

Article

FeatherFace: Robust and Lightweight Face Detection via Optimal Feature Integration

Dohun Kim, Jinmyung Jung  and Jinhyun Kim 

Division of Data Science, College of Intelligent Software Convergence, The University of Suwon,
Hwaseong 18323, Republic of Korea; dohun0714@naver.com

* Correspondence: jmjung@suwon.ac.kr (J.J.); jinhyun.kim@suwon.ac.kr (J.K.)

Abstract: Face detection in resource-constrained environments presents challenges, due to the computational demands of state-of-the-art models and the complexity of real-world conditions, such as variations in scale, pose, and occlusion. This study introduces FeatherFace, a lightweight face-detection architecture with only 0.49 M parameters, designed for high accuracy and efficiency in such environments. Leveraging MobileNet-0.25 as a backbone, FeatherFace incorporates advanced feature-integration strategies, including a bidirectional feature pyramid network (BiFPN), a convolutional block attention module (CBAM), deformable convolutions, and channel shuffling. Evaluated on the WIDERFace dataset, FeatherFace achieves an overall average precision (AP) of 87.2%, with notable performance gains of 4.0% AP on the Hard subset compared with the baseline. Ablation studies highlight the critical role of multiscale feature integration and the strategic placement of attention mechanisms in addressing detection challenges such as small or occluded faces. With its compact design and reduced inference time, FeatherFace bridges the gap between the reliability of computationally intensive models and the need for deploying robust models in highly resource-constrained environments, such as edge devices and embedded systems. This work provides valuable insights for developing robust and lightweight models suited to challenging real-world applications.

Keywords: face detection; feature integration; lightweight deep learning model



Academic Editor: Silvia Liberata Ullo

Received: 10 January 2025

Revised: 24 January 2025

Accepted: 26 January 2025

Published: 27 January 2025

Citation: Kim, D.; Jung, J.; Kim, J. FeatherFace: Robust and Lightweight Face Detection via Optimal Feature Integration. *Electronics* **2025**, *14*, 517. <https://doi.org/10.3390/electronics14030517>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Face detection serves as the cornerstone in numerous computer vision applications, including surveillance, authentication systems, expression analysis, and human–computer interactions [1–8]. Recent advances in deep learning have significantly improved face detection accuracy; however, reliably detecting faces in unconstrained environments remains a formidable challenge [9–15]. This complexity stems from extreme variations in factors such as scale, pose, occlusion, expression, and illumination associated with face detection. For example, faces in images are, on average, approximately five times smaller than general objects in standard training datasets, making the task inherently more challenging than generic object detection [11,16]. The WIDERFace dataset exemplifies these challenges, serving as a comprehensive benchmark for assessing face detection performance under such diverse and demanding conditions [17].

RetinaFace, one of the most well-known deep learning-based face detection methods, has demonstrated state-of-the-art performance on this challenging dataset [9]. However, such high-performance models typically rely on computationally intensive backbones,

such as ResNet50 or ResNet152, which demand substantial processing power and memory [18,19]. These requirements render them impractical for deployment in highly resource-constrained environments, such as edge devices and Internet of Things (IoT) platforms, where power, memory, and computational capacity are severely limited. Consequently, there is a growing demand for robust and efficient face detection models designed to operate accurately within these constraints, driven by the increasing adoption of edge-based applications, such as smart home systems, wearable devices, and the IoT [20–23].

In response, recent research has focused on lightweight neural architectures capable of maintaining high detection accuracy with reduced computational demands. Notable approaches include employing compact backbone networks in conjunction with advanced multiscale feature aggregation modules, such as feature pyramid network (FPN), weighted-fusion feature pyramid network (WFPN), and bidirectional feature pyramid network (BiFPN), which have yielded substantial performance gains [24–26]. Attention mechanisms, including the convolutional block attention module (CBAM), squeeze-and-excitation network (SENet), and separated and enhancement attention module (SEAM) have also been incorporated, to help models concentrate on salient facial features across different scales [27–29]. Although many of these lightweight models reach parameter sizes ranging from approximately 1 M to 7 M [29–35], they may still be excessively large for highly resource-constrained systems. Meanwhile, more recent approaches have demonstrated models with parameter counts of approximately 0.5 M or less, yet they often suffer from reduced performance, achieving only 82% to 84% overall AP on the WIDERFace dataset when evaluated with VGA-resolution input images [36–38].

Here, we introduce FeatherFace, an ultra-lightweight face detection architecture containing only 0.49 M parameters—drastically smaller than RetinaFace’s 25 M-parameter architecture. Despite its compact design and stringent computational constraints, FeatherFace achieves an improved overall AP of 87.2% in the WIDERFace dataset through optimal feature-integration strategies.

Building upon the RetinaFace baseline, we replace the conventionally heavy ResNet backbone with MobileNet-0.25, achieving a 50-fold reduction in the parameter count. To maintain detection accuracy under the tighter parameter budget, we incorporate enhanced feature-integration strategies that effectively combine multiscale feature information and emphasize critical facial cues. This strategy yields a 4.0% improvement in AP on the WIDERFace Hard subset, underscoring the effectiveness of our lightweight yet robust design. Specifically, systematic evaluations of various multiscale feature aggregation methods and attention mechanisms reveal that incorporating the BiFPN and CBAM together results in a 2.3% AP gain on the WIDERFace Hard subset. Furthermore, the adoption of the channel shuffle module and deformable convolution in the detection heads provides an additional 1.7% increase in AP.

By achieving a high detection accuracy with minimal computational overhead, our work facilitates the broader deployment of face detection systems, particularly in highly resource-constrained environments, such as edge devices and IoT platforms. We attribute these performance gains to the more effective integration of multiscale features and adaptive receptive fields, which are crucial for accurately detecting small faces, handling occlusions, and dealing with unfavorable illumination. Our ablation studies further validate this conclusion. We believe that the optimized feature-integration methodologies presented here can extend beyond face detection to other computer vision tasks, such as object detection and segmentation, particularly in similarly constrained computational settings.

2. Materials and Methods

2.1. WIDERFace Dataset

The WIDERFace dataset served as the primary training and evaluation benchmark for FeatherFace [17]. Comprising 32,203 images and 393,703 annotated face bounding boxes, WIDERFace captures an extensive array of real-world complexities, such as scale, pose, occlusion, expression, illumination, and makeup, across 61 event categories (Figure S1). The dataset was partitioned into training (40%), validation (10%), and test (50%) splits. During training, the model leveraged annotated bounding boxes, along with five key facial landmarks (eyes, nose, and mouth corners). WIDERFace also provides three difficulty subsets—Easy, Medium, and Hard—that reflect increasing levels of detection challenges. These subsets facilitated a thorough performance assessment of the trained model under diverse real-world conditions.

2.2. Model Architecture

FeatherFace incorporates four interconnected components: a lightweight backbone for initial feature extraction, multiscale feature aggregation for integrating information across different resolutions, attention mechanisms that focus the model on pertinent spatial and channel-wise features, and a detection head for final bounding-box regression and classification (Figure 1a). This design achieves robust face detection while maintaining minimal computational overhead, making it well suited for resource-constrained environments.

2.2.1. Backbone

In this study, the MobileNet-0.25 backbone was selected for its exceptionally lightweight architecture and computational efficiency. By leveraging depthwise separable convolutions, this network substantially reduces the number of parameters and associated computational cost compared with traditional convolutional networks, while preserving robust feature-extraction capabilities. It has a parameter footprint that is approximately fifty times smaller than that of ResNet50, but maintains consistent performance in feature extraction. This compact design makes it particularly advantageous for deployment on mobile or embedded devices, where computational and memory resources are often constrained.

