A Deep Learning based Unoccupied Parking Space Detection Method for City Lots

Hamid Reza Tohidypour, Yixiao Wang, Panos Nasiopoulos, and Mahsa T. Pourazad

The University of British Columbia, Canada
TELUS Communications Inc., Canada
Contact email: htohidyp@ece.ubc.ca

Research

- Light Field (LF) Compression
- LF Refocusing
- LF Quality Metrics
- LF view synthesis

Education

- Ph.D. 2018-present; UBC
- M.S. 2013-2015; DU
- B.Sc. 2009-2013; DU

Work

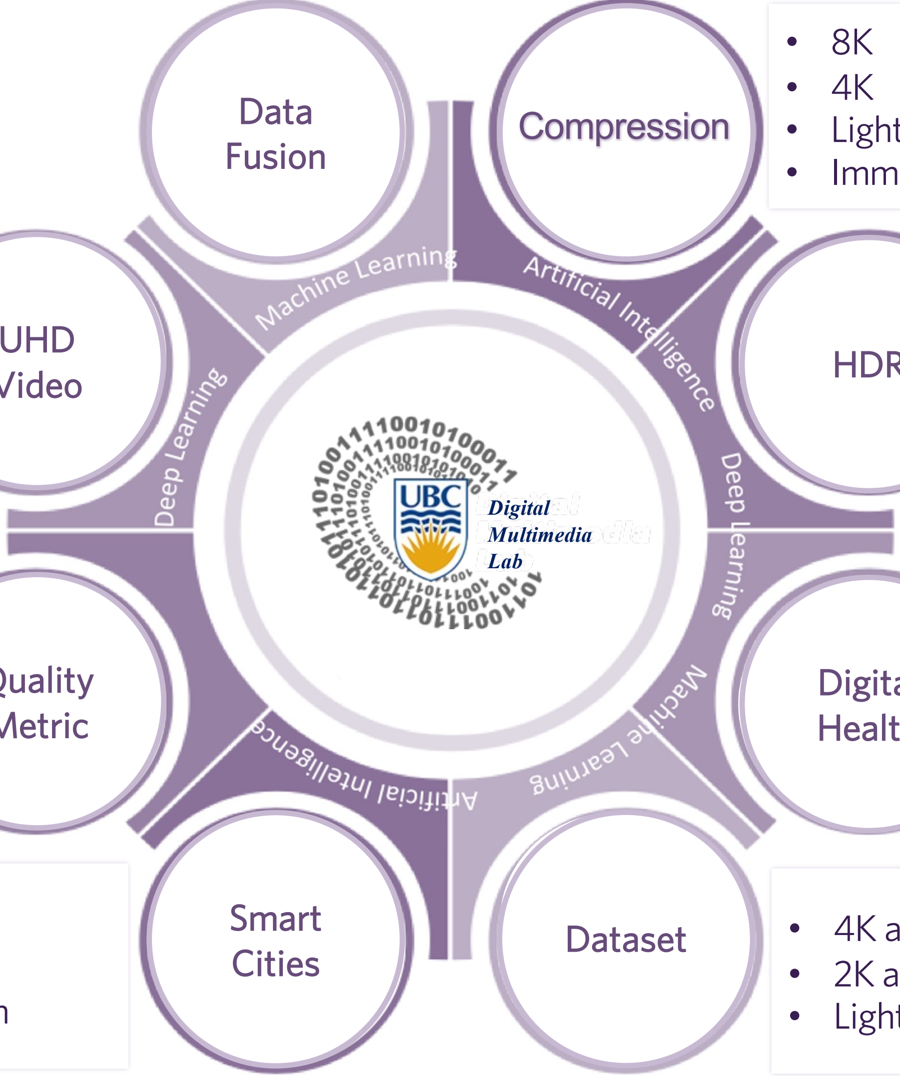
- Research Assistant; DML
- Teaching Assistant; UBC
- Lecturer; DU; BRAC U

