



# Parking Spot Number Detection Using Fisheye Camera

**Sponsor:** General Motors (GM)

**Team Member:** Yuanli Zhu

**Course Instructor:** Dr. Jigang Wu

Guoxin Yin

**Group ID:** 1

Weihao Xia

**Date:** July 15th, 2021

Shengan Zhang

Yixuan Wang

# About Us



Name	Yuanli Zhu	Guoxin Yin	Weihao Xia	Shengan Zhang	Yixuan Wang
Major	ECE+CS	ECE+CS	ECE+DS	ECE+CS	ECE+CS
Email	18901990109@sjtu.edu.cn	yingguoxin2017@sjtu.edu.cn	anthony12590@sjtu.edu.cn	jacob_zsa@sjtu.edu.cn	joy_wang@sjtu.edu.cn

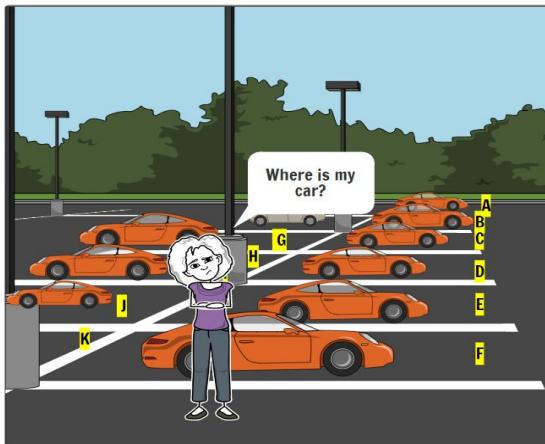
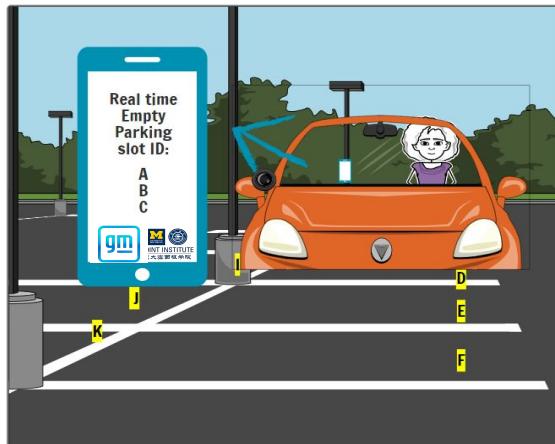
# Content

- Introduction
  - Problem Definition & Project Goal
  - Customer Requirements & Engineering Specifications
- Concept Generation & Selection
- System Implementation
  - Car + Parking Slot
  - Algorithm - Parking Spot Number Detection
  - App - Display Results
- Validation Result
- Conclusion & Future Work

# Problem Definition & Project Goal



Record parking spot ID in real time



Reference:

Use characters from <https://www.storyboardthat.com/storyboard-creator>

# Customer Requirements and Engineering Specifications

CR:

- Record parking spot ID
- Real-time response

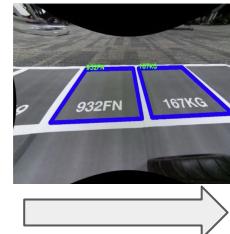
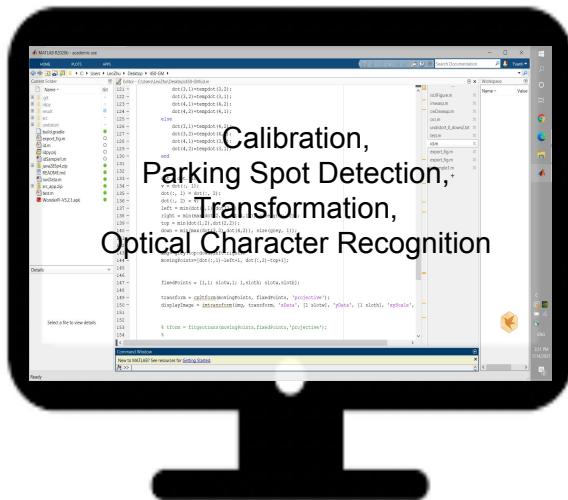
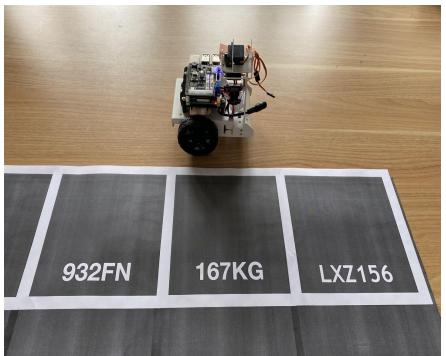
ES:

