SAVING LIVES FROM ABOVE: PERSON DETECTION IN DISASTER RESPONSE USING DEEP NEURAL NETWORKS

Reza Bahmanyar,[*] Nina Merkle

Remote Sensing Technology Institute, German Aerospace Center (DLR), Wessling, Germany
(reza.bahmanyar, nina.merkle)@dlr.de

KEY WORDS: Aerial Imagery, Deep Neural Networks, Drone Imagery, Person Detection, Small Object Detection, UAVs

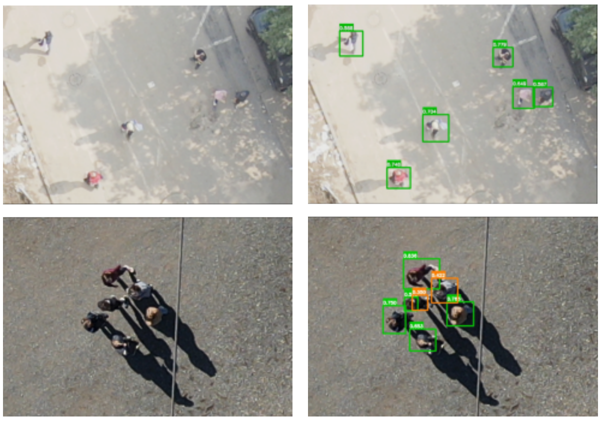
ABSTRACT:

This paper focuses on person detection in aerial and drone imagery, which is crucial for various operations such as situational awareness, search and rescue, and safe delivery of supplies. We aim to improve disaster response efforts by enhancing the speed, safety, and effectiveness of the process. Therefore, we introduce a new person detection dataset comprising 311 annotated aerial and drone images, acquired from helicopters and drones in different scenes, including urban and rural areas, and for different scenarios, such as estimation of damage in disaster-affected zones, and search and rescue operations in different countries. The amount of data considered and level of detail of the annotations resulted in a total of 10,050 annotated persons. To detect people in aerial and drone images, we propose a multi-stage training procedure to improve YOLOv3's ability. The proposed procedure aims at addressing challenges such as variations in scenes, scenarios, people poses, as well as image scales and viewing angles. To evaluate the effectiveness of our proposed training procedure, we split our dataset into a training and a test set. The latter includes images acquired during real search and rescue exercises and operations, and is therefore representative for the challenges encountered during operational missions and suitable for an accurate assessment of the proposed models. Experimental results demonstrate the effectiveness of our proposed training procedure, as the model's average precision exhibits a relevant increase with respect to the baseline value.

1. INTRODUCTION

In recent years, the use of aerial imagery from aircraft and drones for disaster monitoring, management, and search and rescue missions has increased significantly. This advanced technology has proven to be highly effective in providing real-time, high-resolution images and videos of disaster-affected areas, enabling first responders and emergency personnel to assess the damage and critical aspects of the situation and respond quickly. The use of aerial imagery significantly minimizes the time and effort required to locate affected areas and individuals, facilitating rapid and effective intervention by scanning large areas in a short period of time. These data also provide rescue teams with vital information about the path to follow during search and rescue operations, and enabling effective decision making and resource allocation during disaster management, resulting in more efficient and effective response efforts. This has helped optimizing the use of resources and minimizing the risk of human casualties during critical situations (Mohd Daud et al., 2022).

In this paper, we focus in particular on the detection of people in aerial and drone imagery, which is often a critical requirement for situational awareness, search and rescue missions, safe delivery of supplies to affected areas, and safe landing of drones. We hope this will improve the speed, safety, and effectiveness of disaster response efforts and in turn help saving lives. Accurately detecting a person in large aerial and drone images with complex scenes is a challenging task due to several factors, including the very small size and pixel coverage of people in comparison to the large number of pixels in the image, the numerous poses and articulated nature of the human body, as well as variations in lighting and weather conditions, object occlusion, and changes in camera angles. In order to overcome these challenges and achieve higher accuracy in detecting people in large aerial and drone images with complex scenes, effective and appropriate algorithms and training data are required.

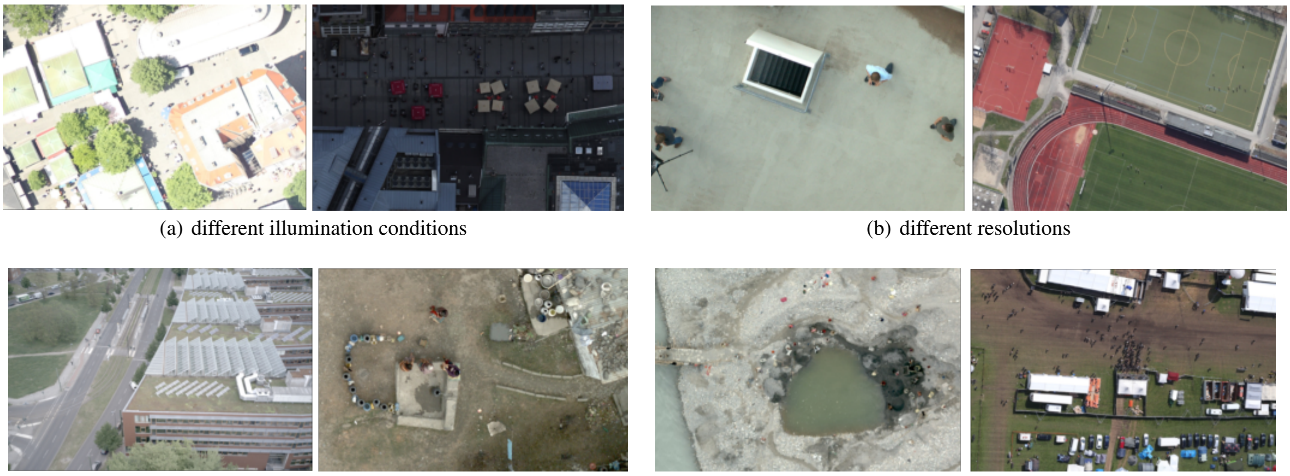


(a) Aerial/drone image (b) Predictions

Figure 1. Sample detection results. Bounding box colors are: predictions with 🟧 confidence < 0.5 and 🟩 confidence ≥ 0.5.