Skills

- Deep Learning
- UI design and front-end development
- Software development

Interests

- Internet
- Movies
- Travelling
- Cooking



- 8K
- 4K
- Light Field
- Immersive video

- Tone Mapping
- Inverse Tone Mapping

- Fall Detection
- Medical Image processing

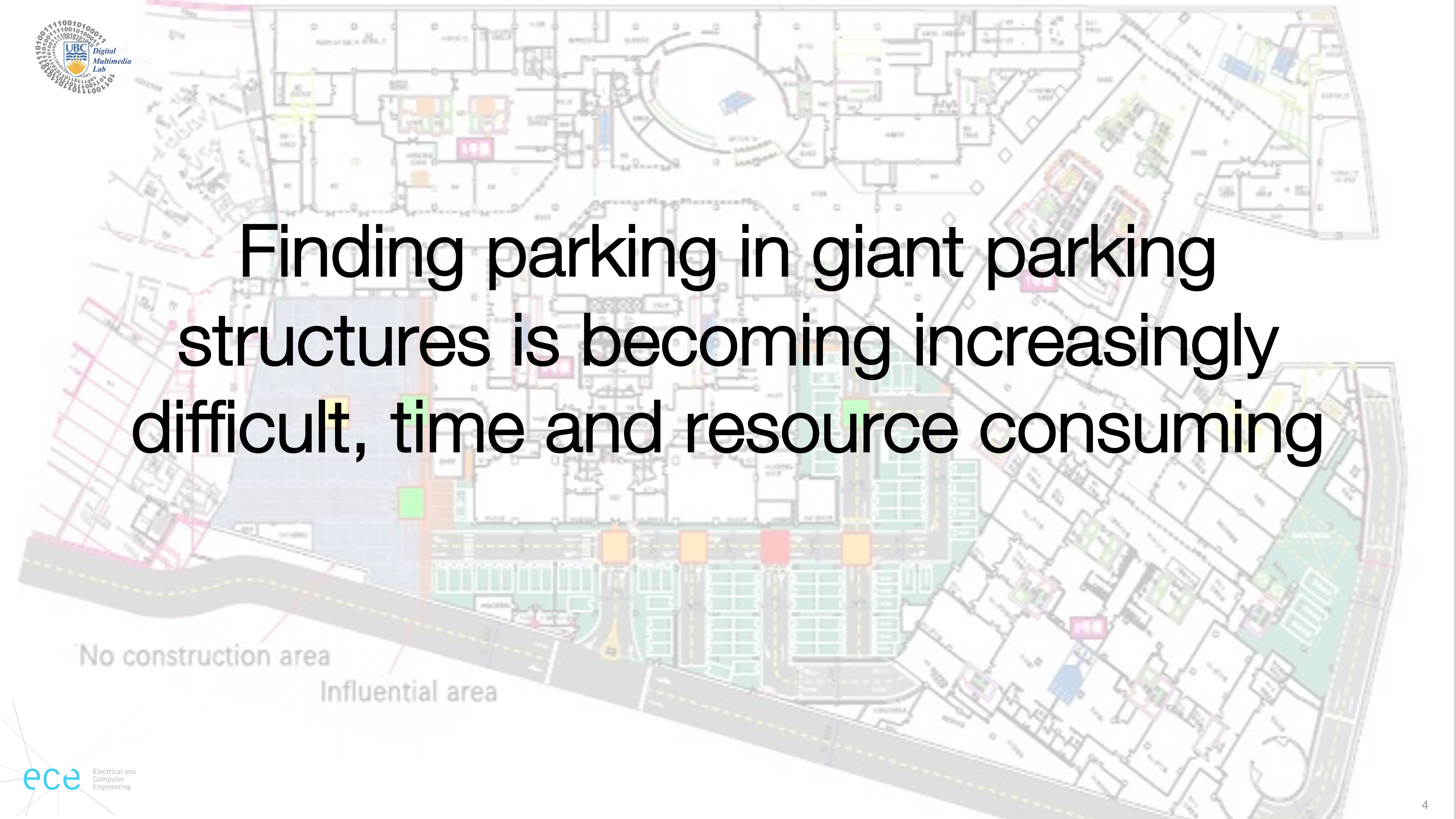
- 4K and 8K pairs
- 2K and MOS pairs
- Light field

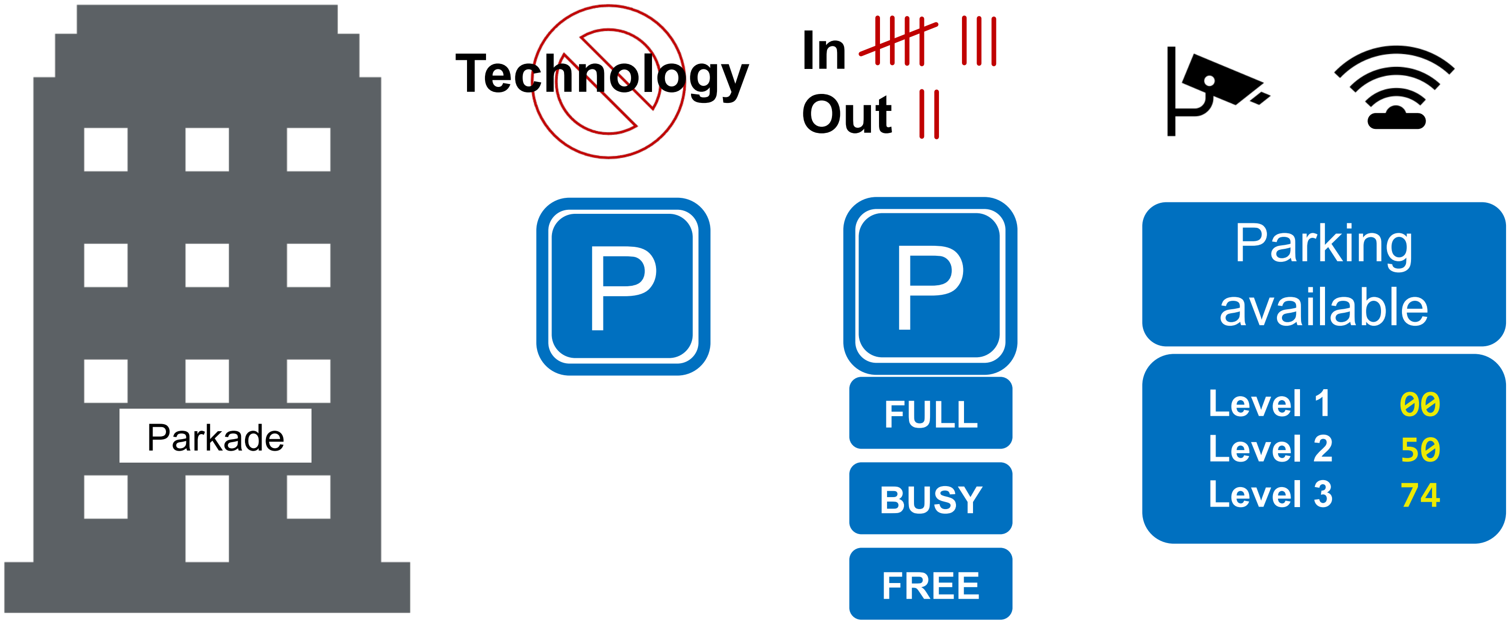
- Street Signs
- Street Parking
- Road Deterioration

