Object Detection for Rescue Operations by High-altitude Infrared Thermal Imaging Collected by Unmanned Aerial Vehicles

Andrii Polukhin*[1][0009−0000−7650−5185], Yuri Gordienko[1][0000−0003−2682−4668], Gert Jervan[2][0000−0003−2237−0187], and Sergii Stirenko[1][0000−0001−5478−0450]

[1] National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine
*pandrii000@gmail.com
[2] Tallinn University of Technology, Tallinn, Estonia
gert.jervan@taltech.ee

Abstract. The analysis of the object detection deep learning model YOLOv5, which was trained on High-altitude Infrared Thermal (HIT) imaging, captured by Unmanned Aerial Vehicles (UAV) is presented. The performance of the several architectures of the YOLOv5 model, specifically 'n', 's', 'm', 'l', and 'x', that were trained with the same hyperparameters and data is analyzed. The dependence of some characteristics, like average precision, inference time, and latency time, on different sizes of deep learning models, is investigated and compared for infrared HIT-UAV and standard COCO datasets. The results show that degradation of average precision with the model size is much lower for the HIT-UAV dataset than for the COCO dataset which can be explained that a significant amount of unnecessary information is removed from infrared thermal pictures ("pseudo segmentation"), facilitating better object detection. According to the findings, the significance and value of the research consist in comparing the performance of the various models on the datasets COCO and HIT-UAV, infrared photos are more effective at capturing the real-world characteristics needed to conduct better object detection.

Keywords: Deep Learning · Object Detection · You Only Look Once · YOLO · Average Precision · AP · Unmanned Aerial Vehicles · UAV · Infrared Thermal Imaging

1 Introduction

Unmanned aerial vehicles (UAVs) are frequently used in many different industries, such as emergency management [1], mapping [2], traffic surveying [3], and environment monitoring [4]. Since UAVs can now load artificial intelligence (AI) algorithms as edge computing devices [5], the utility of the aforementioned applications has increased with the development of deep learning and edge computing

[6]. The rapid expansion of object detection applications has prompted numerous broad datasets to be proposed to boost algorithm training and evaluation [7,8,9,10].

This paper proposes the use of unmanned aerial vehicles (UAVs) equipped with thermal infrared (IR) cameras to locate missing individuals in wilderness settings. Detecting abandoned individuals using standard UAV technology, which lacks IR capabilities, is challenging due to a variety of factors, such as terrain, temperature, and obstacles. Infrared thermal imaging has been identified as a promising method for enhancing object detection in adverse weather conditions and challenging environments [11]. Moreover, the use of IR-equipped UAVs has the potential to facilitate rescue operations in conditions such as complete darkness, fog, and heavy rain [12,1]. However, outdated neural network models, which can result in reduced performance, inaccurate positive and false negative predictions, and difficulties in running the software on current and continuously evolving hardware, are the primary limitations of previous studies.

The significance and value of our research are to provide insights into the potential use of infrared thermal imaging on UAVs for object detection in challenging weather conditions, particularly in rescue operations. Special attention will be paid to the investigation of the dependence of some characteristics (average precision, inference time, and latency time) on different sizes of deep learning models, comparing infrared and standard datasets. This is especially important for understanding the feasibility of the usage of relatively small models for Edge Computing devices with regard to the deterioration of their performance with a decrease of model sizes in various applications described in our previous publications [13,14,15].

2 Background and Related Work

The deep learning methodology has sped up the development of the object detection field in recent years. The development of object detection apps has been facilitated by large datasets for object detection [16,17,18]. Over time, we' ve seen advancements in the accuracy and overall performance of object detection systems that have allowed apps to identify and classify objects more accurately.

Many datasets of aerial perspective were presented for the AI job with the UAV platform with the development of AI and the deployment of UAVs for many domains such as forest fire prevention [19], traffic monitoring [3], disaster assistance [20], and package delivery [21]. We can effectively employ object detection to save more people by using infrared imaging.

2.1 Neural Network Object Detection Methods

Convolutional neural networks (CNNs) and large-scale GPU processing have enabled deep learning to achieve remarkable success in modern computer vision [22,23,24,25]. CNNs, which concentrate on processing spatially local input to learn the visual representation, are now the de facto method for a variety of vision-related tasks. An object detection model typically consists of two parts: a backbone that extracts features from the image, and a head that predicts object classes and bounding boxes. The choice of backbone architecture for object detectors often depends on the complexity of the model and the platform on which it is intended to be run. For instance, architectures such as VGG [23] or ResNet [24] are typically employed as the backbone for detectors that run on GPU platforms due to their higher computational demands. On the other hand, models such as MobileNet [25] may serve as suitable backbones for detectors that run on CPU platforms, as they have lower computational complexity and are thus better suited for resource-constrained settings.

