electronics

Article

FAFD: Fast and Accurate Face Detector

Namho Kim, Jun-Hwa Kim and Chee Sun Won *

Department of Electrical and Electronic Engineering, Dongguk University-Seoul, 30, Pildong-ro 1-gil, Jung-gu, Seoul 04620, Korea; namho96@dgu.ac.kr (N.K.); jhkim414@dongguk.edu (J.-H.K.)
* Correspondence: cswon@dongguk.edu

Abstract: Deep Neural Networks (DNN) have contributed a significant performance improvement in face detection. However, since most models focus only on the improvement of detection accuracy with computationally expensive structures, it is still far from real-time applications with a fast face detector. The goal of this paper is to improve face detection performance from the speed-focusing point of view. To this end, we propose a novel Fast and Accurate Face Detector (FAFD) to achieve high performance on both speed and accuracy performance. Specifically, based on the YOLOv5 model, we add one prediction head to increase the detection performance, especially for small faces. In addition, to increase the detection performance of multi-scale faces, we propose to add a novel Multi-Scale Image Fusion (MSIF) layer to the backbone network. We also propose an improved Copy-Paste to augment the training images with face objects in various scales. Experimental results on the WiderFace dataset show that the proposed FAFD achieves the best performance among the existing methods in a Speed-Focusing group. On three sub-datasets of WiderFace (i.e., Easy, Medium, and Hard sub-datasets), our FAFD yields average precisions (AP) of 95.0%, 93.5%, and 87.0%, respectively. Also, the speed performance of the FAFD is fast enough to be included in the group of speed-focusing methods.

Keywords: face detection; convolution neural network; YOLOv5; image augmentation

Citation: Kim, N.; Kim, J.-H.; Won, C.S. FAFD: Fast and Accurate Face Detector. *Electronics* **2022**, *11*, 875. https://doi.org/10.3390/electronics11060875

Academic Editor: Savvas A. Chatzichristofis

Received: 10 February 2022
Accepted: 8 March 2022
Published: 10 March 2022

1. Introduction

Face detection is an important first step in face image analysis problems such as face alignment [1,2], face recognition [3,4], and face attribute analysis [5,6]. Applying face detection to real-world images is a challenging task due to the diverse characteristics of face images in scale variations, occlusions, illuminations, poses, and expressions [7]. Based on the recent development of Convolutional Neural Networks (CNN), various approaches [8–17] have been proposed to improve the face detection performance. In [8–10], they reported performance improvement in a multi-task manner by combining the facial landmark localization with the existing object detector. Another group of methods mainly focused on improving the architecture of CNN-based networks [11–16]. For example, a method to automatically find an effective network structure based on NAS (Neural Architecture Search) was proposed in [11]. In RefineFace [12], they adopted its own modules such as Selective Two-step Regression (STR), Selective Two-step Classification (STC), Scale-aware Margin Loss (SML), Feature Supervision Module (FSM), and Receptive Field Enhancement (RFE). The context module to a specific pyramid was used to increase the receptive field in Euclidean grids [13,14]. An additional layer such as a large size filter [15] for each prediction head was added to combine context information. An anchor-associated layer [16] to the C3 stage based on the Single Shot MultiBox Detector (SSD) [18] was also used for an anchor matching strategy to increase recall performance for small faces. Besides, loss functions [11,12,17] and data augmentation [13,14] suitable for face detection have been used.

However, the studies mentioned above have focused mainly on improving the detection performance rather than the speed performance. Therefore, they relied heavily on applying computationally expensive methods with large-scale images. However, since

there are increasing demands on face detection applications in mobile environments with low computational resources equipped with CPU and edge-computing devices [19], it is important to maintain a trade-off between accuracy and speed. To this end, based on the YOLOv5 object detector [20], we propose a novel Fast and Accurate Face Detector (FAFD). Our FAFD achieves a high performance both on speed and accuracy for VGA (Video Graphics Array, 640 × 480) resolution images.

The main contributions of this paper are summarized as follows:

1. We propose an additional module for the YOLOv5 to improve the detection performance for small faces. The prediction head is designed to receive a low level (thus, a high-resolution) feature map.
2. To boost the multi-scale face detection performance, a Multi-Scale Image Fusion (MSIF) layer is proposed to effectively learn multi-scale faces.
3. To diversify face objects with various scales in training the DNN, a Copy-Paste technique for the face object is proposed.
4. Within the category of the speed-focusing face detectors, the proposed method has achieved state-of-the-art (SOTA) performance. Specifically, the proposed FAFD-Small achieved 95.0%, 93.5%, and 87.0% APs in Easy, Medium, and Hard sub-datasets of WiderFace [7] dataset, respectively. Also, our FAFD-Nano model achieved 93.0%, 91.1%, and 84.1% APs in Easy, Medium, and Hard sub-datasets, respectively. These results are the best ones among the speed-focusing methods.

2. Related Work

According to the guideline proposed by [21], we divide the existing face detection techniques into two groups: Accuracy-Focusing Face Detection and Speed-Focusing Face Detection. The criterion of this categorization is mainly based on the floating-point operations (FLOPs) of the model. That is, the Accuracy-Focusing methods have a computational magnitude of about TFLOPs, whereas the Speed-Focusing ones have a computational magnitude of about GFLOPs or 10 GFLOPs.

