Stereo-RCNN

Team #32
Ji-il Park (20195355)
Hyunyong Jeon(20193813)
Leon Thormeyer(20196530)
Project option : #1(Reproduce the existing paper)
Github repository : https://github.com/hyunyongjeon/repo.git

[Introduction]

1. Why do we need object detection on a runway

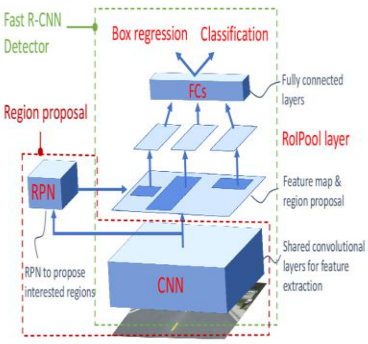
<Runway snowplow of each country(Korea, Russia, Germany)>

Autonomous Runway snowplows need to remove stacked snow when it snows. It means that it is difficult to use LiDAR for object detection because it is vulnerable to falling snow. Therefore, we need to detect the object by only using camera vision. Just for the case, that we need a 3D object detection ability by using only camera vision. For these reasons, we decided to find a 3D object detection model based on stereo camera vision.

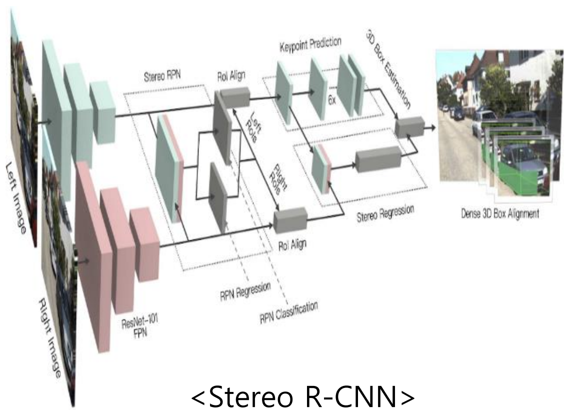
2. What is Stereo-RCNN

   We searched for a target paper which focuses on object detection for autonomous vehicles. Of those, we decided to have the Stereo-RCNN paper as our target paper, because it includes various session like AI models, RPN sessions, LiDAR point cloud and so on. We chose it because we thought the paper offers a variety of approaches to alter it optimize for our use case.

   (1) Network architecture of Stereo-RCNN



<Faster R-CNN>



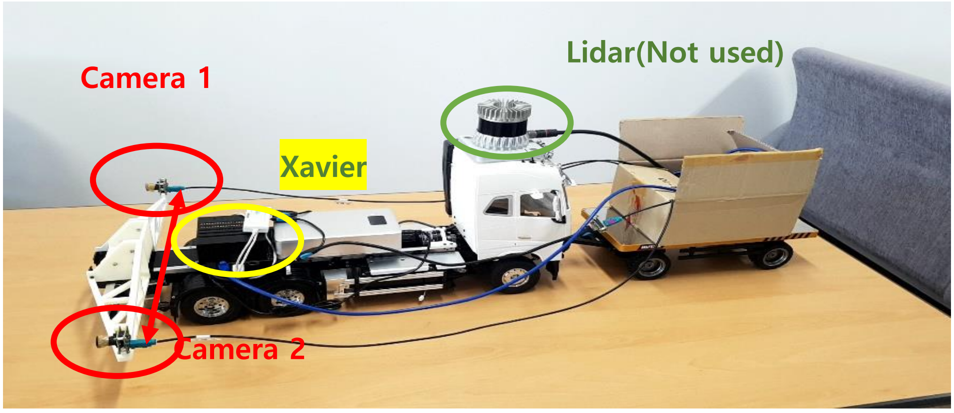
<Stereo R-CNN>

Stereo R-CNN extends Faster R-CNN for stereo inputs to simultaneously detect 3D bounding boxes by using the left and right images. This algorithm is essential for autonomous driving with vision cameras without using LiDAR[1].

[1] Faster R-CNN use mono image, so only make 2D bounding box.

