ON THE ACCURACY OF YOLOV8-CNN REGARDING DETECTION OF HUMANS IN NADIR AERIAL IMAGES FOR SEARCH AND RESCUE APPLICATIONS

J. Berndt[1]*, H. Meißner[1], Th. Kraft[1]

[1] Institute of Optical Sensor Systems, German Aerospace Center, 12489 Berlin, Germany -
(Julian.Berndt, Henry.Meissner, Thomas.Kraft)@dlr.de

KEY WORDS: Deep Learning, YoloV8, Human Detection, CNN, Convolutional Neural Network, UAV, Aerial Images.

ABSTRACT:

The use of deep learning techniques especially in conjunction with convolutional neural networks (CNN) has attracted major attention of the remote sensing community. Main use cases are object detection, image classification and image segmentation.
The paper will focus on object detection, specifically on detection of humans. In search and rescue applications it is common to map larger areas with downward facing cameras. However, there are many training data sets for CNNs showing oblique images which strongly differ from nadir aerial images used for real-time maps.
To circumnavigate this issue, an unique data set was created. It solely contains nadir images at different ground sample distances (GSD) varying from one to five centimetres. Diversity of the training data is ensured through various flights using an unmanned aerial vehicle (UAV) at different locations. GSD dependency is valuable prior knowledge as it enhances the difficulty associated with human detection in aerial images. An image, depicting a human at one centimetre GSD contains much more information than the same human depicted in an image of three centimetres. That is one reason why networks trained on a variety of ground sample distances possibly struggle to detect humans reliably on a certain GSD.
The unique data set consists of four subsets (divided by GSD). Each subset contains 1000 manually annotated humans, augmented by rotation and colour shift resulting in 12000 training samples used to train the new released YoloV8 CNN. The entire training and test process is unified to ensure comparable input conditions.

1. INTRODUCTION

In search and rescue (SAR) operations, the fast and accurate detection of human beings can decide whether lives can be safed or not. The validity of the detection results is crucially important to ensure the success of rescue missions. The capabilty to identify and locate individuals in diverse or challenging environments can influence the effectiveness and efficiency of search and rescue teams in noteworthy ways. There are traditional ways of human detection e.g. manual visual search in images or the use of thermal imaging but these come with limitations which can impede the effectiveness in such complex scenarios.

However, the recent years have brought the rise of neural networks and deep learning which have revolutionized the field of computer vision and broke new ground for enhancing object detection in SAR-scenarios (Rodin et al., 2018, Bejiga et al., 2017). Neural networks have emerged as a powerful tool for processing and analyzing large volumes of visual data. That has enabled the development of state of the art (SOTA) systems for SAR applications (Martinez-Alpiste et al., 2021).

The primary objective of this paper is to explore the utilization of YOLOv8 (you only look once - version 8) networks in human detection for search and rescue scenarios. By leveraging the capabilities of neural networks, we aim to address the possible limitations that come with ground sampling distance (GSD)-dependency and investigate this with several tests. In addition, we are investigating whether the neural networks of one GSD range function better when they are operated in a higher GSD range. This effect would be conceivable if the purely mathematical calculation of the GSD deviates from the actual one, e.g. due to smear (Meißner, 2020).

* Corresponding author

2. PREREQUISITES AND RELATED WORK

This chapter explains the related work of this paper and the prerequisites that need to be taken into account when dealing with the topic.

2.1 YOLOv8

YOLOv8 (you only look once - version 8) (Jocher et al., 2023, Terven and Cordova-Esparza, 2023) is an object detection algorithm which is designed to accurately and efficently detect objects in images or video streams. This algorithm is published by ultralytics (Jocher et al., 2023) and can be utilized in various applications. Out-of-the-box support is given for object detection, segmentation and classification (Jocher et al., 2023). It is easy to apply to use-cases since it is accessible through a Python package. There are different pretrained models (trained on the COCO-128 dataset) which can be used for training, starting from checkpoints. This is recommended by the creators and was adapted for the following experiment. One of the key features of YOLOv8 is the anchor-free approach. That means it does not rely on predefined anchor boxes to generate object proposals. Instead, YOLOv8 directly predicts the bounding boxes and class probabilities for each object in the input image.

