France

Université Paris Cité
LIPADE

Team FengHu
TIGER Challenge

METHOD DESCRIPTION

Submitted to

TIGER challenge committee

Method description for the lymphocytes and plasma cells detection task

By

Amine MARZOUKI

UPARISCITE1 ALGORITHM

FengHu Team members:

Nicolas LOMENIE	Associate Professor
Camille KURTZ	Associate Professor
Zhuxian GUO	PhD Student
Qinghe ZENG	PhD Student
Amine MARZOUKI	Master's student

Contact: firstName.lastName@u-paris.fr

June 2022

Table of Contents

List of Figures

General introduction

Most of the optimizations that is done in a deep learning system focuses on the model, whether it was the model's architecture in general or the way its weights are optimized. In our approach, we decided to go with a *data-centric* mindset, focusing more on how can we optimize the dataset while keeping the model fixed such that it yields the best performance.

Many sampling and data augmentations techniques were tried, though we tried to keep it as simple as possible, we also share some of the things we tried that didn't work well.

Because we followed a *data-driven* approach, we had to choose the model that will be fixed somehow, the literature exposes many different flavors, variants and architectures for the object detection [1], but since we want to get the best performance with the simplest way possible we chose the most famous one which is YOLO [2]. While we found a tweaked version of this model specifically designed for lymphocytes detection [3], we decided to use the original vanilla version, and given the amount of the data we had, we chose to stick with the small variant.

1. Team

The **FengHu team** consists of 5 members coming from the **Laboratory of Informatics Paris Descartes (LIPADE)** that belongs to the Faculty of Mathematics and Computer Science in the Université Paris Cité.

1.1 Team members

- **Nicolas LOMENIE** An associate professor at Université Paris Cité who teachs computer science and mathematics applied to artificial intelligence and computer vision. He is responsible for an apprenticeship training at the Master level Vision and Intelligent Machine, and co-leader of a double degree in this field with the University of Tunis El Manar (National School of Engineering of Tunis).
- **Camille KURTZ** An associate professor of computer science (maître de conférences, HDR) at Université Paris Cité, in the IUT de Paris - Rives de Seine, and researcher at the LIPADE laboratory, SIP team. He conducts scientific research in AI on pattern recognition, image analysis and computer vision. More precisely, he is interested in the underlying problems related to image representations, with visual, semantic and perceptual aspects.
- **Qinghe ZENG** a PhD student working at LIPADE, you can find more about her **here**.
- **Zhuxian GUO** a PhD student working at LIPADE, you can find more about him **here**.
- **Amine MARZOUKI** a research engineer apprentice at LIPADE, you can find more about him **here**.

1.2 Contact

You can reach out to the professors at: firstname.LASTNAME@u-paris.fr

For the students, you can reach out at: firstname.LASTNAME@etu.u-paris.fr

2. Data

Concerning the dataset, we only used the ROI level annotations under the WSIrois dataset, specifically the tissue-cells folder where there was 1879 images of different sizes and one annotation json file in COCO format.

Since we decided to use YOLO, we had to convert the annotations to YOLO format, for that we have used the roboflow API in order to do so, we also used this tool to do our pre-processing step.

Obviously the first step to do is to split these images into smaller patches, since a simple resize would not be a good solution. After inspecting the sizes of these images, we noticed that we have a discontinuous function in terms of sizes, since there was a big gap between the two subsets as you can see in the images sizes histogram below Fig1.

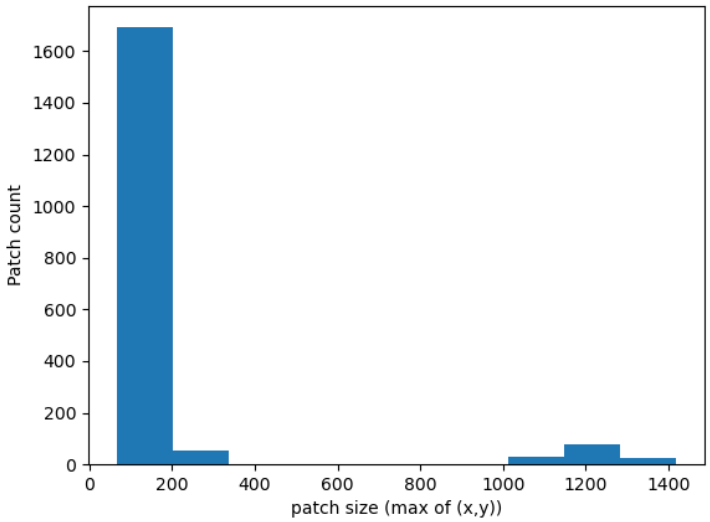

Figure 1. *Histogram of the images sizes*

2.1 Creating the first dataset

Although we wanted to try different patches sizes, we chose to stick to 256. Doing a basic resize strategy to all of these images will result in a very poor patches, so we chose to develop two different strategies for each subset.

