YOLO-LITE:
A Real-Time Object Detection Web Implementation

Rachel Huang
B.S. Computer Engineering | Georgia Institute of Technology

Live Demo: https://reu2018dl.github.io

INTRODUCTION

Object detection is a field of computer vision that deals with locating and classifying real-world objects. Most current implementations are able to achieve real-time object detection with a Graphics Processing Unit (GPU). However, many devices such as cellphones or portable laptops lack such support. In order to increase accessibility of object detection, a lighter model, YOLO-LITE, was developed to run without a GPU.

BACKGROUND

YOLO: (You-Only-Look-Once)

- The architecture of YOLO-LITE was based on an already established model called YOLO which detects objects and predicts bounding boxes simultaneously.
- **Process**:
  - Split input image into an **S** x **S** grid.
  - In each grid, **B** bounding boxes are predicted.
  - Each bounding box is given a confidence score, the likelihood of an object being in each box. The equation is given below:

$$Confidence = Pr(Object) * IOU_{pred}^{truth}$$

  with IOU being intersection over union.
  - At the same time, each grid cell predicts C class probability. Below shows the class specific probability for each grid cell:

$$Pr(Class_i|Object) * Pr(Object) * IOU_{pred}^{truth}$$
$$= Pr(CLass_i) * IOU_{pred}^{truth}$$

- Below are the performances of the different versions of YOLO:

Model	Layers	FLOPS(B)	FPS	mAP
YOLOv1	26	Not reported	45	63.4 (VOC)
YOLOv1-tiny	9	Not reported	155	52.7 (VOC)
YOLOv2	32	62.94	40	48.1 (COCO)
YOLOv2-tiny	16	5.41	244	23.7 (COCO)

Datasets:

Two object recognition datasets were used to train YOLO-LITE:

- **VOC 2007 + 2012**
  - Training set combines dataset from both 2007 and 2012
  - Contains about 5,011 training images with 20 classes.
- **COCO 2014**
  - Training set contains about 40,775 images with 80 classes.

EXPERIMENTS

Goals:

- The goal of YOLO-LITE was to obtain a model with an mAP 30% and 10 FPS and implement onto a website.
- Several variables were tested when training YOLO-LITE:
  - Batch normalization
  - Input image size: 416x416, 300x300, 224x224, 112x112
  - Activation function: leaky ReLu, Linear
  - Number of layers
  - Number of filters

VOC Training:

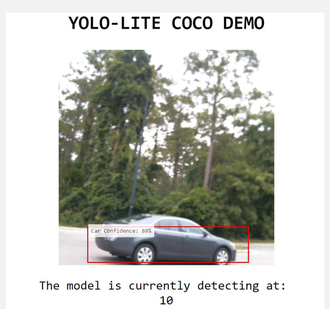
Model	Variable Tested	Layers	FLOPS(B)	Loss	Epochs
Tiny-YOLOv2	Retraining	9	6.97	1.26	35,200
Tiny-YOLOv2	No batch	9	6.97	.85	40,800
Trial 1	Less layers	7	28.64	1.91	20,000
Trial 2	224x224 image size	9	1.551	1.37	40,200
Trial 2	No batch	9	1.551	1.56	44,600
Trial 3	Less layers	7	.482	1.64	166,659
Trial 3	**No batch**	**7**	**.482**	**1.64**	**116,600**
Trial 4	Layer order	8	1.025	1.93	98,700
Trial 5	Layer order	7	.426	2.4	42,600
Trial 6	Activation function	7	.482	1.91	159,000
Trial 7	More filters	7	.618	2.3	101,100
Trial 8	+1 Layer	8	.49	1.3	174,700
Trial 9	300x300 image size	7	.846	1.5	211,600
Trial 10	416x416 image size	7	1.661	1.55	87,700
Trial 11	112x112 image size	7	.118	1.35	478,900
Trial 12	Reduced FLOPS	8	.71	1.74	72,500
Trail 12	No batch	8	.71	.92	168,000
Trial 13	Reduced FLOPS	8	1.083	1.42	59,300
Trial 13	No batch	8	1.083	.77	99,000

COCO Training:

- The best trial from VOC, trial 3-no batch, was taken and retrained on COCO.
- The resulting model ran at about 21 FPS with 12.26% mAP.

Web Implementation:

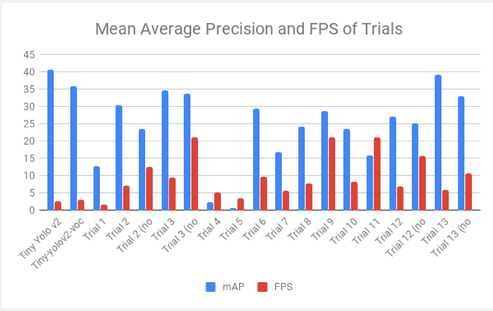
- Once the best VOC and COCO model was obtained through training, the model was then converted to JavaScript and implemented onto a website in order to run without a GPU. A snapshot of the COCO model is pictured to the right:



RESULTS

The mAP and FPS from the **VOC** trials are shown below:

Model	mAP	FPS
Tiny-YOLOv2	40.48%	2.4
Tiny-YOLOv2	35.830%	3
Trial 1	12.64%	1.56
Trial 2	30.24%	6.94
Trial 2	23.49%	12.5
Trial 3	34.59%	9.5
Trial 3	**33.57%**	**21**
Trial 4	2.35%	5.2
Trial 5	0.55%	3.5
Trial 6	29.33%	9.7
Trial 7	16.84%	5.7
Trial 8	24.22%	7.8
Trial 9	28.64%	21
Trial 10	23.44%	8.2
Trial 11	15.91%	21
Trial 12	26.9%	6.9
Trail 12	25.16%	15.6
Trial 13	39.04%	5.8
Trial 13	33.03%	10.5



Best models from both **COCO** and **VOC** are shown below:

Model	Best mAP	Best FPS
VOC	33.57%	21
COCO	12.26%	21

CONCLUSION

YOLO-LITE was able to surpass the initial goal with 21 FPS. While the model achieved 33.57% mAP with **VOC**, only 12.26% mAP was achieved with **COCO**. This decrease in mAP may be due to the larger training set and number of classes. Future work may include increasing mAP through the following methods:

- Use of Region-based Convolution Neural Networks in combination with YOLO model.
- Pretraining on ImageNet or CIFAR-10 dataset.

REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 779–788, 2016.

[2] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. arXiv preprint, 2017.

ACKNOWLEDGEMENTS

This research was done through the National Science Foundation under DMS Grant Number 1659288. I would like give a special thanks to Jonathan Pedoeem, Dr. Cuixian Chen, and Dr. Yishi Wang for their support.