Multiple Cars Tracking Basing on Kalman Filter

ECE 251B –Digital Signal Processing Ⅱ

Team #13
Pengluo Wang; Kaiyu Zhang;
Shihao Luo

UC San Diego
Electrical and Computer Engineering

Contents

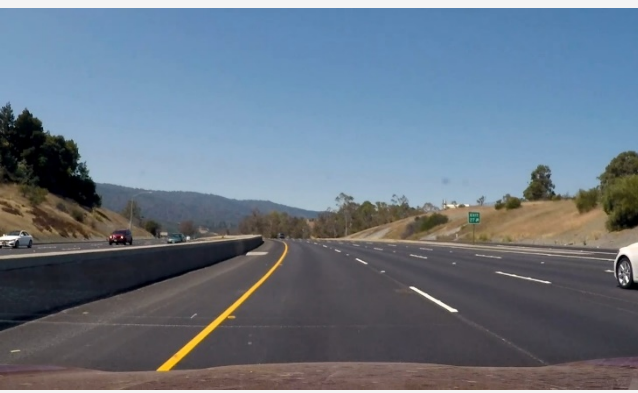
Problem

- Background
  - The automatic driving is one of the most important technical achievement, which still needs more promotion
  - Implement a set of algorithm to combine car detection and tracking together, to provide convenience for car driving.

- What we did
  - Use different methods implement the car detecting and tracking, and add the window marking the multiple cars inside the video.
  - For car detection, we choose Tensorflow Object Detection API to make detection
  - After getting the information of detected cars(location and velocity), we make the prediction using Kalman Filter, and combine the prediction & measurement value as the real tracking value, to finish whole algorithm.

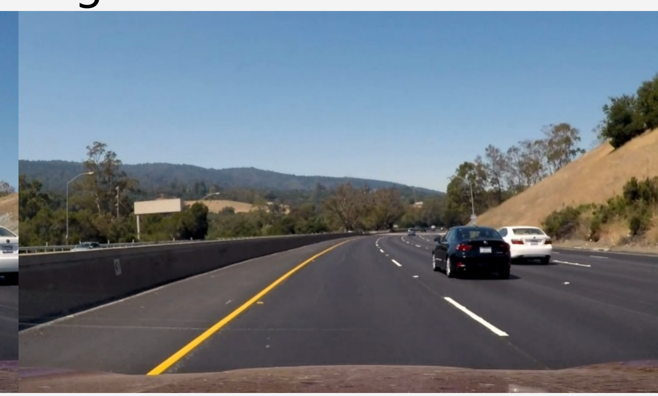
Dataset

- Video Data
  - The Data we use is a set of video taken by the dashcam in front of our car, which is real and reliable.
  - The Data was taken in our daily life, including different situation: No Cars, One Car, More than One Cars in the visual area.
  - The cars in the video have several typical behaviors including intersection and



(A)

(B)



(C)

