Long-Range Human Detection in Drone Camera Images

Joris Heemskerk
University of Applied Sciences Utrecht
joris.heemskerk@hotmail.com

Tina Mioch
University of Applied Sciences Utrecht
tina.mioch@hu.nl

Henry Maathuis
University of Applied Sciences Utrecht
henry.maathuis@hu.nl

Huib Aldewereld
University of Applied Sciences Utrecht
huib.aldewereld@hu.nl

ABSTRACT

In recent years, drones have increasingly supported First Responders (FRs) in monitoring incidents and providing additional information. However, analysing drone footage is time-intensive and cognitively demanding. In this research, we investigate the use of AI models for the detection of humans in drone footage to aid FRs in tasks such as locating victims. Detecting small-scale objects, particularly humans from high altitudes, poses a challenge for AI systems. We present first steps of introducing and evaluating a series of YOLOv8 Convolutional Neural Networks (CNNs) for human detection from drone images. The models are fine-tuned on a created drone image dataset of the Dutch Fire Services and were able to achieve a 53.1% F1-Score, identifying 439 out of 825 humans in the test dataset. These preliminary findings, validated by an incident commander, highlight the promising utility of these models. Ongoing efforts aim to further refine the models and explore additional technologies.

Keywords

Human Detection, Computer Vision, Drones.

INTRODUCTION

In the last years, the use of drones to aid First Responders (FRs) has become more and more prevalent, including applications such as monitoring fires from an unobstructed perspective (e.g., (Georgiades et al., 2019)) and assessing building damage in difficult-to-access areas (e.g., (Adams et al., 2014)). The camera footage recorded by drones can also be used, for example, to help detect victims, solve logistical problems, and locate missing persons (e.g., (Khan & Neustaedter, 2019)). Currently, FRs usually fly the drones manually and interpret the camera footage without any system help. However, analysing drone images is time-intensive and cognitively demanding, making the interpretation of the camera footage prone to errors. Automating aspects of this analysis, e.g., the detection of humans, can help support FRs in their tasks and decision-making and possibly lead to an overall better understanding of the environment and reduce the time needed to provide aid to victims.

Past research focused on detecting objects like roofs, cars, debris, and vegetation from cameras in UAVs or helicopters (Pi et al., 2020) and also on detecting missing people in for example nature environments (Martinez-Alpiste et al., 2021). We aim to expand upon this by introducing more complex scenarios, like residential and industrial areas, with drones flying at varying heights, angles, and scales. For this, we created a custom dataset based on drone footage from the Dutch Safety Region *Rotterdam-Rijnmond*. This paper explores the use of a state-of-the-art object detection model for real-time long-range human detection in crisis scenarios. It should be noted that this explicitly differs from human identification, in which the goal is to exactly recognize individuals, and thus explicitly excludes surveillance. We only aim to locate humans present in the camera footage, and only during incidents. As such, the central question guiding this research is: *How can humans be located in drone camera footage in order to positively impact the decision-making of first responders?* As we are still in progress of investigating this question, our paper presents first steps and preliminary results.

Identifying humans in drone data can benefit the decision-making process of FRs in different ways. First of all, locating and marking all humans present can help to determine whether humans require rescue or are located in potentially hazardous areas. Secondly, knowing the total number of humans present at a given moment can assist the FRs in addressing logistical challenges, such as determining the number of humans requiring evacuation or shelter. We analyse the preliminary findings together with an incident commander of the *Safety Region Rotterdam-Rijnmond*, to analyse the impact the current leading model can have on their decision-making process.

In this paper, we first explore some of the related background information and describe how we created and prepared our dataset, our model choice, and our evaluation methods. After this, we present our preliminary results, both the quantitative performance as well as qualitative results of an evaluation by an incident commander from the *Safety Region Rotterdam-Rijnmond*. Finally, we discuss our findings and describe future work.

BACKGROUND

In recent years, advancements in Computer Vision and Machine Learning have revolutionized the field of object detection (Zaidi et al., 2022), contributing to innovative solutions in various domains. Specifically, automatically detecting humans in images has been popular (Nguyen et al., 2016). Most of the time, this regards detecting humans from a short-range perspective at a relatively close distance (e.g., from CCTV footage or handheld mobile phone pictures). This is reflected by widely used datasets such as COCO (Lin et al., 2014), which contains many images of, amongst others, humans, from different angles, backgrounds, and poses. However, in these datasets, humans always make up a large part of the image by size. For our research, this is not the case, as humans in our footage take up a very small part of the images, meaning there are fewer pixels to describe the humans with, making their detection more difficult. There are also various long-range datasets. However, most of those do not contain humans (e.g., MOHR (Zhang et al., 2021) and DroneVehicle (Sun et al., 2022)). An example of a dataset that does allow long-range human detection is TinyPerson (Yu et al., 2020), which consists of helicopter footage, recorded above crowded beach scenes. Whilst the scale of the humans in the drone camera footage recorded by FRs resembles this dataset, the humans in the TinyPerson dataset are sometimes even smaller than the ones in our drone footage. In addition, the beach backgrounds in the TinyPerson dataset do not resemble ours, and they vary relatively little compared to the footage recorded by FRs.

