What is the Key to Preventing Training Failures in Object Detection Models?

Insoo Kim
Aiffel research 12th
insookim43@gmail.com

Abstract

In object detection, a model must not only determine the location of bounding boxes but also perform classification tasks. If learning signals are not effectively transmitted to the model, training may fail. Various failure patterns can be observed depending on the characteristics of object detection models, including common deep learning failures such as underfitting and overfitting. This study analyzes these failure patterns and discusses appropriate hyperparameter tuning and engineering techniques to address them. The code is available on the website at https://github.com/insookim43/AIFFEL_quest_rs/blob/main/GoingDeeper/Ex05/face_detection.ipynb.

1 Introduction

Various failure patterns can be observed during the training process of object detection models.

Common failure cases Common failure cases in deep learning models include:

- Underfitting: When a model has not learned enough from the training data, resulting in poor performance.
- Overfitting: When a model adapts too much to the training data, leading to decreased performance on new data.

In object detection, a model must not only determine the location of bounding boxes but also classify objects. If learning signals are not effectively transmitted, the model may fail to learn.

Well-known engineering techniques To address these failures, various well-known engineering techniques are employed:

- Hard Negative Mining: Since the proportion of background is much larger than that of faces, this technique re-trains samples where the label is negative, but the confidence score is high. By doing so, the model is penalized more for false negative samples where non-face regions are misclassified as faces. This strengthens the model against false negatives near the ambiguous boundary between positive and negative samples.
- Learning Rate Warm-up: This method allows the learning rate to gradually increase in the early stages of training. Different learning rates are applied depending on the training step. Figure [ref].

Various failure patterns However, even with these techniques, various failure patterns may persist:

- Early stage: The model has not yet learned appropriate feature representations. Loss values are high, and predictions are close to random, often detecting only background.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

- Mid stage: As the model gradually learns objects, mean Average Precision (mAP) begins to improve. A balance between False Positives (FP) and False Negatives (FN) is required, and the risk of overfitting increases.
- Late stage: The model reaches near-convergence. Fine-tuning is necessary to optimize final performance, and misprediction causes must be analyzed. This paper discusses appropriate hyperparameter tuning and engineering strategies to address these issues.

2 Experimental Setup

Dataset The WIDER FACE dataset was used for facial detection experiments. This dataset contains images with varying scales, poses, and occlusions. It was randomly divided into train, validation, and test sets in a 40%/10%/50% ratio:

- Train set: img count: 12,880, box count: 159,397
- Validation set: img count: 3,226, box count: 39,697
- Test set: img count: 16,097

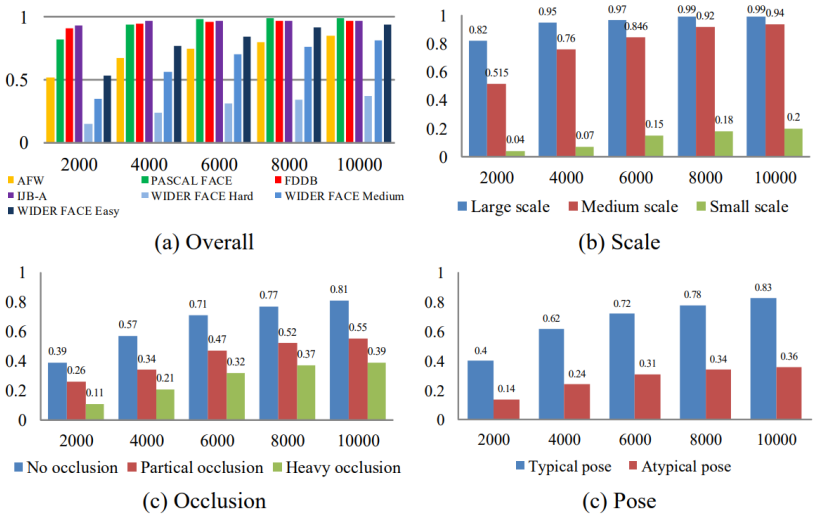

Figure 1: WIDER FACE variability.

Figure [dataset example]: The dataset consists of images containing faces of varying sizes.

