Proceedings of the OAGM Workshop 2022 DOI: 10.3217/978-3-85125-954-4-09

Exploring Learning-Based Approaches for Bomb Crater Detection in Historical Aerial Images

Marvin Burges, Sebastian Zambanini, Robert Sablatnig
Computer Vision Lab, TU Wien
1040 Vienna, Austria
{mburges, zamba, sab}@cvl.tuwien.ac.at

Abstract

Many countries were the target of air strikes during World War II. The heritage of these attacks is still present today, as numerous unexploded bombs are uncovered yearly in Central Europe. While these bombs pose a significant explosion hazard, they can be inferred from the existence of craters. Therefore, analyzing aerial images from World War II surveillance flights allows for preliminary risk estimation. In this paper, we train and evaluate 12 different object detector architectures and compare them to a crater detection algorithm on our custom historical aerial dataset. We show that modern detectors, in combination with a large enough historical aerial crater dataset, can outperform a current method for crater detection, achieve a precision of 0.6 and a recall of 0.6 on our dataset, and can process large remotely sensed images within seconds, rather than minutes. Additionally, pretraining and different dataset extensions are evaluated and discussed.

1. Introduction

Although the last air raids of World War II happened more than 70 years ago, UneXploded Ordnances (UXOs) still pose a significant explosion hazard for European construction projects [11]. Specialized companies provide a preliminary risk estimation by reviewing and interpreting aerial images from World War II surveillance flights over the area of interest. To generate these risk estimations, historical aerial images have to be georeferenced, and all objects that indicate increased combat activity have to be marked. Currently, both the georeferencing task and the search for increased combat activity are performed manually by specialists. The goal of our work is to automatically generate "explosive ordnance maps" from selected images, by detecting increased combat activity. These "explosive ordnance maps" indicate whether an area is likely to be contaminated and therefore may contain UXOs, while in uncontaminated areas, UXOs are unlikely [11]. To achieve this goal, we survey the performance of existing fully-automatic object detectors in detecting increased combat activity by training them on our custom dataset consisting of historical aerial images.

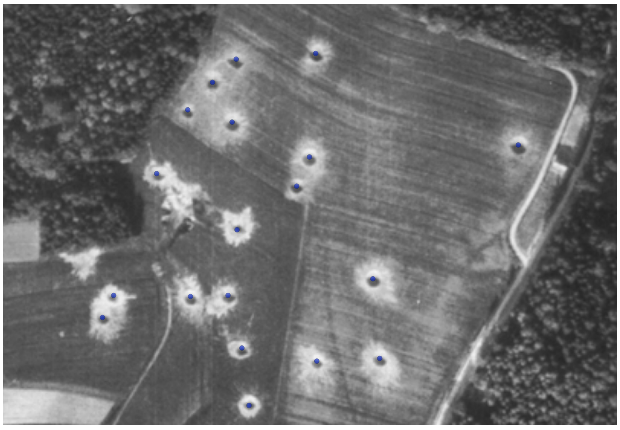


Figure 1. An example detection result of the best performing detector on a rural area. Blue point: Prediction, best viewed in color.

Specifically, we focus on the detection of bomb craters, as they are the most abundant type of warfare-related object visible in aerial images. Furthermore, they represent direct evidence for the presence of an UXO, as it is assumed that 10 - 15 % of all bombs dropped during World War II did not explode [2]. We concentrate on the detection of warfare-related objects instead of the prediction and segmentation of potentially contaminated areas due to explainability reasons. As this is a task with potentially dangerous consequences, the network results always have to be verified by an expert. An example for this task is given in Fig. 1, where the automatic detections on a image from the H2OPM dataset [25] are given. In this work, we compare Convolutional Neural Networks (CNNs)-based detectors on the task of automatic detection of bomb craters and compare our findings with a crater detection method from the literature [2]. We specifically focus our comparison on network architectures that can be retrained on consumer hardware, as we intend to focus on domain adaptation in future work, which can require an end-user to retrain on newly obtained data of a different domain. We will include the best performing network in a plugin for the geographic information system *QGIS*[1], for which we will release the code[2] and the weights[3] for the best performing networks can be used directly with the original YoloV5 implementation of Glenn Jocher *et al.* [8].