- DL based video quality
- LF spatial and angular quality

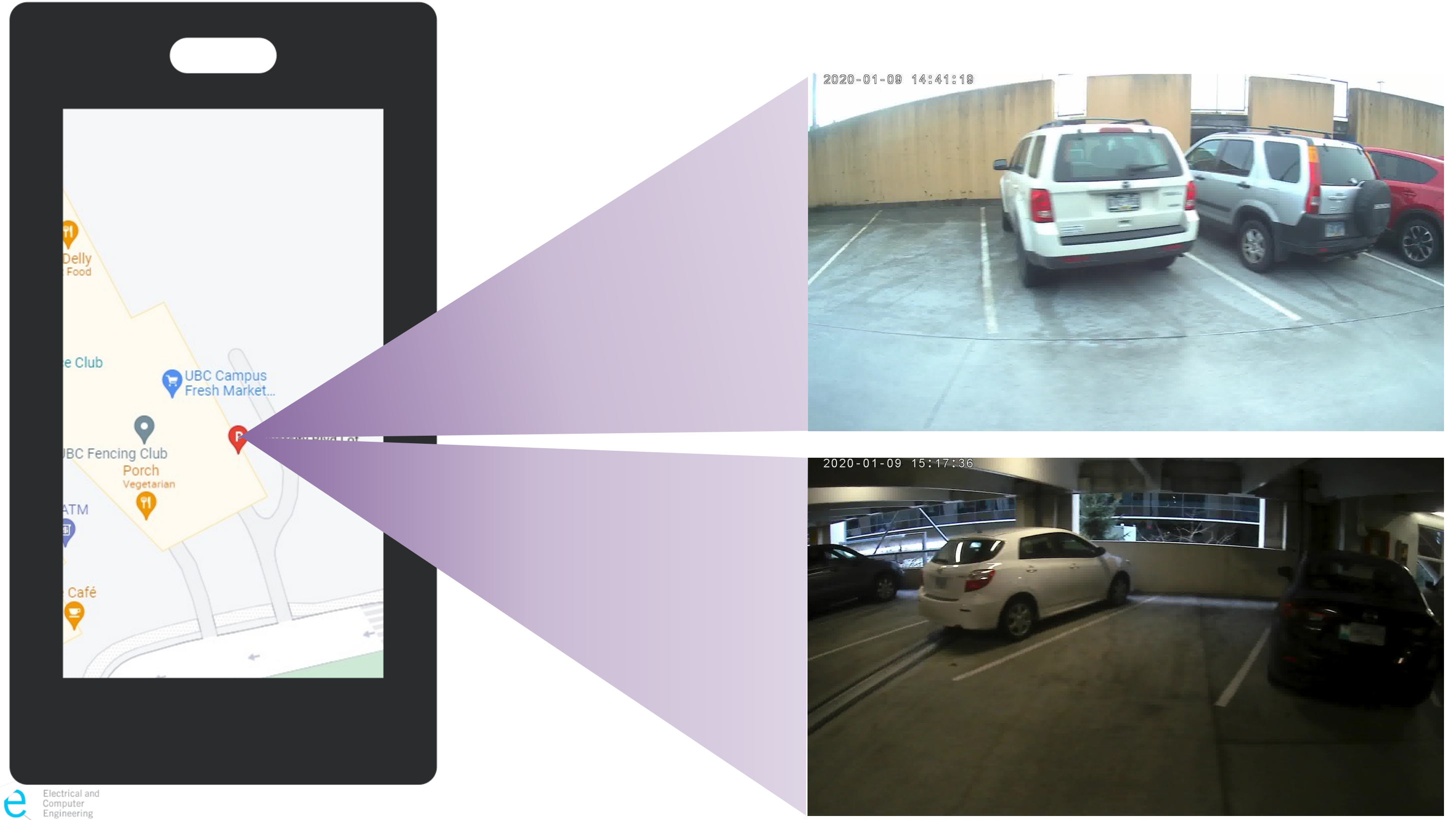
- 4K, 8K, LF capture
- Calibration
- Rendering

