SkyDeploy Project Update

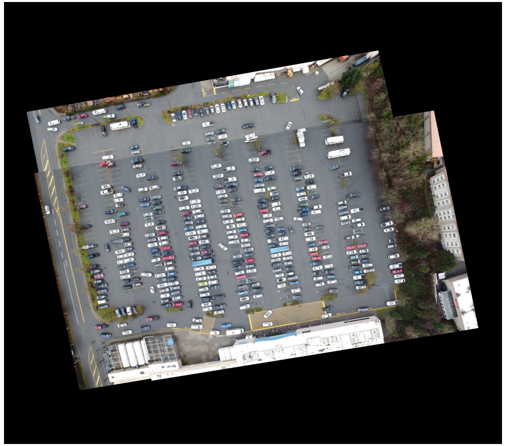
November 9th 2021

Tables of Contents

Project Overview

Project Overview

Image Stitching



- Image Initialization
- Global Alignment

Vehicle Detection

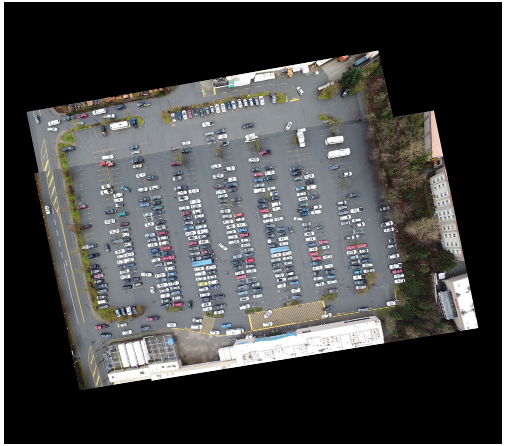
- Image Annotation
- Object Detection Network

Heatmap Generation

- Vehicle Counting
- Mosaic Registration

Project Overview

Image Stitching



- Image Initialization
- Global Alignment

Vehicle Detection

- Image Annotation
- Object Detection Network

Heatmap Generation

- Vehicle Counting
- Mosaic Registration

Image Stitching

Image Stitching

Input: Set of images from parking lot

Output: Mosaic image

- Modify perspective of each image so 3D correspondences align

Steps:

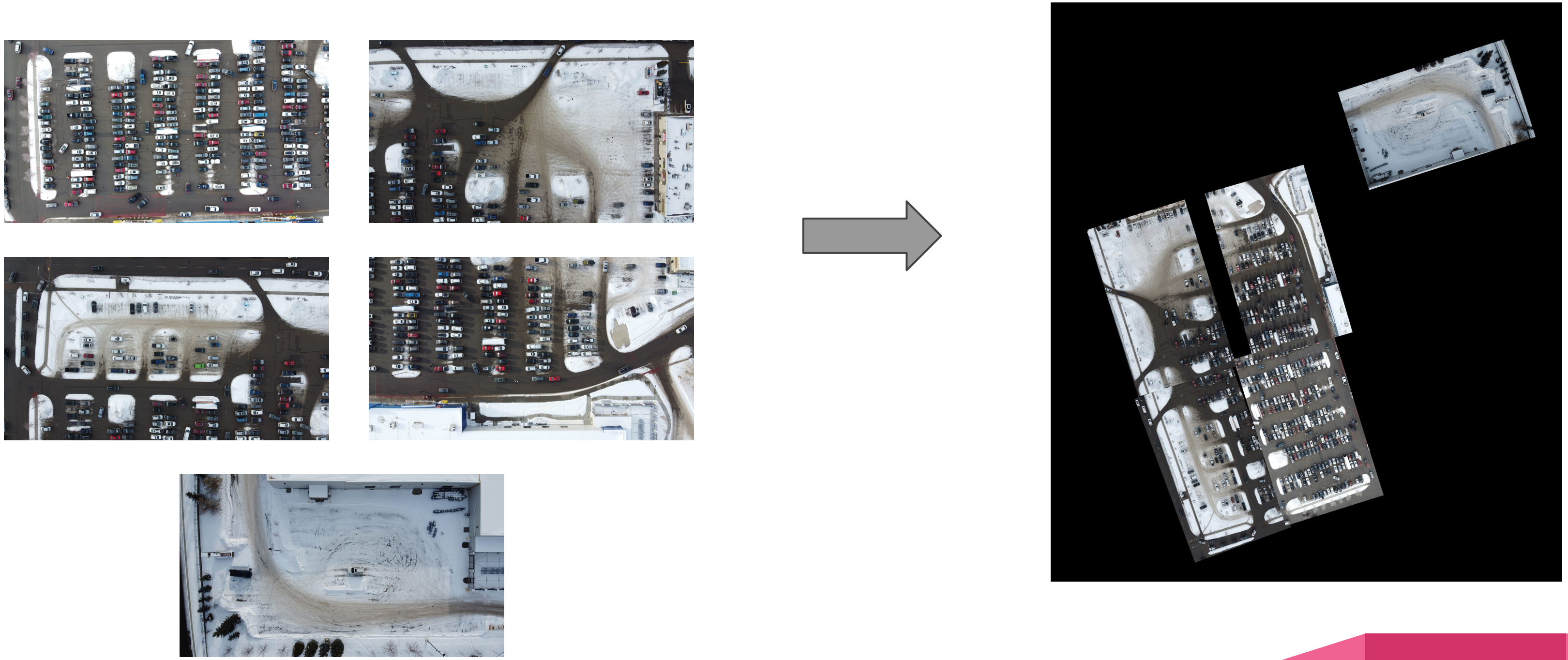
- Estimate Relative Transformations
- Image Initializations
- Global Alignment
- Performance Evaluation

General Update Notes - Image Stitching

- Image stitching is accurate and efficient
- Mosaics can be accurately registered across time intervals
- Sufficient for the vast majority of cases
    - Further optimizations planned once end-to-end system is built

Image Initializations

Main Idea: Initialization image locations in stitched image.



Algorithm

Initializations
Relative Transformations
Global Alignment

1. Initialize location and orientation of images in mosaic using drone metadata
2. Get Keypoints/Descriptors using SIFT
3. Determine pairwise keypoint matches with KNN
4. Filter best matches with Lowes ratio
5. Determine true relative transformations by comparing number of matches with minimum matches (inlier) threshold
6. Solve for pairwise relative transformations using RANSAC
7. Construct symmetric graph in which each node is an image and edges represent transformations
8. Define loss between the ground truth relative transformations and the relative transformation defined by the global location of the images in the mosaic
9. Use back propagation to compute gradients of the parameters defining the pose of the image with respect to the previously defined loss and update parameters to minimize this loss

Image Initialization

Image Initializations

Method: Use GPS, orientation and altitude information to map an image from world coordinates to stitched image coordinates. The resulting coordinates act as initializations for the global alignment step.