The remainder of this paper is structured as follows. First, in Sec. 2 the related work on object detection in historical aerial images and related domains is presented, followed by the data used for this paper in Sec. 3. The experimental setup is described in Sec. 4 and results and discussion are given in Sec. 5. Finally, a summary and potential future work is presented in Sec. 6.

2. Related Work

As the amount of related work on the topic of crater detection in historical aerial images is limited, we include selected methods for mars and moon crater detection, which are visually similar to bomb craters.

Brenner *et al.* [2] developed an approach to automatically detect craters in historical aerial images using a machine learning approach based on a CNN. They use a sliding window in combination with DenseNet [10] to extract candidate crater positions from the image and then use post-processing to refine the detections. These post-processing steps include a spatial proximity prior, as bombs are dropped in clusters thus, "lonely" bombs are likely a false positive, non-cluster suppression, as due to the overlap of the sliding window, bombs should be detected multiple times, as outliers are detections that are not part of a cluster detection and a non-maxima suppression to reduce multiple detections. Overall, their approach achieved a precision/recall of 90.7%/91.3% with the same amount of craters and background images in the test set. However, with a more realistic distribution of around 1:250, the precision is reduced to 4%. In [11], Kruse *et al.* assume that multiple images of the same area exist, based on which they propose an approach that combines the individual detection results of a stochastic approach based on marked point processes. This increased the F1-score from 39 % (based on single images) to 67 % (based on multiple images). In [12] Kruse *et al.* further evaluate this method by examining the influence of random number generation. They also compare their approach to a Faster RCNN object detector [17] trained on their dataset. The results show that the CNN can outperform their approach if the correct threshold is selected. However, they also note that in a scenario where only a limited amount of training data is available, their approach delivers superior results.

Wu et. al. [23] propose a Crater Detection Algorithm (CDA) called SUNnet 3+, that is based on the UNET architecture [18] and detects craters in the digital elevation model of Mars. The CDA proposed in [9] aims at detecting lunar craters in images in real-time by a crewed lunar lander during the landing procedure, based on a modified YoloV4 [1] architecture.

The listed publications show that, while learned crater detectors have been evaluated before, only one publication trains an object detector on this task. Brenner *et al.* [2] extracts regions via a sliding window and classifies the patches via a CNN and Clermont *et al.* [4] use a blob detector, also in combination with a CNN as classifier. Only in [12] Kruse *et al.* train and evaluate an object detector and show its potential. The primary challenge, frequently mentioned, was the lack of training data [4, 11, 12]. However, due to an industry partnership, we have access to a dataset suitable for training. Hence, this paper evaluates different object detectors trained on crater samples. We expect that by training object detectors with learned regions proposals we can outperform current crater detection methods.

