Focal Gradient Harmonized Loss Function for Object Detection

Muye Sun¹, Hongbo Huang²*, Zhiying Hu¹, Qinglin Ran³ and Yahui Liu⁴

¹School of Computer Science, Beijing Information Science and Technology University, Beijing, 100101, China
²Institute of Computing Intelligence, Beijing Information Science and Technology University, Beijing, 100192, China
³School of Electromechanical Engineering, Beijing Information Science and Technology University, Beijing, 100192, China
⁴School of Information Management, Beijing Information Science and Technology University, Beijing, 100192, China

*Corresponding author’s e-mail: hhb@bistu.edu.cn

Abstract. Mainstream detectors in the field of object detection usually can be divided into one-stage and two-stage detectors. Compared to the two-stage detectors, one-stage detectors are more efficient and elegant, but are more easily disturbed by imbalanced data during the training. Inspired by the gradient harmonized single-stage detector (GHM detector)[1], we propose a focal gradient harmonizing loss function for classification (FGHM-C). FGHM-C loss function optimizes the method of generating example weights during training, thereby alleviating the impact of long-tailed distribution. Our proposed model achieves 42.0 mAP on COCO test-dev set, which is 0.4 points higher than the original GHM detector.

1. Introduction
In the field of object detection, the one-stage detectors have been greatly developed in recent years. It’s faster and simpler, but its accuracy is inferior to the two-stage detector. The main problem that leads to this phenomenon is the serious imbalance between easy and hard, positive and negative examples. Most images contain a lot of background (negative) and easy examples. They overwhelm important examples, thus affecting the training process and reducing the model performance. In order to solve this problem, a large number of methods have been proposed in recent years. Example mining based methods such as OHEM[2] pays more attention to hard examples, but abandon some easy examples, which will cause the distribution of data to be changed, thereby impairs the ability of the model to distinguish easy examples. For the positive and negative imbalance, focal loss[3] achieved remarkable results, but it needs to adjust the parameters for different situations, making the training process complicated. Inspired by focal loss, GHM detector[1] was proposed to solve this imbalance problem and reduce parameter adjustments. They pointed out that the class imbalance problem can be summarized to the gradient norm distribution’s imbalance. The GHM detector determines the weight of the example by attaching a harmonizing parameter to the gradient of each example according to the gradient density. However, in the training process, there are still some defects in the approximate estimation of the gradient density.
In this work, we proposed FGHM-C to replace GHM-C as the novel classification loss function. In our approach, we improved the approximate estimation method of the gradient density. This method generates unequal length regions based on the distribution of gradient norm to estimate the gradient density better. After a simple tuning of parameters, our model’s mAP is 0.4 points higher than origin GHM detector on COCO test-dev set[4].

2. Related Work

In many real-world scenarios, the data is often imbalanced. Such as there is a class of data in the dataset that contains far more data than the other classes. When the distribution related to input attributes affects performance, imbalance problems arise. In the last few years, imbalance learning in object detection at several levels has received a lot of attention[2][5][6][7]. To achieve optimal training and deeply exploit the potential of model architectures, it is crucial to reduce the imbalance in training process.

2.1. Focal Loss

In the field of object detection, the one-stage detector has always been disturbed by the foreground-background imbalance. It used the concept of anchor boxes from two stage detectors like Faster R-CNN. For two-stage detectors, the region proposal networks[8] filters a large number of background anchors, which greatly eases positive and negative imbalance. Without region proposal networks, the one-stage detector considered the entire feature map as region proposals, therefore implanted a large number of background class anchors. Focal loss[3] was proposed to solve this problem in an elegant way. It reformulated the cross-entropy loss function by introducing two parameters $\alpha$ and $\beta$. This method reduced the loss of easily classified examples i.e. background class. Compared with OHEM[2], focal loss does not drop examples, so it avoids changing the data distribution.

2.2. Gradient Harmonized Single-Stage Detector

Focal loss has achieved good results in eliminating the foreground-background imbalance, but hyper-parameters need to be tuned according to different data distribution. The design of focal loss is data-independent and lacks exploration of data distribution, which is crucial for balancing foreground-background imbalance. Gradient harmonized single-stage detector was proposed to solve these problems. The imbalance of class and difficulty can be considered as the imbalance of the gradient norm distribution. Gradient norm represents the classification difficulty of the example. Therefore, the gradient density is defined to measure the distribution of examples’ gradient norm. Fig.1[1] shows in a converged one-stage detector, the numbers of easy examples is huge, and the number of extremely hard examples is slightly higher than the number of moderately difficult examples. The author proposed a gradient harmonizing mechanism (GHM) to efficiently train a one-stage model that takes into account the weights of examples with different gradient norm. After using GHM, a large number of easy examples’ weight can be greatly reduced, and the outliers can also be relatively reduced.

