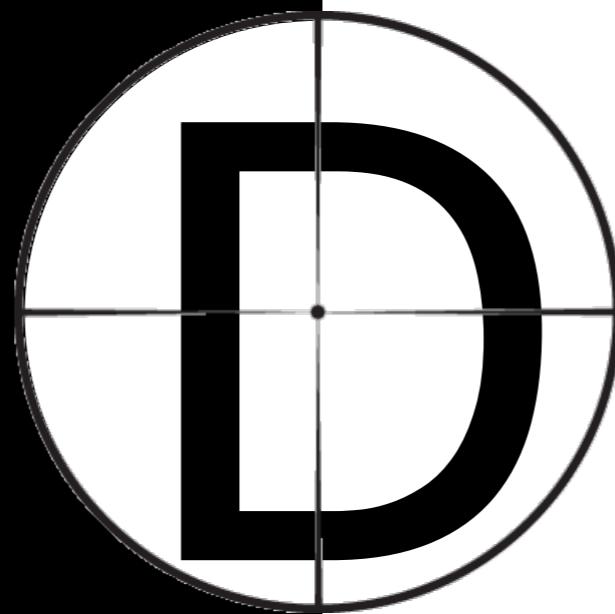


OBJECT



DETECTION

GROUP



BACKGROUND

-
-
-

Potential of self-driving car

Need robust perception systems

Deep learning (specifically CNNs) advances



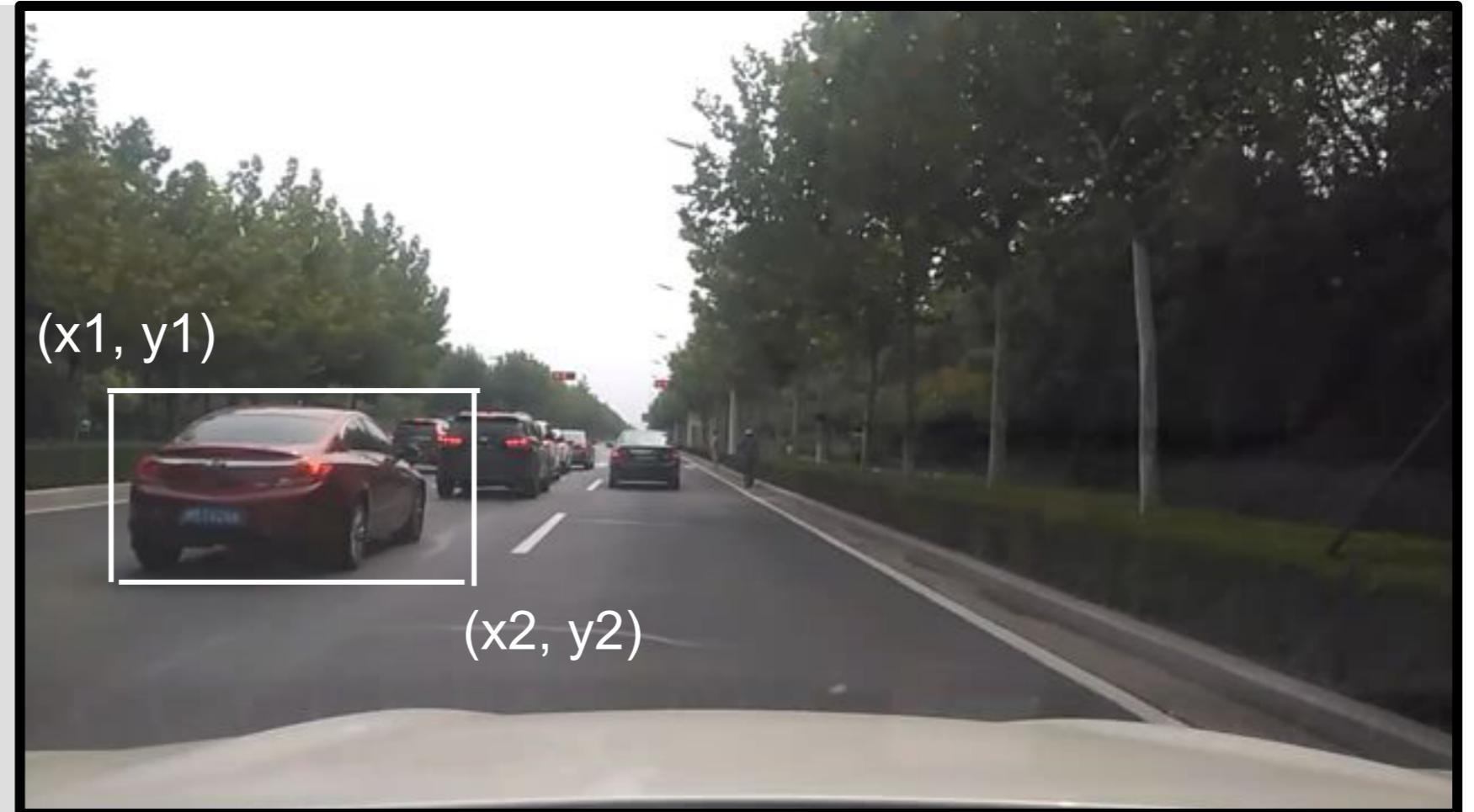
DATA

Dataset	Source	Object	Quantity
1	UCar Technology Inc.	car, human, cyclist and traffic lights	10,000
2	KITTI: Object Detection Evaluation 2012	car, human	7,481

Download data: UCar DATA: [training set](#); [test set](#); [KITTI](#)