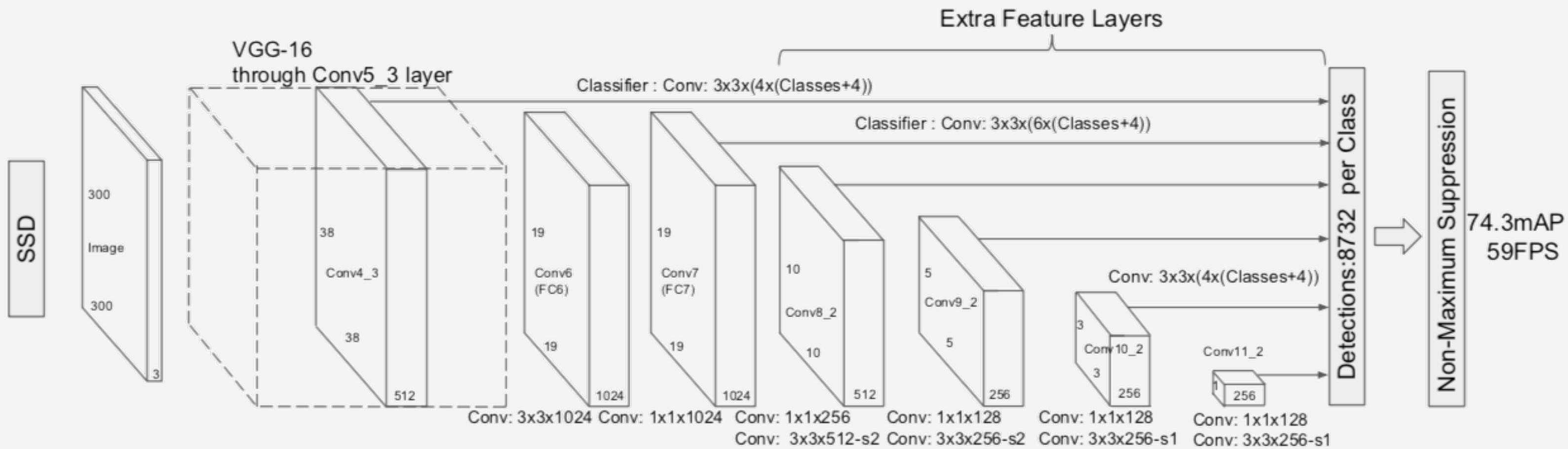
Method Introduction-Detection

- Tensorflow Object Detection API
  - TensorFlow Object Detection API is an open source framework built on top of TensorFlow that makes it easy to construct, train and deploy object detection models.
  - To use this build-in model, we applies SSD with MobileNet, a Single-shot-detection with mobilenet architecture trained on COCO test set.

- SSD - Single-Shot-Detection
  - SSD algorithm is an object detection algorithm that directly predicts the coordinates and categories of the objection with bounding box.
  - SSD is much faster than those traditional detection models without losing the accuracy, by adding several separate prediction filters

Method Introduction-Detection

- SSD - Basic Frames
  - Using a small convolutional filter to predict object categories and offsets in bounding box locations
  - Using separate predictors (filters) for different aspect ratio detections
  - Applying these filters to multiple feature maps from the later stages of a network in order to perform detection at multiple scales
  - Train the data with ground truth boxes, which make the model can be used for detection and prediction