2.2.2. Multiscale Feature Aggregation

The MobileNet-0.25 backbone generates three levels of image feature pyramids: P3, P4, and P5. P5 represents the most semantically rich and abstracted features, but has the lowest spatial resolution. Conversely, P3 offers the highest spatial resolution with detailed fine-grained features, but lacks substantial semantic context. Therefore, effective aggregation of these feature levels is critical for robust face detection, particularly in addressing challenges posed by images with varying face sizes across images.

FeatherFace employs the BiFPN for multiscale feature aggregation [25]. The BiFPN enables efficient bidirectional information flow between higher- and lower-level feature maps through top-down and bottom-up pathways (Figure 1a). The top-down pathway enhances lower-level features by infusing them with semantic information from higher-level feature maps, while the bottom-up pathway transfers finer-grained details from lower-level feature maps to higher-level representations. The BiFPN further refines this process, using weighted feature fusion, where learnable weights dynamically adjust the relative importance of feature maps at different scales. This approach increases the adaptability of the fusion mechanism, enabling the network to capture intricate interdependencies across varying feature resolutions. Consequently, the combined feature maps exhibit improved richness and robustness, particularly in challenging multiscale detection scenarios.

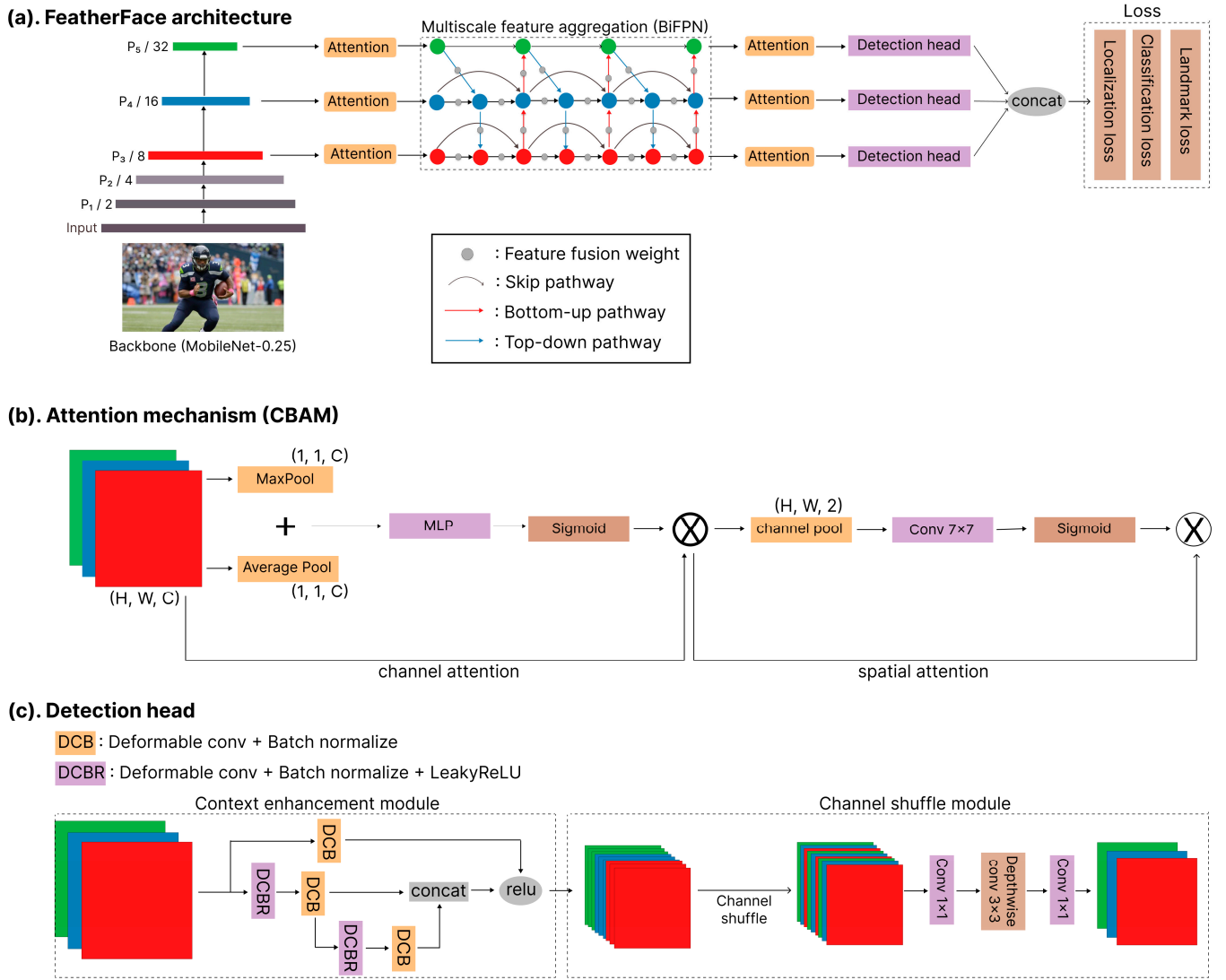


Figure 1. Overview of the FeatherFace architecture and its components. (a) FeatherFace integrates a MobileNet-0.25 backbone, attention mechanisms, multiscale feature aggregation, and detection heads. The integration of these modules jointly enhances feature representation, significantly improving the model’s accuracy and robustness. (b) The convolutional block attention module (CBAM) applies both channel and spatial attention to refine features critical for accurate face detection. (c) The detection heads incorporate a context enhancement module, which uses deformable convolutional networks (DCNs) to capture multiscale contextual information, and a channel shuffling module to facilitate effective inter-channel information exchange, further enriching feature representation.

Skip connections complement the BiFPN by mitigating information loss during feature fusion. In the bottom-up pathway, they help preserve fine-grained details that would otherwise be lost in the aggregation process. Similarly, in the top-down pathway, skip connections help restore essential semantic information to lower-level features. Additionally, these connections address gradient vanishing issues in deep networks, improving training stability and convergence. When integrated with the BiFPN’s bidirectional fusion, skip connections enable seamless and efficient information transfer across feature levels.

Formally, given multi-scale feature maps $(P_3^{in}, P_4^{in}, P_5^{in})$, where P_i^{in} represents the input feature map at level i , the aggregated new feature map P_i^{out} is computed as follows:

$$P_i^{td} = \text{Conv} \left(\frac{w_1 \cdot P_i^{in} + w_2 \cdot \text{Upsample}(P_{i+1}^{in})}{w_1 + w_2 + \epsilon} \right) \quad (1)$$

$$P_i^{out} = Conv \left(\frac{w'_1 \cdot P_i^{in} + w'_2 \cdot P_i^{td} + w'_3 \cdot Downsample(P_{i-1}^{out})}{w'_1 + w'_2 + w'_3 + \epsilon} \right) \quad (2)$$

where P_i^{td} is the intermediate top-down feature at level i , $Conv$ denotes the convolutional operation, and w_i and w'_i are learnable weights. Max pooling is used for the *Downsample* operation in the bottom-down pathway, while ϵ ensures numerical stability.

Depthwise separable convolutions are utilized for the convolutional operation to enhance computational efficiency, with batch normalization and SiLU (swish) activation applied after each convolution. This configuration delivers high performance while minimizing computational overhead, enabling the model to merge features effectively within a compact computational footprint. As a result, the architecture is highly suitable for deployment in resource-constrained environments.