3. Data

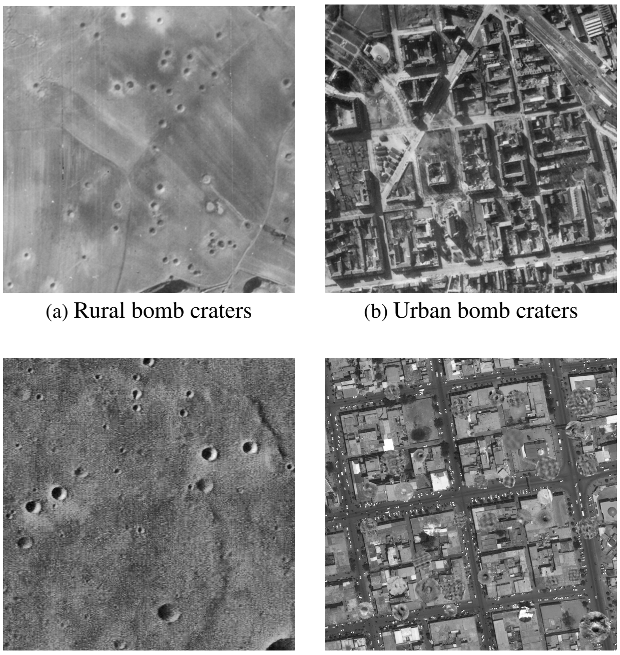
To the best of our knowledge, no dataset for crater detection in historical aerial images is publicly available. Therefore, we use our dataset, which covers both urban and rural areas. These images originate from finished projects in which experts georeferenced the historical aerial images and annotated the bomb craters. In total, 111 images are georeferenced, and a total of 19,506 craters have been marked in the images. Note, the analysis per image is only performed within the Region Of Interest (ROI). As a result, we had to ignore all regions outside of the ROI, as no ground truth data is available outside of the ROI. The images were made between 1943 and 1945 and contain craters with a minimum size of 1m, an average size of 8m, and a maximum size of 17m. The minimum image size is $2,274 \times 2,388$, the average is $11,626 \times 10,864$, and the maximum image size is $16,714 \times 16,973$. All images are split into 960×960 images with an overlap of 10% for training for a total of 3,711 images. The Ground Sampling Distance (GSD) for all images is normalized to 0.25m, and they cover an area of 505 km^2. An example of a rural image can be seen in Fig. 2a, one of an urban image in Fig. 2b, both images are from the H2OPM dataset [25] and are also part of our dataset. Additionally, we experimented on 12 panchromatic Martian satellite images [6] with a crater size of less than 5 km and a total amount of

[1] `https://qgis.org/en/site/`

[2] `https://github.com/mburges-cvl/QGIS_Plugin_for_OAGM_2022`

[3] `https://owncloud.tuwien.ac.at/index.php/s/AxarN33AnClCDhA`, PW: "oagm2022"

around 42,000 craters (Fig. 2c). These images were also split into 960×960 images with an overlap of 10% for training, which resulted in 384 images. We also generated a synthetic dataset, which is based on the XVIEW dataset [13] were 16,931 cut-out craters from our training dataset are imprinted onto the XVIEW images. An example of the synthetic data is presented in Fig. 2d. We discarded the original classes of the XVIEW dataset, converted the images to grayscale and to a GSD of 0.25, and augmented the crater patches before imprinting them in the image. The augmentation consists of rotation, horizontal flipping, and Fourier Domain Adaptation (FDA) [24], with the target being the XVIEW-image. This resulted in 17,234 960 × 960 images with a total of 603,190 craters and a similar GSD to the original project images.



(a) Rural bomb craters

(b) Urban bomb craters

(c) Martian craters

(d) Synthetic bomb craters

Figure 2. Four example images from our dataset. a) and b) Images from company projects, c) Martian satellite image from [6] and d) Image from the XVIEW-Dataset [13] with synthetic craters.

4. Experimental Setup

In this section, we present the experiments conducted for this paper. We start by evaluating different object detectors on our historical crater dataset, then compare the results with the crater detection method proposed by Brenner *et al.* [2]. We continue with the pretraining and synthetic data results and finish with the evaluation metrics.

4.1. Evaluation of State-of-the-Art Object Detectors

For this work, we evaluated 12 different object detection architectures. An overview of the networks can be seen in Tab. 1, which highlights the trainable parameters as well as whether the networks use one stage or two stages. Single-stage networks perform classification and regression on dense anchor boxes without generating a sparse ROI set, while two-stage networks first generate sparse region proposals, which are then further regressed and classified in a second stage [14]. We chose these specific networks as they are capable of running on consumer hardware and because these networks have been repeatably used for few-shot learning. We trained all networks on the historical aerial dataset with an image size of 960 × 960 pixels and a batch size of 2 (32 for YoloV5n), until the Average Precision (AP) started plateauing, which was between 10 - 50 epochs depending on the dataset size (larger datasets required more epochs). For the data augmentation, we relied on the Albumentations framework [3]. We used common augmentations like blur, random brightness and contrast changes, rotation, translation, and histogram equalization, but we also used the mosaic augmentation method proposed in [1] as well as FDA. All networks were pretrained on COCO before being finetuned on our crater dataset and used the same anchors (if applicable).

