Instream wood detection using YOLOv4 object detection algorithm

Christophe Reymond
UNIVERSITY OF LAUSANNE

Abstract

In-stream wood detection is a fastidious and labour intensive task of river scientists. Machine learning and especially object detection algorithms can help them tremendously in this task. Here we propose an automatic detection of in-stream wood using the YOLOv4 algorithm. Its detecting abilities will be tested on an orthophoto taken from a drone flight at the Sense river, between Bern and Fribourg, in Switzerland.

1. Introduction

Instream wood is a key feature of a river floodplain, it has huge effects of the morphology of the river, the creation of different habitats for the riparian fauna and flora as well as a important role for the water quality (Jones et al., 2014; Lester Boulton, 2008).

Thus, to study most of the processes in a river, it is important to also study the in-stream wood. Field survey to measure and mark pieces of wood are time consuming and difficult, they have to be performed again and again in all sorts of meteorological conditions to study the evolution of the wood supply in the river. On the other hand, flying a drone is easy, it does not require a lot of technical knowledge and the image processing gets faster and faster as computer get more and more powerful. Flying UAVs is already common practice in river studies to produce DEMs and orthophotos which are crucial data. Hence, using drone flight data to automatically detect wood through image detection algorithms could save lots of time and effort.

Thus, the aim of this work is to detect wood in a floodplain using aerial imagery collected through a drone flight. This high quality orthophoto will be explored with the YOLOv4 image detection algorithm. YOLOv4 is a state-of-the-art object detection algorithm that has achieved impressive performance on a variety of benchmarks. Developed by Joseph Redmon and Ali Farhadi, YOLOv4 is an extension of their previous work on the YOLO (You Only Look Once) algorithm, which was known for its high accuracy and speed. YOLOv4 builds on this foundation, using a combination of techniques, such as feature pyramid networks, cross stage partial connections, and multimodal training, to further improve its performance (Bochkovskiy, Wang, Liao, 2020). In experiments, YOLOv4 has demonstrated superior accuracy and speed compared to other popular object detection algorithms, making it a promising choice for a wide range of applications and especially here for wood detection.

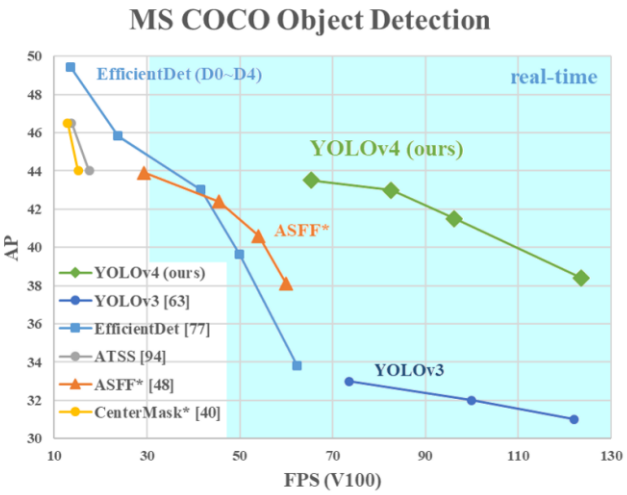

Figure 1. Comparison of the YOLOv4 and other state-of-the-art object detectors. YOLOv4 runs twice faster than EfficientDet with comparable performance. Improves YOLOv3's AP (average precision) and FPS (frames per second) by 10% and 12%, respectively. For the MS COCO dataset (known in the literature). (Bochkovskiy, Wang, Liao, 2020)

2. Dataset

2.1. Orginial dataset

The raw dataset used to test the effectiveness of the YOLOv4 image detection algotrithm is a orthophoto of the Sense floodplain around Plaffeien. This natural braided river is situated between Bern and Fribourg in Switzerland. The entire orthophoto has a length of about 2.0 km and a width of 180 m at its largest point. The orthophoto has a pixel size of 2.3 cm for the whole region, it is more than enough to to have great zoomed in pictures of in-stream wood in the floodplain.