![Fig.1 The distribution of the gradient norm from a converged single stage detection model. Since the easy negatives have a dominant number, log axis is used to display the fraction of examples.](image-url)
3. Methodology

3.1. Gradient Harmonizing Mechanism

In the original GHM detector, a gradient density function is defined, which is used to calculate the gradient density of the examples, thereby attaching a harmonizing parameter. The definition process of the original paper[1] is as follows.

In classification task, the cross-entropy loss is often used, \( p \in [0,1] \) is the predicted probability, and \( y \in \{0,1\} \) is the ground-truth label of the examples.

\[
CE(p,y) = \begin{cases} -\log(p) & y = 1 \\ -\log(1-p) & y = 0 \end{cases}
\]  

(1)

Assume that \( x \) is the model’s output, \( p = \text{sigmoid}(x) \). The definition of gradient norm \( g \) is as follows:

\[
\begin{align*}
g &= |p - y| = \begin{cases} 1 - p & y = 1 \\ p & y = 0 \end{cases} \\
g = |p - y| = \begin{cases} 1 - p & y = 1 \\ p & y = 0 \end{cases}
\end{align*}
\]  

(2)

Gradient density function is defined as:

\[
GD(g) = \frac{1}{\varepsilon} \sum_{i=1}^{N} \delta_{\varepsilon}(g_k, g)
\]  

(3)

\( g_k \) represents the gradient norm of the \( k \)-th example.

\[
\delta_{\varepsilon}(g_k, g) = \begin{cases} 1 & g - \varepsilon < g \leq g + \varepsilon \\ 0 & \text{otherwise} \end{cases}
\]  

(4)

\( GD(g) \) represents the number of examples in a region centered on \( g \) with a length of \( \varepsilon \) and normalized by the length of the region.

In order to reduce the amount of calculation, divided the range space of \( g \) into \( M \) unit regions equally, and the length of one region \( \varepsilon = \frac{1}{M} \). Then count the examples lying in every region. Thus, the gradient density can be approximated. After approximate estimation, the gradient density is defined as:

\[
GD(g) = \frac{R_{\text{ind}(g)}}{\varepsilon}
\]  

(5)

\[
\text{ind}(g) = t \quad (t-1)\varepsilon \leq g < t\varepsilon
\]  

(6)

\( R_{\text{ind}(g)} \) denote the number of examples with gradient norm \( g \) lying in region which index = \( \text{ind}(g) \).

The GHM-C loss function is defined as:

\[
L_{\text{GHM-C}} = \sum_{i=1}^{N} \frac{CE(p_i, y_i)}{GD(g_i)}
\]  

(7)

It can be seen that the gradient of the example is inversely proportional to the gradient density. This suppresses the gradients of easy and extremely hard examples.

3.2. Focal Gradient Harmonized Loss Function

In the GHM-C loss function, the unit regions are divided equally. Gradient density can be estimated by calculating the number of examples lying into each region. During the experiment, adjusting the number of regions \( M \) will have a certain impact on performance. As the \( M \) value rises, so does the performance, until \( M = 30 \)[1]. If \( M \) continues to rise, performance will decrease. The reason behind this phenomenon is that if there are too few unit regions, the granularity of the gradient density will be relatively low, resulting in insufficient accuracy. And if there are too many unit regions, the discontinuous examples distribution will cause the density of some regions to lose representativeness,
and this problem will be more obvious in regions with higher gradient density. It can be seen that it is necessary to reconsider the method of dividing the areas, that is, to have a smaller granularity, and to avoid generating regions with abnormal density. To this end, we designed the FGHM-C loss. This loss function modified the method of region generation in the GHM-C loss.

First divide the range space of g into two parts: scatter region areas (scatter areas) and focal region areas (focal areas). Scatter areas correspond to the area where the gradient norm distribution of the examples is dense, i.e. areas where very easy examples and extremely hard examples are concentrated. The remaining area is focal areas. Fig.2 shows the red area as the focal areas, and the blue areas as the scatter areas. There are many regions inside these two areas. Our method is to make the regions in focal areas shorter than the regions in scatter areas. Because the gradient density of examples in focal areas is relatively smaller and smoother, it will be less affected by the discontinuous examples’ distribution. And the regions in scatter areas will have a longer length to make the result more representative.