2.2 Object Detection

Object Detection is the foundation of this paper. It is a technique used to locate instances of objects in images or frames of videos (Amit et al., 2020). The detection is usually achieved by using deep learning or machine learning. The main goal of object detection in this paper is to mimic the ability of a human to perceive and locate an object in an image. Object detection is an

enormously important technology for the future e.g. in the driving assistant sector (Mao et al., 2022) or in our focussed case search and rescue scenarios (Martinez-Alpiste et al., 2021).

There are mainly two variants for object detection models.

Two-Stage Detectors

With this technology the first stage identifies parts of the image which might contain the object (regions of interest or RoI). The second stage then detects the objects in said RoI´s. These detectors are usually slower but more precise when it comes to results in comparison to single-stage detectors (Soviany and Ionescu, 2018).

Single-Stage Detectors

In a single-stage object detector, the detection is performed in a single pass through the network. It directly predicts the bounding boxes and class labels for objects in an image. They divide the input image into a grid of cells and predict bounding boxes and class labels directly for each cell. These detectors are usually faster but less precise when it comes to results (Soviany and Ionescu, 2018).

Considering the above stated, a single-stage detector was chosen because the final goal should be to use such networks in real-time mapping applications where the speed of a single-stage detector outperforms the higher accuracy of a two-stage detector with view on the use-case (SAR).

YOLO is designed to detect larger or prominent objects in an image and some articles focus on modification of the YOLO object detector to improve its performance in detecting smaller objects (Benjumea et al., 2021). When dealing with detection of humans in aerial images this fact led to the investigation presented in this paper that GSD is an important factor in terms of precision and overall accuracy. As mentioned before a pre-trained network was transformed via transfer learning to fit the use-case of human detection.

2.3 Nadir/Oblique Aerial Images

This paper focuses on the use of nadir aerial images. An aerial image is a photography of any kind and sprectrum (e.g. visual, thermal infrared, near infrared, multispectral, etc.) taken from a carrier system moving above the earth´s surface. The camera capturing the scene can be aligned in different angles depending on the purpose of the images. A nadir aerial image is taken orthogonal to the scene. An oblique aerial view would deviate from this (Paine and Kiser, 2012). The difference between the two is shown in Figure 1.

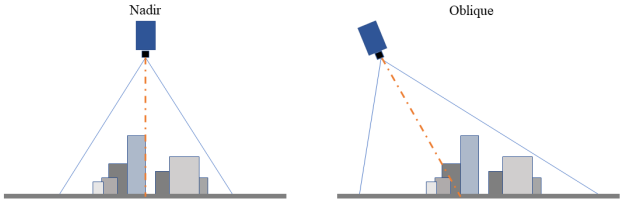

Figure 1. Nadir and oblique view.

2.4 Search and Rescue

A search and rescue (SAR) scenario refers to a situation where efforts are undertaken to locate, assist, and rescue individuals who are in distress, danger, or missing (Frost and Stone, 2001). SAR operations are typically carried out in various challenging environments.

Search and rescue scenarios can involve a wide range of situations, such as:

1. Missing Person
2. Natural Disasters
3. Maritime and aviation incidents
4. Urban disasters
5. Medical emergencies

The goal in all of these situations is to help the individuals as fast and effectively as possible. SAR missions can be assisted by UAV´s, computer vision and the use of neural networks to improve the image processing (Waharte and Trigoni, 2010).

2.5 GSD

The ground sampling distace (GSD) refers to the dimensions of a single pixel in an image, in relation to the surface it covers on the ground. When calculating the GSD of an image one has to consider camera´s sensor properties and focal length, as well as the distance between the sensor and the ground when the photo is taken. This is usually represented by the altitude of the drone or aircraft the camera is attached to (Draeyer and Strecha, 2014).

This paper investigates the influence GSD-specific-training has on the applicability of neural networks in different GSD sections.