TRAINING DATA-EXAMPLE

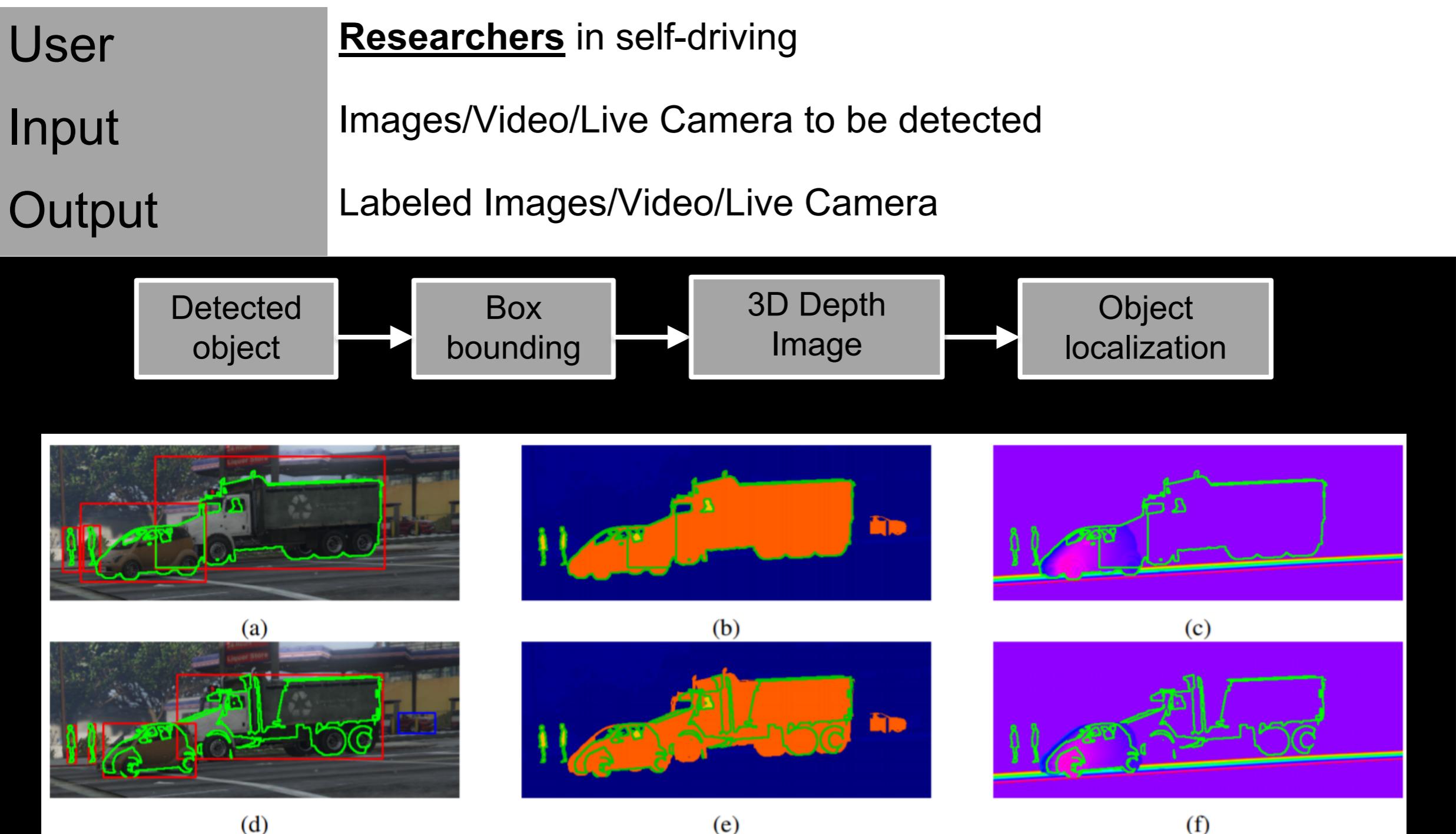
Image



**Label Map
(bounding box +
class name)**

```
{"61213.jpg": [  
    [319.00032, 185.50008, 329.83296, 217.33344, 2],  
    [258.99968, 187.99991999999997, 296.49984, 218.00016, 1],  
    [182.83328, 179.49996000000002, 237.16672, 222.5001600000002, 1],  
    [162.0, 181.9998, 199.00032, 209.00016, 1],  
    [46.0, 176.3334, 182.0, 252.8334, 1],  
    [232.83328, 188.1666, 257.00032, 207.8334, 1],  
    [233.49952, 161.25012, 248.7500800000003, 169.0002, 20],  
    [305.49952, 187.25004, 312.49984, 208.75032, 2]]}
```

