ELECTRICAL & COMPUTER ENGINEERING

UC DAVIS COLLEGE OF ENGINEERING

Towards Self-Driving Car: Pedestrian and Traffic Sign Detection

Christopher G. Uy, Kalani H. Murakami, Minh Truong, Gabriel J. Lee

Department of Electrical and Computer Engineering, University of California, Davis CA 95616

Project Goals

Apply deep learning to detect pedestrians and classify traffic sign in real-time, two key tasks for self-driving cars.

Approach & Milestones

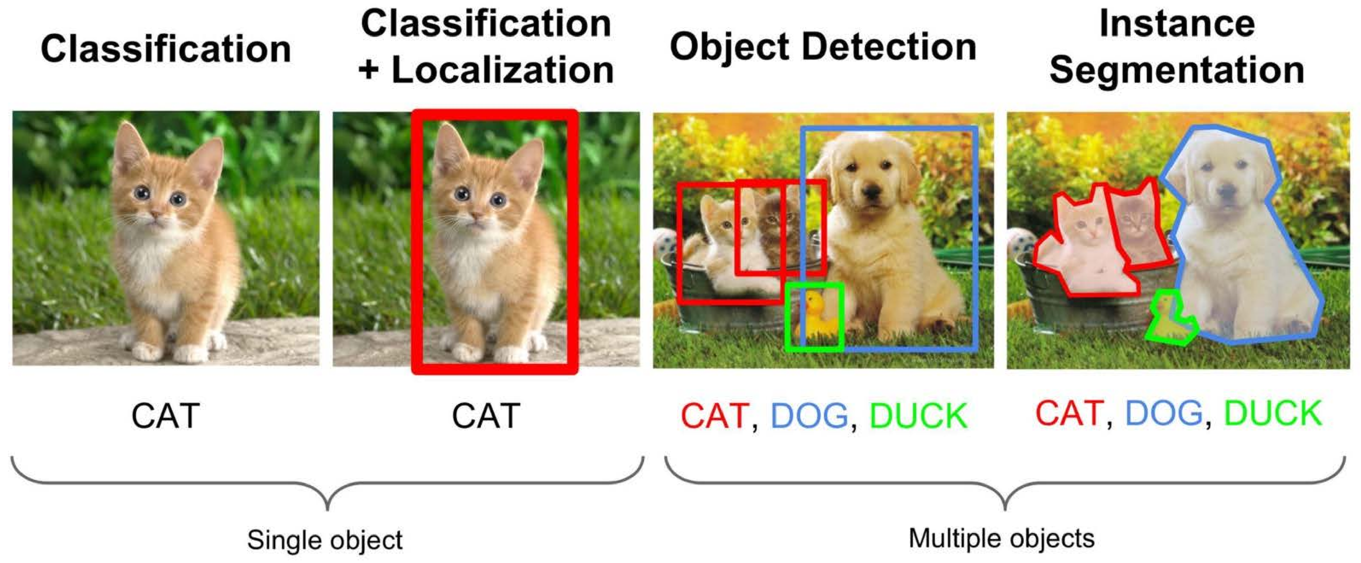
We used a modified YOLOv2 detection/classification algorithm and were able to run it in 30 fps (frames per second). Our bottleneck is the communication between the algorithm processor and the car.

