Module 1: Object Detection -- Part B

CSC490H1 2022: Making Your Self-driving Car Perceive the World

Gongyi Shi, Anny Runxuan Dai

Part B : Exploration

Motivation

There are two problems we try to tackle in Part B: (i) our target heatmap only counts for the centroid of vehicles, while other features, such as size and heading, are also critical factors for auto-driving planning. We choose to scale the isotropic Gaussian kernel based on the covariance matrix of x and y coordinates to count for the heading and size in heatmap; (ii) our heatmap loss function, weighted mean squared error (MSE), can be improved by more sophisticated loss functions. In our study, we explore the performance of our model based on Focal Loss [3] and MSE with hard mining [4].

There are many other possible approaches to the two problems. For the first problem, we can replace the loss functions with more sophisticated loss functions for heading and size channels, instead of combining the two in the heatmap. Furthermore, we can also have pre-set heading and size knowledge based on additional info. For instance, vehicles' heading tends to be parallel to the street. If we are given an HD map, we can set prior knowledge of vehicles' headings. For the second problem, we can preprocess the target, labeling vehicles that are not detected by the LiDAR point cloud (See Fig. 1), and weight them less in the loss function.

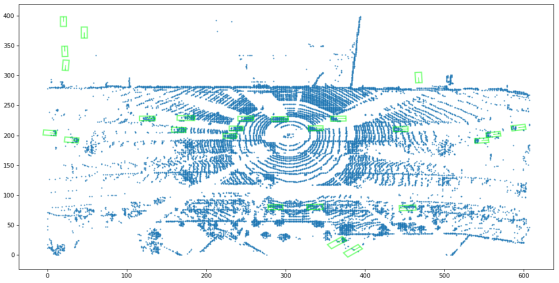


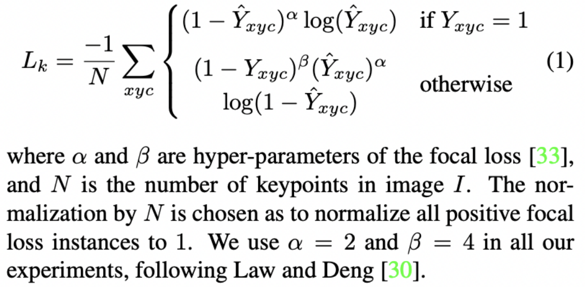
Fig. 1. A selected target point cloud frame. Notice the four vehicles on the upper left corner are entirely invisible to the LiDAR.

The most important reason we choose our approach on the first problem is simply the dataset does not have extra info to build the priors. While exploring other loss functions may be promising, we are more interested in exploring the cross-impact among channels, as individual channel loss functions are tested in the second problem. We choose to test on Focal Loss and MSE with hard mining, because both loss functions increase the weight of hard examples, forcing the training to focus more on false positive and true negative eliminations. Thus, we suggest the average precision may converge faster than the original model.

Techniques

In our model, we will be using Focal loss, MSE with hard mining, and modified Gaussian kernel. See `loss_target.py` and `loss_function.py` for our implementations.

For focal loss, we will be using the following equation quoted from "Objects as Points"[1].



$$L_k = \frac{-1}{N}\sum_{xyc}\begin{cases}(1-\hat{Y}_{xyc})^{\alpha}\log(\hat{Y}_{xyc}) & \text{if } Y_{xyc}=1\\ (1-Y_{xyc})^{\beta}(\hat{Y}_{xyc})^{\alpha} & \text{otherwise}\\ \quad\log(1-\hat{Y}_{xyc}) & \end{cases} \tag{1}$$

where α and β are hyper-parameters of the focal loss [33], and N is the number of keypoints in image I. The normalization by N is chosen as to normalize all positive focal loss instances to 1. We use $\alpha = 2$ and $\beta = 4$ in all our experiments, following Law and Deng [30].