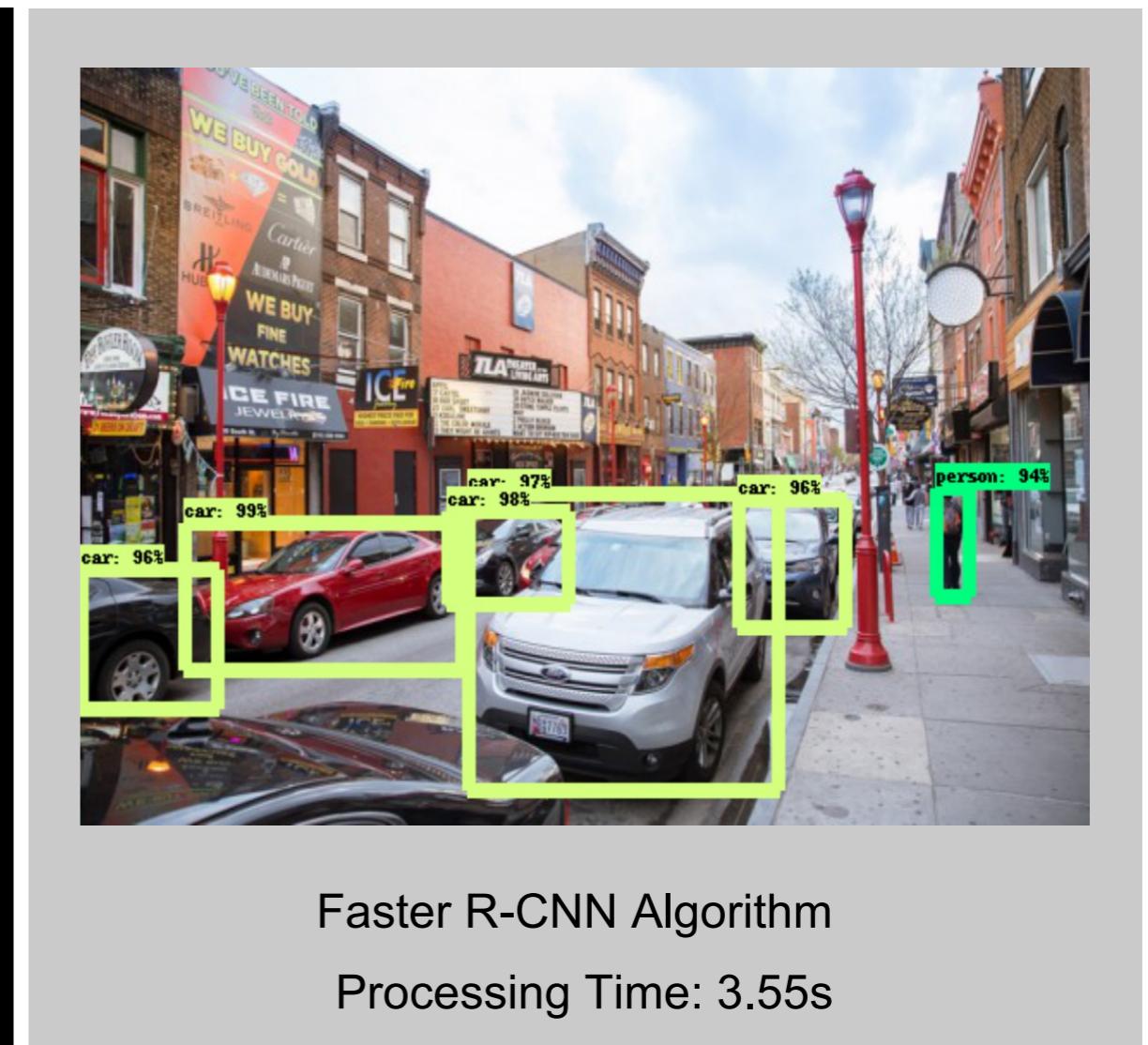
