

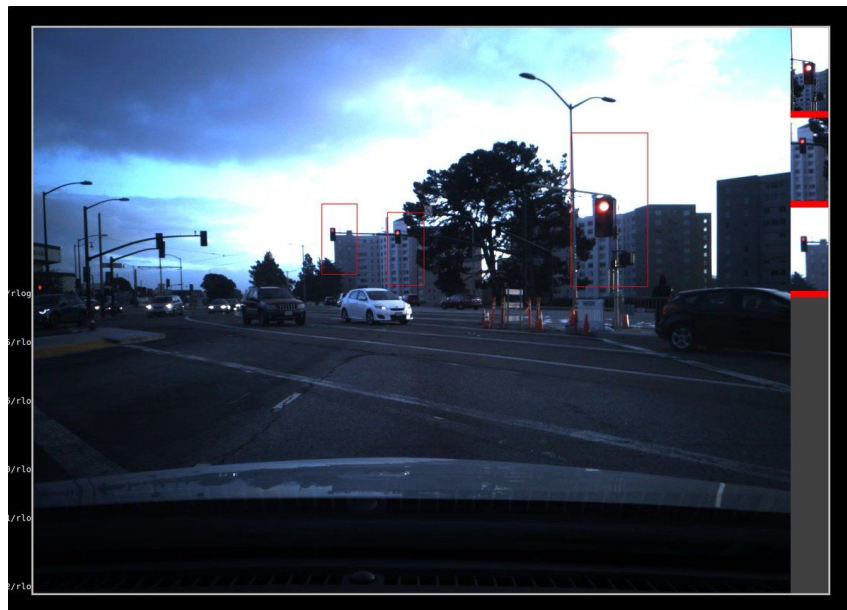
# OpenPilot

## Street Lights and Signs Recognition

By comma(Elliot)

### 1) Introduction

Although there has been teaser for traffic lights by comma however it didn't go anywhere due to reason openpilot wasn't ready. Openpilot is able to handle lateral control(steer) and longitudinal control(ACC-Active Cruise Control) pretty well however there wasn't full stop and go solution until comma pedal(interceptor) was released earlier of 2018 for Honda's only and by end of the year we had for Toyota and Tesla which wouldn't be possible with effort from community.



Model teaser for traffic lights recognition on [twitter](#)

Besides pedal there were other things that were required to solve the puzzle where last year 2018 was BIG year where comma released multiple tools where community can do a lot more with OP. In 2018 a lot of new cars were ported which makes this project even more reason for devel. Following is list some of tools with description involved for this project that comma released which will make this project possible with description involved on the project

- Forward Collision Warnings([Medium](#))-Although it won't be used for traffic lights and signs project however it is good reference where OP can do breaking using

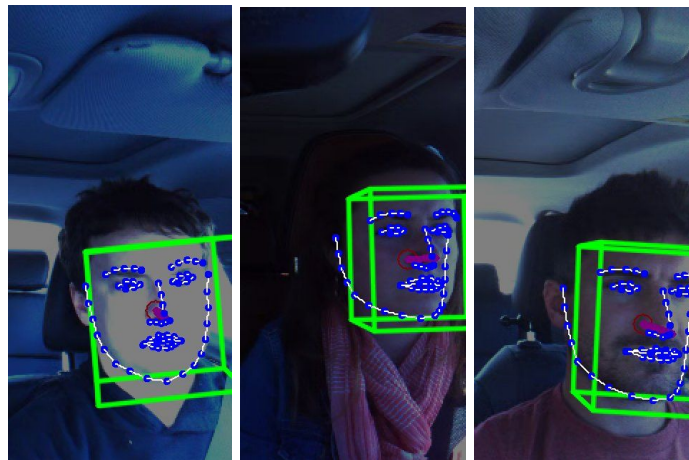
longitudinal control. You can see [this](#) video where FCW is used on real time situation.