Model We employed the Single Shot Detector (SSD), a one-stage detection model. SSD, similar to YOLO, is a high-speed detection model and is well-suited for the WIDER FACE benchmark, which contains a diverse distribution of face sizes.

The SSD model utilizes anchor boxes to enhance detection performance. By extracting anchor boxes from multiple layers of the feature map, SSD generates bounding box outputs at various scales. These outputs are compared with ground truth boxes, selecting those with high Intersection over Union (IoU) scores. This significantly improves speed compared to traditional sliding window-based R-CNN models.

Training Settings

- **Epochs:** 20
- **Optimizer:** Stochastic Gradient Descent (SGD)
- **Batch size:** 32
- **Loss functions:** Huber loss (L1 loss) and sparse categorical entropy loss

- **Hard Negative Mining:** The ratio of negative to positive samples was set to 3:1
- **Learning Rate Scheduler:** Learning rate warm-up was applied

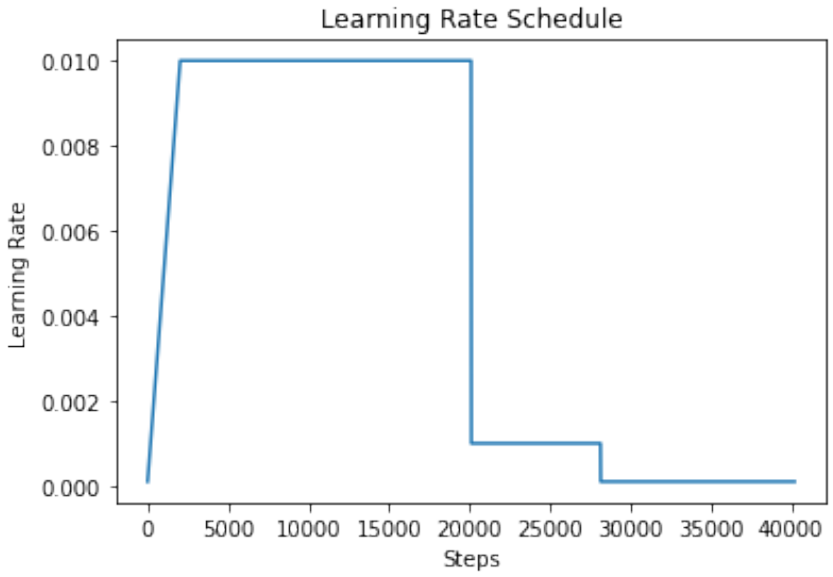

Figure 2: Example of an uploaded image.

To investigate the impact of different loss weightings, we experimented with the following weight configurations:

- **(class_weight: 1.0, box_weight: 0.01)**
- **(class_weight: 1.0, box_weight: 0.05)**
- **(class_weight: 1.0, box_weight: 0.1)**
- **(class_weight: 1.0, box_weight: 0.2)**
- **(class_weight: 1.0, box_weight: 0.5)**
- **(class_weight: 1.0, box_weight: 1.0)**
- **(class_weight: 1.0, box_weight: 2.0)**
- **(class_weight: 1.0, box_weight: 5.0)**
- **(class_weight: 1.0, box_weight: 10.0)**

Evaluation Metric

- **Mean Intersection over Union (IoU):** Used to assess localization error.

3 Analysis

3.1 Early-stage Failure Cases

Figure **??** illustrates common failure cases.

One common failure case occurs when one of the loss terms does not improve from its initial value.

Causes and Interpretation If a loss term does not decrease, it indicates that the model is relying on only one loss function. Figure **??** shows cases where localization loss and classification loss remain stagnant during training, along with the corresponding mean IoU performance over 20 epochs.

