Article

Not peer-reviewed version

Investigating training datasets of real and synthetic images for swimmer localisation with YOLO

Mohsen Khan Mohammadi * , Toni Schneidereit , Ashkan Mansouri Yarahmadi , Michael Breuß

Posted Date: 7 February 2024

doi: 10.20944/preprints202402.0446.v1

Keywords: Object detection; Swimmer safety; Synthetic data; Background removal; YOLO architecture; Image augmentation

Article

Investigating Training Datasets of Real and Synthetic Images for Swimmer Localisation with YOLO

Mohsen Khan Mohmmadi *, Toni Schneidereit, Ashkan Mansouri Yarahmadi and Michael Breuß

Institute for Mathematics, Brandenburg University of Technology Cottbus-Senftenberg, Platz der Deutschen Einheit 1, 03046 Cottbus, Germany; schneton@b-tu.de (T.S.); yarahmadi@b-tu.de (A.M.Y.); breus@b-tu.de (M.B.)

* Correspondence: khanmmoh@b-tu.de

Abstract: In this paper we develop and explore a methodical pipeline for swimmer localisation in outdoor environments. The developed framework is intended to be used for enhancing swimmer safety. A main issue we deal with by the proposed approach is the lack of real world training data in such outdoor environments. Natural lighting changes, dynamic water textures and possibly barely visible swimming persons are key elements to approach. We account for these difficulties by adopting an effective background removal technique with available training data. This allows us to edit swimmers into natural environment backgrounds for the use in subsequent image augmentation. We created 17 training datasets with real images, synthetic images and a mixture of both to investigate different aspects and characteristics of the proposed approach. The datasets are used to train a YOLO architecture for the possible future application in real-time detection. The trained framework is then tested and evaluated on outdoor environment imagery acquired by a safety drone to investigate and confirm the usefulness for outdoor swimmer localisation.

Keywords: object detection; swimmer safety; synthetic data; background removal; YOLO architecture; image augmentation

1. Introduction

Swimmer safety in outdoor environments, which includes swimming pools [1], lakes [2], and seas [3], is a critical concern. One may refer to the report [4] ranking drowning as the third most significant cause of accidental injury worldwide [5], leading to approximately 320, 000 individuals to lose their lives due to drowning, which accounts for 7% of all injury-related fatalities [6].

The engineering of a drowning alarm system based on computer vision incorporates several complex tasks that requires robust algorithms. In this paper we focus on swimmer localization [7], which is in practice a major step within any supervised swimmer safety pipeline. The computer vision framework has to tackle probable complexities that an outdoor environment may exhibit. A possible candidate to develop a robust localisation is to make the swimmers independent from their background, which might be achieved by a background removal algorithm [8]. This may also facilitate training of computer vision methods since lack of training data taken in real world outdoor environments is an important issue when approaching the task.

Recently, drones equipped with a variety of sensors, namely visual, thermal, sonar, and also GPS systems, have displayed the potential to greatly enhance swimming safety in open water environments and already act as a valuable tool for lifeguards [9].

In our swimmer detection application, it is crucial that our utilised drone is embedded with a vision-based swimmer safety system to avoid inaccurate localisation of irrelevant floating objects, such as balls or boats, as swimming persons. The wrongly localised object can distract the supervisory drone, and this may lead to paying attention or even flying in a direction far from those who actually need help. To prevent such errors, it is important to adopt a cautious approach and follow established guidelines, such as those recommended in [10].

Our core vision-based algorithm of choice is in general a YOLO architecture [11–15] that is extremely fast, framing the localisation task as a regression problem achieved by adoption of a convolutional network. In current study, the canonical model architecture of YOLO henceforth mentioned as YOLOv1 [11] is discussed, and its different aspects in contrast to the adopted YOLOv3 [13] are highlighted. The YOLOv3 itself consists of different versions, which differ in the size of the architecture. This classification is according to the memory storage size, but the modelling principle is the same [16] for all YOLOv3 models. We report our results based on YOLOv3, motivated by the performance study [17] on different YOLO architectures. Let us note at this point, that the focus of the manuscript is not on the detection framework, but on the datasets utilised for training.

As required by our context, we opt to train and later deploy our model on data acquired from totally different environmental conditions. More specifically, we aim to train YOLO with different datasets consisting of real and synthetic images. The latter are acquired by background removed indoor pool images of swimmers, merged with real lake environment backgrounds. The real images are taken by a drone flying over a lake.

Literature Review

The task of object detection might be tackled based on their removal from complex and dynamic surroundings, where a background pixel value may change due to periodical or irregular movements [18]. The aquatic backgrounds are considered as highly dynamic surroundings, since the water texture results in a highly unstable background environment due to light reflections, waves, and other dynamic elements. With this, we opt to investigate the background removal as a pivotal role and present a concise overview of the existing research landscape in object detection within an aquatic background.

In the domain of interest concerning our current work, a notable part of the literature focuses on feature engineering approaches. For instance, the work [19] pursued the detection goal solely relying on color gradient and low-level techniques, with roots in subtraction of consecutive frames aiming to remove the swimmer's background. Though this enables a fast inference, if resulted in an indoor-specific threshold based approach having limited generalization ability. Let us note that, the authors did not provide a model or metric for result comparison. In [20] a video-based assistant system was developed to assist coaches to acquire swimmer positions leading to estimate their lap times. This is once again achieved through the utilization of a straightforward background modeling technique followed by an engineered blob detection technique, both based on modelling the color space of the background water.