Although these datasets have different characteristics compared to our dataset, we can still learn from these applications to inform our model choices. For example, Tang et al. (2023) trained a Faster R-CNN model along with several YOLO models on the TinyPerson dataset, achieving their best Average Precision of 9.5% with their modified version of YOLOv7. Additionally, YOLOv5 and Faster R–CNN have been trained and compared on drone footage recorded in nature backgrounds (Bachir & Memon, 2024), achieving mAP 0.5 scores of 96.9% and 91.0% respectively. Lastly, a skip-based Fully Convolutional Network (FCN) was trained on a set of long-range infrared images, resulting in an F1-Score of 98.93% (Haider et al., 2021). For this research, we analyse the performance of the YOLOv8 model, a leading model in the YOLO series.

DATA

In this paper, we train and evaluate our models on a custom-made dataset based on the drone footage of the *Safety Region Rotterdam-Rijnmond*. They have a historical record of some of the incidents where drones were used, going back to 2019. The footage is recorded in Full HD (1920 by 1080 pixels), 30 frames per second, by Zenmuse H20N cameras[1] on DJI Matrice 300 RTK drones[2]. The data contains a lot of variation, e.g., regarding light conditions (day and night), environmental background (e.g., industrial areas, rural, urban, indoor), and camera angles and distance (see Tables 1 and 2). For the creation of our dataset, we reduced the variation in the backgrounds by selecting exclusively daytime footage in industrial and residential areas, keeping only the video footage that was not zoomed in. From this, we only kept the 12 scenarios that contained humans and did not exclusively contain FRs. In the Methodology Section, we elaborate on the methods we used for preparing and labelling the footage from these scenarios to create a dataset. For examples of the selected footage, see Figure 1. In later stages of this research, we aim to incorporate more varied scenarios or train separate models for different situations.

METHODOLOGY

In this section, we go over the different steps we took to get to our preliminary results. Firstly, we elaborate on the methods used to prepare our custom dataset. Secondly, we explain our labelling process, after which we introduce our model choice. Lastly, we describe our evaluation set-up, where we differentiate between a quantitative and a qualitative evaluation.

[1] https://enterprise.dji.com/zenmuse-h20n
[2] https://enterprise.dji.com/matrice-300

Table 1. Scenario frequency for all available data. Some incidents span multiple categories, in which case they are counted in all.

Scenario category	Frequency
Industrial	67
Nature	50
Water	48
Urban	41
Indoor	8

Table 2. Light condition frequency for all available data.

Light condition	Frequency
day	104
night	53

Figure 1. Example images from the selected incidents, that cover a range of camera angles, backgrounds, and scales.

Data Preparation

In this section we explain the steps we took to go from the footage of the 12 selected incidents, as discussed in the Data Section, to an image dataset, containing 5,357 images. We also elaborate on the methods used to reduce this dataset into various representative and more balanced subsets.

The 12 incidents combined, contain a total of 87 videos with an average runtime of 7.5 minutes. We converted these videos into a set of images. We extracted the images at an interval of 150 frames (equivalent to 5 seconds) to mitigate redundancy, because the footage does not differ much from frame to frame, as the drones often move slowly or are stationary. This results in a total of 6,694 pictures. By filtering out the images without humans, 5,357 images were left (80.0%). We still noticed an overlap in some of the images, as well as an imbalance in the remaining data, as some incidents are over-represented by the number of images (due to different lengths of the original video footage).

In order to solve the imbalance problem, whilst simultaneously mitigating the perceived image overlap, we balanced the dataset by using a technique that reduces the entire dataset into a subset of x images. This needs to be done in such a way that the resulting subset is representative of the original dataset. A way to do this would be to not group the images by the scenario they stem from to create the reduced subset, but to cluster the data by using the backbone of a classification model, i.e., using a classification model to embed each image into a feature space, where all images that are similar as seen by the model will be close to each other. A classification model extracts certain features from an image, e.g., the relative frequency of green lines or the redness of the image, such that we can group the images on these features. A frequently used classification model is ResNet-50 (He et al., 2016). This model is pre-trained on the ImageNet 2012 classification dataset (Russakovsky et al., 2015), a dataset containing 1.28 million training images spread across 1000 classes, including animals, vehicles, common household objects, and more. The ResNet-50 backbone embeds each image into a 2,048 feature space. In this space, images that are close to each other are considered to be similar by ResNet-50. The classification step that follows the embedding is not needed, as the label does not help reduce the dataset.

