object Detection Using Image Slicing and Re-Slicing

6.1.1 Evaluation on VisDrone Test-Set

In Table 3 and Table 4 a clear upward trend in mAP scores is observed across all grid and vertical slicing models as the resolution increases from 160 to 320 to 640. This trend is expected, as increasing image resolution naturally enhances detection accuracy by providing more detailed object features for the model to process thus reinforcing the validity of the proposed technique.

The mAP (small) and mAP (medium) scores of both grid and vertical slicing models at all resolutions are higher than those of their respective baseline models. Notably, for small objects at lower resolutions

– 160 and 320 – the growth in mAP is substantial. For the grid-based slicing strategy, the increases are +170% and +144% for resolutions 160 and 320, respectively. For the vertical slicing strategy, the increases are +60% and +68%, respectively. However, there is a decline in model performance for large objects at all resolutions in both strategies. This suggests that large objects are most adversely affected by slicing, which can cut them abruptly, especially considering the density of objects in each image of the VisDrone dataset.

Slicing has shown improved performance for small and medium objects as the model is able to focus on tiny features by processing the slices individually, resulting in improvements over the baseline performance. Furthermore, given the nature of the images and objects in the VisDrone dataset, the grid-based slicing showed better results than vertical slicing because more objects are abruptly cut in vertical slicing.

6.1.2 Evaluation on KITTI Test-Set

Unlike the VisDrone dataset, the slicing technique applied to KITTI images did not yield significant improvements. Tables 5 and 6 show that, for grid-based slicing, the mAP scores of the models at all resolutions and for objects of all scales were observed to be lower than those of the baseline models. In the case of vertical slicing, particularly at the 160×160 resolution, a slight increase in mAP scores for small and medium objects was observed, and for the 320×320 resolution, the mAP score for small objects surpassed the baseline by +62%. However, all other mAP scores were lower than those of the baseline models.

This suggests that for ARAS applications, especially when images are from different perspectives and vehicles are commonly located in specific areas, the slicing strategy plays a crucial role. It was observed that grid-based slicing resulted in most medium-sized or close-range objects – those appear-

Table 3: Comparison of YOLOv8n model performance at different resolutions, comparing models with **grid-based image slicing** to baseline models (*) on VisDrone-Test set. Values indicate the percentage increase (↑) or decrease (↓) compared to the baseline model without slicing. Total number of slices is 4.

Model	Slicing	mAP@50 (Small)	mAP@50 (Medium)	mAP@50 (Large)
160×160	Yes	0.0027 ↑170%	0.0262 ↑128%	0.0720 ↓0.7%
160×160*	No	0.0010	0.0115	0.0725
320×320	Yes	0.0234 ↑144%	0.1006 ↑31%	0.1834 ↓8%
320×320*	No	0.0096	0.0768	0.1999
640×640	Yes	0.0625 ↑56%	0.1961 ↑4%	0.2756 ↓11.7%
640×640*	No	0.0400	0.1880	0.3120

Table 4: Comparison of YOLOv8n model performance at different resolutions, comparing models with **vertical image slicing** to baseline models (*) on VisDrone-Test set. Values indicate the percentage increase (↑) or decrease (↓) compared to the baseline model without slicing. Total number of slices is 4.

Model	Slicing	mAP@50 (Small)	mAP@50 (Medium)	mAP@50 (Large)
160×160	Yes	0.0016 ↑60%	0.0190 ↑65%	0.0660 ↓9%
160×160*	No	0.0010	0.0115	0.0725
320×320	Yes	0.0161 ↑68%	0.0903 ↑18%	0.1855 ↓7%
320×320*	No	0.0096	0.0768	0.1999
640×640	Yes	0.0532 ↑33%	0.1944 ↑3%	0.2829 ↓9%
640×640*	No	0.0400	0.1880	0.3120

ing larger – being cut abruptly across both horizontal and vertical axes. Additionally, there was likely very little background contextual information available for small objects in each image slice, negatively affecting model performance. Conversely, with vertical patches, notably at lower resolutions, small objects likely appeared whole in the image slices. The mosaic augmentation with scale and transform augmentations also helped the model achieve decent performance with small and medium objects. For the 640×640 resolution model, it is likely that the resolution of the slices was too low, resulting in pixelated features, which led to worse performance than the baseline.