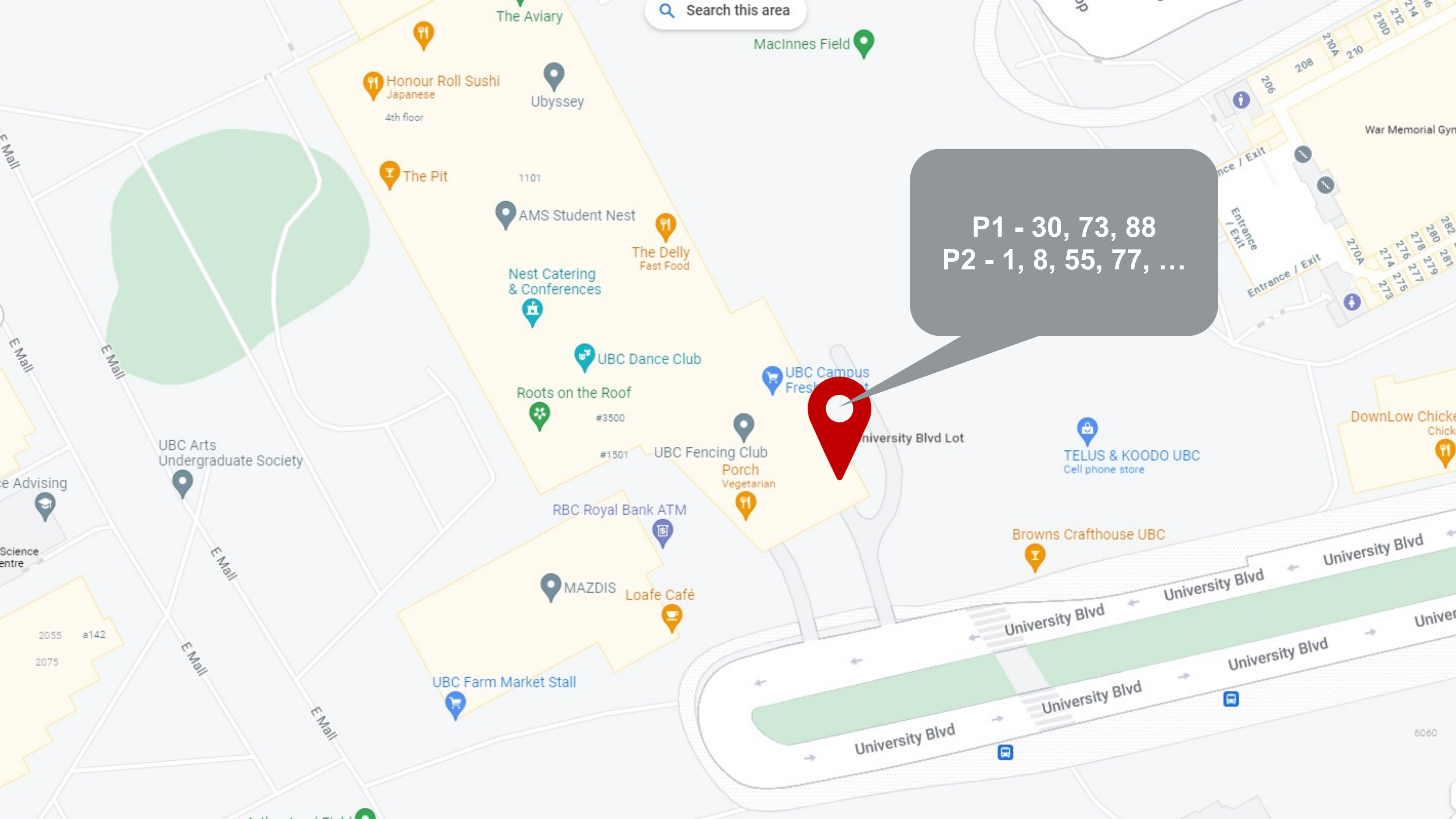
Finding parking in giant parking structures is becoming increasingly difficult, time and resource consuming