2.1. Accuracy-Focusing Face Detection

Most existing methods [9,10,17,22–24] belong to the Accuracy-Focusing group, which tries to increase accuracy in average precisions (AP) with high-performance graphic cards. For example, RetinaFace [9] exploits manually annotated five face landmarks information based on RetinaNet [25]. Also, for better localization, Cascade regression [26] is applied with multi-task loss in RetinaFace. YOLO5-Face [10] and Mask-Face [22], which are based on YOLOv5 [20] and MaskR-CNN [27], respectively, also adopted five landmark information as RetinaFace [9]. Based on S^3FD [16], DSFD [17] proposed three key contributions such as Feature Enhance Module (FEM), Progressive Anchor Loss (PAL), and Improved Anchor Matching (IAM). For these modules, the detection performance of DSFD is improved by enhancing the feature maps and providing proper initialization for the regression. To obtain an efficient and accurate face detector, SRN [23] applied focal loss [25] to multi-step detection [28]. Unlike the previous methods, CSP [24] applies an anchor-free detection head and obtains the advantages of both global and local feature maps using multi-scale features extracted from modified ResNet-50 [29].

2.2. Speed-Focusing Face Detection

Although the accuracy-focusing face detectors mentioned in the previous section can achieve a high detection performance, in practice, they are infeasible for most mobile applications due to computationally expensive models and large-scale images. To solve this problem, some modifications [30–34] have been made to achieve high speed while maintaining detection performance as much as possible. Based on SSD [18], Faceboxes [30] designed an efficient face detection backbone by applying the multiple scale convolutional layers (MSCL) and the rapidly digested convolutional layers (RDCL). Also, YuFaceDetectNet [31] and ULFG [32] improved the detection performance by adding more convolutional

layers to each stride. Inspired by FaceBoxes [30], FaceBoxesImproved (FBI) [33] adopted a knowledge distillation method to boost the performance of a lightweight face detector. LFFD [34] considered RF (Receptive Fields) as another way of defining anchors as natural anchors that can cover continuous facial scales.

Note that the upper mentioned methods in the Speed-Focusing group do not consider the multi-scale faces. In this paper, to improve the detection performance as a Speed-Focusing method, we propose a scheme of Multi-Scale Image Fusion (MSIF) to treat multi-scale faces. Our MSIF is based on the standard Path Aggregation Network (PA-Net) [35], where multi-scale images are merged into a single feature map to provide richer information on multi-scale faces at a marginal extra cost.

3. Proposed Methods

3.1. Overall Structure of the Proposed Framework

Our FAFD is based on the YOLOv5 network structure. The overall structure of the FAFD is shown in Figure 1. The backbone of FAFD uses the CSPDarknet53 structure with SPPF (Spatial Pyramid Pooling Fast) module [20] and the Neck adopts the PA-Net [35]. Since the original YOLOv5 has low detection performance for small faces, one prediction head is added to the FAFD to improve the performance on small faces. In addition, the first convolution layer of the backbone was replaced with a Multi-Scale Image Fusion (MSIF) layer to improve multi-scale face detection.