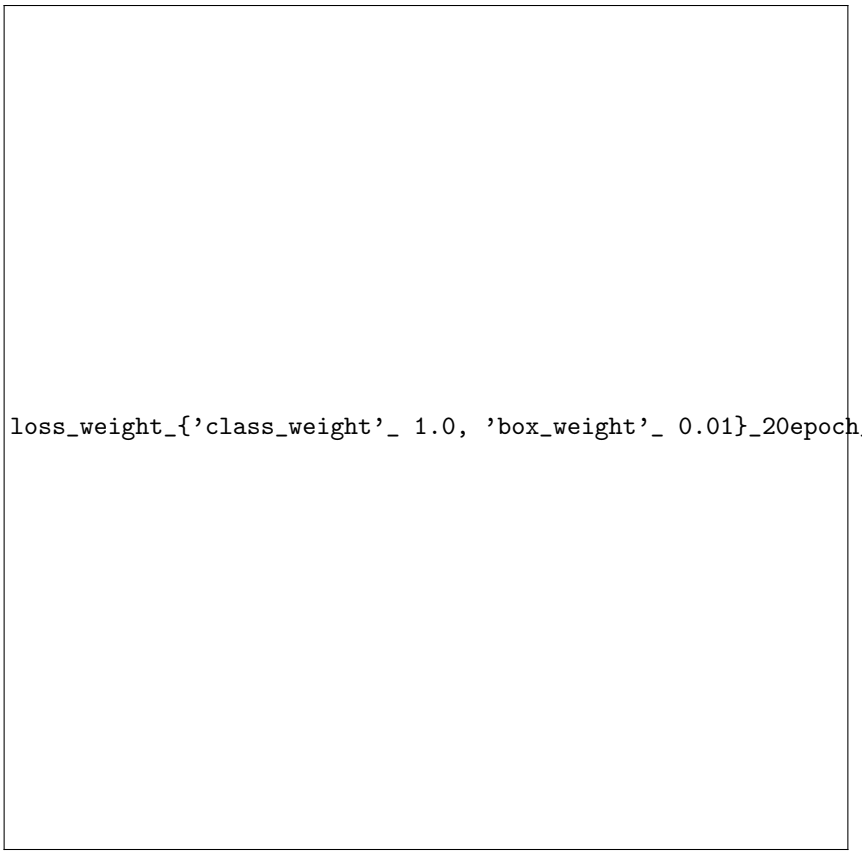

Figure 3: Common failure case

When the loss ratio is appropriate, both losses consistently decrease per iteration. In such cases, mean IoU continues to improve even after the rapid convergence phase, indicating that the model is effectively learning the localization task. This is evident in Figure **??**.

3.2 Mid-stage Failure Cases

Training Behavior During this phase, the mean IoU improves, but classification loss stagnates, and incorrect predictions persist with high confidence.

Causes and Interpretation Despite increasing mean IoU, qualitative analysis reveals that the model struggles with small object predictions and suffers from significant classification errors. The model remains underfitted at this stage.

Since validation performance continues to improve, the solution is to keep training the model until it reaches saturation. Once validation performance plateaus, training should be halted to prevent overfitting.

3.3 Late-stage Failure Cases

In the late stage, the model reaches near-convergence. Fine-tuning is necessary to optimize final performance, and the causes of mispredictions must be analyzed.

4 Conclusion

This study analyzed common failure cases in deep learning, such as underfitting and overfitting, as well as specific failure patterns unique to object detection models.

Since object detection requires both localization and classification, proper learning signals must be delivered to the model at each stage to avoid training failures. Various failure patterns were observed, each requiring appropriate engineering and hyperparameter tuning. The weighting ratio between loss terms plays a crucial role in model training, affecting both convergence speed and task performance.

To address underfitting, appropriate evaluation metrics must be used to diagnose the issue. Additionally, to prevent overfitting and maintain generalization performance, training should be stopped at an optimal point.

5 Future Research Directions

Further exploration is needed to determine optimal weight ratios for loss functions and investigate their impact on different datasets. Additionally, developing adaptive loss weighting strategies could improve model robustness and reduce the need for manual hyperparameter tuning.

Another potential research direction is improving failure detection methods in object detection training, enabling early intervention to prevent severe performance degradation. Lastly, exploring alternative architectures and optimization techniques may yield insights into mitigating the observed failure cases.

References

[1] Yang, S., Luo, P., Loy, C. C., Tang, X. (2016). WIDER FACE: A Face Detection Benchmark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 5525-5533). ResearchGate

[2] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 21-37).

[3] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 779-788).