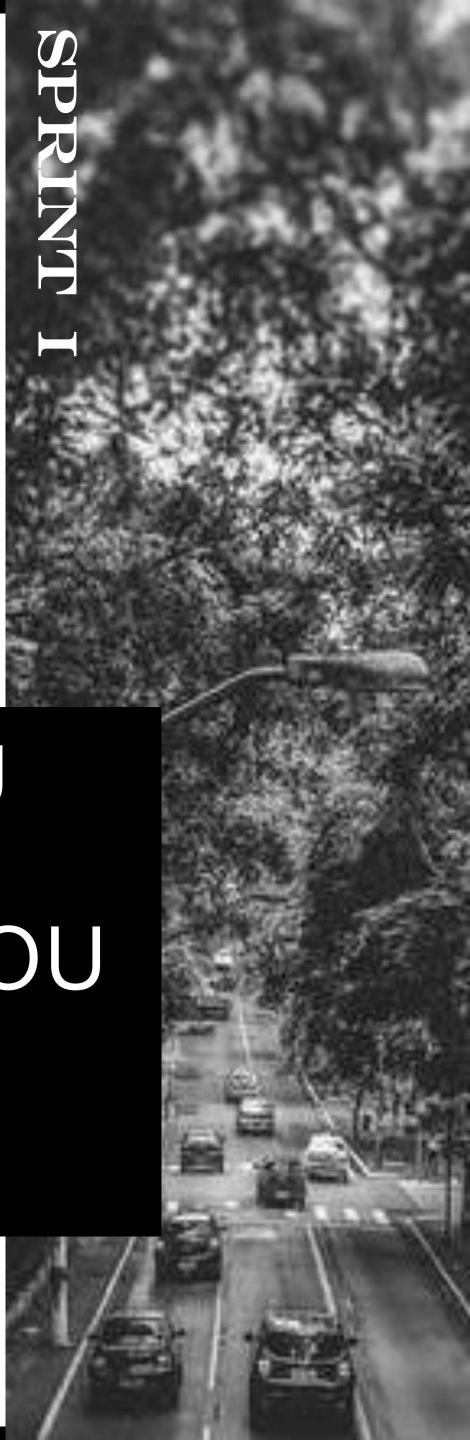
LYFT 3D OBJECT DETECTION FOR AUTONOMOUS VEHICLES

FENGXU TU
ZIYU ZHAO
YUHAO ZHOU

TEAM



SPRINT I

PRODUCT DEFINITION

The statement of the meaning of the product

PRODUCT MISSION

Find out the classes and positions of objects on the road

- Sensor data from 7 cams and 3 LiDARs
- Calibrations and Geographic data are available as well
- Using these data to find out the 3D volume (box) of every object and classifying the categories of objects

USER STORIES

How does the user think

Lyft, who is the sponsor of the competition, as a ridesharing company, want to use this algorithm in their autonomous vehicles system.

TARGET USERS

Who will use our product

Ridesharing Companies

- Lyft, as the sponsor of the Kaggle Competition
- Uber, who uses LiDAR data as well in their system
- Other Ridersharing Companies who will be interested in autonomous vehicles.

Startups Companies

- Since our works are open source, it could inspire some startups
- Even the data and parameters in the model could cause copyright problem, the algorithm and model themselves are free to use

Other Researchers

- We will open source our code and basically every document. So it is more than welcome to check the code and algorithm out

MINIMUM VIABLE

At least, we need …

Algorithms and Models

- The results from our method have to be at least better than from white noise

Implementations

- A fully functional implementation of our algorithm

Documentations

- A report about our method and the working theory
- Docs to help people to build and use our product and code

PRODUCT ANALYSIS

Detailed examination of the product

SIMILAR PRODUCTS

What already exist

Waymo

- Google's self-driving car project
- Strong, powerful, but only dataset is available.

Autopilot

- Tesla's self-driving car system
- Already in use as a commercial application.

Others

- There are a lot of open source framework in the field of self-driving cars for us to study.
- Like Apollo Auto, etc.

PATENT RELATED

Is there any limitation

TOOLS

- Programming language
- Widely used library like numpy, OpenCV
- We only choose open source tools which are under liberal license

ALGORITHMS

- The widely used ML and DL algorithms like UNet, GCN, GAN don’t have a patent problem in our use case.

ALGORITHM DESIGN

Thoughts behind the algorithm

MAJOR COMPONENTS

The major submodules of the algorithm

3D Projection

- Lyft, as the sponsor of the Kaggle Competition
- Uber, who uses LiDAR data as well in their system
- Other Ridersharing Companies who will be interested in autonomous vehicles.

Classification

- Lyft, as the sponsor of the Kaggle Competition
- Uber, who uses LiDAR data as well in their system
- Other Ridersharing Companies who will be interested in autonomous vehicles.

Objec Detection

- Lyft, as the sponsor of the Kaggle Competition
- Uber, who uses LiDAR data as well in their system
- Other Ridersharing Companies who will be interested in autonomous vehicles.

TECHNOLOGY SELECTION

Which technology is the best, to us

Languages

- Python
- All the team members are familiar with Python
- The ecosystem (community) of Python is better than others in this field
- The native performance of Python is bad. But most of the performance sensitive works in our code are just a sort of DSL. The framework will handle the performance problems.

DL Frameworks

- Since DL is one of the most necessary and important part in our algorithm, we have to carefully choose DL frameworks.
- Keras, to run simple model tests and demos.
- PyTorch, to complete more complicate DL task

Other Libs

- Classic ML lib: scikit-learn
- Data Utils:
  - Official: nuscenes-devkit
  - 3rd-Party: pandas, numpy, scipy
- CV lib: OpenCV (cv2 binding for Python)
- Viz lib: seaborn, matplotlib
- Others: some scripts from GitHub

TEST AND VALIDATION

How to evaluate our work

The Kaggle Competition has provided the way they evaluate the algorithm.

Threshold

- Intersection over Union (IoU):

$$IoU(A, B) = \frac{A \cap B}{A \cup B}$$

Average Precision

$$\frac{1}{|thresholds|} \sum_{t} \frac{TP(t)}{TP(t) + FP(t) + FN(t)}$$

THANK YOU