electronics

MDPI

Article

Face Detection Based on DF-Net

Qijian Tang, Yanfei Li, Yinhe Cai, Xiang Peng and Xiaoli Liu *

Key Laboratory of Optoelectronic Devices and System of Ministry of Education and Guangdong Province, College of Physics and Optoelectronic Engineering, Shenzhen University, Shenzhen 518060, China
* Correspondence: lxl@szu.edu.cn

Abstract: Face data have found increasingly widespread applications in daily life. To efficiently and accurately extract face information from input images, this paper presents a DF-Net-based face detection approach. A lightweight facial feature extraction neural network based on the MobileNet-v2 architecture is designed and implemented. By incorporating multi-scale feature fusion and spatial pyramid modules, the system achieves face localization and extraction across multiple scales. The proposed network is trained on the open-source face detection dataset WiderFace. The hyperparameters such as bottleneck coefficients and quality factors are discussed. Comparative experiments with other commonly used networks are carried out in terms of network model size, processing speed, and network extraction accuracy. Experimental results affirm the efficacy and robustness of this method, especially in challenging facial poses.

Keywords: face detection; deep learning; DF-Net; lightweight; spatial pyramid module

1. Introduction

With the continuous development of societal technology, an increasing number of fields are utilizing facial data with authentic scale information, which offers higher robustness and richer details. These applications span domains such as film production, facial recognition, virtual reality, and medical fields [1–4]. The utilization of facial data is expanding, with the foremost and critical step being face detection. In any facial application system, the accuracy and speed of face detection directly affect the overall system's performance [5].

Facial detection can be categorized into two research directions [6]. The first is the traditional approach, which involves manually extracting features for facial detection. For instance, the Viola–Jones method [7] employs Haar feature extraction algorithms (linear features, edge features, center features, and diagonal features). However, traditional detection algorithms are not only time-consuming and labor intensive due to the need for manual feature extraction, but they also have limited feature representation capabilities. In complex environments, they often lack robust detection performance. With the introduction of convolutional neural networks (CNNs) in 2012 [8], led by Hinton and others, more and more researchers have delved into studying and innovating upon this technology. As a result, facial detection has seen significant advancements with the advent of deep learning. Facial detection algorithms based on deep learning can be divided into two main categories: (1) Two-stage methods, which first generate candidate regions and then use convolutional neural networks to predict the targets. These methods are known for their high accuracy but tend to be slower in terms of detection speed. (2) Single-stage methods, which directly predict targets using neural networks. These methods strike a balance between speed and accuracy. These advancements in deep learning have contributed to significant improvements in facial detection technology. However, facial detection is affected by factors such as environmental conditions and obstructions, which still present numerous challenges for achieving both speed and accuracy in detection.

Citation: Tang, Q.; Li, Y.; Cai, Y.; Peng, X.; Liu, X. Face Detection Based on DF-Net. *Electronics* **2023**, *12*, 4021. https://doi.org/10.3390/electronics12194021

Academic Editors: Hilario Gómez Moreno and Sergio Lafuente-Arroyo

Received: 15 August 2023
Revised: 20 September 2023
Accepted: 22 September 2023
Published: 24 September 2023

Addressing the current challenges of cumbersome deployment and sluggish network inference in face detection, this paper presents a novel facial extraction algorithm named Detection Face Net (DF-Net). It devises and implements a streamlined facial extraction neural network rooted in the MobileNet-v2 architecture. This network is endowed with multi-scale feature cascading and spatial pyramid modules, which collectively culminate in a proficient and precise face detection mechanism.

The ensuing sections are structured as follows: Section 2 elucidates the architecture of the designed DF-Net network, expounding upon the intricate details of its constituent modules. Section 3 verifies the proposed approach through experimentation, contrasting its outcomes with those of other methodologies. This comparative analysis substantiates the efficacy of the proposed algorithm. Finally, Section 4 concludes the manuscript.

