Vol. 8, No. 4, August 2026

ISI Journal of Information Systems and Informatics

Published By Asosiasi Doktor Sistem Informasi Indonesia

Performance Trade-Off Analysis of Faster R-CNN with Grid-Based Histogram for Student Face Detection

Arie Satia Dharma[1], Herimanto[2], Ranty Deviana Siahaan[3], Luna Sweeta Pangaribuan[4], Lamboy Albertson Sirait[5]

[1,2,3,4,5]Informatics, Faculty of Informatics and Electrical Engineering, Del Institute of Technology, Indonesia

Received:
February 20, 2026
Revised:
June 6, 2026
Accepted:
August 5, 2026
Published:
August 30, 2026

Corresponding Author:
Author Name*:
Arie Satia Dharma
Email*:
ariesatia@del.ac.id

DOI:
10.63158/journalisi.v8i4.1681

© 2026 Journal of Information Systems and Informatics. This open access article is distributed under a (CC-BY License)

Abstract. Face detection supports applications such as security, identity management, human–computer interaction, and academic information systems. Faster R-CNN is known for strong detection accuracy, but its Region Proposal Network produces many candidate boxes, including background proposals, which may increase processing cost. This study replaces the conventional anchor-generation process with a Grid-Based Histogram method to improve inference efficiency while retaining competitive detection performance. Experiments were conducted on 1132 annotated student profile images collected from the Campus Information System of Institut Teknologi Del. The standard and modified models were evaluated using mean Intersection over Union (IoU), mean Average Precision at IoU 0.50 (mAP@50), and average latency per image with an inference batch size of one. Standard Faster R-CNN achieved an IoU of 0.7595, an mAP@50 of 0.9818, and a latency of 0.170 s per image. The modified model obtained an IoU of 0.7519, an mAP@50 of 0.9719, and a latency of 0.147 s per image. Thus, latency decreased by about 13.53%, with small reductions in localization and detection accuracy. The novelty of this study lies in applying a Grid-Based Histogram as a lightweight replacement for conventional anchor generation in Faster R-CNN, resulting in a preliminary speed–accuracy trade-off rather than an overall performance improvement.

Keywords: Face Detection, Faster R-CNN, Region Proposal Network, Grid-Based Histogram, Inference Latency, Object Detection

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

1. INTRODUCTION

Face detection is a computer-vision task that identifies the presence and spatial location of human faces in digital images or video frames [1][2][3]. Unlike face recognition, which determines a person's identity, face detection only locates facial regions, typically by producing bounding boxes around detected faces. Face detection serves as an important initial stage in security systems, human–computer interaction, student attendance applications, and other systems that require facial regions as inputs for subsequent processing. Despite substantial progress, face detection remains challenging because facial appearance can vary because of pose, expression, illumination, image resolution, scale, occlusion, and background complexity [4][5]. Convolutional Neural Networks (CNNs) have been widely adopted because they can learn hierarchical visual representations directly from image data [6][7]. They handle changes in position, size, and orientation effectively, making them suitable for object detection applications. Among CNN-based object detectors, Faster R-CNN is a two-stage architecture that combines feature extraction, region proposal generation, object classification, and bounding-box regression within a unified framework.

Several studies have significantly contributed to the understanding and enhancement of the Faster R-CNN model. One study implemented Faster R-CNN integrated with IoT demonstrated excellent performance in smart office face recognition systems, achieving an accuracy of 99.3% with VGG-16 [8]. Another research focused on Fine-tuned Modified Faster R-CNN model achieved a superior accuracy of 98.26%, outperforming several existing object detection approaches, including Fast R-CNN, R-CNN, YOLOv5, SSD, and RetinaNet [9]. The experimental results demonstrate that Faster R-CNN achieved excellent performance in Javanese script detection, obtaining a mean Average Precision (mAP) of 0.8381, accuracy of 96.31%, precision of 96.53%, recall of 96.38%, and F1-Score of 96.41%. The high accuracy and evaluation scores indicate that Faster R-CNN is highly effective in detecting and localizing Javanese script characters, outperforming conventional text detection approaches for small and complex objects. [10]. Another research demonstrate that the improved Faster R-CNN achieved a superior mean Average Precision (mAP) of 99.5% with 29.8 FPS, outperforming state-of-the-art object detection methods such as SSD, YOLOv3, RetinaNet, Cascade R-CNN, FCOS, and CornerNet-Squeeze, thereby proving its effectiveness and reliability for real-time traffic

sign detection in autonomous driving systems [11]. These findings confirm the strength of Faster R-CNN in object localization. However, its two-stage structure requires additional processing for proposal generation and proposal-wise classification, which can limit inference efficiency.

Several alternative architectures have been developed specifically to balance face-detection accuracy and processing speed. MTCNN uses a three-stage cascaded architecture that progressively generates, refines, and classifies candidate face regions while jointly estimating facial landmarks [12]. RetinaFace employs single-stage dense prediction and multi-task learning to localize faces and facial landmarks across different scales [13]. YOLO-based face detectors adapt one-stage detection architectures for efficient face localization and can be configured in different model sizes for desktop, embedded, and mobile environments [14]. Similarly, SSD removes the separate proposal-generation stage and predicts object classes and bounding boxes directly from multi-scale feature maps [15]. Lightweight detectors such as BlazeFace further reduce computational requirements through compact feature extraction and an efficient anchor design intended for mobile devices [16].

These approaches provide important alternatives when inference speed is the primary consideration. Nevertheless, replacing Faster R-CNN entirely with a one-stage or lightweight detector does not explain whether the proposal-generation stage within a two-stage detector can itself be simplified. The present study therefore retains the Faster R-CNN architecture and focuses specifically on improving the efficiency of its proposal-generation mechanism. The objective is not to claim that the proposed model is faster than RetinaFace, MTCNN, YOLO-based detectors, SSD, or other lightweight detectors, because these models are not evaluated under the same experimental conditions. Instead, they provide relevant reference architectures for understanding the broader speed–accuracy trade-off in face detection. Previous work has attempted to improve conventional RPN performance through contextual feature enhancement, adaptive anchor configurations, improved proposal classification, and optimized loss functions [17]. Building on this research direction, the present study replaces the conventional anchor-generation mechanism with a Grid-Based Histogram module.