Figure 2 is a visualisation of how all images with people were clustered, using t-distributed Stochastic Neighbour Embedding (t-SNE) (Van der Maaten & Hinton, 2008), which is a technique that reduces larger feature spaces into just two dimensions, trying to keep data points close together that are also close in the original space. It is clearly

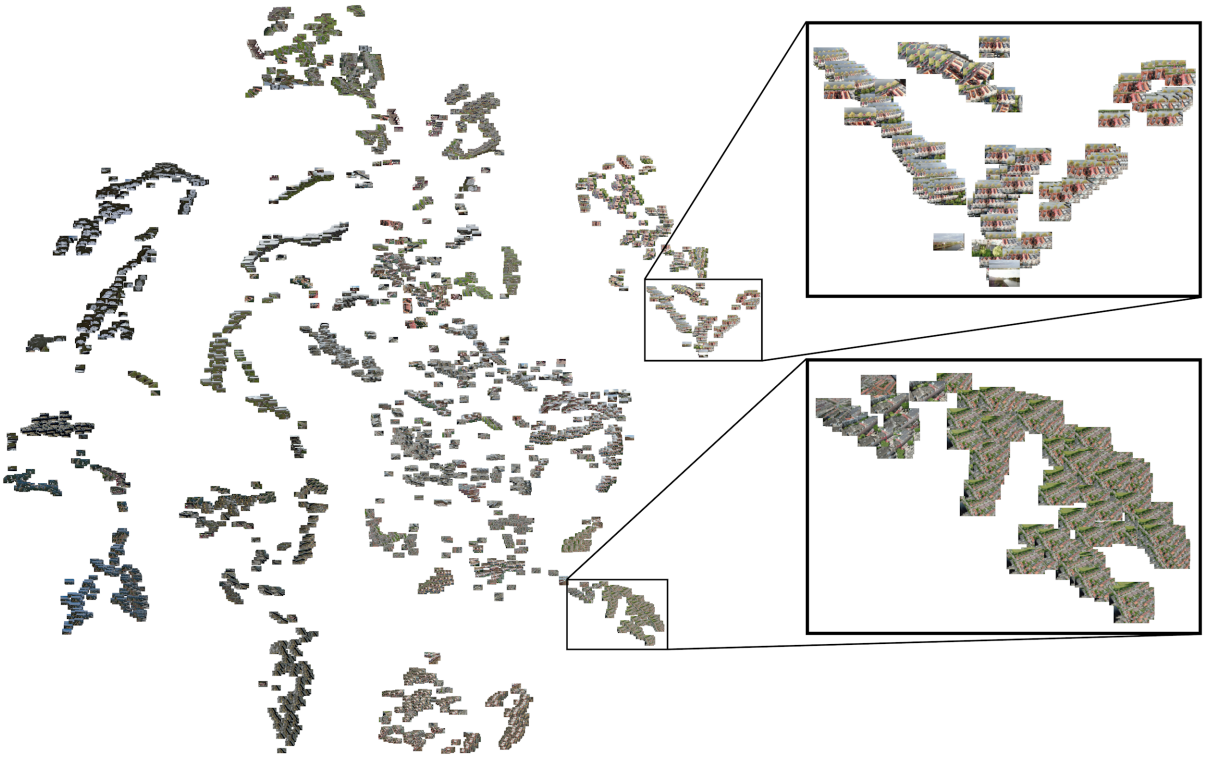

Figure 2. A t-SNE representation of the feature space, made by ResNet-50 on all the fire brigade's images that contain people.

visible in the zoomed-in regions of Figure 2 that the images with very similar structured content are close together and that the two regions are distinct from each other.

To generate a representative subset, the following heuristic was used. First, a random point was selected in the original feature space, after which the furthest possible point was selected as the second point. For this, the cosine similarity was used, to prevent the subset from primarily consisting of outliers. The third point is defined as the point that is the furthest possible point from both points, again using the cosine similarity. The next point is the furthest from all former points, and so on. Using this heuristic, multiple formations are repeatedly created, each with a different starting point. The formation that has the highest minimum cosine similarity will be kept. This is the formation that consists of the points that are the furthest possible from each other.

Using this technique, we created multiple subsets, decreasing in size, such that each subset is created from the closest bigger subset. This will make us able to easily increase the number of images in later stages of this research, knowing that the images we already prepared and labelled can be reused in bigger subsets. In Figure 3, a subset of 100 images, as selected from a subset of 200 images, is visualised in the entire feature space.