Forward Collision Warning indicator on the dashboard

- Safety and Driver Attention([Medium](#))- On Medium, It doesn't talk much details on dilemma but you can find them on [Github](#) where OP had it since OP(OpenPilot) release of v0.2.2 similar to AP(AutoPilot). I won't be going in much details regarding dilemma but it is good source which will be used for traffic recognition. Driver monitor uses machine learning [model](#) which works in assets of [dilemma](#).

❖ **Summary of dilemma and to be used on traffic recognition** : If user is distracted, OP will disengage and come to full stop which is similar situation to traffic lights and signs, Where we need to come to full stop and then resume that's the only difference but similar logic will be applied from dilemma.



Driver monitor in action.



openpilot user getting a distracted alert (real user photo, not actor)

- Beginning to maps- In OP release v0.5.7, OP finally had the first release of using maps. Although not [HD maps](#) yet but instead [OSM](#)(OpenStreetMap) where OP can finally slow down at curves using maps data along with it stay within speed limit which is all done using API calls. This will be second stage logic for traffic recognition where real maps data will be used to get more accurate location for stopping. OSM does have mapping points where [traffic lights](#) and [signs](#) can be detected.



Example of slow down at curves and staying within speed limit using OSM

- Last piece of puzzle was visiond which is finally [open source](#) in recent release 0.5.8 where we can finally had new hardware for OP when needed for the project.

- For better training of model we end up using other new tools [OP tools](#) and [comma2k19](#) 2k will be used data purpose for mapping and OP tools for debug as shown on video teasers below.

## 2) TLAS(traffic lights and signs) problem explanation

The problem is not detecting TLAS but instead the issue is where to stop and when to resume. As mentioned above examples where OP is already can come to full stop and resume using longitudinal control. When it comes to just full stop by itself it can do that by FCW, Driver monitor and last of all it can slow down using maps data by OSM.

Since it's complicated problem i wanted to come up with localization based solution where no maps are used for stage 1 but instead logic is used. Due to reason EON currently has 1 camera detect vehicles and object in front, TLAS won't be L3 solution at all it will be more of L2 where human still has to react to situations although which will change in future where new hardware can be added since visiond is open source. Following i will describe the IF's situation with explanation. There are more situations with stop signs then traffic lights.

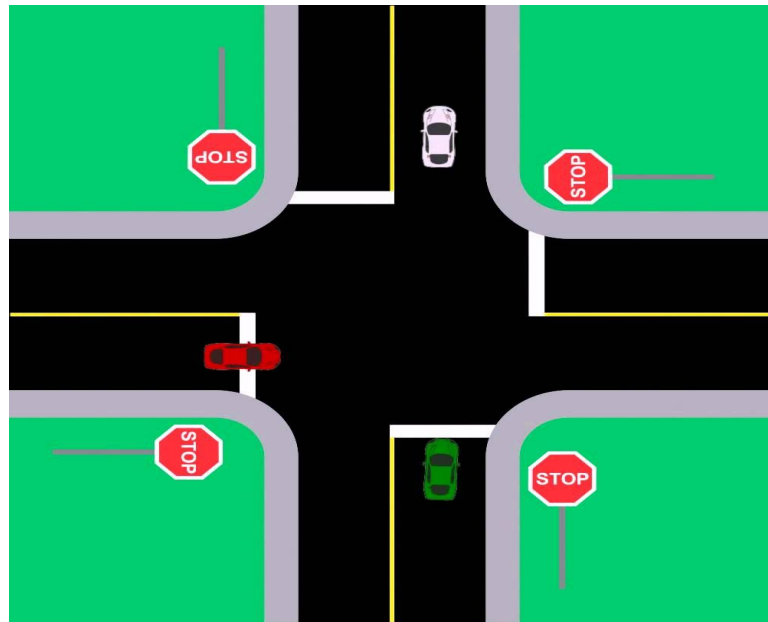
- I will start the stop signs if you look at the camera view of EON it's not able to detect full view left and right view so there are blind spots at intersection that EON can't see.