6.2 Small/Distant Object Detection Using Image Centre ROI Cropping

6.2.1 Evaluation on VisDrone Test-Set

The Table 7 and Table 8 presents the mAP scores of YOLOv8n models trained on the VisDrone dataset with image resolutions of 160×160, 320×320, and 640×640 for small, medium, and large objects. The column labeled “0%” indicates the baseline mAP scores with no cropping applied (i.e., the original images), while the columns labeled “10%” to “40%” show the mAP scores with the respective amounts of cropping applied to the images.

The mAP scores for small and medium objects exhibit an increasing trend as more zoom is applied

to the images. This is expected because cropping in makes small or distant objects appear larger and closer, which helps the model detect objects that are otherwise challenging to identify.

However, for the model trained with 640×640 resolution images, the trend in mAP scores for large objects is observed to be irregular. The performance slightly decreased during the first cropping iteration (10% crop), then increased slightly more during the 20% crop iteration. Following that, a decreasing trend is observed in the subsequent cropping iterations (30% and 40% crop). This lower detection performance for large objects could be a result of context loss due to aggressive cropping. Additionally, since the model is not familiar with images with such high levels of cropping, which can be inferred from the image augmentation parameters used during training, the model showed decreased performance.

A possible explanation for this irregular trend is that excessive cropping may cause large objects to exceed the receptive field of the model’s convolutional layers. When objects become too large relative to the input dimensions, the model might struggle to capture the entire object within its feature maps, leading to incomplete or fragmented detections. Moreover, aggressive cropping can crop out essential contextual information surrounding large objects, which is crucial for accurate detection and classification. The combination of these factors could disrupt the model’s ability to generalize well on large objects at higher cropping levels, resulting in the observed fluctuations in

Table 5: Comparison of YOLOv8n model performance at different resolutions, comparing models with **grid-based image slicing** to baseline models on KITTI-Test set. Values indicate the percentage increase (↑) or decrease (↓) compared to the baseline model without slicing. The asterisk (*) represents baseline models with no image slicing applied. Total number of slices is 4.

Model	Slicing	mAP@50 (Small)	mAP@50 (Medium)	mAP@50 (Large)
160×160	Yes	0.0002 ↑N/A	0.041 ↓44%	0.257 ↓36%
160×160*	No	0.0000	0.073	0.400
320×320	Yes	0.094 ↓17%	0.232 ↓28%	0.447 ↓29%
320×320*	No	0.113	0.320	0.627
640×640	Yes	0.334 ↓30%	0.476 ↓21%	0.625 ↓15%
640×640*	No	0.474	0.601	0.738

Table 6: Comparison of YOLOv8n model performance at different resolutions, comparing models with **vertical image slicing** to baseline models on KITTI-Test set. Values indicate the percentage increase (↑) or decrease (↓) compared to the baseline model without slicing. The asterisk (*) represents baseline models with no image slicing applied. Total number of slices is 4.

Model	Slicing	mAP@50 (Small)	mAP@50 (Medium)	mAP@50 (Large)
160×160	Yes	0.026 ↑N/A	0.091 ↑25%	0.273 ↓32%
160×160*	No	0.0000	0.073	0.400
320×320	Yes	0.183 ↑62%	0.240 ↓25%	0.451 ↓28%
320×320*	No	0.113	0.320	0.627
640×640	Yes	0.332 ↓30%	0.456 ↓24%	0.603 ↓18%
640×640*	No	0.474	0.601	0.738

performance.

6.2.2 Evaluation on KITTI Test-Set

The Table 9 and Table 10 presents the performance of the YOLOv8n models with resolutions of 160, 320, and 640 on the KITTI test set. For small objects, an increasing trend in mAP scores for 160x160 resolution was observed up to the 20%-2% cropping iteration. However, the score decreased from 0.044 to 0.039 (-12.82%) in the subsequent 30%-3% cropping iteration but unexpectedly increased again in the final 40%-4% iteration from 0.039 to 0.066 (+69.23%). This fluctuation might be due to a coincidental deviation in the aspect ratio during the 30%-3% cropping, which adversely affected the appearance of small objects. In the last iteration, the 40%-4% cropping possibly directed the model's focus toward areas rich in small objects, with an aspect ratio similar to the square aspect ratio used during training, thereby improving the mAP for small objects. For 320 resolution, mAP for small objects showed increasing trend till the 3rd iteration however, for the final iteration it dropped down slightly than baseline (-8%).