Table 1. Evaluated networks with their respective parameter count in million and their architecture style.

Name	Parameters (M)	Stages
YoloV5n [8]	1.9	One
YoloV4 [1]	27.6	One
YoloV7 [22]	37.1	One
Faster RCNN R 50 [17]	41.3	Two
Faster RCNN R 101 [17]	60.2	Two
YoloV3 [16]	65.3	One
ScaledYoloV4(-p5) [21]	70.2	One
EfficientDet(-d7x) [19]	76.8	One
YoloV5x [8]	86.7	One
YoloX [7]	99.0	One
Faster RCNN X 101 [17]	104.4	Two
YoloR(-d6) [20]	151.0	One

4.2. Comparison with State-of-the-Art Crater Detection Methods

To the best of our knowledge, no historical aerial crater detection frameworks for comparison are publicly available. We therefore chose to compare the best performing detector from Sec. 4.1 to the detector proposed by Brenner *et al.* in [2] on our crater data. However, as we did not have access to the original version, the approach was re-implemented. Similar to the original implementation, we used a 40-layer DenseNet with an input size of 32 × 32 pixels and trained it on a binary classification problem. We trained the network for 100 epochs on the 42.172 crater patches extracted by a sliding window from the training set with roughly the same amount of negative examples. Differently to the original approach, however, we use patches of size 80 × 80 pixels instead of 20 × 20 pixels to achieve a similar window size of $20m \times 20m$ as our dataset has a GSD of 0.25m instead of 1m. These patches are then resized to the network input size of 32 × 32 pixels. We evaluated the approach in two ways. First the classification way, where we extracted all 1.614 crater patches from the validation set as well as the same amount of negative samples (resulting in a 1:1 ratio of positive and negative samples). Second in the (more realistic) object detection way, where the network is applied as a sliding window to the validation images (resulting in roughly a 1:250 ratio of positive and negative samples). In a final experiment, we compared the run-time of the approach from Brenner *et al.* with YoloV5n and YoloV5x on one example image with the size of $10,644 \times 10,042$.

4.3. Pretraining and Synthetic Data

We also experimented with different strategies to improve the training result. One idea was to pretrain the network. In our case, we chose COCO, the Mars dataset described in Sec. 3 and the XVIEW dataset. Additionally, we experimented with increasing the raw crater dataset size by adding a combination of the Martian and synthetic data to our historical aerial crater dataset. The intention with the Martian data was to add more crater variants, while the idea for the synthetic data was to add more urban structures to the dataset. We again chose the best performing detector from Sec. 4.1 and trained it on the different dataset combinations.

5. Results and Discussion

In this section, we present the results of the experiments. We start by presenting and discussing the quantitative results in Sec. 5.1 and finish with a qualitative analysis of two example images in Sec. 5.2. Both show the difficulty related to detecting craters in historical aerial images.

5.1. Quantitative Results

The trained object detectors presented in Sec. 4.1 were evaluated on the test set of the historical crater dataset described in Sec. 3. The results can be seen in Fig. 3. The graph shows the precision recall curve for all networks presented in Tab. 1. It is visible that YoloV5n, YoloV5x, and YoloR perform similarly and better than the other tested networks. One can also see that YoloV3 and Faster RCNN R 50 perform significantly worse. This is due to the coarse search grid of YoloV3, which hinders the detection performance of small objects in the images, as stated by Pham *et al.* [15]. Similarly, the Faster RCNN family has issues with small objects, as has been shown by Eggert *et al.* [5]. This could be a possible explanation for why the Faster RCNN family is outperformed overall by the Yolo variants and why Faster RCNN R 50, in particular, is underperforming. Overall it is shown that the majority of the tested networks have similar results, which could be due to the training set being comparatively small for an object detection dataset (compared to COCO, for instance). Overall, YoloV5n is one of the best performing networks and also has the lowest amount of trainable parameters which makes the training less prone to overfitting.