3. EXPERIMENTS

3.1 Requirements

There are some specifications on the hardware and software side that should be mentioned before talking about the experiment itself.

3.1.1 Hardware The hardware specifications on the machine used for training the neural networks were the following (Table 1):

Part	Specs	
CPU	AMD Ryzen 9 3900X	@3.8 GHz
GPU	nVidia RTX 3090	@24GB VRAM
RAM	Crucial Ballistix 64GB	@2666 MHz

Table 1. Hardware Specifications.

3.1.2 Software The software specifications and versions of the used software can be found underneath in Table 2.

3.1.3 Camera Parameters To make the data from different GSD´s comparable it is important that the same sensor was used to capture all of it. The camera parameters are listed below in Table 3.

Part	Version
OS	Windows 10 Pro
YOLOv8	8.0.45
PyTorch	2.0.1 + cu117
CUDA	11.3
nVidia-Driver	531.79

Table 2. Software Specifications.

Part	Specs
Sensor width	35.9936mm
Sensor height	23.9168mm
Image width	4864px
Image height	3232px
Principal distance	50.874261mm
Manufacturer	SVS Vistek

Table 3. Camera Parameters.

3.1.4 Training Data In order to carry out this experiment, annotated data was needed. This consisted of aerial images from the above mentioned camera in different GSD ranges. The images used in training were all caputered by the same sensor of the German Aerospace Center over several years. The GSDs available ranged from 1cm to 5cm resolution. According to this, a classification of the available images was made into four classes. Class 1 (1-2cm), Class 2 (2-3cm), Class 3 (3-4cm) and Class 4 (4-5cm).

The images were then annotated and exported using wekantaro's software ("LabelMe") (Wada, n.d.). 1000 persons were annotated per class. Considering that this is a somewhat small amount of training data for a neural network, it was decided to augment the annotated data. Since the labels created during this process cannot be used directly in YOLOv8, a custom script was used to adapt the labels.

The augmentation was done with a specially designed software, which is expected to grow significantly in functionality in the future. With the already available augmentation, a twelvefold increase in data was achieved.

For each annotation, a box of 320 pixels was drawn in all directions around the center pixel of a person. The result is an image with dimensions of 640*640 pixels with at least one annotation in the centre of the image. If more than one person is annotated in the 640*640 pixels, all annotations will of course be included. After converting the individual annotations of the large image into individual sections, these were rotated three times by 90° each and two colour shifts were made per rotation (including the initial tile).

In order to be able to make moderate shifts in the colour, the average of the pixel values is formed. This is then placed into a five-part, equidistant grid from 0 to 65536. Depending on the position in the grid, either one shift up and down or two steps up or down could be made.

Through this process, the number of training data was increased from 1000 annotated images to 12000. Of these 12000 images, 11800 per class were then used to train the neural networks and 200 were retained for validation. In the course of the development, it became apparent that 200 annotated images were not sufficient for the validation. As a result, 100 individuals per class were annotated again and the following files were augmented. Thus, the number of 200 validation images grew to 1400. Consequently, a ratio of about 90:10 between the training and validation set was achieved.

The whole process can be visualised as follows in Figure 2:

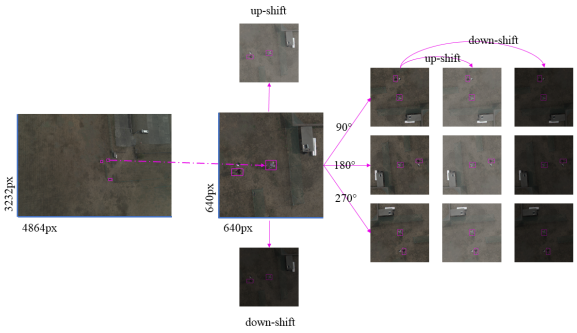

Figure 2. Augmentation Process.

3.2 Training

The training was carried out as shown in the following matrix (Table 4).

NN Size / GSD	1-2cm	2-3cm	3-4cm	4-5cm
S	...	...	...	...
M	...	...	...	...
L	...	...	...	...
XL	...	...	...	...