GPS Coordinates + Orientation + Altitude

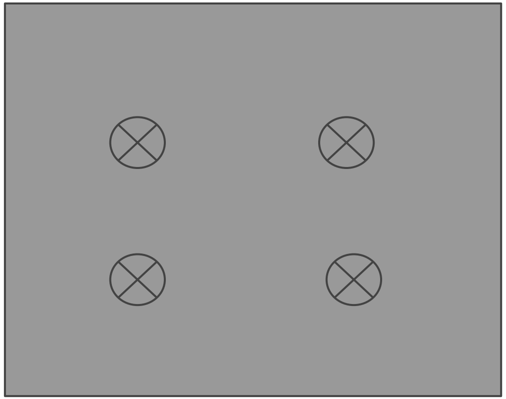
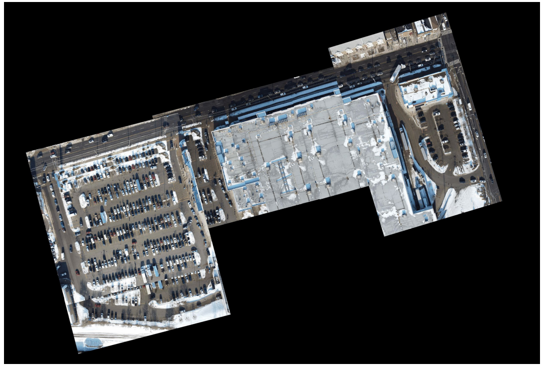


Image Initializations



Relative Transformations

Coordinate Systems - Pixel Coordinates

Image I_0: x_0, y_0

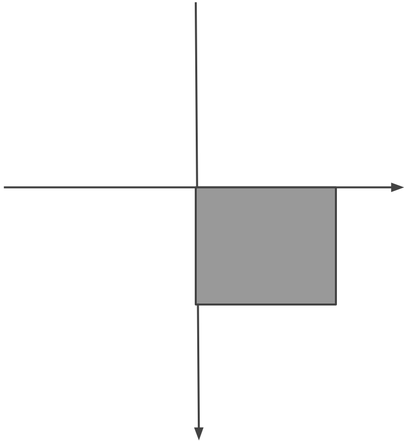
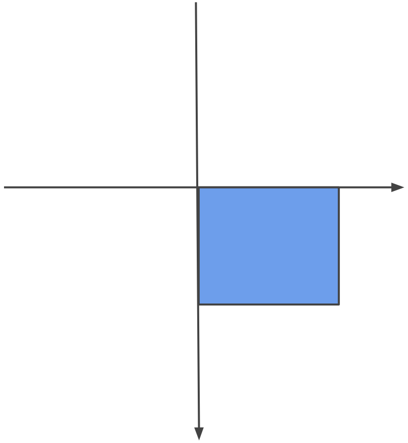


Image I_1: x_1, y_1



Relative Transformations

Transformations between the pixel coordinates of a pair of images

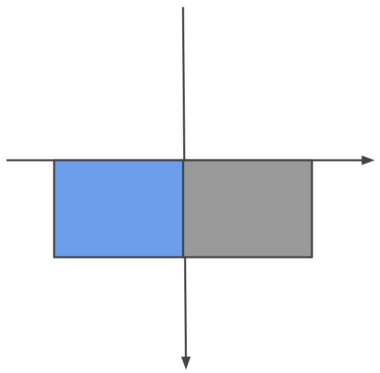
Let A_0 be a partial 2d affine transformation that maps pixel coordinates in I_1 to pixel coordinates in I_0

There are 4 degrees of freedom: Scale parameter, s, rotation parameter Θ, and translation parameters t_X and t_Y

True Relative Transformations are computed using SURF and RANSAC

$$\begin{bmatrix} \cos(\theta)\cdot s & -\sin(\theta)\cdot s & t_x \\ \sin(\theta)\cdot s & \cos(\theta)\cdot s & t_y \end{bmatrix}$$

Image I_0 Coordinates



Global Alignment

Global Alignment

Initialization of Global Image Transformations provided: Θ, t_X, t_Y and s

Use the global transformations for a pair of images I_0 and I_1 to compute the estimated relative transformations between images

Project the four corners of Image I_1 onto the pixel coordinates of Image I_0 using estimated transformation

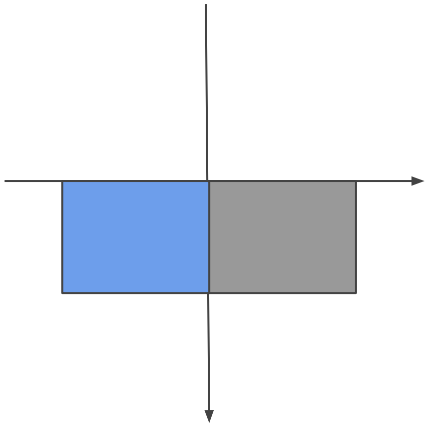
Project the four corners of Image I_1 onto the pixel coordinates of Image I_0 using true transformation

Compute loss as summations of the squared error over the 4 points for each transformation in the Image Set

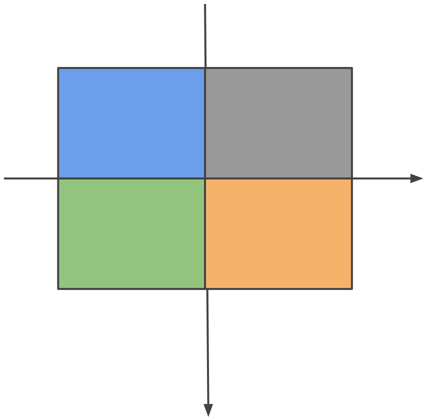
Minimize loss with respect to Θ, t_X, t_Y and s

