YOLO-LITE

RACHEL HUANG

REU UNCW 2018

JULY 24, 2018

Collaborators

Special thanks to all my collaborators on this project:

Dr. Cuixian Chen

Dr. Yishi Wang

Jonathan Pedoeem

Summerlin Thompson

Outline

I. Introduction
   I. Object Detection
   II. Convolutional Neural Networks
   III. YOLO
II. YOLO-LITE
   I. Architecture
   II. Results
III. Web Implementation
IV. Conclusion

Introduction

YOLO-LITE

Web Implementation

Conclusion

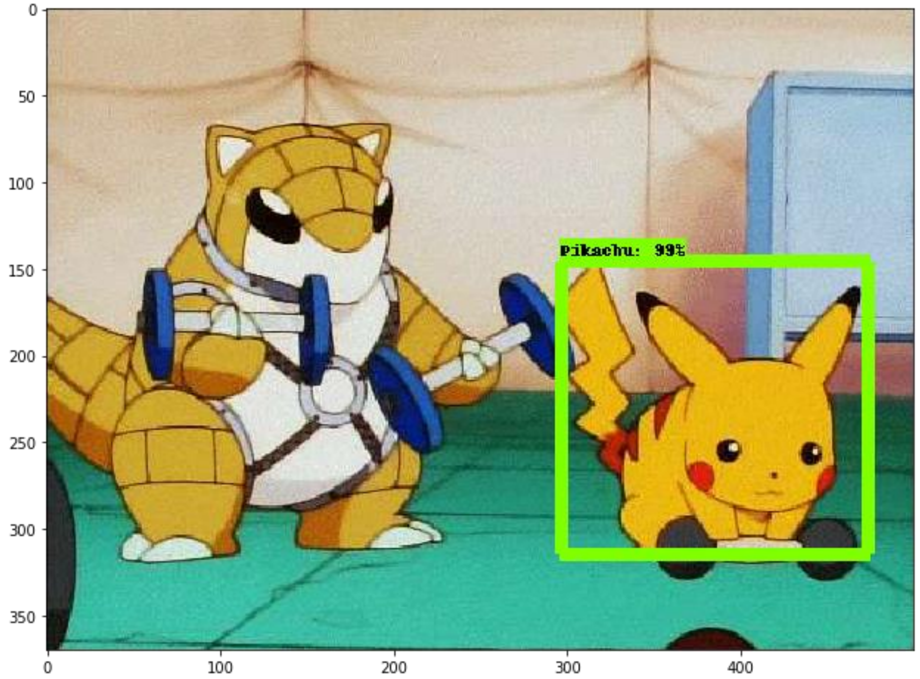
INTRODUCTION

What is Object Detection?

Definition: A field of computer vision to **detect** and **classify** objects.

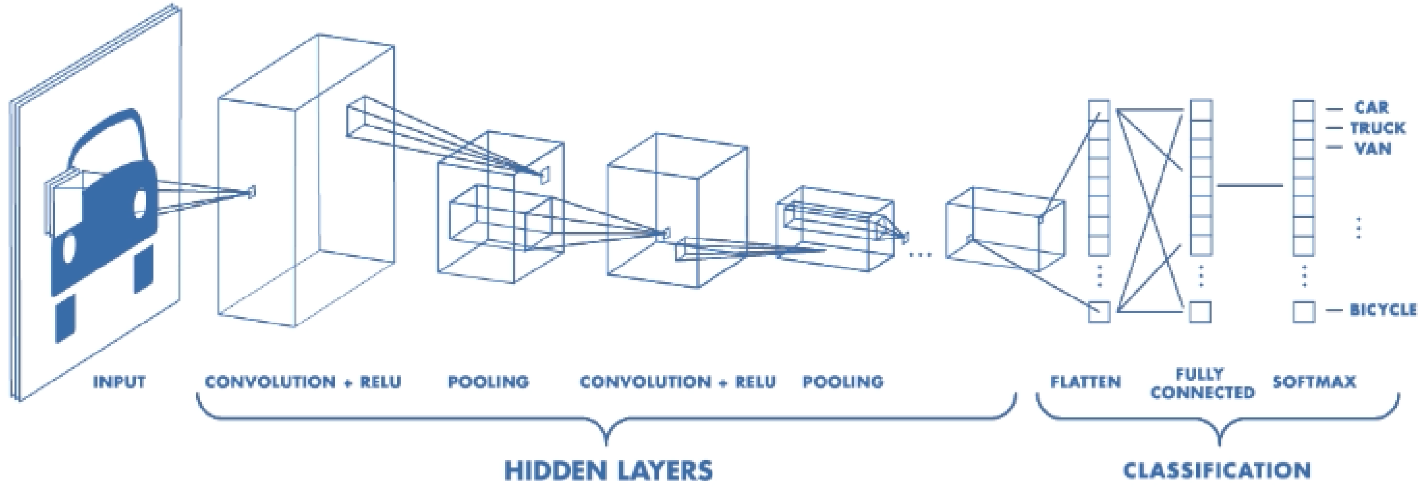
Applications:

