Recent Tesla Model X AutoPilot Accident Analysis and Possible Solution via Traffic Sign Detection

Yufan Xue
Stanford University

Background

- Tesla Autopilot is an advanced driver-assistance system feature offered by Tesla that has lane centering, adaptive cruise control, self-parking, ability to auto change lanes without requiring driver steering. Recent report shows a Tesla Model X was involved into a fatal crash while the autopilot mode was on.
- we analysis the car accident and come up with a possible solution to avoid car crash by using traffic sign detection. To detect traffic sign, we implement Faster R-CNN and trained on GTSDB Dataset. The overall model is applied on the GTSDB benchmark and achieves 86.35%, 81.35% and 81.00% AUC (area under the precision-recall curve) for Prohibitory, Danger and Mandatory signs, respectively.

Problem Statement

- **Problem:** Given a car driving on the highway image/video, detect the traffic sign.
- **Approach**: Traffic sign detection via implementing Faster R-CNN
- **Evaluation metrics**
  - Train: RPN loss, ROI loss. Accuracy on evaluation dataset
  - Test: Same loss. Test dataset accuracy, confusion matrix

Dataset

- The German Traffic Sign Detection Benchmark (GTSDB), a single-image detection assessment
- 900 images, 600 for training, 300 for evaluation
- 1360 x 800 pixels in PPM format
- CSV format ground truth, for example: 00001.ppm; 983;388;1024;432;40. It following format as below
  - Leftmost image column of the ROI
  - Upmost image row of the ROI
  - Rightmost image column of the ROI
  - Downmost image row of the ROI
- Image contain zero to six traffic signs. The size of the traffic signs in the image very from 16x16 to 128x128

Figure 1. The place Tesla Model X hit the concrete barrier.

Figure 2. The image in GTSDB dataset.

Methods

- Dive deep into each component of Faster R-CNN
  - Image Preprocessing: resize it to desired target size so the larger dimension won't exceed the max size.
  - Anchor Generation Layer: produces a set of bounding boxes of different sizes and aspect ratios spread all over the input image
  - Anchor Target Layer: select promising anchors that can be used to train RPN network to distinguish between foreground and background and generate good bounding box regression coefficients for the foreground boxes.
  - Region Proposal Layer (RPN): from a list of anchors, identify background and foreground anchors; modify the position, width and height of the anchors by applying regression coefficients.
  - Proposal (Generate ROI) Layer: takes the anchor boxes produced by anchor generation layer and prunes the number of boxes by applying non-maximum suppression based on the foreground class scores. Then transform bounding box by applying the regression coefficients generated by the RPN to the corresponding anchor boxes.
  - Proposal Target Layer: select from the list of ROIs output by the proposal layer, then perform crop pooling from the feature maps produced by the head network and passed to the tail network to calculates predicted class scores and box regression coefficients
  - Crop Pooling: Proposal Target Layer produces ROIs, this layer will extract the regions corresponding to these ROIs from the feature maps. The extracted feature maps run through the tail network to produce the object class probabilities distribution and regression coefficients for each ROI.
  - Classification Layer: input Crop Pooling's output, output class scores and bounding box