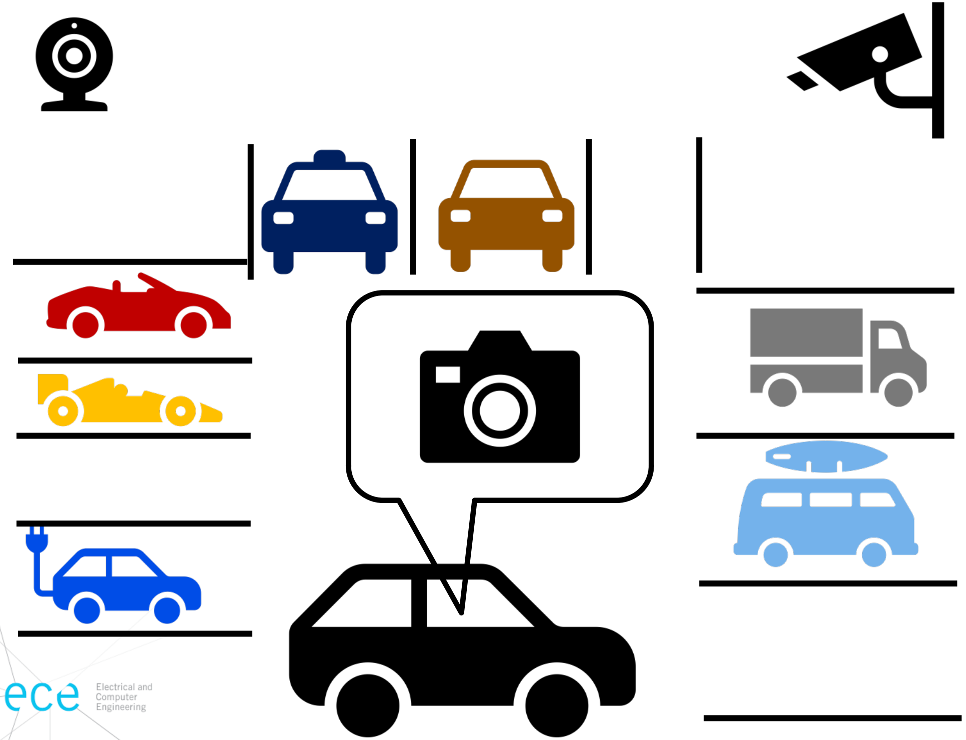

Parkade
Technology
In
Out
P
P
FULL
BUSY
FREE
Parking available
Level 1 00
Level 2 50
Level 3 74

Properly managing **parking lots** and displaying available **parking information** to the **drivers before entering** the parking lot has been a challenge.

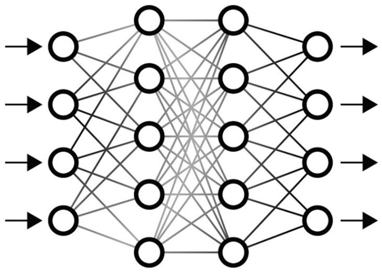


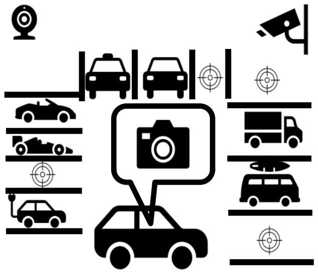

P1 - 30, 73, 88
P2 - 1, 8, 55, 77, …
Search this area
The Aviary
MacInnes Field
Honour Roll Sushi
Japanese
4th floor
Ubyssey
The Pit
1101
AMS Student Nest
The Delly
Fast Food
Nest Catering
& Conferences
UBC Dance Club
Roots on the Roof
#3500
UBC Campus
#1501
UBC Fencing Club
Porch
Vegetarian
University Blvd Lot
TELUS & KOODO UBC
Cell phone store
RBC Royal Bank ATM
Browns Crafthouse UBC
MAZDIS
Loafe Café
UBC Farm Market Stall
UBC Arts
Undergraduate Society
E Mall
University Blvd
War Memorial Gym
Entrance / Exit
2055
a142
2075
6060