Data Labelling

For this paper, we have labelled a subset of 100 images. An example of a labelled image can be found in Figure 4. In the case of a crowd, we attempted to label every single human individually. We also labelled partially obstructed people, as they appear frequently and their detection might prove just as vital when it comes to detecting for example victims.

Model Choice

We trained the Machine Learning model YOLOv8[3] to detect humans in the fire brigade's footage. YOLOv8 is being regarded as state-of-the-art (Solawetz & Franchesco, 2023). Due to its multi-scale prediction method and anchor-free architecture (Jocher et al., 2024), it is specifically good at detecting small-scale objects, making it very applicable to the problem at hand. YOLOv8 is pre-trained on the COCO dataset (Lin et al., 2014). In this paper, we will fine-tune the pre-trained model.

[3] https://github.com/ultralytics/ultralytics

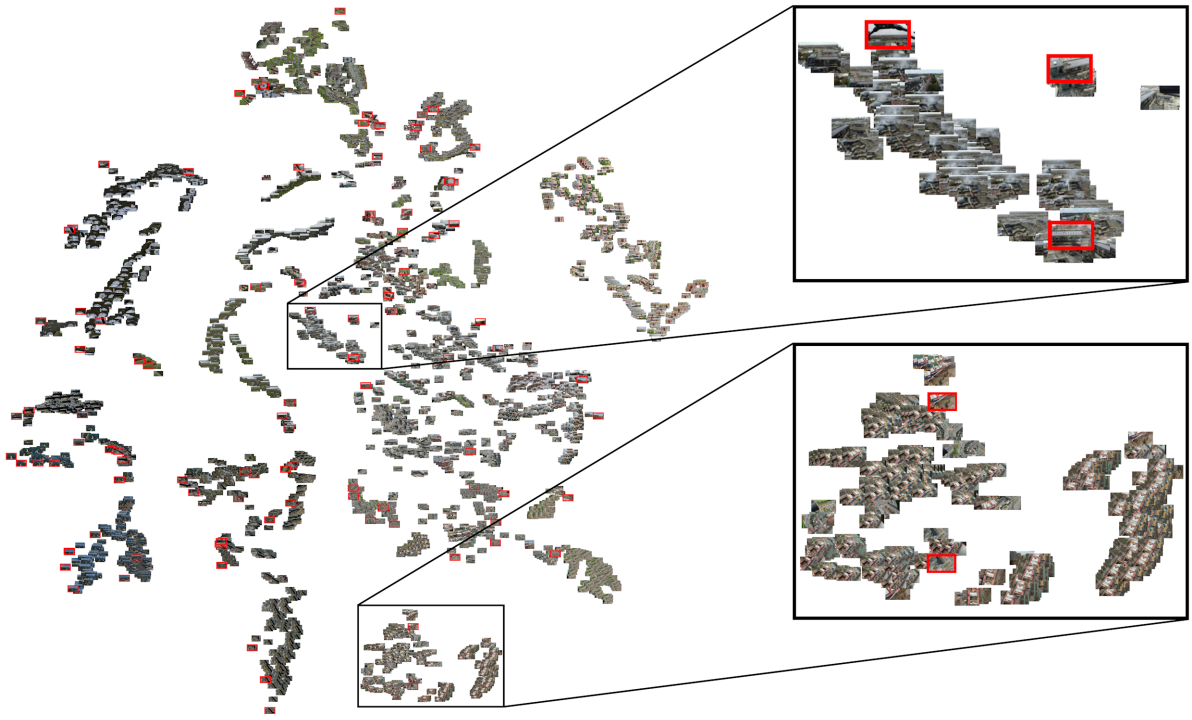

Figure 3. A t-SNE representation of the feature space, where the images with the red borders denote them as selected in the subset of 100.

Figure 4. An image with the labels overlaid. We zoomed in on some of the labelled humans.

Various versions of YOLOv8 are available, including *nano*, *small*, *medium*, *large*, and *extra large*, referring to the size of the model and its parameters. The larger the model, the more information it can store. A larger model has the capacity to learn more detailed representations of what a human is, but could also more easily overfit on smaller datasets. For our research, we decided to train these different versions and compared their results. We also explored the effects of different batch sizes, i.e., the number of images the model is exposed to before adjusting its weights. Larger batch sizes reduce the likelihood of the model overfitting the training data, but can negatively affect how long it takes the model to converge. In addition, we varied the input resolutions of the model, which have to be a multiple of the default 640 by 640 pixels, meaning we were only able to additionally analyse 1280 by 1280 pixels. The other parameters were left as default. The models' initial learning rates were 0.01. The stochastic gradient descent (SGD) optimization algorithm was used with a default momentum of 0.9 and attenuation coefficient of 0.0005. A constant seed of 42 was set for reproducibility purposes. The default mosaic augmentations were used. Each model was trained for 100 epochs, as we found this to be enough for all of the models to converge. The best scoring model, based on F1-Score, out of all these epochs was kept. Lastly, we randomly divided the 100 labelled images into a training, test, and validation set with a ratio of 8:1:1.