In realm of background removal, an emerging trend incorporates deep learning based approaches [21–23] to accomplish the background removal task. A comparison study in [22] revealed the practicability of YOLOv5 [15] in contrast to other feature engineering based approaches [24,25] to remove distinct moving particles from their liquid background contexts. The proposed method in [21] incorporates an autoencoder along with a U-Net to accomplish the static and dynamic backgrounds generation tasks, respectively. Here a set of static backgrounds are used to train an autoencoder. Later, the trained autoencoder is evaluated by a moving-free foreground and the generated result is subtracted from the input image leading to a binary image used to train a U-Net which requires pixel-level labelled training images.

By considering the outdoor lakes as our environment of operation, we realize the importance of a robust background removal approach helping us to achieve our goals to detect swimmers [26]. To attain our desired pipeline we coupled a deep learning based background removal scheme called U2-Net [23] with a light weight YOLOv5 to comprise a robust vision based system that can effectively operate in real world environments.

Our Contribution

We proposed a robust pipeline to localise swimmers in an outdoor environment under uncontrolled conditions. Our pipeline performs localising swimmers across a lake using a YOLOv3 [13] architecture being trained on a a variety of real and synthetic images systematically forming different datasets for investigating the prediction accuracy. Our training datasets are augmented to account various environmental conditions not covered by the very limited amount of training images we have. In total, we successfully created 17 training datasets and one completely different validation dataset to investigate the impact of synthetically constructed images on the prediction accuracy. The main idea that caught our attention to produce this set of synthetic images is the lack of any publicly available data set of outdoor swimmers.

2. YOLO Models

In what follows, the YOLOv1 model along with its advancements leading to YOLOv3 architecture is discussed. Let us mention at this point, that we are using the YOLO framework as a tool and the focus of this manuscript is on the training with various datasets.

2.1. YOLOv1

The architectural representation of YOLOv1 is depicted in Figure 1, drawing inspiration from the GoogLeNet model [27]. YOLOv1 consists of 24 convolutional layers and performs downsampling on input training images sized at 448 × 448 × 3, ultimately producing a 7 × 7 × 30 predictor tensor.

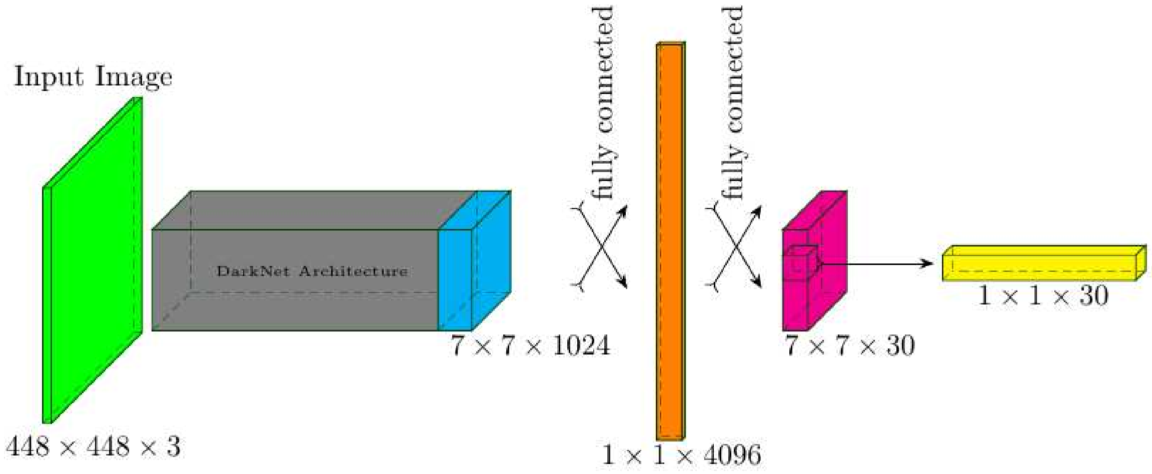


Figure 1. Utilising a 1 × 1 × 4096 layer, the YOLOv1 architecture processes input training images sized at 448 × 448 × 3, leading to a downsampling and generating a 7 × 7 × 30 predictor tensor. From the final layer, a 1 × 1 × 30 predictor vector, highlighted in yellow, is extracted. This vector encompasses 20 conditional probabilities and two vectors, each sized at 1 × 1 × 5, to represent the 20 labelled classes featured in the PASCAL Visual Object Classes Challenge (VOC) [28]. Within each 1 × 1 × 5 predictor, information is provided for two bounding box properties: the centroid (x, y), height (h), width (w), and their respective Intersection over Union (IOU) box scores [29].

In the application of YOLOv1, a grid with dimensions 7 × 7 is superimposed onto input training images sized at 448 × 448, encompassing all three channels. Within each grid cell, YOLOv1 makes predictions for two bounding boxes along with their corresponding class probabilities. These predictions are stored in 1 × 1 × 20 and two 1 × 1 × 5 vectors, respectively. The two aforementioned vectors collectively constitute the 1 × 1 × 30 predicted vector, highlighted in yellow in Figure 1. The YOLOv1 architecture integrates the components of object detection and classification into a cohesive and unified framework [11]. This integration is accomplished by employing a compound cost function, denoted as, which comprises three components: *localisation loss*, *confidence loss*, and *classification loss*. Collectively, these loss components contribute to the overarching training objective of YOLOv1, facilitating concurrent tasks of object detection and classification.

2.2. YOLOv3

One notable advantage that sets YOLOv3 apart from YOLOv1 is its ability to predict across multiple scales. This capability is accomplished through the incorporation of Darknet-53 [30], initially designed as a 53 layer network trained on ImageNet [31]. To enhance its detection capabilities, an additional 53 layers are added, originally a 53-layer network trained on ImageNet. resulting in a comprehensive 106-layer fully convolutional architecture for YOLOv3. Within the stacked-Darknet comprising 106 layers, upsampling and concatenation techniques are employed three times, generating feature maps with dimensions of $13 \times 13 \times 255$, $26 \times 26 \times 255$ and $52 \times 52 \times 255$. As documented in [13], the production of these feature maps involves up-sampling the corresponding feature maps from the two preceding layers by a factor of 4. Subsequently, these up-sampled maps are concatenated with their corresponding earlier feature maps from the network. According to the authors of [13], the utilisation of the technique referred to as the Feature Pyramid Network (FPN) [32], aims to acquire significant semantic information from the up-sampled features and more detailed information from the earlier feature maps.