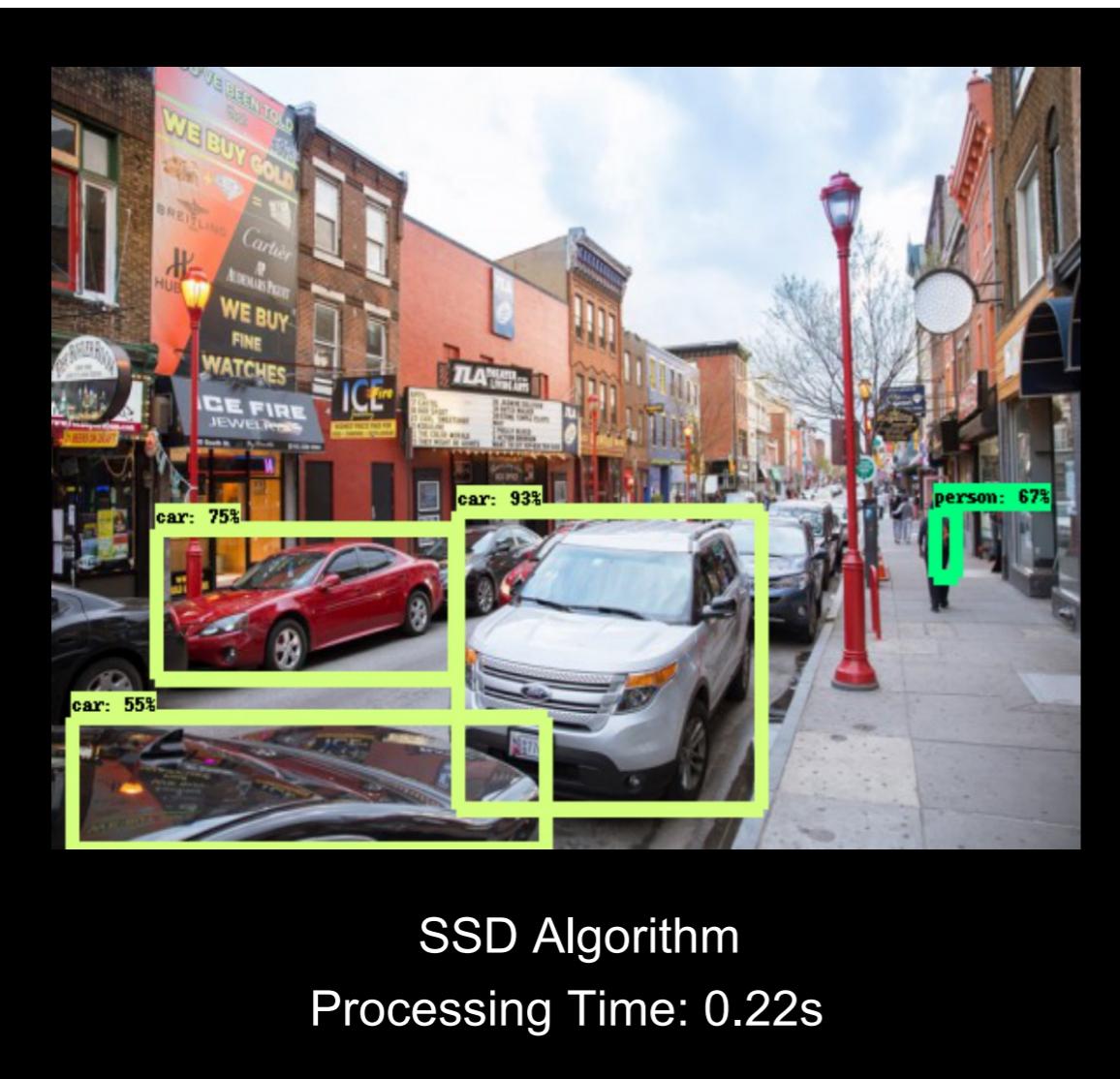
USE CASE1: Self-driving detection