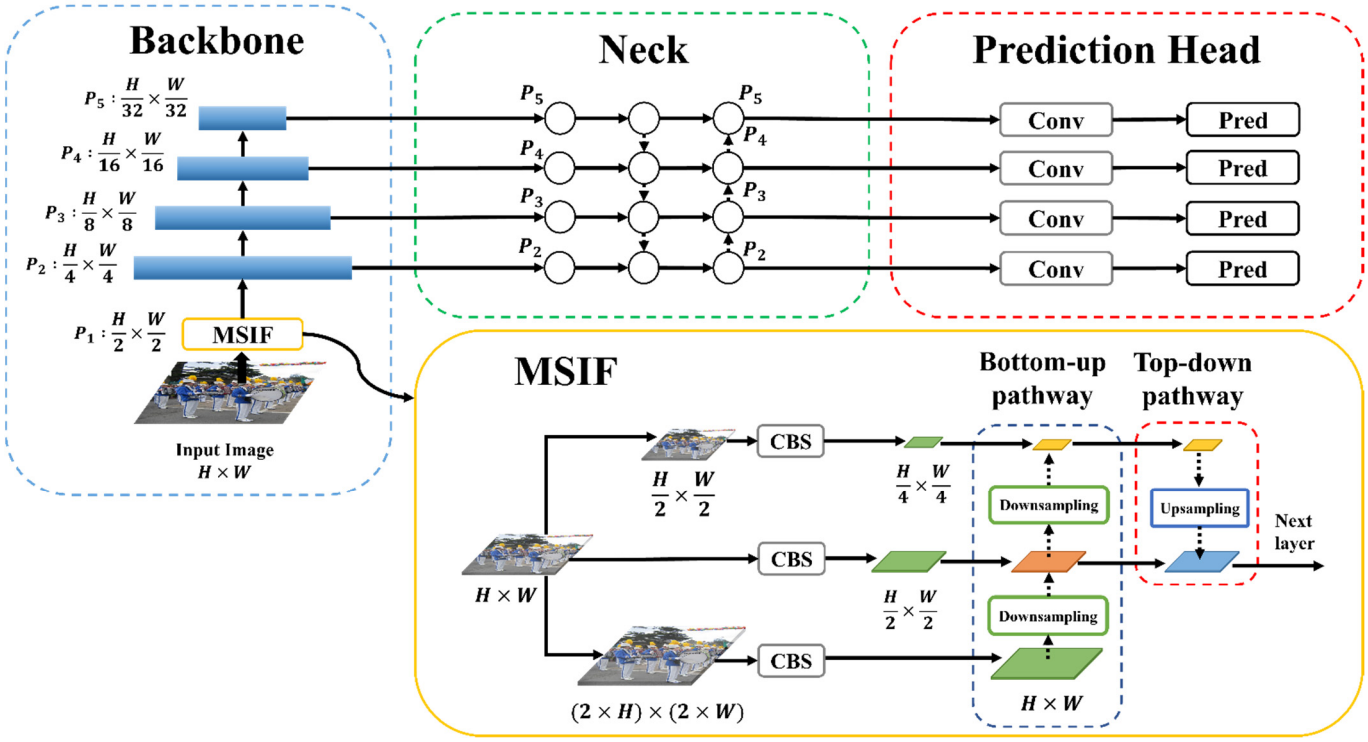

Figure 1. The overall structure of the proposed FAFD, where H and W are the height and width of the input image, respectively.

3.2. Additional Prediction Head for Small Faces

Since the high-resolution feature map at the bottom of the feature pyramid contains weak semantic information but strong spatial local features, it is useful to improve the detection performance for small faces [36]. In the original YOLOv5 structure, feature maps of P3, P4, and P5 layers in Figure 1 corresponding to {1/8, 1/16, 1/32} of the input image size are used for the prediction. In this paper, as shown in Figure 1, the feature map of the P2 layer corresponding to {1/4} of the input image size is added. To utilize the feature map of the P2 layer, one prediction head is also used, and the neck structure is changed

according to the additional head. This additional prediction head allows the model to make a more robust prediction on the small faces.

3.3. Multi-Scale Image Fusion

We propose to include a Multi-Scale Image Fusion (MSIF) layer to the basic YOLOv5. The goal of the MSIF is to provide multi-scale faces at the input level of the network. That is, as shown in Figure 1, the MSIF extracts the pixel level features for three different scales of the input image. The feature maps of the three scales are fused into one feature map through bottom-up and top-down structures. Then, the combined feature map is transferred to the next layer of the YOLOv5. The pipeline of the proposed method is as follows:

1. Resize a $W \times H$ input image to form an image pyramid with the size of $\{(W/2) \times (H/2),\ W \times H,\ (2 \times W) \times (2 \times H)\}$.
2. Feature maps are extracted by applying the CBS (Convolution + Batch Normalization + SiLU) layer as shown in Figure 2a to the three resized images in (i). In the figure, k, s, and p represent kernel size, stride, and padding, respectively. The size of the extracted feature map becomes half of each image pyramid.

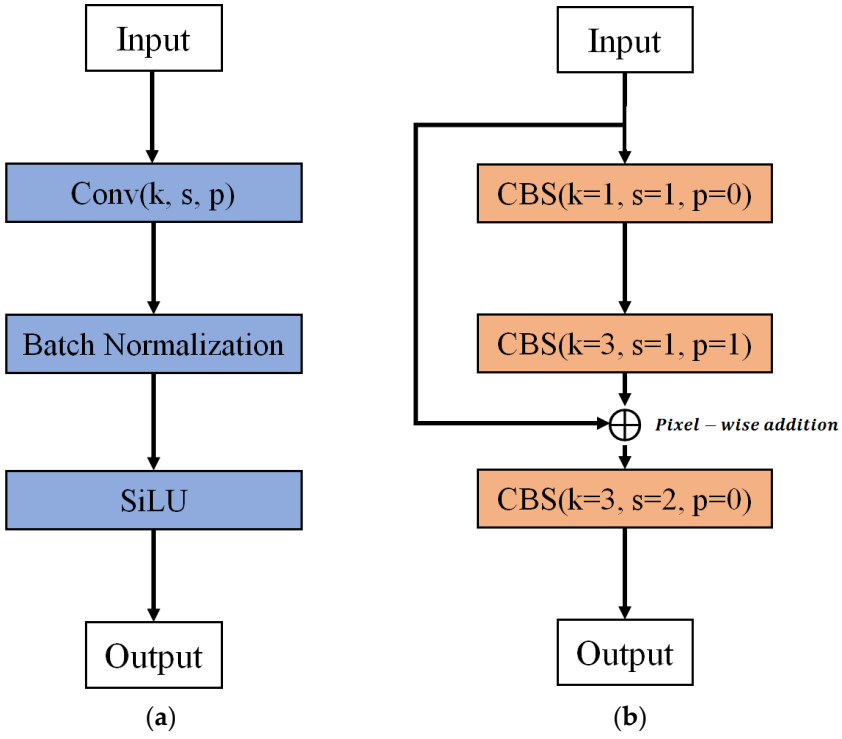

Figure 2. Sub-layers of (**a**) CBS layer and (**b**) Down-sampling layer.

3. For the extracted hierarchical feature map in (ii), the down-sampling layer of Figure 2b is applied to reduce the size. We need two down-sampling processes to reach the lowest-resolution map (see Figure 1).
4. The feature map of (iii) is up-sampled by a bi-linear interpolation. Then, before passing it to the next layer, a pixel-wise addition is performed with a medium-level feature map in (iii).