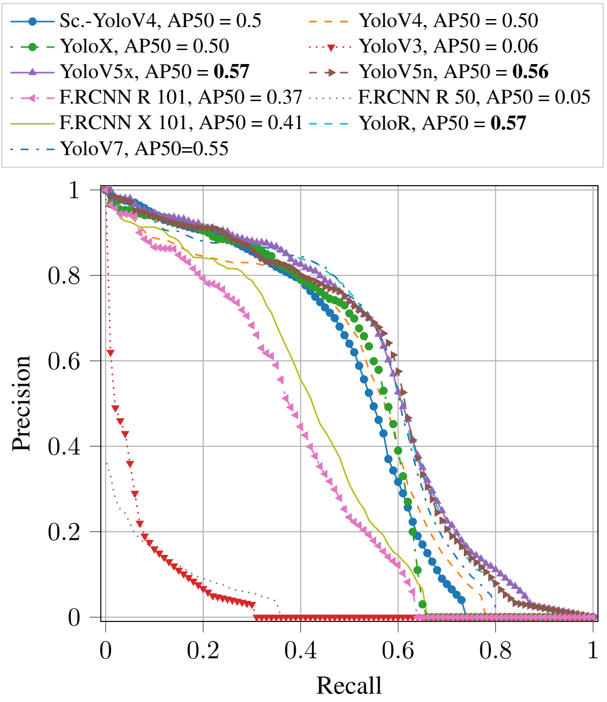


Figure 3. Precision-Recall-Curves for the different network architectures trained on the crater dataset.

The comparison with the approach of Brenner *et al.* is presented in Tab. 2. It shows that their approach achieves a precision of 0.91 and a recall of 0.87 during evaluation with a 1:1 distribution of positive and negative samples on our dataset, however their real-world performance is significantly worse. We were unable to achieve any meaningful detections with the approach, which is also reflected in a precision of 0.05 with a recall of 0.01 during the evaluation with a 1:250 distribution. To further show the edge YoloV5n has, we present the time for a detection of an image with a, for this task, common size of 10644 × 10042 in Tab. 3. It is visible that YoloV5n is 58 times faster than the approach proposed by Brenner *et al.* with a precision of 0.6 and a recall of 0.6.

We further explored YoloV5n on different dataset com-

Table 2. Precision and recall for Brenner *et al.* on a synthetic (1:1) and a more realistic (1:250) distribution of positive to negative training patches, compared to YoloV5n. YoloV5n does not use a sliding window to generate patches but instead is applied to the whole image.

Approach	Precision	Recall	Positive-Negative-Ratio
Brenner *et al.*	0.91	0.87	1:1
Brenner *et al.*	0.05	0.01	1:250
YoloV5n	**0.6**	**0.6**	1:250

Table 3. Runtime comparison of the best performing detector and the approach from Brenner *et al*. Measurement: A $10,644 \times 10,042$ example image, split into 144 960×960 image patches (with overlap) for YoloV5. GPU: Nvidia T500 (Mobile).