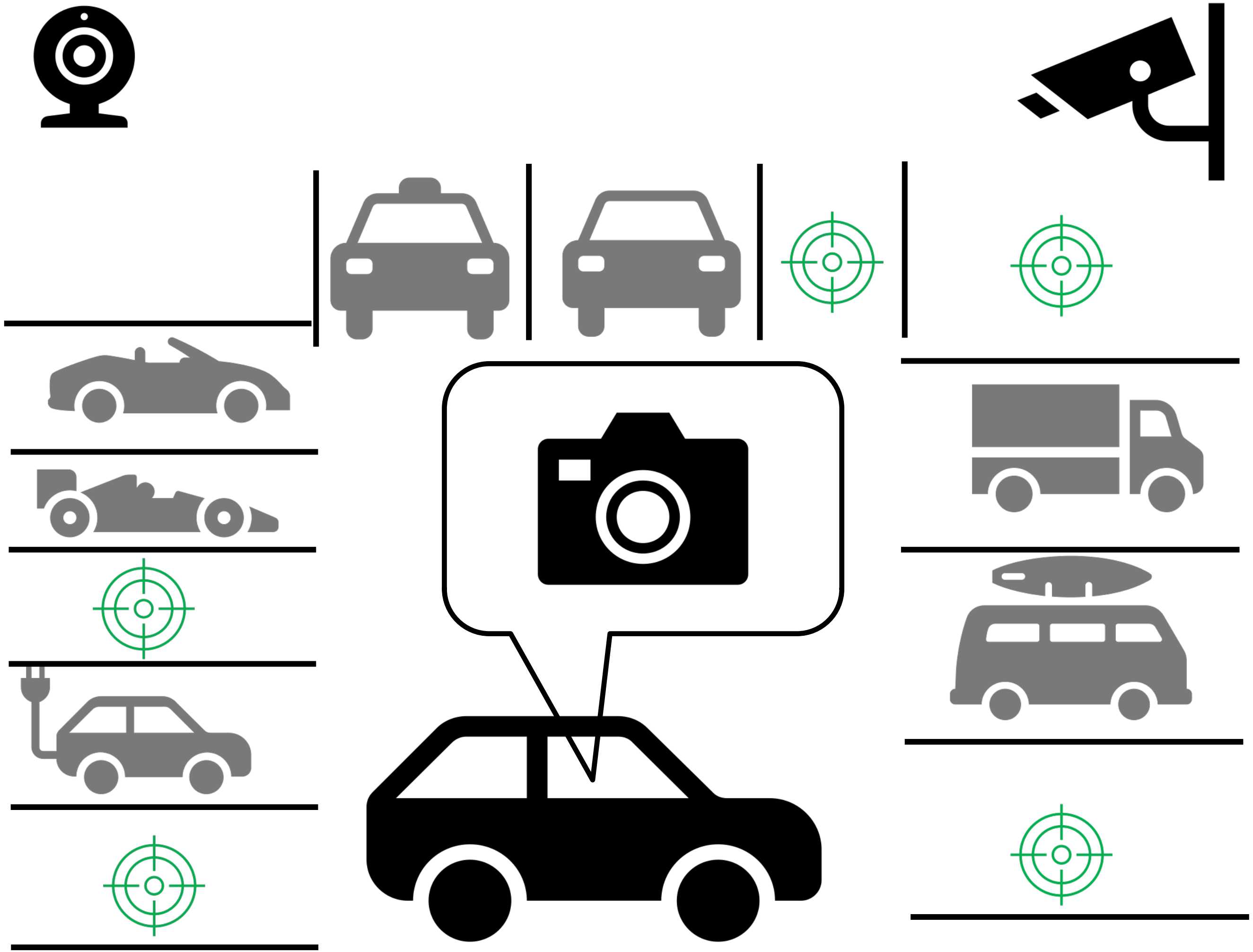
Our Approach



- deep learning based real-time occupancy detection system, using a video feed from security and parking inspection vehicle-mounted cameras.
  - to accurately detect and count the number of **uniquely** available parking spots in real-time.







Our Solution

- We chose YOLOv4 as the object detection and classification neural network
  - Tested against other Deep Learning architectures
  - Showed the best performance for this task

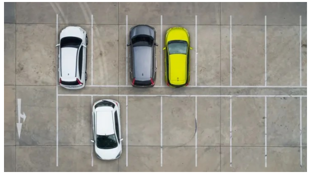
Dataset: Data Collection

- We first had to determine what type of commercial camera may be used that meets our requirements
  - Framerate 30 fps
  - Resolution (at least HD)
  - Affordability
  - compactness

AKASO EK700

Dataset: Data Collection

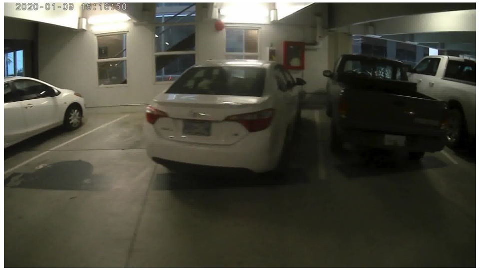
- In our implementation, an AKASO EK700 camera was mounted on top of a moving vehicle.
- Since the angle and distance of the camera from the parking spaces may affect the shape of the objects of interest in the frame, we tried to cover all practical scenarios of distances and angles.
- In addition, we tried to cover a huge variety of parking scenarios, with different types of cars, cars that were not "properly" parked and parking lots with different ways of indicating parking spaces.



Electrical and Computer Engineering

Dataset: Data Collection

- A large number of videos were captured and converted to single frames for labelling and training the network.
- To avoid overfitting from using similar frames, we chose only the frames with some significant difference.
- A total of 4000 unique frames was used for training our network.



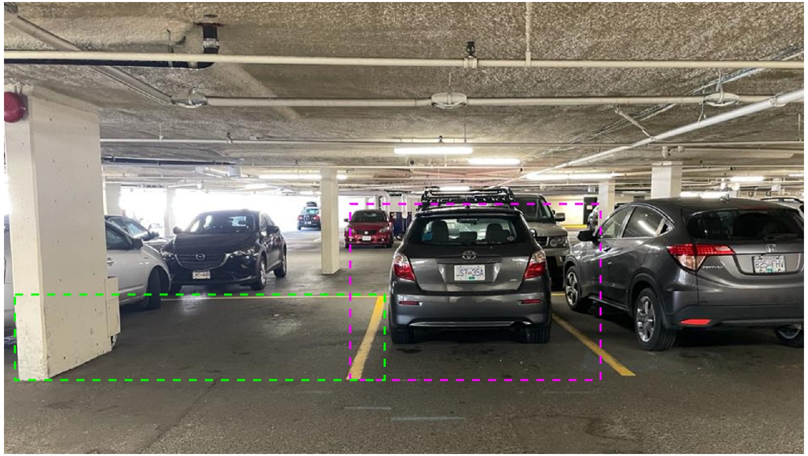
Dataset: Labels

- Recall: Our main objective is to design a deep learning-based approach that detects unoccupied parking spots and counts the uniquely available parking spaces.
- In order to facilitate the training process, we decided to assign two classes, a class for unoccupied parking and a class for occupied parking spaces.