2. Related Work

2.1. Face Detection Method

Due to the pivotal role of face detection, numerous researchers have proposed a range of related algorithms. In the early stages, most face detection algorithms relied on traditional feature extraction and classifier training processes. For instance, Viola and Jones introduced a face detection algorithm in 2001 capable of detecting front-facing faces, although its effectiveness on side profiles was limited [9]. Felzenszwalb et al. [10–12] presented a component-based object detection algorithm, known as Deformable Part Model (DPM), in 2008. While versatile in detecting faces of varying orientations and poses, the algorithm's complexity resulted in prolonged runtime. With the evolving landscape of deep learning in computer vision and the advancements in convolutional neural networks within ImageNet classification tasks [13–15], neural networks have progressively become the mainstream technology for target detection [16,17]. One noteworthy approach, the cascade CNN, blended traditional techniques with deep learning [18]. It built upon the foundation of the Viola–Jones algorithm [19], enhancing the classifier with convolutional networks to attain robust face detection outcomes. Expanding on this, the Multi-Task Convolutional Neural Network (MTCNN) extended the cascade CNN concept, employing multiple cascaded convolutional neural networks for face detection [20]. This method, while effective, presented deployment challenges due to its multi-cascade architecture. Face RCNN, an evolution of Faster RCNN proposed by Wang et al., further refined face detection [21]. By introducing online difficult sample mining and multi-scale training mechanisms, the network's face detection prowess was significantly augmented. Nonetheless, the introduction of several modules in the network somewhat compromised its inference speed. Researchers have subsequently introduced various approaches to enhance detection speed, such as the YOLO series (YOLOv6 [22], YOLOv7 [23]), RetinaFace [24], and more. Li et al. [25]. Proposed an improved anchor box matching method by integrating new data augmentation techniques and anchor design strategies into a dual-camera face detector, which provides better initialization for the regressor and consequently enhances face detection performance. Qi et al. [26]. Improved detection performance by using the Wing loss function and replacing the Focus module in the Backbone with the StemBlock module, building upon YOLOv5. While these methods improve detection speed, it is important to note that they often come at the cost of a decrease in accuracy.

The introduction of deep learning has significantly improved the effectiveness of facial detection and has become the mainstream approach in contemporary facial detection. It has found widespread applications in various domains.

2.2. Multi-scale Feature Fusion Module

Multi-scale feature fusion is an essential research direction in the field of computer vision, aiming to effectively combine image features from different scales to enhance the performance of image analysis and understanding. With the advancement of deep learning, architectures such as convolutional neural networks (CNN) have taken a dominant role in computer vision tasks. Deep networks can automatically learn multi-scale features from

data, but how to fuse features from different levels remains a research focus. In the realm of computer vision, addressing the issue that CNNs require fixed input image sizes leading to unnecessary accuracy loss, researchers such as Kaiming He et al. introduced the concept of pyramid pooling [27]. By incorporating pyramid pooling layers into CNNs, it becomes possible to perform pooling on features at different scales, thereby achieving multi-scale information fusion. Multi-scale feature fusion holds significant practical value. For example, Qian Wang et al. combined deep CNNs and multi-scale feature fusion to propose a method for detecting multiple classes of 3D objects [28]. This method enables the detection of various objects of interest within a single framework. Another innovation comes from Han et al., who introduced a novel convolutional neural network called MKFF-CNN [29]. This network combines multi-scale kernels with feature fusion and is capable of recognizing gestures, serving the purpose of human-computer interaction. In a similar vein, Chen et al. devised a model named MSF-CNN for multi-scale fusion [30]. This model is employed to train a facial detection system, achieving accurate face detection. Later, Lin et al. integrated the concepts of pyramid structures and multi-scale feature fusion, resulting in the Feature Pyramid Network (FPN) [31]. FPN combines low-level and high-level features to create an object detection system that excels in accuracy, localization, and detection speed. Due to the advantages of FPN in object detection, this paper opts to utilize the FPN module for facial detection when conducting their research.

3. DF-Net Network Design

To enhance the face detection model's inference speed, this paper introduces a lightweight face detection algorithm. By adopting MobileNet-v2 as the foundational framework, the entire network's inference speed is optimized, ultimately enabling real-time face detection and extraction. The algorithms presented herein are executed on a CPU, utilizing test images with a resolution of 1280 × 1240 pixels. Notably, the algorithm achieves an impressive processing speed of 57 fps (frames per second), thereby attaining real-time performance. As depicted in Figure 1, the overarching architecture of the network is depicted. DF-Net predominantly comprises the MobileNet-v2 backbone network, a multi-scale feature cascade module, a spatial pyramid module, and a combined loss function.

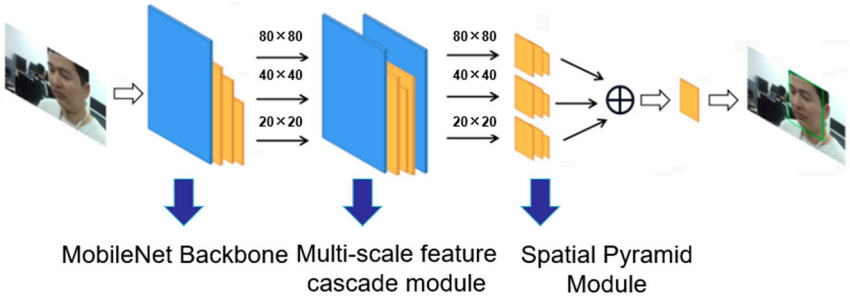

Figure 1. The overall framework of DF-Net.

