Instream Large Wood detection trough YOLOV4

Aldo Fornari, Université de Lausanne, FGSE

Abstract

An image detector using YOLOV4 algorithm (using darknet framework) is trained to detect instream large wood in the river. The model is trained with augmented pictures and tested with normal pictures.

1. Introduction

Trough landslide and mass movements, the catchment forest provide large wood (LW) to the river; LW is defined as wood component surpassing 1 m of length and 10 cm of diameter (Ruiz- Villanueva et al., 2018).
The fluvial dynamics of LW are multiple: formation of steps, development of dams, protection of banks and creation of islands (Baillie, 2011). These geomorphic processes reduce flow erosivity by increasing hydraulic resistance (Abernethy et al., 1997) and reducing flow velocity compared to non-wooded stream (Elosegi et al., 2016). Furthermore, the reductions of flow velocity and transport capacity, increase sediment deposition (Wohl and Scott, 2016).
The LW plays also a key role in ecology by increasing diversity of invertebrate communities (Pilotto et al, 2014), increasing diversity and size of fishes (Nagayama and Nagano, 2012; Wats et al., 2018), and providing favorable habitats for the establishments of trout (Wats et al., 2018). In agreement with Pettit and Naiman (2005), instream wood is cornerstone in the resilience and recovery of the fluvial ecosystem in response of a disturbance.
Nevertheless, instream wood during a flood event can causes several damages to infrastructures close to river like bridges, roads, and buildings (Lassettre and Kondolf, 2012; Ruiz-Villanueva et al., 2018). This latter hazardousness caused by river has prompted historical removal of accumulation of wood (Wohl, 2014).

Therefore, it is important to quantify the presence of wood in a stream to understand the river health and hazardousness. Manually quantify wood can result in a difficult task, especially outside office hour. But this task can be operated automatically trough a machine learning algorithm.

The aim of this work is to develop a code able to detect LW in a river using a YOLOV4 algorithm implemented trough a darknet framework. YOLO (*You Only Look Once*) is a fully conventional network (FCN) used in the field of image detection that only looks once at the image compared convolutional neural network (CNN). CNN need to be run many times, is slower and less efficient compared to a FCN as YOLO (Géron, 2019). Darknet is an open source framework that allows to run YOLO image detection written in C and CUDA (Reddy et al., 2021). The code is written on Google Colab with a Jupyter notebook that uses python language. The link to the code is provided at the end of the report.

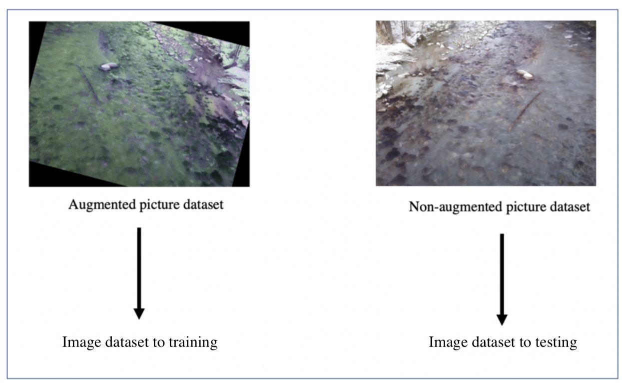

Figure 1. Augmented dataset used for training and "normal" dataset used for testing.

2. Dataset

The river acquisitions were taken with smartphones installed at the edge of the stream or over the bridge. The smartphones took a video during which wood was manually thrown upstream the river to be captured from the camera. This operation was necessary since instream LW was not present during the days of the recordings. The videos were framed to obtain a dataset of thousands of pictures.

The first step to train a YOLO object detector is to manually label images with an annotation tool. This latter operation was done with LabelIMG which allows the user to draw a rectangle over a figure's object and save the

coordinates of the labelling in a .txt file with the same name as the picture.
The datasets of pictures that will be used for testing are augmented to avoid over fitting: the images are shifted, rotated and their lightning are changed to force the model to be more tolerant (Figure 1).

3. Methodology

First, the GPU (Graphics Processing Unit) is enabled to allow graphical computations to run faster. The darknet is cloned in the repository from AlexeyAB's repository and the makefile are changed to GPU and OPENCV enabled. CUDA compiler driver then is verified. The darknet can now be built. The pre-trained weights for YOLOV4 learned from MS COCO dateset which have been trained up to 137 convulutional layers are downloaded.
The file necessary to run the training are uploaded in Google Drive in a .zip file. Colab is connected to Drive and the dataset is unzipped and uploaded to the cloud.

At first it was attempted to train the image detection with two datasets of 2000 augmented pictures and test it for 1 dataset of 2000 pictures. But this attempt failed since the computational time estimated was more than 20 hours and Google Colab kick out an user after 12 hours. In addition to that, Google Colab blocks access of GPU to users that surpassed the 12 hours. Because of these limitations 12 augmented pictures of two different datasets were used for training and 4 images of one dataset were used for testing. This latter operation took 5 hours.

```
IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
mean average precision (mAP@0.50) = 0.808824, or 80.88 %
```

Figure 2. Mean average precision (maP) used to evaluate the object detection model.

4. Results and discussion

The intersection over union (IoU) is defined as the ratio of the area of intersection and area of union of the predicted bounding box and ground truth bounding box (labeled objects in the test dataset)(Redmon et al., 2016).
The precision is the ratio between the true positives and the total number of the objects retrieved (true positives + false positives). The mean average precision (mAP) is the integral over the precision (Hendry and Chen, 2019).

According to Redmon et al. (2016) a correct classification, a true positive, is determined at IoU > 0.5. The mAP at ioU thresold = 50 % is thus the precision of the correct classifications of the model. Figure 2 shows the mean average precision of the model calculated at IoU threshold 0.5 equal to 80.88%, which is a decent result.