Within YOLOv3, each output tensor with dimensions $1 \times 1 \times 255$ encompasses a total of $B = 3$ bounding boxes. These boxes are characterised by six attributes: centroid coordinates, dimensions, objectness score, and a set of 80 [33] conditional class confidences. With YOLOv3 making predictions on three distinct scales, a total of nine "derived" bounding boxes are anticipated. These derivations are executed using a set of predefined "anchor boxes," which are initially supplied to YOLOv3 during the preprocessing stage known as "dimension clusters" [12].

The primary purpose of anchor boxes is to establish a constrained set of predefined shapes derived from the dataset and the available ground truth boxes. This allows for a comparison during the training phase, where the ground truth boxes are matched against these anchors, and the model learns the transformations between the predefined anchors and the actual ground truth boxes. In this context, the model is trained by selecting the anchor box with the highest Intersection over Union (IOU) with the ground truth box. Utilising a K-means clustering approach [34], a total of nine anchor boxes are determined, each representing the mean anchor box within one of nine established clusters across different scales. This clustering is performed in reference to the COCO dataset [33]. The primary advantage of these prior boxes lies in their ability to enhance YOLOv3's capacity to predict multiple objects, accommodating various height and width aspect ratios on different scales.

3. Data preparation and experimental setup

In order to investigate the specific impact of real and synthetic training images on the detection accuracy, we propose a systematic testing approach. We relate the term (*i*) "real images" to images of swimmers captured by a done flying over a lake and (*ii*) "synthetic images" to images merged from two datasets. Let us stress the difficulty of our task that has roots in the dissimilarity of distributions from which the datasets are captured, namely the lighting conditions, the distance of the camera to the swimmers, varying angles of view and the limited availability of such data.

At this point we want to state that all images have been captured with the consent of the visible people. The used images are covered by data privacy regulations.

Real images

We have extracted 150 real images from videos of a drone flying over a lake with one or more swimmers visible. The swimmers have been captured in several different positions. Three example swimming styles are displayed in Figure 2. We used the original drone footage and cropped the images into a resolution of 416 by 416 for a more detailed representation of the swimming people.

Let us note at this point, that we target the issue of very limited data availability. That is, not only is the number of real images very limited, but also the number of swimming people and swimming styles. In addition, a major challenge is that the only available environmental condition in those images is sunshine. This has caused light reflections on the water surface and may introduce issues when detecting on, e.g, cloudy days. However, in the upcoming data augmentation paragraph we will further address and take on this issue.

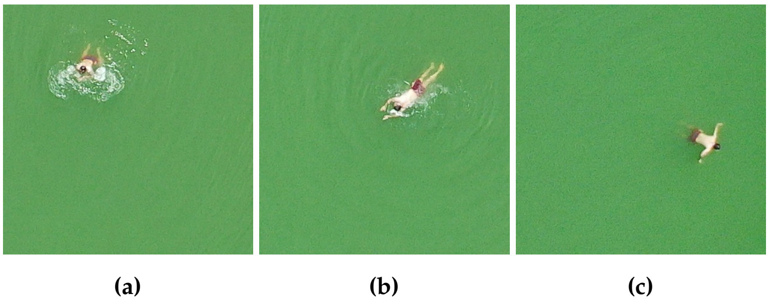


Figure 2. Three examples captured of one person, representing the real images. Different swimming styles may cause the appearance of foam around the swimmer. The limited number of swimming persons and swimming styles available motivated the creation of synthetic images.

Synthetic images

We have also created 150 synthetic images by using the Kaggle data set [35] that has 301 images each of size 600 × 400 pixels. Those images were captured by a camera located at a relatively far distance from the objects of interest and its surrounding water. The Kaggle images are acquired in an indoor environment with predefined lighting conditions, so that the swimmer can be captured along with its background water. One major contribution of our work is the proposed pipeline for merging two different distributions of data sets. Here, we consider some of the lake images that contain no swimmers as our target background. The motivation for this was to merge the lake water texture with the swimmers from the indoor environment. Our merging plan was to embed those indoor swimmers, with their backgrounds removed using U2-Net [23], on random positions and within the target backgrounds. Figure 3a-3d shows two swimming people from the Kaggle dataset and their background remove versions. Directly below, in Figure 3e-3g we have displayed a subset of our merged images showing the lake background images with the extracted swimmers.

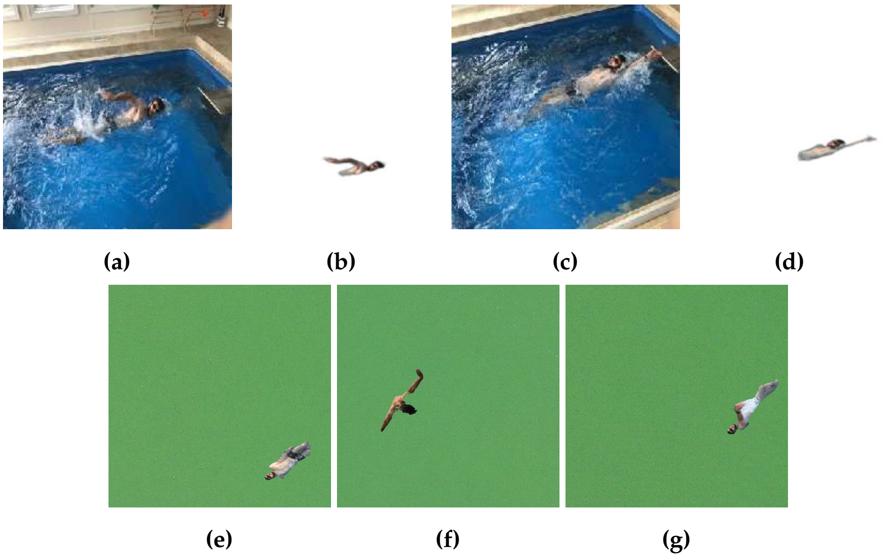


Figure 3. A pair of indoor pool swimmers (a) and (c) along with their corresponding background removed images (b) and (d) performed using U2-Net [23]. Swimmers of different swimming styles (e)-(g) have already been merged with lake backgrounds.