- Self-driving vehicles
- Traffic monitoring
- Video surveillance



https://towardsdatascience.com/detecting-pikachu-on-android-using-tensorflow-object-detection-15464c7a60cd

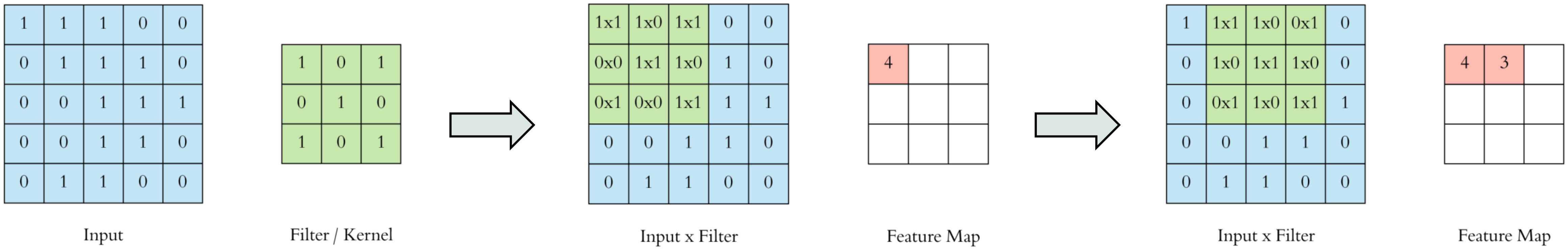
Convolutional Neural Networks



- Convolution
- Activation Function
- Pooling
- Classification

https://medium.freecodecamp.org/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050

Convolutional Neural Networks



https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

YOLO

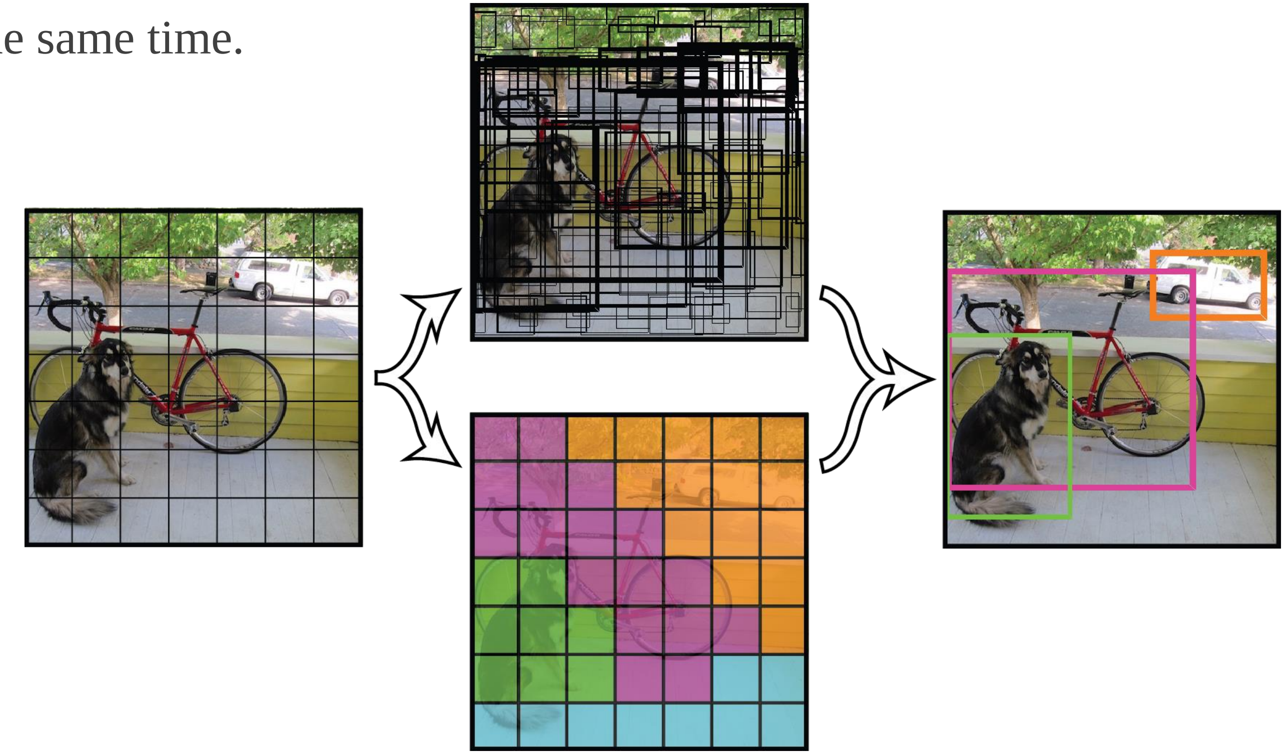
You-Only-Look-Once: **Detects** and **classifies** at the same time.