Table 4. Neural Network Size vs. GSD.

It was decided to do this in order to be able to compare all neural networks at the end. In this way, it becomes obvious, for example, which network with which data generally performed best or which network in a class (GSD) particularly stands out.

The training consisted of the following parameters per cell of the matrix (Table 5):

Parameter	Value
Image Size	640*640px
Epochs	150
Batch Size	8
Worker Count	8

Table 5. Training Parameters.

After the training, the 16 created networks were run on the validation data of each class to obtain comparison parameters. These can then be used to evaluate the experiment.

4. RESULTS

In this chapter, the results of the individual networks are shown. The results were generated with the data already mentioned for validation. For the evaluation of the results, various parameters can be output by the validation methods of YOLOv8. The chosen parameters are examined in more detail below:

4.1 Parameters

In this following subsection the parameters for the evaluation of the networks are presented.

4.1.1 Detection confidence The confidence score is a measure of how certain or confident the model is in its predictions. It provides an estimate of the reliability or trustworthiness of the network's output for a given input.

The confidence score is typically associated with classification tasks, where the neural network is trained to assign input data into different categories or classes. When making predictions, the network computes a probability distribution over all possible classes and assigns a confidence score to each class. The confidence score reflects the network's belief in the correctness of its prediction for that particular class (Wenkel et al., 2021).

In the case of this work there is only the class "human" which is tested with the validation data.

A high confidence score indicates that the network is very confident in its prediction for a given class, while a low confidence score suggests more uncertainty or ambiguity. In some cases, the confidence score can be used to set a threshold for decision-making. For example, if the confidence score is below a certain threshold, the network might indicate that it is unsure about the prediction and leave it out of the results or it requires human intervention.

4.1.2 True and false positives True positives and false positives are terms used in the context of binary classification tasks, where the neural network is trained to classify inputs into one of two classes: positive and negative.

True Positives (TP): True positives refer to the cases where the neural network correctly predicts a positive class when the actual class is indeed positive. In other words, the network correctly identifies the presence of the target condition or event (Hoiem et al., 2012).

False Positives (FP): False positives occur when the neural network incorrectly predicts a positive class when the actual class is negative. In this case, the network falsely identifies the target condition or event when it is not present (Hoiem et al., 2012).

Two further parameters can be derived from this.

Precision, also known as positive predictive value, is defined as the ratio of true positives to the sum of true positives and false positives. It measures the proportion of correctly identified positive instances out of all instances predicted as positive. Precision indicates how reliable the positive predictions are.

Recall, also called sensitivity or true positive rate, is the ratio of true positives to the sum of true positives and false negatives. It measures the ability of the classifier to correctly identify positive instances out of all actual positive instances. Recall reflects the classifier's ability to detect the target condition or event.

4.1.3 mAP 0.5 The term "mAP 0.5" refers to the mean Average Precision at an Intersection over Union (IoU) threshold of 0.5. It is commonly used as an evaluation metric for object detection tasks performed by neural networks (Padilla et al., 2020).

In object detection, the goal is to identify and localize objects within an image. The IoU is a measure of the overlap between the predicted bounding box and the ground truth bounding box for an object. An IoU threshold of 0.5 means that if the IoU between the predicted box and the ground truth box is greater than or equal to 0.5, the detection is considered correct.

The mean Average Precision (mAP) is a summary metric that measures the overall performance of an object detection model across multiple IoU thresholds. It combines precision and recall values at different IoU thresholds to calculate an average precision value for each class. The mAP 0.5 specifically focuses on the IoU threshold of 0.5.

A high mAP 0.5 value indicates that the neural network performs well in object detection tasks, particularly in terms of accurately localizing objects with a reasonable overlap with the ground truth bounding boxes. It implies that the model achieves a good balance between precision and recall at an IoU threshold of 0.5.

However, it's important to note that mAP 0.5 alone may not provide a comprehensive evaluation of a neural network's performance. It is typically used in conjunction with other IoU thresholds (such as 0.75 or 0.9) to assess the model's robustness across different levels of bounding box overlap. Additionally, considering other metrics like precision, recall, and F1 score can provide a more comprehensive understanding of the network's performance.