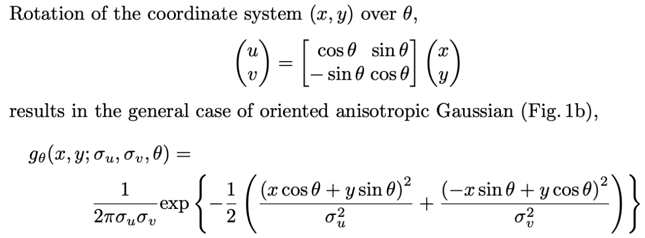
This equation is a modified version of the binary focal loss function that considers the values in the heatmap. We will be testing on four different pairs of alpha and betas to find the best pair of values of the parameters.

For MSE with hard mining, we take the average of the top k difference of the MSE loss. We test on three different values of hyperparameter *k*.

For the anisotropic Gaussian kernel, we will be using a modified version of the previous equation from part A to get the values in the heatmap.

$$\mathcal{Y}^{(n)}_{\text{heat},i,j} = \exp\left(-\frac{(x_n-i)^2+(y_n-j)^2}{\sigma}\right)$$

We have modified the following equation from "Fast Anisotropic Gauss Filtering." [2]:



Rotation of the coordinate system (x, y) over θ,

$$\begin{pmatrix}u\\v\end{pmatrix} = \begin{bmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{bmatrix}\begin{pmatrix}x\\y\end{pmatrix}$$

results in the general case of oriented anisotropic Gaussian (Fig. 1b),

$$g_\theta(x, y; \sigma_u, \sigma_v, \theta) = \frac{1}{2\pi\sigma_u\sigma_v}\exp\left\{-\frac{1}{2}\left(\frac{(x\cos\theta + y\sin\theta)^2}{\sigma_u^2} + \frac{(-x\sin\theta + y\cos\theta)^2}{\sigma_v^2}\right)\right\}$$

, where x and y are the differences between the x, y coordinates of the pixel and the centroid.

And our final equation is:

$$exp[-\ (\frac{((x_p - x_c)cos\theta + (y_p - y_c)sin\vartheta)^2}{\sigma_x} + \frac{(-(x_p - x_c)sin\theta + (y_p - y_c)cos\vartheta)^2}{\sigma_y})$$

Where $\sigma_x = \frac{\sigma \cdot xsize}{2}$ and $\sigma_y = \frac{\sigma \cdot ysize}{2}$ are the scales we choose, and $\sigma$is the scale factor in Part A.

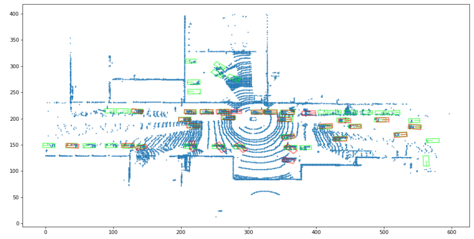
Evaluation

Due to limited computational resources, it is hard to train the model until convergence. Practically, we choose to set up a baseline of AP, and we will compare the number of iterations of different methods that first exceed 0.2. The parameters that generate this result with the lowest amount of epoch will be our best choice. For hyperparameter tuning in Focal Loss and MSE with hard mining, if exceeding 0.2 at the same epoch, we choose the set of hyperparameters with the largest AP.

For Focal Loss, based on Figure 1.1, we choose alpha = 2, beta = 4 as the hyperparameters.

alpha	beta	epoch	AP
2	4	5	0.26
2	2	7	0.26
4	2	7	0.20
4	4	6	0.21

Fig. 2.1 Resulting AP table with different sets of hyperparameters and corresponding epoch



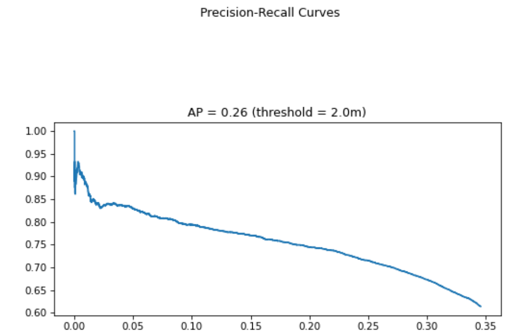


Fig. 2.2 Detection and PR curve with epoch = 5, alpha = 2, beta = 4 on input frame 000 and τ = 2.0m