3.4. Copy-Paste Augmentation for Face

For training our FAFD, a large number of images are required to overcome overfitting and generalization problems [37]. We need annotated training images with the location and the size information of the bounding boxes with the faces. Therefore, it is important to provide annotated face images with different locations in the entire image space and different face scales as much as possible. To this end, the Copy-Paste method [38–40] can be

adopted to diversify the face locations in the image space and the face scales. For example, the copied faces are randomly scaled [38] and pasted in various background images [39]. Note that the Copy-Paste augmentation can be also used to alleviate the class-imbalance problem by making more copies from the minority classes (e.g., small faces) [40]. To the best of our knowledge, no study that adopts Copy-Paste method to face detection has been reported.

Unlike other application domains, in face detection, surrounding areas of the face region such as the neck and shoulders also provide useful information [13]. However, the bounding boxes of the ground truth provide the location and the size of the facial region only. For example, using the source images in Figure 3a with the face bounding boxes and the destination image in Figure 3b, we have the pasted faces with the bounding boxes as in Figure 3c. On the other hand, in this paper, we propose a modified Copy-Paste method that can include the shoulder-face region as well as the face. Then, we have the pasted image as in Figure 3d.

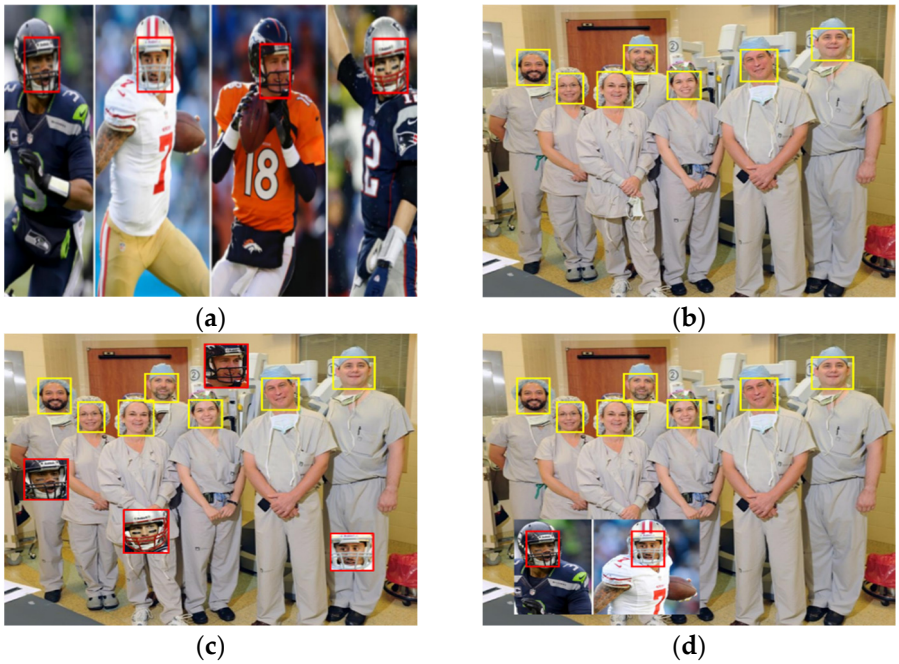

Figure 3. Copy-Paste methods for face images: (**a**) source (reference) image, (**b**) target (original) image, (**c**) pasted image with bounding box only, and (**d**) pasted image with the shoulder and neck as well as face.

On top of the augmentation techniques of the basic YOLOv5 such as geometric transformation, mosaic [41], and mixup [42], our Copy-Paste method is used as shown in Figure 4. In Figure 4, f_s, S_c, and S_r represent the length of the shorter side of the image, crop-scale factor, and resize-scale factor, respectively. The detailed process of the proposed Copy-Paste is as follows:

(i) The reference image I_R for the original image I_O, is chosen at random from the training dataset.
(ii) Get f_s from I_R.
(iii) S_c is randomly determined within the experimentally defined range of [0.1, 0.4].
(iv) I_C is obtained by randomly cropping an area of size ($f_s \times S_c$, $f_s \times S_c$) from the reference image I_R.
(v) Using the bounding box information of the face objects in the reference image, check if the cropped region contains any face object. Specifically, if the cropped I_C contains a center point of the bounding box for the face, proceed to (vi). Otherwise, return to (iii) for re-cropping.

(vi) In this step, we need to check if the cropped I_C includes any other body parts such as the shoulder as well as the face. To this end, we employ a simple method of comparing the sizes of the face bounding boxes in I_C with the size of I_C. So, if widths and heights of all bounding boxes in I_C are less than 60% than those of I_C, then we proceed to (vii). Otherwise, we assume that I_C consists mainly of the facial region and we return (iii) for re-cropping. Note that the thresholding value of 60% is determined experimentally.
(vii) S_r is randomly determined in the experimentally defined range of [0.5, 1.5].
(viii) Resize I_C by the scale factor S_r.
(ix) Paste I_C into I_O in the non-overlapping region with the existing face region.