4.2 Experiment outcome

The upcoming diagrams and analyses show the outcome of the experiment performed on the trained networks and validation datasets.

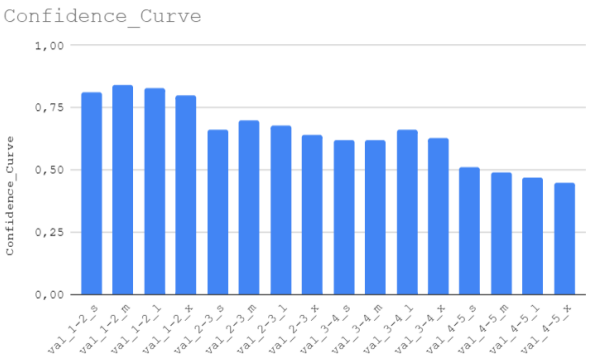

Figure 3. Confidence Distribution.

This diagram (Figure 3) shows the distribution of the confidence across the different nets and GSD ranges. It is clearly visible that the confidence values are highest in the 1-2cm class and decrease with increasing GSD. This pattern is only interrupted by the l- and x-net in the GSD region of 3-4cm.

For the 1-2cm class, it can generally be said that all nets achieved a confidence of over 0.75 on the corresponding validation data. The highest value of the class was achieved by the m-net, closely followed by the l-net.

The 2-3cm class achieved a lower confidence overall with an average of 0.67. In this class, the m-net also achieved the highest confidence with a value of 0.7. The other nets slightly lagged behind.

The neural networks of the next class (3-4cm) achieved an average confidence of 0.6325. Unlike in the two previous classes, it was not the m-network that achieved the highest value, but the l-network. The other three trained neural networks (s, m, x) were fairly evenly matched with values of 0.62 and 0.63 respectively.

The neural networks of class 4-5cm achieved by far the worst results on the associated validation data. The confidence average here is 0.48. The networks steadily decrease in their confidence values from s (0.51) to x (0.45).

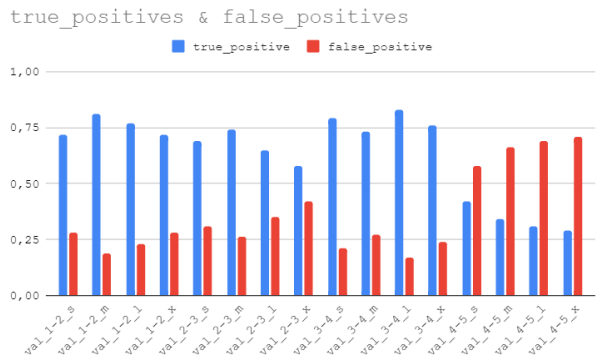

Figure 4. True and false positives.

Figure 4 shows the distribution of false positives and true positives of all tested neural networks. In general, it can be stated that the true-positives are best in the GSD ranges 1-2cm and 3-4cm. The nets of the GSD range 2-3cm performed mediocre in terms of all results. The nets of the last GSD range again performed worst.

The neural nets of the 1-2cm class achieved a true-positive rate of 0.755 on average, which was the second highest rate in the tests. In this class, the m-net had the highest true-positive rate at 0.81 and the s- and x-nets the worst at 0.72.

The next GSD class (2-3cm) had an average true-positive rate of 0.665. This means that the neural networks performed slightly worse than the previous class in terms of true-positives by an average of 5.5 percent. The highest result in this class was achieved by the m-net with 0.74 and the worst by the x-net with a value of 0.58.

The class 3-4cm achieved on average the highest results for true-positives with a value of 0.7775. The top performer in this class is the l-net with a true-positives rate of 0.83. The worst result in this class was achieved by the m-net with a value of 0.73.

The neural networks of the last class (4-5cm) had by far the worst true recognition rates. The average of this class is 0.34. The best true-positive rate is shown by the s-net, with a value of 0.42. The worst net is the x-net with a rate of 0.29.