Approach	Runtime
QGIS + Brenner *et al.*	361s
QGIS + YoloV5n	**6.2s**

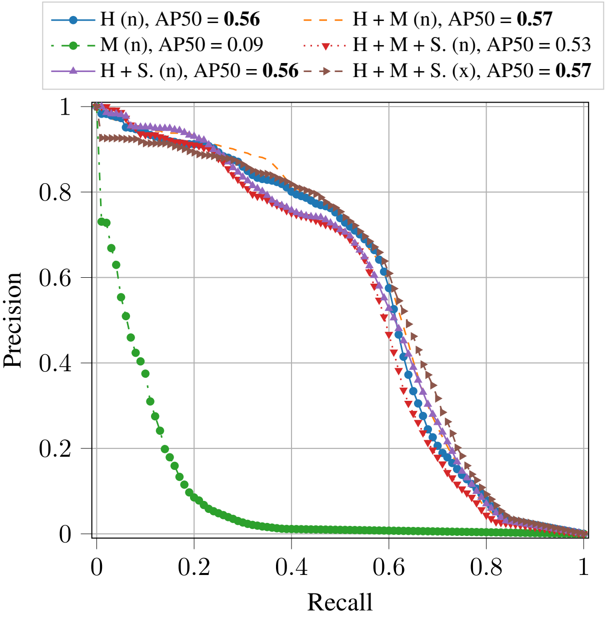


Figure 4. Precision-Recall-Curve for the different training dataset combinations. (n) refers to YoloV5n, and (x) refers to YoloV5x. H = Historical, M = Martian and S = Synthetic.

binations. The results are presented in Fig. 4, in addition to YoloV5n, we present YoloV5x trained on the combined datasets. We chose to add YoloV5x to the comparison, as it performed similar to YoloV5n but had more parameters and might benefit more from the additional data. Overall it is visible that training on the Martian dataset alone does not result in a suitable detector. While it can detect craters in rural areas (i.e. fields), it has a high amount of false positives in urban areas. Training on synthetic data only resulted in a detector that was unable to detect any crater correctly. The curve was thus omitted, while training on any combination of the synthetic and Martian data with our historical dataset resulted in a similar performance. This is likely due to the fact that the Martian domain is too trivial as it only contains well-defined craters in rock and sand. The only challenges with this dataset are overlapping crater or crater contained in a larger crater. Both are rare in historical images and therefore do not contribute much to the overall precision. A likely reason why the synthetic dataset does not contribute to a better performance is because the issue with urban areas is not the false-positive rate on human-made structures but the high irregularity of the craters. But, as we insert cut-out historical craters into XVIEW images we do not increase crater verity.

Lastly, we experimented with different pretraining strategies: no pretraining, COCO pretrained weights, and XVIEW pretrained weights. In Fig. 5 one can see the AP plotted for 20 training epochs. It is visible that after 20 training epochs, all networks achieve similar results, which was also verified with the test set where all networks again performed closely. A similar effect can be seen in the validation loss, which is presented in Fig. 6. It is apparent that after 16 epochs, the loss of all methods is similar. However, it is also visible in both plots that the network pretrained on the COCO dataset requires only about 5 epochs to achieve peak performance. This, considering previous results, means that YoloV5n can be rapidly retrained on and finetuned on a new domain.

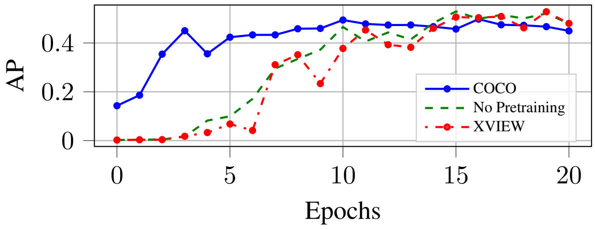


Figure 5. Pretraining evaluation: AP per epoch during training.

5.2. Qualitative Results

We present two detection results of YoloV5n in Fig. 7 and Fig. 1, which show a rural, but snow-covered, area and a field. While YoloV5n can detect all craters flawlessly in Fig. 1, which shows a barren field, it struggles to catch all craters in Fig. 7, which similarly presents a field, however with snow coverage and fresh craters as well as older craters. It was also observed, that in an urban environments YoloV5n is performing even worse than in domain shifted images (i.e snow). This shows that the detector has

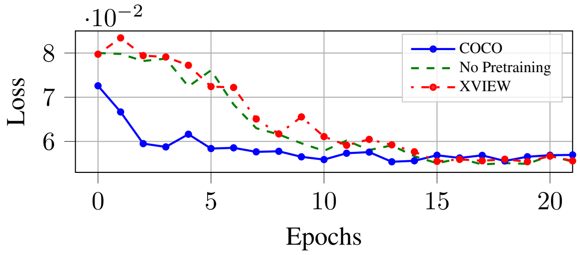


Figure 6. Pretraining evaluation: Bounding box loss per epoch during training.

a very high recall and precision for more straightforward tasks, like the detection of craters in rural areas (Fig. 1). However, it struggles with more complicated tasks, such as the detection in an urban domain. The primary challenge here is the fact that craters are highly irregular due to interference with artificial structures. In contrast to this, in rural areas and especially in barren fields, craters tend to look similar. Additional challenges are due to the low image quality, like low contrast and noise. A further issue is a domain change (i.e., fresh craters and old snow-covered craters), where training images are rare for the new domain. This can be seen in Fig. 7, where the detector can detect 5 out of 13 craters with snow coverage, which are sparsely represented in the dataset.