Background

Deep Learning in Computer Vision

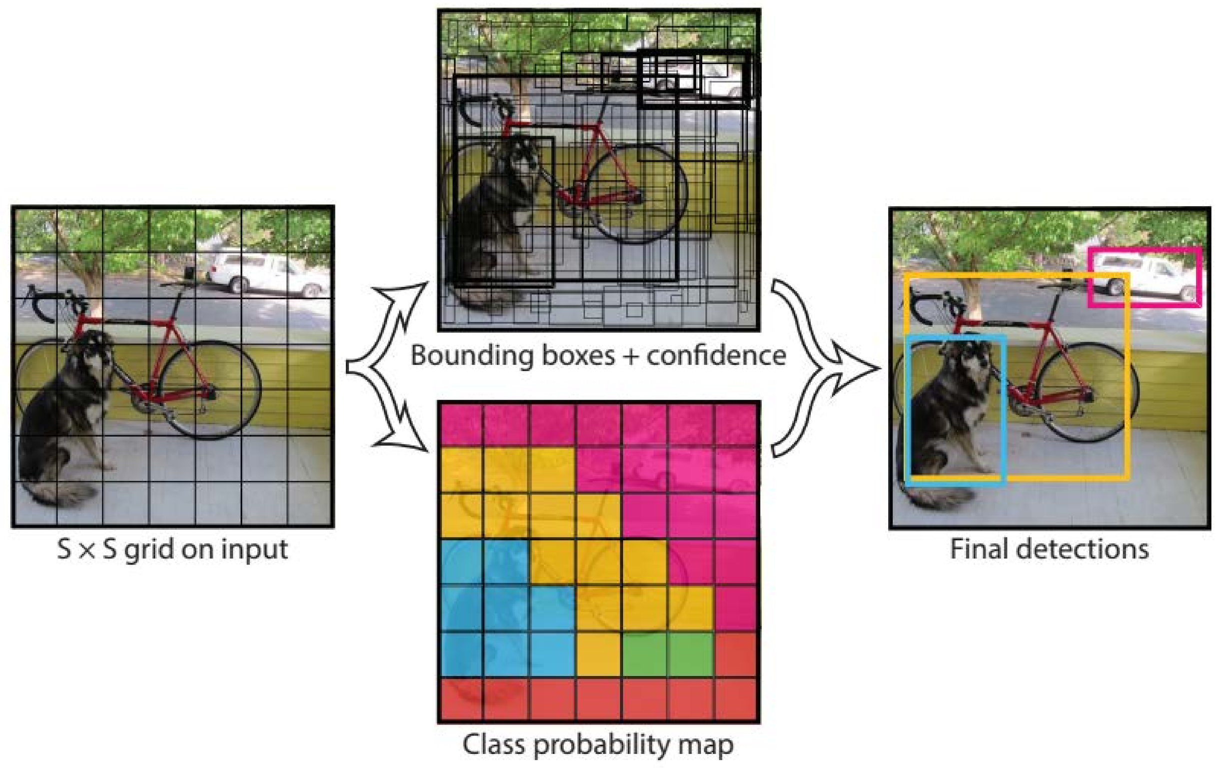
Deep learning allows computational models to learn data in multiple layers of abstraction. With deep learning and computer vision, we use a neural network to detect features in the image and associates it with the intended classifier. A efficient computer vision algorithm usually has a confidence level of at least 90%.

Classification vs. Detection



Object detection tell us where an object is within an image. Object classification tell us what that object is that is being detected. A self-driving car needs both a detection algorithm (to narrow down the position of objects of interests such as pedestrian and different signs), as well as a classification algorithm to identify object type (e.g., stop sign vs. speed limit sign). We used a modified version of YOLOv2 algorithm that unifies these 2 processes. This speeds up the processing time and allows us to perform object detection/classification in real time. YOLOv2 also achieves high accuracy for multiple classes.