3.1. Backbone Network

The role of the neural network's backbone network is to extract a sequence of high-dimensional features from the input data. Due to the feature extraction demands of the backbone network and its inherently elevated dimensionality and depth, the processing speed of this network directly influences the overall neural network's performance. To adhere to the real-time requirements of face detection, this paper employs MobileNet-v2 as the backbone network, leveraging the specialized attributes of MobileNet-v2's depth separable convolution to ensure real-time efficacy for the entire face detection algorithm. We chose MobileNetV2 instead of MobileNetV3 for a reason. When conducting research experiments, the main goal is to maintain a small size while achieving real-time performance and deployment on mobile devices. These two advantages are also required in many practical engineering applications. However, MobileNetV3 is generally more complex than MobileNetV2, accompanied by larger model sizes and higher computational costs. Since

the performance requirements of the tasks in this paper are not very high, and considering limited computing resources, we chose to use MobileNetV2 as it provides a better trade-off between speed and model size. Table 1 presents a comparison between the algorithm DF-Net's use of the MobileNet-v2 backbone network and the conventional MobileNet. This paper omits the fully connected layer in the rear of MobileNet. In the table, "Input" represents the input feature map's dimensions, encompassing image height, width, and channels. "Conv" and "Depthwise Conv" denote traditional convolution and depthwise separable convolution, respectively. "C" signifies the number of processing channels for convolution or depthwise separable convolution, while "n" indicates the repetitions in the current layer. "S" represents the stride of convolution or depthwise separable convolution. As the entire backbone network employs depthwise separable convolutions, it attains swift processing speed. Additionally, MobileNet-v2's bottleneck structure is dynamic, allowing its scaling factor to be adjusted as per specific requirements.

Table 1. MobileNet-v2 network structure.

Input	Operator	c	n	s
640 × 640 × 3	Conv3 × 3	16	1	2
320 × 320 × 16	Depthwise Conv3 × 3	32	2	2
160 × 160 × 32	Depthwise Conv3 × 3	64	3	2
(S3)80 × 80 × 64	Depthwise Conv3 × 3	128	3	2
40 × 40 × 128	Depthwise Conv3 × 3	128	3	1
(S2)40 × 40 × 128	Depthwise Conv3 × 3	256	1	2
(S1)20 × 20 × 256	Depthwise Conv3 × 3	256	1	1

3.2. The Multi-Scale Feature Cascade Module

Given the diverse requirements of face detection encompassing varying sizes, positions, and feature attributes, establishing a capacity for multi-scale processing within the algorithm becomes essential. As such, three distinct output feature maps of varying scales are derived from the backbone network and subsequently utilized as inputs, with each scale capturing face information at different magnitudes. This approach concurrently extends the network's receptive field towards faces, thereby enhancing the accuracy of facial information extraction.

In the context of a deep convolutional neural network, as it transitions from one input feature map to the next, irrespective of whether the convolution employs a stride of 1 or 2, the convolutional kernel comprehensively scans the entire feature map. However, this traversal process gives rise to a challenge. During convolution, targets occupying a larger pixel space inherently receive better feature representation than those encompassing fewer pixels. Consequently, the subsequent input feature map tends to emphasize features of more spatially extensive targets. Furthermore, the deep convolutional neural network entails numerous convolution operations, each potentially leading to some degree of information loss, especially for smaller targets. Notably, convolution with a stride of 2 tends to retain pixels from larger targets while inadvertently discarding those from smaller ones. In this context, facilitating multi-scale feature extraction across the feature map stands as a pivotal task for the network itself.

As illustrated in Figure 2, the diagram depicts a multi-scale feature cascade module. To begin with, three distinct scale output feature maps, denoted as FeatureMap1, FeatureMap2, and FeatureMap3, are extracted from the output of the backbone network. Post the high-dimensional feature extraction accomplished by the backbone network, each of the three-scale feature maps holds their individual scale-related information. Specifically, FeatureMap1's resolution is rectified through linear interpolation to align with FeatureMap2, ensuring that their feature information on different scales does not intersect. Following this alignment, FeatureMap1 and FeatureMap2 are channel-wise merged, culminating in a consolidated feature map, subsequently subjected to a 1 × 1 convolution to manage

channel transformation. This convolution is characterized by parameters acquired through network learning, with an identical cascading process for FeatureMap3 and FeatureMap2.