This study investigates whether integrating a Grid-Based Histogram into Faster R-CNN can reduce proposal-processing overhead and inference latency while maintaining competitive face-detection performance. We hypothesize that it will produce a measurable efficiency gain with only a limited reduction in localization and detection performance. The main contribution of this study is an empirical evaluation of Grid-Based Histogram proposal generation as a lightweight alternative to conventional anchor generation within Faster R-CNN, with particular attention to the resulting speed–accuracy trade-off in student face detection.

2. METHODS

This study employed a comparative experimental design to evaluate two face-detection models namely the standard Faster R-CNN and a modified Faster R-CNN incorporating a Grid-Based Histogram mechanism. Both models used the same dataset partition, image preprocessing, VGG-16 backbone, optimization settings, hardware environment, and evaluation protocol. The only architectural difference was located in the proposal-generation stage. The standard model used conventional dense anchor generation, whereas the modified model used histogram-guided anchor generation. The Grid-Based Histogram module replaced only the conventional anchor-generation mechanism and did not replace the entire Region Proposal Network. The experimental procedure consisted of data collection, image annotation, dataset partitioning, preprocessing, Faster R-CNN (standard Faster R-CNN and modified Faster R-CNN), and performance evaluation, as illustrated in Figure 1.

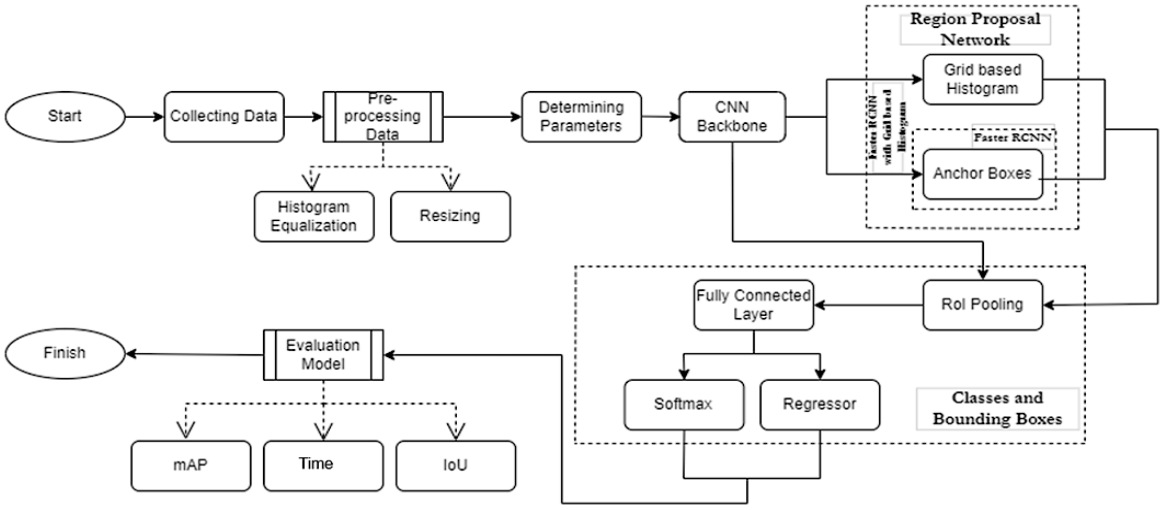

Figure 1. Flowchart of Experiment

2.1. Collecting Data

The dataset consisted of 1,132 student profile images collected from students of Institut Teknologi Del from the 2020 to 2024 cohorts. Google Drive was utilized as a platform for data collection and storage. An example of the dataset used in this study is shown in Figure 2.

Figure 2. Dataset Example

Each facial region was annotated using a rectangular bounding box that covered the main facial area using Roboflow. The annotations were reviewed to identify incomplete, excessively large, or incorrectly positioned bounding boxes before the images were included in the experiment. Each image contained one face, as illustrated in Figure 3.

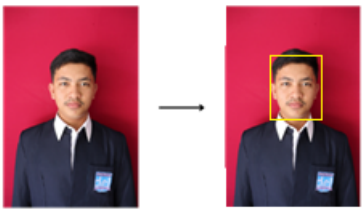

Figure 3. Face Annotation

The 1,132 images were divided into mutually exclusive training, validation, and test subsets. The training subset was used to update the model weights and biases. The validation subset was used to select the batch size, number of epochs, learning rate, Grid-Based Histogram configuration, and decision thresholds. The test subset was reserved exclusively for the final performance evaluation. The dataset partition is illustrated in Table 1.

Table 1. Dataset Partition

Dataset Subset	No. Images	Percentage
Training	906	80.04%
Validation	113	9.98%
Test	113	9.98%
Total	1132	100%

2.2. Pre-processing Data

The research proceeds to the data preprocessing stage, this stage is essential for enhancing the quality of the input data and eliminating noise or irrelevant information. The primary goal of data preprocessing is to help the model more easily identify important facial features, thereby improving the overall performance of the face detection process [18][19]. The resizing technique is a crucial step in the preprocessing stage of image processing, as it ensures that all input images have a uniform size before being fed into the model [8]. In this study, each image is resized to 300 × 300 pixels. This adjustment is intended to enable the model to process the data more efficiently and quickly. The resizing process, as part of the overall preprocessing workflow is illustrated in Figure 4.

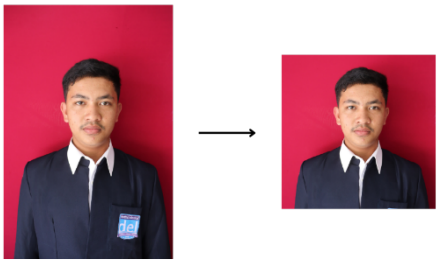

Figure 4. Resizing Technique

Histogram equalization was then applied to improve image contrast and reduce variations in illumination. A contrast-adjustment parameter of 0.4 was used based on preliminary validation experiments. This value was selected based on an analysis indicating that it produces balanced lighting neither too bright nor too dark, while preserving important facial features. The process of histogram equalization implemented in this study is illustrated in Figure 5.