Advances in algorithms based on Deep Neural Networks (DNNs) in recent years have led to significant improvements in object detection performance. These include two-stage approaches such as the R-CNN (Region-based Convolutional Neural Network) family (Girshick et al., 2016, Girshick, 2015, Ren et al., 2015), one-stage approaches such as the YOLO (You Only Look Once) family (Redmon et al., 2016, Redmon and Farhadi, 2017, Redmon and Farhadi, 2018, Wang et al., 2022), and anchor-free approaches such as CenterNet (Zhou et al., 2019). One-stage object detectors have several advantages over two-

[*] Corresponding author



(a) different illumination conditions

(b) different resolutions

(c) different viewing angles

(d) different scenarios

Figure 2. Samples of the dataset showing different illumination conditions (a), resolutions (b), viewing angles (c), and scenarios (d).

stage object detectors including speed, simplicity, end-to-end training, and high accuracy. This makes them well-suited for a variety of applications, including object detection in autonomous driving, robotics, and surveillance systems.

At the beginning of our research, YOLOv3 (Redmon and Farhadi, 2018) was the most stable available variant of the YOLO family, and we chose it for our experiments and adaptation. Since then, however, the state of the art has evolved to YOLOv7 (Wang et al., 2022), which has proven to be faster and more accurate on benchmark datasets. We propose a multi-stage training procedure to enhance the capabilities of YOLOv3 in learning crucial features of people in aerial and drone images, enabling it to effectively handle variations in scenes, scenarios, people poses, and scales across a wide range of spatial resolutions and viewing angles. Although our analysis and suggestions are based on YOLOv3, they can also be applied to the newer variants of the YOLO family as the challenges posed by aerial and drone imagery often require adaptation of the methods, regardless of the specific variant employed.

Datasets play a crucial role in training and testing DNNs as they allow to identify patterns and relationships in the data, generalize to new scenarios, evaluate performance, and compare different techniques. However, collecting and annotating large aerial images can be a time-consuming and expensive process, especially for real-world applications such as disaster management and search and rescue, where privacy and security concerns related to the use of the data constitute additional hindrances. As a result, there is a lack of publicly available reference datasets for person detection in aerial imagery, thing which hinders the development of algorithms that can be effectively used in real-world applications.

In recent years, researchers have attempted to overcome this problem by creating new datasets. One example is the VisDrone2019 dataset (Zhu et al., 2021), a large-scale benchmark dataset containing over 2.6 million bounding boxes of pedestrians, cars, bicycles, and tricycles in drone images acquired over China. The VisDrone dataset has been used for various tasks, including detecting and tracking objects and estimating crowd density. The UAVHuman dataset (Li et al., 2021) was originally created for action recognition, enabling a better understanding of human behavior in UAV-related scenarios, and can also be used for person detection. The dataset includes multimodal video sequences and frames for action recognition, pose estimation, person re-identification, and attribute recognition from drones, acquired at an altitude ranging from 2 to 8 meters. Datasets such as DLR-ACD (Bahmanyar et al., 2019) were created for crowd counting and density estimation tasks, which can also be used for person detection. However, the lower spatial resolution of their images, as they are typically taken from higher altitudes, limits their suitability for person detection. Recently, the Manipal-UAV person detection dataset (Akshatha et al., 2023) was released, which contains 13,462 image samples from 33 videos taken at different flight altitudes in different locations and weather conditions. This dataset contains a total of 153,112 annotated persons. The SeaDronesSee dataset (Varga et al., 2022) was created specifically for search and rescue missions. It contains images of people in open water, and annotations have been created for both people detection and tracking tasks. The dataset was collected from various heights and angles of view, ranging from 5 to 260 meters.

While these datasets cover a wide range of altitudes and weather conditions and contain a large number of annotated people, none of them, except for SeaDronesSee, are specifically designed for real-world search and rescue missions. We believe that an appropriate dataset for disaster management and search and rescue missions should include images with a spatial resolution (Ground Sampling Distance or GSD) ranging from 1 to 6 cm/pixel, acquired from diverse relevant scenes and scenarios. This paper contributes a novel person detection dataset consisting of 311 annotated aerial and drone images captured by multiple flying vehicles, such as helicopters and drones, in various scenes and scenarios. The dataset includes both urban and rural environments, acquired in the frame of extreme events or search and rescue missions and exercises in multiple countries. The images were acquired from varying flight altitudes, ranging from 80 to 300 meters, and feature diverse GSDs from 0.2 to 6 cm/pix. The overview in Figure 4 provides the geographical distribution of the images, while Figure 2 demonstrates their diversity through a set of sample images. We performed a thorough manual process to annotate people in the images, resulting in a total of 10,050 annotated individuals. Figure 3 shows some examples of the annotation results.