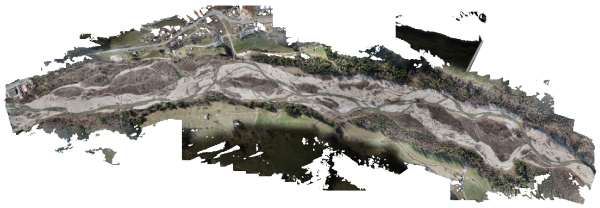

Figure 2. Orthophoto of the Sense river at Plaffeien

2.2. Partitioning of the dataset

The in order to better detect wood in the floodplain, the large orthophoto must be divided into smaller images to train and test the algorithm. The approximate size of these sub-images are around 100 m by 80 m. The images must be zoomed in enough for the algorithm to have a clear view on the wood pieces while having a wide enough view to detect more than a single piece per image. With this kind of division of the floodplain, each image for the training and the validation have around 20 wood pieces in their frame on average.

Figure 3. Example of a sub-image used for training

On the picture presented above, bounding boxes are put around each wood pieces and wood jam to train the YOLOv4 algorithm as a label is associated with each of these pieces.

3. Methodology

YOLOv4 is used through the darknet framework on google Colab, it already has a set of pre-trained weights, they just must be adjusted using the training data to be able to detect the object of interest. These weights are adjusted using the labelled images using the bounding boxes from the orthophoto of the Sense floodplain. The images were labelled using LabelImg, an open-source graphical image annotation tool. This tool helped for the annotation as it could export the annotated images into the right format for YOLO. The format is an image in jpg format with a text file with the same name as the image, containing the relevant information about the bounding boxes.

The large orthophoto is divided into smaller images to have a clear view on the wood pieces. These smaller images are the ones used for training and testing. A total of 28 images were used for training and testing, each of them having around 20 wood pieces in their frame. the split was made at 75%, there are 21 images for training and 7 for testing.

The images used for the training are used for adjusting the pre-trained weights of the YOLOv4 algorithm on the darknet framework, using the COCO dataset. These weights are adjusted with the labeled images during training to better detect the wood pieces in the floodplain of the Sense. The YOLOv4 algorithm in thus trained for the specific purpose of detecting wood in the environment when and where the drone flight was made.

4. Results

The YOLOv4 model trained with the images took a very long time to train : more than 8 hours in total. These types of CNN take a very long time to train and it is a problem when using some king of online platform when computational powers are limited. This is something to keep in mind for further research. It would have been quite difficult to training YOLOv4 on more images, on Colab, due to computational needs.

The average loss during training can be visualized on Figure 4 below. It decreases quite fast until it reaches a plateau around a value of 1 for the average loss. The model shouldn't be trained after this plateau as it could easily be overfitted. For this reason, training was stopped at 4000 iterations, when the average loss value was 1.3754. Figure 4, doesn't display values before 900 iterations as training was done in two sessions and the this chart wasn't saved.

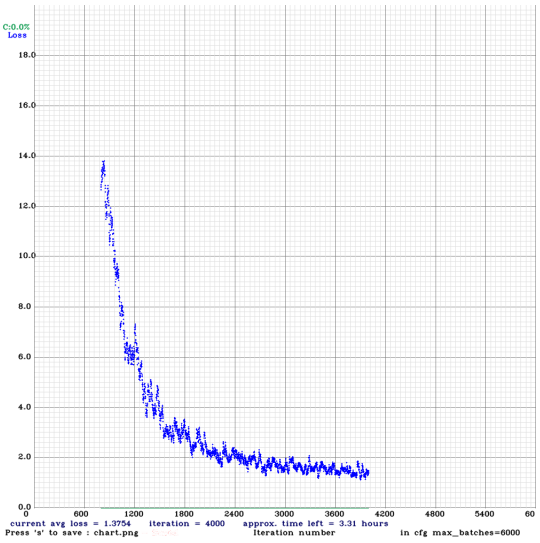

Figure 4. Average loss value during training

Nonetheless, once training has been completed, YOLOv4 detects the wood pieces with incredible speed, it takes less than a second to complete the testing on all 7 images. However, the value for the mean average precision is 29.26 % which isn't a very high value. However, considering the rather small amount of pictures used for training and testing and the difficulty of the task presented for the algorithm, I would still argue that this value is acceptable. The region of study is quite difficult to apprehend as standing wood is mixed with the dead wood pieces we're trying to detect and there are patches of wood with different shapes and colors which further increases the difficulty of the detection for the algorithm.