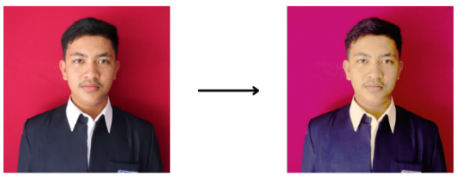

Figure 5. Histogram Equalization

The histogram equalization applied during preprocessing must be distinguished from the Grid-Based Histogram mechanism used during proposal generation. Preprocessing histogram equalization was calculated from the raw-image pixel intensities before the

image entered the CNN. In contrast, the Grid-Based Histogram proposal mechanism was calculated from normalized CNN feature-map activations after feature extraction.

2.3. Hyperparameter

Hyperparameters are configuration parameters that are set prior to the training process and remain constant throughout the training phase [7][20], Hyperparameter selection was performed using the validation subset rather than the test subset to prevent information from the final evaluation data from influencing model development. The key hyperparameters under consideration include learning rate, batch size, and epoch count. To determine the most effective combination of batch size and epoch count EarlyStopping was used with val_accuracy as the monitored metric and a patience value of 5, a series of preliminary experiments were conducted as shown in Table 2.

Table 2. Hyperparameter Experiments

Experiment	Optimizer	Learning Rate	Batch Size	Epoch
1	Adam	0.00001	16	10;50;100;500;1000
2	Adam	0.00001	32	10;50;100;500;1000
3	Adam	0.00001	64	10;50;100;500;1000

Experiments were conducted on a server equipped with an NVIDIA B200 GPU, 192 GB of HBM3e VRAM, and 2 TB of storage. For the software, we used Python 3.10, PyTorch 2.8.0, CUDA 12.8 and cuDNN 9.7.0. Training duration was not recorded in the current experiment; therefore, the computational comparison focuses on inference latency and FPS under the same hardware and software environment.

2.4. Training Faster R-CNN

The input image is processed through a series of convolutional layers, where the network learns to recognize facial patterns and extract relevant features by applying convolutional filters. Each successive convolutional layer captures increasingly abstract representations of the image, enabling the model to identify complex and high-level visual features [21]. This process results in the generation of a feature map that retains essential spatial information about the location and structure of the face. In this study, the convolutional neural network architecture employed is VGG-16, known for its depth

and strong capability in capturing fine-grained feature details [22]. The mathematical operations involved in generating the feature map are presented following this explanation. The overall process of the convolutional backbone is illustrated in Figure 6.

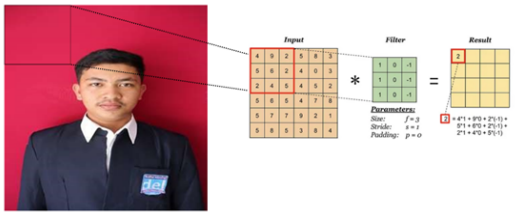

Figure 6. CNN Backbone Step

The input image is represented as a two-dimensional array and processed using a convolutional filter (kernel) with dimensions of 3 × 3, a stride of 1, and no padding (padding = 0) to generate a corresponding feature map. The filter slides across the input array, performing element-wise mathematical operations specifically, a dot product between the filter values and the corresponding input pixels. This operation produces a single value at each position, which becomes an element of the resulting feature map. The overall mechanism of this feature extraction process is illustrated in Figure 7.

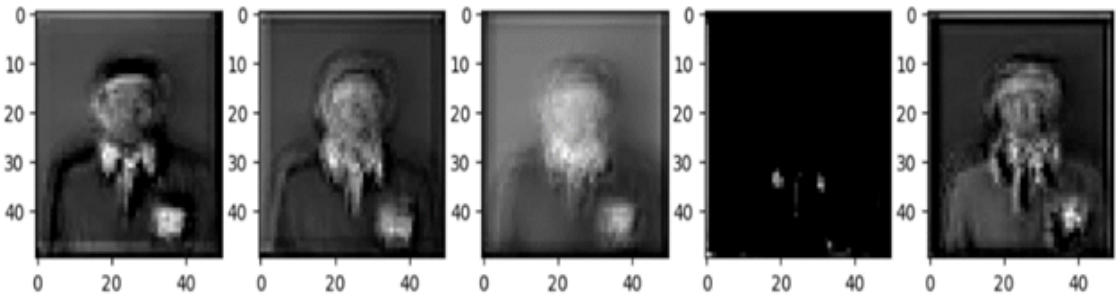

Figure 7. Feature Map

2.4.1. Region Proposal Network

RPN (Region Proposal Network) is a small neural network that operates on the final feature map of the convolution layer and predicts the presence of objects and estimates the object's bounding box. At this stage, two branches will differentiate between the two approaches, the steps are as follows:

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

1) RPN Using Standard Anchor Box Generation

In this model, the feature map generated from the Convolutional Neural Network backbone stage is further processed to generate anchor boxes, which are candidate regions that may contain faces. These anchor boxes are then evaluated based on how well they align with the actual face objects present in the image, using predefined criteria such as overlap thresholds [23][24]. The process of the anchor box generation is presented in Algorithm 1.

Algorithm 1. Standard Anchor Box Proposal Generation in RPN
Input: Feature map, anchor scales 32, 64, and 128, aspect ratios 1:1, 1:2, and 2:1, feature-map stride, and non-maximum suppression threshold.
Output: Final proposal set.