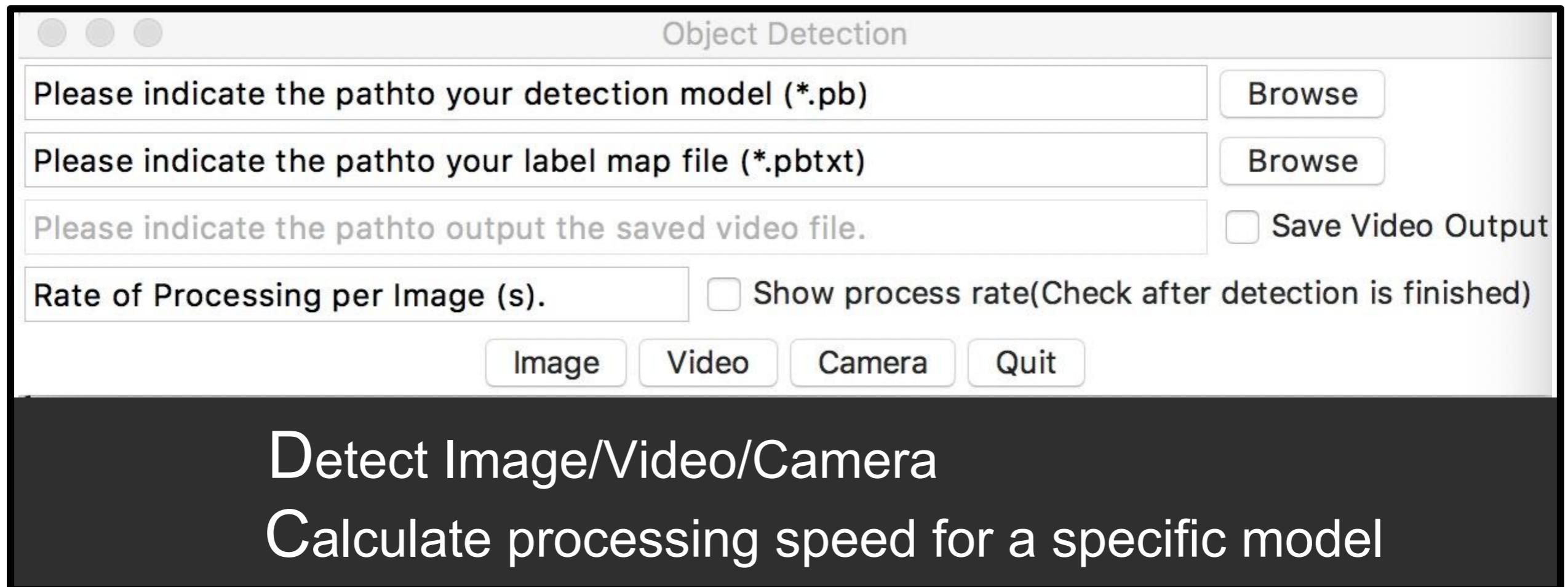
Driving in the Matrix: Can Virtual Worlds Replace Human-Generated Annotations for Real World Tasks?
(M. Johnson-Roberson, Charles Barto, Rounak Mehta, Sharath Nittur Sridhar, Karl Rosaen, Ram Vasudevan), *In IEEE International Conference on Robotics and Automation, 2017.*

USE CASE2: Compare different algorithm

User	<u>Researchers</u> in self-driving
Input	Different Object Detection Models, Photos to be detected, LabelMaps
Output	Labeled Images