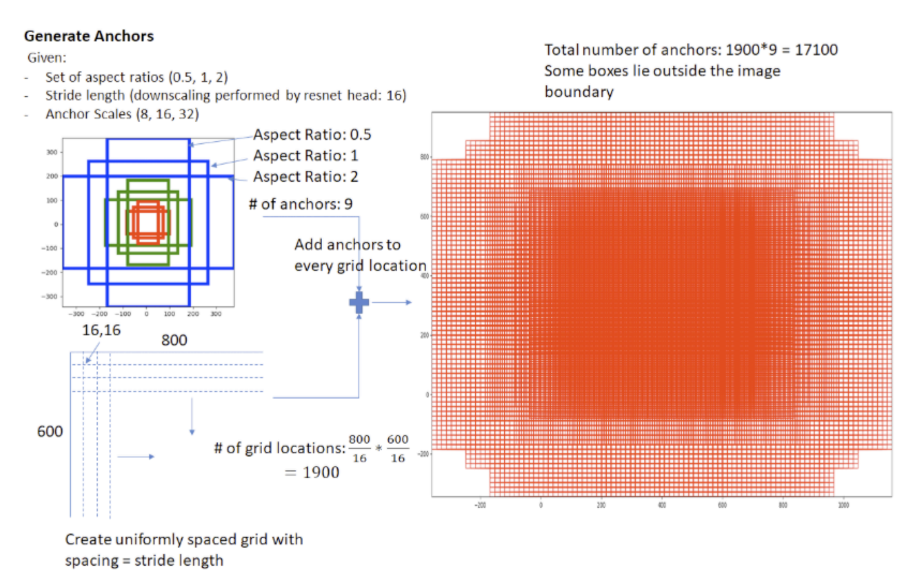

Figure 3. Anchors generated by anchor generation layer

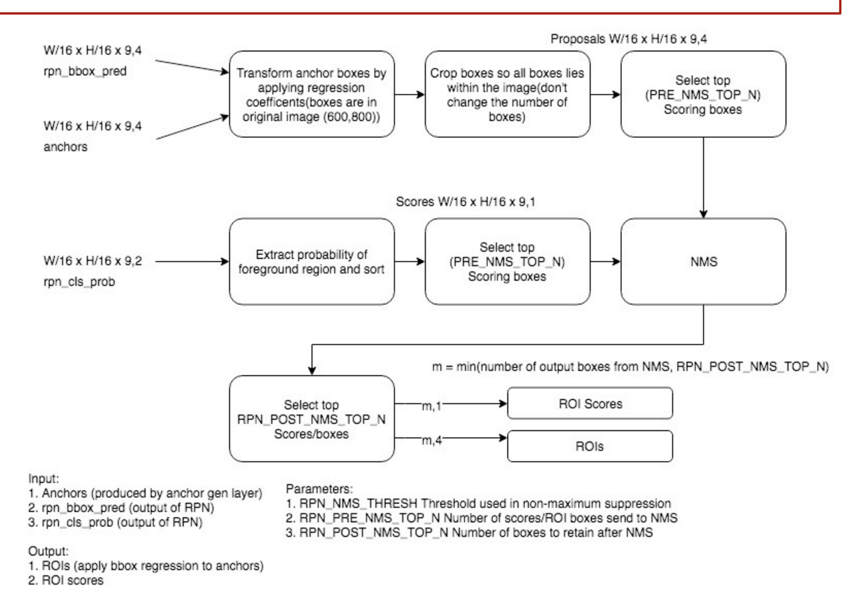

Figure 4. Proposal (Generate ROI) Layer

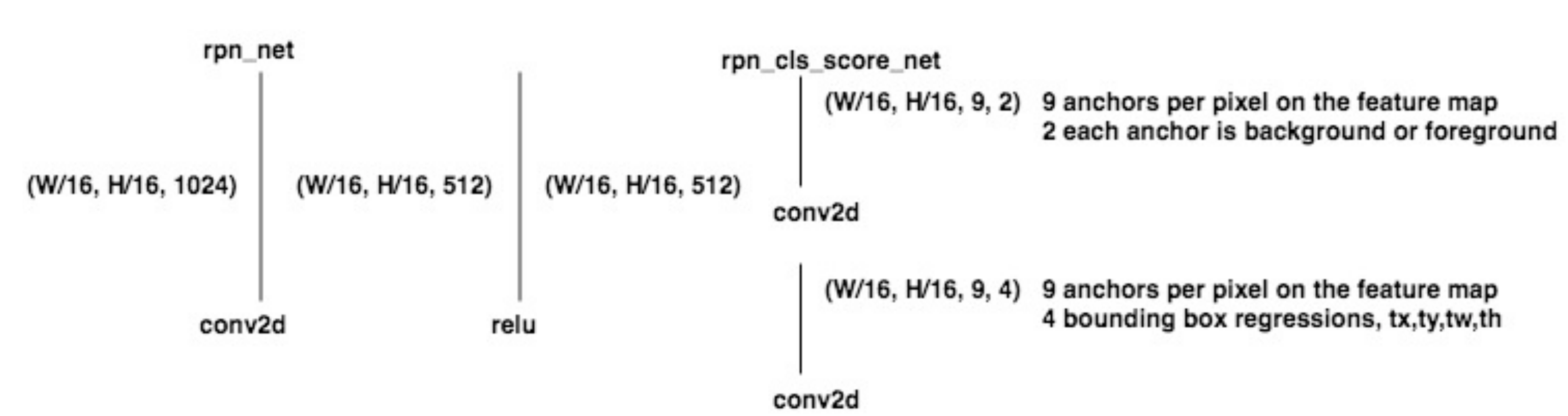

Figure 5. Region Proposal Layer (RPN)

Results

- Using pre-trained VGG-18 on Imagenet then train RPN and ROI net on GTSDB dataset
- Trained on GTSDB dataset and evaluate it on its test dataset
- Classification training loss on RPN network reduced to 0.0289 at 170 epochs
- RPN regression training loss reduced to 0.019 at 170 epochs
- ROI classification loss reduced to 0.031 at 150 epochs
- ROI regression loss reduced to 0.09 at 170 epochs
- Validation accuracy on GTSDB val dataset reaches around 71%