1. Receive the feature map from the VGG-16 backbone.
2. Determine the height and width of the feature map.
3. Create an empty list to store all generated anchor boxes.
4. For each spatial location on the feature map:
    a. Map the feature-map location to the corresponding center position in the original 300 x 300 image.
    b. Generate anchor boxes using scales 32, 64, and 128.
    c. Combine each anchor scale with aspect ratios 1:1, 1:2, and 2:1.
    d. Add all generated anchor boxes to the anchor list.
5. Clip anchor boxes that exceed the image boundaries.
6. Remove invalid anchor boxes after clipping.
7. Pass valid anchor boxes to the RPN classification and regression branches.
8. Refine the anchor boxes using the predicted bounding-box offsets.
9. Sort all proposals based on objectness scores.
10. Retain the top 2,000 proposals before non-maximum suppression.
11. Apply non-maximum suppression to remove highly overlapping proposals.
12. Retain the top 300 proposals after non-maximum suppression.
13. Return the final proposal set for RoI Pooling and subsequent classification.

In the standard Faster R-CNN model, anchor boxes were generated at each spatial location of the VGG-16 feature map. Since the input images were resized to 300 x 300 pixels, the anchor scales were set to 32, 64, and 128 pixels to represent small, medium, and relatively large facial regions. Each scale was combined with three aspect ratios, namely 1:1, 1:2, and 2:1, resulting in nine anchor boxes at each feature-map location. The most promising anchor boxes are selected and passed forward as object proposals for further classification and localization.

2) RPN Using Grid-Based Histogram Generation

In the modified model, the conventional dense anchor-generation mechanism was replaced with a Grid-Based Histogram anchor generator. The remaining RPN components, including objectness classification and bounding-box regression, were retained. The Grid-Based Histogram technique is applied after the Convolutional Neural Network backbone stage. The configuration of the Grid-Based Histogram proposal-generation mechanism used in the main experiment is presented in Table 3.

Table 3. Grid-Based Histogram Proposal Configuration

Parameter	Value
Input image size	300 x 300 pixels
Histogram source	CNN feature-map activations
Grid size	20 x 20
Histogram bins	32
Selection threshold	0.5
Anchor scales	32, 64, 128
Aspect ratios	1:1, 1:2, 2:1
Candidate boxes per selected grid cell	9
Proposals before NMS	1,200
Proposals after NMS	180
NMS threshold	0.5

The detailed steps of the Grid-Based Histogram proposal-generation process are presented in Algorithm 2.

Algorithm 2. Grid-Based Histogram Proposal Generation
Input: Feature map, grid size 20 x 20, 32 histogram bins, proposal-selection threshold 0.5, anchor scales 32, 64, and 128, aspect ratios 1:1, 1:2, and 2:1, and non-maximum suppression threshold 0.5.
Output: Final proposal set.
1. Receive the feature map from the VGG-16 backbone.
2. Apply ReLU activation to remove negative activation values.
3. Average the activation values across all feature-map channels to produce a single activation map.
4. Normalize the activation map into a value range between 0 and 1.
5. Divide the normalized activation map into 20 x 20 grid cells.
6. Create an empty list to store candidate proposals.

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

```
7. For each grid cell:
   a. Extract all normalized activation values inside the grid
      cell.
   b. Calculate a normalized histogram with 32 bins.
   c. Calculate the histogram score from the normalized histogram
      values.
   d. If the histogram score is greater than or equal to 0.5, use
      the grid cell as a proposal  center.
   e. Generate candidate boxes using anchor scales 32, 64, and 128
      with aspect ratios 1:1, 1:2, and 2:1.
   f. Convert the candidate-box coordinates from feature-map
      coordinates into original image coordinates.
   g. Clip the candidate-box coordinates so that they remain inside
      the image boundary.
   h. Use the histogram score as the initial proposal score.
   i. Add the generated candidate boxes to the candidate proposal
      list.
   j. If the histogram score is lower than 0.5, skip the grid cell.
8. Pass all candidate proposals to the retained RPN objectness and
   bounding-box regression branches.
9. Refine each proposal using the predicted bounding-box offsets.
10.Sort all proposals based on their objectness scores.
11.Retain a maximum of 1,200 proposals before non-maximum
   suppression.
12.Apply non-maximum suppression to remove highly overlapping
   proposals.
13.Retain the top 180 proposals after non-maximum suppression.
14.      Return the final proposal set.
```

The histogram score is not treated as the final detection confidence. Instead, it is used as an initial filtering score to select informative grid cells. Grid cells with scores greater than or equal to 0.5 are used as proposal centers, while grid cells below this threshold are discarded. For each selected grid cell, candidate boxes are generated using anchor scales of 32, 64, and 128 pixels with aspect ratios of 1:1, 1:2, and 2:1. The histogram score is assigned as the initial proposal score, and the generated boxes are then refined by the retained RPN objectness and bounding-box regression branches before non-maximum suppression is applied.

2.4.2. RoI Processing and Final Detection

The proposals generated by either the conventional anchor generator or the Grid-Based Histogram module were mapped to the shared CNN feature map. RoI Pooling converted each proposal into a fixed-size feature representation. The resulting features were processed by the classification and bounding-box regression heads to classify each proposal as face or background and refine its coordinates.

These proposal regions are selected from the feature map generated by the Convolutional Neural Network backbone through the Region of Interest (ROI) Pooling layer, which standardizes the size of the feature regions. The resulting bounding boxes are then refined through a non-maximum suppression process, which selects the most accurate bounding box by eliminating overlapping boxes with lower confidence scores [25]. This stage ensures that the final detection output contains only the most relevant bounding box for each detected object.

2.5. Evaluation Metrics

The predicted bounding boxes were evaluated against the ground-truth annotations using Intersection over Union (IoU). For a predicted bounding box and a ground-truth bounding box, The IoU value was calculated using Equation (1).

$$\text{IoU} = \text{Area of Intersection} / \text{Area of Union} \quad (1)$$

Precision and recall were used to measure the correctness and completeness of the detection results. These metrics were calculated using Equations (2) and (3).

$$\text{Precision} = TP / (TP + FP) \quad (2)$$

$$\text{Recall} = TP / (TP + FN) \quad (3)$$

where TP, FP, and FN represent true-positive, false-positive, and false-negative detections, respectively.