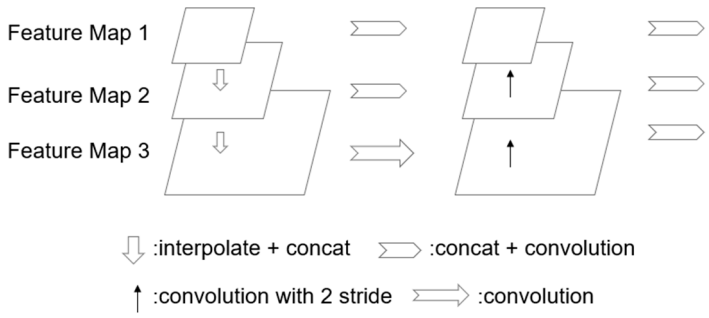

Figure 2. Multi-scale feature cascade module.

In the ensuing steps, FeatureMap3 undergoes a convolution operation, succeeded by a stride-2 convolution. The latter operation is intended to harmonize FeatureMap3's resolution with the cascaded FeatureMap2, thereby enabling a channel-wise fusion devoid of any intermingling of feature information across scales. Following this fusion, another 1×1 convolution is executed to adjust channel numbers. The resultant FeatureMap3 encompasses three distinct scale-specific feature maps. This process mirrors that of FeatureMap1 and FeatureMap2. Consequently, following the traversal of the multi-scale feature cascade module by the three diverse-scale feature maps, the output feature maps collectively encompass diverse scale-associated feature information. Additionally, to cater to the network's imperative reasoning speed, this study has opted to substitute all convolutions within the multi-scale feature cascade module with deep separable convolutions. This strategic substitution translates to reduced computational overhead and enhanced calculation speed.

3.3. The Feature Pyramid Module

The convolution operation in a convolutional neural network involves a weighted summation process between a sliding window and the feature map. Consequently, the dimensions of the convolution kernel dictate the quantity of features the ongoing convolution operation can extract from the feature map. When a 3×3 convolution kernel traverses the feature map, the resultant output feature map contains high-dimensional features achieved through weighted summation of every 3×3 section of the input feature map. Likewise, when performing convolution operations of 5×5 or 7×7, the features in the output feature map represent high-dimensional attributes of the 5×5 or 7×7 segment of the input feature map, constituting the receptive field of the convolution kernel. The ability to extract features within a certain neighborhood size of the feature map is contingent upon the use of convolution kernels of varying sizes, which correspond to distinct receptive fields. In instances where the target within the current feature map is relatively large, the relatively small receptive field derived from the application of diminutive convolution kernels may not adequately encompass the target's characteristics. Conversely, employing large-sized convolution kernels might fall short in encapsulating intricate target details. Faces, for instance, incorporate both minute details such as eyes, nose, and mouth, along with overarching information that relates to the holistic facial structure. Consequently, relying solely on a single-sized convolution kernel for extracting facial features would fail to comprehensively incorporate all pertinent information.

To comprehensively capture target features spanning from intricate details to overarching context, this paper employs the feature spatial pyramid structure depicted in Figure 3. Within this structure, three convolution kernels with distinct receptive fields, namely, 3×3, 5×5, and 7×7, are employed. Given that convolution operations can potentially compromise some original information, the outputs of these three convolutional processes are merged with the initial input feature map in the channel domain. This approach ensures a fusion of feature extraction results from diverse receptive fields. While larger

convolution sizes can expand the receptive field, they also introduce more parameters and computations. Therefore, employing multiple smaller convolutions as replacements can yield equivalent outcomes as larger convolutions but with reduced parameter count. Alternatively, dilated convolutions can be utilized to augment the receptive field without increasing the parameter count.

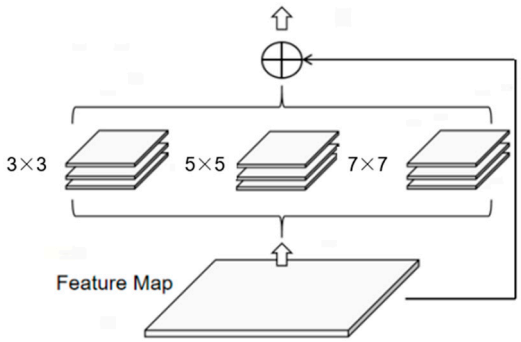

Figure 3. Feature pyramid module.