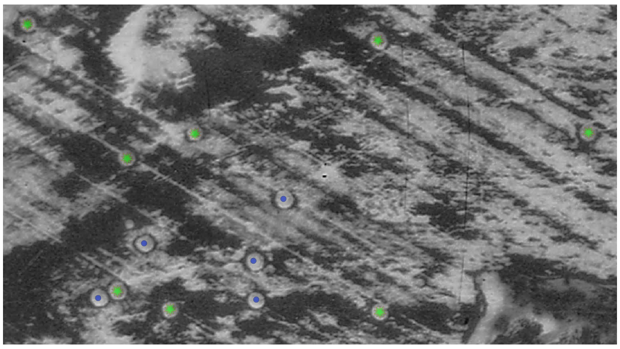


Figure 7. Snowy rural bomb crater detections from [25]. Blue points: Detections, Green Star: Predictions, best viewed in color.

A solution for this problem could be an interactive approach, where an end-user could improve the detections by removing false positives or adding false negatives, and the network then retrains based on the changes. Another possibility could be that the user preemptively marks one or a few craters in the image, which the network then uses as additional domain information during the detection.

6. Conclusion and Future Work

In this paper, 12 State-Of-The-Art detectors were compared on the task of detecting craters on 111 historical aerial images, and the best detector was then compared to another approach from the literature [2]. The detections obtained by these detectors can be used in the predominantly manual process of generating an “explosive ordnance map”, which indicate areas that could be contaminated with UXOs, a significant explosion hazard for construction processes in Europe. We showed that, while the tested detectors were unable to achieve sufficient accuracy to be used fully automatically, the best detector, YoloV5n achieves a precision of 0.6 with a recall of 0.6 in real-world use cases. Furthermore, it only requires 6 seconds to process an image of average size $(10,644 \times 10,042)$, while the approach from the literature requires about 360 seconds for the same image and only has a precision of 0.05 and a recall of 0.01. This combination of accuracy and speed allows for a quick and sufficiently correct preliminary overview over an area, which then can be manually finetuned to an “explosive ordnance map” by an expert. YoloV5n also only requires between 5 - 7 epochs for training when pretrained on COCO, which allows for rapid retraining of the network. Another possibility to exploit YoloV5n would be to use it as a base detector for an interactive learning method or a few-shot learning strategy, where an expert corrects the detections and finetunes the network. This idea will be explored in the future. A further insight is that training on synthetic or Martian data does not significantly improve the detection accuracy of the network, primarily due to the fact that the biggest challenges are irregular craters or unseen crater variants.

In general, we see learning-based approaches in favor of algorithmic approaches like [12]. Our and the results of related work demonstrate the difficulty of crater detection in historical images, which makes the use of semi-automatic approaches inevitable for practical reasons. Learning-based approaches offer the needed flexibility to allow for an on-the-fly adaptation to specific image domains, which will be the direction of our future work.

Acknowledgments

This work was supported by the Austrian Research Promotion Agency (FFG) under project grant 880883. Acquisition of historical aerial imagery: Luftbilddatenbank Dr. Carls GmbH; Sources of historical aerial imagery: National Archives and Records Administration (Washington, D.C.) and Historic Environment Scotland (Edinburgh).

References

[1] A. Bochkovskiy, C-Y. Wang, and H-Y. M. Liao. Yolov4: Optimal speed and accuracy of object detection. arXiv, abs/2004.10934, 2020.