The F1-score was calculated as the harmonic mean of precision and recall:

$$\text{F1score} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (4)$$

Mean Average Precision at IoU 0.50 was used to measure detection performance at an IoU threshold of 0.50. Since this study only detected one object class, namely face, mAP@50 is equivalent to AP@50 for the face class was calculated using Equation (5).

$$\text{mAP@50} = (1 / C) \times \Sigma \text{ AP@50 for all classes} \quad (5)$$

where C is the number of object classes. In this study, C = 1 because only the face class was evaluated. Average inference latency was calculated by dividing the total inference time by the number of processed test images, as shown in Equation (6).

$$\text{Average Latency} = \text{Total Inference Time} / \text{Number of Test Images} \quad (6)$$

The latency was reported in seconds per image using an inference batch size of one. FPS was calculated as the inverse of average latency, as shown in Equation (7).

$$\text{FPS} = 1 / \text{Average Latency} \quad (7)$$

In this study, latency measurement included the model inference stage and post-processing, including bounding-box regression and non-maximum suppression, but excluded image-loading and preprocessing time. The same timing scope was applied to both the standard Faster R-CNN and the Faster R-CNN with Grid-Based Histogram.

3. RESULTS AND DISCUSSION

3.1. Results

This section presents the evaluation results of the standard Faster R-CNN with Anchor Box proposal generation and the modified Faster R-CNN with Grid-Based Histogram proposal generation. Both models were trained and evaluated using the same dataset partition, image preprocessing, VGG-16 backbone, optimization settings, hardware environment, and evaluation protocol. The analysis covered overall detection performance, detailed detection metrics, proposal-generation performance, grid-size ablation, and qualitative detection results.

3.1.1. Performance Comparison

Table 4 presents the overall performance of the two models in terms of mean Intersection over Union, mean Average Precision at an IoU threshold of 0.50, mean inference latency, and processing throughput.

Table 4. Evaluation Results of Both Models

Experiment	IoU	mAP@50	Latency (s/image)	FPS
Standard Faster R-CNN	0.7595	0.9818	0.170	5.88
Faster R-CNN with Grid-Based Histogram	0.7519	0.9719	0.147	6.80

The standard Faster R-CNN achieved a mean IoU of 0.7595 and an mAP@50 of 0.9818, whereas the modified model obtained a mean IoU of 0.7519 and an mAP@50 of 0.9719. Thus, the modified model showed absolute reductions of 0.0076 in mean IoU and 0.0099 in mAP@50. In terms of computational performance, the mean latency decreased from 0.170 s to 0.147 s per image so the percentage reduction was calculated as 13.53%. The corresponding throughput increased from 5.88 FPS to 6.80 FPS, representing an increase of approximately 15.65%. These results demonstrate an observed speed–accuracy trade-off: the modified model processed images more efficiently but produced slightly lower localization and detection performance.

3.1.2. Detailed Detection Performance

Table 5 presents the precision, recall, F1-score, and mAP@[0.50:0.95] results. The standard Faster R-CNN achieved higher values across all detection metrics. Compared with the standard model, the modified model showed absolute decreases of 0.0113 in precision, 0.0065 in recall, 0.0088 in F1-score. Its mAP@[0.50:0.95] decreased by 0.0260, which was larger than the reduction observed at IoU 0.50. The larger difference in mAP@[0.50:0.95] suggests that the modified model experienced greater difficulty when stricter localization thresholds were applied. Therefore, its main limitation appears to be precise bounding-box localization rather than a substantial loss in the ability to identify facial regions.

Table 5. Detailed Detection-Performance Metrics

Experiment	Precision	Recall	F1-Score	mAP@50	mAP@[0.50:0.95]
Standard Faster R-CNN	0.9624	0.9415	0.9518	0.9818	0.7245
Faster R-CNN with *Grid-Based Histogram*	0.9511	0.9350	0.9430	0.9719	0.6985

The results reported in this study, including mAP@50, were obtained from a single complete training run and a single evaluation setting for each model. Repeated inference runs and repeated training with different random seeds were not conducted in the current experiment. Therefore, repeated-run statistics, repeated-seed statistics, and confidence intervals are not reported. The results should be interpreted as descriptive experimental findings that show the observed speed–accuracy trade-off between the standard Faster R-CNN and the modified Faster R-CNN with Grid-Based Histogram, rather than as statistically conclusive evidence.

3.1.3. Proposal-Generation Performance

Because the architectural modification was introduced at the proposal-generation stage, the number and quality of proposals were evaluated separately as presented in Table 6.

Table 6. Proposal-Generation Comparison

Experiment	Proposals before NMS	Proposals after NMS	Proposal recall@0.50	Proposal recall@0.70	Proposal recall@0.90
Standard Faster R-CNN	2,000	300	0.9912	0.9245	0.6512
Faster R-CNN with *Grid-Based Histogram*	1,200	180	0.9845	0.9120	0.6385

The Grid-Based Histogram mechanism reduced the number of candidate proposals from 2,000 to 1,200 before non-maximum suppression and from 300 to 180 after non-maximum suppression. Both reductions correspond to a 40% decrease in proposal count. This reduction explains why the modified model achieved lower inference latency, because fewer candidate regions had to be processed by the retained RPN objectness branch, bounding-box regression branch, RoI Pooling layer, and final classification stage. Although the number of proposals decreased, the modified model maintained proposal recall values of 0.9845, 0.9120, and 0.6385 at IoU thresholds of 0.50, 0.70, and 0.90, respectively. However, the differences became more visible at stricter IoU thresholds, indicating that the modified proposal generator produced slightly less precise candidate regions than the standard anchor-generation mechanism.

3.1.4. Grid-Size Ablation Study

Table 7 presents the performance of three Grid-Based Histogram configurations. All configurations used 32 histogram bins, an activation threshold of 0.5, and the same preprocessing and normalization procedure.