Table 1. Performance metrics

mAP@0.50	av. loss at the end of training
29.26 %	1.3754

Considering all these performance metrics, it is difficult to evalute the real performance of the trained YOLOv4 algorithm here without displaying a picture where it tries to detect wood pieces in the frame. Figure 5, below displays the attempt of the model to detect wood. To me, it seems like it is doing a decent job, it does not detect all the wood pieces but it manages to highlight the main wood pieces and patches on this rather difficult image. There are places with standing wood and the river, plenty of different rocks and pebbles, it still does a good job at identifying the main features of the wood pieces in the floodplain.

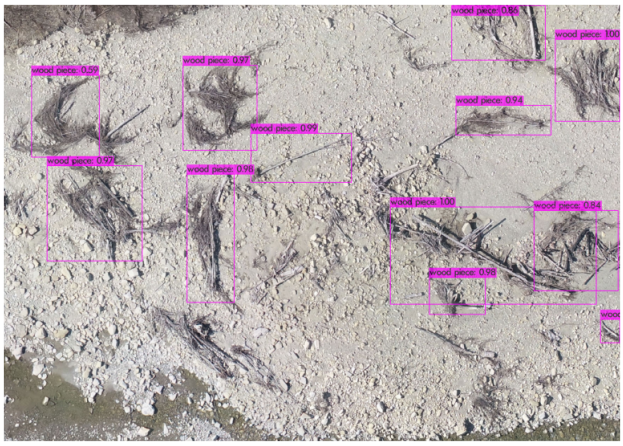

Figure 5. Result of the trained YOLOv4 algorithm on a test image. Predicted in 8.403000 milli-seconds

Overall, I fell like YOLOv4 has been trained sufficiently and works with a acceptable precision for the study site when taking into account the computational requirements and the training time it takes to fully train this object detection algorithm. However, the small amount of pictures used for training would probably make the model only work for the floodplain of the Sense with the environmental conditions when the pictures were taken. The model would be much less precise in a different region with different lighting, or even ground colors.

Improving the overall detecting abilities of the YOLOv4 model would require much more images in different settings but would also require much more time for training. Nonetheless, it would help scientists tremendously in their search for in-stream wood in the floodplain of their study area. Once trained, the speed at which YOLOv4 operates would help gain a ton of time with data available in any river research.

5. Conclusion

YOLOv4 has shown some abilities to detect wood at the study site, its predictive powers could be improved with many more training images but this would even further increase a rather large training time (8+ hours for the training for this report). Nonetheless, it has potential to save a lot of time for many studies as it a quite robust, easy and time efficient algorithm once trained.

Further studies could improve the robustness of the algorithm by training it on many more images and testing it on different floodplains at different times during the year, with different level of vegetation. If trained enough, this YOLOv4 could revolutionize river and in-stream wood research probably.

Link to the code and data

The code used for this report is available on Github with this hyperlink, everything was coded on Google Colab. The data used to train and test the algorithm can be found here

References

Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y. M. (2020, 22 avril). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv. Repéré à http://arxiv.org/abs/2004.10934

Jones, K. K., Anlauf-Dunn, K., Jacobsen, P. S., Strickland, M., Tennant, L., Tippery, S. E. (2014). Effectiveness of In-stream Wood Treatments to Restore Stream Complexity and Winter Rearing Habitat for Juvenile Coho Salmon. Transactions of the American Fisheries Society, 143(2), 334-345. https://doi.org/10.1080/00028487.2013.852623

Lester, R. E., Boulton, A. J. (2008). Rehabilitating Agricultural Streams in Australia with Wood: A Review. Environmental Management, 42(2), 310-326. https://doi.org/10.1007/s00267-008-9151-1

Acknowledgements

I would like to sincerely thank Tom Beucler and all the teaching assistants of the Machine learning class, this class made me do this whole project and helped me develop a real interest toward machine learning. I would like to thanks Janbert Aarnink who helped me at the beginning of the project, to select the algorithm, and to Florent Rouge which did a similar project, we helped eachother during this project. To the anonymous peer-reviewers which helped me improve this project, I thank them for their contributions. Also, this project would have never be completed without the help of The AI Guy and its tutorials.