YOLOv2: You Only Look Once

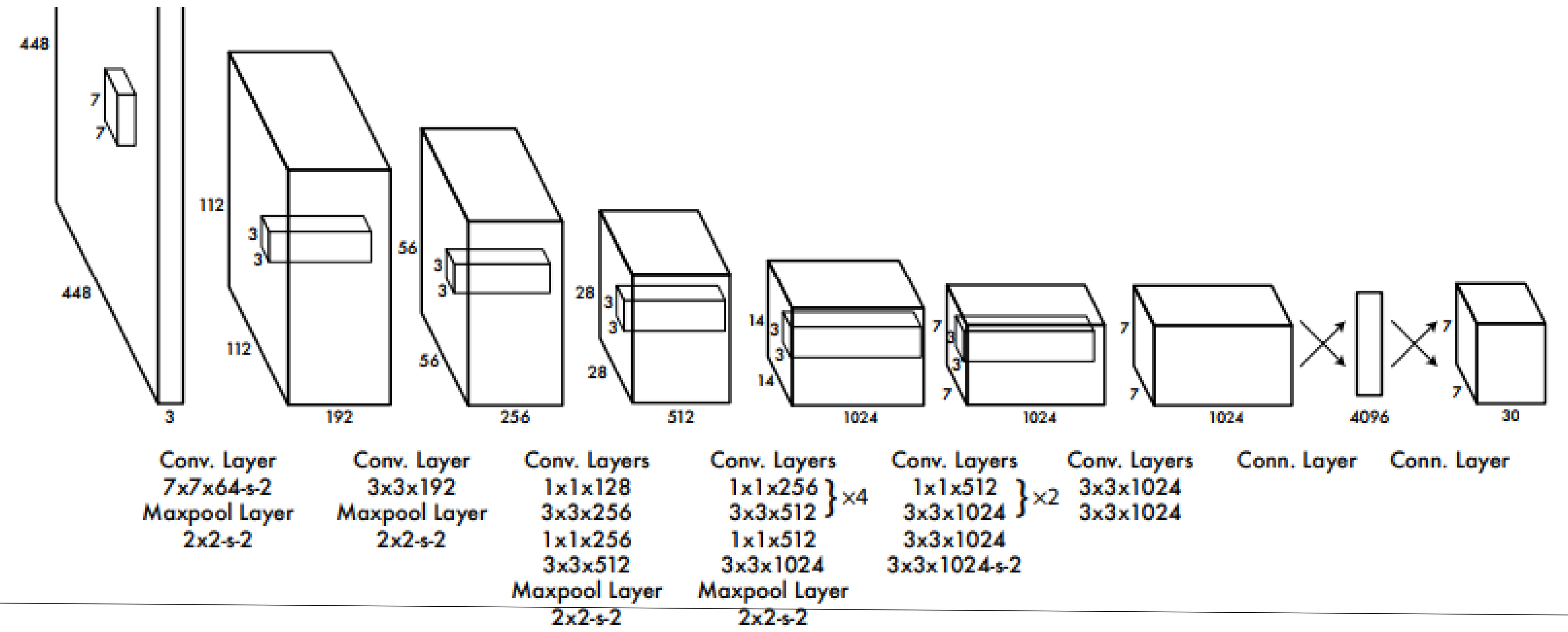


YOLO divides an image into grid cells. Each cell is responsible for proposing B bounding boxes and confidence levels for those boxes. For each object proposal in a cell, we have the coordinates of the center of the bounding box with respect to the cell, the width and height of the box with respect to the entire image, and is the confidence level of the cell's prediction, and a vector of probabilities corresponding to every classes. The class with highest probability represents the class of the bounding box and, consequently, the object itself.

YOLO Architecture

The platform that supports YOLO is called Darknet. Using Darknet, we were able to train our data in 2 different networks, Tiny YOLO and YOLOv2. Tiny YOLO has 16 layers in total:

- First 12 layers alternate between maxpool and convolutional layers.
- Convolutional layers generalize an object and make it position invariant
- Maxpool layers pick out the most important feature of the previous layers and reduce the dimension.
- Last 4 layers essentially form a non-linear voting system that looks at the features extracted by the previous 12 layers and assigns confidence levels to them.



Dataset Augmentation

German Traffic Sign Dataset:

Fifty different classes with over 50000 images in total, including the 35 speed limit, stop sign, and the pedestrian site.