The focal region areas are determined by two parameters, \( s_l \) and \( s_h \). The range of the focal region area is \( [s_l, s_h] \). Therefore, the range of the scatter area is \( [0, s_l] \cup [s_h, 1] \). Define \( L_f = (s_h - s_l) \) as the whole length of the focal areas, \( L_s = (1 - L_f) \) as the whole length of the scatter areas, \( r \) as the ratio of region length inside scatter areas and focal areas, means if the length of the focal areas’ region is \( l \), the length of the scatter areas’ region is \( rl \). And there are \( F_{\text{num}} \) regions in focal area and \( S_{\text{num}} \) regions in scatter area. Next we calculate \( F_{\text{num}} \) and \( S_{\text{num}} \). \( M \) is the total number of the regions.

\[
F_{\text{num}} = \frac{M \cdot rL_f}{rL_f + L_s}
\]

\[
S_{\text{num}} = M - F_{\text{num}}
\]

Finally we divide the focal area \( [s_l, s_h] \) into \( F_{\text{num}} \) regions equally, and the scatter area \( [0, s_l] \cup [s_h, 1] \) into \( S_{\text{num}} \) regions equally. In this way, we have \( M \) regions in which the length of the regions in scatter area is \( r \) times longer than the regions in focal area. We define all these regions as FR, the gradient density can be redefined as:

\[
\tilde{GD}(g) = \frac{FR_{\text{ind}(g)}}{e_{\text{ind}(g)}}
\]

\( FR_{\text{ind}(g)} \) denote the number of examples with gradient norm \( g \) lying in region which index = \( \text{ind}(g) \). \( e_{\text{ind}(g)} \) is the length of the region with index = \( \text{ind}(g) \). The FGHM-C loss function is defined as:

\[
L_{\text{FGHM-C}} = \sum_{i=1}^{N} \frac{CE(p, y)}{GD(g)}
\]

Fig. 2 Our proposed FGHM-C loss function defines the red area as the focal areas and blue areas as the scatter areas.
4. Experiments

4.1. Dataset and Evaluation
We implemented experiments on mmdetection[9]: an open source object detection toolbox based on PyTorch. Our model is trained and evaluated on the challenging MS COCO[4] dataset. The COCO dataset is a large-scale object detection, segmentation and captioning dataset. It contains 115k images for training (train-2017) and 5k images for validation (val-2017). There are also 20k images without disclosed labels in test-dev. The model is trained on train-2017, validated on val-2017 and tested on test-dev set. Our reported results including mAP, $\text{AP}_{0.5}$ (AP at IoU=0.50), $\text{AP}_{0.75}$ (AP at IoU=0.75), $\text{AP}_s$ (small objects), $\text{AP}_M$ (medium objects) and $\text{AP}_L$ (large objects).

Table 1. Results of varying number of regions for FGHM-C loss.

| M  | AP  | $\text{AP}_{0.5}$ | $\text{AP}_{0.75}$ | $\text{AP}_s$ | $\text{AP}_M$ | $\text{AP}_L$ |
|----|-----|-------------------|-------------------|--------------|--------------|------------|
| 15 | 36.7| 54.9              | 39.1              | 19.7         | 39.9         | 48.3       |
| 20 | 37.1| 55.5              | 39.5              | 20.2         | 40.7         | 48.6       |
| 25 | 37.3| 55.9              | 39.8              | 20.5         | 40.7         | 49.7       |
| 30 | 37.4| 55.9              | 39.8              | 20.5         | 40.4         | 50.0       |
| 35 | 37.3| 55.9              | 39.6              | 20.7         | 40.6         | 49.3       |

4.2. Implementation Details
RetinaNet [3] and ResNet-50 with FPN[7] is used as network structure and backbone network. We implemented FGHM-C to replace GHM-C loss function, and retain the GHM-R loss function for regression. The focal area parameters $s_l$ and $s_h$ are set to 0.2 and 0.9 respectively according to the gradient norm distribution in Fig.1. The ratio parameter $r$ are set to 2.0. Multiple M values are set and evaluated. The results are shown in Table 1. When M = 30, the model shows better performance compared to other settings. After evaluating the model on the validation set, we use the 32x8d FPN-ResNeXt-101 and RetinaNet model to train on train-2017 and test on COCO test-dev set.

4.3. Optimization
We use the stochastic gradient descent (SGD) algorithm to optimize the model. The model is trained on RTX 2080 Ti, and mini-batch size is 2 images. All models are trained for 12 epochs. The initial value of the learning rate is set to 0.00125, and divide it by 0.1 at 9th and 12th epoch. Other parameters follow the default settings of mmdetection [9] if not specifically noted.