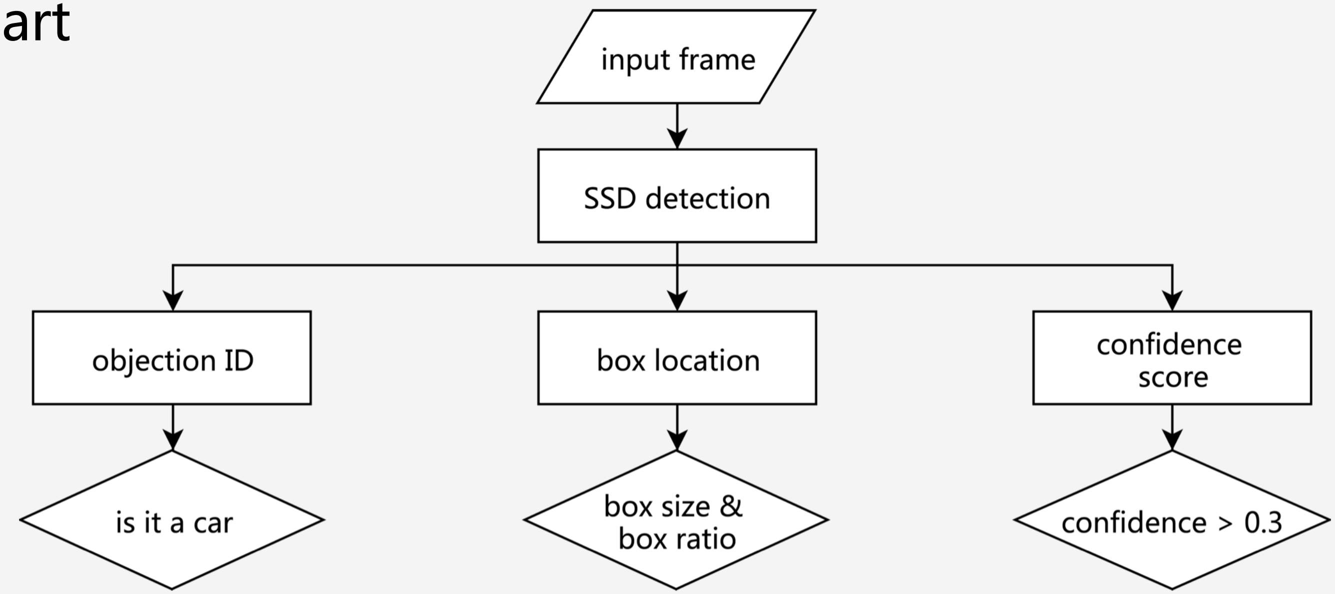
Method Introduction-Detection

- SSD – Outcome
  - This model has been trained with COCO data set, which can detect the following objects related to transportation:

Label Number	Objection
1	Person
2	Bicycle
3	Car
4	Motorcycle
5	Bus
6	Train
7	Traffic Light
8	Stop Sign

Method Introduction-Detection

- Flow Chart



- SSD-Advantages
  - High Speed
  - High Accuracy
  - Can be applied with low resolution and video data

UC San Diego
Electrical and Computer Engineering

Method Introduction-Detection

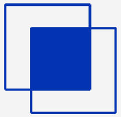
- SSD – Outcome
  - Using a test image, we have the following outcome:

Method Introduction-Tracking

- Kalman Filter
  - Kalman Filter is a kind of adaptive filter.

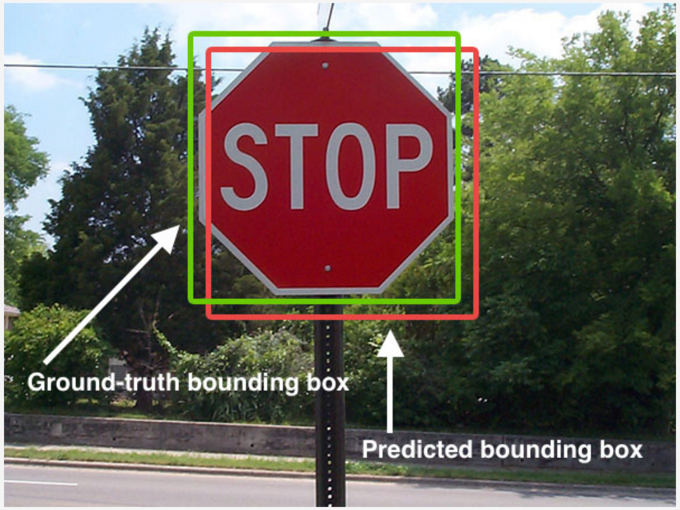
Intersection-over-Union (IoU)

- IoU
  - A parameter shows the accuracy of detection and prediction.
  - Can be understood as the accuracy rate for each image



$$\mathrm{IoU} = \frac{\mathrm{DetectionResults} \cap \mathrm{TrackingResults}}{\mathrm{DetectionResults} \cup \mathrm{TrackingResults}}$$

  - For example:



Detection-to-Tracker Assignment

- Linear-Assignment
  - One of the fundamental combinatorial optimization in order to find a maximum weight matching (or minimum weight perfect matching) in a weighted bipartite graph.
  - Implemented by package *sklearn* in Python.
  - Produce the max matching between detection and tracking box in our project. The results are stored in list matched, unmatched_detections, and unmatched_trackers.