Global Coordinates

Image I_0 Coordinates

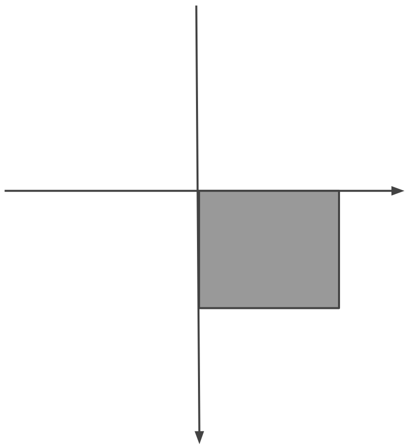


Global Coordinates x, y

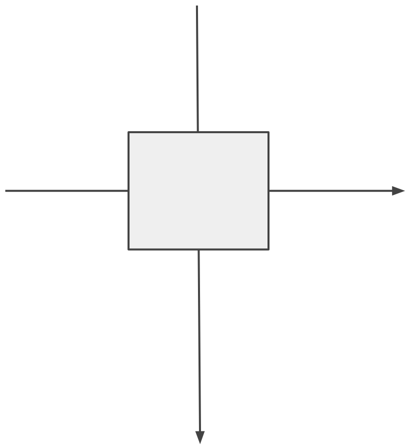


Pixel Coordinates to Global Coordinates

1) Image I_0: x_0, y_0

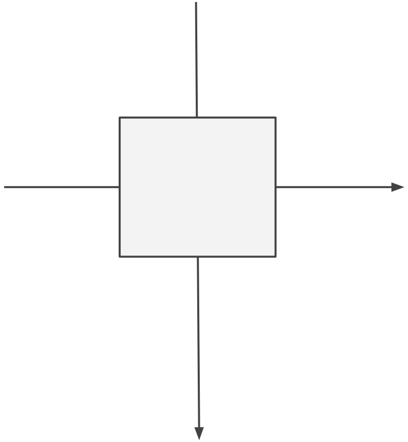


2) Translate to origin

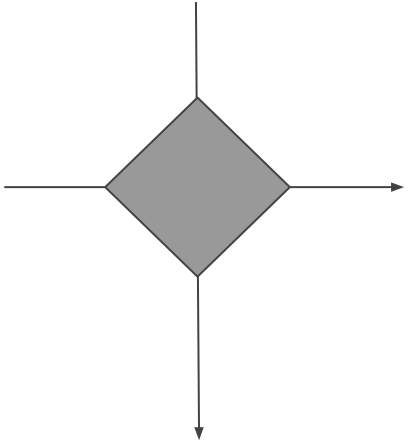


Pixel Coordinates to Global Coordinates

3) Scale about origin

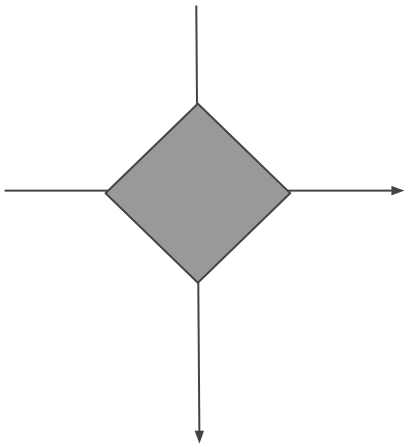


4) Rotate about origin

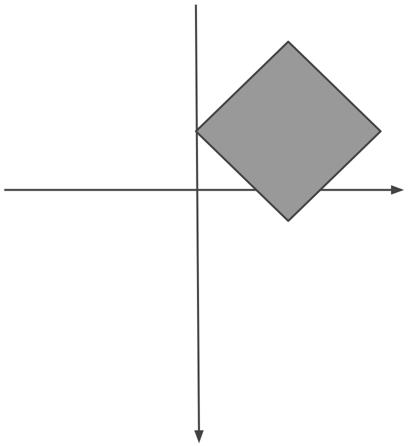


Pixel Coordinates to Global Coordinates

5) Translate



Global Coordinates x, y



Global Transformation

$$\begin{bmatrix} 1 & 0 & x_f \\ 0 & 1 & y_f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_r \\ 0 & 1 & -y_r \\ 0 & 0 & 1 \end{bmatrix}$$

Global Transformation to Relative Transformation

Image $I_0 = x_0, y_0$

Image $I_1 = x_1, y_1$

$x_0, y_0 = x_1, y_1$

Transformation Matrix $T_0 : x_0, y_0 => x, y$

Transformation Matrix $T_1 : x_1, y_1 => x, y$

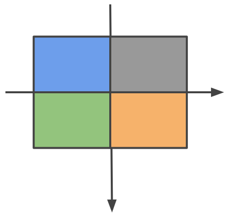
$T_0 * I_0 = x = T_1 * I_1$

$T_0^{-1} * T_1 * I_1 = I_0$

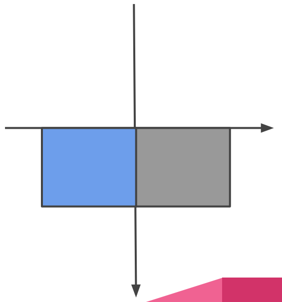
$R_e * I_1 = I_0$

R_e : Est. relative transformation between I_1 and I_0

Global Transform



Relative Transform:



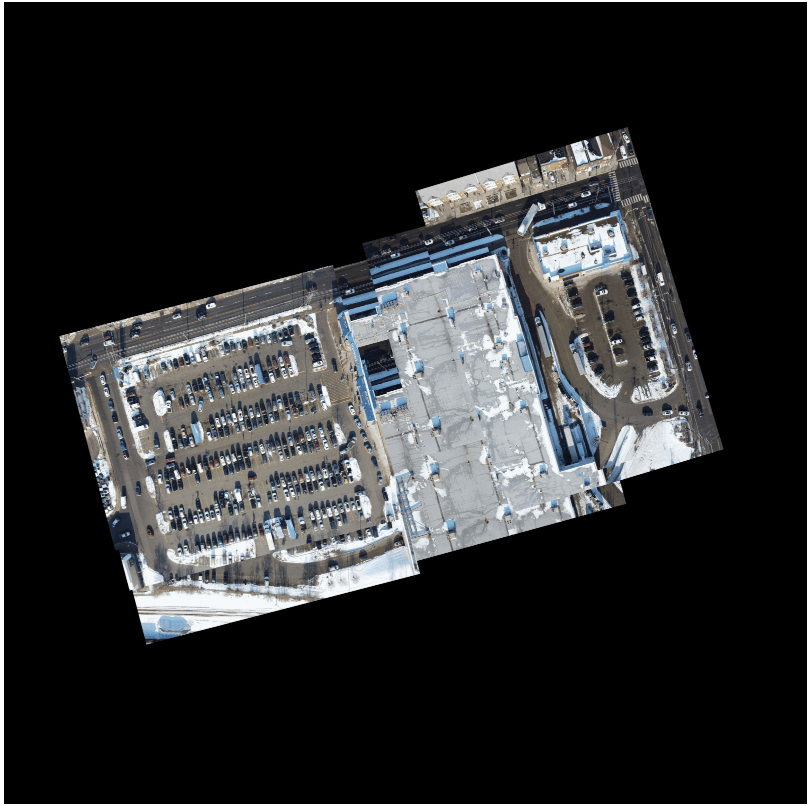
Loss and Optimization

Loss is calculated by taking the squared error between the points transformed by R_e and R

Minimize loss with respect to Θ, t_X, t_Y and s

Results

Global Alignment



Performance Evaluation

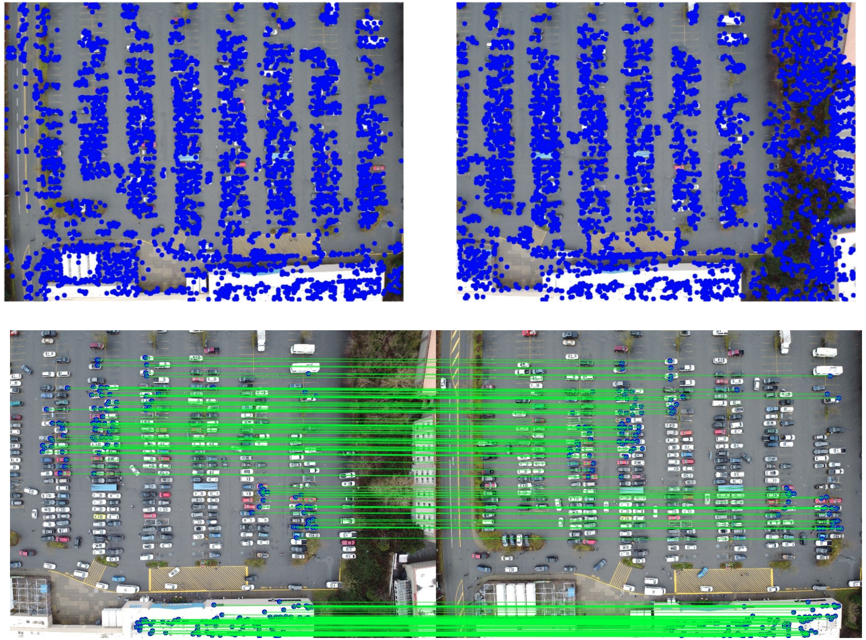
- Plotting image initializations, key points, matches