A tiling strategy was selected for the first subset which consists of the 135 images who have a size of over 900x900. Additionally, since we wanted to have a patch size that is as close as possible to 256x256, we select either a 4x4 or 5x5 tiling approach depending on the image size. Regarding the other subset, which represents the majority here, a simple stretching was done in order to upsample the image so it can fit the 256 size. We summarize this process in the following figure 2.

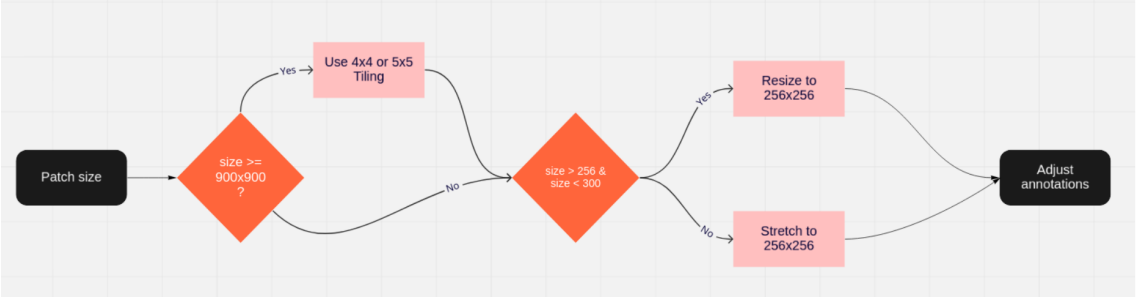

Figure 2. *Preprocessing pipeline*

In order to feed the images to the model the same way they are displayed like they they are stored on disk, an Auto-Orient preprocessing step was applied for both subsets.

Finally, the annotations were adjusted to match to the newly resulted patches and were converted to YOLO format, it is important to mention that all of these steps was done by using Roboflow[1].

2.2 Creating the second dataset

In this version, only the images who had at least one annotation were selected, then for each image it was splitted into multiple patches of 256x256 while discarding any incompleted lymphocytes due to this operation.

To tackle the problem of the smaller patches or the images who are smaller than the required size, different padding strategies were explored before we settled by gray padding.

2.3 Final dataset

After merging the two datasets into one, we sampled the data at an image level, ensuring that there will be no leak when splitting the dataset into train/validation sets. Such sampling requires that all the patches of an image will always be present in the same set.

[1] You can find the webapp at `https://app.roboflow.com/`

The resulting dataset had 8799 patches in total which then were splitted into 80/20 for the the train/validation sets respectively.

Hereby an example of the resulting patches with their annotations from the validation set of the constructed dataset (Fig3).