Blind spots at intersection found on [cabana demo](#)



- Another problem is prediction- If your and another vehicle comes at the intersection at same time there is no safe way to predict when to resume along with it what if there is pedestrian crossing from opposite directions and then wants to come to your direction which are in general traffic situations that we as human can understand but as far as ML(machine learning) goes it will take a while until it's close to perfect and to get there we need more hardware for OP in order to even get to that level.

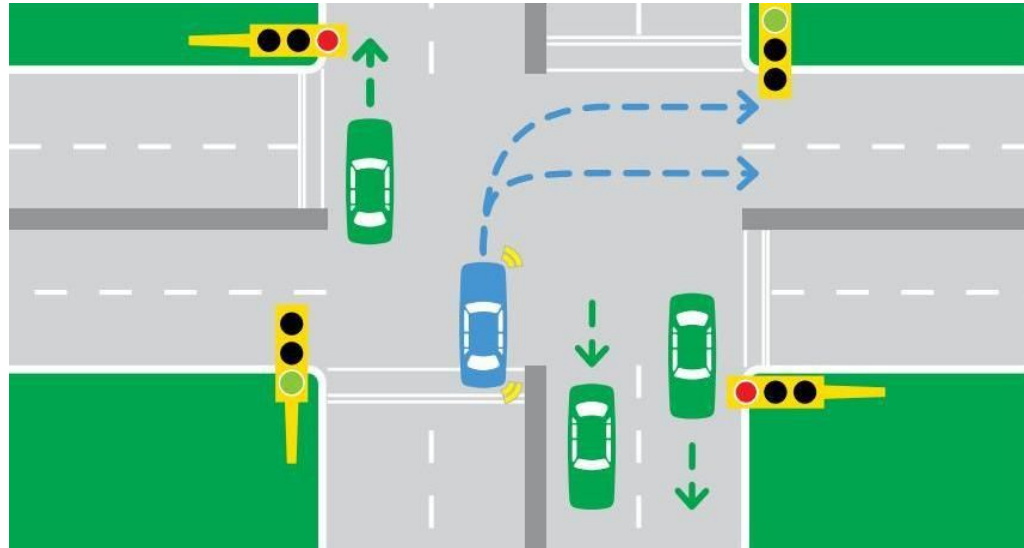


Intersection problem with other cars



Intersection problem with pedestrian.

- Biggest problems are road conditions- Such as lights not working, missing lanes this is where maps comes handy in general.
- **Traffic lights problems**



- Issue with lights is not detecting the colors but it's stopping location.
- Above is perfect world traffic signal intersection however it can vary on each intersection if paint is missing it will be hard to find perfect stop location. Comparing the situation with stop signs where it's predictable when to stop if stop sign is detected stop within 4 feet however with traffic lights it's a different situation.
  - After spending entire day driving in different cities, where I was looking at traffic lights intersections what I noticed was not all traffic lights are equal.
    - If traffic lights were present before the intersection just like stop signs it would have been very easy to predict where to stop however it's not that case. Most of the traffic lights are opposite side of the intersection and if the intersection has 4 way lanes on each side with total of 8 lanes total the traffic light will be too far to see and stopping location will be hard to predict if no lanes are present.

### 3) TLAS L2 Solution

Traffic lights and signs have completely different situations so for that reason there are two solutions for each situation.

- I will start with stop signs again since i already have explained the situations with stops signs. Solution for stop signs is simple,ML will detect Stop signs which are located on right side if present if not OSM or Maps in general will come handy in future but in stage 1 it will detect for stop signs besides that it will look for the crosswalk border line if present if not stop it will stop within 4 feet however using data speed needs to be within OSM speed limit where safely can be stopped within 150 feet.



“Stop signs should be placed so that the sign roadside edge is at least 6-12 feet from the edge of pavement, with a 2 foot minimum for curbed roadways. Placement of the stop sign is to be on the right side of the roadway at either the stop line or (4) feet in advance of a crosswalk. If there is neither a stop line nor a crosswalk the sign should be placed where traffic can stop and clearly see both directions before entering the intersection, and no greater than 50 feet from the intersection.”

