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

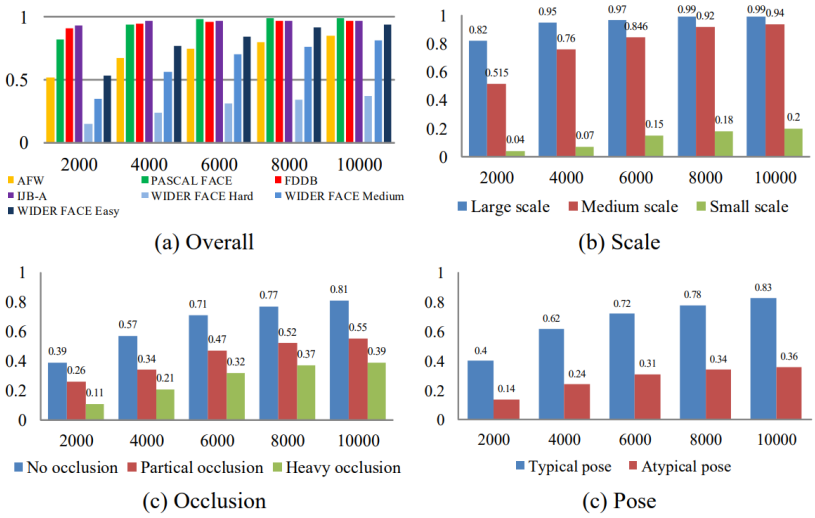

Figure 1: WIDER FACE variability.

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

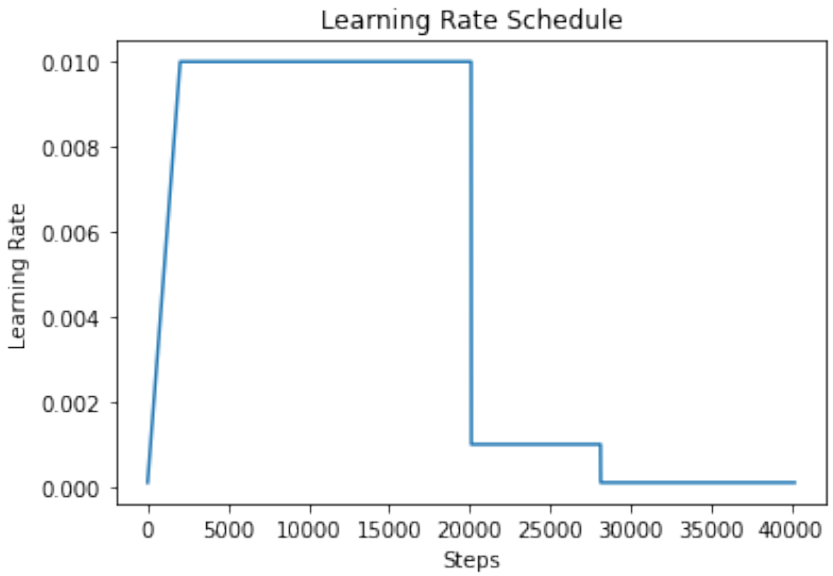

Figure 2: Example of an uploaded image.

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)**

Metric To objectively evaluate the performance of a model, various metrics are used. In object detection models, both localization performance and classification performance must be assessed.

Mean IoU (Intersection over Union, mIoU)

- **Definition:** IoU measures the overlap between the predicted bounding box and the ground truth box.
- **Formula:**

$$IoU = \frac{|PredictedBox \cap GroundTruthBox|}{|PredictedBox \cup GroundTruthBox|} \tag{1}$$

- **Description:**
  - IoU values close to 1 indicate a high degree of overlap between the predicted and ground truth boxes.
  - A predicted box is considered a True Positive (TP) when

$$IoU \geq 0.5. \tag{2}$$

  - Mean IoU (mIoU) calculates the average IoU across the entire dataset, providing an overall measure of localization performance.
  - A limitation of IoU is that detecting small objects often results in lower values, making it difficult to achieve high IoU.

TP, FP, FN (True Positive, False Positive, False Negative)

- **True Positive (TP)**
  - The model correctly detects an actual object.
  - A predicted box is considered a true positive if
$$IoU \geq 0.5. \tag{3}$$
- **False Positive (FP)**
  - The model incorrectly detects an object where none exists.
  - This occurs when
$$IoU < 0.5 \tag{4}$$
    or when a predicted box appears in a region without a ground truth object.
  - A high number of false positives leads to lower precision.
- **False Negative (FN)**
  - The model fails to detect an actual object.
  - This happens when a ground truth box exists, but either no prediction is made or
$$IoU < 0.5. \tag{5}$$
  - A high number of false negatives results in lower recall.

3 Analysis

3.1 Early-stage Failure Cases

3.1.1 Issues

- The loss value remains extremely high from the beginning and does not decrease.
- The model detects only the background or makes random predictions.