[Experimental Design]

1. Hardware setup
   (1) RC car model



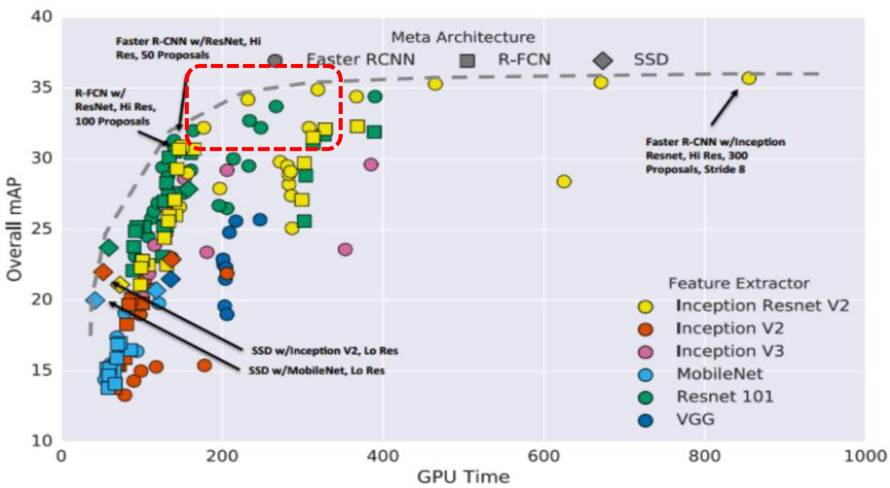
To get test data, we try to build our own test-bed RC car model which contains sensors and a computer.

1) Sensors : Stereo camera(Baseline : 0.54m/consider KITTI model) + LiDAR
2) Computer : Jetson AGX Xavier Dev.Kit(GTX 1060Ti grade performance)

2. Algorithm modification

   In terms of algorithm modification, two attempts were made. Because the Stereo-RCNN model consists of two big parts, the AI model (ResNet), and the Stereo RPN (RPN regression, RPN classification), we tried to modify these parts consecutively.

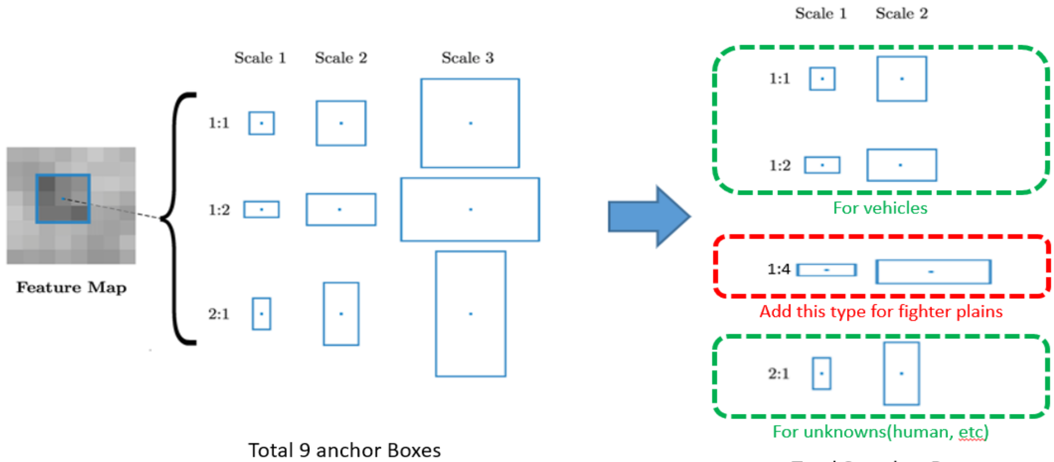
   (1) AI model
   1) Try #1(Resnet algorithm improvements)
      - Since the Resnet-152 model is already applied, performance improvement is limited
   2) Try #2(Resnet+Google inception, VGG model)



      - Resnet+Google-inception is known for good performance but is rather bad detecting the object (for unknown reason).
      - VGG model performs poorly compared to ResNet, as it is talked about in several papers.