- Visually Assessing Stitched Images

- Visually assessing quality of mosaic registration across time intervals

- Squared Error of overlapping images regions

Keypoint Detection and Matching



Performance Evaluation

- Plotting image initializations, key points, matches
- Visually Assessing Stitched Images
- Visually assessing quality of mosaic registration across time intervals
- Squared Error of overlapping images regions

Pairwise Stitching

Performance Evaluation

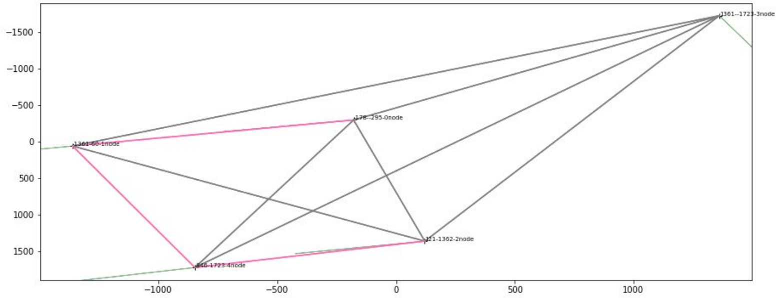
- Plotting image initializations, key points, matches

- Visually Assessing Stitched Images

- Visually assessing quality of mosaic registration across time intervals

- Squared Error of overlapping images regions

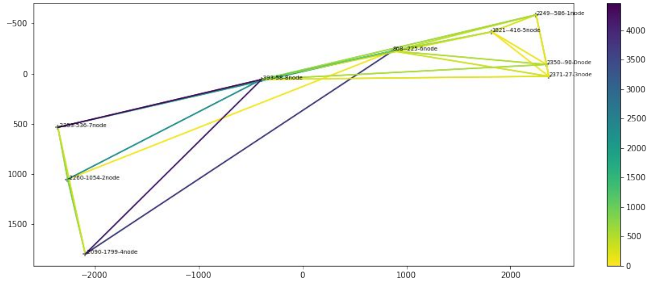
GPS Initializations



Performance Evaluation

- Plotting image initializations, key points, matches
- Visually Assessing Stitched Images
- Visually assessing quality of mosaic registration across time intervals
- Squared Error of overlapping images regions

Link Loss Diagram



Performance Evaluation

- Plotting image initializations, key points, matches
- Visually Assessing Stitched Images
- Visually assessing quality of mosaic registration across time intervals
- Squared Error of overlapping images regions

Mosaic Error Map

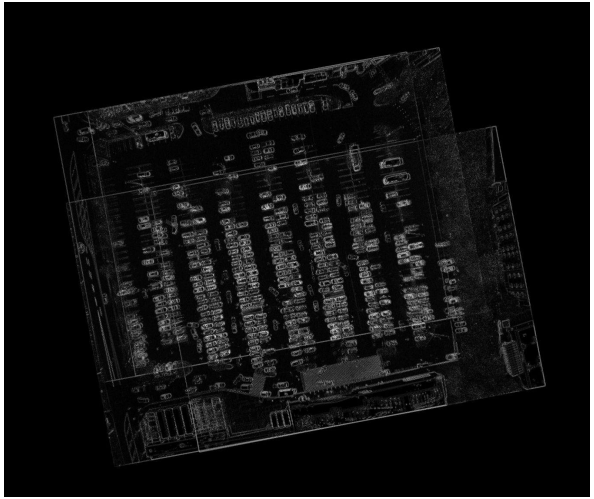
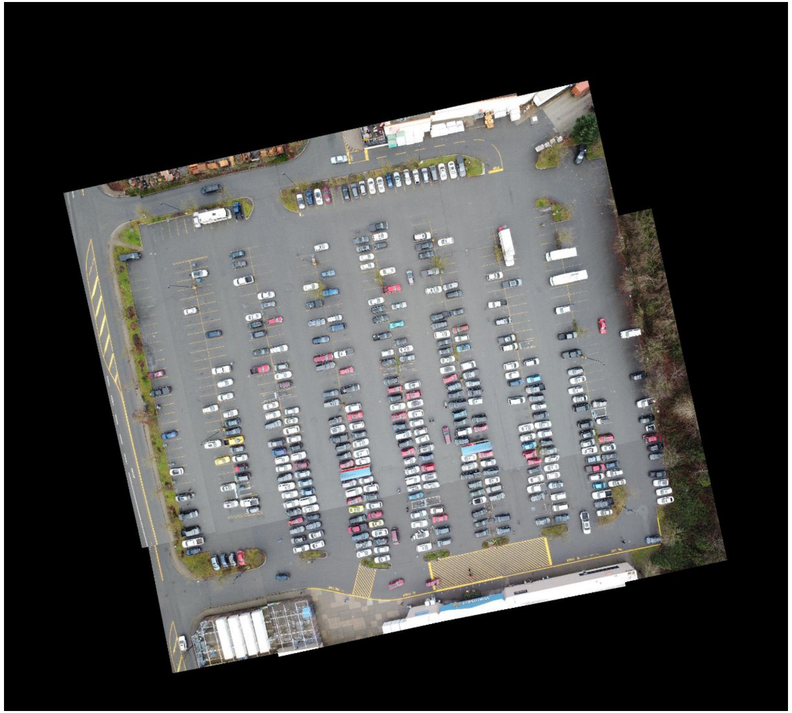


Image Stitching Results



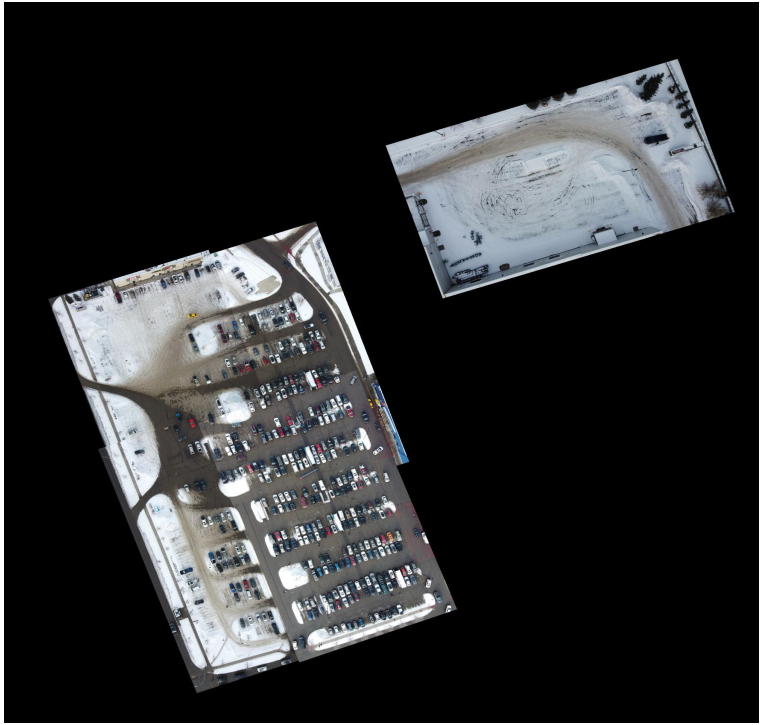
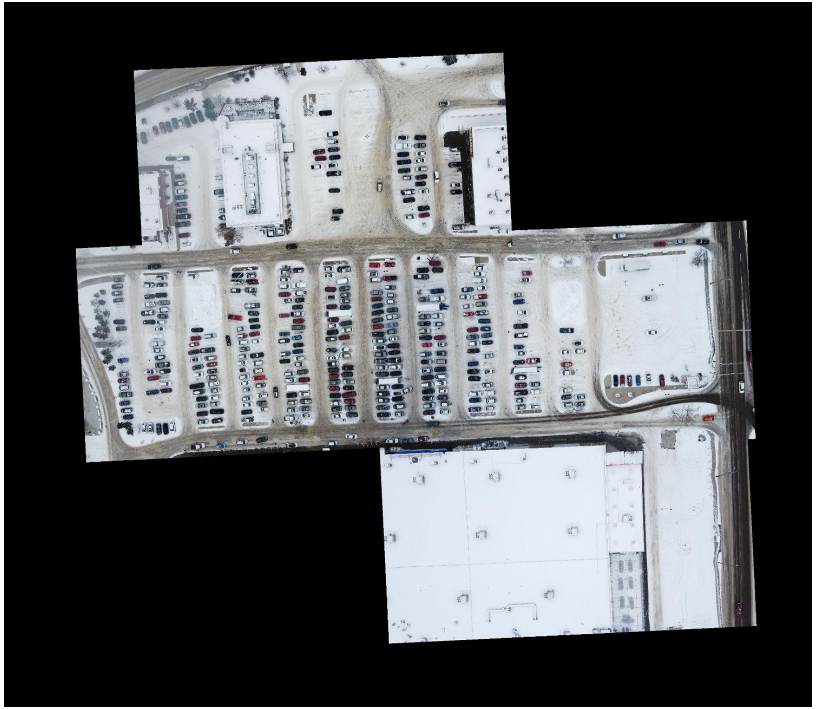


Image Stitching Results

Image Stitching Results



Next Steps - Image Stitching

- Explore deep learning based image stitching methods

- Finetune current image stitching pipeline

Vehicle Detection

Vehicle Detection

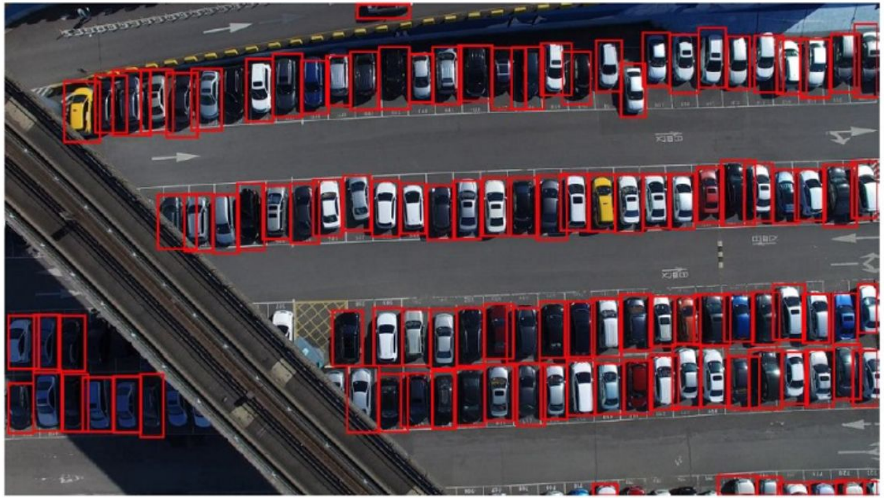
Input: Drone Image

Output: For each object:

- **Category label**
- **Bounding box**

Steps:

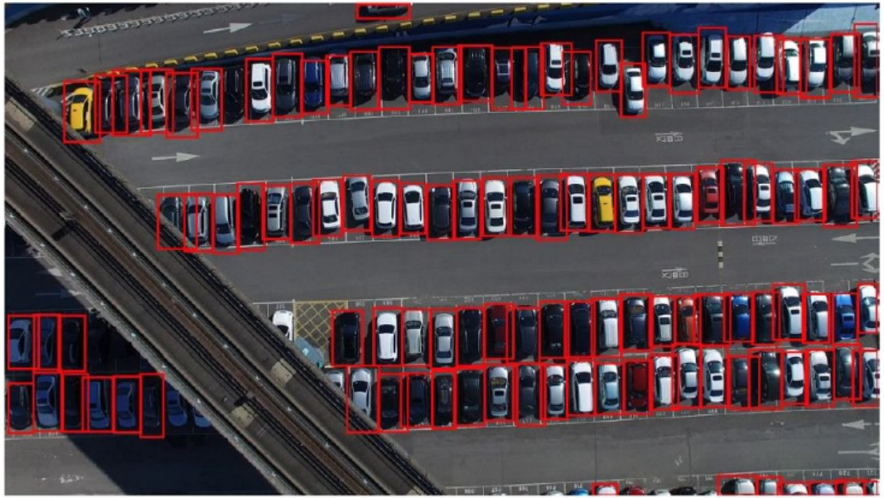
- Train Vehicle Detection Network
- Evaluate Performance
- Apply to new samples



Counting number: 114 cars
Ground Truth: 121 cars

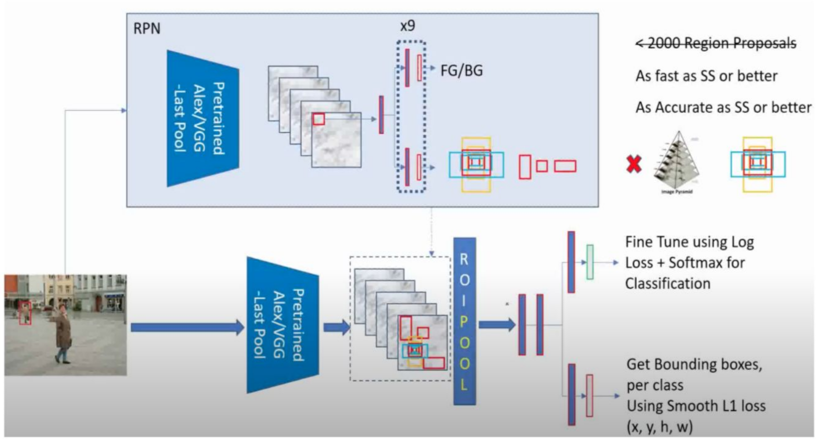
General Update Notes - Vehicle Detection

- Vehicle Detection is very accurate
  - Low False Positive Rate
  - Robust to occlusion and lighting conditions
- Leveraging open source datasets
  - More training data
  - More diverse examples
- Sky Deploy Image Annotation in progress
- Exploring ways to robustly detect commercial vehicles



Counting number: 114 cars
Ground Truth: 121 cars

Object Detection Approaches



Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. 2015.