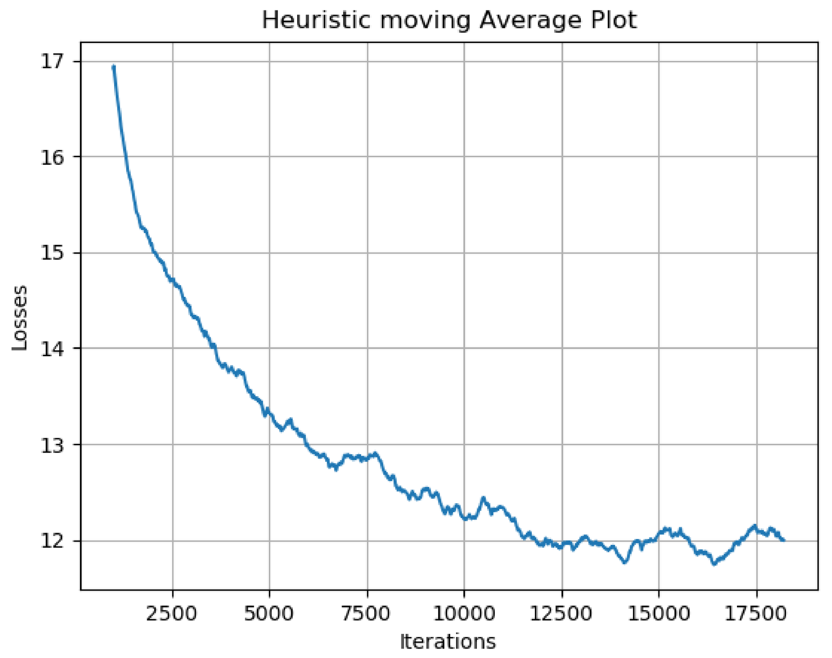
LISA American Traffic Sign Dataset:

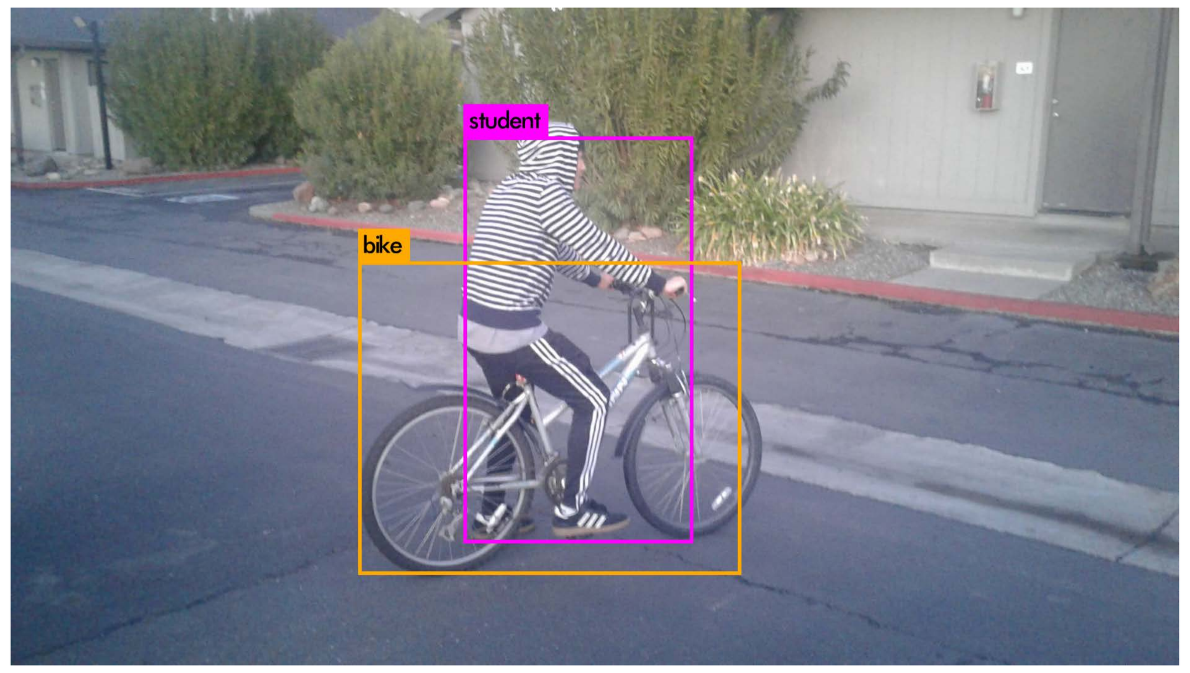
Forty-seven different signs with over 7785 annotations, including stop sign, 15/25/50 speed limit sign.