NN Size / GSD	1-2cm	2-3cm	3-4cm	4-5cm
S	0.84	0.521	0.53	0.434
M	0.85	0.562	0.5	0.427
L	0.847	0.544	0.514	0.408
XL	0.827	0.515	0.495	0.404

Table 6. mAP 0.5 for Neural Network Size vs. GSD.

The matrix above (Table 6) shows the distribution of the mAP 0.5 values across all tested nets. It is clearly visible that the neural nets of the GSD range 1-2cm achieved the highest results. These are followed by the nets of the range between 2 and 3 centimetres. This is followed by the nets of the range between 3 and 4 centimetres and the worst values of this parameter are shown by the nets which were trained on the largest GSD range.

As with the other parameters above, the neural networks in the GSD range 1-2 cm perform best and those in the GSD range 4-5 cm perform worst.

The overall picture of all the individual parameters can be seen very well if you look at the summary below (Figure 5). In this graph, all parameters for each net in all GSD ranges are plotted and it is clear that the nets in the first GSD range produce the best results. The two middle GSD ranges produce good to medium results and the nets of the largest GSD range produce by far the worst results.

This suggests that there is a dependency related to the GSD.

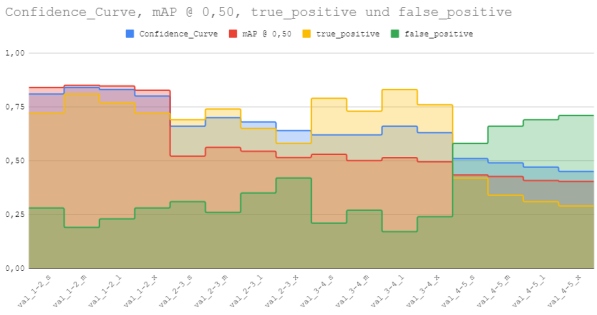

Figure 5. Summarized results.

4.3 Additional Results - Crosscheck

In addition to examining the nets in the assigned GSD range, it was also interesting to see how nets of a class, e.g. trained on 1-2cm data, perform in the validation of a higher class. A so called crosscheck was done. This idea came up because the GSD, which is used for the division of the classes, is calculated. It may well be less accurate than the calculation suggests due to various factors. Factors that can lead to a deterioration of the actual GSD are:

1. Smearing due to exposure time during overflight
2. Focus of the lens
3. Demosaicing process

Smearing due to a long exposure time can lead to an increase in the actual GSD. This is calculated with the help of the camera and aircraft parameters. These are assumed to be idealised, so that sharp images are assumed. A blur can therefore increase the GSD.

As already mentioned in the case of smearing, the GSD is calculated on the assumption that the camera is in focus. If it is not and the photos are blurred as a result, the GSD will also be worse here.

For this cross-check, it was decided to use the respective s- and m- nets, as they were always among the best. Then they were each tested on a higher GSD class. The results of these additional tests are as follows:

In the three following tables "Conf." stands for confidence score.

NN Size / Parameter	Conf.	True Positives	mAP 0.5
S	0.44	0.24	0.418
M	0.49	0.28	0.496

Table 7. Crosscheck I.

4.3.1 1-2cm nets on 2-3cm data It can be seen (Table 7) that the results are significantly worse compared to the GSD-assigned use. There are a few detection hits now and then, but no results that should be relied upon as the true-positive rate hovers around 0.25. The GSD of the 1-2cm class data were apparently so accurate that they cannot be applied to the other classes.

NN Size / Parameter	Conf.	True Positives	mAP 0.5
S	0.67	0.57	0.660
M	0.70	0.67	0.663

Table 8. Crosscheck II.

4.3.2 2-3cm nets on 3-4cm data The results of this investigation (Table 8) are less bad. Even though both nets did not perform as well as in the correlated data domain, they still produced results that could be worked with. Especially the m-net achieved relatively high confidence values and a true-positive rate of 0.7. The deviation of the GSD training data from the 2-3cm class were apparently sufficient to apply the nets at least partially to data of a higher class.