There are two main architectures of the head: one-stage and two-stage options. The R-CNN series, which includes the Fast R-CNN [26], and Faster R-CNN [27], is the most typical two-stage object detector. The most typical models for one-stage object detectors are YOLO [28] and SSD [29].

The YOLO model [28] family architecture is one of the best object detection algorithms known for its speed and accuracy, which can be pre-trained using the COCO dataset [30]. YOLO applies one neural network to divide the picture into areas and forecast probability and bounding boxes. The architecture of a single-stage object detector like YOLO consists of three parts: backbone, neck, and head. YOLO v5 uses CSPNet [31] as its backbone to extract important features, PANet [32] as its neck to produce feature pyramids, and a similar head to that of YOLO v4 [31] to carry out the final detection step.

2.2 Infrared Object Detection

Apart from neural network approaches [33], there are several traditional methods available for identifying objects in infrared images [34,35,36,37]. These methods primarily focus on distinguishing between three elements in infrared images: the object, the background, and the image noise. The main objective is to suppress the background and noise to enhance the object and identify it using various techniques. One such algorithm [34] employs a spatial filtering-based technique for infrared object detection, searching for various background and object gray values. The background is then selected and suppressed to enable the identification of the object. Another technique [35] incorporates shearlet-based histogram thresholding and is based on a practical image denoising approach, offering significant improvement but with a high computational cost. Traditional infrared object identification techniques often use artificially created feature extractors such as Haar [36] or HOG [37], which are effective but not robust to shifts in object diversity.

2.3 UAV Infrared Thermal Datasets

The use of UAVs equipped with infrared thermal cameras can significantly enhance mission accuracy while reducing costs and resource requirements, particularly when dealing with large volumes of data. In this context, several existing datasets have been developed for Infrared Thermal UAV object detection tasks.

For instance, the HIT-UAV dataset [7] comprises 2898 infrared thermal images extracted from 43470 frames captured by a UAV in various scenes, including information such as flight altitude, camera perspective, and daylight intensity. The FLAME dataset [8], on the other hand, includes raw heatmap footage and aerial movies captured by drone cameras, and is used for defining two well-known studies: fire classification and fire segmentation. The PTB-TIR dataset [9], containing 60 annotated sequences with nine attribute labels for pedestrian tracking, is commonly used for evaluating thermal infrared pedestrian trackers. Finally, the BIRDSAI dataset [10] is a long-wave thermal infrared dataset for Surveillance with Aerial Intelligence, which contains images of people and animals at night in Southern Africa, along with actual and fake footage to enable testing of algorithms for the autonomous detection and tracking of people and animals. Although not used in the present study, the availability of these datasets contributes to the development and evaluation of new infrared thermal UAV object detection methods.

2.4 Object Detection Metrics

Supervised object detection methods have recently produced outstanding results, leading to a demand for annotated datasets for their evaluation. A good object detector should locate all ground truth objects with high recall and recognize only relevant objects with high precision, and an ideal model would have high precision with increasing recall. Average precision (AP) summarizes the precision-recall trade-off based on expected bounding box confidence levels. Formally, the equation of AP is defined as follows:

$$AP = \int_0^1 p(r)dr,$$

where p and r are precision and recall values for the same threshold correspondingly.

3 Methodology

We conducted an analysis of the HIT-UAV dataset, which includes five categories of annotated objects: Person, Car, Bicycle, Other Vehicle, and DontCare. We used the YOLOv5 object detection algorithm and split the dataset into three sets for training, validation, and testing. Our analysis showed that the Car, Bicycle, and Person categories make up the majority of the dataset. We utilized all five categories for training but only evaluated the top three due to the shortage of training data for the OtherVehicle and DontCare categories. We also analyzed the distribution of instances across annotated object categories per image and found that most images contain fewer than 10 instances, with some outliers having more than 30 instances per image. We used all available YOLOv5 architectures and evaluated their performance with two GPUs Tesla T4. We also used various augmentation techniques, such as hue, saturation, and value modifications, random translations and scaling, horizontal flipping, and mosaic image generation, to improve the model's robustness and prevent overfitting. We provide more details on the analysis in this section.

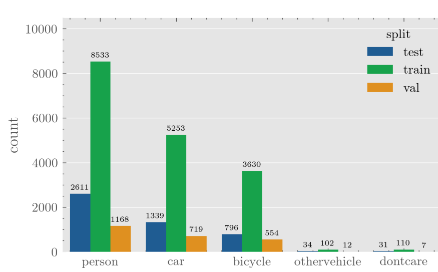
3.1 Exploratory Data Analysis (EDA)

We trained and evaluated object detection algorithms on the HIT-UAV dataset. The dataset was split into three sets: 2008 photos for training, 287 images for validation, and 571 images for testing. In total, the training set consisted of 17,628 instances, the validation set had 2,460 instances, and the testing set contained 4,811 instances, introducing this statistics in Table 1.

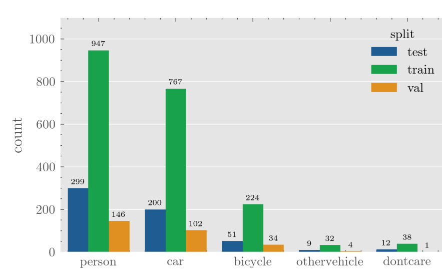
Subset	Images	Instances
Train	2008	17628
Validation	287	2560
Test	571	4811

Table 1: Dataset size for train, test, and validation subsets.

The HIT-UAV dataset includes five categories of annotated objects: Person, Car, Bicycle, and Other Vehicle, which are frequently observed in rescue and search operations. Additionally, there is an ”unidentifiable” category called DontCare, which is used for objects that cannot be assigned to specific classes by an annotator, particularly in cases where the objects appear in high-altitude aerial photos. It can be challenging to determine whether these objects contain something of importance, but there may be an important object present.



(a) Distribution of instances per category.



(b) Distribution of images per category.

Fig. 1: Distribution of the categories across the instances and images.

We conducted an analysis of the distribution of the five annotated object categories across instances and images, as depicted in Figure 1a and Figure 1b, respectively. Our analysis revealed that the Car, Bicycle, and Person categories make up the majority of the dataset. For subsequent experiments, all five categories were used for training, but only the top three categories were used for evaluation. This is because there is a shortage of training data for the OtherVehicle and DontCare categories, which limited the model's ability to learn them effectively. Consequently, the model performed poorly for these categories, resulting in an underestimation of the average AP metric across all categories.

We analyzed the distribution of instances across all annotated object categories per image, as shown in Figure 2. Our analysis indicates that the majority of images contain fewer than 10 instances. However, there are some outliers with more than 30 instances per image, which likely correspond to crowded locations with numerous people.

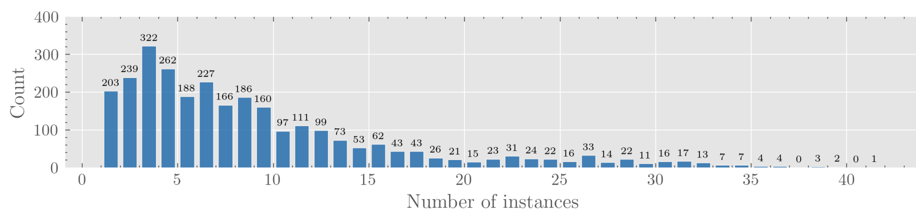
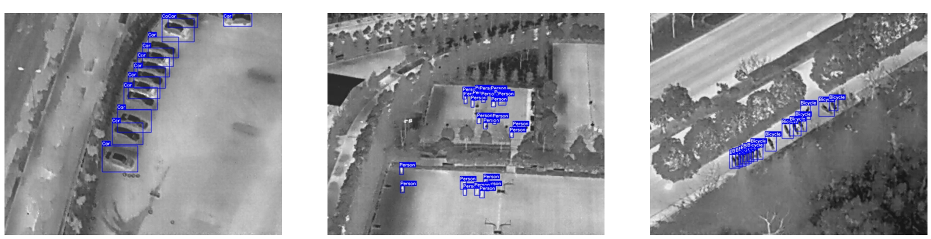


Fig. 2: Distribution of the instances per image.

Also, we have visualized the annotations for the most labeled classes Car in Figure 3a, Person in Figure 3b, and Bicycle in Figure 3c.



(a) Class ”Car”. (b) Class ”Person”. (c) Class ”Bicycle”.

Fig. 3: Example of the annotated classes ”Car”, ”Person”, and ”Bicycle”.

3.2 Model Selection

In order to ensure reliable and reproducible training results, we utilized the standard and freely available YOLOv5 single-stage object detector in conjunction with the open HIT-UAV dataset. YOLOv5 offers five different sizes of architecture, ranging from the extra small (nano) size model denoted by 'n', to the small ('s'), medium ('m'), large ('l'), and extra large ('x') models.

As shown in Figure 4, the number of parameters for each model size increases nearly exponentially.

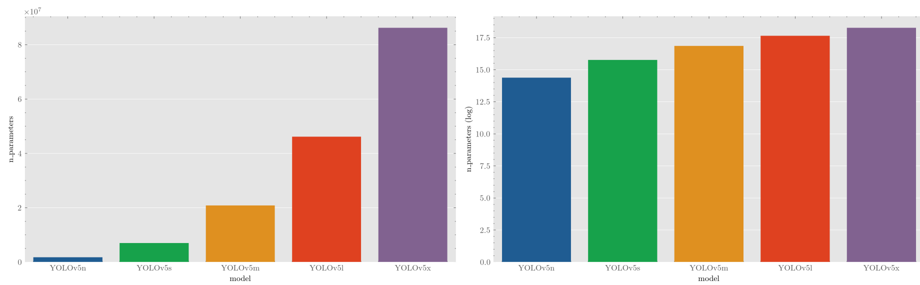


Fig. 4: Number of parameters (left) and log of the number of parameters (right) of the different YOLO v5 sizes.

3.3 Experimental Workflow

The training time required for each variant varies. We conducted an analysis of all the variants using the same dataset, augmentation hyperparameters, training configuration, and hardware. To evaluate performance, we measured the average precision, inference time (ms/image), latency time (ms/image), and frames per second for each category, as well as an average across "Person", "Car", and "Bicycle" categories.

Using the free computational resources, offered by the Kaggle Platform, we trained and evaluated the performance of all YOLO v5 architectures with two GPUs Tesla T4, which have 15 GB of memory, or 30 GB in total, CUDA Version 11.4. All models were trained to similar hyperparameters, which provided 300 training iterations and a batch size of 28. SGD is the optimizer, with a weight decay of 0.0005, a learning rate of 0.01, and a momentum of 0.937. Priorities are set at 0.05 for the box loss function, 0.5 for classification loss, and 0.1 for objectness loss. The thresholding value for IOUs is 0.2.

The YOLOv5 model employs a range of augmentation techniques to improve the performance of object detection. These techniques include the modification of hue, saturation, and value channels, random translations and scaling, horizontal flipping, and generating mosaic images. The hue channel is modified with a portion of 1.5%, while the saturation and value channels are adjusted up to 70% and 40% relative to the original value, respectively. The image is randomly translated up to 10% of its height and width, and the image size is scaled up to 50% of its original size. Additionally, the model applies horizontal flipping with a 50% chance to the image. Finally, the model generates mosaic images by combining multiple images, which is always used during training. These augmentation techniques enhance the model's robustness to different scenarios and prevent overfitting by increasing the diversity of the training dataset.

The hyperparameter values and training setup is aggregated and introduced in Table 2.

Parameter	**Value**	**Augmentation**
GPUs	2 × Tesla T4	-
Memory	30 GB	-
CUDA Version	11.4	-
Optimizer	SGD	-
Weight decay	0.0005	-
Learning rate	0.01	-
Momentum	0.937	-
Loss function	-	
Box	0.05	-
Classification	0.5	-
Objectness	0.1	-
IOU threshold	0.2	-
Image modifications	-	
Hue	1.5%	-
Saturation	70%	-
Value	40%	-
Random translation	-	Up to 10% of height and width
Image scaling	-	Up to 50% of original size
Horizontal flipping	-	50% chance
Mosaic images	-	Always

Table 2: Training setup and hyperparameters for YOLOv5 models

In addition, we compare our results with YOLOv5 models trained on the COCO 2017 dataset [38]. The models were trained for 300 epochs using A100 GPUs at an image size of 640. The training process utilized an SGD optimizer with a learning rate of 0.01 and a weight decay of 0.00005. Our training setup closely follows this configuration. These models are used to evaluate the performance of real and infrared thermal imaging.

4 Results and their Discussion

Our evaluation of the HIT-UAV dataset reveals that YOLOv5 achieves an impressive average precision, as demonstrated by the results presented in Table 3.

We have compared the results of our model trained on the HIT-UAV dataset with the one trained on the COCO dataset. YOLOv5 (x) outperforms other variants in terms of average precision (AP), as evident from Figure 5a and Figure 5b. However, it is observed that YOLOv5 (x) has the longest inference and latency times. YOLOv5 (n), on the other hand, exhibits the fastest inference and latency times but performs poorly in terms of both AP and FPS. YOLOv5 (l) and YOLOv5 (m) demonstrate an optimal balance between accuracy and speed, making them a preferable choice for practical applications. Alternatively, YOLOv5 (s) can be used when speed is of higher priority over accuracy.

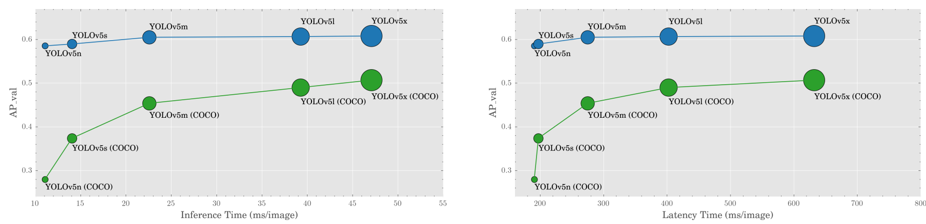
Model	**Dataset**	**Person**	**Car**	**Bicycle**	**FPS**	**AP**
YOLO v5 (n)	HIT-UAV	49.4%	75.4%	50.8%	90	58.5%
YOLO v5 (s)	HIT-UAV	50.8%	73.6%	52.5%	71	58.9%
YOLO v5 (m)	HIT-UAV	**51.0%**	75.1%	55.4%	44	60.5%
YOLO v5 (l)	HIT-UAV	50.9%	**75.7%**	55.4%	25	60.6%
YOLO v5 (x)	HIT-UAV	50.0%	75.4%	**57.0%**	21	**60.8%**
YOLO v5 (n)	COCO	-	-	-	90	28.0%
YOLO v5 (s)	COCO	-	-	-	71	37.4%
YOLO v5 (m)	COCO	-	-	-	44	45.4%
YOLO v5 (l)	COCO	-	-	-	25	49.0%
YOLO v5 (x)	COCO	-	-	-	21	50.7%

Table 3: The evaluation for the YOLO v5.

Across all YOLOv5 size ranges, the results obtained from the HIT-UAV dataset suggest that the Car category's AP value is significantly higher than those of other categories. This could be attributed to YOLOv5's superior detection capabilities for large objects such as automobiles, as opposed to relatively smaller objects like bicycles or persons. In highly crowded images, the "Person" category's AP value is not as high as that of "Car." This is because the YOLOv5 algorithm encounters significant difficulties in such scenarios, where the model starts to miss some of the smaller features. Specifically, the model struggles to perform well in very cluttered images.

The COCO results indicate that the original YOLOv5 (n) model achieves an AP of 28%. However, when trained on the HIT-UAV dataset, the YOLOv5 (n) model reaches an AP of 58%. This suggests that aerial image information is more advantageous for detection tasks than natural photos. It is worth noting that while infrared photos are more effective in capturing certain real-world characteristics required for better object recognition, their advantage is limited to certain classes, such as detecting persons who emit heat or vehicles with metal signatures highlighted in infrared images. For other classes, RGB images could be more beneficial.

The above results indicate the following observations:



(a) Inference time (ms/image). (b) Latency time (ms/image).

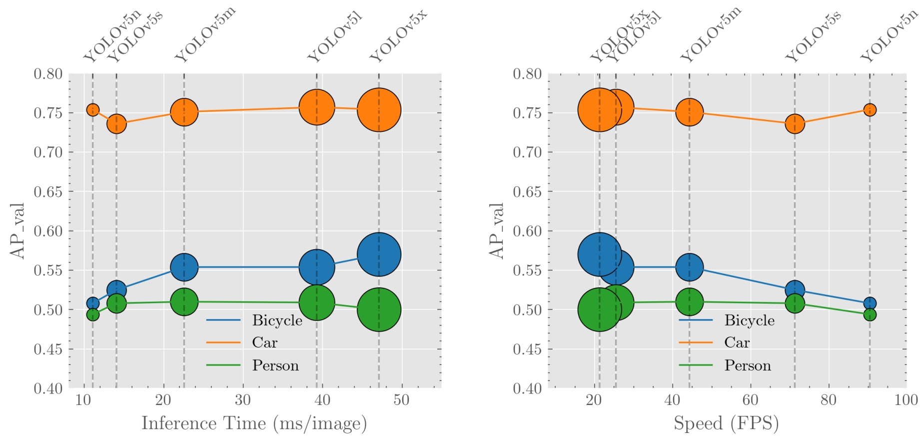
Fig. 5: Comparison of the inference and latency time (ms/image) vs average precision of YOLOv5 models trained on the HIT-UAV and COCO datasets for different model sizes on GPU Tesla T4.

1. A significant amount of unnecessary information is removed from infrared thermal pictures, providing a "pseudo segmentation" effect and facilitating object detection. Due to the clearly visible properties of the objects in infrared thermal pictures, the typical detection model may achieve great recognition performance with few images.
2. In the infrared thermal aerial photos, we can see that the model does a good job of capturing large objects. Small objects, including persons, might easily be mistaken for infrared sensor noise, which raises the false positive rate.

The first observation is supported by our previous results on the impact of ground truth annotation (GT) quality on the performance of semantic image segmentation of traffic conditions, where the mean accuracy values of semantic image segmentation for coarse GT annotations are higher than for the fine GT ones [39,40]. The infrared images give similar coarse representations of objects in comparison to their actual visual appearance.

A visual comparison of the bounding box predictions for the trained models is also provided in Figure 7.

The research shows that YOLOv5 performs better on the HIT-UAV dataset than on COCO, indicating the advantages of using IR aerial imagery over RGB imagery for object detection tasks for certain classes. However, it should be noted that HIT-UAV has a much smaller number of images and classes than COCO, making it an easier benchmark. Additionally, the HIT-UAV dataset only includes a limited number of classes, which may have influenced the reported performance difference between the two datasets. Also, the comparison between the two datasets is not entirely fair due to differences in IoU thresholds and image perspective. Therefore, it is suggested that in the future, the HIT-UAV dataset should be expanded with a larger number of classes to better represent real-world scenarios.



(a) Inference time (ms/image). (b) Speed (FPS).

Fig. 6: Comparison of the inference time (ms/image) and speed (FPS) vs average precision on different object categories of different model sizes on GPU Tesla T4 for HIT-UAV.

5 Conclusion

This paper presents an analysis of object detection training using the YOLOv5 architecture with an infrared thermal UAV dataset. The dataset comprises five categories of objects with varying sizes, locations, and numbers per image. In this study, YOLOv5 models of sizes (n), (s), (m), (l), and (x) were trained and tested using the HIT-UAV dataset. Our findings reveal that the HIT-UAV dataset shows a lower degradation of average precision with model size compared to the COCO dataset. This can be attributed to the removal of extraneous information from infrared thermal images, resulting in better object detection performance for certain classes, such as persons who emit heat or vehicles with metal signatures highlighted in infrared images. These findings have significant implications for using small deep learning models on Edge Computing devices for rescue operations that utilize HIT-UAV-like imagery, as their performance deterioration decreases with a decrease in model size. Moreover, our study shows that infrared thermal images significantly enhance object detection capabilities by filtering out unnecessary information and improving the recognition of certain classes compared to visual light images. However, for other classes, RGB images may be more effective. These benefits increase the feasibility of autonomous object detection using UAVs in crucial nighttime activities, such as city surveillance, traffic control, and person search and rescue. Further research is necessary to validate these results and expand the dataset with more images and categories to better represent real-world scenarios and facilitate a fair comparison of the model performance with HIT-UAV and COCO 2017 datasets. Overall, this algo-

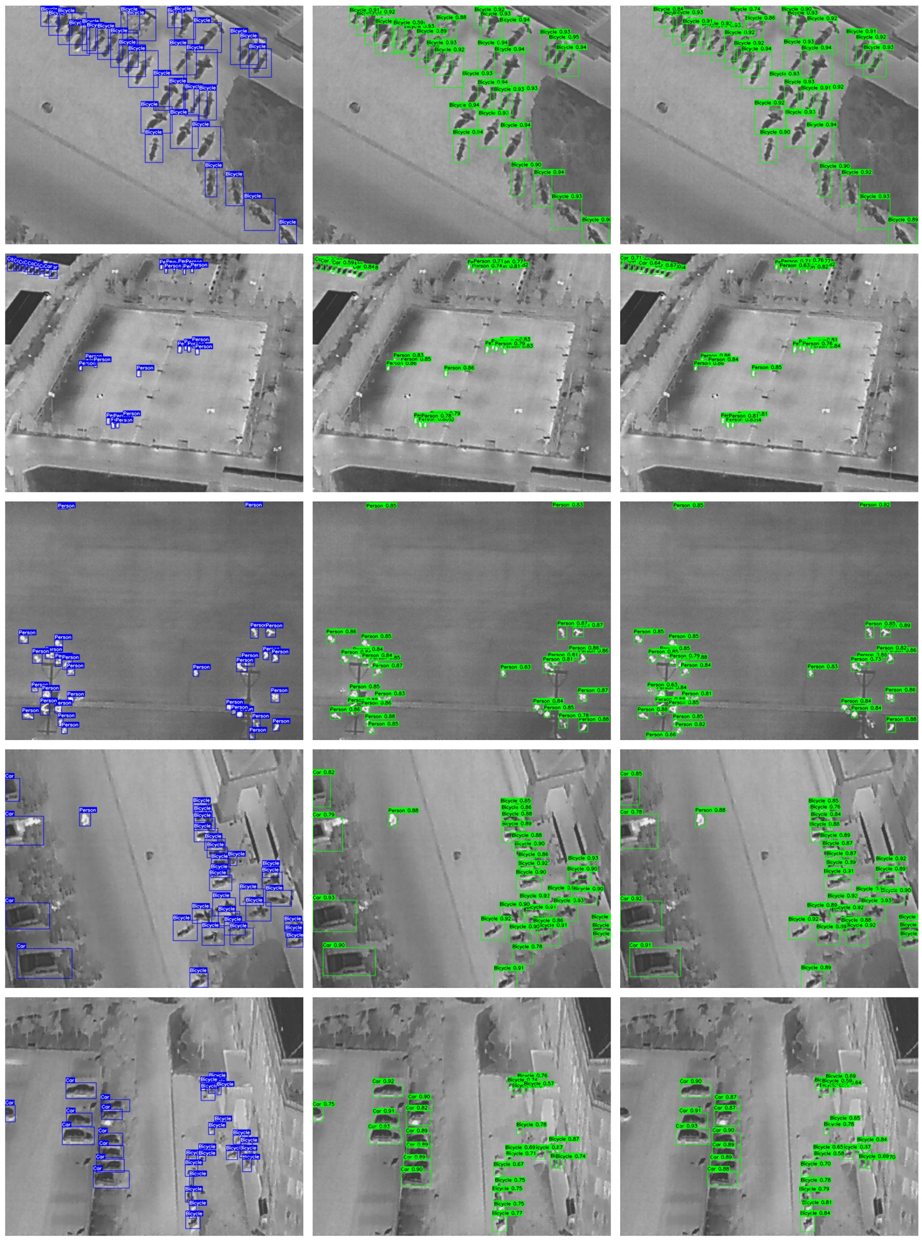


Fig. 7: Sample results of the YOLO v5. Left is the original annotations, the middle is the smallest YOLO v5 (n), and the right is the largest YOLO v5 (x).

rithm's evaluation can contribute to the development of efficient object detection techniques for nighttime applications.

6 Acknowledgements

This research was in part sponsored by the NATO Science for Peace and Security Programme under grant id. G6032.

References

1. Piero Boccardo, Filiberto Chiabrando, Furio Dutto, Fabio Tonolo, and Andrea Lingua. UAV Deployment Exercise for Mapping Purposes: Evaluation of Emergency Response Applications. *Sensors*, 15(7):15717–15737, July 2015.
2. Ana de Castro, Jorge Torres-Sánchez, Jose Peña, Francisco Jiménez-Brenes, Ovidiu Csillik, and Francisca López-Granados. An Automatic Random Forest-OBIA Algorithm for Early Weed Mapping between and within Crop Rows Using UAV Imagery. *Remote Sensing*, 10(3):285, 2018.
3. Konstantinos Kanistras, Goncalo Martins, Matthew J. Rutherford, and Kimon P. Valavanis. A survey of unmanned aerial vehicles (UAVs) for traffic monitoring. *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 221–234, May 2013.
4. Danilo Avola, Gian Luca Foresti, Niki Martinel, Christian Micheloni, Daniele Pannone, and Claudio Piciarelli. Aerial video surveillance system for small-scale UAV environment monitoring. *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, August 2017.
5. Qian Liu, Long Shi, Linlin Sun, Jun Li, Ming Ding, and Feng Shu Shu. Path Planning for UAV-Mounted Mobile Edge Computing With Deep Reinforcement Learning. *IEEE Transactions on Vehicular Technology*, 69(5):5723–5728, May 2020.
6. Fangxin Wang, Miao Zhang, Xiangxiang Wang, Xiaoqiang Ma, and Jiangchuan Liu. Deep Learning for Edge Computing Applications: A State-of-the-Art Survey. *IEEE Access*, 8:58322–58336, 2020.
7. Jiashun Suo, Tianyi Wang, Xingzhou Zhang, Haiyang Chen, Wei Zhou, and Weisong Shi. HIT-UAV: A High-altitude Infrared Thermal Dataset for Unmanned Aerial Vehicles, April 2022.
8. Alireza Shamsoshoara. The FLAME dataset: Aerial Imagery Pile burn detection using drones (UAVs), November 2020.
9. Qiao Liu, Zhenyu He, Xin Li, and Yuan Zheng. PTB-TIR: A Thermal Infrared Pedestrian Tracking Benchmark. *IEEE Transactions on Multimedia*, 22(3):666–675, March 2020.
10. Elizabeth Bondi, Raghav Jain, Palash Aggrawal, Saket Anand, Robert Hannaford, Ashish Kapoor, Jim Piavis, Shital Shah, Lucas Joppa, Bistra Dilkina, and Milind Tambe. BIRDSAI: A Dataset for Detection and Tracking in Aerial Thermal Infrared Videos. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1736–1745, March 2020.
11. Jürgen Beyerer, Miriam Ruf, and Christian Herrmann. CNN-based thermal infrared person detection by domain adaptation. In Michael C. Dudzik and Jennifer C. Ricklin, editors, *Autonomous Systems: Sensors, Vehicles, Security, and the Internet of Everything*, page 8, Orlando, United States, May 2018. SPIE.

12. E. Levin, A. Zarnowski, J. L. McCarty, J. Bialas, A. Banaszek, and S. Banaszek. Feasibility study of inexpensive thermal sensors and small uas deployment for living human detection in rescue missions application scenarios. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLI-B8:99–103, June 2016.
13. Yuri Gordienko, Yuriy Kochura, Vlad Taran, Nikita Gordienko, Alexandr Rokovyi, Oleg Alienin, and Sergii Stirenko. Scaling analysis of specialized tensor processing architectures for deep learning models. *Deep learning: Concepts and architectures*, pages 65–99, 2020.
14. Yuri Gordienko, Yuriy Kochura, Vlad Taran, Nikita Gordienko, Oleksandr Rokovyi, Oleg Alienin, and Sergii Stirenko. "last mile" optimization of edge computing ecosystem with deep learning models and specialized tensor processing architectures. In *Advances in computers*, volume 122, pages 303–341. Elsevier, 2021.
15. Vlad Taran, Yuri Gordienko, Oleksandr Rokovyi, Oleg Alienin, Yuriy Kochura, and Sergii Stirenko. Edge intelligence for medical applications under field conditions. In *Advances in Artificial Systems for Logistics Engineering*, pages 71–80. Springer, 2022.
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, February 2015.
17. Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, June 2012.
18. 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, June 2010.
19. S. Sudhakar, V. Vijayakumar, C. Sathiya Kumar, V. Priya, Logesh Ravi, and V. Subramaniyaswamy. Unmanned Aerial Vehicle (UAV) based Forest Fire Detection and monitoring for reducing false alarms in forest-fires. *Computer Communications*, 149:1–16, January 2020.
20. Horea Bendea, Piero Boccardo, S Dequal, Fabio Giulio Tonolo, Davide Marenchino, and Marco Piras. Low cost UAV for post-disaster assessment. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37, January 2008.
21. John Gunnar Carlsson and Siyuan Song. Coordinated Logistics with a Truck and a Drone. *Management Science*, 64(9):4052–4069, September 2018.
22. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.
23. Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition, April 2015.
24. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*, page 12, December 2015.
25. Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv:1704.04861 [cs]*, page 9, April 2017.
26. Ross Girshick. Fast R-CNN. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, December 2015.

27. Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv:1506.01497 [cs]*, page 14, January 2016.
28. Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 10, Las Vegas, NV, USA, June 2016. IEEE.
29. Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single Shot MultiBox Detector. *arXiv:1512.02325 [cs]*, 9905:17, 2016.
30. Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, volume 8693, pages 740–755, Cham, 2014. Springer International Publishing.
31. Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. YOLOv4: Optimal Speed and Accuracy of Object Detection, April 2020.
32. Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path Aggregation Network for Instance Segmentation, September 2018.
33. Shasha Li, Yongjun Li, Yao Li, Mengjun Li, and Xiaorong Xu. YOLO-FIRI: Improved YOLOv5 for Infrared Image Object Detection. *IEEE Access*, 9:141861–141875, 2021.
34. Zhao Kun. Background Noise Suppression in Small Targets Infrared Images and Its Method Discussion. *Optics & Optoelectronic Technology*, 2004.
35. T S Anju and N R Nelwin Raj. Shearlet transform based image denoising using histogram thresholding. *2016 International Conference on Communication Systems and Networks (ComNet)*, pages 162–166, July 2016.
36. Paul Viola and Michael J. Jones. Robust Real-Time Face Detection. *International Journal of Computer Vision*, 57(2):137–154, May 2004.
37. N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, June 2005.
38. Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, Kalen Michael, TaoXie, Jiacong Fang, imyhxy, Lorna, 曾逸夫 (Zeng Yifu), Colin Wong, Abhiram V, Diego Montes, Zhiqiang Wang, Cristi Fati, Jebastin Nadar, Laughing, UnglvKitDe, Victor Sonck, tkianai, yxNONG, Piotr Skalski, Adam Hogan, Dhruv Nair, Max Strobel, and Mrinal Jain. Ultralytics/yolov5: V7.0 - YOLOv5 SOTA Realtime Instance Segmentation, November 2022.
39. Vlad Taran, Nikita Gordienko, Yuriy Kochura, Yuri Gordienko, Alexandr Rokovyi, Oleg Alienin, and Sergii Stirenko. Performance evaluation of deep learning networks for semantic segmentation of traffic stereo-pair images. In *Proceedings of the 19th International Conference on Computer Systems and Technologies*, pages 73–80, 2018.
40. Vlad Taran, Yuri Gordienko, Alexandr Rokovyi, Oleg Alienin, and Sergii Stirenko. Impact of ground truth annotation quality on performance of semantic image segmentation of traffic conditions. In *Advances in Computer Science for Engineering and Education II*, pages 183–193. Springer, 2020.