- ❖ Due to safety reasons as mentioned above on the situations, It will only come to full stop and not able to resume by itself so for that situation you all you to do is cruise resume so you could continue on same intersection when safe where you do not have to press on brake or gas pedal.

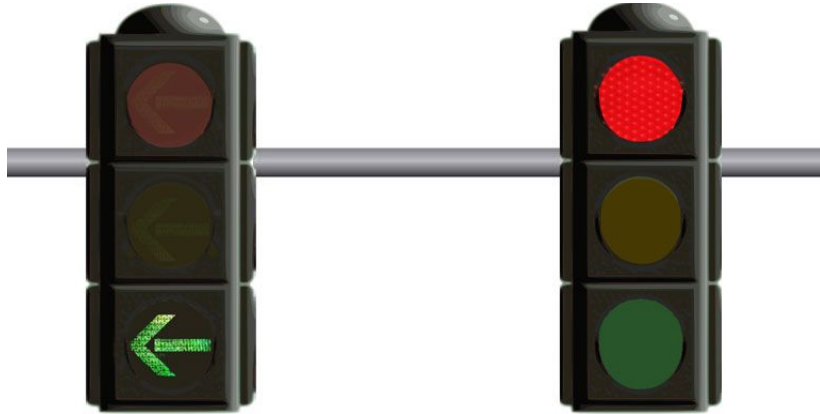


- Traffic lights will follow similar logic as stop sign where it will come to full stop after detecting the traffic light being red or yellow and it will stop before the crosswalk if present in perfect world if not OSM will be used as backup. Major difference with traffic lights will be auto resume since we can somewhat predict the situation not fully since it's L2 solution.



- ❖ Traffic lights with arrows going left or right will be completely ignored since OP can't make full turns it will be meant for straight lanes only.





- ❖ In perfect world when there is green light there shouldn't be any pedestrian on your way which is reason why auto resume for traffic lights is more predictables but you will be still required to pay attention since it will be L2 based detection. If emergency situation does occur that's where FCW will come handy but don't rely on it too much just in case of if situations. I'm not saying auto resume will be safe this is where dilemma comes in as well where making sure you are paying attention for any situation in general.
  - ★ For safety reasons we might as well add another factor where before resume you will be required to touch the steering wheel just to confirm which can be probably used for stop signs as well. Will need more feedback.
- ❖ Complex problem about lights is that the lights are usually located after the intersection unlike stop signs before the intersection for that situation. For localization purpose we need to figure out typical intersection size with crosswalks before the border white line. It will be safer if stopped before the white line even if it's like 5 feet off rather than crossing the intersection with danger. So in general solution main focus for stopping is white line before intersection this is where ML will help.
- ❖ Another solution will be using traffic lanes end spot, the traffic lights poles before the intersection if present. Best practice for stopping will be using OSM so at least speed can be decreased before the intersection.

Here is [paper](#) for intersection which goes on details size.

#### 4) Model(ML)

Model will be look at for just stop signs as far as lights goes it will only detect red, yellow and green if any arrows or any other signs will be ignored for now.

- Model will be heavily trained based on intersection's white line when detected prepare to slow down or completely stop if any TLAS are present.
  - ★ Traffic model will be running along with driver model where it will need visiond access. So it will be parallel model to driver model which will have completely different purpose. As or right now extra hardware is needed.
- @willem was nice enough to share a simple neural network to recognize the color of traffic lights on official comma's repo located at

<https://github.com/commaai/trafficlights>

#### 5) Prediction Summary

- Stop signs

The model will be trained to look for white lines along with it look for stop signs where stopping location will be within 4 feet after sight however before the stop sign prediction for slowing will be needed this is where OSM will come handy. Users will be required of course manually edit OSM to have prediction to slow down before stop sign.

- Traffic lights

The model will be trained to look for white lines no matter if it will be stop sign or traffic lights for localization solution. OSM will be used again for more accurate location if needed user can manually edit the maps again at OSM. Predicting stopping location for lights is a challenge since it's not like stop signs at all where stop sign comes first then the intersection so stopping location is not possible without maps and localization. Since it's real time situation we are dealing with API calls can be tricky so i have thought of localization solution where geo location will be used. solution which will be used for daily commute.