Table 7. Ablation Study of Grid Size

Grid size	Histogram bins	Threshold	Mean IoU	mAP@50	Latency (s/image)
10 x 10	32	0.5	0.7321	0.9455	0.121
20 x 20	32	0.5	0.7519	0.9719	0.147
30 x 30	32	0.5	0.7554	0.9780	0.165

The 10 x 10 grid configuration achieved the lowest latency of 0.121 s per image, but it also produced the lowest mean IoU and mAP@50. Increasing the grid size to 20 x 20 improved mean IoU to 0.7519 and mAP@50 to 0.9719, while maintaining lower latency than the standard Faster R-CNN. The 30 x 30 grid configuration achieved the highest performance among the modified configurations, with a mean IoU of 0.7554 and mAP@50 of 0.9780, but its latency increased to 0.165 s per image, approaching the latency of the standard model. Therefore, the 20 x 20 grid was selected as the most balanced configuration under the evaluated conditions.

3.1.5. Qualitative Detection and Failure Case Analysis

Qualitative analysis was used to support the quantitative results and observe how both models behaved on the test images. The examples were grouped into successful detection, inaccurate localization, and missed-detection or low-IoU cases. A detection was considered successful when the predicted box overlapped the ground-truth face with an IoU of at least 0.50. Inaccurate localization occurred when the face was detected, but the bounding box was not well aligned, such as when it included hair, ears, clothing, or background. A missed-detection or low-IoU case occurred when the model failed to produce a prediction that sufficiently matched the ground truth. Since the test set mainly consisted of controlled student profile images, challenging cases such as blur, occlusion, low illumination, non-frontal pose, or unusual cropping were not sufficiently available. Therefore, the qualitative evaluation focused only on the observable cases in the test set.

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

As shown in Figure 10, both models were generally able to detect frontal student faces under controlled profile-photo conditions. However, the Faster R-CNN with Grid-Based Histogram occasionally produced less precise bounding boxes than the standard Faster R-CNN. In some examples, the modified model included irrelevant facial surroundings or produced boxes that did not align as tightly with the annotated face region.

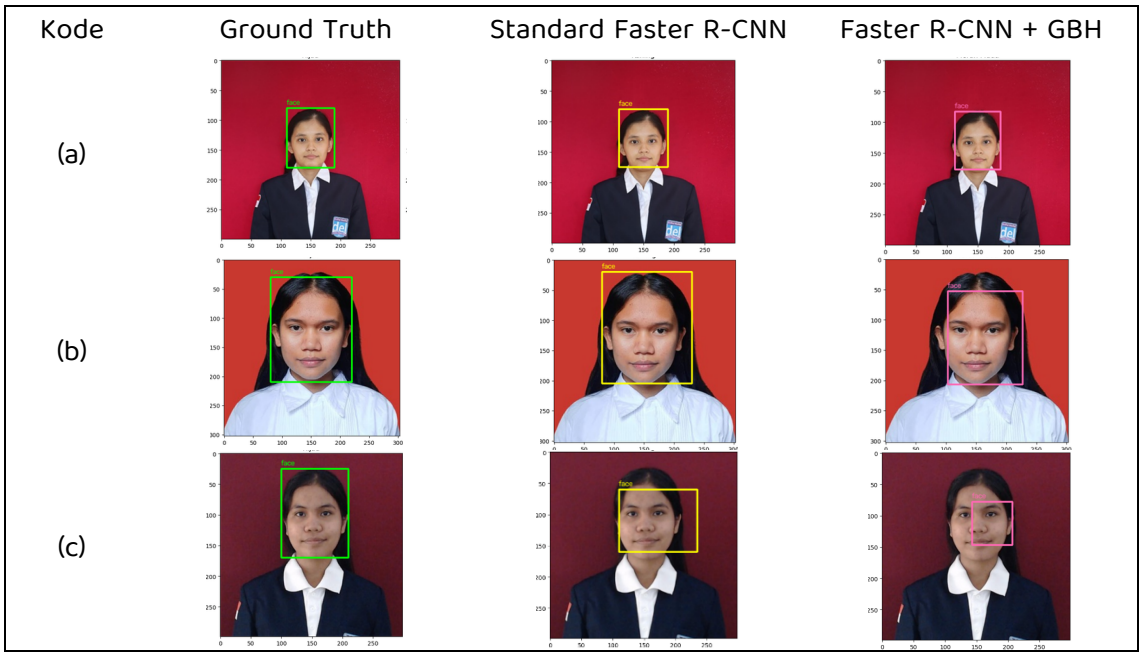

Figure 8. Qualitative detection examples: (a) successful detection, (b) inaccurate localization case, (c) low-confidence

3.1.6. Baseline Scope

Lightweight detector baselines such as YOLO, SSD, MobileNet-SSD, RetinaFace, and MTCNN were not included in the current experimental scope. Therefore, this study does not claim superiority over these architectures. Future work should include lightweight detector baselines to provide a broader comparison between two-stage and one-stage face-detection models in terms of accuracy, latency, and deployment feasibility.

3.2. Discussion

The experimental results indicate that the proposed Grid-Based Histogram mechanism introduces a speed–accuracy trade-off in Faster R-CNN-based face detection. The standard Faster R-CNN achieved a higher mean IoU of 0.7595 and mAP@50 of 0.9818, whereas the modified Faster R-CNN with Grid-Based Histogram obtained a mean IoU of

0.7519 and mAP@50 of 0.9719. The absolute decrease of 0.0099 in mAP@50 is relatively small, suggesting that the modified model still maintains competitive face-detection performance under the evaluated dataset conditions. However, the larger decrease in mAP@[0.50:0.95] indicates that the modified model is less precise when stricter localization thresholds are applied.

The main advantage of the modified model is its lower computational cost during inference. The average inference latency decreased from 0.170 s per image to 0.147 s per image, corresponding to a 13.53% reduction. This improvement is mainly caused by the reduction in the number of candidate proposals. The standard Faster R-CNN generated 2,000 proposals before non-maximum suppression and retained 300 proposals after non-maximum suppression, whereas the Grid-Based Histogram model generated 1,200 proposals before non-maximum suppression and retained 180 proposals after non-maximum suppression. This 40% reduction in proposal count means that fewer candidate regions had to be processed by the RPN refinement, RoI Pooling, and final classification stages.