Validation images

In order to have a meaningful evaluation of the results, we use a complete separate and different dataset to test and validate the results. The images were also taken by a drone flying over a lake, on a different day with different people. In contrast to the real images from the training dataset, up to nine persons are visible in the validation set, which can be seen in Figure 4. We have carefully selected those images because of substantial differences compared to the real training images, namely more people are visible in one image, either standing in the water or showing additional swimming styles on a different background. In total, the validation dataset consists of 50 images.

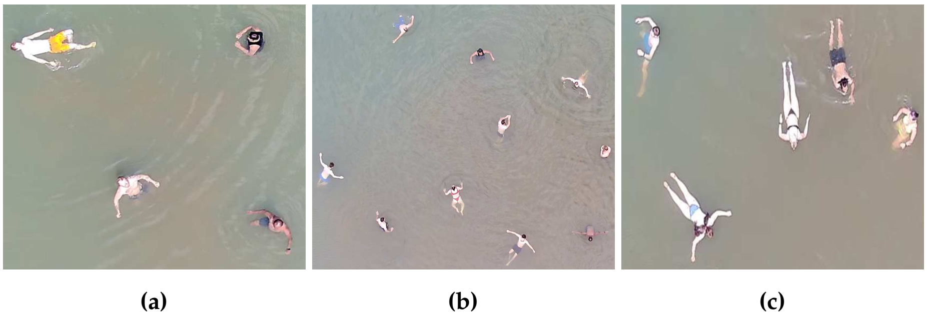
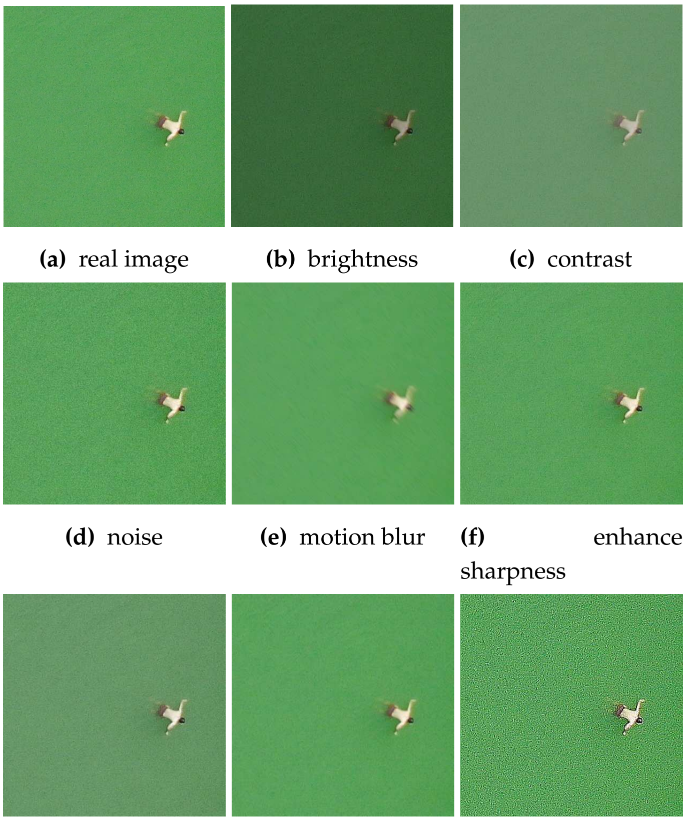


Figure 4. A representation of three examples from the validation dataset, which is different from the real images in the training dataset. Differences between these datasets are, e.g., the number of visible persons showing various swimming styles, the background (colour and texture) and camera settings.

Data augmentation

We have augmented the very limited amount of training images to create the datasets for our investigations. That is, editing images to change certain characteristics can simulate environmental conditions and increase the available number of images for training. More precisely, we have applied image processing techniques to change the following characteristics: *(i)* **brightness** (simulate different lighting conditions), *(ii)* **contrast** (enhance differences between lighter and darker regions), *(iii)* **noise** (simulate challenging lighting conditions, different distances and weather conditions), *(iv)* **motion blur** (simulate camera movement blur), *(v)* **enhance sharpness** (differences between objects and lighting condition changes), *(vi)* **enhance colour** (make object look different), *(vii)* **smooth** (enhance merging synthetic images), *(viii)* **enhance edge** (increase texture visibility) Figure 5 shows an example of the augmentations applied to a real image (Figure 5a). Although some augmentation technique look similar, the features extracted by CNN-based frameworks like YOLOv3 are always different and add flexibility to the dataset. The image augmentation techniques were implemented using the library imgaug [36].



(a) real image **(b)** brightness **(c)** contrast

(d) noise **(e)** motion blur **(f)** enhance sharpness

(g) enhance colour **(h)** smooth **(i)** enhance edge

Figure 5. Visualisation of the utilised augmentation techniques for an example image. Each image (real and synthetic) has been augmented several times with each method and different parameters.