- ★ There will be more likely a JSON where you can add all of your geo location for your daily commute which will be saved on EON so there won't be any issue for your daily commute since stopping location will be predicted.

## 6) Mapping with OP using Localization

After pulling hair for weeks, I have finally came up with a solution for traffic lights which won't be possible without maps .Since it's real time situation we are dealing with API calls can be tricky so i have thought of localization solution where geolocation will be used. Please note this won't be true for every road condition however it will be good solution for daily commute. Daily commute will be beginning to something awesome where we could even figure out mapping even more since there will be trail errors to begin with and learn from mistakes afterwards.

### Offline maps basics

It will works using offline maps where the values of geolocation will be used. If you have used offline maps, You are probably familiar with downloading file where files can be pretty large good example will be google maps where you can have can download areas to navigate offline or any other app in general.

### How will we map?

OSM has done a lot of mapping already so why build from scratch? When mapping already exists. We will just export data from OSM and use it based on our preferences.

### How will it work?

Using geolocation by defining values of slowdowns used as prepare to stop if needed and stopping location with speed limit in JSON format where we can finally have prediction planned except for running into deadzone.

Current maps on OP can already slow down at curves so we will be using similar logic as current.Following is current OP code for mapping as you can see where speed limits are defined.

<https://github.com/commaai/openpilot/tree/devel/selfdrive/mapd>

### How can we run maps offline?

I think best technique for this project will be reverse geocoder where we can have all of the values downloaded on EON and if match of geolocation code it based on it so there will be slow downs and stops with speed limits based on the geolocation.

Good Tools for reverse geocoders.

1. <https://openaddresses.io/>
2. <https://github.com/thampiman/reverse-geocoder>

Other approach will be running OSM offline

[https://wiki.openstreetmap.org/wiki/Using\\_OpenStreetMap\\_offline](https://wiki.openstreetmap.org/wiki/Using_OpenStreetMap_offline)

### 7) Wish List for OSM

It was suggested by user kp623, We need something like waze to add traffic lights and signs for OSM so if you are app developer, Please feel free to contact me. For now to map everything in general you will have to use OSM website or you can use one of the following editors.

<https://wiki.openstreetmap.org/wiki/Editors>

### 8) Background

I wrote full post regarding the project at [opc](#) where i did mention most of the details from here but it was project that was needed for me to work on next with help of community of course like anything else i have been involved in.

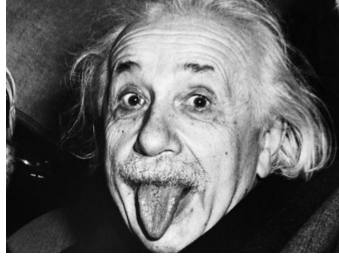
- ★ Please note this project won't be upstreamed due to safety reasons however it will be a gimmick that we all can enjoy soon with help of community members.



## 9) Conclusion

I know it won't be a easy task for traffic lights but stop signs are different situation. Like they say anything is possible when you put your mind and soul into it. Please do forgive me if i have made any grammar errors, English is not my first language. Following are community members who are going to be part of the project so far:)

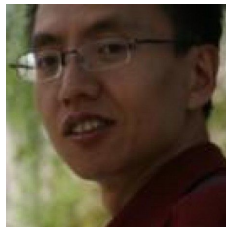
1. Mad genius who brought us auto lane change who i call albert einstein  
@BogGyver



2. The king of gimmicks from Germany @arneschwarck



3. Last one is special i call him the real life Iron man his name is @Alex Wang  
He first brought the OP with flexray driver then visiond on pc and now he will helping with this project running model along with driver model:)



4. And me of course the guy with bat wings who got the plan and vision to begin something awesome.



## 10) Work in progress:)

**Big thanks to Alex who has already started the model training. Following is teaser.**

### 1. First Teaser



### 2. Localization Test with OSM