3.4. Definition of Loss Function

Equation (1) represents the loss function adopted by DF-Net. The essence of this loss function can be segmented into three key components. Firstly, the classification loss is employed to ascertain whether an object is a face. Secondly, the regression loss gauges the accuracy of the face detection frame. Lastly, the face feature point detection regression loss contributes to the precise localization of the face detection frame.

$$\begin{cases} total_{Loss} = Loss_{class} + Loss_{bbox} + a \times Loss_{keypoint} \\ Loss_{class} = CE(p,y) = -(y \times \log(p) + (1-y) \times \log(1-p)) \\ Loss_{bbox} = IoU(A,B) = 1 - \frac{A \cap B}{A \cup B} \\ Loss\_keypoint = Smooth\_L1\_Loss(x) = f(x) = \begin{cases} \frac{1}{2}x^2, & if\ |x| < 1 \\ |x| - \frac{1}{2}, & otherwise \end{cases} \end{cases} \quad (1)$$

The classification loss, denoted as $Loss_{class}$, serves to discern whether an entity is a face or not. 'p' signifies the network's predicted value, while 'y' stands for the true value from the dataset. To accomplish this classification distinction, the two-class cross-entropy loss function is applied. This facilitates the network in discerning the disparities between a face and its surroundings, with the objective of minimizing the cross-entropy loss. Conversely, for forecasting the regression loss of the face detection bounding box, an *IoU* (Intersection over Union) loss function is employed. Here, 'A' symbolizes the face detection box predicted by the network, and 'B' signifies the actual face detection box. The network endeavors to minimize the disparity between the predicted outcome and the actual outcome by reducing the intersection and union ratio between the two bounding boxes. This progressive approach helps the network gradually converge towards the genuine face detection box.

4. Experimental Result and Analysis

4.1. Experimental Environment

The training environment setup for this paper is outlined in Table 2. The computational setup includes an Intel Core i5-11260H CPU, an NVIDIA RTX 3050 GPU, and 32 GB of memory. The algorithm is developed using the Pytorch deep learning framework and implemented using the Python programming language. During subsequent experimental and algorithmic tests, the deployment and execution of the algorithm on the CPU are undertaken.

Table 2. Experimental environment configuration.

Item	Parameter
Operating system	Windows11
CPU	Intel Core i5 11260H
CPU frequency	3.90 GHz
GPU	NVIDIA RTX 3050
Memory	32 GB
Deep learning framework	Pytorch
Programming language	Python

4.2. WiderFace Dataset

The WiderFace dataset is employed in this paper. The inception of the dataset dates back to 2015, originated by the Chinese University of Hong Kong [32]. This dataset holds a more comprehensive and inclusive classification of facial images. With a voluminous compilation of nearly 400,000 instances of facial detection data, it encompasses 61 intricate classifications to capture diverse facial attributes. In this data set, an instance of this diversity is exemplified in the "Scale" category, encapsulating multiple faces within a larger scene. Similarly, the "Occlusion" category solely consists of faces subjected to occlusion circumstances. Expanding beyond facial classification, the WiderFace dataset also encompasses facial feature points. These points consist of five salient facial features—two eye pupils, the nose tip, and two mouth corners. To elaborate on the dataset division, 90% of the data are allocated for training purposes, while the remaining 10% is dedicated to the test set.

Given the dataset's inclusion of facial feature point information, these points can be incorporated into the detected faces, introducing a supplementary constraint to the face detection process. This integration necessitates the inclusion of a quality factor denoted as α, taking values within the range of 0.25, 0.5, 0.75, and 1. This strategic selection of α values prevents excessive interference with the core face detection loss, effectively preserving its primacy. This auxiliary loss framework enforces the constraint and integration of facial feature points within the larger context of the face detection algorithm.

4.3. Network Training

During the training phase, the images in the training set are resized to a uniform size. To preserve the inherent texture and contextual details of the images, grayscale filling is employed. This approach ensures that the image's inherent information remains intact while achieving size uniformity. Training employs the Adam optimizer, and the pre-trained MobileNet-v2 backbone network from ImageNet is used. The learning rate is set to 0.001, with a rate decay mechanism implemented. After every 50 training iterations, the learning rate is reduced by a factor of 10. The training batch size is configured as 5. The DF-Net algorithm in this study undergoes 150 training iterations. Figure 4 illustrates the loss convergence following network training, revealing that network convergence is achieved within approximately 120 iterations.