- |                 |              |
|-----------------|--------------|
| • Accuracy      | > 90%        |
| • Speed         | 0.1-1s/frame |
| • Field of View | 170°         |
| • ID Length     | 1-10         |
| • Characters    | a-z,0-9,A-Z  |

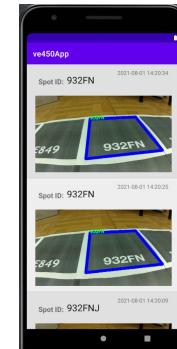
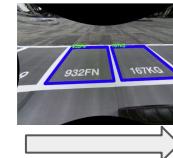
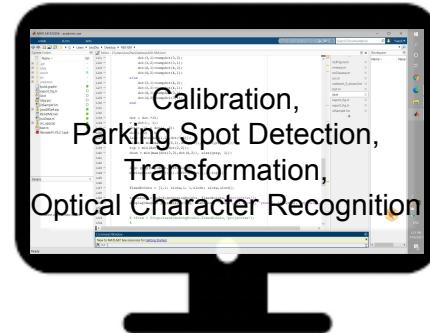
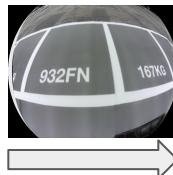
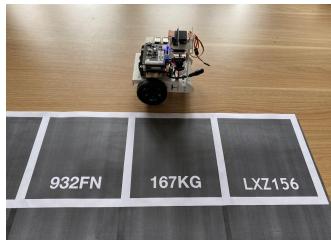


JOINT INSTITUTE  
交大密西根学院

# Flowchart and Concept Generation



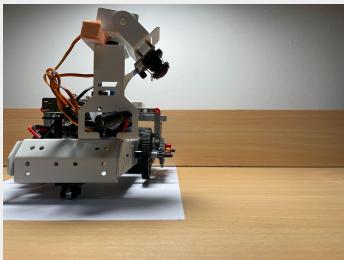
# Concept Generation and Selection



	<b>Hardware</b>	<b>Algorithm</b>	<b>App</b>
Method 1	Real car + real parking lot	Retrain OCR	✓ Android
Method 2	✓ Small car + simulated parking lot	✓ Preprocess + OCR	IOS
Reason	(Economical + flexible + easier demo)	(Higher accuracy +faster + less input dataset)	(Wider development)

# Preparation Overview

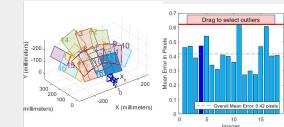
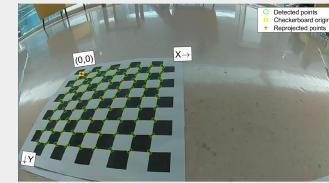
Hardware