Evaluation Set-up

To evaluate our models, we use both quantitative and qualitative evaluation methods, which we describe in the following sections.

Quantitative Evaluation

To evaluate the performance of the model, we considered Precision, Recall, F1-Score, and mAP (Mean Average Precision). The precision and Recall of the model are calculated as follows:

$$Precision = \frac{TP}{TP + FP}, \qquad Recall = \frac{TP}{TP + FN},$$

where TP = True Positive, FP = False Positive, and FN = False Negative. Precision here quantifies the ratio of correctly predicted humans. Recall quantifies the proportion of actual humans that were correctly identified.

We combine recall and precision into an F1-Score to get a single score providing insight into both how often the model correctly identifies humans and how many humans it misses. A high F1-Score means the model is both accurate (with minimal False Positives) and thorough (with minimal False Negatives) in finding humans in images:

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}.$$

Since this score provides a balanced measurement of both False Positives and False Negatives, we primarily used this metric to intercompare models. For the best-performing model, based on the F1-Score, we will also analyse the exact number of True Positives, False Positives, and False Negatives, with the use of a confusion matrix. This gives us a more accurate and exact number to analyse if the best-performing model is business-acceptable or not.

Lastly, we used the mean Average Precision metric (mAP), which is commonly used in assessing the performance of object detection models. The Average Precision (AP) summarises the shape of the precision/recall curve, which shows how precision and recall change as we vary the threshold for what counts as a "correct" detection. The area under this curve gives us the average precision. This is done by taking the precision at each recall point and then averaging over all recall values. Mathematically, it's represented as:

$$AP = \sum_{r \in \text{recalls}} (r_i - r_{i-1}) \cdot p_i,$$

where r_i and r_{i-1} are consecutive recall values and p_i is the precision corresponding to the recall r_i.

The mAP is then computed as the average of AP over all classes:

$$mAP = \frac{1}{N} \sum_{k=1}^{N} AP_k,$$

where N is the number of classes and AP_k is the AP of class k. The mAP is a measure of accuracy for object detection models, typically assessed at various Intersection over Union (IoU) thresholds. IoU represents the percentage of overlap between the predicted bounding box and the ground truth bounding box:

$$IoU = \frac{A \cap B}{A \cup B},$$

where A is the actual bounding box and B is the bounding box as predicted by the model. For example, a mAP 0.5 score indicates the Mean Average Precision at an IoU threshold of 50%.

Qualitative Evaluation

Whilst these quantitative evaluation metrics enable us to intercompare models, they do not allow us to evaluate if any single model is business acceptable, i.e., can positively impact the decision-making of FRs. Relevant questions concern, for example, the potential effects of people being overlooked in the footage (False Negatives), people being marked multiple times, background objects being marked as people (False Positives), and the confidence values given for each detection. As a first step towards evaluating the model, we discussed the model results with an incident commander by presenting different scenarios with corresponding model output. The scenarios we presented differed in the number of humans that were present (e.g., a few humans or a crowd) and the setting of the incident (e.g., an industrial or residential area). The participant was asked to discuss the model output, including the impact on decisions, the desirability of different model behaviour, and other aspects relating to the understandability and usability of the model.

PRELIMINARY RESULTS

In this section, we describe the quantitative and qualitative preliminary results.

Quantitative Results

In this section, we present the preliminary results of the experiments we have run so far. We analyse the accuracy of all the different YOLOv8 models (i.e., *nano*, *small*, *medium*, *large*, and *extra large*) for a varying number of batch sizes and input resolutions. The training and testing was performed on a HP ZBook Studio 15.6 G8 laptop[4] with a Nvidia RTX A3000 GPU with 6GB of VRAM (Video Random-Access Memory). The relatively limited VRAM resulted in certain tests not being runnable, which explains why not every model is trained on every batch size.

640 by 640 resolution results

In order to analyse the effect different batch sizes have on the performance of the different models, we first ran them all with a 640 by 640 pixel input resolution, as we were not able to run larger batch sizes with higher input resolutions. This means that the images are scaled down, losing some detail in the progress. In return, we were able to train more model configurations. Due to potentially promising results, the YOLOV8s model on a batch size of 16 was trained on CPU.