Experimental setup

We have prepared 17 training datasets in total used for two experiments to investigate:

1. the impact of replacing real images by synthetic images.
2. the benefits of adding synthetic images to real images.

In experiment 1 we have trained YOLOv3 11 times, were each dataset has a fixed amount of 150 (raw) images before augmentation. The images were augmented 40 times each, so that in total one dataset consists of 6150 images. The 150 raw images are entirely covered by real images in the first dataset. Subsequently, the real images get gradually replaced by the synthetic images as shown in Table 1.

Table 1. Training dataset composition of real and synthetic images for each setup in experiment 1 and experiment 2

dataset	experiment 1	experiment 2
1	150 real + 0 synthetic	150 real + 25 synthetic
2	135 real + 15 synthetic	150 real + 50 synthetic
3	120 real + 30 synthetic	150 real + 75 synthetic
4	105 real + 45 synthetic	150 real + 100 synthetic
5	90 real + 60 synthetic	150 real + 125 synthetic
6	75 real + 75 synthetic	150 real + 150 synthetic
7	60 real + 90 synthetic	
8	45 real + 105 synthetic	
9	30 real + 120 synthetic	
10	15 real + 135 synthetic	
11	0 real + 150 synthetic	

For experiment 2 we have trained YOLOv3 6 times with a continuously increasing number of raw images, achieved by adding more and more synthetic images to the 150 real images, as displayed in Table 1. Again, each image has been augmented 40 times using the above mentioned image processing techniques.

4. Results

Please let us highlight again the importance of the framework accuracy when it comes to swimmer safety. Such a detection must be accurate to ensure the correct recognition of people struggling in outdoor water environments.

For training we have used the large YOLOv3 architecture together with 100 epochs for training and the standard parameters. The datasets have been separated for training (80%), testing (10%) and validation (10%) as it is required by the framework. For validation we have used a separate dataset with 50 images also taken by a drone showing different persons and conditions.

To quantify the results, we consider the mean Average Precision (mAP) metric. More specific, both mAP@.5 and mAP@.5:.95. The metrics use the average precision measure which results from the Intersection over Union (IoU) of the ground truth bounding box and the predicted bounding box. Defining a threshold for the IoU determines if a detection is True Positive (TP), when IoU is above the threshold, or False Positive (FP), otherwise. The letter means in other words, the model has made a prediction which is not enough overlapping with the ground truth. A False Negative (FN) detection occurs when the model predicts a correct instance as false. The precision is then computed as TP over the sum of TP and FP. One also needs to consider the recall, which is computed as TP over the sum of TP and FN. The precision is than plotted over recall and the area below the graph defines the Average Precision (AP). An additional averaging of the APs for all classes (e.g., dog, cat) defines the mAP. Setting the IoU threshold to 0.5 (or 50%) returns the mAP@.5 metric. More meaningful, however, is mAP@.5:.95. Here we have several thresholds, iteratively ranging from 0.5 to 0.95 in steps of 0.05. The mAP is then averaged (again) over all thresholds, resulting in mAP@.5:.95.

Turning to experiment 1, where we investigate the impact of replacing real images by synthetic images (see Table 1).

In Figure 6 we find the mAP@.5 and mAP@.5:.95 plotted over the datasets. Please see Table 1 for the specific combinations of real and synthetic images per dataset. On a qualitative level, there is no general trend visible. When it comes to a quantitative analysis, dataset 2 (90% real, 10% synthetic) provides the best mAP@.5 = 0.986, followed by dataset 6 (50% real, 50% synthetic) with mAP@.5 = 0.983 and dataset 7 (40% real, 60% synthetic) with mAP@.5 = 0.982. In the ranking for the second metric, dataset 6 has the best mAP@.5:.95 = 0.797, followed by dataset 3 (80% real, 20% synthetic) with mAP@.5:.95 = 0.792 and dataset 7 with mAP@.5:.95 = 0.791. While the worst results are always connected to dataset 10 and 11, dataset 1 with 100% real images is performance-wise somewhere in the middle. Those results are clear indications that features extracted from the synthetic images have similar characteristics as the real swimmers. One reason is related to the fact that the swimming persons stand out from the background, just like the merged swimmers in the synthetic images. However, since the validation dataset is completely different from the training dataset, especially with more swimming/moving styles, a different background and small waves, we find the results to be meaningful on different levels. In Figure 7 we can see one image from the validation set, detected using the trained YOLOv3 framework for dataset 1 (100% real, 0% synthetic), dataset 6 (50% real, 50% synthetic) and dataset 11 (0% real, 100% synthetic). Those detections confirm the results from Figure 6, where dataset 6 shows one of the best results. While in Figure 7a the swimmer on the left is detected with a prediction of 0.56, the detection of this person failed in Figure 7c. On the other hand, this swimmer is predicted with 0.98 for the dataset using 50% real and 50% synthetic images.