Steps:

I. Divide image into S x S grid.

II. Creates bounding boxes in each grid cell.

III. Each cell predicts class probability.



YOLO diagram [2]

YOLO

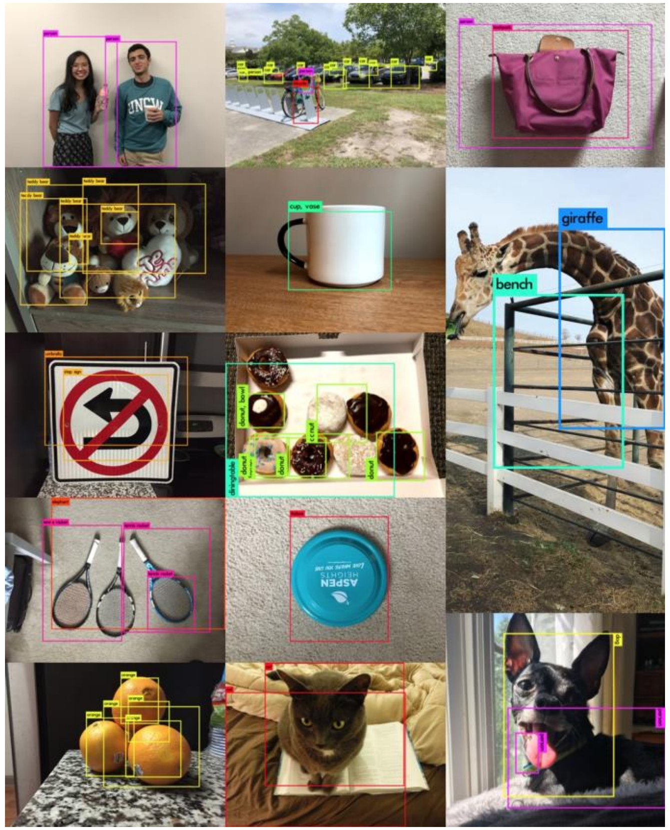
Version	mAP	FPS (GPU)	Dataset
YOLOv1	63.4	45	VOC
Tiny-YOLOv1	52.7	155	VOC
YOLOv2	48.1	40	COCO
Tiny-YOLOv2	**23.7**	**244**	**COCO**
YOLOv3	57.9	20	COCO
Tiny-YOLOv3	33.1	220	COCO

Tiny-yolov2-VOC without a GPU runs at around **2.4** FPS.

YOLO-LITE

YOLO-LITE

Developed to run **real-time object detection** on portable devices such as a laptop or cellphone without a **Graphics Processing Unit** (GPU).

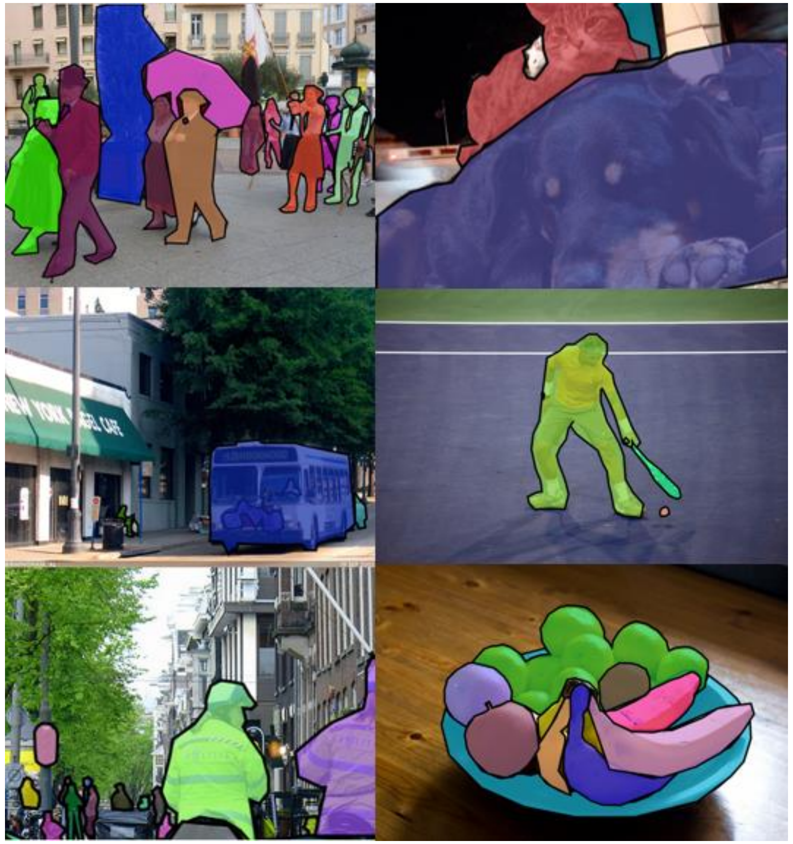


Goals:

- Achieve **real-time** object detection: 10 FPS.
- **mAP:** 30%.
- Implement resulting models onto a **website**.