For downstream applications such as attendance systems or face-recognition pipelines, the small reduction in mAP@50 may or may not be critical depending on the operational requirements. In controlled attendance scenarios where users face the camera directly and images are captured under stable lighting conditions, the modified model may still be acceptable because its mAP@50 remains high at 0.9719. However, even small decreases in precision, recall, and localization accuracy can affect downstream recognition performance if the detected bounding boxes crop the face too loosely or exclude important facial regions such as the chin, cheeks, or forehead. Therefore, the proposed model should be evaluated further in an end-to-end attendance or face-recognition pipeline before deployment.

In modern object detection research, contextual feature representation and multi-scale learning are important factors for improving localization accuracy, particularly for small, scale-varying, or partially occluded face instances [26]. Therefore, the decrease in localization precision in the modified model may be related to the simplified histogram-based proposal-generation process, which may not preserve spatial and contextual information as effectively as the standard RPN anchor-generation mechanism.

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

The improved efficiency should not be interpreted as evidence of real-time video performance. Although the modified model increased throughput from 5.88 FPS to 6.80 FPS, this value is still below commonly used real-time video rates. Therefore, the contribution of the proposed method should be interpreted as improved inference efficiency under the evaluated dataset conditions, rather than as real-time face-detection performance. Recent research on lightweight object detection systems also indicates that simplifying detection architectures can improve processing efficiency while maintaining adequate detection performance [11][27][28].

Another contribution of this study is demonstrating that modifications to the Region Proposal Network stage remain a promising area of research in the development of deep learning-based object detection. Various recent studies have sought to improve Faster R-CNN by integrating feature enhancement, lightweight backbones, and proposal-generation optimization to achieve a better balance between speed and accuracy. One effective approach is combining multi-scale feature learning with deformable convolutions, which can improve the model's ability to detect small objects and handle complex spatial variations [29][30]. Compared with this approach, the Grid-Based Histogram method proposed in this study offers a simpler and computationally lighter alternative, although it still requires further refinement to minimize the loss in detection precision.

Nevertheless, this study has several limitations. First, the dataset limits the generalizability of the findings. The images consist of student profile photos collected from a controlled academic environment, where faces are generally centered, frontal, and captured under relatively consistent conditions. Therefore, the dataset may not represent unconstrained surveillance or real attendance environments that include varying head poses, occlusions, motion blur, low illumination, crowded backgrounds, and multiple faces per image. Second, repeated training with different random seeds was not conducted. Therefore, the variability of mAP@50, IoU, precision, recall, and F1-score across different training runs could not be estimated. Future work should improve the proposal-generation mechanism by incorporating contextual features, adaptive grid selection, feature fusion, or lightweight attention modules while preserving the computational advantages of the current approach. Future studies should also include repeated-seed experiments and repeated inference-time measurements to provide

stronger statistical evidence regarding the stability and robustness of the proposed method.

4. CONCLUSION

This study aimed to evaluate whether replacing the conventional anchor-generation mechanism in Faster R-CNN with a Grid-Based Histogram approach could improve inference efficiency while maintaining competitive face-detection performance. The results show that the proposed modification achieved this objective in terms of efficiency, as the average latency decreased from 0.170 s per image to 0.147 s per image, corresponding to a 13.53% reduction. However, this efficiency gain was accompanied by a small decrease in detection performance, with mAP@50 decreasing from 0.9818 in the standard Faster R-CNN to 0.9719 in the modified model. Therefore, the proposed method should be interpreted as an efficiency-oriented speed–accuracy trade-off rather than an overall improvement over the standard Faster R-CNN. The scope of the findings is still limited because the dataset was collected from a single controlled source, consisting of student profile images from Institut Teknologi Del. Therefore, the results may not fully reflect more challenging face-detection conditions, such as unconstrained surveillance settings, real attendance scenarios, pose variation, occlusion, motion blur, low-light images, or images containing multiple faces. Future work should validate the proposed approach using external face-detection datasets, examine its performance in more realistic deployment environments, and strengthen the proposal-refinement stage through adaptive grid selection, contextual feature integration, feature fusion, or lightweight attention mechanisms.

REFERENCES

[1] D. Hendrawan, Wahyuni, and P. Adytia, "Comparative Performance Analysis of YOLOv12 and RF-DETR in Face Detection," *J. Inf. Syst. Informatics*, vol. 8, no. 2, 2026, doi: 10.63158/journalisi.v8i2.1561.

[2] H. F. Al-Selwi, N. Hassan, H. B. A. Ghani, N. A. B. A. Hamzah, and A. B. A. Aziz, "Face mask detection and counting using you only look once algorithm with Jetson Nano and NVDIA giga texel shader extreme," *IAES Int. J. Artif. Intell.*, vol. 12, no. 3, 2023, doi: 10.11591/ijai.v12.i3.pp1169-1177.

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

[3] D. Mamieva, A. B. Abdusalomov, M. Mukhiddinov, and T. K. Whangbo, "Improved Face Detection Method via Learning Small Faces on Hard Images Based on a Deep Learning Approach," *Sensors*, vol. 23, no. 1, 2023, doi: 10.3390/s23010502.

[4] R. Qi, R. S. Jia, Q. C. Mao, H. M. Sun, and L. Q. Zuo, "Face Detection Method Based on Cascaded Convolutional Networks," *IEEE Access*, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2934563.

[5] C. Dhanamjayulu *et al.*, "Identification of malnutrition and prediction of BMI from facial images using real-time image processing and machine learning," *IET Image Process.*, vol. 16, no. 3, 2022, doi: 10.1049/ipr2.12222.

[6] D. Bhatt *et al.*, "CNN variants for computer vision: History, architecture, application, challenges and future scope," *Electron.*, vol. 10, no. 20, 2021, doi: 10.3390/electronics10202470.

[7] T. A. Prasetyo, T. L. Gaol, N. F. Sipahutar, T. Siahaan, T. E. Manik, and R. Chandra, "Refining Tomato Disease Recognition: Hyperparameter Tuning on ResNet-101 Architecture For Precise Leaf-Based Classification," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 34, no. 2, 2024, doi: 10.11591/ijeecs.v34.i2.pp1204-1213.