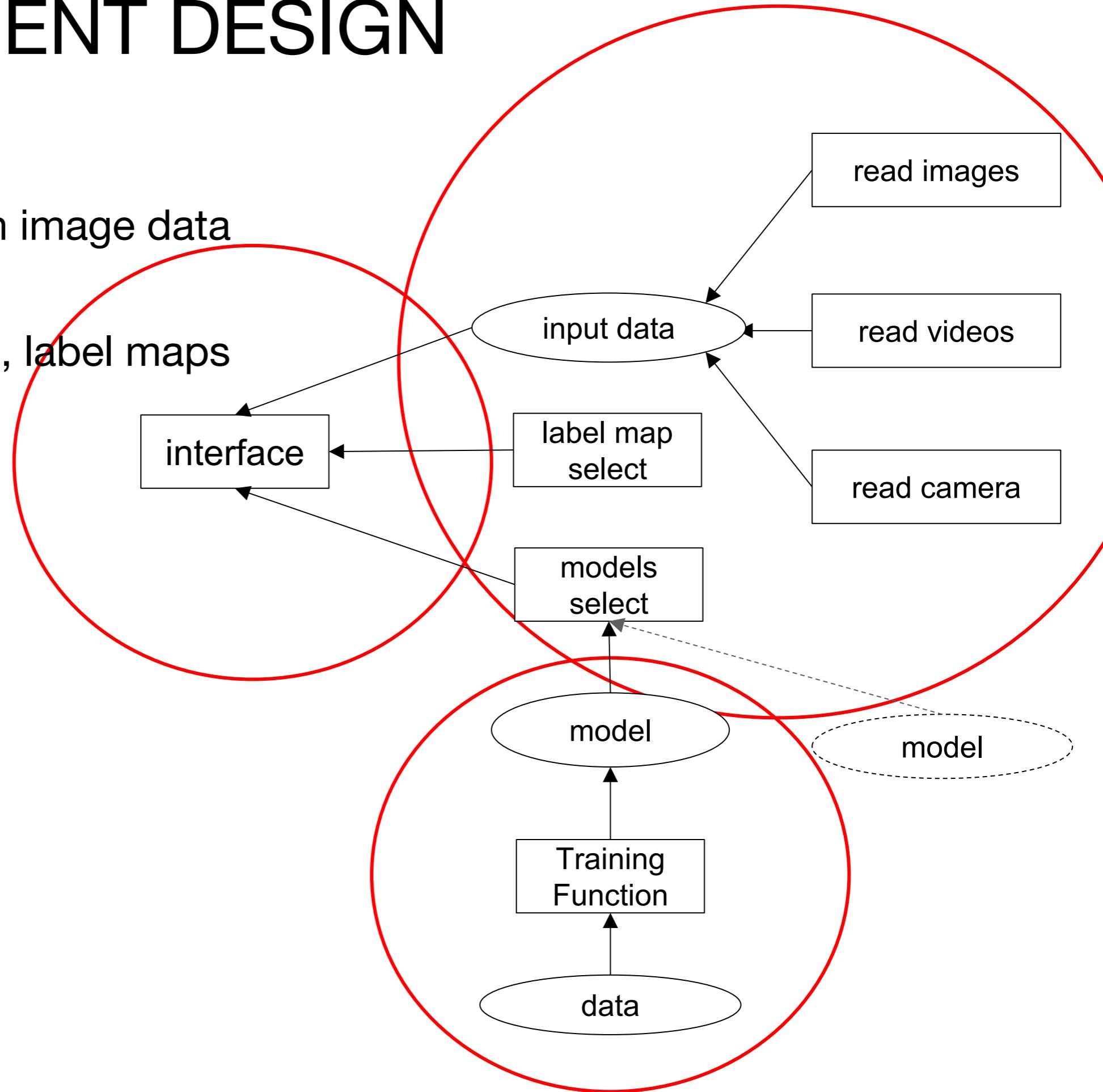


DEMO



COMPONENT DESIGN

- **Train a model**
 - Train the model with image data
- **Interface**
 - Load detect models, label maps
 - Read in inputs
 - Show result

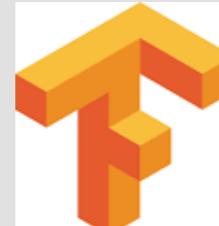


PROJECT STRUCTURE

```
└── Doc/  
└── examples/  
└── objectdetection/  
    ├── data/  
    ├── ssd_mobilenet/  
    └── submodule/  
    └── tests/  
└── README.md  
└── LICENSE  
└── setup.py
```

Doc
example
objectdetection
.gitignore
1.png
LICENSE
README.md
setup.py

LESSON LEARNED

- Design and implemented a complete software (detection algorithm, component, interface, test...)
- Git version control
-     (Tkinter)
- Dealing with different data format (images, json, csv, pbtxt , camera input, tensorflow tf record, etc)

FUTURE WORK

- Further tune hyperparameters in Faster R-CNN Model
- Upgrade user interface
- Add input data from other sensors (radar, lidar) to make more robust detection.

Thank you!

<https://github.com/UWSEDS-aut17/uwseds-group-zero>