In Table 3 we see that, judging by the F1-Scores, the two best-scoring models are YOLOv8s with a batch size of 12 and YOLOv8m with a batch size of 8. We do see a difference in Precision and Recall between the models. The YOLOv8s model seems to have a lower Precision score, but a higher Recall, which means that this model has more True Positives, at the cost of some extra False Positives.

1280 by 1280 resolution results

Due to our expectation that a higher input resolution would lead to better results, as there would be more pixels to describe the people with, we also tested the 1280 by 1280 input resolution. Not many of the tests would run on our hardware, which is why the larger batch sizes for the YOLOv8n model were trained on the CPU. In order to have a result for each model, we also introduced a batch size of 1. The 1280 by 1280 input resolution results in an overall higher performance compared to the 640 by 640 input resolution results (see Table 4). The YOLOv8n models show higher scores than the other models, with the batch size 16 model scoring the best in its F1-Score.

The validation set consists of 825 labelled people. This means that this is the worst possible False Negative score that can be reached, as well as the best possible True Positive score. In Table 5 we see that more than half (439/825) of the people are found. However, the model also marks 276 False Positives.

Qualitative Results

Based on the experimental results, for this evaluation, we used the output of the YOLOv8n model that was trained with an input resolution of 1280 by 1280 with a batch size of 16. During the discussions of the different scenarios and model outputs, the participant mentioned that it is important to differentiate between two aspects. First, the model can create False Positives, i.e., mark humans in the images where there are no humans (see examples in Figure 5). The participant mentioned that in general, False Positives should not be problematic, as having more False Positives probably also corresponds to having more True Positives and because it is better to find too many humans than too few, given that finding humans corresponds to the overall goal of the system. Also, the FR can

[4] https://support.hp.com/us-en/document/ish_4225460-4216961-16

Table 3. Comparative results of the varied YOLOv8 models with different batch sizes for an input resolution of 640 by 640 (%). The best-performing models, based on F1-Score, are highlighted in bold.

Model configuration		Performance metrics (%)			
Model name	batch size	Precision	Recall	F1	mAP 0.5
YOLOv8n	2	19.8	15.5	17.4	8.4
	4	24.3	16.6	19.7	10.1
	8	29.5	15.5	20.3	11.2
	12	25.5	20.2	22.6	11.6
	16	27.4	16.2	20.4	10.3
YOLOv8s	2	7.5	17.8	10.6	8.3
	4	24.1	17.0	19.9	10.9
	8	25.9	22.7	24.2	14.3
	12	**26.8**	**23.0**	**24.8**	**14.7**
	16	25.6	22.5	24.0	14.5
YOLOv8m	2	5.8	13.3	8.1	5.8
	4	37.0	18.4	24.6	16.3
	8	**34.6**	**19.3**	**24.8**	**16.4**
YOLOv8l	2	0	0	0	0
	4	7.9	13.1	9.8	6.6
YOLOv8x	2	7.1	3.6	4.8	1.9
	4	6.2	13.9	8.6	5.8

easily (visually) verify the result in the image. Second, the model can create False Negatives, i.e., not detecting humans in the image (see examples in Figure 6). The participant mentioned that this is more problematic, though it might not have a big impact if the humans are clearly visible. In any way, the participant mentioned that the model already provides output that is very helpful and would like to use it.

DISCUSSION

In this paper, we explored the use of state-of-the-art Object Detection models to detect people in long-range drone footage, to aid the decision-making process of FRs. We looked at the performance of different versions of the YOLOv8 model, for different batch sizes and input resolutions. Our best model (YOLOv8n with a batch size of 16 and an input resolution of 1280 by 1280 pixels) was able to detect 439/825 people in the test data, with 276 False Positives. This model had an F1-Score of 53.1%. An incident commander of the Dutch Fire Services feels confident, based on these preliminary results, that this model can aid the FRs' decision-making process for different use cases, such as locating people in the incident area faster and with less cognitive effort and aiding with decision-making regarding the deployment of teams and logistical efforts. The model can only be used to locate humans in the camera footage and cannot be used to identify (and monitor) individuals; furthermore, the model will only be used during incidents.

As a first start in this research, we reduced the variation in the drone footage and selected residential and industrial footage in daytime. In future iterations, we would like to expand the current model to include more different types of scenarios regarding environmental background and other light conditions. If this proves challenging, we alternatively will explore making specialized models for specific use cases or adding other sensor data, such as infrared.

Also, for now, we decided not to look at the effect of introducing images with no humans into the dataset. This may have caused the large number of False Positives of our model, as the model may have learned the number of people that should be present in each image. In the next steps, we will introduce different amounts of images without humans present to analyse the effect on the number of False Positives.

To create representative subsets of the selected dataset, we used a heuristic. Whilst we believe our approach to be grounded and methodical, we recognise that this approach has not yet been validated. It may also be the case that a different approach would result in a more applicable model. In the future, we will validate our approach and compare it to different techniques, using an already labelled dataset, to back up our claims.

Figure 5. Example predictions of the model. Each red box is a detected human, the number stands for the prediction certainty. We zoomed in on some False Positives (rooftop objects recognised as humans).

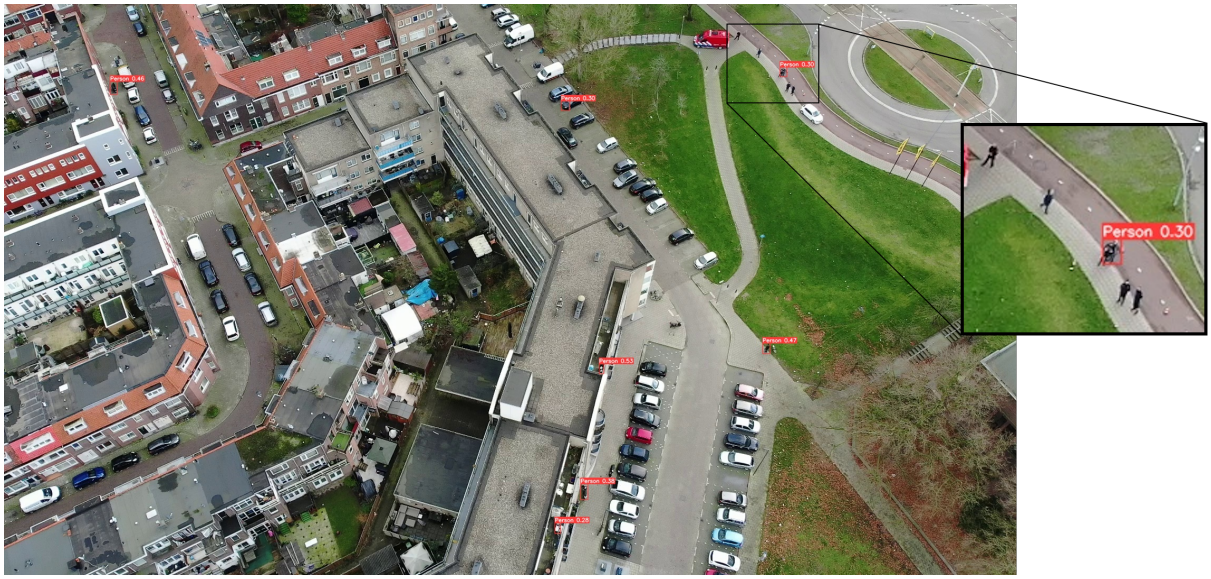

Figure 6. Example predictions of the model. Each red box is a detected human, the number stands for the prediction certainty. We zoomed in on some False Negatives (unfound humans by the model).

Table 4. Comparative results of the varied YOLOv8 models with different batch sizes for an input resolution of 1280 by 1280 (%). The best performing model, based on F1-Score, is highlighted in bold.

Model configuration		Performance metrics (%)			
Model name	batch size	Precision	Recall	F1	mAP 0.5
YOLOv8n	1	38.8	31.0	33.2	21.2
	2	42.6	38.3	40.3	27.5
	4	49.7	44.0	46.7	38.7
	8	54.1	47.8	50.7	43.3
	12	55.8	48.6	51.9	43.7
	16	**58.8**	**48.5**	**53.1**	**43.8**
YOLOv8s	1	16.2	34.8	22.1	20.7
	2	13.9	38.1	20.3	21.8
YOLOv8m	1	37.6	38.7	38.1	26.8
	2	42.8	23.1	30.0	22.7
YOLOv8l	1	15.6	35.4	21.7	16.9
YOLOv8x	1	7.5	25.1	11.5	10.7

Table 5. Confusion matrix of the YOLOv8n model with a batch size of 16 and an input resolution of 1280 by 1280 pixels on the test dataset containing 825 labeled humans.

		Predicted Classes	
		Human	Background
Actual Classes	Human	439	386
	Background	276	-

In addition to selecting the data, we also labelled it. In the current stage of our research, we were only able to label 100 images. We recognise that this is not a lot. We expect the addition of extra labelled data to have a significantly positive impact on the accuracy of the models. We will continue to label data until we see diminishing returns and also explore the use of active learning to aid our labelling process. With this technique, an AI model decides which images should be next to be labelled. Another technique that we think could aid us in our labelling process is the use of semi-automated labelling. An Object Detection model like the one we built could provide the first iteration of labels, after which we would only need to correct the output. This could save a large amount of time spent on labelling images.