Similarly, for the 640×640 resolution model, excessive cropping led to a decline in mAP scores for small, medium, and large objects. It is likely that the reduced resolution of the ROI resulted in pixelated features, causing the model's performance to be worse than the baseline. Likewise, for medium objects at the 320×320 resolution, the mAP scores de-

clined after the second iteration through to the last iteration, with a decrease of up to 46.3%. This suggests that excessive cropping at these resolutions reduces the effective resolution of the objects, thereby negatively impacting the model's performance in both cases.

7 CONCLUSIONS AND FUTURE WORK

In this study, we addressed the challenge of detecting small and distant objects in Advanced Rider Assistance Systems (ARAS) implemented on resource-constrained hardware platforms. Recognizing that traditional convolutional neural networks (CNNs) struggle with small object detection due to limited feature representation and contextual information, we evaluated two computationally inexpensive image preprocessing techniques: image center region-of-interest (ROI) cropping and image slicing and re-slicing. Our experiments utilized the YOLOv8-nano model at input resolutions of 160×160, 320×320, and 640×640 pixels, conducted on the VisDrone and KITTI datasets.

Experiments with image center region-of-interest (ROI) cropping on both the VisDrone and KITTI datasets reveal a positive trend in detecting small objects as the cropping factor increases. However, aggressive cropping can cause the model to miss a sub-

Table 7: Comparison of YOLOv8n model performance at different resolutions and cropping factors on the VisDrone test set (Part 1). The asterisk (*) represents baseline results with no cropping applied. The column names are in the format: crop percentage from left/right/top – crop percentage from bottom of the image. S – Small, M – Medium, and L – Large objects. Values indicate the percentage increase (↑) or decrease (↓) compared to the baseline.

Model/ mAP@50	0%*			10% - 1%			20% - 2%		
	S	M	L	S	M	L	S	M	L
160 x 160	0.001	0.011	0.072	0.001 ↑0.0%	0.019 ↑72.7%	0.115 ↑59.7%	0.003 ↑200.0%	0.037 ↑236.4%	0.115 ↑59.7%
320 x 320	0.009	0.076	0.199	0.014 ↑55.6%	0.108 ↑42.1%	0.210 ↑5.5%	0.030 ↑233.3%	0.140 ↑84.2%	0.226 ↑13.6%
640 x 640	0.040	0.188	0.312	0.059 ↑47.5%	0.227 ↑20.7%	0.302 ↓3.2%	0.088 ↑120.0%	0.243 ↑29.3%	0.345 ↑10.6%

Table 8: Comparison of YOLOv8n model performance at different resolutions and cropping factors on the VisDrone test set (Part 2). Continuation from Table 1. Values indicate the percentage increase (↑) or decrease (↓) compared to the baseline.

Model/ mAP@50	0%*			30% - 3%			40% - 4%		
	S	M	L	S	M	L	S	M	L
160 x 160	0.001	0.011	0.072	0.009 ↑800.0%	0.061 ↑454.5%	0.121 ↑68.1%	0.021 ↑2000.0%	0.068 ↑518.2%	0.121 ↑68.1%
320 x 320	0.009	0.076	0.199	0.053 ↑488.9%	0.157 ↑106.6%	0.170 ↓14.6%	0.060 ↑566.7%	0.138 ↑81.6%	0.170 ↓14.6%
640 x 640	0.040	0.188	0.312	0.109 ↑172.5%	0.235 ↑25.0%	0.252 ↓19.2%	0.101 ↑152.5%	0.191 ↑1.6%	0.252 ↓19.2%

Table 9: Comparison of YOLOv8n model performance at different resolutions and cropping factors on the KITTI test set (Part 1). The asterisk (*) represents baseline results with no cropping applied. The column names are in the format: crop percentage from left/right/top – crop percentage from bottom of the image. S – Small, M – Medium, and L – Large objects. Values indicate the percentage increase (↑) or decrease (↓) compared to the baseline.