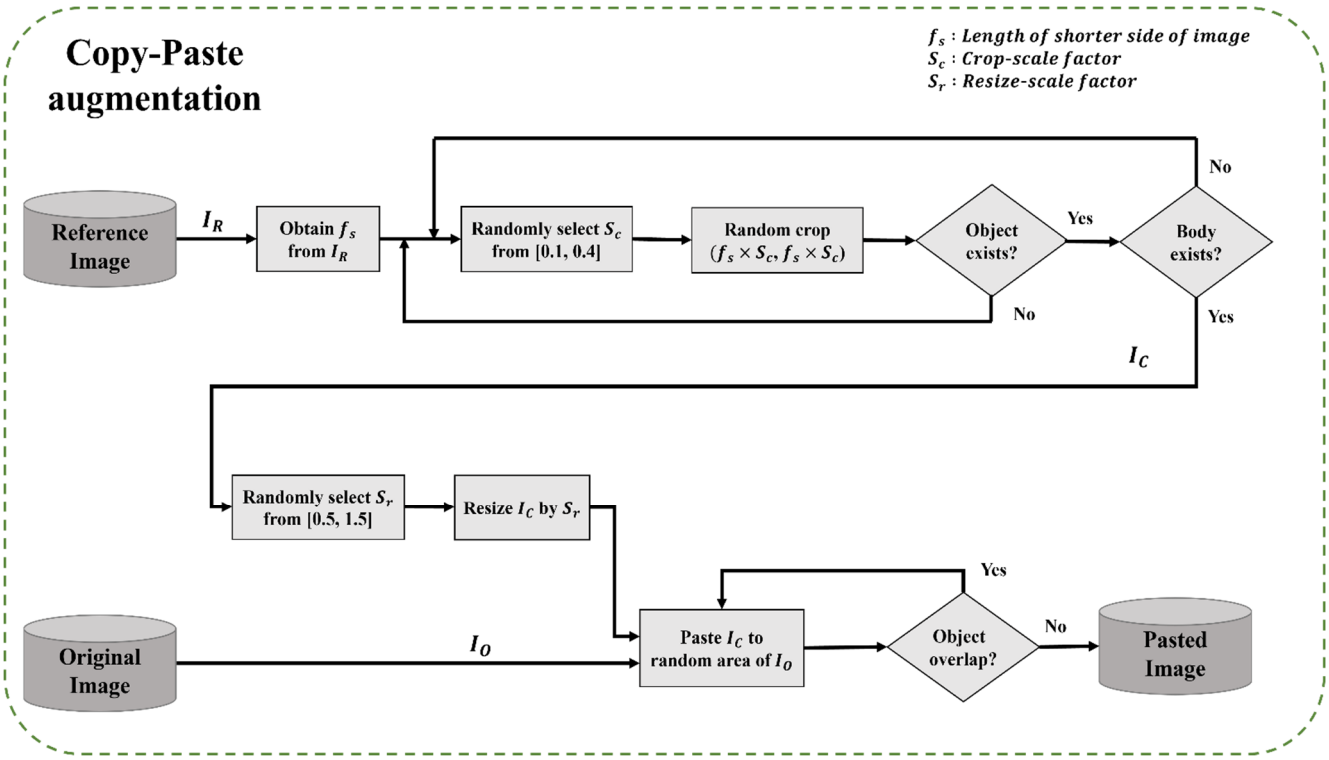

Figure 4. The pipeline of our Copy-Paste algorithm.

4. Results

4.1. Dataset

We use WiderFace dataset [7] to train and validate our proposed FAFD. The WiderFace dataset has 393,703 face objects in 32,203 images, including various situations in terms of pose, scale, expression, occlusion, and lighting condition. Images in the WiderFace dataset are grouped into 61 social event classes that are much more diverse and closer to real-world scenarios.

The WiderFace dataset is split into train (40%), validation (10%), and test (50%) sub-datasets. Each split sub-dataset has three levels of difficulty: Easy, Medium, and Hard according to the detection rate of EdgeBox [43]. Compared to other sub-datasets, the Hard sub-dataset contains a large number of tiny faces that are between 10–50 pixels tall. So, it is quite challenging for the WiderFace dataset to achieve high detection performance on a hard sub-dataset.

4.2. Experiments

We implemented the proposed FAFD in Pytorch 1.10.1 environment with NVIDIA RTX 6000 GPU. YOLOv5 consists of five models: YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x [44]. Sharing the basic module, these five models have different depths and sizes. The depth of the layers is determined by the number of repetitions of the

bottleneck layer and the size is determined by the channel size of the entire layers. In this paper, although our methods introduced in Section 3 can be used for any type of YOLOv5, considering the trade-off between speed and accuracy performance, we chose YOLOv5s and YOLOv5n models for our experiments. Therefore, there are two models of our proposed FAFD, YOLOv5n-based FAFD-Nano, and YOLOv5s-based FAFD-Small. The parameter values of k (kernel size), s (stride), and p (padding) of the CBS layer in the MSIF were set to 6, 2, and 2, respectively.