As mentioned above, we only had access to relatively limited VRAM to run our models and for that reason could not train every model on every batch size. Current efforts aim to access better hardware so that all of the models can be run to compare performances.

Moreover, for this research, we only analysed YOLOv8 models; we realize that other models might also perform well. In the future, we will explore different models, such as R-CNNs and Vision Transformers, and see whether these provide better alternatives.

Lastly, for this research, we were only able to speak with one incident commander, which means that our results might not accurately reflect the actual implications of our preliminary results. As next steps, we aim to expand upon this evaluation by interviewing multiple stakeholders from a variety of FR functions and by evaluating the use of the model in the field.

ACKNOWLEDGMENTS

We would like to thank the *Safety Region Rotterdam-Rijnmond* for their participation in this research. This research was supported by the University of Applied Sciences Utrecht (HU) through a '*promotievoucher*'.

REFERENCES

Adams, S. M., Levitan, M. L., & Friedland, C. J. (2014). High resolution imagery collection for post-disaster studies utilizing unmanned aircraft systems (uas). *Photogrammetric Engineering & Remote Sensing*, *80*(12), 1161–1168. https://doi.org/10.14358/PERS.80.12.1161

Bachir, N., & Memon, Q. A. (2024). Benchmarking yolov5 models for improved human detection in search and rescue missions. *Journal of Electronic Science and Technology*, 100243. https://doi.org/https://doi.org/10.1016/j.jnlest.2024.100243

Georgiades, G., Papageorgiou, X. S., & Loizou, S. G. (2019). Integrated forest monitoring system for early fire detection and assessment. *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*, 1817–1822. https://doi.org/10.1109/CoDIT.2019.8820548

Haider, A., Shaukat, F., & Mir, J. (2021). Human detection in aerial thermal imaging using a fully convolutional regression network. *Infrared Physics & Technology*, *116*, 103796. https://doi.org/https://doi.org/10.1016/j.infrared.2021.103796

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 770–778.

Jocher, G., Laughing-q, Chaurasia, A., & Cagatay Akyon, F. (2024). *Yolov8*. Retrieved February 6, 2024, from https://docs.ultralytics.com/models/yolov8/

Khan, M. N. H., & Neustaedter, C. (2019). An exploratory study of the use of drones for assisting firefighters during emergency situations. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–14. https://doi.org/10.1145/3290605.3300502

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. *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 740–755. https://doi.org/10.1007/978-3-319-10602-1_48

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. https://doi.org/https://doi.org/10.1016/j.eswa.2021.114937

Nguyen, D. T., Li, W., & Ogunbona, P. O. (2016). Human detection from images and videos: A survey. *Pattern Recognition*, *51*, 148–175. https://doi.org/https://doi.org/10.1016/j.patcog.2015.08.027

Pi, Y., Nath, N. D., & Behzadan, A. H. (2020). Convolutional neural networks for object detection in aerial imagery for disaster response and recovery. *Advanced Engineering Informatics*, *43*, 101009. https://doi.org/https://doi.org/10.1016/j.aei.2019.101009

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, *115*, 211–252. https://doi.org/10.1007/s11263-015-0816-y

Solawetz, J., & Franchesco. (2023). *What is yolov8? the ultimate guide*. Retrieved February 11, 2024, from https://blog.roboflow.com/whats-new-in-yolov8/

Sun, Y., Cao, B., Zhu, P., & Hu, Q. (2022). Drone-based RGB-infrared cross-modality vehicle detection via uncertainty-aware learning. *IEEE Transactions on Circuits and Systems for Video Technology*, *32*(10), 6700–6713. https://doi.org/10.1109/TCSVT.2022.3168279

Tang, F., Yang, F., & Tian, X. (2023). Long-distance person detection based on YOLOv7. *Electronics*, *12*(6). https://doi.org/10.3390/electronics12061502

Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of machine learning research*, *9*(11).

Yu, X., Gong, Y., Jiang, N., Ye, Q., & Han, Z. (2020). Scale match for tiny person detection. *Proceedings of the IEEE/CVF winter conference on applications of computer vision (WACV)*, 1257–1265.

Zaidi, S. S. A., Ansari, M. S., Aslam, A., Kanwal, N., Asghar, M., & Lee, B. (2022). A survey of modern deep learning based object detection models. *Digital Signal Processing*, *126*, 103514. https://doi.org/https://doi.org/10.1016/j.dsp.2022.103514

Zhang, H., Sun, M., Li, Q., Liu, L., Liu, M., & Ji, Y. (2021). An empirical study of multi-scale object detection in high resolution uav images. *Neurocomputing*, *421*, 173–182. https://doi.org/https://doi.org/10.1016/j.neucom.2020.08.074