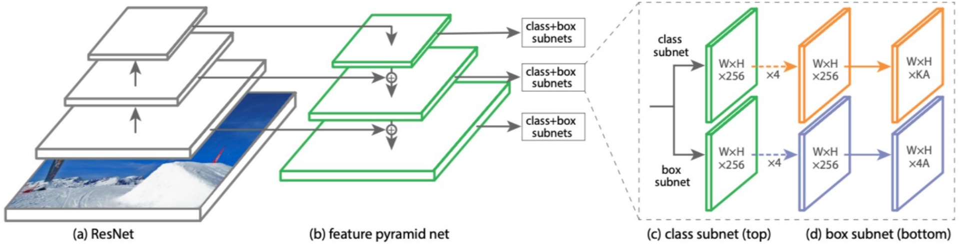


Figure 3. The one-stage **RetinaNet** network architecture uses a Feature Pyramid Network (FPN) [20] backbone on top of a feedforward ResNet architecture [16] (a) to generate a rich, multi-scale convolutional feature pyramid (b). To this backbone RetinaNet attaches two subnetworks, one for classifying anchor boxes (c) and one for regressing from anchor boxes to ground-truth object boxes (d). The network design is intentionally simple, which enables this work to focus on a novel focal loss function that eliminates the accuracy gap between our one-stage detector and state-of-the-art two-stage detectors like Faster R-CNN with FPN [20] while running at faster speeds.

Focal Loss for Dense Object Detection. 2017.

Vehicle Detection Approaches

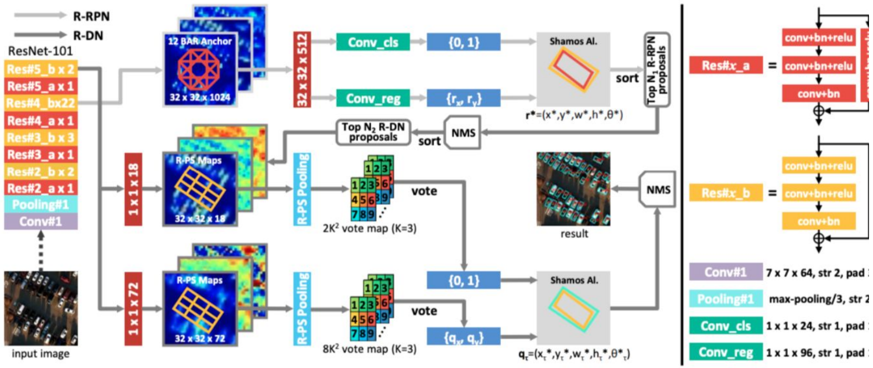


Fig. 2. Overall architecture of the proposed R^3-Net. There are two main branches. One is rotatable region proposal network (R-RPN), and the othe rotatable detection network (R-DN). R-RPN is for generating rotatable region of interests (R-RoIs) for R-DN, and then R-RoIs are fed into R-DN to proc final rotatable detection boxes. Key: str = stride; pad = padding; conv = convolution; bn = batch normalization; relu = ReLU function.

R^3-Net: A Deep Network for Multi Oriented Vehicle Detection in Aerial Images & Video. 2019.

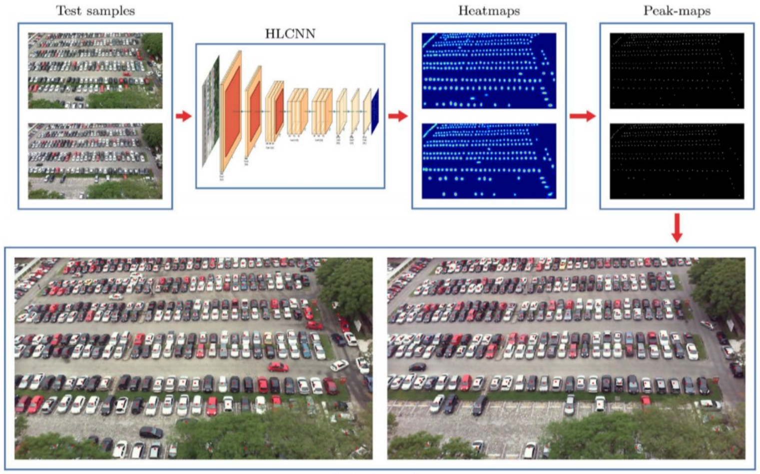
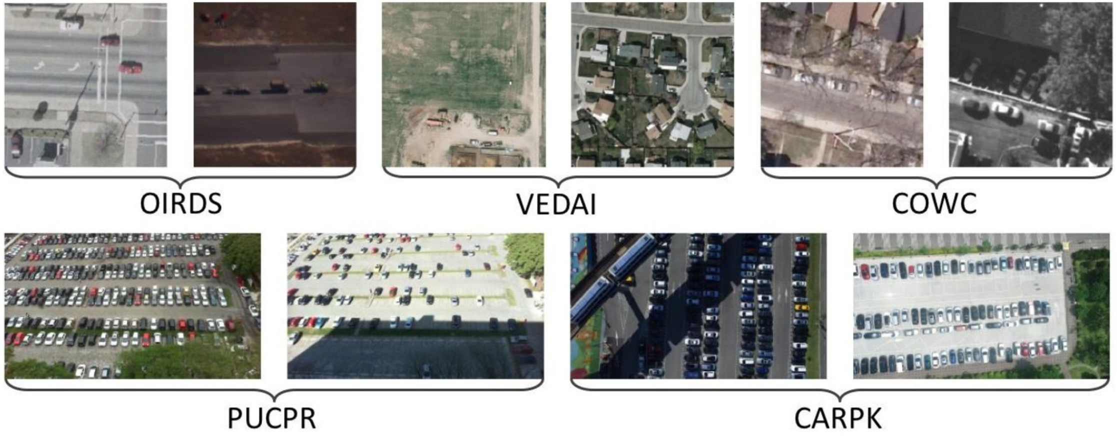