For MSE with hard mining, based on Figure 2.1, we choose k = 150 as the hyperparameter.

k	epoch	AP
50	7	0.21
100	7	0.24
150	7	0.25

Fig. 3.1 Resulting AP table with different sets of hyperparameters and corresponding epoch

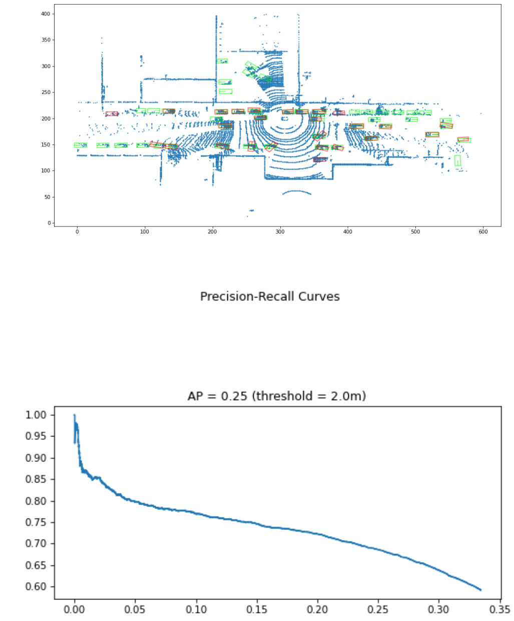


Fig. 3.2 Detection and PR curve with epoch = 7, k = 150 on input frame 000 and τ = 2.0m

While sharing the same number of iterations, we can see that the heading prediction of the anisotropic Gaussian kernel (Figure. 4) is more accurate than the MSE with hard mining (Figure 3.2).

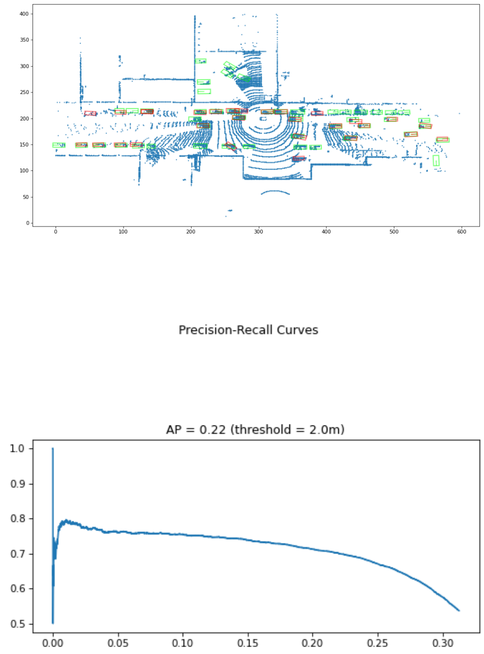


Fig. 4 Detection and PR curve with the anisotropic Gaussian approach at epoch = 6 on input frame 000 and τ = 2.0m

To compare with part A, the AP first exceeds 0.2 with a value of 0.27 at epoch 6.

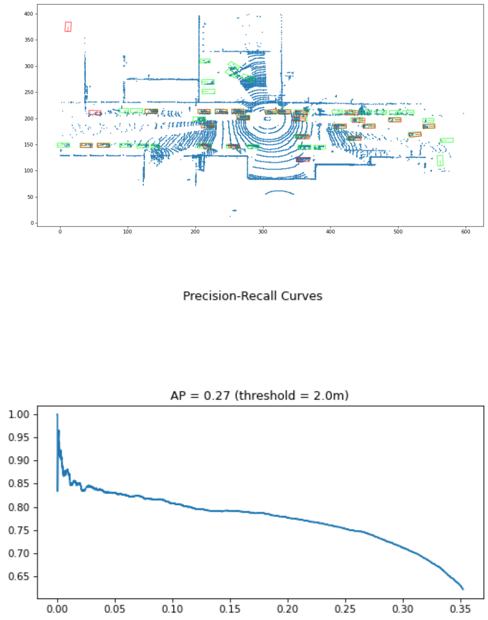


Fig. 5 Detection and PR curve for part a at epoch = 6 on input frame 000 and τ = 2.0m