In order to evaluate the performance of our proposed training procedure for YOLOv3, we divided the dataset into two subsets for training and testing. We use the training set to train

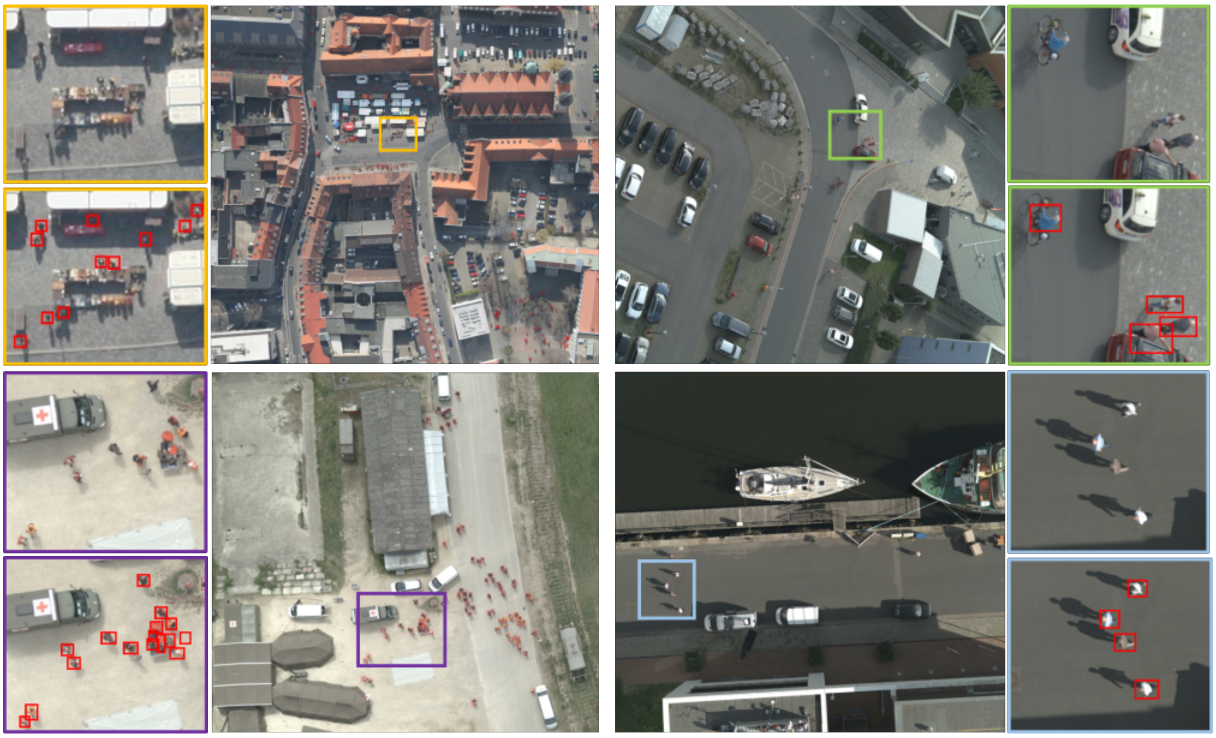


Figure 3. Samples of the annotated dataset. Each person is annotated with an individual bounding box. This figure shows image samples from the training set.

YOLOv3 according to our proposed procedure and assess its performance on the independent test set. Experimental results demonstrate the effectiveness of our training procedure and the suitability of our dataset for operational missions.

2. PERSON DETECTION DATASET

In order to train and evaluate our person detection method and to investigate the current challenges of real-world applications, we created a new dataset for person detection, containing 311 annotated aerial and drone images acquired by the 4K (Kurz et al., 2014) and MACS (Brauchle et al., 2019) camera systems mounted on helicopters and drones, respectively, as well as by the DJI's Phantom-4 and Movic-Pro drones. Image sizes vary between 4864 × 3232 px, 5184 × 3456 px, 5616 × 3744 px, and 8000 × 6000 px. Particular care was taken to select images having different GSD, cloud cover, and acquired with different weather conditions, sun positions, viewing angles, seasons, times of day, types of scenes (urban, suburban, rural, park, and recreation sites), and application scenarios (rescue, crowd events, construction). The selected images have a GSD between 0.2 and 6 cm/pix and were acquired between 2012 and 2022 over different regions in Germany, the Netherlands, Switzerland, Spain, France, and Nepal. Figure 4 gives an overview of the spatial distribution of the images, while Figure 2 illustrates the diversity of the images through several examples.

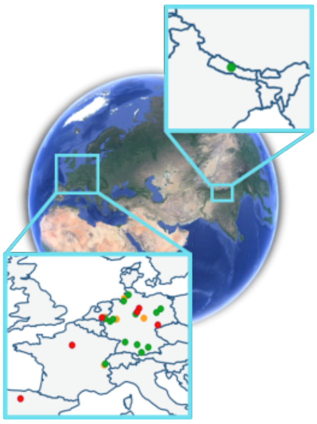
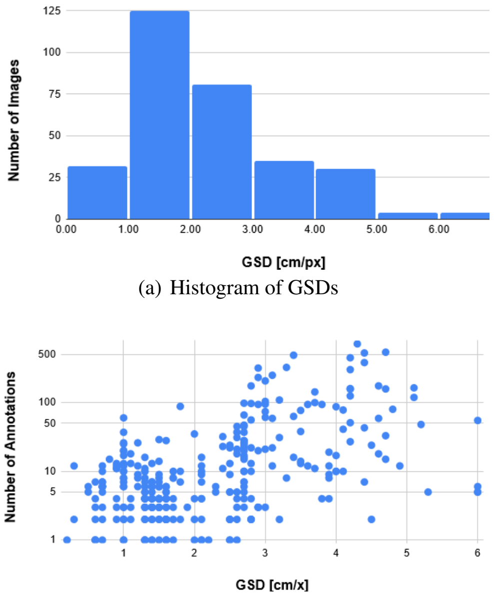


Figure 4. Illustration of the distribution of the aerial and drone images within the training 🟩, validation 🟧 and test 🟥 sets in Germany, the Netherlands, Switzerland, France, Spain, and Nepal. *Image source: Google Earth Pro; Data: SIO, NOAA, U.S. Navy, NGA, GEBCO; Image: IBCAO; Image: Landsat/Copernicus; 12/14/2015.*