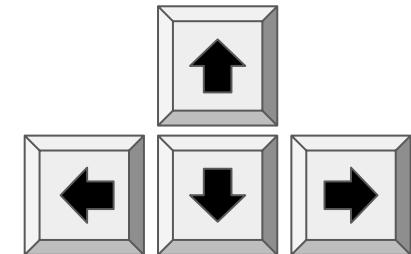
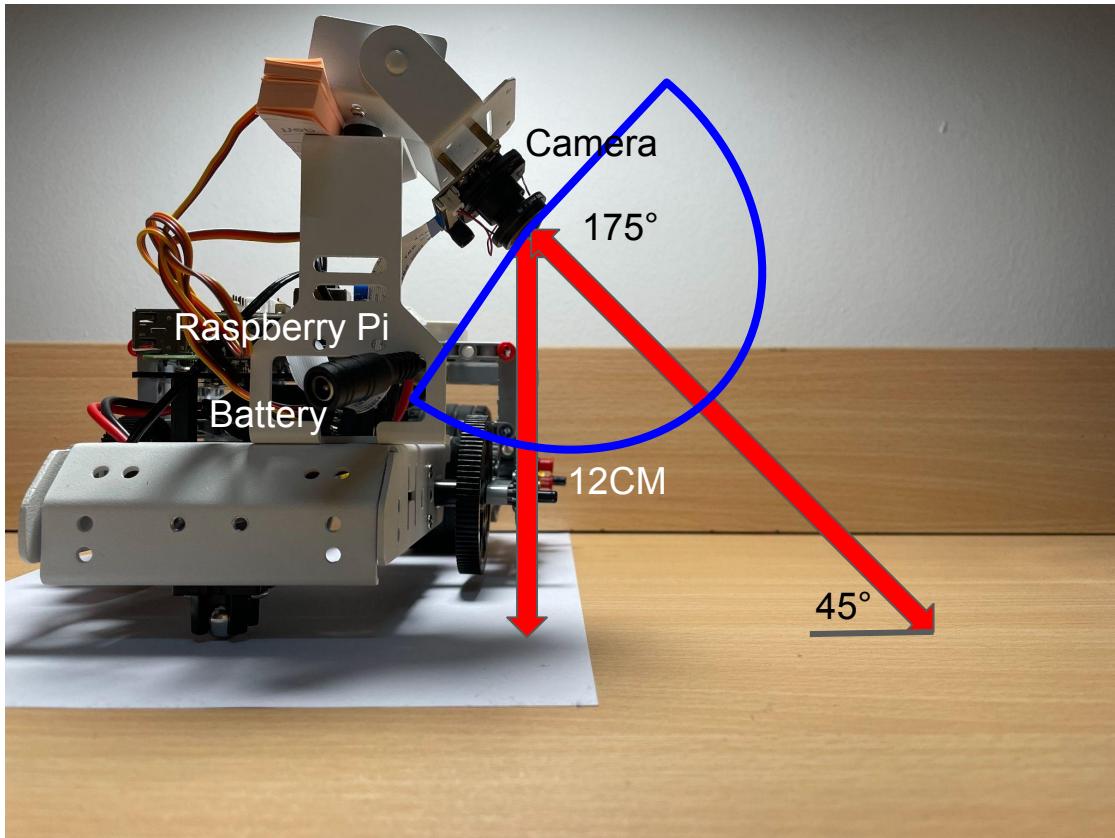
Simulation  
Parking Lot



