Custom YOLOv3 Model for Human Detection

Miiyu Fujita
McGill University, MAIS202

Problem Statement

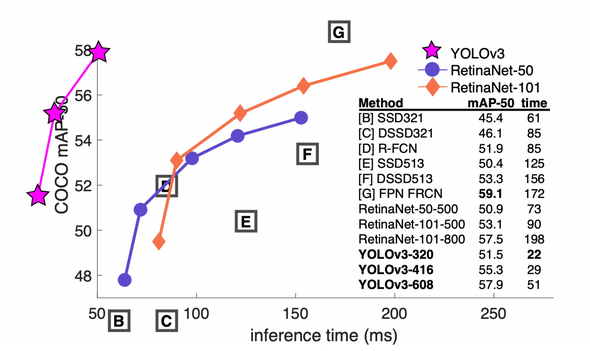
- The Custom Human Detection model's main goal is to identify human intruders in home environments, to ensure the safety of the house occupants.
- The model is intended to be used on an environment in which no occupants are expected to be present. Thus, the model aims to detect any human subjects within an environment.

Introduction

- Upon identifying the problem to solve, I was tasked with completing the following steps: choosing an adequate model, selecting a dataset, training the model, and finally testing the model's performance.

Choosing the Model

- Considering the nature of the problem (human intruder detection), it was clear that the model had to have a high speed of detection coupled with satisfactory accuracy of predictions.
- Due to this constraint, the **YOLOv3** (You Only Look Once) model was attractive, in the sense that it provided accurate predictions despite high detection speeds:
- The following is a plot of the accuracy vs speed of the YOLOv3 model (purple star) compared to other models, demonstrating that in terms of speed of detection, the YOLOv3 model is an adequate choice [1].



Method	mAP-50	time
[B] SSD321	45.4	61
[C] DSSD321	46.1	85
[D] R-FCN	51.9	85
[E] SSD513	50.4	125
[F] DSSD513	53.3	156
[G] FPN FRCN	**59.1**	172
RetinaNet-50-500	50.9	73
RetinaNet-101-500	53.1	90
RetinaNet-101-800	57.5	198
YOLOv3-320	51.5	**22**
YOLOv3-416	55.3	29
YOLOv3-608	57.9	51

Selecting a Dataset

- The initial dataset of choice was the COCO dataset, as it was a large-scale dataset that captured everyday scenery, which I believed would be advantageous for detection in home environments [2].
- However, having chosen to work with the YOLOv3 model, I opted instead to use **Google's Open Images Dataset V6**, as this dataset was not only also large in scale, but was much more convenient when it came to the data pre-processing stage. To train a custom YOLOv3 model, one must use data that is of a specific format. As such, Google's Open Images Dataset V6 was an extremely convenient dataset to use, considering all images in the dataset came with corresponding bounding box information.
- To download the data, I made use of the **OIDv4 toolkit**, as it allowed me to download both the images for the dataset as well as annotation files for each. I then proceeded to use an additional python script written by theAIGuy to convert the .txt annotation files to the proper format for training [3] .

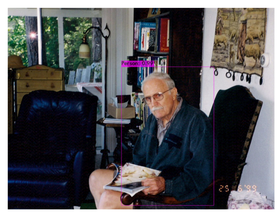
The Model: Preliminary Results

- In my first attempt at training the model, I worked with two classes: "Person" and "Furniture", as I believed this would encapsulate the home environment in which I intended the detections to take place. 400 instances of each class were used, and I chose to go through 4000 iterations during the training process. [4]
- Using these two classes proved unfruitful, as the model was extremely overfitted, with an mAP of 99.61% on the training set and very low confidence in predictions when tested on other images.
- On the left is a test run on a sample image from the test set. The model fails to detect the two women separately.

The Model: Final Results

- My choice to use both "Person" and "Furniture" classes in the preliminary training stage proved ineffective and failed to optimize the confidence of detections. To account for this, I aimed to train the model on a dataset solely consisting of "Person" subjects, and to increase the number of instances used from 400 to 800, while keeping the number of iterations the same. [4]
- By doing so, I prevented blatant overfitting, and increased the mAP on my test set to 27.85%, obtaining an F1 score of 0.40. Detection times on several test images averaged around 40 milliseconds, and the total detection time for 160 test images was 5 seconds.
- Testing this newly trained model on the test image used in the preliminary stage, there is an observable improvement. There is both an increase in the confidence of predictions as well as number of predicted subjects.

Test Runs:



- Person: 0.99

- Person: 1.00
- Person: 0.99

- Person: 0.99

Conclusion

- My custom YOLOv3 model for Human Detection is now able to detect human subjects in home environments (as shown in the test runs). When presented with an image, it can readily detect human subjects quickly with relative confidence.
- In terms of future projects, I would be interested in implementing this human detection model into a larger project that would be able to alert home occupants when a human subject is identified. I believe also that further investigation into how this model could be applied to video footage (security camera footage) would also prove to be fruitful.

References

[1] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. arXiv, 2018.

[2] Common Objects in Context. What is COCO?. Retrieved from https://cocodataset.org/#home

[3] theAIGuy (2020). *Creating a YOLOv3 Custom Dataset | Quick and Easy | 9,000,000+ Images.* Available at: https://www.youtube.com/watch?v=_4A9inxGqRM&ab_channel=TheAIGuy (Accessed: 17 February 2021).

[4] theAIGuy (2020). *YOLOv3 in the CLOUD : Install and Train Custom Object Detector (FREE GPU).* Available at: https://www.youtube.com/watch?v=10joRJt39Ns&ab_channel=TheAIGuy (Accessed: 17 February 2021).

Contact

Miiyu Fujita
McGill University
E-mail: miiyu.fujita@mail.mcgill.ca
LinkedIn: https://www.linkedin.com/in/miiyu-fujita