While the BiFPN was ultimately chosen for its superior performance, alternative multiscale feature aggregation strategies, such as the FPN and the WFPN, were also evaluated during development [24,26]. Through this comprehensive comparative analysis, this study aimed to identify the most optimized feature-integration strategies for this task.

2.2.3. Attention Mechanism

FeatherFace incorporates the CBAM to enhance feature representation and emphasize critical aspects for face detection [27]. The CBAM dynamically refines feature maps by prioritizing significant information and suppressing irrelevant details, resulting in more robust feature representations and improved detection accuracy in challenging real-world scenarios.

The CBAM introduces a dual attention mechanism that sequentially applies channel and spatial attention (Figure 1b). The channel attention module identifies the importance of individual feature map channels by employing global average pooling and max pooling. These operations aggregate spatial information across the feature map, producing scalar values that capture the significance of each channel. These values are processed through a multilayer perceptron (MLP) and normalized using a sigmoid activation function to generate channel attention weights. These weights are then applied to the input feature map via elementwise multiplication, selectively enhancing critical channels.

The spatial attention module focuses on identifying and emphasizing salient spatial regions within the feature map. During the channel pool step, max pooling and average pooling are applied along the channel dimension to summarize critical information for each spatial location, producing two spatial summary maps. These maps are concatenated and passed through a convolutional layer to generate a spatial attention map, which highlights essential spatial regions. The spatial attention map is normalized to the range [0, 1] using a sigmoid activation function and applied elementwise to the input feature map, ensuring that significant spatial locations are emphasized while irrelevant areas are suppressed.

Formally, given an input feature map $\mathbf{F} \in \mathbb{R}^{C \times H \times W}$, where C , H , and W represent the number of channels, height, and width, respectively, CBAM computes a channel attention map $\mathbf{M}_c \in \mathbb{R}^{C \times 1 \times 1}$ and a spatial attention map $\mathbf{M}_s \in \mathbb{R}^{1 \times H \times W}$, as follows:

$$\mathbf{M}_c(\mathbf{F}) = \sigma(MLP(AvgPool(\mathbf{F})) + MLP(MaxPool(\mathbf{F}))) \quad (3)$$

$$\mathbf{M}_s(\mathbf{F}) = \sigma(Conv_{7 \times 7}([AvgPool(\mathbf{F}); MaxPool(\mathbf{F})])) \quad (4)$$

where σ denotes the sigmoid function, $Conv_{7 \times 7}$ represents a convolution operation with a filter size of 7×7 , and $[\cdot; \cdot]$ denotes concatenation. *AvgPool* and *MaxPool* operations for computing the spatial attention map are applied along the channel dimension, while these operations for the channel attention map involve pooling across the spatial dimensions.

The output $\mathbf{F}''$ of the CBAM is computed as

$$\mathbf{F}' = \mathbf{M}_c(\mathbf{F}) \odot \mathbf{F} \quad (5)$$

$$\mathbf{F}'' = \mathbf{M}_s(\mathbf{F}') \odot \mathbf{F}' \quad (6)$$

where $\odot$ denotes element-wise multiplication.

While the CBAM was selected for FeatherFace due to its superior performance, alternative attention mechanisms, such as the SENet, were also evaluated during development, to identify the most optimal approach [28]. By incorporating CBAM, FeatherFace achieves an excellent balance between enhanced feature representation and computational efficiency, further reinforcing its suitability for resource-constrained environments.

2.2.4. Detection Head

The detection head is a critical component responsible for the final regression of bounding boxes and the classification of detected faces. It integrates a context enhancement module and a channel shuffling module to enrich feature representations, enabling the precise localization and accurate detection of faces. These modules work in tandem to capture diverse contextual information and promote effective inter-channel interactions, ensuring robust performance while maintaining computational efficiency.

The context enhancement module plays a pivotal role in extracting multiscale contextual features essential for accurately delineating bounding boxes. Inspired by the single-stage headless (SSH) framework [10], this module employs multiple convolutions with varying receptive fields—such as 3×3 , 5×5 , and 7×7 —to process features at different spatial scales (Figure 1c). This multi-receptive field approach equips the network to capture diverse object contexts effectively, which is critical for accurately identifying faces across varying sizes and poses.

Additionally, we explored incorporating deformable convolutional networks (DCNs) as a replacement for standard convolutions, to further enhance the flexibility and adaptability of convolution operations [39,40]. Standard convolutions rely on fixed kernel sizes and shapes (e.g., 3×3 and 5×5) to sample input data, which inherently limits their ability to effectively model non-rigid transformations and complex geometric patterns. In contrast, deformable convolution overcomes this limitation by introducing learnable offsets to each kernel position. These offsets allow the sampling locations to be dynamically adjusted based on the input data, enabling the network to capture broader and more adaptable receptive fields. This capability significantly improves the model's ability to represent intricate object structures and handle diverse geometric transformations, thereby enhancing its effectiveness in challenging detection scenarios.

The channel shuffle module, inspired by ShuffleNet V2 [41], was employed to enhance feature representation by efficiently facilitating information exchange between channels in a lightweight deep learning architecture. This technique aims to increase the network's representational power while minimizing computational overhead. The channel shuffle mechanism operates by dividing the input tensor into multiple groups and shuffling the channels across these groups to promote stronger inter-channel interactions (Figure 1c). This is followed by a 1×1 convolution to reduce channel dimensions, a depthwise convolution (3×3 kernel, stride 1) for spatial refinement, and another 1×1 convolution to restore the channel count. This approach boosts representational power while maintaining computational efficiency, making it ideal for resource-constrained applications.

2.3. Training Protocol

The model was trained using fixed input image sizes of 640×640 . Data augmentation techniques, including random cropping, color distortion (brightness, contrast, and saturation adjustments), expansion, horizontal flipping, padding to square, resizing, and normalization, were employed to increase the diversity of the training data. An initial learning rate of 0.001 was used, with dynamic adjustments governed by the OneCycleLR scheduler. Optimization was performed using the AdamW optimizer. Training was conducted on an NVIDIA RTX A6000 GPU with 48 GB of memory, utilizing a batch size of 32 for 350 epochs.

Loss Function

The training employed a multitask loss function comprising localization loss (L_{loc}), classification loss (L_{cls}), and landmark loss (L_{landm}). The total loss L was defined as $L = 2 \times L_{loc} + L_{cls} + L_{landm}$, prioritizing localization accuracy.

The localization loss aims to minimize positional discrepancies between the predicted bounding boxes and ground-truth bounding boxes. The intersection over union (IoU) metric was used to measure the overlap between prior boxes and ground-truth bounding boxes. Prior boxes with an IoU exceeding the threshold of 0.35 are designated as positive samples, whereas the other boxes are treated as negatives. For the positive boxes, Smooth L1 loss was applied to compute the difference between the predicted and ground-truth coordinates, effectively refining the bounding-box predictions. The relatively low IoU threshold of 0.35 was employed to learn from a wider range of examples, including those with less precise overlaps, thereby enhancing its robustness and generalization capability.

The classification loss focused on reducing the discrepancy between the predicted class probabilities and ground-truth labels. Positive samples, identified based on the IoU and threshold criteria, were linked to their corresponding ground-truth labels. The hard negative mining technique was applied to address the imbalance caused by excessive negative samples, selecting a subset of negative samples with the highest loss values for inclusion in training. Both positive and selected negative samples were then processed using the Cross Entropy loss function to calculate the classification loss, ensuring the model's ability to distinguish facial features effectively.