Fig. 1 Overview of the proposed car counting framework. Detected center points of the target cars have been indicated by red point in output samples

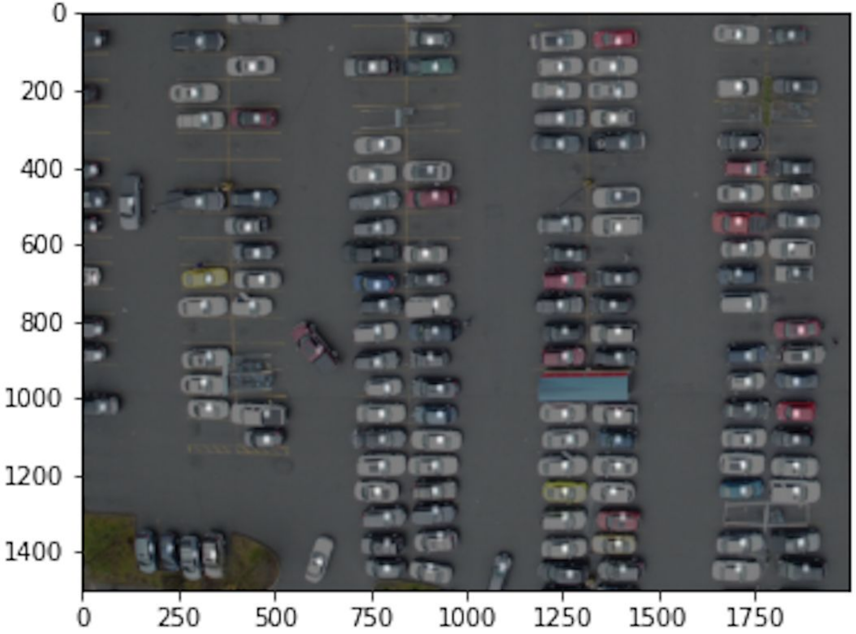
An accurate car counting in aerial images based on convolutional neural networks. 2021

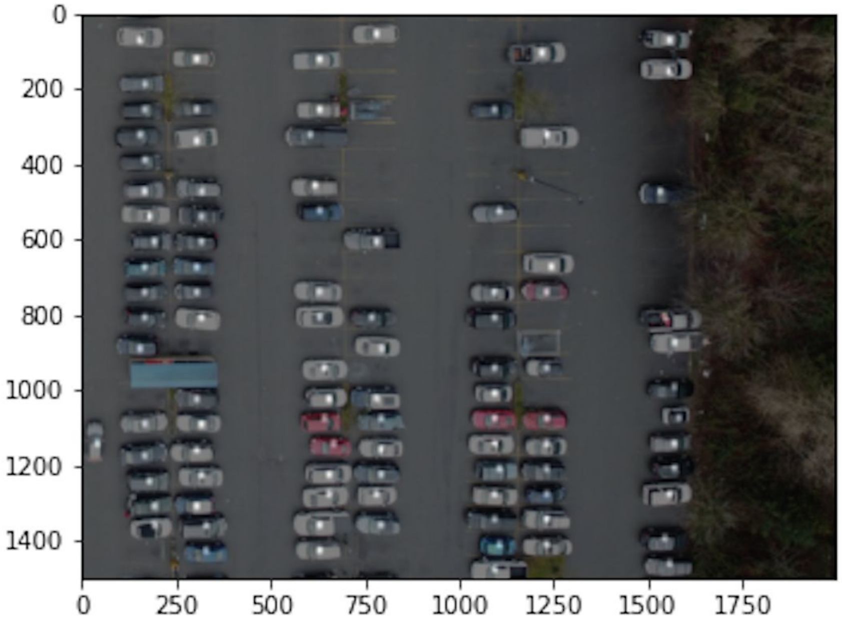
Vehicle Detection Datasets

Dataset	Sensor	Multi Scenes	Resolution	Annotation Format	Car Numbers	Counting Support
OIRDS	satellite	✓	low	bounding box	180	✓
VEDAI	satellite	✓	low	bounding box	2,950	✓
COWC	aerial	✓	low	car center point	32,716	✓
PUCPR	camera	×	high	bounding box	192,216	×
CARPK	drone	✓	high	bounding box	89,777	✓