3.1.2 Known Solutions

- Apply learning rate warm-up.
- Initialize feature extraction layers with pre-trained weights.
- Utilize batch normalization to stabilize training.

3.1.3 Case Analysis

Initially, the model predicts bounding box scores randomly, leading to a large number of positive predictions and, consequently, a high loss value.

During the early phase of the first epoch, the model attempts to lower the box scores to reduce the loss by simply avoiding predictions. However, this reduction does not indicate meaningful learning but rather an avoidance strategy. At this stage, classification loss serves as the primary supervision signal, guiding the model to move away from random predictions and towards object detection.

In the early training stage, the model tends to generate **randomly placed large boxes**. As training progresses, it begins to produce **smaller boxes** as it refines its search patterns.

At this point, **L1 loss has little effect**, meaning that the model is primarily learning through classification loss rather than localization loss.

3.1.4 Key Failure Factors

- **Imbalance Between Box Loss and Classification Loss**
  If classification loss becomes too small, the model may fail to escape the early-stage learning failures.
- **Learning Direction of False Positive Boxes**
  If the model assigns high confidence scores to incorrect classes, it may prioritize reducing class confidence rather than improving localization.
  As a result, the model may become overly conservative in object predictions, failing to explore sufficiently.
  A well-designed box confidence reduction strategy and supervision adjustments are crucial in the early training phase.

3.1.5 Additional Refinements

- **Early Warm-up Phase**
  Introducing an initial warm-up phase can prevent the model from being overly conservative in predicting bounding boxes.
- **Controlling the Rate of Classification Loss Reduction**
  If classification loss decreases too quickly, the model may struggle with sufficient exploration, making localization training difficult.
- **Adaptive Weight Strategy**
  Increasing the weight of classification loss in the early stages while gradually raising the weight of localization loss over time can lead to more effective training.

3.2 Mid-stage Failure Cases

3.2.1 Issues

- While localization performance (*mean IoU*) improves, classification loss stagnates.
- A high number of false positives occur, particularly for small objects, leading the model to predict small bounding boxes randomly.

3.2.2 Proposed Solutions

- **Enhancing Hard Negative Mining**
  Actively filtering out false positives to help the model learn more effectively.
- **Applying Data Augmentation Techniques**
  Utilizing scale transformations and random cropping to improve small object detection performance.
- **Adjusting Object Confidence Threshold**
  Properly tuning confidence scores for small bounding boxes to reduce unnecessary predictions.

3.2.3 Case Analysis

As loss optimization progresses, the model starts generating a large number of small false positive bounding boxes.

However, this does not indicate that the model has learned the precise locations of objects; rather, it adopts a **random prediction strategy by generating multiple small boxes**.

A notable failure pattern emerges where the model predicts bounding boxes even in **regions without faces**, indicating that it has not properly learned meaningful image features.

Interestingly, when these randomly predicted small boxes partially overlap with ground truth boxes, **the supervision signal helps the model adjust its learning direction**.

Qualitative analysis over several epochs reveals two key trends: **small predicted boxes gradually align with ground truth boxes**, and **the number of predicted boxes increases**.

During this learning process, an increasing number of bounding boxes surpass the **IoU threshold of 0.5**.

3.2.4 Failure Case Analysis

- **Dataset Complexity**
  - If the benchmark dataset contains a large number of very small bounding boxes, achieving IoU ≥ 0.5 becomes challenging.
  - Similar issues can also occur with large bounding boxes.
  - If early false positive predictions predominantly consist of small boxes, IoU values may remain below 0.5, even when the boxes are contained within larger ones.

3.2.5 Additional Refinements

- **Comparison Between Anchor-free and Anchor-based Models**
  Evaluating whether anchor-free methods are more effective for learning small objects.
- **Introducing an Adaptive IoU Threshold**
  Applying a lower IoU threshold for small objects to improve evaluation criteria.
- **Applying Loss Scaling Strategies**
  Emphasizing classification loss over localization loss during the early training phase.

3.2.6 Key Takeaways

- **During the mid-stage of training, small false positive boxes increase while classification loss stagnates.**
- **To mitigate random prediction tendencies, methods such as hard negative mining, data augmentation, and confidence threshold adjustment are necessary.**
- **For datasets where achieving IoU ≥ 0.5 is difficult, adjusting evaluation criteria and loss strategies is crucial.**

3.3 Late-stage Failure Cases

3.3.1 Issues

- The model becomes overfitted to the training data.
- Validation performance stagnates or deteriorates.

3.3.2 Proposed Solutions

- **Applying Early Stopping**
  Stopping training at the optimal point to prevent overfitting.
- **Incorporating Regularization Techniques**
  Utilizing methods such as Dropout and L2 regularization to improve generalization.
- **Expanding Data Augmentation Strategies**
  Introducing more diverse augmentations to enhance the model's ability to generalize.

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).