The landmark loss facilitated the precise prediction of facial landmarks, such as the eyes, nose, and lips, which are essential for face detection and recognition tasks. Training on landmark coordinates was restricted to positive samples that met the IoU threshold ($\text{IoU} \geq 0.35$). Smooth L1 loss was employed to compute the differences between the predicted and ground-truth landmark coordinates. The loss was summed across all positive samples and normalized by their total count, to derive the final landmark loss. When optimized alongside localization and classification losses, landmark loss enhances the model's ability to accurately estimate facial features, improving overall detection performance.

2.4. Evaluation

The trained FeatherFace model was evaluated using the original images without scaling, with the pixel values normalized by subtracting the mean values [104, 117, 123]. The evaluation procedure comprised three sequential steps: applying the non-maximum suppression (NMS) algorithm, generating the precision–recall curve, and calculating the average precision (AP).

NMS was utilized to eliminate redundant bounding boxes while retaining only the most confident detections. Specifically, bounding boxes with confidence scores below 0.02 were discarded. Among the remaining boxes, those with an intersection over union (IoU) value greater than 0.4 were suppressed, to ensure unique and accurate predictions.

For the precision–recall curve, precision and recall values were calculated at confidence thresholds ranging from 0.02 to 1.0, in increments of 0.001. Precision was defined as $TP/(TP + FP)$ and recall as $TP/(TP + FN)$, where TP, FP, and FN represent the counts of true positives, false positives, and false negatives, respectively. A detection was considered a true positive if its IoU with the ground truth was 0.5 or higher. By plotting recall on the x -axis and precision on the y -axis, the curve illustrates the trade-off between precision and recall across varying confidence levels.

The AP was determined as the area under the precision–recall curve across all evaluated images. A stricter IoU threshold of 0.5 was applied during the evaluation, to ensure that only detections with a high overlap with the ground truth were considered true positives. In comparison, the IoU threshold for NMS (0.4) was an intermediate value between the training threshold (0.35) and the AP calculation threshold (0.5), balancing the removal of redundant detections with the preservation of valid ones.

Finally, the number of floating-point operations (FLOPs) and total parameters were measured for an input image of size 640×640 , to evaluate the computational efficiency and model compactness. This analysis provides practical insight into the trade-off between detection accuracy and computational resource demands, offering a comprehensive understanding of the model’s suitability for resource-constrained applications.

3. Results

To rigorously assess the effectiveness of the proposed FeatherFace architecture, comprehensive experiments were conducted on the WIDERFace benchmark dataset. Table 1 illustrates incremental modifications—from a simplified RetinaFace baseline to the fully optimized FeatherFace—and demonstrates the resulting improvements in detection accuracy and model compactness.

Table 1. The performance of the FeatherFace face-detection architecture on the WIDERFace validation set.

Architecture						WIDERFace AP (%)		
Feature Aggregation	Attention Mechanism	Attention Placement	Detection Heads	Params	FLOPs	Easy	Medium	Hard
FPN	-	-	Conv	426.6 K	1.030 G	90.5	87.5	74.3
WFPN	-	-	Conv	426.7 K	1.030 G	88.6	85.9	73.7
BiFPN	-	-	Conv	436.1 K	900.8 M	90.9	88.3	74.6
BiFPN	SENet	Backbone + BiFPN	Conv	448.4 K	902.1 M	92.0	89.5	76.4
BiFPN	CBAM	Backbone + BiFPN	Conv	449.7 K	902.5 M	92.2	89.8	76.6
BiFPN	CBAM	Backbone	Conv	447.7 K	901.7 M	91.6	88.8	75.4
BiFPN	CBAM	Backbone + BiFPN + SSH	Conv	451.8 K	903.4 M	90.6	88.9	75.1
BiFPN	CBAM	Backbone + BiFPN	DCN	474.7 K	971.5 M	92.3	90.0	77.5
BiFPN	CBAM	Backbone + BiFPN	DCN + shuffle *	488.7 K	1.013 G	92.7	90.7	78.3

* FeatherFace architecture.

3.1. Performance Evaluation of FeatherFace Architecture

The baseline configuration employed a MobileNet-0.25 backbone with FPN-based feature aggregation and utilized conventional convolutional heads (Conv) for classification and bounding-box regression. This simplified RetinaFace setup achieved AP values of 87.5% on the Medium subset and 74.3% on the Hard subset (Table 1). While this served as a reasonable starting point, the baseline exhibited limitations when dealing with small or occluded faces, and lacked the computational efficiency demanded by resource-constrained environments.

To enhance the model's ability to effectively combine multiscale facial features extracted from the backbone, the baseline FPN was replaced with the BiFPN, a feature aggregation module designed to adaptively learn the relative importance of features at different scales (Figure 2a). As shown in Table 1, the BiFPN provided consistent performance gains across the Easy, Medium, and Hard subsets. For instance, the Medium subset AP improved from 87.5% to 88.3%, indicating a clear advantage over the baseline. In contrast, substituting the FPN with the WFPN showed negligible benefits, suggesting that not all multiscale feature-integration methods are equally effective in ultra-lightweight settings.

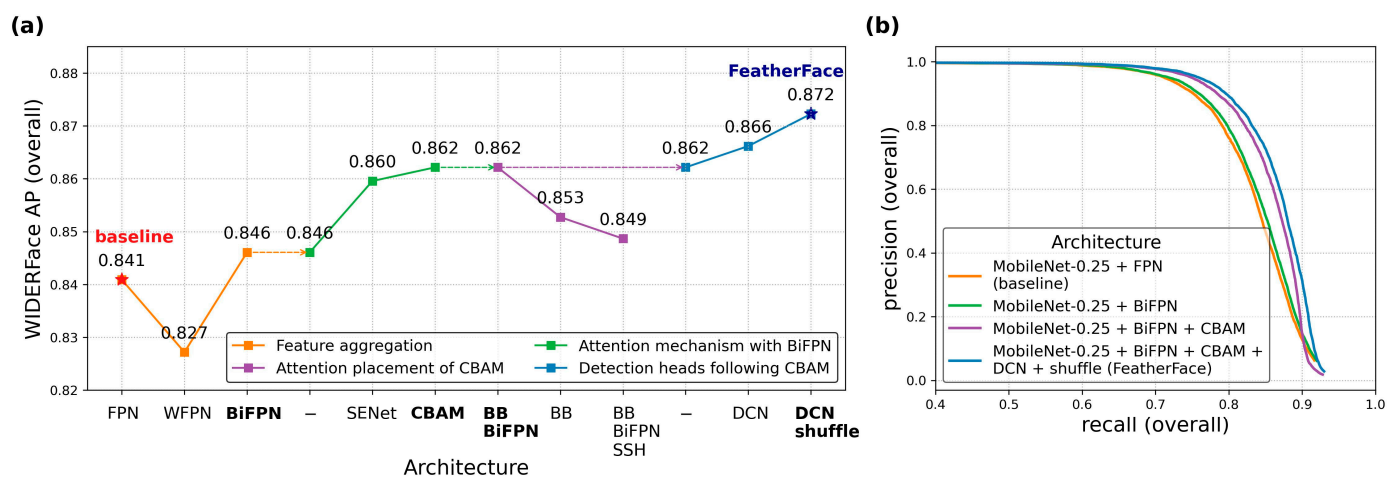


Figure 2. (a) The evolution of the FeatherFace architecture via intermediate phases, illustrated with the overall AP achieved at each phase of development. The process consists of four sequential phases—feature aggregation, attention mechanism, attention placement, and detection heads—where the top-performing configuration from each phase was selected to guide the subsequent phase. The final FeatherFace architecture integrates a MobileNet-0.25 backbone, the bidirectional feature pyramid network (BiFPN) for multiscale feature aggregation, CBAM attention mechanisms applied to both the backbone and BiFPN output, and detection heads enhanced with DCNs and channel shuffling. The overall average precision (AP) represents the model's performance across the Easy, Medium, and Hard subsets of the WIDERFace dataset. Key architectures selected at each phase are highlighted in bold. Starred architectures denote the baseline (beginning) and final FeatherFace (end) architectures, marking the start and culmination of the optimization process. BB: backbone. (b) The overall precision–recall curves comparing the baseline and three key architectures. Additional details, including phase-wise AP and precision–recall curves for the Easy, Medium, and Hard subsets, are presented in Figure S2.