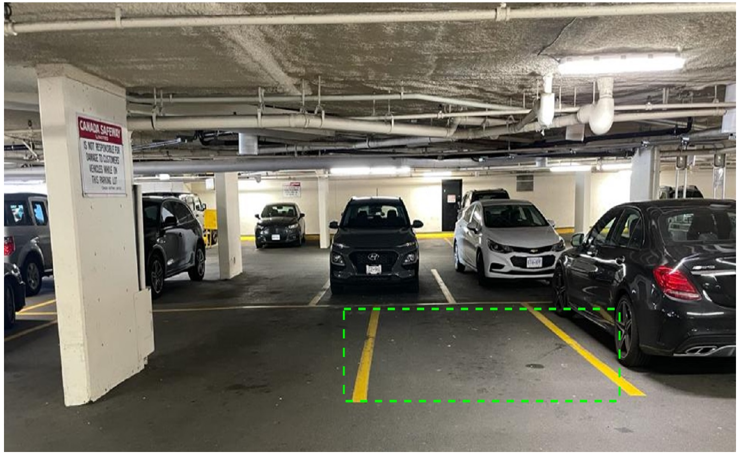
We categorize them into two classes:
Occupied
Unoccupied

Dataset: Labeling Strategy

- Since in this project, we want to count the number of **unique unoccupied parking** spots
  - we considered to label only the first unoccupied spot we encounter in every frame, if it exists.
- In the case of **occupied spots**, we label all the clearly visible spaces



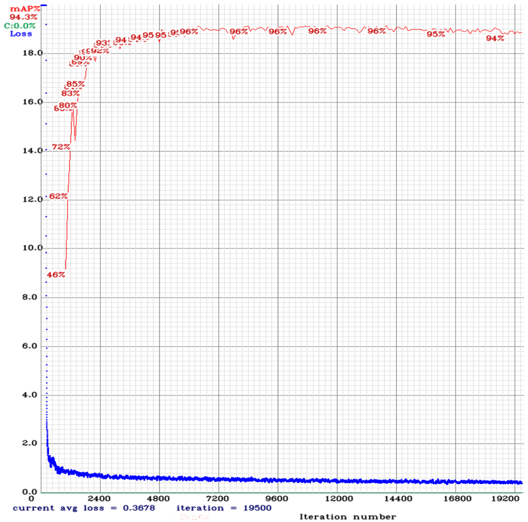
Only the fully visible occupied spot is labelled



Only the **first** unoccupied spot is labelled

Our YOLOv4 model:

- We randomly assigned 80% and 20% of our 4000 unique frames in to training frames and validation frames, respectively.
- We tried several training setups to find the best YOLOv4 model
- Using batch size of 128 and learning rate of 0.0001 we achieved the best mean average precision (96%) for our validation set.



Test Performance

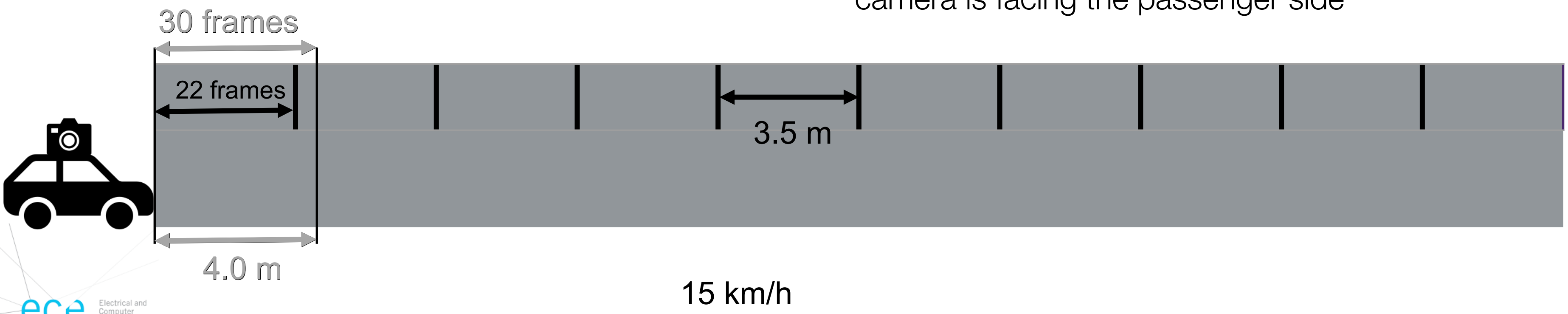
- In order to test our model, we used a MacBook Pro laptop with M1 chip with 8-core CPU, 8-core GPU and 16-core Neural Engine. The laptop proved to be very powerful and run our model in real-time using a **webcam feed**, performing real-time predictions at 30fps.
- Our model's prediction accuracy is measured based on its ability to classify the unoccupied and the occupied parking spots classes.
- In addition, a counting algorithm was developed that used the occupancy detection model to count the number of distinct parking stalls.