NN Size / Parameter	Conf.	True Positives	mAP 0.5
S	0.37	0.17	0.468
M	0.31	0.212	0.504

Table 9. Crosscheck III.

4.3.3 3-4cm nets on 4-5cm data The results of this crosscheck (Table 9) are quite plainly poor. Of the three experiments, all parameters performed worst. With confidence values of 0.37 and less, the results may be usable with human follow-up, but coupled with the very low true-positive rates, they become useless. The GSD classification seems to have been correct again, because a transfer to a higher class does not make sense.

5. DISCUSSION

After evaluating the results, it can be deduced that there is a connection between the GSD of the training data and the results of YOLOv8 neural networks.

It was clearly recognisable that the networks from the lower GSD ranges achieved better detection results than those from higher GSD ranges.

Possible explanations for this are:

5.1 Pixels per Object

One reason for the better detection in a smaller GSD range could be the number of pixels per object. This is higher in a smaller GSD range and thus an annotated object consists of more features that can be used in training. This should theoretically lead to annotated objects being better learned as they are distinguished by more unique features. This should also lead to a reduction of false positives, as the experiments have shown. The features of other objects that look similar in nadir images, such as lampposts or bollards, can be better distinguished from those of humans during detection. This can also be transferred to the opposite side. According to the theory stated, fewer pixels per object should lead to a lower confidence and a higher false-positive rate. This could be shown in the experiment.

5.2 More precise annotation with lower GSD´s

Annotation on images with a lower GSD is easier because the object appear bigger, clearer and there is likely less smearing that hampers the annotation process. Errors in the annotation are in doubt throughout the entire training process and can consequently also negatively influence the results. This influence on the results is less pronounced in lower GSD areas, as the annotation can be carried out much more cleanly. An example of this can be seen in the following Figure 6:

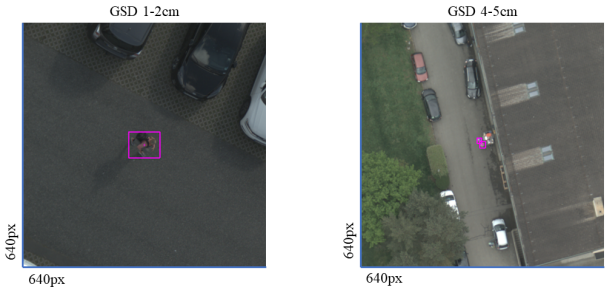

Figure 6. GSD Annotation Comparison.

Even if the scenes the images show are similar, the resolution of a person in the lower GSD range is significantly higher. This then leads to the aforementioned improvement in annotation accuracy.

5.3 More detailed context

Similar to the object itself, a lower GSD also leads to a better learning curve in the network for the context or background of the object. Knowing the background of an object is as important for a neural network as the object itself, because it helps to improve the following points:

1. It helps the network differentiate objects from the background.
2. It enables the network to handle occlusion and clutter.
3. It reduces false positives and false negatives.
4. It enhances the network's ability to generalize to different environments.

This also seems to be true when looking at the results of these experiments.

6. FUTURE RESEARCH

For future or further work, the following things could be done or are being planned:

6.1 Data preparation tool

The tool used for augmenting the data is to be expanded. The team around this work has already collected thoughts on this and these are to be implemented in the near future. The range of functions is currently limited to the rotations and the associated colour shifts. This range of functions should grow. Among other things, it should be possible to:

1. Convert Images from RGB to Mono
2. Include slight displacements of the bounding boxes to lift the training from the centre of the image
3. Apply Gaussian blur
4. Apply artificial edge sharpening

These and more functions should lead to a further diversification of the training data and thus to better results. It should then also be possible to either activate or deactivate all these functions via the programme's GUI.