[8] G. Rajeshkumar *et al.*, "Smart Office Automation Via Faster R-CNN Based Face Recognition and Internet of Things," *Meas. Sensors*, vol. 27, 2023, doi: 10.1016/j.measen.2023.100719.

[9] J. Madake, S. Korpade, H. Kulkarni, R. Kumbhar, and S. Bhatlawande, "Safe and Efficient Traffic Obstacle Detection Using Fine-Tuned Modified Faster R-CNN," in *Lecture Notes in Networks and Systems*, 2024. doi: 10.1007/978-981-97-7571-2_36.

[10] M. Faishal, M. D. Sulistiyo, and A. F. Ihsan, "Javanese Script Letter Detection Using Faster R-CNN," *Indones. J. Artif. Intell. Data Min.*, vol. 6, no. 2, p. 243, 2023, doi: 10.24014/ijaidm.v6i2.24641.

[11] X. Li, Z. Xie, X. Deng, Y. Wu, and Y. Pi, "Traffic sign detection based on improved faster R-CNN for autonomous driving," *J. Supercomput.*, vol. 78, no. 6, 2022, doi: 10.1007/s11227-021-04230-4.

[12] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks," *IEEE Signal Process. Lett.*, vol. 23, no. 10, 2016, doi: 10.1109/LSP.2016.2603342.

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

[13] N. A. Mahamad Amin (Corresponding Author), N. N. Amir Sjarif, and S. S. Yuhaniz, "A Comparative Study of Facial Feature Extraction using MTCNN, RetinaFace and Dlib Face Detector for Personality Traits Recognition," *Malaysian J. Comput. Sci.*, vol. 38, no. 2, 2025, doi: 10.22452/mjcs.vol38no2.2.

[14] D. Qi, W. Tan, Q. Yao, and J. Liu, "YOLO5Face: Why Reinventing a Face Detector," in *Lecture Notes in Computer Science*, 2023. doi: 10.1007/978-3-031-25072-9_15.

[15] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016. doi: 10.1007/978-3-319-46448-0_2.

[16] D. Xie, Y. Zhang, Z. Hu, T. Wang, F. Chen, and P. Hu, "Detect-TPE: A New Framework for Ideal Thumbnail-Preserving Encryption via Face Detection," *IEEE Internet Things J.*, vol. 11, no. 21, 2024, doi: 10.1109/JIOT.2024.3435136.

[17] Y. P. Chen, Y. Li, and G. Wang, "An Enhanced Region Proposal Network for object detection using deep learning method," *PLoS One*, vol. 13, no. 9, pp. e0203897-, Sep. 2018, doi: 10.1371/journal.pone.0203897.

[18] R. D. Siahaan, H. Pardede, I. N. Simbolon, I. Simbolon, and D. J. Gultom, "Enhancing Real-time Herbal Plant Detection in Agricultural Environments with YOLOv8," *J. Inf. Syst. Informatics*, vol. 6, no. 4, 2024, doi: 10.51519/journalisi.v6i4.889.

[19] A. S. Dharma, H. Herimanto, N. F. Simbolon, and A. R. Nababan, "Performance Trade-off of Anchor-Based and Anchor-Free Approaches of Faster R-CNN for Face Detection," *Sink. J. dan Penelit. Tek. Inform.*, vol. 10, no. 1, pp. 771–780, 2026, doi: 10.33395/sinkron.v10i1.15804.

[20] L. Franceschi *et al.*, "Hyperparameter Optimization in Machine Learning," *Found. Trends Mach. Learn.*, vol. 18, no. 6, 2025, doi: 10.1561/2200000088.

[21] A. Biswas and M. S. Islam, "A Hybrid Deep CNN-SVM Approach for Brain Tumor Classification," *J. Inf. Syst. Eng. Bus. Intell.*, vol. 9, no. 1, 2023, doi: 10.20473/jisebi.9.1.1-15.

[22] B. D. Mardiana, W. B. Utomo, U. N. Oktaviana, G. W. Wicaksono, and A. E. Minarno, "Herbal Leaves Classification Based on Leaf Image Using CNN Architecture Model VGG16," *J. RESTI*, vol. 7, no. 1, 2023, doi: 10.29207/resti.v7i1.4550.

[23] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: A Simple and Strong Anchor-Free Object Detector," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 4, 2022, doi: 10.1109/TPAMI.2020.3032166.

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

[24] H. Lyu, F. Qiu, L. An, D. Stow, R. Lewison, and E. Bohnett, "Deer survey from drone thermal imagery using enhanced faster R-CNN based on ResNets and FPN," *Ecol. Inform.*, vol. 79, 2024, doi: 10.1016/j.ecoinf.2023.102383.

[25] G. Cui and L. Zhang, "Improved faster region convolutional neural network algorithm for UAV target detection in complex environment," *Results Eng.*, vol. 23, 2024, doi: 10.1016/j.rineng.2024.102487.

[26] Y. Wang, "Pedestrian and Vehicle Detection Performance Analysis of Faster R-CNN under Extreme Weather Conditions," *Appl. Comput. Eng.*, vol. 80, no. 1, 2024, doi: 10.54254/2755-2721/80/2024ch0066.

[27] S. K. Abbas *et al.*, "Vision based intelligent traffic light management system using Faster R-CNN," *CAAI Trans. Intell. Technol.*, vol. 9, no. 4, 2024, doi: 10.1049/cit2.12309.

[28] D. A. Koutsomitropoulos and I. C. Gogou, "Object Detection Models and Optimizations: A Bird's-Eye View on Real-Time Medical Mask Detection," *Digital*, vol. 3, no. 3, 2023, doi: 10.3390/digital3030012.

[29] W. Zhao, F. Chen, H. Huang, D. Li, and W. Cheng, "A new steel defect detection algorithm based on deep learning," *Comput. Intell. Neurosci.*, vol. 2021, 2021, doi: 10.1155/2021/5592878.

[30] Y. Chen *et al.*, "Lightweight Detection Methods for Insulator Self-Explosion Defects," *Sensors*, vol. 24, no. 1, 2024, doi: 10.3390/s24010290.