Test Scenario

- Service car speed
  - 15 km/h – 20 km/h
  - Driving normally
- Location of camera
  - On top of the car
  - Passenger side
  - Driver side
- Weather conditions
  - Rainy / sunny
  - Indoors/ outdoors

Test Performance

- Unoccupied parking space counting algorithm - always detect the first parking space and only one unoccupied space in each frame.
- Car speed used to determine which frame to consider next, making sure that we will “see” a new parking space in the next frame.
- For instance, car traveling at 15 km/h and camera capturing 30fps
  - Process every 22nd frame
  - Assuming the parking lot is normal size, and the camera is facing the passenger side



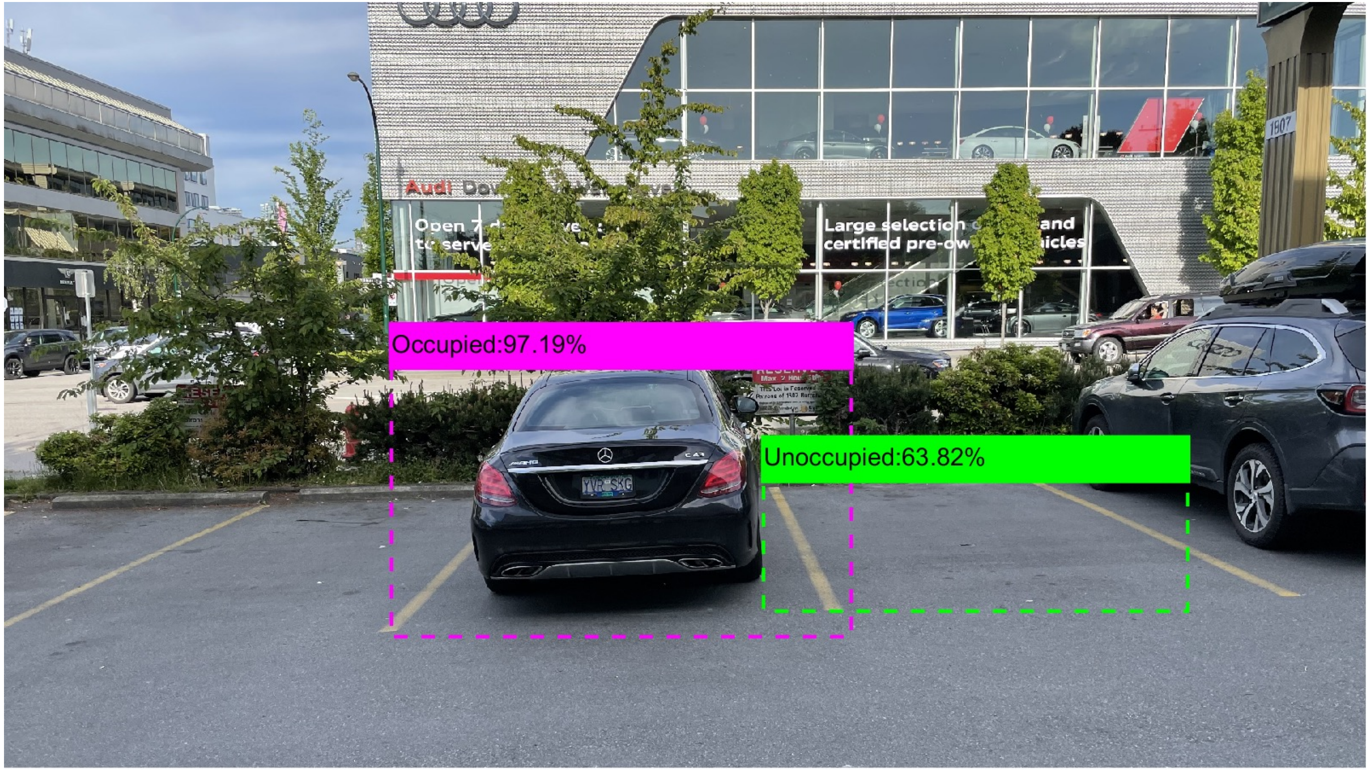
Test Performance

- Prediction accuracy is also determined by the prediction threshold.
- In this regard we considered several threshold values for the confidence thresholds.
- We realized that for our applications threshold of 30% achieved the best performance.

Test results for real time scenario

Parking Status	Average Precision	
	Thr = **30%**	Thr = 50%
Unoccupied	**90.59%**	88.86%
Occupied	**95.66%**	95.52%

Example 1: Detected bounding boxes



An example of test images with the predicted bounding boxes, labels, and the probabilities assigned to them.

Example 2: Detected bounding boxes

An example test image for the case that distance is two times larger than the case of the previous slide.

Conclusion

- We introduced a deep learning-based detection method for **unoccupied parking** spaces in a city parking lot.
- We developed a **unique labeling scheme**, which allowed us to **detect and accurately count** the total number of vacant spaces in the parking lot.
- We used the YOLOv4 network, and we achieved a detection accuracy of **90.59%** for unoccupied spaces and **95.66%** for the occupied spaces.
- As our training and validation dataset was only 4000 frames due to privacy issues, we are confident that this network can achieve even higher accuracy with a larger dataset and the same labeling scheme.

Thank you!
Q/A?