The hyperparameters used in our experiments for training with the WiderFace were as follows: the SGD optimizer with 200 epochs, a weight de-cay of 0.0005, and momentum of 0.937. The first 3 epochs were used for a warm-up. During the warmup period, the momentum of SGD was set to 0.8, and the learning rate was updated from 0.1 to the initial learning rate of 0.01 through one-dimensional linear interpolation. After the warmup, the learning rate was reduced from the initial learning rate of 0.01 to the minimum learning rate of 0.01×0.1 through the cosine annealing function. Finally, since 3 anchor boxes were allocated per each layer of the prediction head, 9 and 12 anchor boxes were used for the baseline YOLOv5 and the proposed FAFD, respectively (see Table 1). In the table, the size of each anchor box was obtained by using the k-means++ [45] algorithm on the distribution of training data.

Table 1. Anchor boxes (In (w,h) 'w' and 'h' represent the width and the height of the anchor box, respectively).

Used Layer	Size of Anchor Box		
Baseline			
First detection layer	(72, 94)	(130, 170)	(229, 304)
Second detection layer	(16, 21)	(26, 33)	(43, 55)
Third detection layer	(4, 5)	(7, 9)	(11, 14)
Proposed FAFD			
First detection layer	(117, 152)	(167, 218)	(249, 340)
Second detection layer	(32, 40)	(46, 63)	(69, 97)
Third detection layer	(12, 14)	(16, 21)	(22, 29)
Fourth detection layer	(4, 5)	(6, 7)	(8, 10)

For the testing phase with VGA-resolution images, the speed performance in terms of latency and the accuracy performance of AP were used for the comparisons. To maintain the aspect ratio of the original image, we first set the length of the longer side to 640 and resized the length of the shorter one proportionally. In the non-maximum suppression (NMS) stage, the IoU threshold and the confidence threshold were set to 0.5 and 0.02, respectively, as in the YOLO5-Face [10].

4.3. Ablation Study

To verify the effectiveness of each component in the proposed method, an ablation study was conducted. The validation dataset of WiderFace was used for model evaluation. As shown in Tables 2 and 3, AP for each sub-dataset calculated with the evaluation toolkit [7] was used as a performance metric.

Table 2. Ablation study for the proposed methods with WiderFace validation dataset.

Method				WiderFace Validation Set		
Baseline	**MSIF**	**Copy-Paste**	**P2**	$\mathbf{AP_{Easy}}$	$\mathbf{AP_{Medium}}$	$\mathbf{AP_{Hard}}$
✓				94.9	93.1	84.2
✓	✓			95.0	93.4	85.5
✓		✓		95.0	93.3	84.2
✓			✓	94.9	93.2	85.7
✓	✓		✓	94.8	93.3	86.9
✓	✓	✓	✓	**95.0**	**93.5**	**87.0**

Table 3. Ablation study for the proposed P2 and MSIF methods with respect to face sizes. AP_{Small}, AP_{Medium}, and AP_{Large} represent the AP performance for Small, Medium, and Large face sub-datasets. AP_{All} is the AP performance for all faces in WiderFace Validation dataset without the size-based categorization.

Method			WiderFace Validation Set			
Baseline	**MSIF**	**P2**	$\mathbf{AP_{Small}}$	$\mathbf{AP_{Medium}}$	$\mathbf{AP_{Large}}$	$\mathbf{AP_{All}}$
✓			78.4	94.7	94.9	83.3
✓	✓		80.4	94.8	95.0	84.8
✓		✓	80.5	94.7	92.8	85.1
✓	✓	✓	82.3	94.6	92.1	86.4

As shown in Table 2, except for a few cases, the individual or combination of the proposed schemes has achieved better AP than the baseline YOLOv5s model. Especially, the proposed Multi-Scale Image Fusion layer (MSIF) has achieved the largest performance gain among the three proposed schemes of P2, Copy-Paste, and MSIF. Specifically, adopting MSIF alone, the AP performance has improved by 1.3% in Hard sub-dataset and by 0.1% and 0.3% in Easy and Medium sub-datasets, respectively, compared to the baseline YOLOv5s. For the Copy-Paste, the AP performance in the Hard sub-dataset is the same as the baseline YOLOv5s. This is because the original YOLOv5 has low detection performance for small faces, so it cannot learn the features of the pasted object well. As shown in the last two rows of Table 2, when MSIF and P2 were applied to increase the detection performance of small faces, the performance of Copy-Paste was increased in all sub-datasets. Finally, with the P2 alone, the AP performance was the same in the Easy sub-dataset, but the AP performance increased significantly to 1.5% in the Hard sub-dataset. This means that the high-resolution feature map of the P2 layer is effective for detecting small-scale faces, but on the contrary, it has no effect on large-scale faces. Note that it is quite a challenging task to improve the detection performance of the Hard sub-dataset.

On top of the performance gain for each individual technique, we have achieved additional improvements in all combinations of the proposed schemes. Among all combinations, when all the three proposed schemes of the additional prediction head (P2), Copy-Paste, and MSIF are included, we have the highest AP performance.

In order to show the effectiveness of the proposed schemes of P2 and MSIF for small faces, we first need to categorize the validation dataset of the WiderFace with respect to the size of the face. To this end, borrowing the size criterion from [7], we divide the faces in the validation dataset of the WiderFace into three size groups: Small (between 10–50 pixels of the height), Medium (between 50–300 pixels of the height), and Large (over 300 pixels of height). Among the three size groups, the Small sub-dataset takes 78.3% of the WiderFace validation dataset. Now, we conduct the ablation study for the proposed P2 and MSIF schemes. As shown in Table 3, each of the proposed MSIF and P2 achieved better AP

than the baseline YOLOv5s model for the Small sub-dataset. Specifically, each of P2 and MSIF has contributed to the performance improvement for Small sub-dataset by about 2%. This proves that the proposed P2 and MSIF are effective, especially for small faces. On the other hand, having the local feature by P2 to the feature fusion pyramid, the detection performance for the Large sub-dataset may deteriorate.