   (2) RPN session
   1) Try #1(Decided to modify RPN part of Resnet)
      - Resize the anchor shape to fit objects on the runway such as airplanes

3. RPN modification



Total 9 anchor Boxes

Total 8 anchor Boxes

1) Assumptions for RPN modification are as follows :
   - Operating environment: Airplane runway
   - Expected target object: Airplanes, vehicles, and unknowns

   So, we need to change the appropriate anchor size for the operation environment.
2) The object to be detected in the runway environment are mainly lateral wide air planes. As a result, we added horizontally wide anchors to identify airplanes (Anchor ratio 1:4).
3) Increasing the ratio to four types may exceed the GPU processing memory, so in consideration of this, the scale needs to be reduced to two types (Scale factor changes from 3-scale to 2-scale).

4. Modification of Stereo camera calibration parameters(Intrinsic & extrinsic)

< 1^{st} left side image of 96 images>

< 1^{st} right side image of 96 images>

1) To find the camera setting suitable for us, the baseline was selected with reference to the KITTI model (0.54m) and calibrated.

   (1) Camera parameters contain focal length (f), principal point (c), distortion (k, p)

   (Left camera) rms = 0.148793, fx = 767.646362, fy = 769.060456, cx = 669.203517, cy = 388.021517, k1 = -0.249038, k2 = 0.068186, p1 = 0.001469, p2 = -0.003465

   (Right camera) (Skip / see the presentation material to find out data)

   (2) Calibration matrix contains Intrinsic, distortion matrix, and so on.

   (Skip / see the presentation material to find out data)

[Test & Results]

Outdoor test preparation

• 1st test (11.27/KAIST Front Door) : not suitable images for testing

* Problems : ① different view angle between test & training images
② Background that makes vehicle identification difficult

• 2nd test (11.30 / Habit @) : Good test image

* Similar to the training image (vehicle shape, view angle, etc...)

1. Data acquisition

1) 1st test (Unsuitable images for testing)

(1) The Dataset is not suitable for testing. We analyzed the 2-reasons for this. First, we photographed the side of a moving vehicle, but in the KITTI dataset, the vehicles are photographed in the direction of travel. Because of that, the view angles of the two data sets are different. Second, we photographed at sunset, so background of the images makes vehicle identification difficult.

2) 2nd test(Suitable images for testing)

(1) The shooting environment is similar to the KITTI images (vehicle shape, view angle, etc.).