4.4. Results and Analysis

The DF-Net network described in this paper utilizes both the Multi-Scale Feature Cascade Module and the Feature Pyramid Module. The Multi-Scale Feature Cascade Module allows multiple feature maps of different scales to pass through it, resulting in output feature maps that carry feature information of various scales. This greatly en-riches the semantic information of the feature maps, making it easier to obtain more accurate facial information. The Feature Pyramid enables multi-scale detection, as faces in different images may have different scales. The Feature Pyramid allows the detector to perform face detection at multiple scales. Regardless of the distance of the face or the scale within the image, the detector can recognize faces. This significantly enhances the robustness and accuracy of face detection. To validate the roles of these two modules, we conducted

experiments by removing each module individually and then training and testing on the WiderFace dataset. We compared the detection results with ground truth data and found that removing either module resulted in a decrease in accuracy of approximately 2~3%. Therefore, experimental validation confirms that both the Multi-Scale Feature Cascade Module and the Feature Pyramid Module contribute to improving the accuracy of face detection.

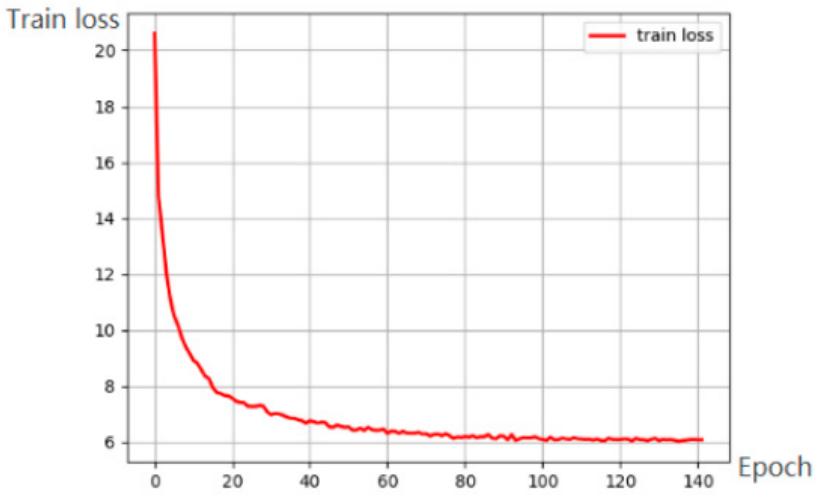

Figure 4. Loss convergence.

To further enhance the model's performance, a backbone network comparison between MobileNet-v1 and MobileNet-v2 is conducted, as depicted in Table 3. In the MobileNet-v2 version, the incorporation of a bottleneck structure enables dynamic adjustment of the channel count transformation ratio, referred to as the bottleneck coefficient. This coefficient is explored at values of 0.25, 0.5, 0.75, and 1. Notably, train and test directly from scratch using the WiderFace dataset. Neither of the two backbones is pre-trained via ImageNet, and identical parameters are maintained. These parameters include a fixed number of training iterations at 50, a learning rate set at 0.001, and consistency in the loss function. The observations from Table 3 indicate that while the bottleneck structure can reduce model size, it entails a channel number transformation that might lead to information loss. Given that the operational speed of the entire network framework in this paper aligns with real-time requirements, a bottleneck coefficient of 1 is adopted in the algorithm.

Table 3. DF-Net comparison of different backbones.

Network Backbone	Easy Accuracy (%)	Medium Accuracy (%)	Hard Accuracy (%)	Model Size (Million Bytes)
MobileNet v1	78.32	73.45	44.64	4.34
MobileNet v2 with 1	82.43	77.21	47.65	4.34
MobileNet v2 with 0.75	79.92	74.88	45.84	4.22
MobileNet v2 with 0.5	79.19	73.25	43.06	4.14
MobileNet v2 with 0.25	79.46	69.28	41.10	3.90

In the formulation of the loss function, this paper introduces a quality factor to the auxiliary loss function, which integrates facial feature point information to constrain facial attributes. As the facial feature points loss function predominantly assumes an auxiliary role, the calibration of the quality factor demands testing. Similar to the bottleneck coefficient, the quality factor is variably set at 0.25, 0.5, 0.75, and 1. As demonstrated in Table 4,

the assessment of DF-Net under distinct quality factors remains constant during experimentation. All other parameters remain fixed, with the bottleneck coefficient set at 0.25 to ensure expedited overall training pace. The outcomes outlined in Table 4 elucidate that a decrease in the quality factor corresponds to an augmented accuracy in network-based facial extraction. This phenomenon is primarily attributed to the dwindling proportion of auxiliary loss from facial feature points, enabling the network to better prioritize the core task of facial detection. Consequently, the diminished influence of facial feature points loss can paradoxically serve as a supplementary constraint on facial detection.

Table 4. DF-Net with different quality factors.