4.4. Comparative Study

Our FAFD model was compared with some speed-focusing methods [30–32,34] reported in [21]. The results include AP comparisons for Easy, Medium, and Hard sub-datasets. Also, the speed performance was compared in terms of latency for VGA-resolution images. For a fair comparison, we export our FAFD-small and FAFD-Nano models from Pytorch to ONNX and inference using ONNXTUNTIME as [21].

As shown in Table 4, our FAFD-Small obtained 95.0%, 93.5%, and 87.0% AP in Easy, Medium, and Hard sub-datasets, respectively, for the WiderFace dataset. Here, our FAFD-Small results for all sub-datasets show the best performance by the increased detection performance of 4.0%, 5.5%, and 9.2% compared to LFFD-v1 [34]. Also, our FAFD-Small shows a better speed performance of 207.1 ms than LFFD-v1 of 239.43 ms. Although FAFD-Nano has lower performance than FAFD-Small, it achieved 93.0%, 91.1%, and 84.1% AP in Easy, Medium, and Hard sub-datasets, respectively. This is the second-best performance with performance gains of 2.0%, 3.1%, and 6.3% to the LFFD-v1. In addition, our FAFD-Nano can run at 94.4 ms.

Table 4. Comparison of our FAFD and the Speed-Focusing face detectors on the WiderFace validation dataset. AVG GFLOPs and AVG Latency are the average GFLOPs and the average Latency, respectively, for 100 images with different sizes.

Model	#Params ($\times 10^6$)	AVG GFLOPs	WiderFace Validation Set			AVG Latency (ms)	
			AP_{Easy}	AP_{Medium}	AP_{Hard}	AVG Latency (ms)	Device
FaceBoxes [30]	1.013	**1.5**	84.5	77.7	40.4	23.68	INTEL i7-5930K
YuFaceDetectNet [31]	**0.085**	2.5	85.6	84.2	72.7	56.28	
ULFG-slim-320 [32]	0.390	2.0	65.2	64.6	52.0	**21.4**	
ULFG-slim-640 [32]			81.0	79.4	63.0		
ULFG-RFB-320 [32]	0.401	2.4	68.3	67.8	57.1	23.17	
ULFG-RFB-640 [32]			81.6	80.2	66.3		
LFFD-v2 [34]	1.520	37.8	87.5	86.3	75.2	185.17	
LFFD-v1 [34]	2.282	55.6	91.0	88.0	77.8	239.43	
FAFD-Nano (Ours)	1.808	8.3	93.0	91.1	84.1	94.4	INTEL i7-7700K
FAFD-Small (Ours)	7.193	29.3	**95.0**	**93.5**	**87.0**	207.1	

We note that the latency performance in Table 4 is measured with two different CPUs (i.e., i7-5930K and i7-7700K). Therefore, for a fair comparison, we also need to check other computing measures such as GFLOPs and the number of model parameters (see Table 4). As shown in the table, the proposed FAFD-Nano and FAFD-Small have smaller GFLOPs than LFFD-v1 and LFFD-v2. In terms of the number of parameters, however, our FAFD-Small has much more parameters than LFFD-v1 and LFFD-v2. Although the number of parameters of FAFD-Small is large, its computing speed in terms of the GFLOPs is not that bad. Recall that, in [21], the criterion for differentiating the Speed-Focusing Face Detector from the Accuracy-Focusing Face Detector is the GFLOPs. Here, both FAFD-Nano and

FAFD-Small still satisfy the GFLOPs criterion for the Speed-Focusing Face Detector, having a computational magnitude of tens of GFLOPs.

Experimental results show that the proposed FAFD achieved the best results in terms of accuracy–speed performance among the speed-focusing methods.

5. Conclusions

For a high-speed face detector with CPU, we have proposed a Fast and Accurate Face Detector (FAFD) based on YOLOv5. In our FAFD, we have designed to effectively detect a wide range of faces. To this end, we have proposed a Multi-Scale Image Fusion (MSIF) layer for the baseline YOLOv5. It is expected that the proposed MSIF can be adopted for other domain-specific datasets as a module to provide multiple-scale images at the input side of the network. To improve the small-scale face detection performance, we have inserted one more prediction head for the high-resolution feature map into YOLOv5. As an augmentation technique, a Copy-Paste technique is employed for face objects, which can provide diversified face images during the training with various scales and locations. The Copy-Paste technique can also alleviate the class-imbalance problem for the minority class of the small faces. Experimental results show that the proposed FAFD achieves the best performance among the speed-focusing methods.

Author Contributions: Conceptualization, C.S.W.; methodology, N.K. and J.-H.K.; software, N.K. and J.-H.K.; validation, N.K. and J.-H.K.; formal analysis, N.K. and J.-H.K.; investigation, N.K. and J.-H.K.; resources, C.S.W., N.K. and J.-H.K.; data curation, N.K. and J.-H.K.; writing—original draft preparation, N.K.; writing—review and editing, C.S.W.; visualization, N.K. and J.-H.K.; supervision, C.S.W.; project administration, C.S.W.; funding acquisition, C.S.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Basic Science Research Program of the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant NRF-2018R1D1A1B07043542. For C.S.W., this work was also supported by the Dongguk University Research Fund of 2021.