To further refine the model's feature representation, attention mechanisms were incorporated at various stages of the network (Figure 2a). Integrating the CBAM into both the backbone and BiFPN outputs led to significant performance gains, particularly on the Hard subset, where AP improved from 74.6% to 76.6% (Table 1). Placing the CBAM after both the backbone and BiFPN resulted in the most substantial improvements, outperforming other configurations such as positioning the module exclusively after the backbone or across multiple network stages. These results emphasize the importance of leveraging attention

mechanisms to enhance feature representation by emphasizing informative spatial and channel-wise features.

The final FeatherFace architecture further incorporates deformable convolutions and channel shuffling, leading to a substantial performance boost (Figure 2a). FeatherFace achieves AP scores of 92.7%, 90.7%, and 78.3% on the Easy, Medium, and Hard subsets, respectively, demonstrating a clear advantage over both the baseline and intermediate configurations. The total 4.0% AP improvement on the WIDERFace Hard subset over the simplified RetinaFace baseline underscores the robustness of the FeatherFace architecture in challenging face-detection scenarios. Beyond these AP enhancements, FeatherFace demonstrates superior precision–recall curves compared with all intermediate and baseline architectures, further supporting its effectiveness and reliability (Figure 2b). These trends are consistently observed across individual analyses of the Easy, Medium, and Hard subsets, as highlighted in Figure S2, reinforcing the model’s comprehensive robustness across diverse detection challenges.

In summary, FeatherFace achieves an overall AP of 87.2% on the WIDERFace dataset, while maintaining a parameter count of just 0.489 M—a substantial reduction compared with the 25 M parameters of the original RetinaFace. This compact architecture makes the model exceptionally well suited for deployment in resource-constrained environments and applications requiring real-time performance. These results demonstrate that the optimized feature-integration strategies—including multiscale feature aggregation, attention mechanisms, deformable convolutions, and channel shuffling—are instrumental in achieving high detection accuracy while substantially minimizing computational resource requirements.

3.2. Robustness of FeatherFace Across Six Challenging Scenarios

To evaluate the robustness of FeatherFace, we analyzed its performance across six challenging scenarios in the WIDERFace Hard subset: blur, expression, illumination, occlusion, pose, and scale. These scenarios represent common challenges in real-world face detection, where varying image qualities and conditions can significantly impact detection accuracy.

As illustrated in Figure 3a, FeatherFace consistently outperformed the simplified RetinaFace baseline across all six scenarios. Notable improvements were observed in handling expression variations, where FeatherFace achieved a significantly higher AP of 95.5% compared with the baseline’s 88.5%. Additionally, FeatherFace demonstrated superior performance in detecting occluded and smaller faces, with AP scores of 64.3% and 62.9%, respectively, compared with the baseline’s 57.6% and 58.4%, respectively.

In addition to the quantitative results, the qualitative comparisons depicted in Figure 3b illustrate FeatherFace’s enhanced bounding-box precision relative to the baseline. FeatherFace consistently identified a greater number of correctly detected faces in challenging scenarios such as occluded, blurred, or smaller faces, effectively reducing both false positives and false negatives. These visual examples further support the quantitative findings and highlight the practical advantages of FeatherFace in real-world applications.

These results validate FeatherFace’s robustness across a diverse range of challenging scenarios, including exaggerated expressions, partially occluded faces, numerous small faces, and motion or focus blur. Its consistent outperformance of the baseline highlights the effectiveness of optimized feature-integration strategies in enhancing facial representation, thereby addressing real-world challenges in face detection.