	D_1	D_2	D_3	D_4
T_1	0	0	1	0
T_2	1	0	0	0
T_3	0	0	0	0

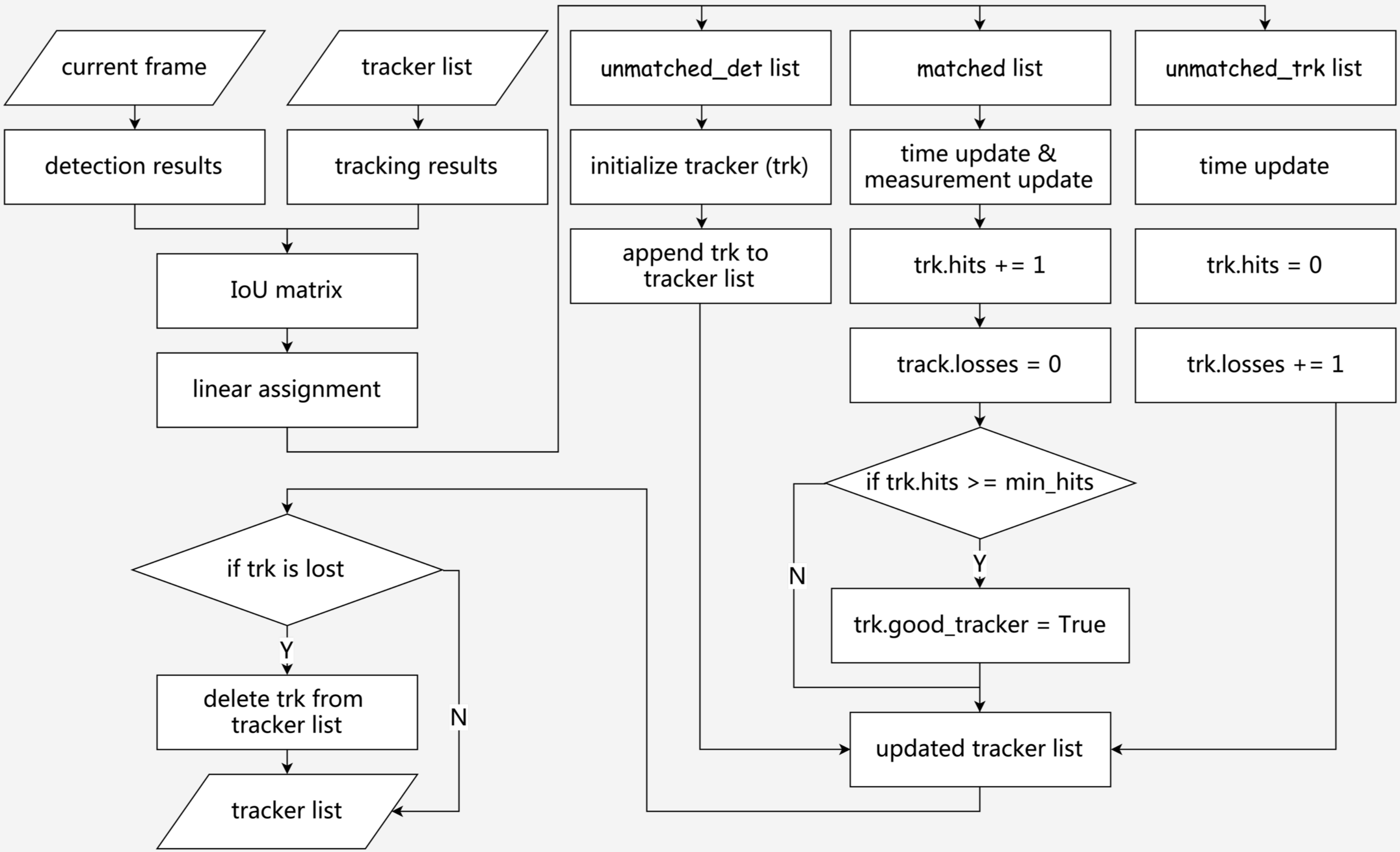
→ IoU matrix

Pipeline

- Sequential processing
  - Process each frame of the input video, implemented by package MoviePy.
  - Deals with matched, unmatched_detections, and unmatched_trackers obtained from linear assignment of the IoU matrix for each frame.

- False alarm & missed detection
  - Introduce two important parameters, min_hits and max_losses: min_hits stands for the minimum number of consecutive matches to establish a tracker; max_loss stands for the maximum number of consecutive losses to delete a current track.
  - Introduce a variable to discriminate false alarm and real target, a tracker is called *good tracker* if it has reached min_hits.

Pipeline diagram



Solution & Frame

Solution & Frame

UC San Diego
Electrical and Computer Engineering

Solution & Frame

Solution & Frame

Solution & Frame

Solution & Frame

Result

Reference