We generated high-quality annotations for each image through a manual process using the CVAT annotation tool (Sekachev and et al., 2020). An annotation policy was created in order to define the main criteria of the annotation process, in order to ensure consistency across the dataset and to provide a basis for its further expansion. Each person was then annotated with a bounding box. We performed a multi-stage quality check in order to correct or remove erroneous annotations and add missing ones to guarantee the quality of the dataset, which contains in its final form a total of 10,050 annotated persons. Figure 3 shows a few examples of the annotations. In Figure 5, we present some statistics of the dataset. From the plots, it can be seen that the majority of the images have a smaller GSD size. While 78.8% of the images have a GSD≤ 3cm/pix, they contain only 25.5% of the annotations. This is due to the fact that the images having a higher GSD were taken from higher altitudes, things which allows them to cover larger areas and be acquired over populated areas, such as urban areas, where it is usually not allowed to fly at lower altitudes.

https://doi.org/10.5194/isprs-annals-X-1-W1-2023-343-2023 | © Author(s) 2023. CC BY 4.0 License.



(a) Histogram of GSDs

(b) Number of annotations vs. GSDs

Figure 5. Overview of statistical characteristics of the dataset: (a) histogram of the GSDs and (b) number of annotations vs. GSDs

In order to train, validate, and test our neural networks, we divided the 311 images into three disjoint sets: 1) the training set consisting of 259 images with 6934 annotations, 2) the validation set consisting of 25 images with 2706 annotations, and 3) the test set consisting of 27 images with 410 annotations. Figure 4 shows the distribution of images from different sets, where special care has been taken in order to select the test images from locations as different from one another as possible. To evaluate the ability of the developed algorithms in real-world situations, the test set includes urban scenes as well as images from search and rescue exercises and missions. In addition, there are a few images of the same region/area that are part of different sets, which have been acquired during different flight campaigns and at different points in time.

3. YOLOV3 FOR PERSON DETECTION

For the person detection procedure used in this paper, we follow YOLOv3 (Redmon and Farhadi, 2018) with some adaptations. YOLOv3 is a one-stage object detection CNN from the YOLO family, designed to achieve reasonable detection accuracy in a time and computationally efficient manner (Redmon and Farhadi, 2018). Darknet-53 is used as backbone for the feature extraction process. It consists of 52 convolutional layers, benefits from a residual structure, and downscales the input image 32 times. Darknet-53 has been shown to outperform its predecessor Darknet-19 (Redmon and Farhadi, 2017), ResNet-101, and ResNet-152 (He et al., 2016) in object detection. Figure 6 illustrates the network architecture of YOLOv3.

Darknet-53 is connected to several convolutional layers which upsample the extracted features in two steps. A bounding box prediction is done using three prediction heads at three spatial scales, 1/32, 1/16, and 1/8, inspired by feature pyramid networks. At each scale, the feature pixels are considered as a grid, and three bounding boxes are predicted for each one, resulting in tensors of $N \times N \times [3 * (4+1+C)]$, where 4 is for the bounding box offsets, 1 is for the objectness prediction, and C is the number of classes. Thus, the size of the predicting tensors and the number of predictions depends on the size of the input images. In our experiments, with an input of 416×416 pixels and only one class, we have tensors of $(13 \times 13 \times 18)$, $(26 \times 26 \times 18)$, and $(52 \times 52 \times 18)$, and 10,647 predicted bounding boxes. The final bounding boxes are then selected according to their confidence score and using the Non Maximum Suppression (NMS) technique to choose the best bounding box from multiple overlapping bounding boxes that may represent each object.

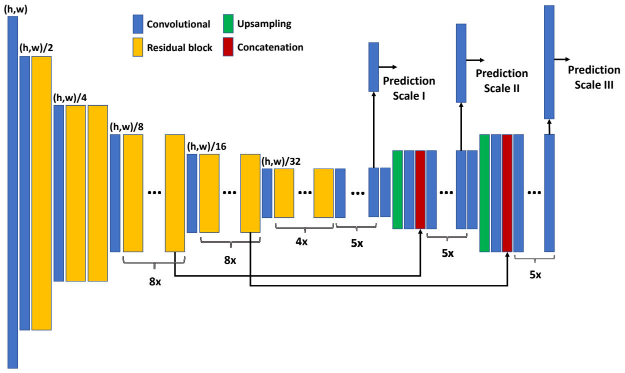


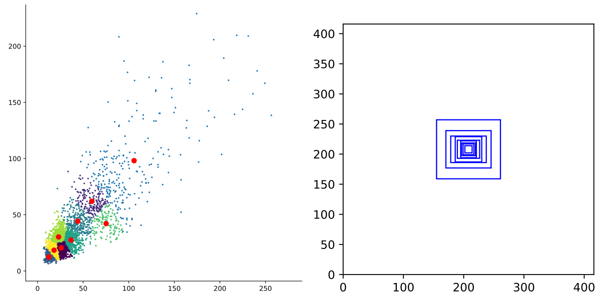
Figure 6. The network architecture of the YOLOv3.

3.1 Training procedure

In order to apply the mesh to the images, we split the images into patches of 416×416 pixels with a 10% overlap. This results in 35,991 training image patches. Due to the wide coverage of the images and the small size of single persons, the number of image patches without people, negative (Neg) patches, is more than 9× larger than the number of image patches with people, positive (Pos) patches, namely 32,471 versus 3,520. In addition, although 76.5% of the training image patches have a GSD better than 3 cm/pix, only 48.7% of the Pos image patches are within this GSD range, containing only 35% of the annotated people. Considering that the pose appearance of people in images with a GSD ≤ 3 cm/pix exhibits higher variability than in images with higher GSDs, the dataset naturally suffers from a large sample imbalance. This can cause the trained model to be biased towards the detection of smaller objects,i.e., the people in the images with a higher GSD.