Calibration  
Parameters

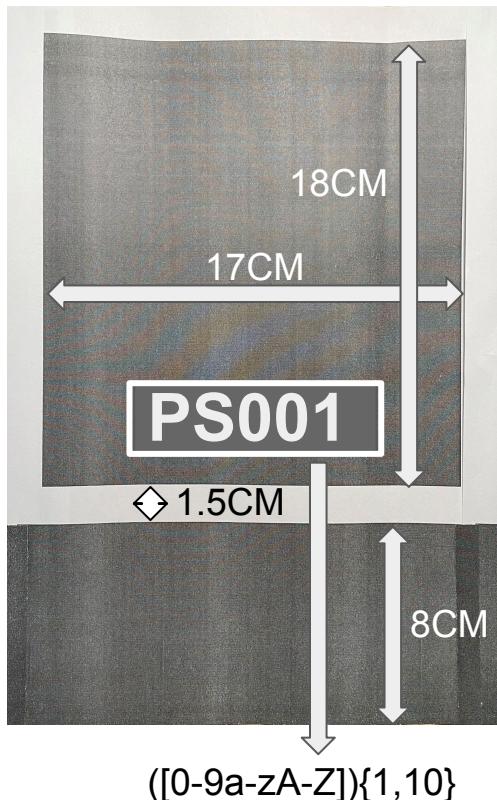


# Hardware

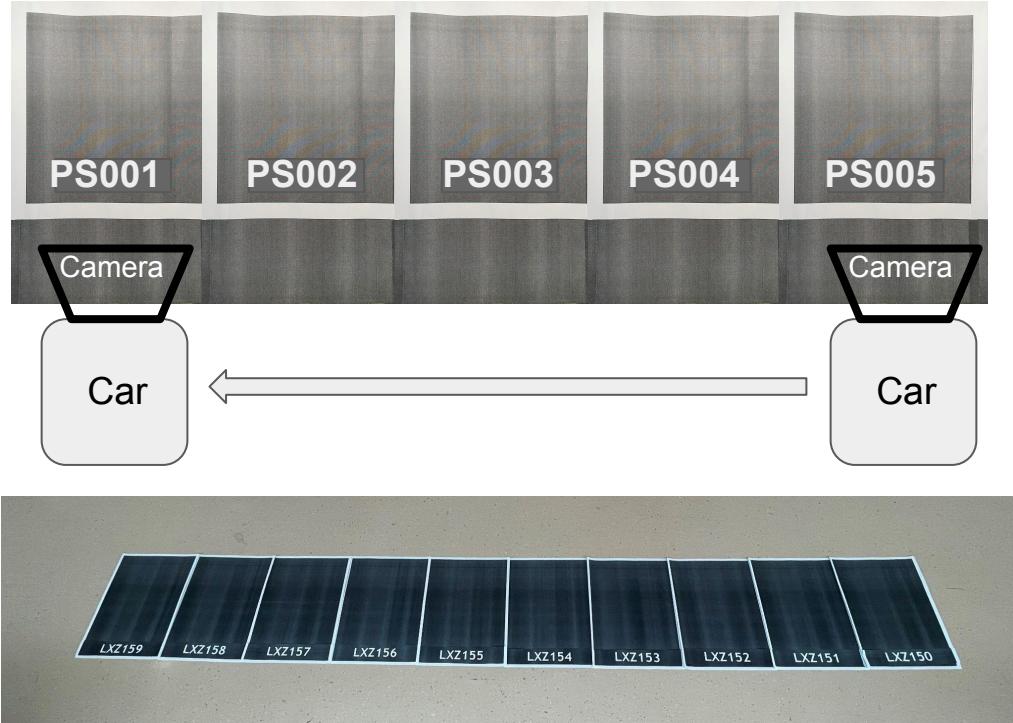


# Simulation Parking Lot

A basic spot unit



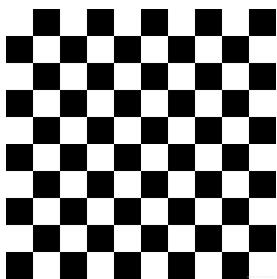
Consecutive spot units



# Calibration Parameters



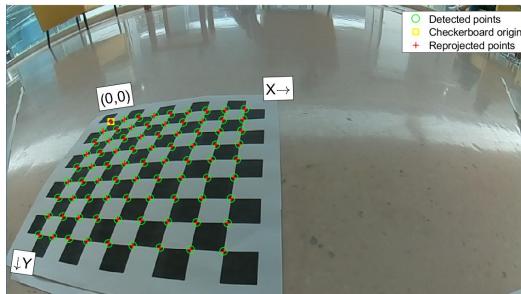
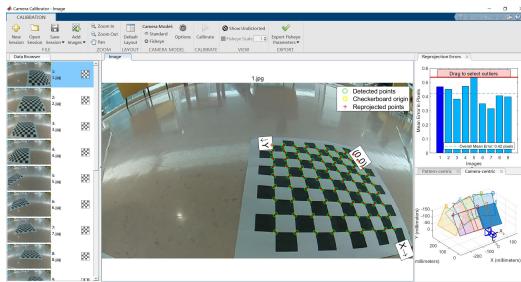
JOINT INSTITUTE  
交大密西根学院