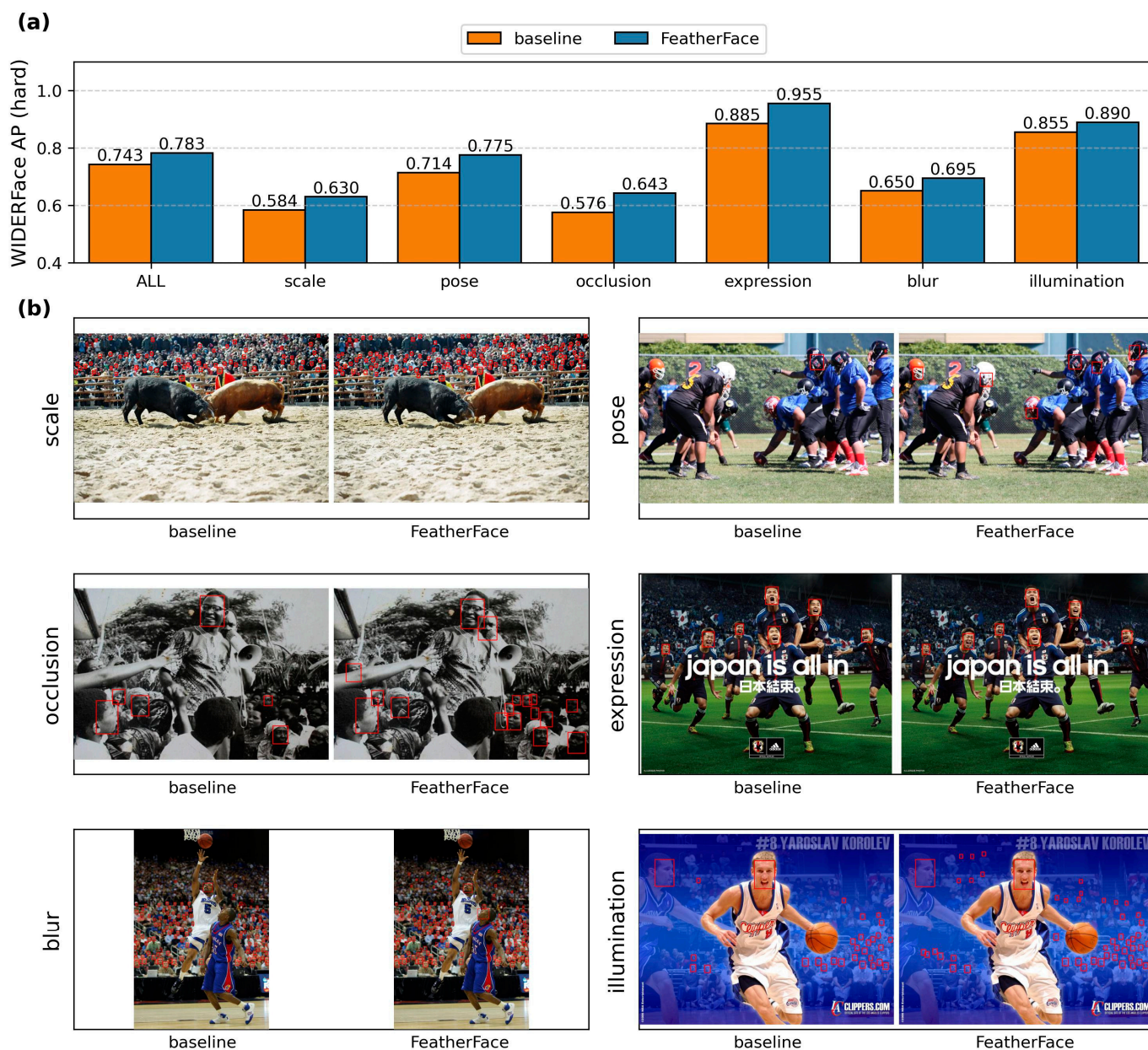


Figure 3. (a) Comparison of APs between the baseline and FeatherFace architectures across six challenging scenarios. Images were categorized based on the dominant challenge they presented, with a category assigned if a specific challenge-related facial box constituted more than 50% of the total facial boxes in the image (allowing for multiple category mappings). FeatherFace consistently outperformed the baseline across all scenarios, demonstrating superior detection accuracy under diverse and demanding conditions. (b) Representative images demonstrating the prediction results of the baseline and FeatherFace models for each scenario. Red boxes indicate the predicted facial boxes generated by each architecture.

3.3. Ablation Studies

An ablation study was conducted to evaluate the influence of key design choices in the FeatherFace architecture, focusing on two critical aspects: multiscale feature inputs to the BiFPN and the depth of the BiFPN layers.

3.3.1. Multiscale Feature Inputs to Bidirectional Feature Pyramid Network (BiFPN)

The BiFPN module originally utilized multiscale feature inputs from three feature pyramid levels, denoted P3, P4, and P5, where P3 represents the feature level with the

highest spatial resolution. To assess the contribution of each feature level, various configurations were tested by selectively excluding certain inputs. The tested configurations included single-level inputs (P3), (P4), and (P5), pairwise combinations (P3, P4), (P4, P5), and (P3, P5), and the original full input configuration (P3, P4, P5).

The BiFPN layers were removed when a single feature level was used, as there was no need for multiscale aggregation. The results show that P3, the highest-resolution feature level, performed better on the Hard subset (AP of 60.4%) than P5, which achieved only 46.9% (Table 2). This outcome aligns with our expectations, as the WIDERFace Hard subset predominantly includes smaller faces, where features with high spatial resolution (such as those from P3) are better suited for accurate detection. Conversely, P5, which represents the lowest resolution but more semantically rich feature level, delivered the highest performance on the Easy subset, achieving an AP of 40.6%, significantly outperforming P3, which achieved only 23.0%. This result suggests that lower-resolution features are more effective for detecting larger faces, which are more prevalent in the Easy subset.

Table 2. The ablation study results for the FeatherFace architecture, analyzing the impacts of different multiscale feature input configurations and the number of BiFPN layers.

Feature Inputs	Number of BiFPN Layers	Params	FLOPs	Easy	Medium	Hard
P3	0	60.62 K	489.4 M	23.0	55.0	60.4
P4	0	192.7 K	462.0 M	29.9	59.1	58.9
P5	0	364.8 K	431.4 M	40.6	61.5	46.9
P3, P4	3	263.7 K	907.4 M	85.6	86.6	75.9
P4, P5	3	395.6 K	531.0 M	80.6	82.3	68.2
P3, P5	3	389.9 K	857.8 M	83.8	86.4	77.0
P3, P4, P5	1	449.7 K	914.0 M	92.5	90.3	77.8
P3, P4, P5	2	469.2 K	963.3 M	92.8	90.4	78.1
P3, P4, P5	3 *	488.7 K	1.013 G	92.7	90.7	78.3
P3, P4, P5	4	508.1 K	1.062 G	92.7	90.4	77.9

* FeatherFace architecture.

The pairwise combinations of the feature levels demonstrated similar trends. Combinations that included P3 consistently achieved higher performance on the Hard subset, whereas those that included P5 showed superior performance on the Easy subset. Notably, pairwise combinations of feature levels resulted in significantly higher AP scores (average of 80.7%) than configurations using single feature levels (average of 48.4%). This finding underscores the importance of optimally integrating features from multiple scales to address the diverse face sizes and scales present in real-world datasets.

3.3.2. Reduction in Number of BiFPN Layers

The original three-layer configuration was adjusted to include one, two, and four layers, to analyze the effect of the BiFPN layer depth. Overall, the number of BiFPN layers had a relatively minor impact on the detection performance, with AP differences between configurations being less than 0.4% in any pairwise comparison (Table 2). The three-layer BiFPN configuration achieved the highest AP performance across subsets, but the single-layer BiFPN configuration offered a viable alternative for scenarios prioritizing model compactness, with only a marginal decrease in performance.

3.4. Comparative Analysis with Existing Lightweight Face Detectors

To further evaluate the performance and efficiency of FeatherFace, we compared it with various lightweight and high-performance face detectors, as summarized in Table 3. The comparison includes complex architectures with high parameter counts (e.g., RetinaFace),

lightweight models with medium parameter counts ranging from 1 M to 7 M, and ultra-lightweight models with approximately 0.5 M parameters or fewer.

Table 3. Comparative performance of FeatherFace and existing lightweight face detectors on WIDER-Face dataset. Metrics include AP on Easy, Medium, and Hard subsets, along with parameter counts and floating-point operations (FLOPs).

Architecture			WIDERFace AP (%)			
Face Detector	Params	FLOPs	Easy	Medium	Hard	Overall
RetinaFace [9]	25.1 M	32.2 G	95.6	94.2	86.3	92.0
YOLOv5 s [36]	7.08 M	5.75 G	94.3	92.6	83.2	90.0
SCRFD-10 GF [37]	3.86 M	9.98 G	95.2	93.9	83.1	90.7
LFFDv2 [32]	1.45 M	6.87 G	83.7	83.5	72.9	80.0
YOLOv5n [36]	1.73 M	2.11 G	93.6	91.5	80.5	88.5
FaceBoxes [30]	1.01 M	0.275 G	76.2	57.2	24.2	52.5
SCRFD-0.5GF [37]	0.57 M	0.508 G	90.6	88.1	68.5	82.4
MTCNN [42]	0.50 M	4.6 G	85.1	82.0	60.7	75.9
YOLOv5n0.5 [36]	0.447 M	0.571 G	90.8	88.1	73.8	84.2
FeatherFace (ours)	0.489 M	1.01 G	92.7	90.7	78.3	87.2

FeatherFace demonstrated competitive performance among models with similar parameter counts and computational budgets. Specifically, it achieved an overall AP of 87.2% with just 0.49 M parameters, compared with SCRFD-0.5GF (82.4%), MTCNN (75.9%), and YOLOv5n0.5 (84.2%). While heavier models such as RetinaFace delivered higher absolute AP scores, their significantly larger parameter counts and computational requirements make them less suitable for deployment in highly resource-constrained environments.

These results suggest that FeatherFace effectively balances efficiency and detection accuracy, making it a strong candidate for deployment in real-world applications requiring lightweight and scalable solutions. By maintaining a minimal parameter count and competitive detection accuracy, FeatherFace highlights the potential of optimized feature-integration strategies for efficient face detection in resource-limited scenarios.