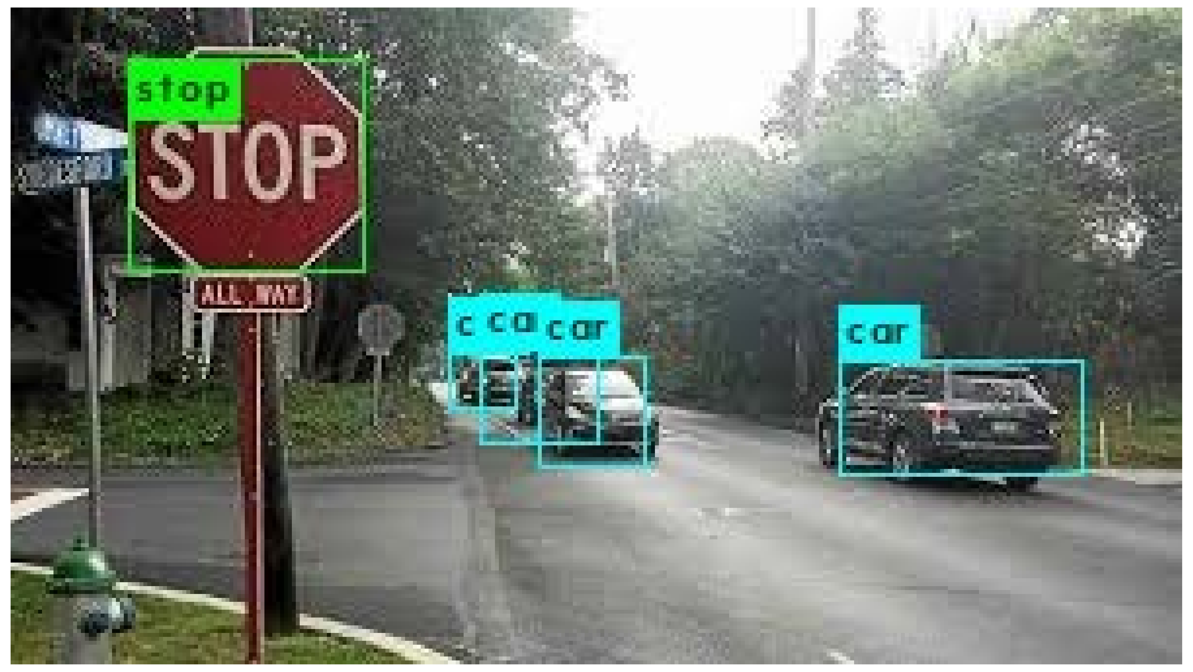
Training and Processing

We select images/labels of the wanted classes from COCO data set. By default, COCO gives us 2 files: images and labels. We augment the dataset with images but keep the labels the same.

We plotted the loss value, i.e., the error between label and prediction. Loss value decreases as our modes are optimized over time.







Results

Most important classes are: Pedestrian, Bike, Car and Stop sign, all of which are successfully detected with the average accuracy of 95%.

Acknowledgement: The Titan Xp used for this research was donated by the NVIDIA Corporation. We also acknowledge the REU support from NSF CCMI-1301496 grant.