6.2 Generating training data with neural networks

A second tool is to be created for the post-processing of detection runs. Since a bounding box with image coordinates is created for each person found in an image, this can be read out afterwards. This information should be evaluated by a human in a tool with a viewer. In this way, the image with the respective bounding box could be displayed and the viewer would then have to indicate, for example, via the various directions of the arrow keys, whether an object sought is actually located a bounding box. If so, the box is saved, if not, it is discarded. In this way, the application of neural networks could be used directly for the generation of new training data, which would be included in the next training run.

6.3 Allround network testing

A neural network is to be created from all training data of the different GSD areas. This should then be compared with the results of the confidence matrix and the other parameters. This would make it possible to determine whether a generalisation of the training data in the GSD area would improve or worsen the results.

7. CONCLUSION

In summary, the experiment has shown a dependence of YOLOv8 networks on different GSD ranges. In general, it can be said that neural networks trained with training data from a smaller GSD range produce better results when it comes to detection confidence and the number of true positives.

If this GSD dependency is taken into account, satisfactory detection results can be achieved (see example, Figure 7).

Furthermore, the topic has some future research needs and has not yet been fully explored.

Figure 7. Example: Human detection.

REFERENCES

Amit, Y., Felzenszwalb, P., Girshick, R., 2020. Object detection. *Computer Vision: A Reference Guide*, 1–9.

Bejiga, M. B., Zeggada, A., Nouffidj, A., Melgani, F., 2017. A convolutional neural network approach for assisting avalanche search and rescue operations with UAV imagery. *Remote Sensing*, 9(2), 100.

Benjumea, A., Teeti, I., Cuzzolin, F., Bradley, A., 2021. YOLO-Z: Improving small object detection in YOLOv5 for autonomous vehicles. *CoRR*, abs/2112.11798. https://arxiv.org/abs/2112.11798.

Draeyer, B., Strecha, C., 2014. White paper: How accurate are UAV surveying methods. *Pix4D White Paper*, 4.

Frost, J. R., Stone, L. D., 2001. Review of search theory: Advances and applications to search and rescue decision support.

Hoiem, D., Chodpathumwan, Y., Dai, Q., 2012. Diagnosing error in object detectors. *European conference on computer vision*, Springer, 340–353.

Jocher, G., Chaurasia, A., Qiu, J., 2023. YOLO by Ultralytics.

Mao, J., Shi, S., Wang, X., Li, H., 2022. 3d object detection for autonomous driving: A review and new outlooks. *arXiv preprint arXiv:2206.09474*.

Martinez-Alpiste, I., Golcarenarenji, G., Wang, Q., Alcaraz-Calero, J. M., 2021. Search and rescue operation using UAVs: A case study. *Expert Systems with Applications*, 178, 114937.

Meißner, H., 2020. *Determination and improvement of spatial resolution obtained by optical remote sensing systems*. Humboldt University Berlin (Germany).

Padilla, R., Netto, S. L., da Silva, E. A. B., 2020. A survey on performance metrics for object-detection algorithms. *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, 237–242.

Paine, D. P., Kiser, J. D., 2012. *Aerial photography and image interpretation*. John Wiley & Sons.

Rodin, C. D., de Lima, L. N., de Alcantara Andrade, F. A., Haddad, D. B., Johansen, T. A., Storvold, R., 2018. Object classification in thermal images using convolutional neural networks for search and rescue missions with unmanned aerial systems. *2018 International Joint Conference on Neural Networks (IJCNN)*, 1–8.

Soviany, P., Ionescu, R. T., 2018. Optimizing the trade-off between single-stage and two-stage object detectors using image difficulty prediction. *arXiv preprint arXiv:1803.08707*.

Terven, J., Cordova-Esparza, D., 2023. A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond. *arXiv preprint arXiv:2304.00501*.

Wada, K., n.d. Labelme: Image Polygonal Annotation with Python.

Waharte, S., Trigoni, N., 2010. Supporting search and rescue operations with uavs. *2010 International Conference on Emerging Security Technologies*, 142–147.

Wenkel, S., Alhazmi, K., Liiv, T., Alrshoud, S., Simon, M., 2021. Confidence score: The forgotten dimension of object detection performance evaluation. *Sensors*, 21(13), 4350.