In order to address these challenges, we split the training data into the images with a GSD ≤ 3 cm/pix and the images with a GSD > 3 cm/pix. We train the network in three consecutive steps:

- **Step-1:** The main idea of the first step is to learn general features, including different background features. We train the network on the whole training set for 300k iterations. We do not use data augmentation as it causes early divergence.
- **Step-2:** The idea of the second step is to focus on learning the target features in higher resolution images. Therefore, we continue to train the model from Step-1 on the Pos image patches with GSD ≤ 3 cm/pix for 35k iterations. As the network has already seen these patches, we apply intensive data augmentation including scaling, rotation, translation, and illumination changes.
- **Step-3:** In the last step, the network should learn all the target features. Therefore, we train the model from Step-2 on all Pos image patches of the training set for 66k iterations with an intensive data augmentation, as in Step-2.



(a) (b)

Figure 7. Distribution of the bounding boxes in height and width space and the anchor boxes as cluster centers depicted by red dots (a), illustration of the anchor boxes (b).

3.2 Experimental setup

We use the source code published by the authors in (Redmon and Farhadi, 2018) as the base code for our implementation and experiments. We select the number of training iterations by observing that the loss curve reaches a plateau and the validation results stop improving. The validation set, consisting of 25 images, was used to calculate the validation results. After setting up the overall training procedure, we trained the network with all images in the training set and applied the final model to the test set. For all training steps (Step-1 to Step-3), we set the batch size to 50. The learning rate for the whole training procedure was set to 5e−7 and with the same scheduling mechanism as (Redmon and Farhadi, 2018). We used the Adam optimizer with a decay of 5e−4 and we employed the pre-trained model on the MS COCO dataset (Lin et al., 2014), which turned out to be the best fit for our problem considering the overall object sizes. The training process was realized on a GeForce RTX 2080 Ti GPU for training and inference, respectively.

3.2.1 Loss calculation In order to calculate the loss, we consider the objectness and bounding box errors. For the bounding box prediction loss, we calculate the complete IoU (CIoU) (Zheng et al., 2022) between the predicted and target bounding boxes, similar to the original paper (Redmon and Farhadi, 2018):

$$\mathcal{L}_{box} = 1 - CIoU. \qquad (1)$$

We estimate the objectness scores for the predicted bounding boxes using logistic regression, as in the original paper. Then we compute the objectness loss ($\mathcal{L}_{obj}$) by computing the cross-entropy between the predicted and target scores. Finally, we compute the total loss by combining the two loss values:

$$\mathcal{L} = \mathcal{L}_{obj} + 0.05 \; . \; \mathcal{L}_{box}. \qquad (2)$$

3.2.2 Anchor boxes To select appropriate anchor boxes, we perform a statistical analysis on the bounding boxes of the training set. We evaluate various properties of the bounding box distribution in height and width space, including the outliers, and apply k-means clustering to the distribution for nine clusters, resulting in the following anchor boxes: (12 × 12), (18 × 18), (23×20), (26×28), (37×30), (44×42), (59×44), (75×62), (106×98). Figure 7 shows the distribution of the bounding boxes, the anchor boxes as cluster centers (red dots), and the visualization of the anchor boxes with respect to the image patch size. Experimental results show significant improvement with defined anchor boxes over the original setup.

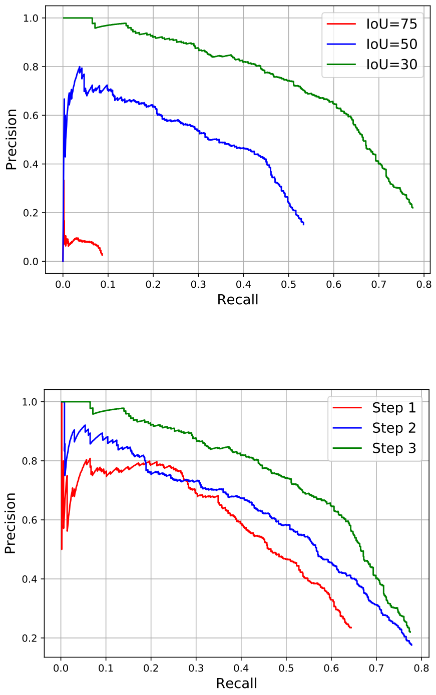


Figure 8. Precision-Recall curves for different IoU thresholds (top), and different training steps with IoU=30 (bottom).

3.3 Evaluation metric

For quantitative evaluation, we use Average Precision (AP), the weighted average of Precision at different prediction confidence thresholds. The weight is the increase in Recall from the previous threshold. AP summarizes the Precision-Recall curve. More correct predictions result in a better Precision-Recall curve and higher AP. The best possible score is 1 and the worst possible score is 0. Precision and Recall are:

$$Precision = \frac{TP}{TP + FP}, \qquad (3)$$

$$Recall = \frac{TP}{TP + FN}, \qquad (4)$$

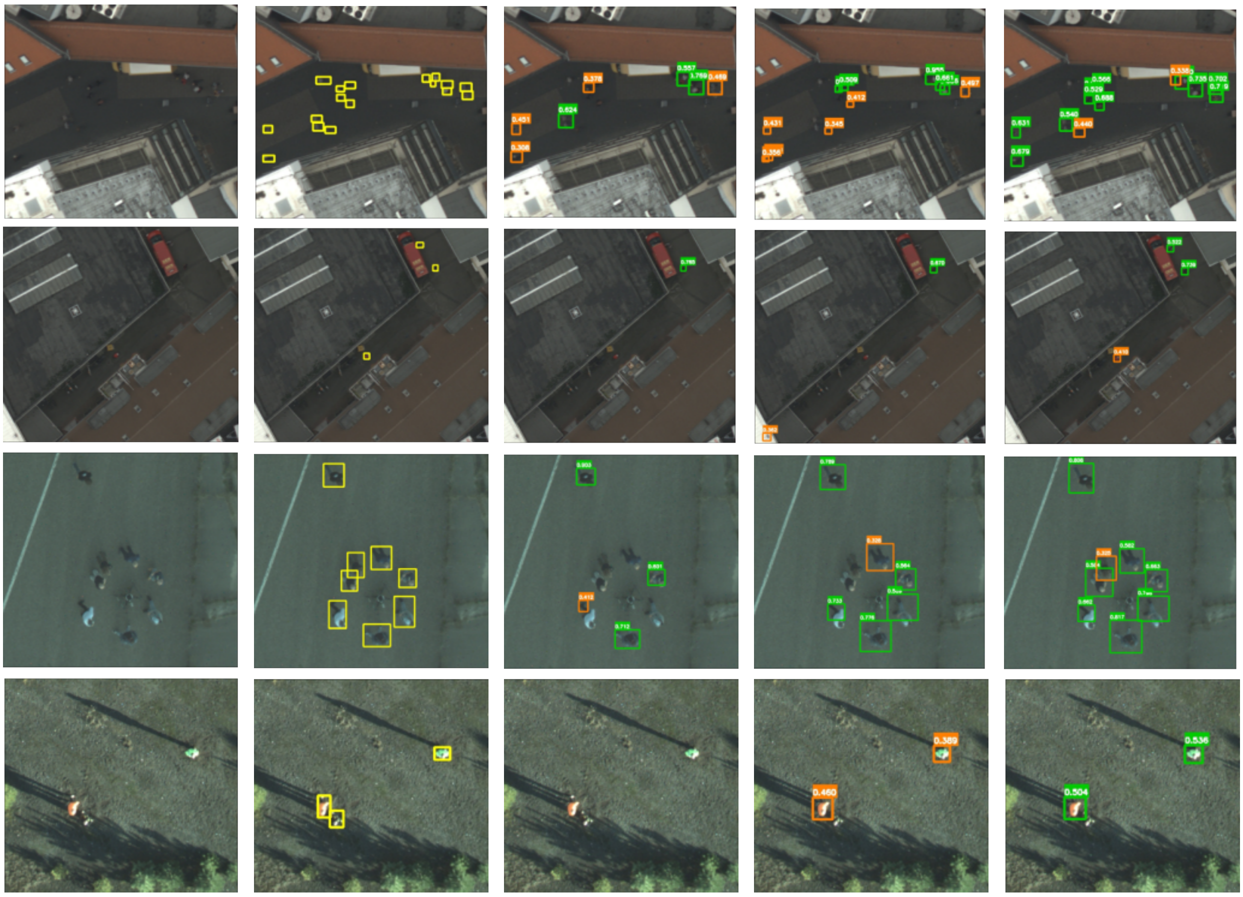
where True Positive (TP) and False Positive (FP) indicate whether a predicted bounding box is correct. False Negative (FN) indicates the number of actual bounding boxes that were missed. To assign the predicted bounding boxes to the actual ones, we use the Intersection of Union (IoU) metric, which quantifies the closeness of the two bounding boxes with a value between 0 and 1. For complete overlap of the bounding boxes, IoU is equal to 1. To consider a match as true, we set a threshold for IoU.

4. RESULTS AND DISCUSSION

We evaluate our trained model on the test set. We assume that our test set can clearly show the performance of our model for operational missions because it covers various challenges that are present in real-world applications, as its images were taken during search and rescue exercises and missions. We compare the results quantitatively in Table 1 and qualitatively in Figure 9 to demonstrate the impact of different training steps on model performance. In Table 1, AP_{30}, AP_{50}, AP_{75}, specify the value of AP by selecting the IoU threshold as 0.3, 0.5, and 0.75,

Training	LR	# Iter	Pos/Neg	GSD	Aug	AP_{30}	AP_{50}	AP_{75}	$AP_{\leq 3}$	$AP_{> 3}$
Step 1	5e−7	300k	Pos + Neg	All	-	0.41	0.23	0.02	0.34	0.67
Step 2	5e−7	35k	Pos	$\leq 3\ cm$	✓	0.49	0.25	0.01	**0.60**	0.62
Step 3	5e−7	71k	Pos	All	✓	**0.60**	0.29	0.01	0.54	**0.76**

Table 1. The training steps with their specifications and the APs of their results. $AP_{\leq 3}$ and $AP_{> 3}$ refer to the results for the test images with the GSDs ≤3 and >3 cm/pix, respectively.



(a) Aerial/drone image (b) Ground truth (c) Step-1 (d) Step-2 (e) Step-3

Figure 9. Detection results for the three training steps (see Table 1). The GSDs of the images from top to bottom are 4.2, 4.2, 1.2, and 2.9 cm/pix. Bounding box colors are: 🟨 ground truth, predictions with 🟧 confidence < 0.5 and 🟩 confidence ≥ 0.5.

respectively. The best results are obtained by setting the IoU threshold to 0.3 (IoU=30). Due to the very different shapes and sizes of the bounding boxes (see the distribution in Figure 7) the model has difficulties fitting the prediction exactly to the ground truth bounding boxes. For most of the intended applications of this model such as in search and rescue missions, the main goal is to identify the presence of people, and delineating their exact contour is not important. Therefore, setting the IoU=30 is sufficient. Figure 8 (a) also shows a better Precision-Recall curve for IoU=30.