Model/ mAP@50	0%*			10% - 1%			20% - 2%		
	S	M	L	S	M	L	S	M	L
160x160	0.000	0.073	0.400	0.009 ↑-	0.099 ↑35.6%	0.446 ↑11.5%	0.044 ↑-	0.141 ↑93.2%	0.463 ↑15.8%
320x320	0.113	0.320	0.627	0.139 ↑23.0%	0.360 ↑12.5%	0.654 ↑4.3%	0.172 ↑52.2%	0.369 ↑15.3%	0.619 ↓1.3%
640x640	0.474	0.601	0.738	0.484 ↑2.1%	0.605 ↑0.7%	0.744 ↑0.8%	0.533 ↑12.4%	0.577 ↓4.0%	0.675 ↓8.5%

Table 10: Comparison of YOLOv8n model performance at different resolutions and cropping factors on the KITTI test set (Part 2). Continuation from Table 1. Values indicate the percentage increase (↑) or decrease (↓) compared to the baseline.

Model/ mAP@50	0%*			30% - 3%			40% - 4%		
	S	M	L	S	M	L	S	M	L
160x160	0.000	0.073	0.400	0.039 ↓-	0.114 ↑56.2%	0.351 ↓12.3%	0.066 ↑-	0.121 ↑65.8%	0.029 ↓92.8%
320x320	0.113	0.320	0.627	0.183 ↑61.9%	0.282 ↓11.9%	0.490 ↓21.9%	0.104 ↓8.0%	0.172 ↓46.3%	0.052 ↓91.7%
640x640	0.474	0.601	0.738	0.291 ↓38.6%	0.453 ↓24.6%	0.580 ↓21.4%	0.059 ↓87.5%	0.112 ↓81.4%	0.086 ↓88.3%

stantial number of large objects located outside the cropped area, leading to a decline in detection performance for such object sizes. Excessive cropping narrows the field of view, missing many medium and large objects, highlighting the need for balanced cropping levels to optimize detection across different object scales. Similarly, image slicing and re-slicing demonstrate strong improvements in detecting small objects, particularly with the grid-based slicing strategy on the VisDrone dataset, where performance surpasses the baseline. However, on the KITTI dataset, while improvements are seen for small objects at lower resolutions, image slicing adversely affects the detection of medium and large objects, as fragmentation at slice borders impacts their visibility.

There are some limitations of these techniques as well. Even though they are computationally efficient, especially the image slicing-reslicing technique utilized in this work adds 4x CPU latency—considering the inference is running on a CPU and the four slices must be processed individually in sequence when deployed in ARAS systems. Therefore, the tradeoff between detection accuracy and latency should be studied. Both techniques can negatively impact performance depending on the camera perspective and the specific application objectives.

To reconcile the need for improved small object detection for ARAS applications with the constraints of limited hardware resources, future research could explore adaptive techniques that dynamically adjust

the cropping level based on the speed of the ego vehicle. Implementing a multi-task learning approach using more advanced and precise image segmentation models can make object detection models more aware of the scene context, especially for distant objects. Additionally, incorporating advanced data augmentation strategies during training, such as simulated zooming and context-aware slicing, could enhance the model's robustness to varying object scales and appearance without incurring runtime computational costs.

ACKNOWLEDGEMENTS

This research was conducted with the financial support of Research Ireland (12/RC/2289.P2), at the Research Ireland Insight Centre for Data Analytics at Dublin City University, and Luna Systems. We would like to express our gratitude to Luna Systems for their invaluable support throughout the course of this research.