3.5. Generalization of FeatherFace Core Across Diverse Backbones

To evaluate the versatility and effectiveness of the FeatherFace Core components—comprising BiFPN, CBAM, deformable convolutions, and channel shuffling—when applied to various backbone networks, additional experiments were conducted using four distinct backbones: Darknet, EfficientNet, MnasNet-0.5, and ShuffleNet-V2-0.25. Table 4 summarizes the comparative results, highlighting the performance improvements achieved by integrating the FeatherFace Core with each backbone.

The results demonstrate consistent performance improvements across all backbones upon integrating the FeatherFace Core. Its application to MnasNet-0.5 resulted in a substantial increase in overall AP, from 63.0% to 86.2%. This improvement was achieved with only a marginal increase in computational complexity, as reflected in the modest changes to parameter count and FLOPs. Similar trends were observed across all other backbones, underscoring the versatility and effectiveness of the FeatherFace Core modules. This generalizability highlights the potential of the FeatherFace Core as a modular solution for building robust and efficient face detection models tailored to diverse deployment scenarios, including resource-constrained environments.

Table 4. Effectiveness of FeatherFace Core when applied to various backbone networks.

Backbone	Network Configuration	Params	FLOPs	Easy	Medium	Hard	Overall
CSPDarknet53	Baseline (RetinaFace core)	29.9 M	51.0 G	94.7	93.5	87.1	91.8
	+FeatherFace Core	29.2 M	46.1 G	95.2	93.9	88.0	92.3
EfficientNet-B0	Baseline (RetinaFace core)	3.88 M	3.96 G	93.0	91.5	83.9	89.5
	+FeatherFace Core	3.93 M	3.81 G	93.5	92.0	84.5	90.0
MnasNet-0.5	Baseline (RetinaFace core)	0.607 M	0.893 G	69.2	71.2	48.6	63.0
	+FeatherFace Core	0.649 M	0.933 G	91.9	89.8	76.9	86.2
ShuffleNet-V2-0.25	Baseline (RetinaFace core)	0.265 M	0.649 G	90.7	87.6	72.6	83.6
	+FeatherFace Core	0.326 M	0.713 G	92.2	89.9	75.9	86.0
MobileNet-0.25	Baseline (RetinaFace core)	0.427 M	1.03 G	90.5	87.5	74.3	84.1
	+FeatherFace Core	0.489 M	1.01 G	92.7	90.7	78.3	87.2

4. Discussion and Conclusions

In this study, we introduced FeatherFace, a lightweight yet robust face detection framework designed for resource-constrained environments. FeatherFace leverages a combination of advanced feature-integration strategies, including multiscale feature aggregation, attention mechanisms, deformable convolutions, and channel shuffling, to achieve high detection accuracy with an exceptionally low parameter count of 0.49 M. These design choices address key challenges in face detection—such as blurring, small faces, and partial occlusions—making FeatherFace well suited for complex real-world applications. Furthermore, its compact size paves the way for broader deployment on edge devices, embedded systems, and real-time monitoring platforms where computational resources are severely limited.

Building upon the strengths of state-of-the-art frameworks such as RetinaFace and leveraging lightweight backbones such as MobileNet, FeatherFace incorporates feature-integration strategies such as BiFPN, CBAM, DCN, and channel shuffling to jointly enhance detection robustness and computational efficiency. This combination of modules, collectively referred to as the FeatherFace Core, has also demonstrated its effectiveness and adaptability across diverse backbone architectures, as detailed in Section 3.5.

Our experiments on the WIDERFace dataset highlight its notable performance gains, especially on the Hard subset, which represents the most challenging real-world scenarios. By achieving an overall AP of 87.2% with an exceptionally compact model size, FeatherFace bridges the gap between high-performing but computationally expensive models and those optimized for efficiency.

Ablation studies revealed the pivotal role of strategic feature fusion, underscoring how high-resolution features (P3) help detect small faces, whereas more semantically rich, lower-resolution features (P4, P5) enhance the detection of larger faces. These findings illustrate how thoughtful multiscale feature integration can significantly boost face detection performance, especially under challenging face detection scenarios.

Future work could explore smaller and more efficient backbone networks to further reduce parameter counts while sustaining performance. Automated neural architecture search (NAS) offers another promising direction for tailoring the FeatherFace design to diverse hardware profiles systematically. Knowledge distillation or model pruning may also be employed to maintain high accuracy with lower computational overhead and faster inference times.

FeatherFace exemplifies how architectural optimizations can simultaneously achieve robustness and efficiency in face detection. These design principles can potentially be extended to a broader range of vision tasks, including object detection and semantic segmentation, particularly in similarly constrained computational environments. We believe that the strategies demonstrated in this work provide a foundation for developing resource-efficient and high-performing models capable of meeting the practical demands of real-world scenarios, thereby advancing the frontier of lightweight computer vision systems.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/electronics14030517/s1>, Figure S1: Representative images from the WIDERFace dataset showcasing various challenging scenarios in face detection, including scale, pose, occlusion, expression, makeup, and illumination; Figure S2: The phase-wise AP and the precision-recall curves for the Easy, Medium, and Hard subsets of the WIDERFace dataset.

Author Contributions: Conceptualization, J.J. and D.K.; methodology, D.K. and J.K.; software and experiments, D.K.; validation and investigation, J.K. and D.K.; writing—original draft preparation, J.K.; writing—review and editing, D.K., J.J. and J.K.; visualization, J.J.; supervision, J.J. and J.K.; project administration, J.J. and J.K.; funding acquisition, J.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by a National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (NRF-2022R1C1C1008823).

Data Availability Statement: The WIDERFace dataset can be accessed and downloaded from its official website (<http://shuoyang1213.me/WIDERFACE> (accessed on 25 January 2025)). The PyTorch implementation of FeatherFace is publicly available on GitHub (<https://github.com/dohun-mat/FeatherFace> (accessed on 25 January 2025)).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AP	Average precision
FPN	Feature pyramid network
BiFPN	Bidirectional feature pyramid network
CBAM	Convolutional block attention module
SENet	Squeeze-and-excitation network
SSH	Single-stage headless
DCN	Deformable convolutional network
NMS	Non-maximum suppression
FLOPs	Floating-point operations

References

1. Viola, P.; Jones, M.J. Robust Real-Time Face Detection. *Int. J. Comput. Vis.* **2004**, *57*, 137–154. [[CrossRef](#)]
2. Schroff, F.; Kalenichenko, D.; Philbin, J. FaceNet: A Unified Embedding for Face Recognition and Clustering. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 815–823.
3. Zhang, F.; Zhang, T.; Mao, Q.; Xu, C. Joint Pose and Expression Modeling for Facial Expression Recognition. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 3359–3368.
4. Deng, J.; Guo, J.; Xue, N.; Zafeiriou, S. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.