Figure 3 displays the results of the image detector run on two different pictures of the same river. In Figure 3a. the trained model detected the log, but it did not predicted bounding box did not covered well all the length of the log, indeed the precision is only at 0.37. The model detected well the length of the log in Figure 3b., in fact the precision is 0.93. But the model did not detected a second smaller log in figure 3b.

The model can detect instream LW, but it is far from being a complete image detector. The main limitation of this project is the few images chosen to train the image detector. It would have been more relevant to have thousand of pictures to train the model. This problem would have probably been solved by choosing a simpler algorithm. Indeed YOLOV4 is used in elaborate detection like videos.

A non-technical limitation of this project is due to the manual thrown of logs into the river. In fact, the presence of the wood is usually related to storms, events when trees and wood are thrown into the stream by wind. The flow during a storm is not clear as in Figure 3, but muddy and full of sediments which can make it difficult for the algorithm to recognize logs. For this reason, it is important to train and test datasets of instream large wood during storm event, since these events are responsible of flood.

5. Conclusion

We trained an a image detector using YOLOV4 algorithm under darknet framework to recognize LW into the river. The model can detect instream large wood, but it is far from being able to recognize wood on real conditions. It would have been interesting to run the model on video to see if the model can track the log as it moves. In the perspective of natural hazards, it would be pertinent to train a model able to measure the length and width of large woods and to identify them (giving an number to each LW detected).

Log Detection Code

The Github link to the script coded to run the training and the log detection using Google Colaboratory.

The files used for the training are available in the following Google drive folder.

Acknowledgements

I cannot express enough thanks to Janbert Aarnink, SNSF Doctoral Student of Institute of Earth Surface Dynamics of Université de Lausanne, to providing me the image datasets with labels (same for the augmented datasets).

My completion of this project could not have been accomplished without the tutorials of The AI Guy to run darknet for a custom object detection using YOLOV4.

References

[1]Abernethy, B., Rutherfurd, I. D. (1998). Where along a river's length will vegetation most effectively stabilise stream banks? Geomorphology, 23(1), 55–75. https://doi.org/10.1016/S0169-555X(97)00089-5

Elosegi, A., Díez, J. R., Flores, L., Molinero, J. (2017). Pools, channel form, and sediment storage in wood-restored streams: Potential effects on downstream reservoirs. Geomorphology, 279, 165–175. https://doi.org/10.1016/j.geomorph.2016.01.007

Géron, Aurélien (2018). Hands-on Machine Learning with Scikit-Learn, Keras Tensorflow (2nd ed). O'Reilly

Hendry, Chen, R.-C. (2019). Automatic License Plate Recognition via sliding-window darknet-YOLO deep learning. Image and Vision Computing, 87, 47–56. https://doi.org/10.1016/j.imavis.2019.04.007

Lassettre, N. S., Kondolf, G. M. (2012). Large woody debris in urban channels River Research and Applications, 28(9), 1477–1487. https://doi.org/10.1002/rra.1538

Nagayama, S., Nakamura, F., Kawaguchi, Y., Nakano, D. (2012). Effects of configuration of instream wood on autumn and winter habitat use by fish in a large remeandering reach. Hydrobiologia, 680(1), 159–170. https://doi.org/10.1007/s10750-011-0913-z

Pettit, N. E., Naiman, R. J. (2005). Flood-deposited wood debris and its contribution to heterogeneity and regeneration in a semi-arid riparian landscape. Oecologia, 145(3), 434–444. https://doi.org/10.1007/s00442-005-0143-z

Pilotto, F., Bertoncin, A., Harvey, G. L., Wharton, G., Pusch, M. T. (2014). Diversification of stream invertebrate communities by large wood. Freshwater Biology, 59(12), 2571–2583. https://doi.org/10.1111/fwb.12454

Reddy, B. K., Bano, S., Reddy, G. G., Kommineni, R., Reddy, P. Y. (2021). Convolutional Network based Animal Recognition using YOLO and Darknet. 2021 6th International Conference on Inventive Computation Technologies (ICICT), 1198–1203. https://doi.org/10.1109/ICICT50816.2021.9358620

Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779–788. https://doi.org/10.1109/CVPR.2016.91

Ruiz-Villanueva, V., Badoux, A., Rickenmann, D., Böckli, M., Schläfli, S., Steeb, N., Stoffel, M., Rickli, C. (2018). Impacts of a large flood along a mountain river basin: The importance of channel widening and estimating the large wood budget in the upper Emme River (Switzerland). Earth Surface Dynamics, 6(4), 1115–1137. https://doi.org/10.5194/esurf-6-1115-2018

Baillie, B. The physical and biological function of wood in New Zealand's forested stream ecosystems. (2011). 155.

Watz, J., Calles, O., Carlsson, N., Collin, T., Huusko, A., Johnsson, J., Nilsson, P. A., Norrgård, J., Nyqvist, D. (2019). Wood addition in the hatchery and river environments affects post-release performance of overwintering brown trout. Freshwater Biology, 64(1), 71–80. https://doi.org/10.1111/fwb.13195

Wohl, E. (2014). A legacy of absence: Wood removal in US rivers. Progress in Physical Geography: Earth and Environment, 38(5), 637–663. https://doi.org/10.1177/0309133314548091

Wohl, E., Scott, D. N. (2017). Wood and sediment storage and dynamics in river corridors: Wood and Sediment Dynamics in River Corridors. Earth Surface Processes and Landforms, 42(1), 5–23. https://doi.org/10.1002/esp.3909

(a)

(b)

Figure 3. Results of running the object detection with a dataset that was not used for training and testing with tresh flag around the object indicating the accuray of the detection(a) Custom detector for one log (b) Custom detector for two logs.