REFERENCES

- (2024). Germany: E-scooter accidents and fatalities on the rise – DW – 07/26/2024 — dw.com. <https://www.dw.com/en/germany-e-scooter-accidents-and-fatalities-on-the-rise-a-69775992>. [Accessed 22-10-2024].
- Ait-Moula, A., Riahi, E., and Serre, T. (2024). Effect of advanced rider assistance system on powered two wheelers crashes. *Heliyon*, 10(4).
- Akyon, F. C., Altinuc, S. O., and Temizel, A. (2022). Slicing aided hyper inference and fine-tuning for small object detection. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 966–970. IEEE.
- Bai, Y., Zhang, Y., Ding, M., and Ghanem, B. (2018). Sodmtgan: Small object detection via multi-task generative adversarial network. In *Proceedings of the European conference on computer vision (ECCV)*, pages 206–221.
- Borrego-Carazo, J., Castells-Rufas, D., Biempica, E., and Carrabina, J. (2020). Resource-constrained machine learning for adas: A systematic review. *IEEE Access*, 8:40573–40598.
- Chen, C., Zhang, Y., Lv, Q., Wei, S., Wang, X., Sun, X., and Dong, J. (2019). Rrnet: A hybrid detector for object detection in drone-captured images. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*, pages 0–0.
- Chen, D., Hosseini, A., Smith, A., Nikkhah, A. F., Heydari, A., Shoghli, O., and Campbell, B. (2024). Performance evaluation of real-time object detection for electric scooters. *arXiv preprint arXiv:2405.03039*.
- Du, D., Zhu, P., Wen, L., Bian, X., Lin, H., Hu, Q., Peng, T., Zheng, J., Wang, X., Zhang, Y., et al. (2019). Visdrone-det2019: The vision meets drone object detection in image challenge results. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*, pages 0–0.
- Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013). Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237.
- Ghiasi, G., Lin, T.-Y., and Le, Q. V. (2019). Nas-fpn: Learning scalable feature pyramid architecture for object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7036–7045.
- Guo, L., Liu, H., Pang, Z., Luo, J., and Shen, J. (2024). Optimizing yolo algorithm for efficient object detection in resource-constrained environments. In *2024 IEEE 4th International Conference on Electronic Technology, Communication and Information (ICETCI)*, pages 1358–1363. IEEE.
- Hong, M., Li, S., Yang, Y., Zhu, F., Zhao, Q., and Lu, L. (2021). Sspnet: Scale selection pyramid network for tiny person detection from uav images. *IEEE geoscience and remote sensing letters*, 19:1–5.
- Jocher, G., Qiu, J., and Chaurasia, A. (2023). Ultralytics YOLO.
- Kisantal, M. (2019). Augmentation for small object detection. *arXiv preprint arXiv:1902.07296*.
- Li, J., Liang, X., Wei, Y., Xu, T., Feng, J., and Yan, S. (2017). Perceptual generative adversarial networks for small object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1222–1230.
- Li, K., Wang, Y., and Hu, Z. (2023). Improved yolov7 for small object detection algorithm based on attention and dynamic convolution. *Applied Sciences*, 13(16):9316.
- Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125.
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). 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.
- Liu, S., Qi, L., Qin, H., Shi, J., and Jia, J. (2018). Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8759–8768.
- Liu, Y., Sun, P., Wergeles, N., and Shang, Y. (2021). A survey and performance evaluation of deep learning methods for small object detection. *Expert Systems with Applications*, 172:114602.
- Ma, S., Lu, H., Liu, J., Zhu, Y., and Sang, P. (2024). Layn: Lightweight multi-scale attention yolov8 network for small object detection. *IEEE Access*.

- Tang, S., Zhang, S., and Fang, Y. (2024). Hic-yolov5: Improved yolov5 for small object detection. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6614–6619. IEEE.
- Wang, M., Yang, W., Wang, L., Chen, D., Wei, F., KeZiEr-BieKe, H., and Liao, Y. (2023). Fe-yolov5: Feature enhancement network based on yolov5 for small object detection. *Journal of Visual Communication and Image Representation*, 90:103752.
- Yang, Z., Wang, X., Wu, J., Zhao, Y., Ma, Q., Miao, X., Zhang, L., and Zhou, Z. (2022). Edgeduet: Tiling small object detection for edge assisted autonomous mobile vision. *IEEE/ACM Transactions on Networking*, 31(4):1765–1778.
- Zhao, H., Zhang, H., and Zhao, Y. (2023). Yolov7-sea: Object detection of maritime uav images based on improved yolov7. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 233–238.
- Zhao, M., Cheng, L., Yang, X., Feng, P., Liu, L., and Wu, N. (2019). Tbc-net: A real-time detector for infrared small target detection using semantic constraint. *arXiv preprint arXiv:2001.05852*.