In this table, we can also see that training in different steps contributes significantly to the performance, and the final model achieves an AP of 0.60, which is about 46% improved compared to the AP of the first step. Figure 8 (b) shows that different steps reduce the number of missing people and increase the correctness of the predicted bounding boxes. Table 1 also evaluates the training influence on the low and high GSD images. $AP_{\leq 3}$ and $AP_{> 3}$ refer to the results for the test images with the GSDs ≤3 and >3 cm/pix, respectively. $AP_{\leq 3}$ and $AP_{> 3}$ are calculated by averaging the image-wise APs for the respective GSDs. Consequently, their calculation differs from the overall APs, which consider all detections across the entire dataset simultaneously. Therefore, their weighted average does not reflect the other APs. The results verify our assumption about the data imbalance and the design of our training strategy. Training on the whole dataset in Step-1 significantly biases the model towards the images with higher GSD. Further training the model only on the images with GSD≤ 3 cm/pix in Step-2 brings the model into balance by significantly improving its performance on the images with lower GSD. Finally, by training on the entire dataset in Step-3, we recover the reduced detection performance for the high GSDs in Step-2. In addition, training only on the Pos image patches in Step-2 and -3 allows the network to focus on learning the target features faster without increasing the FP detections, since the model has already learned the background diversity in Step-1.

Figure 9 visually demonstrates the model performance for different training steps. We include examples from different scenarios and GSDs to give a more comprehensive impression of the performance. In all examples, moving from Step-1 to Step-

3 leads to the detection of missing people and increases the network's confidence in the detected bounding boxes. Furthermore, this helps getting rid of some erroneously detected bounding boxes. Figure 10 shows further example results of the final model. Although the model misses a few people in the first example, the overall detection performance is reasonable considering the complexity of the scene, where many objects which could be confused with people are present among the ruins. In the second and third examples, where the images have very small GSDs, all people are correctly detected. In the second example, we can see some multiple detections that can be removed by post-processing. In the last example, with a higher GSD and a more complex scene, we can see some false detections. It is common in high GSD images that many objects look very similar to people. Overall, we can rate the detection performance in this example as good, since most of the people are correctly detected.

In many applications, such as search and rescue missions, the inference time and computational complexity play a crucial role due to the limited time and computational resources available. In this regard, we find YOLOv3 to be relatively fast and efficient. Our model can process 1 Megapixel in 0.12 seconds, for a GSD of 3 cm/pix, which means 1 km^2 in 2.3 minutes.

4.1 Examples of failed attempts

For simplicity, similar to many existing crowd datasets, we annotate people with single points on their bodies. We then automatically create squared bounding boxes centered on the points and size them according to the image GSDs. A visual inspection shows that for the images with higher GSDs, where the viewing angles are closer to the nadir and the people appear almost as circles, the bounding boxes can do a good job of estimating the people's coverage. However, these bounding boxes almost fail to circumscribe the people in the images with better GSDs. People have different poses, and due to the oblique viewing angle of the images, several of their body parts such as hands and legs are visible. The experimental results also show unsatisfactory predictions of the bounding boxes and a low AP.

In order to compensate for the data imbalance, we apply different weighting strategies to the loss calculation as a function of the number of images and the number of annotations for different GSDs. All these strategies result in underfitting and performance degradation. We also try YOLOv3-Tiny (Gong et al., 2019); however, it does not perform well. This could be due to its much shallower backbone, Darknet-19, and the fact that it only performs detection on two courser scales, which results in missing smaller bounding boxes that cover a large portion of our dataset.

5. CONCLUSION AND FUTURE WORKS

Our paper presents a novel dataset for person detection in aerial and drone images, which incorporates diverse scenes and scenarios, including disaster-affected areas and search-and-rescue exercises and missions. The dataset's broad coverage ensures that the algorithms trained and tested on it are suitable for real-world operational missions. We propose a multi-stage training strategy that significantly enhances the detection performance of YOLOv3. Our analysis and suggestions, based on YOLOv3, can be applied to newer variants of the YOLO family. Aerial and drone imagery challenges often require method adaptation, regardless of the specific variant used. Our future work includes expanding our dataset with more diverse images from real-world scenarios, as well as increasing the sample size. We also aim to explore other YOLO variants to achieve improved detection accuracy with lower computational demands.

ACKNOWLEDGMENT

This work was supported by internal DLR funding as part of the Humanitarian Technologies Initiative. The authors would like to thank DLR's Institute of Optical Sensor Systems for providing image data for the presented person detection dataset.

REFERENCES

Akshatha, K., Karunakar, A., B., S., K., P., Dhareshwar, C., Johnson, D., 2023. Manipal-UAV person detection dataset: A step towards benchmarking dataset and algorithms for small object detection. *ISPRS Journal of Photogrammetry and Remote Sensing*, 195, 77–89.

Bahmanyar, R., Vig, E., Reinartz, P., 2019. Mrcnet: Crowd counting and density map estimation in aerial and ground imagery. *BMVC's Workshop on Object Detection and Recognition for Security Screening (BMVC-ODRSS)*.

Brauchle, J., Bayer, S., Hein, D., Berger, R., Pless, S., 2019. MACS-Mar: A Real-time Remote Sensing System for Maritime Security Applications. *CEAS Space Journal*, 11(1), 35–44.

Girshick, R., 2015. Fast r-cnn. *2015 IEEE International Conference on Computer Vision (ICCV)*, 1440–1448.

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2016. Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1), 142-158.