In our quantitative test standard, we can conclude that Focal loss with alpha=2, beta=4 is our current best model. It reaches the baseline of an AP of 0.2 first at epoch 5. Surprisingly, for our MSE with hard mining and anisotropic Gaussian approach, we did not see any improvements with our current parameters, by comparing the number of iterations that first reach AP = 0.2.

Limitations and Future Works

Comparing AP only gives us information about centroids and predictions matching, ignoring the losses of size and heading. We could improve by comparing the overall features' losses with the simple MSE loss function for a given epoch or even converged model, to yield the best metric.

The most intuitive limitation of our approach is that we only have little hyperparameter tuning as we have only tested on a few combinations. We have thought about using ray tune to get the best parameters within a proper range, and smaller steps, but due to limited computational resources, using ray tune is much more time-consuming.

Additionally, as we suggested, we can also filter out the non-detected targets in the LiDAR point cloud in the preprocessing stage to improve the performance of the model detection, as such detections are not necessarily responsible for the LiDAR.

For anisotropic Gaussian method, when we modify the heatmap, our pre-set scaling factors might not be of the best fit. If the area above the threshold is larger than the actual object, multiple objects with close distance might result in dense and hard-to-distinguished shapes, due to overlapping (See Figure 4.). Such a pattern might affect the model's ability to identify the correct number, size and heading of targets. Currently, our size is a little larger than the actual cars, we think that making the size of the significant area about the same as the actual targets should make a better performance.

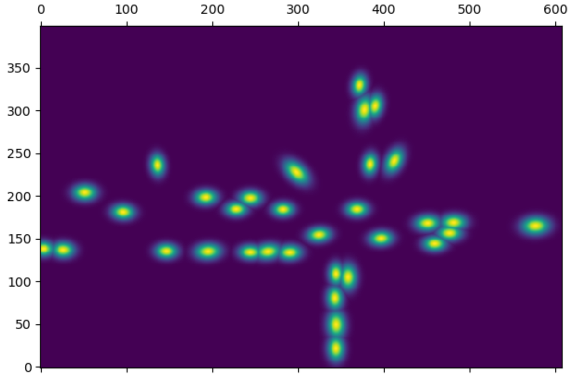


Fig. 6 The target heatmap using the anisotropic Gaussian approach with our scaling factors. Notice that the shape of the distribution is changed by overlapping boundaries, which makes the heading and size of the dense-target area harder to identify.

Furthermore, currently our implementation uses Gaussian distribution to create the heatmap, while the shapes of the targets are rectangles. We can choose to use other distributions that have similar shapes with the targets, while maintaining peak value at target centroids.

References:

[1] Zhou, Xingyi, Dequan Wang, and Philipp Krahenb. “Objects as Points.” arxiv.org, April 25, 2019. https://arxiv.org/pdf/1904.07850.pdf%5D.

[2] Geusebroek, Jan-Mark, Arnold W. M. Smeulders, and Joost van de Weijer. “Fast Anisotropic Gauss Filtering.” springer.com, 2002. https://link.springer.com/content/pdf/10.1007%2F3-540-47969-4_7.pdf.

[3] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Doll ́ar. Focal loss for dense object
detection. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October*
22-29, 2017, pages 2999–3007. IEEE Computer Society, 2017.

[4] Shrivastava, A., Gupta, A., & Girshick, R. (2016). Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 761-769).