Quality Factor	Easy Accuracy (%)	Medium Accuracy (%)	Hard Accuracy (%)
1	74.97	68.82	41.21
0.75	75.92	69.61	39.06
0.5	76.21	69.18	41.22
0.25	76.73	70.83	42.67

Upon defining the aforementioned parameters, the evaluation of the DF-Net facial detection network primarily revolves around a singular facial classification. Consequently, the evaluation is predicated solely upon the utilization of the Average Precision (AP), a standard gauge within the domain of target detection. As depicted in Table 5, a comprehensive comparison is conducted between DF-Net and other renowned face detection networks such as MTCNN, Faster-RCNN, and RetinaFace. The comparative experiment is conducted under consistent conditions, utilizing an identical dataset for training and maintaining uniform learning rates. The face detection performance of the DF-Net network was evaluated on different levels of complexity within the dataset: easy, medium, and hard patterns. The detected faces were compared against the ground truth labels in the dataset. From the data presented in Table 5, it is evident that the DF-Net achieved an accuracy of 90.15% on easy patterns, 85.63% on medium patterns, and 74.89% on hard patterns. In comparison to the other three methods, our approach significantly improves the accuracy of face detection. Additionally, the processing speed of DF-Net reached 57 fps, with a model size of 4.34 M. This not only ensures real-time performance but also maintains a compact model size, making it well suited for deployment on mobile devices while retaining its real-time capabilities.

Table 5. Comparison of DF-Net with other networks.

Network Model	Processing Speed (fps)	Model Size (M)	Easy Accuracy (%)	Medium Accuracy (%)	Hard Accuracy (%)
DF-Net	57	4.34	90.15	85.63	74.89
MTCNN [20]	63	2.8	61.54	64.76	40.75
Faster-RCNN [33]	12	18.2	71.63	68.31	61.37
RetinaFace [24]	67	1.9	69.62	64.49	58.43

As shown in Figure 5, the results of face detection using the method proposed in this paper include wearing masks, sunglasses, side faces, and the presence of objects on the face. It can be seen that the detection results are all accurate.

Figure 5. The detection results of faces in WiderFace Dataset. The green box represents the detected face.

5. Conclusions

To address the existing challenges of cumbersome deployment and sluggish network inference rates in contemporary face detection systems, this study introduces a face detection algorithm founded on DF-Net. To expedite the overall network inference, MobileNet-v2 is employed as the foundational framework. Additionally, the integration of a multi-scale feature cascade module and a spatial pyramid module facilitates comprehensive multi-scale feature extraction from the feature maps. The algorithm is trained on the publicly available WiderFace dataset for face detection, followed by evaluation on a distinct test set post-training. This research extensively scrutinizes the network model's dimensions, processing velocity, and extraction precision. A comparison with three other classic face detection networks reveals a significant improvement in face detection accuracy with DF-Net. Furthermore, it conducts a meticulous exploration of each network hyperparameter through experimental analysis, affirming the efficacy of the proposed algorithm. Ultimately, these endeavors culminate in the achievement of rapid and accurate face detection. DF-Net offers real-time performance without compromising on a compact model size, making it suitable for deployment on mobile devices. These two advantages align well with practical engineering applications. DF-Net can be applied in scenarios such as pedestrian detection in autonomous driving and facial payment in mobile transactions. These scenarios often require face detection on platforms with limited memory and computing capabilities, demanding low-latency and real-time responsiveness. Hence, this method holds substantial application potential. After effectively extracting facial regions, we will further analyze facial features, which is our future research content.

Author Contributions: Conceptualization, X.L. and X.P.; methodology, Q.T. and Y.C.; writing—original draft preparation, Q.T., Y.L. and Y.C.; writing—review and editing, X.L. and X.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (62275173, 62061136005), the Sino-German Cooperation Group (GZ1391, M-0044), and the Shenzhen Research Program (JSGG20210802154541021, JCYJ20220531101204010).

Data Availability Statement: Data underlying the results presented in this paper are not publicly available at this time but may be obtained from the authors upon reasonable request.