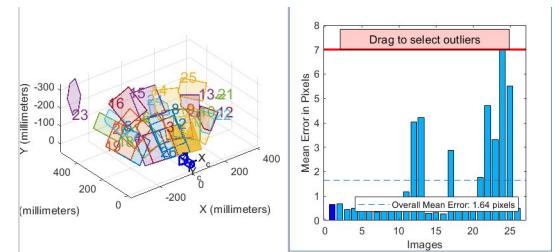
10x10  
checkerboard



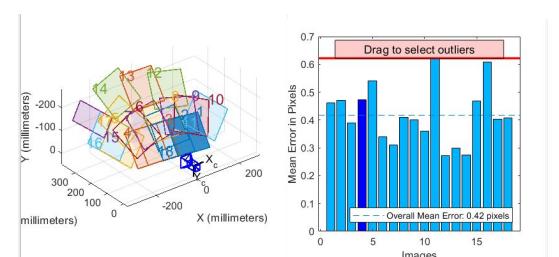
Calibration dataset



Calculate parameters



Before



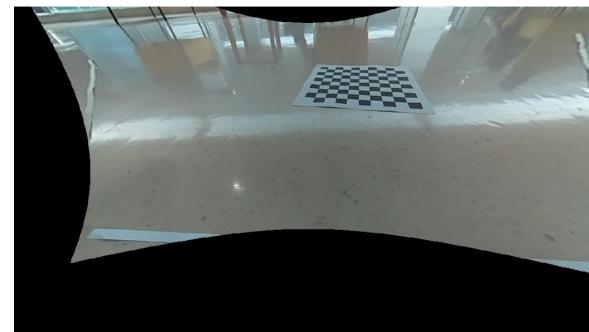
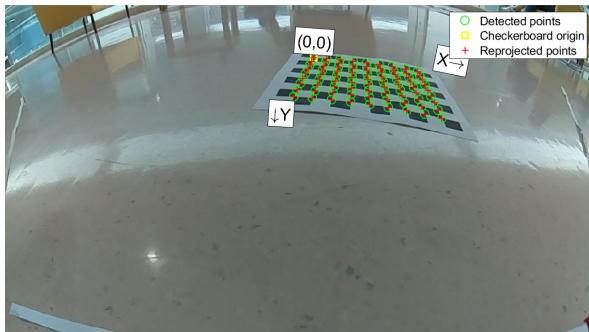
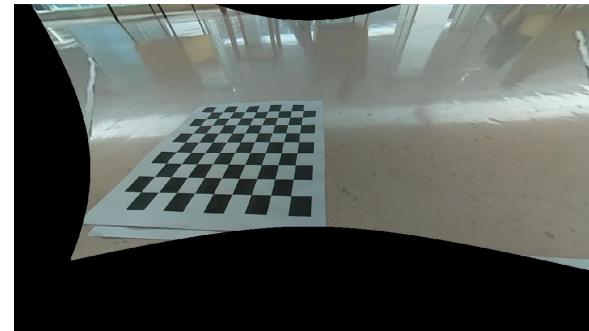
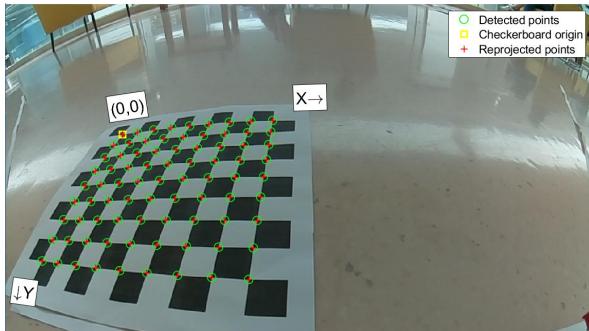
After

Optimize parameters



JOINT INSTITUTE  
交大密西根学院

# Photo calibration



# Algorithm Overview



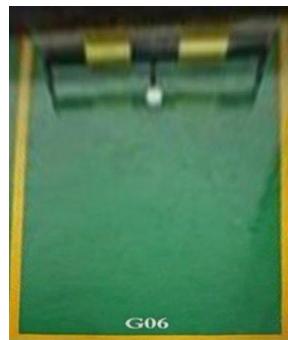
Parking Spot  
Detection



Projective  
Transformation



Optical Character  
Recognition (OCR)



(Matlab imtransform  
function [1])