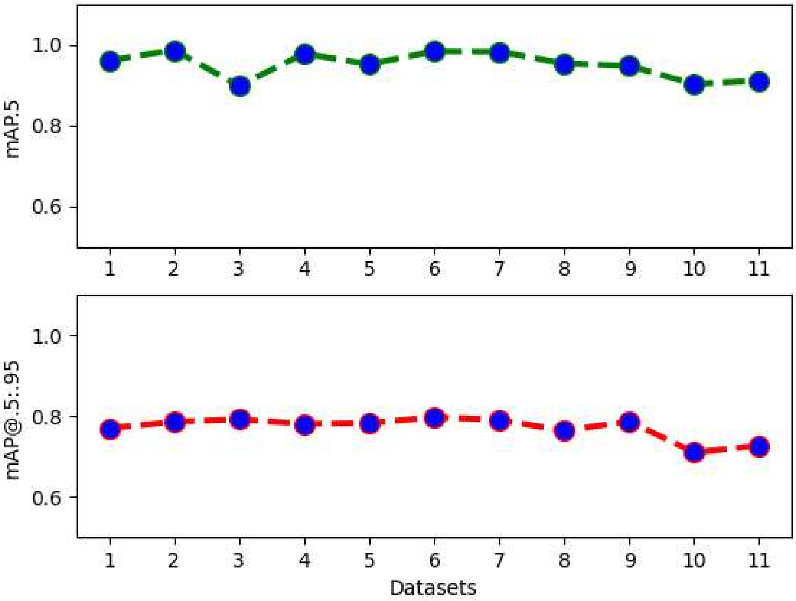
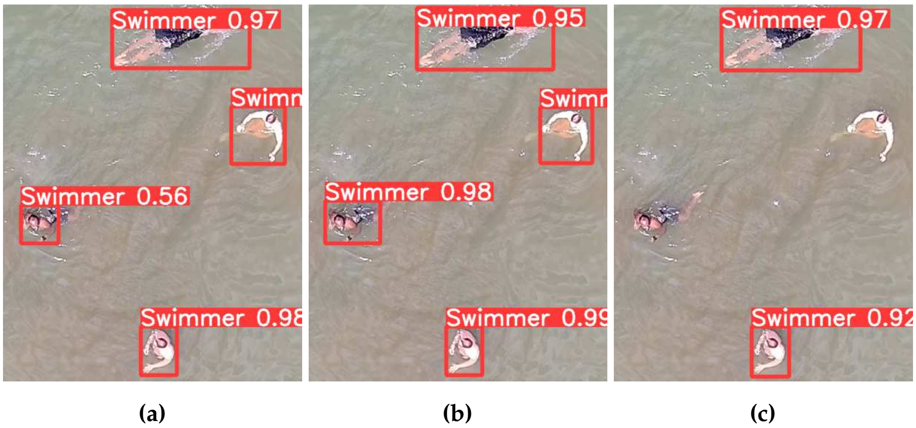


Figure 6. Results for both mAP@.5 and mAP@.5:.95 metric for experiment 1.

To conclude the first experiment, we see evidence that mixing real images with constructed (merged/synthetic) images can be beneficial for the detection accuracy. However, a clear statement of which ratio is the best is difficult to make. The best results are connected to 90% real, 10% synthetic, 50% real, 50% synthetic and 40% real, 60% synthetic.



(a) (b) (c)

Figure 7. A detected image based on the results from YOLOv3 training in experiment 1 with 7a dataset 1, 7b dataset 6 and 7c.

In experiment 2 we investigate the impact of adding synthetic images to the real images in order to see whether this is beneficial or not.

In order to evaluate the detection accuracy of YOLOv3 based on the training datasets from Table 1, we again consider the mAP@.5 and mAP@.5:.95 in Figure 8. The quantitative results for dataset 1 (150 real + 25 synthetic) return an mAP@.5 = 0.983 which the highest in this category and still better than dataset 1 from the previous experiment (mAP@.5 = 0.960) with only 150 real images and no synthetic images. We again find indications that the constructed images have a right to exist and datasets can have a certain amount to improve the accuracy. On the other hand, the results for mAP@.5:.95 have an interesting behaviour. Dataset 6 (150 real + 150 synthetic) has the highest mAP@.5:95 = 0.814 of all datasets and also outperforms every single dataset from the previous experiment. This is in general a desirable result, as we find it to indicate a more careful selection of the detected (less FP) objects while perhaps the detection itself is in some cases a little bit less good.

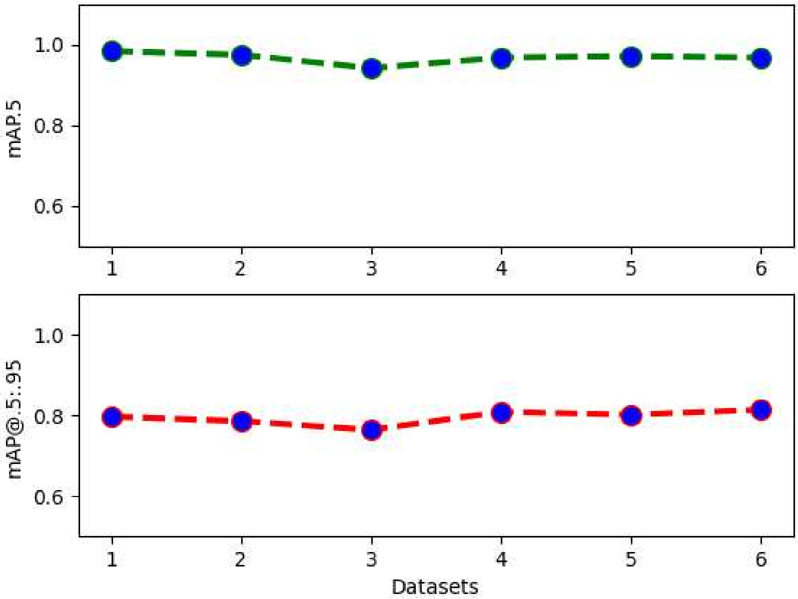
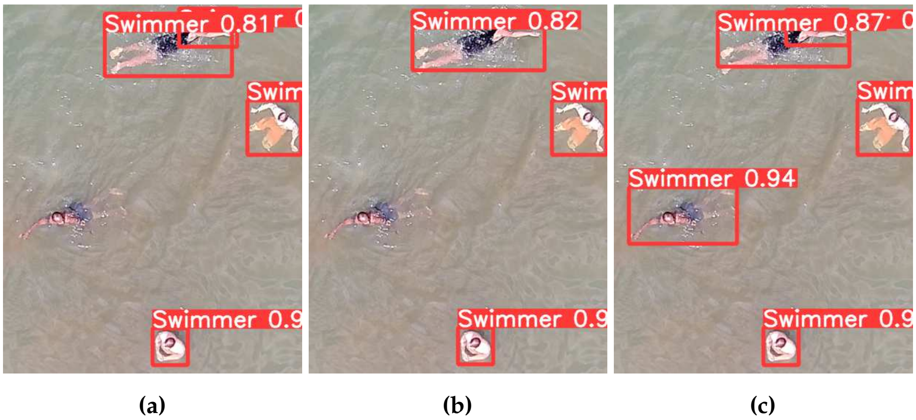


Figure 8. Results for both mAP@.5 and mAP@.5:.95 metric for experiment 2.