4.4. Main Results
The baseline setting follows the GHM detector. Table 2 shows the result of our FGHM-C with M = 30. We can see a gain of 0.4 mAP compared with the best configured GHM detector. Table 3 shows our result on test-dev set, mAP also achieved an increase of 0.4. Compared with GHM detector, our model has improved on $\text{AP}_s$ and $\text{AP}_M$, but decreased on $\text{AP}_L$, indicating that our model is better at detecting small and medium objects. Judging from the accuracy of different object scales, the accuracy of our model has improved more balanced than GHM detector compared to the baseline. This proves that our FGHM-C loss function approximates the gradient density better than the original GHM detector.

Table 2. Comparison with baseline on COCO val-2017.

| method           | AP  | $\text{AP}_{0.5}$ | $\text{AP}_{0.75}$ | $\text{AP}_s$ | $\text{AP}_M$ | $\text{AP}_L$ |
|------------------|-----|-------------------|-------------------|--------------|--------------|------------|
| Focal Loss + SL1 | 35.6| 55.6              | 38.2              | 19.1         | 39.2         | 46.3       |
| GHM-C + SL1      | 35.8| 55.5              | 38.1              | 19.6         | 39.6         | 46.7       |
| GHM-C + GHM-R    | 37.0| 55.5              | 39.2              | 20.4         | 40.3         | 49.1       |
| FGHM-C + GHM-R   | 37.4| 55.9              | 39.8              | 20.5         | 40.4         | 50.0       |
5. Conclusion and Future Work

In this paper, we focused on the training process of the GHM detector and found that because of the approximate accuracy problem in the training process, the potential of the model architecture has not been fully exploited. Based on this discovery, we propose FGHM-C loss function to enhance the performance of the model by optimizing the approximation method. Experiments show that our model achieves better performance than GHM detector.

Despite our method has made performance improvements, there are still some shortcomings: 1) regions generation still relies on the set parameters. We are trying to dynamically generate the regions based on the data distribution, but we have not yet achieved good results. 2) for imbalanced data, it is difficult to define the optimal distribution of gradient and still needs to be explored.

Acknowledgments

This work is supported by Beijing municipal education committee scientific and technological planning project (KM201811232024, KM201611232022) and Beijing excellent talents youth backbone project (9111524401).

References

[1] Li, B., Liu, Y., & Wang, X. (2019). Gradient harmonized single-stage detector. In Proceedings of the AAAI Conference on Artificial Intelligence. Hawaii. pp. 8577-8584.
[2] 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. Las Vegas. pp. 761-769.
[3] Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision. Hawaii. pp. 2980-2988.
[4] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In European conference on computer vision. Zurich. pp. 740-755.
[5] Sun, Y., Wong, A. K., & Kamel, M. S. (2009). Classification of imbalanced data: A review. International journal of pattern recognition and artificial intelligence, 23(04): 687-719.
[6] Pang, J., Chen, K., Shi, J., Feng, H., Ouyang, W., & Lin, D. (2019). Libra r-cnn: Towards balanced learning for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition. Los Angeles. pp. 821-830.
[7] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition. Hawaii. pp. 2117-2125.
[8] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems. Montreal. pp. 91-99.
[9] Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., & Li, X., et al. (2019). Mmdetection: open mmlab detection toolbox and benchmark.

| method                  | network             | AP   | AP_{0.5} | AP_{0.75} | AP_{S}  | AP_{M}  | AP_{L}  |
|-------------------------|---------------------|------|----------|-----------|---------|---------|---------|
| Faster RCNN             | FPN-ResNet-101      | 36.2 | 59.1     | 39.0      | 18.2    | 39.0    | 48.2    |
| GA-Faster-RCNN          | GA-Faster-RCNN      | 39.8 | 59.2     | 43.5      | 21.8    | 42.6    | 50.7    |
| Mask RCNN               | FPN-ResNeXt-101     | 39.8 | 62.3     | 43.4      | 22.1    | 43.2    | 51.2    |
| RDSNet                  | RDSNet              | 40.3 | 60.1     | 43.0      | 22.1    | 43.5    | 51.5    |
| Focal Loss              | RetinaNet-FPN-ResNeXt-101 | 40.8 | 61.1     | 44.1      | 24.1    | 44.2    | 51.2    |
| GHM-C + GHM-R           | RetinaNet-FPN-ResNeXt-101 | 41.6 | 62.8     | 44.2      | 22.3    | 45.1    | 55.3    |
| FGHM-C + GHM-R          | RetinaNet-FPN-ResNeXt-101 | 42.0 | 61.4     | 45.0      | 24.0    | 45.4    | 52.9    |