[1] imtransform <https://www.mathworks.com/help/images/ref/imtransform.html>

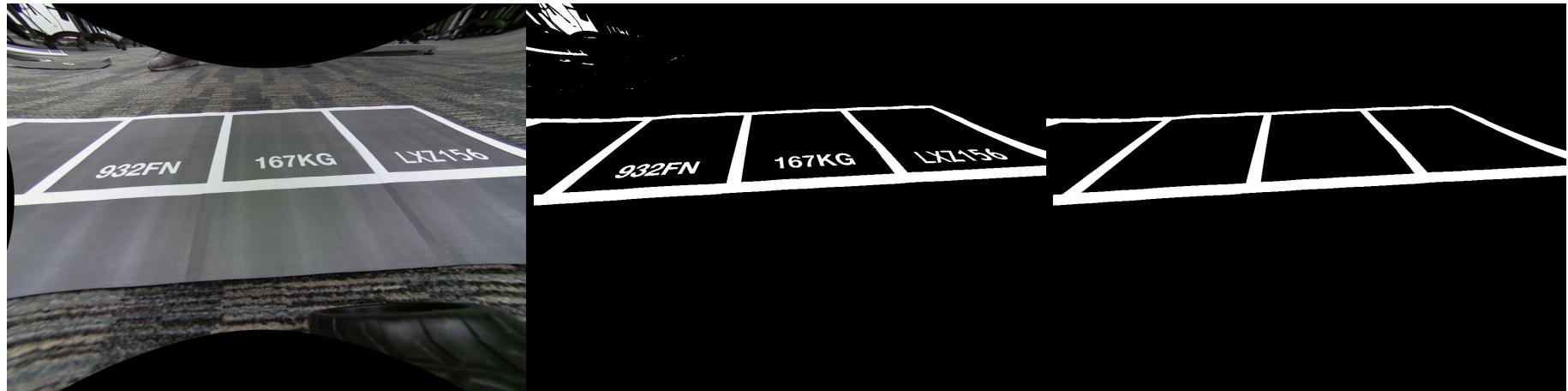
[2] OCR <https://www.mathworks.com/help/vision/ref/ocr.html>



JOINT INSTITUTE  
交大密西根学院

# Parking Spot Detection 1

Parking spot lines have to be continuous



Grey Image



Binary Image



Eliminate Distraction

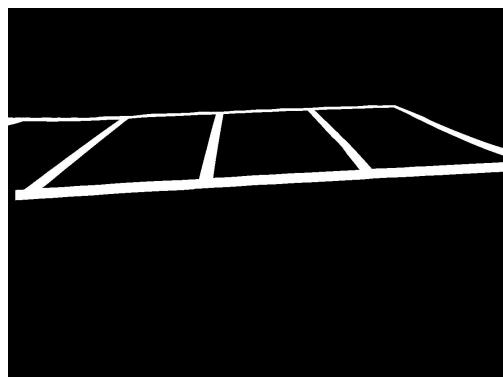
Threshold: 0.7  
(1: white 0: black)

Limitation: Cannot eliminate  
too large white area



JOINT INSTITUTE  
交大密西根学院

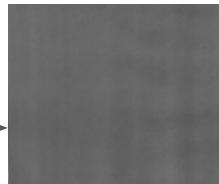
# Parking Spot Detection 1



Separation+  
Corner Detection



Projective  
Transformation



LXZ156



932FN



167KG

Limitation: Extracted slot line has to be continuous

# Parking Spot Detection 2

No long lines in the image



Parking spot with  
non-continuous line

Edge Detection +  
Hough Transformation  
(Detect Line segment)

Line Segment Grouping +  
Linear Fitting



# Backend Server

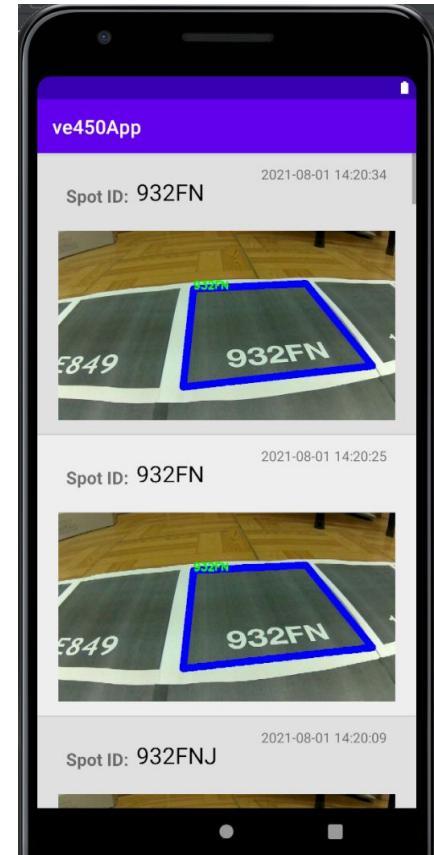
- Database: postgresql

Column	Type	Collation	Nullable	Default
id	character varying(255)			
time	timestamp with time zone			CURRENT_TIMESTAMP
imageurl	text			
newimage	text			

- Web framework: Django + nginx + gunicorn
  - POST:
    - pass the spot ID, the recognized images with parking spot lines highlighted
    - Auto record the current timestamp
  - GET: retrieve the information from the backend database

# App Design

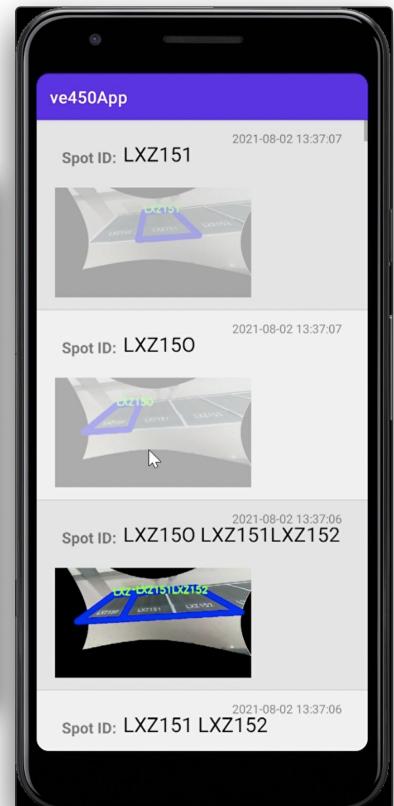
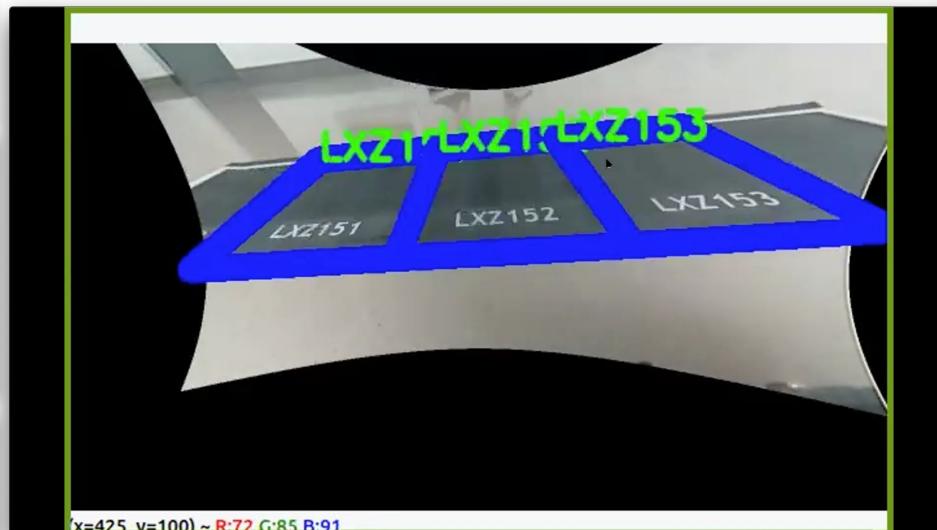
- Platform: Android
  - Development could be done on Windows, macOS and Linux
  - Emulator is better than IOS development.
- UI:
  - Parking Spot ID
  - Timestamp
  - Recognized Image
- Feature:
  - Auto refresh the page every 0.5s
  - Manually refresh the page by a poll-down
  - Show all the captured results if scrolling down the page



# Validation Test



JOINT INSTITUTE  
交大密西根学院





JOINT INSTITUTE  
交大密西根学院



JOINT INSTITUTE  
交大密西根学院

# Test Results

- Accuracy:
  - Accuracy per character: 0.9398
  - Missing rate: 0
- Processing Speed:



Engineering Specifications:

- ✓ Accuracy > 90%
- ✓ Processing Speed 0.1-1 s/frame

	Shoot	Calibration	Algorithm	Send to server	Whole pipeline
Avg time (s)/ frame	0.1887	0.0565	0.1536	0.0917	<b>0.3297</b>
Max time (s)/ frame	0.50	0.28	0.3377	0.3485	<b>0.6128</b>

# Conclusion & Future Work

- High accuracy (**94%** > **90%** in ES)
- Processing rate (**0.33 second per frame** < **1s** in ES)
- Traditional computer vision + pretrained model  
=> fine-tuned models on parking slot dataset

# Thank you!

# Q & A