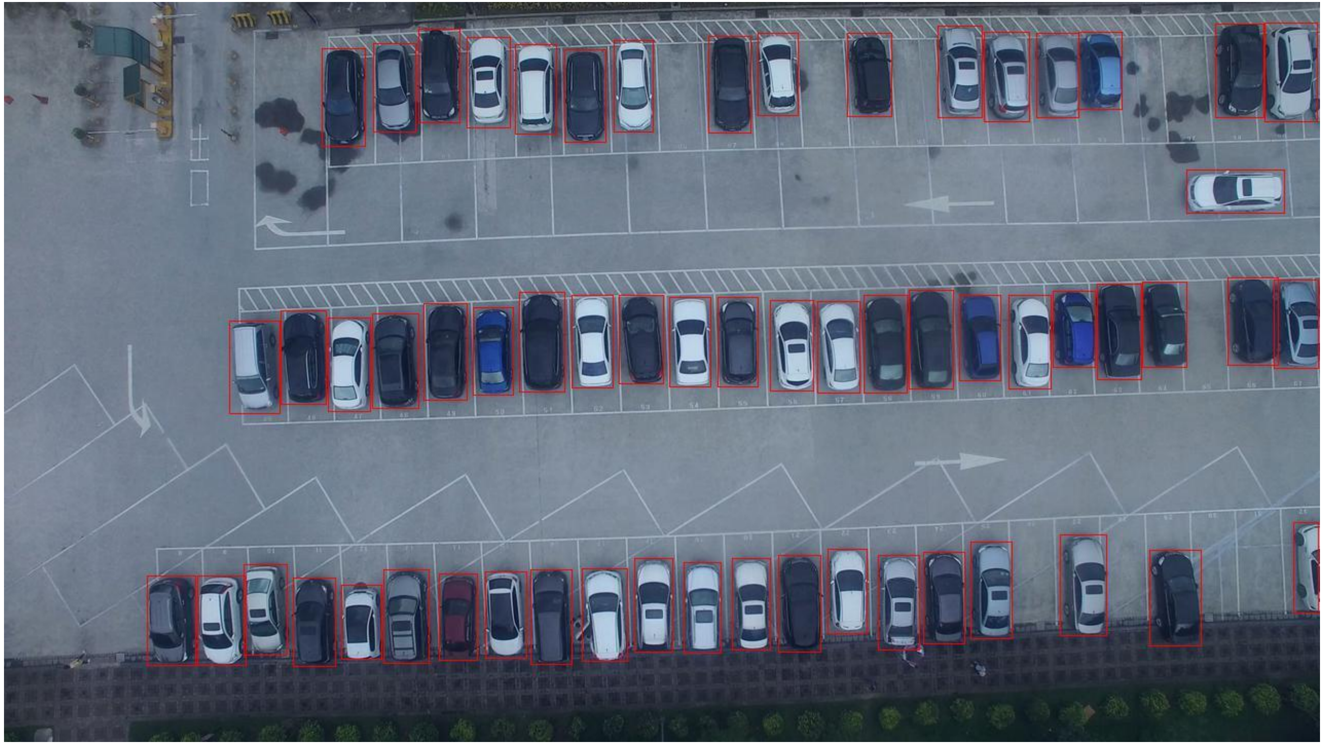
Vehicle Detection Results - HLCNN



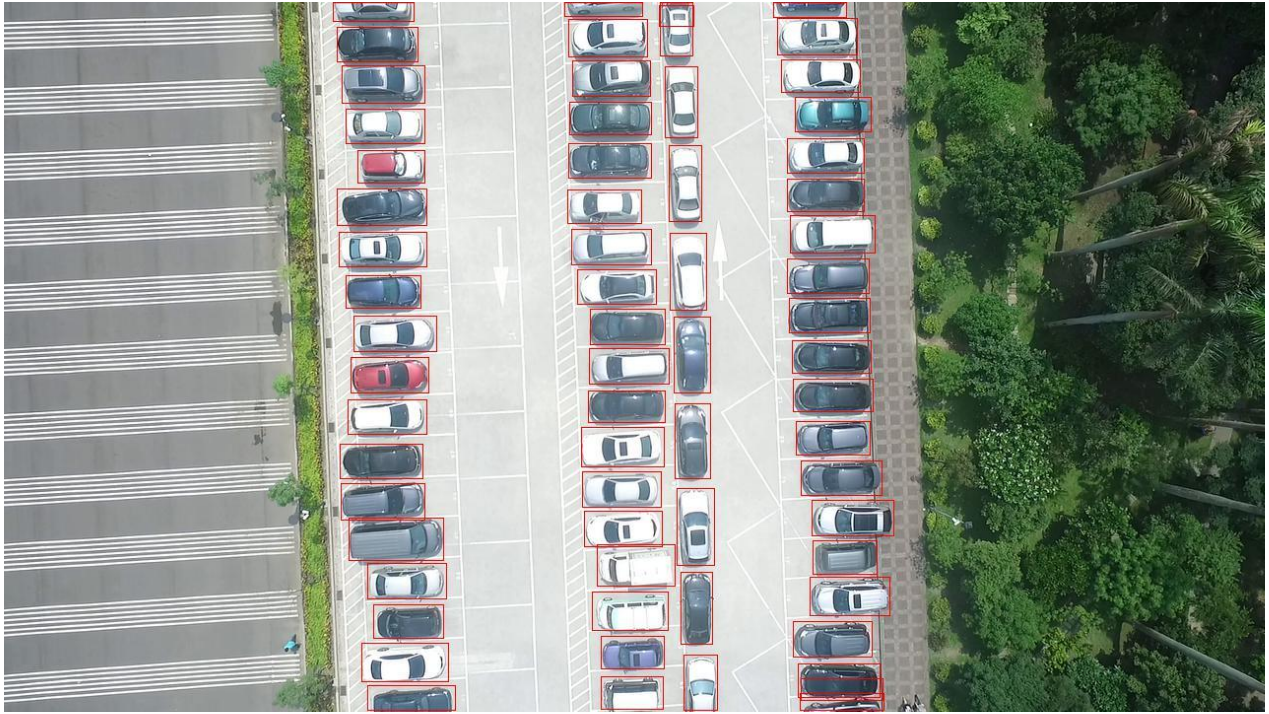


Vehicle Detection Results - RetinaNet

Vehicle Detection Results - FasterRCNN



Vehicle Detection Results - Mobilenet



Vehicle Detection Results - SSD

Next Steps - Vehicle Detection

- Do not count instances of commercial vehicles or vehicle not in parking spots
- Complete Image Annotation
- Finetune approach

Heatmap Generation

Heatmap Generation

Input: Mosaics with corresponding vehicle detections

Output: Heatmap

- Expresses the frequency each parking space is used

Steps:

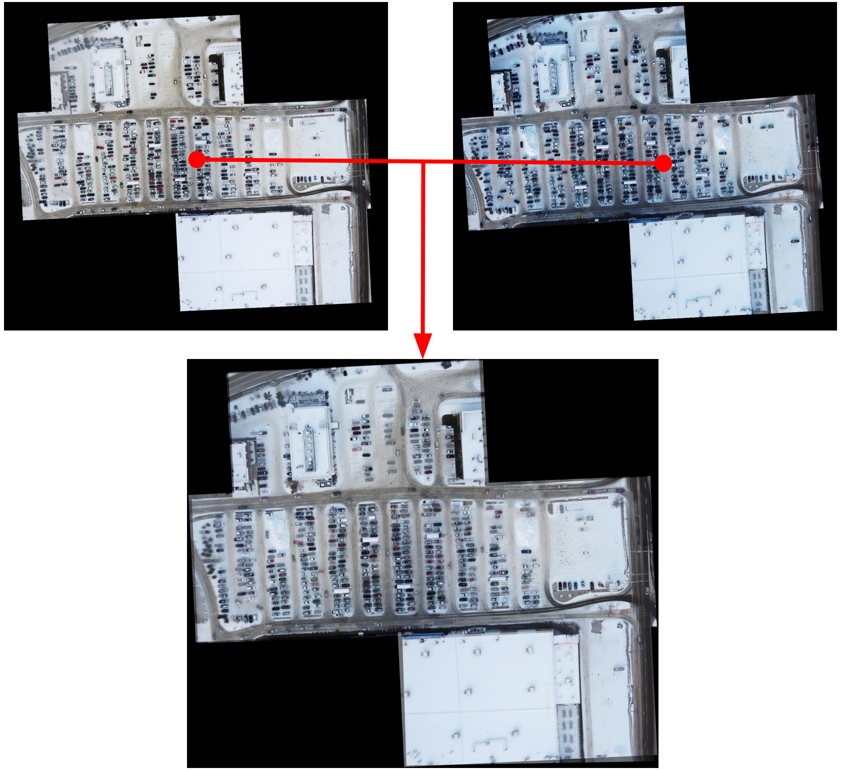
- Register Mosaics
- Vehicle Counting

General Update Notes - Heatmap Generation

- Able to accurately register mosaics across time intervals

- Developing approach to count vehicles mosaics

Mosaic Registration



Mosaic Registration Results

Next Steps - Heatmap Generation

- Working towards the ability to register vehicle detections within image sets

- Implement Vehicle Counting across mosaics

- Seamlessly Integrate into existing pipeline

Next Steps

Next Steps

Build End to End System:

- Working towards the ability to register vehicle detections within image sets
- Implement Vehicle Counting across mosaics
- Seamlessly Integrate into existing pipeline

Refine Object Detection:

- Do not count instances of commercial vehicles or vehicle not in parking spots
- Complete Image Annotation
- Finetune approach

Refine Image Stitching:

- Explore deep learning based image stitching methods
- Finetune current Image Stitching Pipeline

Productionalize Code and User Interface