[2] S. Brenner, S. Zambanini, and R. Sablatnig. Detection of bomb craters in wwii aerial images. In *Proceedings of the OAGM Workshop*, pages 94–97, 2018.

[3] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin. Albumentations: Fast and flexible image augmentations. *Information*, 11(2), 2020.

[4] C. Clermont, D.and Kruse, F. Rottensteiner, and C. Heipke. Supervised detection of bomb craters in historical aerial images using convolutional neural networks. *ISPRS - International Society for Photogrammetry and Remote Sensing*, XLII-2/W16:67–74, 2019.

[5] C. Eggert, S. Brehm, A. Winschel, D. Zecha, and R. Lienhart. A closer look: Small object detection in faster r-cnn. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pages 421–426, 2017.

[6] J. Francis, A.and Brown, T. Cameron, R. Crawford Clarke, J. Dodd, R.and Hurdle, M. Neave, J. Nowakowska, V. Patel, A. Puttock, O. Redmond, A. Ruban, D. Ruban, M. Savage, W. Vermeer, A. Whelan, P. Sidiropoulos, and J-P. Muller. A multi-annotator survey of sub-km craters on mars. *Data*, 5(3), 2020.

[7] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun. Yolox: Exceeding yolo series in 2021. arXiv, abs/2107.08430, 2021.

[8] J. Glenn. ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference, 2022.

[9] T. Hu, C. Zhao, Z. Qian, L. He, and M. Ni. Crater obstacle recognition and detection of lunar landing based on yolo v4. In *2021 33rd Chinese Control and Decision Conference (CCDC)*, pages 1748–1752, 2021.

[10] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, Los Alamitos, CA, USA, 2017. IEEE Computer Society.

[11] C. Kruse, F. Rottensteiner, and C. Heipke. Using redundant information from multiple aerial images for the detection of bomb craters based on marked point processes. *ISPRS - International Society for Photogrammetry and Remote Sensing*, V-2-2020:861–870, 2020.

[12] C. Kruse, D. Wittich, F. Rottensteiner, and C. Heipke. Generating impact maps from bomb craters automatically detected in aerial wartime images using marked point processes. *ISPRS - International Society for Photogrammetry and Remote Sensing*, 5:100017, 2022.

[13] D. Lam, R. Kuzma, K. McGee, S. Dooley, M. Laielli, M. Klaric, Y. Bulatov, and B. McCord. xview: Objects in context in overhead imagery. arXiv, abs/1802.07856, 2018.

[14] X. Lu, Q. Li, B. Li, and J. Yan. Mimicdet: Bridging the gap between one-stage and two-stage object detection. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 541–557, Cham, 2020. Springer International Publishing.

[15] M-T. Pham, L. Courtrai, C. Friguet, S. Lefèvre, and A. Baussard. Yolo-fine: One-stage detector of small objects under various backgrounds in remote sensing images. *Remote Sensing*, 12(15):2501, 2020.

[16] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. arXiv, abs/1804.02767, 2018.

[17] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[18] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[19] M. Tan, R. Pang, and Q. V. Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[20] C. Wang, i-H. Yeh, and H-Y. M. Liao. You only learn one representation: Unified network for multiple tasks. arXiv, abs/2105.04206, 2021.

[21] C-Y. Wang, A. Bochkovskiy, and H-Y. M. Liao. Scaled-yolov4: Scaling cross stage partial network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13029–13038, 2021.

[22] C-Y. Wang, A. Bochkovskiy, and H-Y. M. Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv, abs/2207.02696, 2022.

[23] Y. Wu, G. Wan, L. Liu, Y. Jia, Z. Wei, and S. Wang. Fast and accurate crater detection on martian surface using sun et 3+. In *2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC)*, volume 6, pages 683–687, 2022.

[24] Y. Yang and S. Soatto. Fda: Fourier domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4085–4095, 2020.

[25] Sebastian Zambanini. Feature-based groupwise registration of historical aerial images to present-day ortho-photo maps. *Pattern Recognition*, 90:66–77, 2019.