The predictions for dataset 1 (150 real + 25 synthetic), dataset 3 (150 real + 75 synthetic) and dataset 6 (150 real + 150 synthetic) are shown in Figure 9. In comparison to Figure 7, the current validation image has slight changes and is in fact another frame from the same sequence. However, even minor changes may have an impact on the detection, as the water texture (like waves and foam) and swimming styles change. We clearly observe that the prediction accuracy throughout the three images increases. However, the person at the top is detected as two different swimmers in Figure 9a and Figure 9c. In the previous paragraph we have stated that an increasing mAP@.5:.95 indicates a more careful selection of the detected objects. This statement continuous to hold, since the double detection of the same swimmer is still a TP prediction, even with a better accuracy. The main difference between the detections in Figure 9 is the swimming person on the left, who is only detected with dataset 6 and then directly with a high accuracy. We find this to be a positive effect of the added synthetic images. However, adding more (unaugmented) images to a datasets, which represent different situations, should in theory always have a beneficial impact. Nonetheless, this behaviour clearly indicates that the synthetic images serve their intended purpose.

Concluding this experiment, adding synthetic images to a fixed amount of real images has a benefit. An equal amount of both real and synthetic images returned the best predictions. Therefore we find that the constructed synthetic images are good representations of real swimming persons.



(a) (b) (c)

Figure 9. A detected image based on the results from YOLOv3 training in experiment 2 with 9a dataset 1, 9b dataset 6 and 9c.

5. Conclusion and future work

In the current paper we propose an approach to localise and classify swimmers across a lake captured by a drone. Swimmer safety in outdoor environments is a topic of great importance and so is the responsibility for a working approach when people rely on the system. Our approach uses a large YOLOv3 model with the focus on different training datasets consisting of real and synthetic images. The latter have been constructed/merged from two different datasets. The datasets have been systematically setup, augmented and investigated regarding their impact on the mAP accuracy measure. The trained YOLOv3 networks were tested on a complete different validation dataset.

We have seen that replacing real images by synthetic images for a fixed number of unaugmented images in a dataset has its benefits. The ratio showing the best results was 50%. We can also say that adding synthetic images to a fixed number of real images has a positive effect on the robustness of the detection. There are clear indications that continuing research on constructing synthetic images for datasets with very limited amount of real images is necessary.

The future work will focus on the effects of using different real-time detection frameworks as well as the dataset composition. Improving the appearance of the synthetic images will also be a major part of the future research direction.

Author Contributions: The authors declare, that the main idea and experiments were proposed by Mohsen Khan Mohammadi and the manuscript was written together with Toni Schneidereit , Ashkan Mansouri Yarahmadi and Michael Breuß.

Funding: The authors acknowledge partial funding of their work by Bundesministerium für Digitales und Verkehr (BMDV) within the project *RescueFly* as well as by Bundesministerium für Bildung und Forschung (BMBF) within the project *KI@MINT* ("AI-Lab").

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The authors declare, that upon reasonable request, the data and the code is available from the corresponding author.

Acknowledgments: The authors acknowledge all people who helped capturing the lake images for the dataset. This especially includes the organisers, the visible people and the drone flying team.

Conflicts of Interest: The authors have no conflicts of interest to declare that are relevant to the content of this article.

Abbreviations

The following abbreviations are used in this manuscript:

FPN	Feature Pyramid Network
IoU	Intersection over Union
mAP	mean Average Precision
TP	True Positive
FP	False Positive
FN	False Negative
CNN	Convolutional Neural Network
YOLO	You Only Look Once