2. Simulation result

• **Original algorithm**

• **RPN change algorithm**

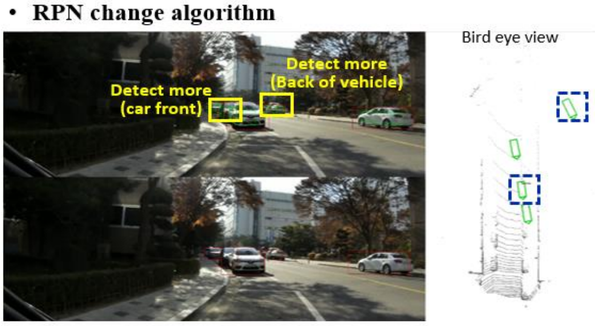
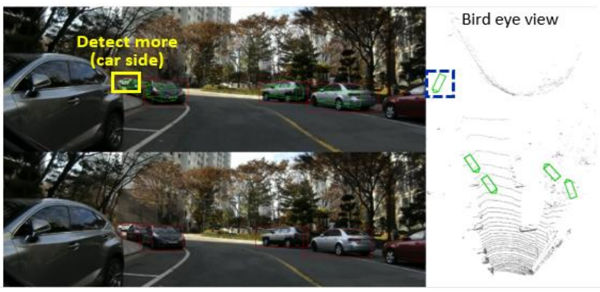
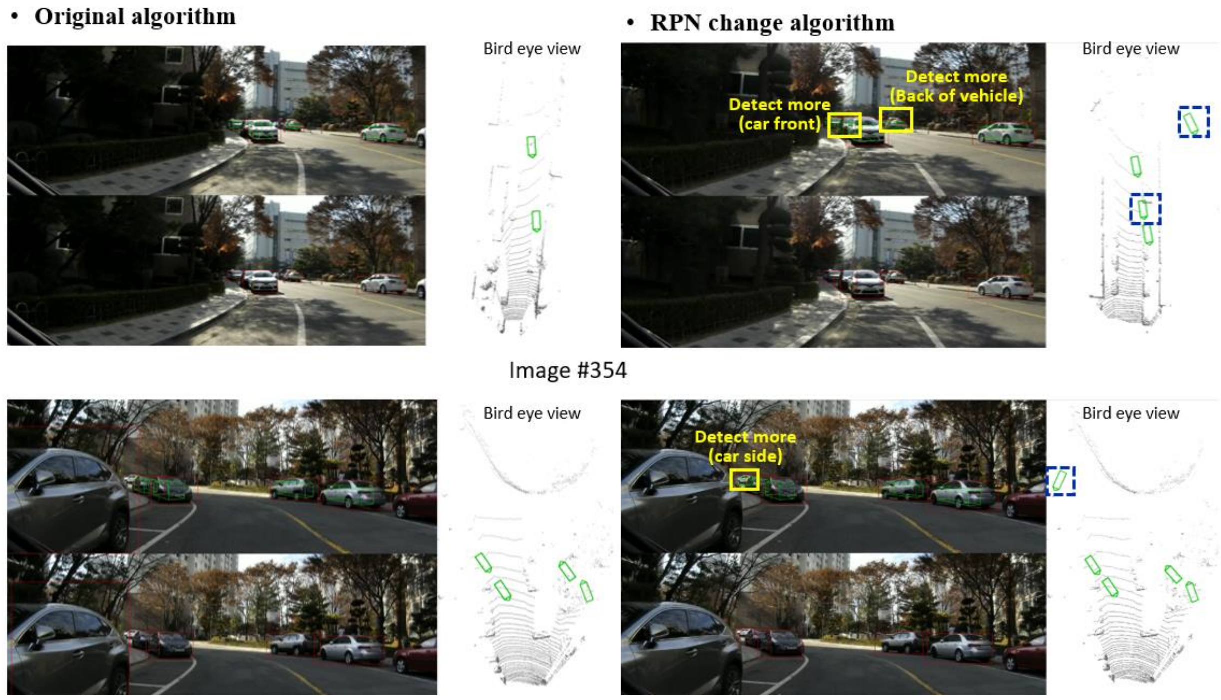


Image #354





[Conclusions & Further Work]

1. Conclusions

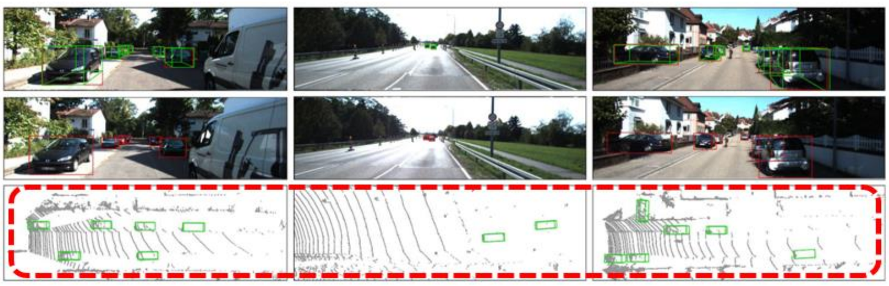
   Better object detection performance despite the reduced number of RPNs (9 → 8). (Image #122, #137, #345) Vehicles are better detected when they are evenly clustered.

   (Image # 137, #345, #370) Find more objects in the distance

   However, the lack of training and testing on wide objects such as airplanes is a limitation. In conclusion, Using RPNs that is appropriate for your environment, rather than a large number of RPNs, will perform better on objects.

2. Further work

   1) Improve with robust stereo vision algorithm even in bad environment (rain, snow, and night)
   2) Create the stereo-camera version of YOLO, and SSD algorithm
   3) Since the collected data doesn't contain noise, it is considered that applying the filtering process will be meaningless to confirm our algorithm. So implement that later
   4) Try to get test results with bird eye view by extracting the LiDAR point-cloud data



[Contribution]

Ji-il Park: Overall project flow design, target algorithm Selection, Hardware design, Modify and test the RPN algorithm, Presentation material production

Hyunyong Jeon : Build a software setup(dataset creation environment), and create a dataset

Leon Thormeyer : Build a hardware setup, participation in image shooting, and presentation.