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

References

1. Sun, Y.; Dong, J.; Jian, M. Fast 3D face reconstruction based on uncalibrated photometric stereo. *Multimed. Tools Appl.* **2015**, *74*, 3635–3650. [CrossRef]
2. Roth, J.; Tong, Y.; Liu, X. Adaptive 3D Face Reconstruction from Unconstrained Photo Collections. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 2127–2141.
3. Yang, Y.; Kan, L.; Yu, J. 3D face reconstruction method based on laser scanning. *Infrared Laser Eng.* **2014**, *43*, 3946–3950.
4. Fan, X.; Zhou, C.; Wang, S. 3D human face reconstruction based on band-limited binary patterns. *Chin. Opt. Lett.* **2016**, *14*, 81101. [CrossRef]
5. Sun, N.; Zhou, C.; Zhao, L. A Survey of Face Detection. *J. Circuits Syst.* **2006**, *6*, 101–108.
6. Li, Y.; Xi, Z. Based on the improved RetinaFace face detection method. *Appl. Sci. Technol.* **2023**, *9*, 1–7.
7. Paul, V.; Michael, J. Robust real-time face detection. *Int. J. Comput. Vis.* **2004**, *57*, 137–154.
8. Hinton, G.; Deng, L.; Yu, D. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Process. Mag.* **2012**, *29*, 82–97. [CrossRef]
9. Jones, M.; Viola, P. Fast multi-view face detection. *Mitsubishi Electr. Res. Lab.* **2003**, *96*, 3–14.
10. Mathias, M.; Benenson, R.; Pedersoli, M.; Van Gool, L. Face detection without bells and whistles. In Proceedings of the ECCV Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 720–735.
11. Yan, J.; Lei, Z.; Wen, L. The fastest deformable part model for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 2497–2504.
12. Zhu, X.; Ramanan, D. Face detection, pose estimation, and land mark localization in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 2879–2886.
13. Krizhevsky, A.; Sutskever, I.; Geoffrey, E.H. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *6*, 84–90. [CrossRef]
14. Gómez-Guzmán, M.A.; Jiménez-Beristaín, L.; García-Guerrero, E.E.; López-Bonilla, O.R.; Tamayo-Perez, U.J.; Esqueda-Elizondo, J.J.; Palomino-Vizcaino, K.; Inzunza-González, E. Classifying Brain Tumors on Magnetic Resonance Imaging by Using Convolutional Neural Networks. *Electronics* **2023**, *12*, 955. [CrossRef]
15. Li, Z.; Tang, X.; Han, J. PyramidBox++ : High performance detector for finding tiny face. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
16. Lei, H.L.; Zhang, B.H. Crowd count algorithm based on multi-model deep convolution network integration. *Laser Technol.* **2019**, *43*, 476–481.
17. Chen, Q.X.; Wu, W.C.; Askar, H. Detection algorithm based on multi-scale spotted target modeling. *Laser Technol.* **2020**, *44*, 520–524.
18. Li, H.; Lin, Z.; Shen, X.; Brandt, J.; Hua, G. A convolutional neural network cascade for face detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 5325–5334.
19. Li, Y.; Lv, X.; Gu, Y. Face detection algorithm based on improved S3FD network. *Laser Technol.* **2021**, *45*, 722–728.
20. Zhan, 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]
21. Wang, H.; Li, Z.; Ji, X.; Wang, Y. Face R-CNN. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
22. Li, C.; Li, L.; Jiang, H. YOLOv6: A single-stage object detection framework for industrial applications. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR), New Orleans, LA, USA, 18–24 June 2022.
23. Wang, C.; Bochkovskiy, A.; Mark Liao, H. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
24. Deng, J.; Guo, J.; Ververas, E.; Kotsia, I.; Zafeiriou, S. RetinaFace: Single-Shot Multi-Level Face Localisation in the Wild. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 5202–5211.
25. Li, J. DSFD: Dual Shot Face Detector. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 5055–5064.
26. Qi, D.; Tan, W.; Yao, Q. YOLO5Face: Why reinventing a face detector. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Tel Aviv, Israel, 13–27 October 2022; pp. 228–244.
27. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef] [PubMed]
28. Wang, Q.; Bhowmik, N.; Breckon, T.P. Multi-Class 3D Object Detection within Volumetric 3D Computed Tomography Baggage Security Screening Imagery. In Proceedings of the IEEE International Conference on Machine Learning and Applications (ICMLA), Miami, FL, USA, 14–17 December 2020; pp. 13–18.

29. Han, L.; Zou, Y.; Cheng, L. A Convolutional Neural Network with Multi-scale Kernel and Feature Fusion for sEMG-based Gesture Recognition. In Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO), Sanya, China, 27–31 December 2021; pp. 774–779.
30. Chen, Q.; Meng, X.; Li, W.; Fu, X.; Deng, X.; Wang, J. A multi-scale fusion convolutional neural network for face detection. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC), Banff, AB, Canada, 5–8 October 2017; pp. 1013–1018.
31. Lin, T.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.
32. Wu, W.; Qian, C.; Yang, S. Look at Boundary: A Boundary-Aware face alignment algorithm. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 2129–2138.
33. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *28*, 1137–1149. [CrossRef] [PubMed]