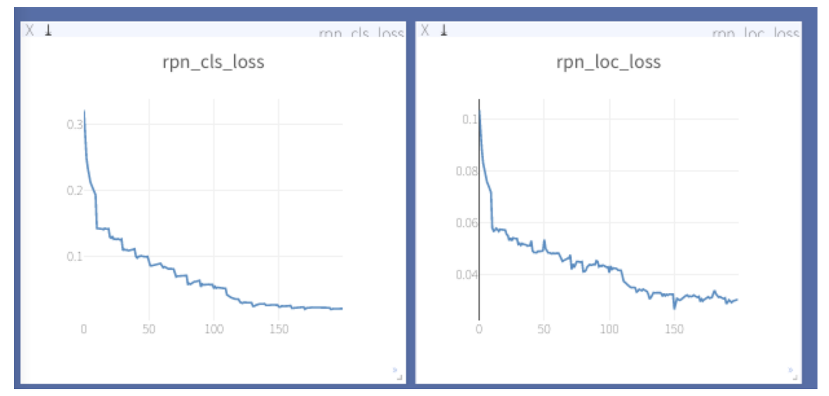

Figure 6. classification and regression loss of RPN network

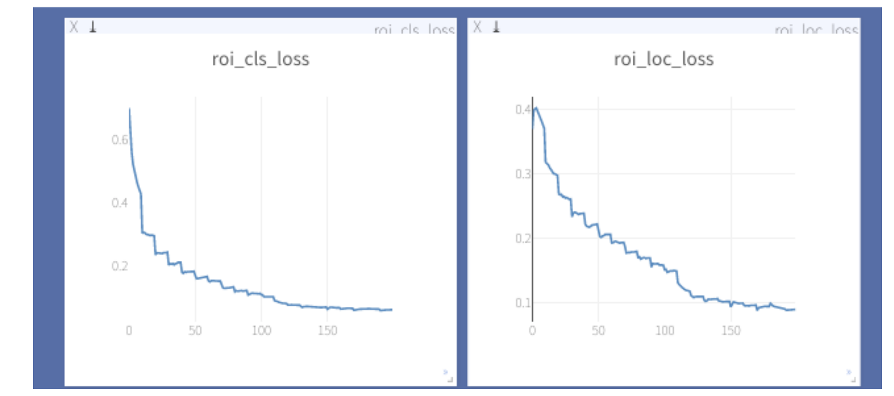

Figure 7. classification and regression loss of ROI network

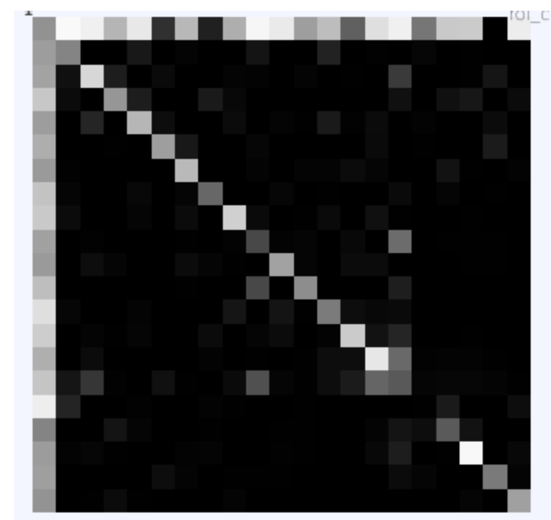

Figure 8. Confusion matrix for 43 classes on GTSDB dataset

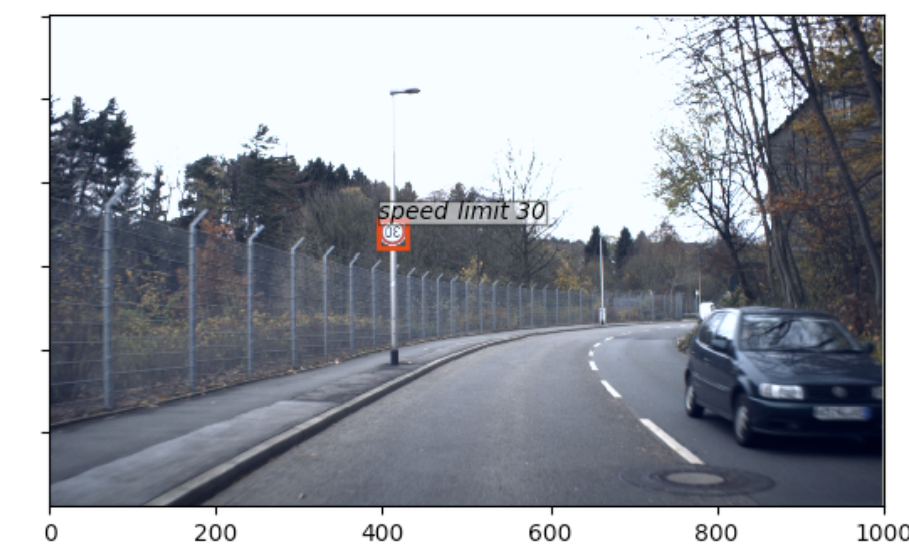

Figure9: Test on GTSDB test dataset

Conclusion

- Faster R-CNN is the state of art Object Detection algorithm
- Implement Faster R-CNN is very very hard, there are lots of tricks which are unknown until we really start writing code to implement it
- Although traffic sign could be detected via a fine-tuned Faster R-CNN model, however the distance between the car and sign is not be able to calculated. Traffic sign detection could give good information to AutoPilot system about whether there is a sign in front of the car and which kind of sign it is in front of the car, but the AutoPilot system would still need other methods to get distance information.
- Future work:
  - Try out YOLO method
  - Try out different traffic sign dataset
  - Use ResNet as feature extractor