5. Wang, H.; Wang, Y.; Zhou, Z.; Ji, X.; Gong, D.; Zhou, J.; Li, Z.; Liu, W. CosFace: Large Margin Cosine Loss for Deep Face Recognition. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 5265–5274.
6. Liu, W.; Wen, Y.; Yu, Z.; Li, M.; Raj, B.; Song, L. SphereFace: Deep Hypersphere Embedding for Face Recognition. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 6738–6746.
7. Kong, C.; Chen, B.; Li, H.; Wang, S.; Rocha, A.; Kwong, S. Detect and Locate: Exposing Face Manipulation by Semantic- and Noise-Level Telltales. *IEEE Trans. Inform. Forensic Secur.* **2022**, *17*, 1741–1756. [[CrossRef](#)]
8. Yakovleva, O.; Kovtunen, A.; Liubchenko, V.; Honcharenko, V.; Kobylina, O. Face Detection for Video Surveillance-Based Security System. *COLINS* **2023**, 69–86.
9. Deng, J.; Guo, J.; Zhou, Y.; Yu, J.; Kotsia, I.; Zafeiriou, S. RetinaFace: Single-Stage Dense Face Localisation in the Wild. *arXiv* **2019**, arXiv:1905.00641.
10. Najibi, M.; Samangouei, P.; Chellappa, R.; Davis, L.S. SSH: Single Stage Headless Face Detector. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; IEEE: Piscataway, NJ, USA; pp. 4885–4894.
11. Hu, P.; Ramanan, D. Finding Tiny Faces. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1522–1530.
12. Zhang, S.; Zhu, X.; Lei, Z.; Shi, H.; Wang, X.; Li, S.Z. S3FD: Single Shot Scale-Invariant Face Detector. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; IEEE: Piscataway, NJ, USA; pp. 192–201.
13. Chi, C.; Zhang, S.; Xing, J.; Lei, Z.; Li, S.Z.; Zou, X. Selective Refinement Network for High Performance Face Detection. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 8231–8238. [[CrossRef](#)]
14. Minaee, S.; Luo, P.; Lin, Z.; Bowyer, K. Going Deeper Into Face Detection: A Survey. *arXiv* **2021**, arXiv:2103.14983.
15. Mamieva, D.; Abdusalomov, A.B.; Mukhiddinov, M.; Whangbo, T.K. Improved Face Detection Method via Learning Small Faces on Hard Images Based on a Deep Learning Approach. *Sensors* **2023**, *23*, 502. [[CrossRef](#)]
16. Li, J.; Zhang, B.; Wang, Y.; Tai, Y.; Zhang, Z.; Wang, C.; Li, J.; Huang, X.; Xia, Y. ASFD: Automatic and Scalable Face Detector. *arXiv* **2022**, arXiv:2201.1078.
17. Yang, S.; Luo, P.; Loy, C.C.; Tang, X. WIDER FACE: A Face Detection Benchmark. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 5525–5533.
18. Zhu, Y.; Cai, H.; Zhang, S.; Wang, C.; Xiong, Y. TinaFace: Strong but Simple Baseline for Face Detection. *arXiv* **2021**, arXiv:2011.13183.
19. Liu, Y.; Wang, F.; Deng, J.; Zhou, Z.; Sun, B.; Li, H. MogFace: Towards a Deeper Appreciation on Face Detection. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 4083–4092.
20. Meddeb, H.; Abdellaoui, Z.; Houaidi, F. Development of Surveillance Robot Based on Face Recognition Using Raspberry-PI and IOT. *Microprocess. Microsyst.* **2023**, *96*, 104728. [[CrossRef](#)]
21. George, A.; Ecabert, C.; Shahreza, H.O.; Kotwal, K.; Marcel, S. EdgeFace: Efficient Face Recognition Model for Edge Devices. *IEEE Trans. Biom. Behav. Identity Sci.* **2024**, *6*, 158–168. [[CrossRef](#)]
22. Mohammad, A.S.; Jarullah, T.G.; Al-Kaltakchi, M.T.S.; Alshehabi Al-Ani, J.; Dey, S. IoT-MFaceNet: Internet-of-Things-Based Face Recognition Using MobileNetV2 and FaceNet Deep-Learning Implementations on a Raspberry Pi-400. *J. Low Power Electron. Appl.* **2024**, *14*, 46. [[CrossRef](#)]
23. Naseri, R.A.S.; Kurnaz, A.; Farhan, H.M. Optimized Face Detector-Based Intelligent Face Mask Detection Model in IoT Using Deep Learning Approach. *Appl. Soft Comput.* **2023**, *134*, 109933. [[CrossRef](#)]
24. Lin, T.-Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 936–944.
25. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 10778–10787.
26. Qi, S.; Song, X.; Yang, J.; Jiang, C.; Xie, T. ACWFace: Efficient and Lightweight Face Detector Based on RetinaFace. *J. Electron. Imag.* **2022**, *31*, 013012. [[CrossRef](#)]
27. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In *Computer Vision—ECCV 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2018; Volume 11211, pp. 3–19. ISBN 978-3-030-01233-5.

28. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
29. Yu, Z.; Huang, H.; Chen, W.; Su, Y.; Liu, Y.; Wang, X. YOLO-FaceV2: A Scale and Occlusion Aware Face Detector. *Pattern Recognit.* **2024**, *155*, 110714. [[CrossRef](#)]
30. Zhang, S.; Zhu, X.; Lei, Z.; Shi, H.; Wang, X.; Li, S.Z. FaceBoxes: A CPU Real-Time Face Detector with High Accuracy. In Proceedings of the 2017 IEEE International Joint Conference on Biometrics (IJCB), Denver, CO, USA, 1–4 October 2017; IEEE: Piscataway, NJ, USA; pp. 1–9.
31. Wang, G.; Li, J.; Wu, Z.; Xu, J.; Shen, J.; Yang, W. EfficientFace: An Efficient Deep Network with Feature Enhancement for Accurate Face Detection. *Multimed. Syst.* **2023**, *29*, 2825–2839. [[CrossRef](#)]
32. He, Y.; Xu, D.; Wu, L.; Jian, M.; Xiang, S.; Pan, C. LFFD: A Light and Fast Face Detector for Edge Devices. *arXiv* **2019**, arXiv:1904.10633.
33. Zhang, Z.; Chao, Q.; Wang, S.; Yu, T. A Lightweight Face Detector via Bi-Stream Convolutional Neural Network and Vision Transformer. *Information* **2024**, *15*, 290. [[CrossRef](#)]
34. Liu, B.; Yu, H. A Lightweight and Accurate Face Detection Algorithm Based on Retinaface. *arXiv* **2023**, arXiv:2308.04340.
35. Liu, W.; Hasan, I.; Liao, S. Center and Scale Prediction: Anchor-Free Approach for Pedestrian and Face Detection. *Pattern Recognit.* **2023**, *135*, 109071. [[CrossRef](#)]
36. Qi, D.; Tan, W.; Yao, Q.; Liu, J. YOLO5Face: Why Reinventing a Face Detector. *arXiv* **2022**, arXiv:2105.12931.
37. Guo, J.; Deng, J.; Lattas, A.; Zafeiriou, S. Sample and Computation Redistribution for Efficient Face Detection. *arXiv* **2021**, arXiv:2105.04714.
38. Wu, W.; Peng, H.; Yu, S. YuNet: A Tiny Millisecond-Level Face Detector. *Mach. Intell. Res.* **2023**, *20*, 656–665. [[CrossRef](#)]
39. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable Convolutional Networks. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
40. Zhu, X.; Hu, H.; Lin, S.; Dai, J. Deformable ConvNets V2: More Deformable, Better Results. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
41. Ma, N.; Zhang, X.; Zheng, H.-T.; Sun, J. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. In *Computer Vision—ECCV 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2018; Volume 11218, pp. 122–138, ISBN 978-3-030-01263-2.
42. Zhang, K.; Zhang, Z.; Li, Z.; Qiao, Y. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal Process. Lett.* **2016**, *23*, 1499–1503. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.