Data Availability Statement: WiderFace: http://shuoyang1213.me/WIDERFACE/ (accessed on 9 February 2022).

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

References

1. Kumar, A.; Marks, T.K.; Mou, W.; Wang, Y.; Jones, M.; Cherian, A.; Koike-Akino, T.; Liu, X.; Feng, C. LUVLi Face Alignment: Estimating Landmarks' Location, Uncertainty, and Visibility Likelihood. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 8236–8246.
2. Ning, X.; Duan, P.; Li, W.; Zhang, S. Real-time 3D face alignment using an encoder-decoder network with an efficient deconvolution layer. *IEEE Signal Processing Lett.* **2020**, *27*, 1944–1948. [CrossRef]
3. Chang, J.; Lan, Z.; Cheng, C.; Wei, Y. Data Uncertainty Learning in Face Recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 5710–5719.
4. Kim, Y.; Park, W.; Roh, M.-C.; Shin, J. Groupface: Learning Latent Groups and Constructing Group-Based Representations for Face Recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 5621–5630.
5. Anzalone, L.; Barra, P.; Barra, S.; Narducci, F.; Nappi, M. Transfer Learning for Facial Attributes Prediction and Clustering. In Proceedings of the 7th International Conference on Smart City and Informatization, Guangzhou, China, 12–15 November 2019; pp. 105–117.
6. Karkkainen, K.; Joo, J. FairFace: Face Attribute Dataset for Balanced Race, Gender, and Age for Bias Measurement and Mitigation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2021; pp. 1548–1558.
7. Yang, S.; Luo, P.; Loy, C.-C.; Tang, X. Wider Face: A Face Detection Benchmark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 5525–5533.
8. Earp, S.W.; Noinongyao, P.; Cairns, J.A.; Ganguly, A. Face detection with feature pyramids and landmarks. *arXiv* **2019**, arXiv:1912.00596.
9. Deng, J.; Guo, J.; Zhou, Y.; Yu, J.; Kotsia, I.; Zafeiriou, S. Retinaface: Single-stage dense face localisationd in the wild. *arXiv* **2019**, arXiv:1905.00641.
10. Qi, D.; Tan, W.; Yao, Q.; Liu, J. YOLO5Face: Why Reinventing a Face Detector. *arXiv* **2021**, arXiv:2105.12931.

11. Zhang, B.; Li, J.; Wang, Y.; Tai, Y.; Wang, C.; Li, J.; Huang, F.; Xia, Y.; Pei, W.; Ji, R. Asfd: Automatic and scalable face detector. *arXiv* **2020**, arXiv:2003.11228.
12. Zhang, S.; Chi, C.; Lei, Z.; Li, S.Z. RefineFace: Refinement Neural Network for High Performance Face Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 4008–4020. [CrossRef] [PubMed]
13. Tang, X.; Du, D.K.; He, Z.; Liu, J. Pyramidbox: A Context-Assisted Single Shot Face Detector. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 797–813.
14. Li, Z.; Tang, X.; Han, J.; Liu, J.; He, R. Pyramidbox^{++}: High performance detector for finding tiny face. *arXiv* **2019**, arXiv:1904.00386.
15. Najibi, M.; Samangouei, P.; Chellappa, R.; Davis, L.S. Ssh: Single Stage Headless Face Detector. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 4875–4884.
16. Zhang, S.; Zhu, X.; Lei, Z.; Shi, H.; Wang, X.; Li, S.Z. S3fd: Single Shot Scale-Invariant Face Detector. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 192–201.
17. Li, J.; Wang, Y.; Wang, C.; Tai, Y.; Qian, J.; Yang, J.; Wang, C.; Li, J.; Huang, F. DSFD: Dual Shot Face Detector. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5060–5069.
18. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. Ssd: Single Shot Multibox Detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37.
19. Zhao, X.; Liang, X.; Zhao, C.; Tang, M.; Wang, J. Real-time multi-scale face detector on embedded devices. *Sensors* **2019**, *19*, 2158. [CrossRef] [PubMed]
20. Jocher, G.; Stoken, A.; Chaurasia, A.; BoroVec, J.; Kwon, Y.; Michael, K.; Changyu, L.; Fang, J.; Abhiram, V.; Skalski, P.; et al. Ultralytics/yolov5: V6. 0—YOLOv5n 'Nano' models, Roboflow integration, TensorFlow export, OpenCV DNN support. *Zenodo Tech. Rep.* **2021**. [CrossRef]
21. Feng, Y.; Yu, S.; Peng, H.; Li, Y.-R.; Zhang, J. Detect Faces Efficiently: A Survey and Evaluations. *arXiv* **2021**, arXiv:2112.01787. [CrossRef]
22. Yashunin, D.; Baydasov, T.; Vlasov, R. MaskFace: Multi-Task Face and Landmark Detector. *arXiv* **2020**, arXiv:2005.09412.
23. Chi, C.; Zhang, S.; Xing, J.; