Project Proposal

Inspiring Excellence

Real Time Triggered Object Detection For Alzheimer Patient Using YOLObile

Prepeared By:

Nibraz Khan - 18201057

Minhajul Abedin - 18301224

Date of Submission: 30 November 2021

Department of Computer Science and Engineering

BRAC University

Research paper summary

[1] Cai, Y., Li, H., Yuan, G., Niu, W., Li, Y., Tang, X., ... & Wang, Y. (2021, May). YOLObile: Real-Time Object Detection on Mobile Devices via Compression-Compilation Co-Design. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 35, No. 2, pp. 955-963).

Cai and et al. mention about current two state of the art object detections works. One is accuracy oriented which is based on large model and because of that it provides high latency. On the other hand, speed-oriented works are based on lightweight model. It has the drawback in terms of accuracy though this type of model has more speed than the model that are used in accuracy-oriented works. However, the most promising DNN (Deep Neural Network) based object detection algorithms proposed so far are two-stage detectors and one-stage detectors. While two stage detectors classify and regress after extracting region of interest (ROI), one stage detector does so without removing ROI. In contrast to two-stage detectors, one-stage detectors maintain a balance trade off between accuracy and speed. Although, one stage detectors, like, YOLO and SSD can achieve excellent average precision for high performing desktop GPUs but failed on mobile devices due to the demand of heavy processing Some lightweight model, like, YOLO-LITE and SSDLite have introduced for mobile devices but suffer from poor precision. Some model compression approaches, including weight pruning, can solve this issue. Still, there are certain shortcomings in terms of parallelism and precision. While pattern-based pruning addresses this, it is limited to the 3×3 convolution layers. Moreover, DNN inference acceleration is insufficient to fulfill the low latency needs of object detection. Therefore, a better pruning scheme and computation method must need to overcome this lacking.

To address this issue, the authors proposed a YOLObile framework that is the combination of a noble block-punched pruning and GPU-CPU collaborative computation scheme. Also, mobile unfriendly operation replacement, re-weighted pruning algorithm and complier assisted acceleration method are the part of this framework. To explain, the block-punched pruning approached has been taken so that it can be applied in

any CONV layer with any kernel size along with fully-connected layers. Instead of pruning entire channel like structured pruning or pruning randomly like unstructured pruning, block-punched pruning followed by patter-based pruning with slide changes. In block-punched pruning, DNN weights of a certain layer will be divided into a number of equal size block and then a group of weights will be pruned from each block at the same location. Besides that, to improve the computational efficiency of DNNs on mobile device, their proposed YOLObile framework adopts a GPU-CPU collaborative computation scheme. In this scheme, GPU will be selected for computing most time-consuming branch as it outperforms CPU in terms of high-parallelism and the remaining issue is determining whether the other branches compute simultaneously on the CPU or sequentially on the GPU. Finally, their proposed framework trained and test on MS COCO dataset and successfully achieved 14 × compression rate (in weights) of YOLOv4 with 49.0 mAP along with 5× speed of YOLOv4 by getting 19.1 frames per second (FPS) inference speed on an off the shelf Samsung Galaxy S20. To sum up, the combination of their two new schemes which are block-punched pruning and GPU-CPU collaborative computation outperformed the previous lightweight models such as YOLOv4, YOLOv3-tiny, SSD.

Despite the fact that their proposed YOLObile framework has outperformed previous proposed model, yet the one limitation that became apparent is that this model will only perform well for the mobile devices with GPU in terms of speed. In contrary, the mobile devices that have only CPU will be less efficient as the collaborative computation scheme cannot be applied there. As a result, non-GPU mobile devices might produce high latency.

We came across this article when looking for Real Time Object Detection on mobile or embedded devices projects, and we found it to be partially comparable to our research as we also need to do the object detection on mobile or embedded devices so that patient can easily carry it. Moreover, the proposed YOLObile framework in this research paper has successfully achieved an outstanding result and outperformed all the previous object detection model that are based on DNNs and mobile devices. Most of the information about the object detection models are well explained in this paper along

with their accuracy compared to the proposed framework. This paper also referenced in some recent works, like, Hand Detection [2] and ODF on Resource Constricted Hardware device [3]. Thus, by examining all of the aspects it was more convenient for us to choose this paper.

Project Proposal

The main goal of this project is to check whether it is possible to make a real time object detection model with the best available DNNs model which can continuously provide fast and most accurate result after trained on a group of triggered objects of Alzheimer patient. After this experiment we can come to a conclusion about the acceptance of using YOLObile framework for this kind of purposes where we need both high accuracy and low latency at the same time. Moreover, the conclusion will guide us for better research in future projects for mobile devices. Apart from that, in future we can test this framework on embedded device such as smart watches and smart glasses and verify the acceptance by examining the resultant accuracy and speed. Our chosen paper has proposed a framework that successfully achieved a good result while outperforming all of the previous DNNs model specially for mobile device. It can maintain both accuracy and speed at the same times where many models have failed to do so [1]. From the outcome of this paper, we can come to the conclusion to work with YOLObile as it has achieved better result than available DNNs model in terms of both speed and accuracy.

In our project, we are going to shortlist some triggered object, make data sets based on the shortlist and real time object detection task of computer vision. To explain, Alzheimer patient have some triggered object in which they get triggered and do some abnormal behavior. Although, the trigger object can be varied from person to person but we can initially shortlist some triggered object for the project. After that, we will manually create the data set for the models with some downloaded videos from internet and making some videos on our own. The model will be trained on this data set with

some prepossessing to detect the triggered objects. Then we will test the framework on some videos and live stream. The process of training and testing will repeat until an acceptable outcome will come. After came up with an acceptable result we will go for deployment where it will search continuously for triggered object and give unsafe signal to the user.

For our dataset, we will download the videos based on our triggered objects from internet and manually make some videos on our own as well. Also, we will do some prepossessing, like, adding noise.

We are planning to use ML-based approach using YOLObile framework as it has achieved satisfactory result over rest of the available DNNs models. As this paper provide the source code of the implementation on their public GitHub repository (https://github.com/nightsnack/YOLObile), we will use it as a baseline of our project. For evaluating the accuracy, we might use the mean Average Precision(mAP) and precision x recall curve. Also, Average Delay (AD) might be used to calculate delay more accurately.

Research paper summary

Despite the fact that we aim to work together, each of us has a unique set of training and experience. Therefore, we have divided the work based on our own skills like below:

1. Nibraz Khan – Researching paper, collecting data set, coding (visualize)
2. Minhajul Abedin - Researching algorithm, pre-processing, coding (train-test)

Lastly, we have designed a flow chart diagram based on our pre-planning which may alter as the project proceeds.

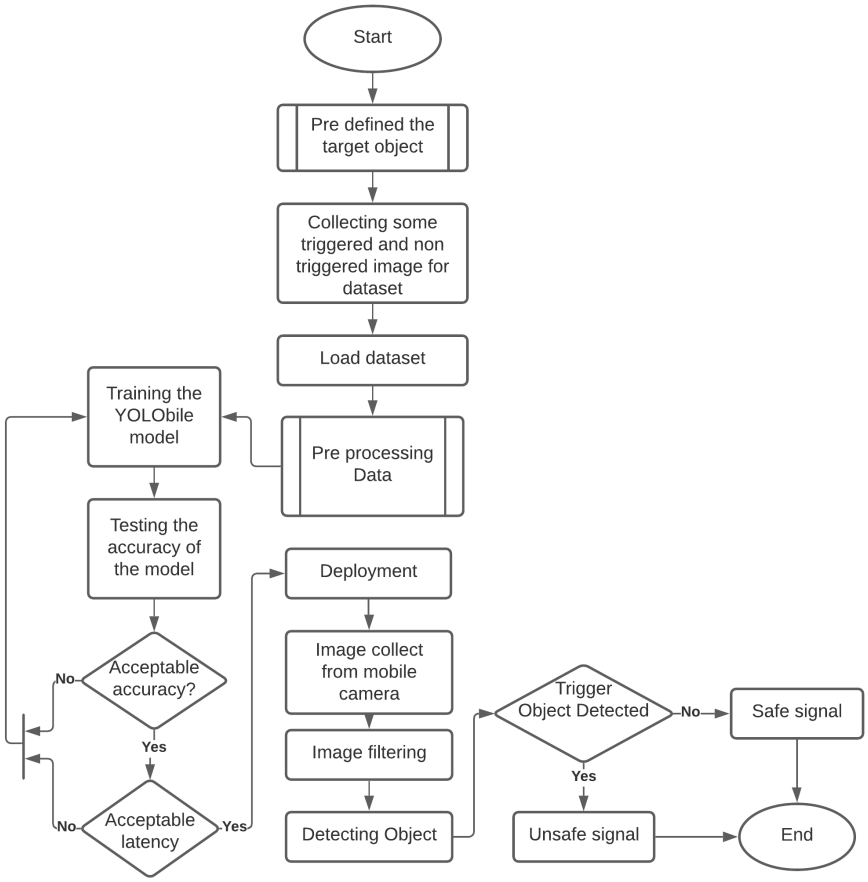

Figure 1: Work flow diagram

References

[1] Yuxuan Cai et al. "YOLObile: Real-Time Object Detection on Mobile Devices via Compression-Compilation Co-Design". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 2. 2021, pp. 955–963.

[2] Mingshuo Liu et al. "An Efficient Real-Time Object Detection Framework on Resource-Constricted Hardware Devices via Software and Hardware Co-design". In: *2021 IEEE 32nd International Conference on Application-specific Systems, Architectures and Processors (ASAP)*. IEEE. 2021, pp. 77–84.

[3] Linjuan Ma and Fuquan Zhang. "A Deep End-to-end Hand Detection Application On Mobile Device Based On Web Of Things". In: *Companion Proceedings of the Web Conference 2021*. 2021, pp. 63–67.