Technical Report: Test 2

SkyNet

1. Methods

Since a delay in perception propagates heavily onto subsequent models such as localization, path planning, and controls, we decided early on to focus more on inference speed rather than accuracy of intersection of union (IoU) scores. As long as sufficiently accurate locations (~IoU 0.5) for the gates can be correctly identified, reasonably large flyable regions could be extracted from the scene at any time step. From here, viable path planning waypoints can be formed that go through the center of the detected flyable regions. Hence, after analyzing multiple models such as YOLO variants [1], SSDs [2], and R-CNN variants [3], we chose Mask R-CNN [4] based on increased accuracy as well as the superior ability compared to previous versions in detecting smaller objects. This becomes useful when detecting gates at a greater distance. Mask R-CNN is one variant of Fast R-CNN, the key difference being an additional branch for predicting an object mask in parallel with an existing branch for bounding box recognition. Typically these R-CNN approaches are slow unless properly optimized (i.e. via TensorRT) and parallelized well, but our implementation is fast enough to account for this.

Our implementation is based on a Mask R-CNN general framework for object instance segmentation. The COCO dataset was initially used, and transfer learning was then employed on the set of pre-trained weights in order to more accurately perform the instance segmentation. Figure 1 shows a bounding box placed successfully over the flyable region of a gate within an image. As shown, the instance segmentation of Mask R-CNN outputs pixel level segmentation. This non-rectangular segmented data can be utilized to easily increase the IoU accuracy by at least a factor of 10% or more. Unfortunately we were unable to complete this within the deadline, but we plan to do it in our future works.

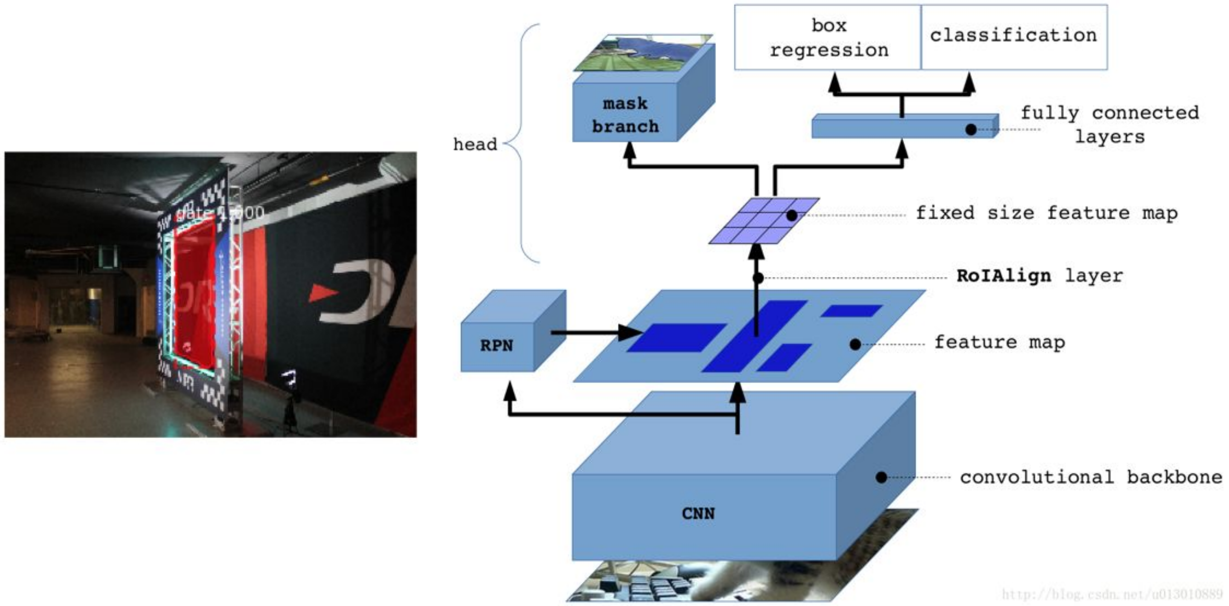

Figure 1: (left) A red bounding box around the flyable region of a gate, seen at a sharp angle. (right) a diagram of the Mask R-CNN architecture used in this approach ([4])

Our network was trained on two classes (gate and background). The images for training and testing were resized to 832x832x3 and we used Resnet50 as the backbone with anchors at 8, 16, 32, 64, and 128. We used a region of interest (ROI) of size 32 per image during training.

2. Future Works

Our team has several proposals for improving our results for more accurate gate detections and their application to real-life drone racing. To start off, data augmentation and analysis of data distribution will be important for further improve the accuracy of our approach. For this task all gates had one same real shape. However in a real drone race, the gates could have different shapes, sizes, and colors. Hence proper data augmentation will be required in order for our model to generalize to many types of gates that may be present in a race.

Subsequent model optimization will be done using CUDA or Tensor-RT optimization, which will shave off critical milliseconds from our computation time per image frame and increase the frame rate we will need to race. Additional model architecture refinement may also improve latency and accuracy. There is a natural tradeoff between the model accuracy and the latency, which we will seek to exploit. We intend to find this ideal tradeoff point once detection and localization/control are combined in silico. Altering the size of the image or using a different backbone network that is heavier or lighter will be studied. The Mask R-CNN inference time for a larger image, as well as a deeper backbone network, will generate a more accurate position in general; however, a smaller image or shallower backbone will allow for much faster computation of the bounding region. We also will not limit ourselves to Mask R-CNN, but will investigate other models that we deem beneficial and give us the mix of accuracy and speed we desire.

We will also take advantage of the fact that our current implementation does instance pixel-wise segmentation. After dividing the detected region into four quadrants, we can deduce the corners of the polygon by finding the point with the most euclidean distance from the center of the instance detection. This will further improve our IoU by allowing for non-rectangular flyable regions. We can also account for situations where there are obstacles such as pillars interrupting the flyable region by collecting all region instances located within certain threshold distances and treating them as belonging to a common flyable region, then finding the four corners as mentioned above. This will increase our IoU accuracy significantly, which can even enable us to estimate the distance to each gate by utilizing the homography of the detected gates.

For detecting distances to each gate as well as pixels, they can be easily computed using semi-global mapping (SGM) on image regions found to be within the frustum of the flyable zone. Using 2D box location along with distance information, the accuracy of our localization routine will improve substantially. However, if SGM methods are prove too computationally expensive for this challenge (particularly on the target hardware platform, NVidia Jetson), monocular visual odometry and depth estimation is a an alternative we will seek. Stereo camera SGM could also be used to generate training datasets for depth estimation training. Furthermore, we are aware that a combination approach utilizing both stereo imaging and monocular depth estimations can be combined to gain further accuracy [5].

We also wish to explore a variety of architectures for multi-task networks to regress the distance of objects and their detections as this technique is known improve both the distance estimation and the object detection accuracy in similar applications.

3. References

[1] Redmon, Joseph et. al. *You Only Look Once: Unified, Real-Time Object Detection*. CVPR (2016).

[2] Liu, Wei; Anguelov, Dragomir; Erhan, Dumitru; Szegedy, Christian; Reed, Scott; Fu, Cheng-Yang; Berg, Alexander. *SSD: Single Shot MultiBox Detector*. ECCV (2016).

[3] Girschick, Ross. *Fast R-CNN*. ICVV (2015).

[4] He, Kaiming; Gkioxari, Georgia; Dollar, Piotr; Girshick, Ross. *Mask R-CNN*. ICCV (2017).

[5] Saxena, Ashutosh; Schulte, Jamie; Ng, Andrew. *Depth Estimation using Monocular and Stereo Cues*. IJCAI (2007).