References

1. Shatnawi, M.; Albreiki, F.; Alkhoori, A.; Alhebshi, M. Deep Learning and Vision-Based Early Drowning Detection. *Information* **2023**, *14*. doi:10.3390/info14010052.
2. Xiao, H.; Li, Y.; Xiu, Y.; Xia, Q. Development of outdoor swimmers detection system with small object detection method based on deep learning. *Multimedia Systems* **2022**, *29*. doi:10.1007/s00530-022-00995-7.
3. Cafarelli, D.; Ciampi, L.; Vadicamo, L.; Gennaro, C.; Berton, A.; Paterni, M.; Benvenuti, C.; Passera, M.; Falchi, F. MOBDrone: a Drone Video Dataset for Man OverBoard Rescue. Image Analysis and Processing – ICIAP 2022; Springer International Publishing: Cham, 2022; pp. 633–644.
4. Handalage, U.; Nikapotha, N.; Subasinghe, C.; Prasanga, T.; Thilakarthna, T.; Kasthurirathna, D. Computer Vision Enabled Drowning Detection System. 2021 3rd International Conference on Advancements in Computing (ICAC), 2021, pp. 240–245. doi:10.1109/ICAC54203.2021.9671126.
5. "Drowning," 27 April 2021. [Online].
6. "Drowning — United States, 2005–2009", CDC, 18 May 2012.
7. Sha, L.; Lucey, P.; Morgan, S.; Pease, D.L.; Sridharan, S. Swimmer Localization from a Moving Camera. *2013 International Conference on Digital Image Computing: Techniques and Applications (DICTA)* **2013**, pp. 1–8.
8. Piccardi, M. Background subtraction techniques: a review. 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583), 2004, Vol. 4, pp. 3099–3104 vol.4. doi:10.1109/ICSMC.2004.1400815.
9. Seguin, C.; Blaquière, G.; Loundou, A.; Michelet, P.; Markarian, T. Unmanned aerial vehicles (drones) to prevent drowning. *Resuscitation* **2018**, *127*, 63–67. doi:10.1016/j.resuscitation.2018.04.005.
10. Nguyen, A.; Yosinski, J.; Clune, J. Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images, 2015, [arXiv:cs.CV/1412.1897].
11. Redmon, J.; Divvala, S.K.; Girshick, R.B.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. *CoRR* **2015**, *abs/1506.02640*, [1506.02640].
12. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. *CoRR* **2016**, *abs/1612.08242*, [1612.08242].
13. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *CoRR* **2018**, *abs/1804.02767*, [1804.02767].
14. Bochkovskiy, A.; Wang, C.; Liao, H.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *CoRR* **2020**, *abs/2004.10934*, [2004.10934].
15. Jocher, G.; Chaurasia, A.; Stoken, A.; Borovec, J.; NanoCode012.; Kwon, Y.; TaoXie.; Michael, K.; Fang, J.; imyhxy.; Lorna.; Wong, C.; Yifu, Z.; V, A.; Montes, D.; Wang, Z.; Fati, C.; Nadar, J.; Laughing.; UnglvKitDe.; tkianai.; yxNONG.; Skalski, P.; Hogan, A.; Strobel, M.; Jain, M.; Mammana, L.; xylieong. ultralytics/yolov5: v6.2 - YOLOv5 Classification Models, Apple M1, Reproducibility, ClearML and Deci.ai integrations, 2022. doi:10.5281/zenodo.7002879.
16. Jung, H.K.; Choi, G.S. Improved YOLOv5: Efficient Object Detection Using Drone Images under Various Conditions. *Applied Sciences* **2022**, *12*. doi:10.3390/app12147255.
17. Schneidereit, S.; Yarahmadi, A.M.; Schneidereit, T.; Breuß, M.; Gebauer, M. YOLO-based Object Detection in Industry 4.0 Fischertechnik Model Environment. *Accepted for publication in Intelligent Systems and Applications: Proceedings of the 2023 Intelligent Systems Conference (Intellisys) Volume 1* **2023**.

18. Xu, Y.; Dong, J.; Zhang, B.; Xu, D. Background modeling methods in video analysis: A review and comparative evaluation. *CAAI Transactions on Intelligence Technology* **2016**, *1*, 43–60. doi:10.1016/j.trit.2016.03.005.
19. Benarab, D.; Napoléon, T.; Alfalou, A.; Verney, A.; Hellard, P. Swimmer's Head Detection Based on a Contrario and Scaled Composite JTC Approaches. *International Journal of Optics* **2020**, *2020*, 1–12.
20. Pogalin, E.; Thean, A.H.; Baan, J.; Schipper, N.; Smeulders, A.W.; others. Video-based training registration for swimmers. *Int. J. Comput. Sci. Sport* **2007**, *6*.
21. Bahri, F.; Ray, N. Weakly Supervised Realtime Dynamic Background Subtraction. *ArXiv* **2023**, *abs/2303.02857*.
22. Kara, E.; Zhang, G.; Williams, J.J.; Ferrandez-Quinto, G.; Rhoden, L.J.; Kim, M.; Kutz, J.N.; Rahman, A. Deep Learning Based Object Tracking in Walking Droplet and Granular Intruder Experiments, 2023, [arXiv:cs.CV/2302.05425].
23. Qin, X.; Zhang, Z.; Huang, C.; Dehghan, M.; Zaiane, O.R.; Jagersand, M. U2-Net: Going deeper with nested U-structure for salient object detection. *Pattern Recognition* **2020**, *106*, 107404. doi:10.1016/j.patcog.2020.107404.
24. Zivkovic, Z. Improved adaptive Gaussian mixture model for background subtraction. Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004., 2004, Vol. 2, pp. 28–31 Vol.2. doi:10.1109/ICPR.2004.1333992.
25. Zivkovic, Z.; van der Heijden, F. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters* **2006**, *27*, 773–780. doi:10.1016/j.patrec.2005.11.005.
26. Yarahmadi, A.M.; Breuß, M.; Mohammadi, M.K. Explaining StyleGAN Synthesized Swimmer Images in Low-Dimensional Space. Computer Analysis of Images and Patterns; Springer Nature Switzerland: Cham, 2023; pp. 164–173.
27. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions, 2014. doi:10.48550/ARXIV.1409.4842.
28. Everingham, M.; Eslami, S.M.; Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes Challenge: A Retrospective. *Int. J. Comput. Vision* **2015**, *111*, 98–136. doi:10.1007/s11263-014-0733-5.
29. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression, 2019. doi:10.48550/ARXIV.1902.09630.
30. Bochkovskiy, A.
31. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. 2009 IEEE conference on computer vision and pattern recognition. Ieee, 2009, pp. 248–255.
32. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection, 2016. doi:10.48550/ARXIV.1612.03144.
33. Lin, T.; Maire, M.; Belongie, S.J.; Bourdev, L.D.; Girshick, R.B.; Hays, J.; Perona, P.; Ramanan, D.; Doll'a r, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. *CoRR* **2014**, *abs/1405.0312*, [1405.0312].
34. Lloyd, S. Least squares quantization in PCM. *IEEE Transactions on Information Theory* **1982**, *28*, 129–137. doi:10.1109/TIT.1982.1056489.
35. Coughlin, S. Swimmers, 2021.
36. Jung, A.B. imgaug. https://github.com/aleju/imgaug, 2018.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.