Gong, H., Li, H., Xu, K., Zhang, Y., 2019. Object Detection Based on Improved YOLOv3-tiny. *2019 Chinese Automation Congress (CAC)*, 3240-3245.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep Residual Learning for Image Recognition. *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

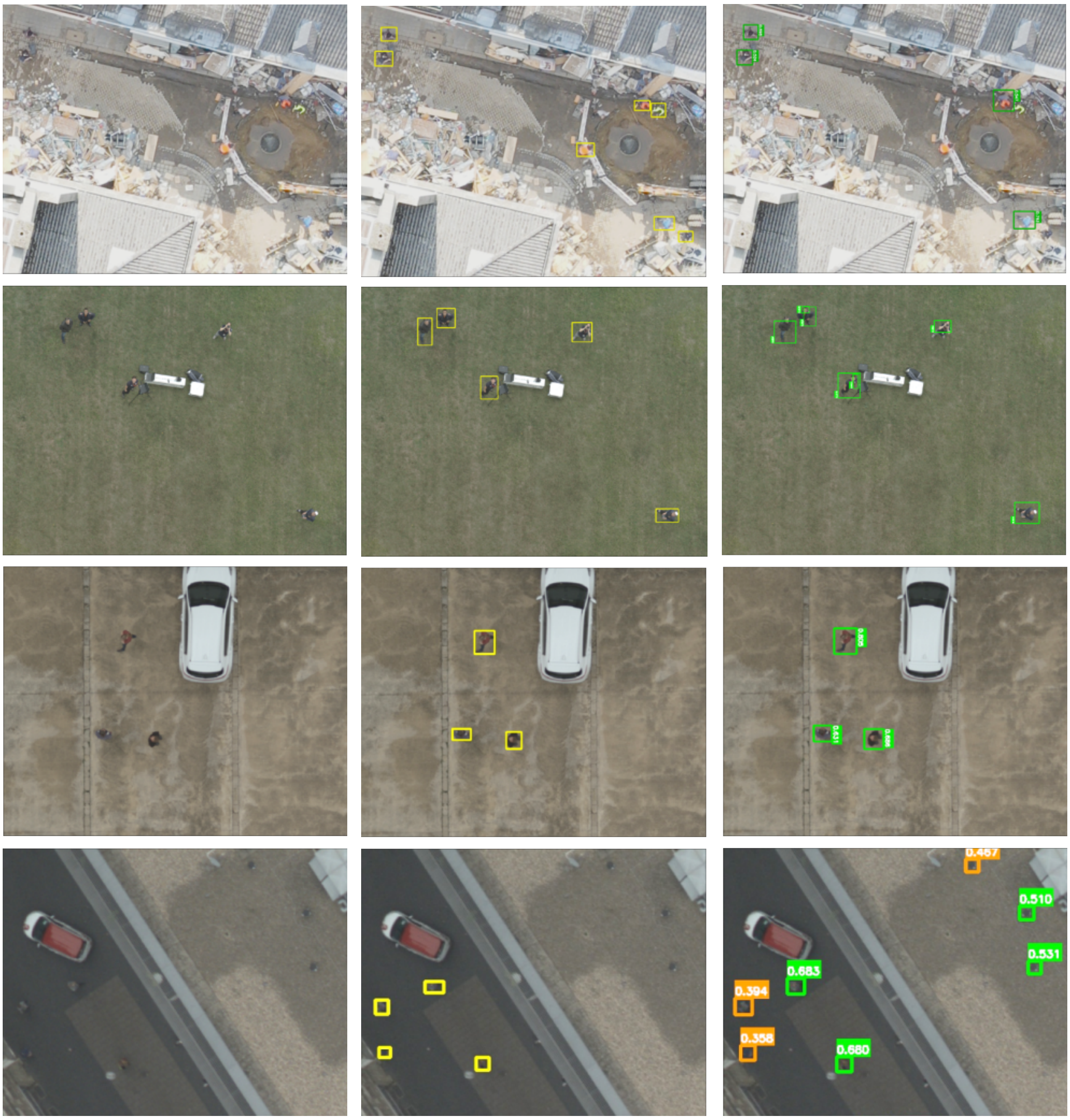
Kurz, F., Rosenbaum, D., Meynberg, O., Mattyus, G., Reinartz, P., 2014. Performance of a real-time sensor and processing system on a helicopter. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 189-193.

Li, T., Liu, J., Zhang, W., Ni, Y., Wang, W., Li, Z., 2021. UAV-Human: A Large Benchmark for Human Behavior Understanding With Unmanned Aerial Vehicles. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 16266–16275.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C. L., 2014. Microsoft coco: Common objects in context. *European Conference on Computer Vision (ECCV)*, 740–755.

Mohd Daud, S. M. S., Mohd Yusof, M. Y. P., Heo, C. C., Khoo, L. S., Chainchel Singh, M. K., Mahmood, M. S., Nawawi, H., 2022. Applications of drone in disaster management: A scoping review. *Science & Justice*, 62(1), 30–42.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788.



(a) Aerial/drone image

(b) Ground truth

(c) Step 3

Figure 10. Examples of final results. The GSDs of the images from top to bottom are 1, 0.6, 0.6, and 4.2 cm/pix. Bounding box colors are: 🟨 ground truth, predictions with 🟧 confidence < 0.5 and 🟩 confidence ≥ 0.5.

Redmon, J., Farhadi, A., 2017. Yolo9000: Better, faster, stronger. *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6517–6525.

Redmon, J., Farhadi, A., 2018. YOLOv3: An Incremental Improvement. *arXiv*.

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Sekachev, B., et al., 2020. opencv/cvat: v1.1.0.

Varga, L. A., Kiefer, B., Messmer, M., Zell, A., 2022. Seadronessee: A maritime benchmark for detecting humans in open water. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2260–2270.

Wang, C.-Y., Bochkovskiy, A., Liao, H.-Y. M., 2022. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *preprint arXiv:2207.02696*.

Zheng, Z., Wang, P., Ren, D., Liu, W., Ye, R., Hu, Q., Zuo, W., 2022. Enhancing Geometric Factors in Model Learning and Inference for Object Detection and Instance Segmentation. *IEEE Transactions on Cybernetics*, 52(8), 8574-8586.

Zhou, X., Wang, D., Krähenbühl, P., 2019. Objects as points. *arXiv preprint arXiv:1904.07850*.

Zhu, P., Wen, L., Du, D., Bian, X., Fan, H., Hu, Q., Ling, H., 2021. Detection and Tracking Meet Drones Challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7380–7399.