Datasets

Dataset	Training Images	Number of Classes
PASCAL VOC 2007 + 2012	5,011	20
COCO 2014	40,775	80



Images from the COCO dataset [1]

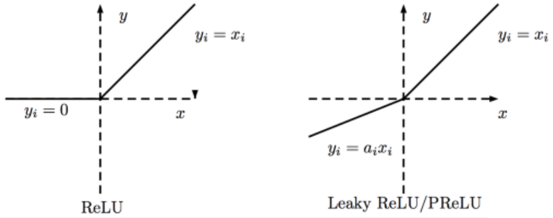
Architectural Elements

Batch Normalization:

- Creates zero mean/unit variance for every input layer in a neural network.
- Increases mAP

Activation Function:

- Creates nonlinearity in order to determine output of neural networks.



Input Image Size:

- Tiny-YOLOv2-VOC input image size: 416 x 416
- YOLO-LITE input image size: 224 x 224

Number of Layers and Filters:

- Tiny-YOLOv2-VOC layers: 9
- YOLO-LITE layers: 7

https://towardsdatascience.com/activation-functions-and-its-types-which-is-better-a9a5310cc8f

Results

Dataset	mAP	FPS
PASCAL VOC	33.57	21
COCO	12.26	21

Tiny-YOLOv2-VOC

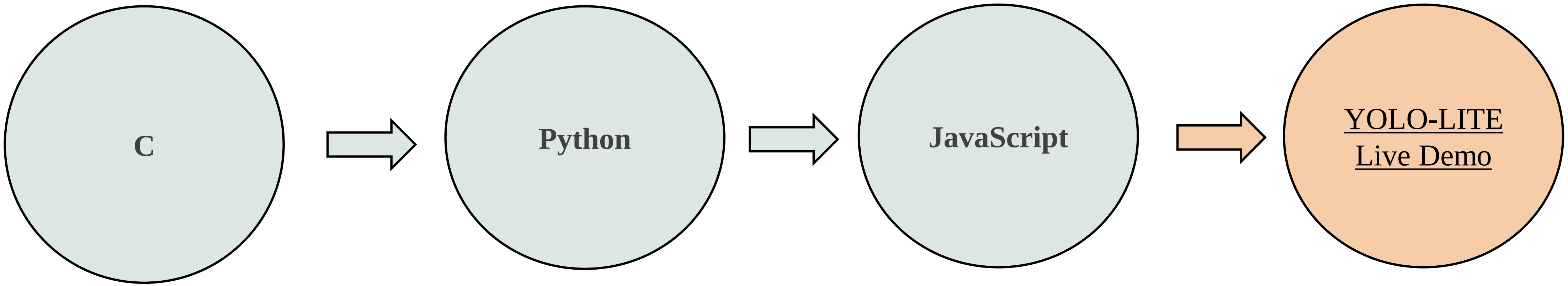
Layer	Filters	Size	Stride
Conv1 (C1)	16	3	1
Max Pool (MP)		2	2
C2	32	3	1
MP		2	2
C3	64	3	1
MP		2	2
C4	128	3	1
MP		2	2
C5	256	3	1
MP		2	2
C6	512	3	1
MP		2	2
C7	1024	3	1
C8	1024	3	1
C9	125	1	1

YOLO-LITE

Layer	Filters	Size	Stride
C1	16	3	1
MP		2	2
C2	32	3	1
MP		2	2
C3	64	3	1
MP		2	2
C4	128	3	1
MP		2	2
C5	128	3	1
MP		2	2
C6	256	3	1
C7	125	1	1

Web Implementation

Web Implementation



Conclusion

Conclusion

YOLO-LITE: **Lighter** model

Achieves **real-time object detection** without **GPU**.

Widely **accessible**

Future Work:

- Increase mAP:
  - **Pretraining** on ImageNet or CIFAR-10
  - Combining **R-CNN** and YOLO
  - **Pruning** weights

References

[1] COCO. Coco - common objects in context. http://cocodataset.org/, Last accessed on 2018-07-18.

[2] 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.

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

[4] PASCAL. The pascal visual object classes home-page. http://host.robots.ox.ac.uk/pascal/VOC/index.html, Last accessed on 2018-07-18.

Questions?