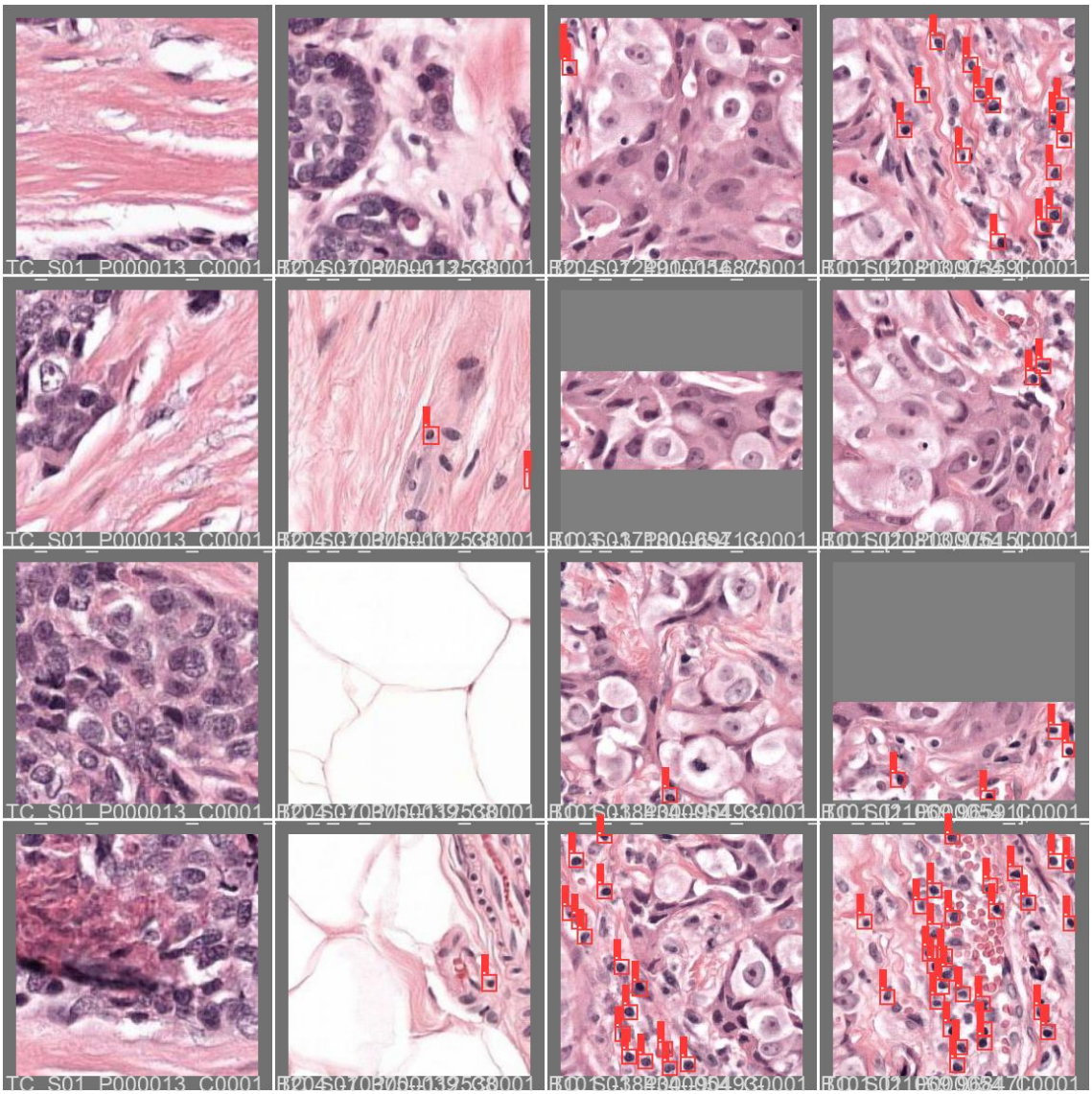

Figure 3. *Example of some patches in the validation set*

3. Method

As the core of our method is mainly described in section 2, here we present the model that we have fixed to use and a small post-processing step.

We used YOLOv5 small [4] pretrained on COCO dataset thanks to its ultralytics pytorch implementation. It was then finetuned for 600 epochs with a batch size of 32 using the default parameters. We also let the default on the fly data augmentation pipeline as well that comes from ultralytics.

The training took over 9 hours using NVIDIA GeForce RTX 3050 Ti Laptop GPU, and we have reached 75.77% in precision and 73.03% in recall. We provide the precision recall curve on the validation set in the following figure 4)

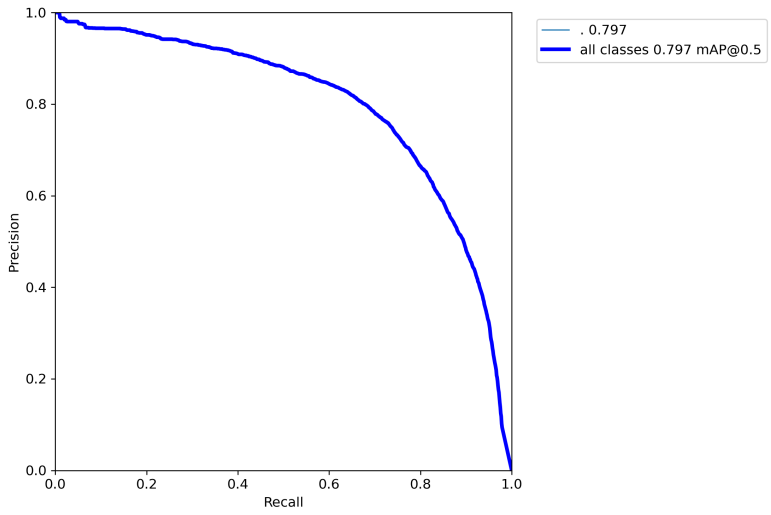

Figure 4. *Precision recall curve on the validation set*

Since in the evaluation only the center of the boxes are taken, we did two post-processing steps:

1. We filtered the inference such that all the predicted boxes will a width and height between 8 and 20 pixels, otherwise we drop the bounding box.
2. We adjust the center of the box in case of a partial bounding box.

On the unseen test set of the platform, we achieved an FROC score of 0.5731[1]

[1]More details on the submission can be found at `https://tiger.grand-challenge.org/evaluation/712cc901-abaa-4080-8ae1-8e0aa977110f/`

Conclusion

We present a very simple approach that placed us in the 6th position of the leaderboard, we are the second team wise regarding the lymphocytes detection task.

Our approach is a data-centric one and we believe that it is an approach that is not well explored until now. We have used the famous YOLOv5 model with its default hyperparameters with a very regular laptop GPU. We proved that we can reach a very good performance in a simple way, where we placed 6th algorithms wise and 2nd team wise in the L1 leaderboard with an **FROC** score of **0.5731**.

References

[1] Zhong-Qiu Zhao et al. “Object Detection With Deep Learning: A Review”. In: *IEEE Transactions on Neural Networks and Learning Systems* 30.11 (2019), pp. 3212–3232.

[2] Licheng Jiao et al. “A Survey of Deep Learning-Based Object Detection”. In: *IEEE Access* 7 (2019), pp. 128837–128868.

[3] Mart van Rijthoven et al. “You Only Look on Lymphocytes Once”. In: *Medical Imaging with Deep Learning*. 2018.

[4] G. Jocher et al. “ultralytics/yolov5: v6. 1-TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference”. In: *Zenodo* 22 (2022).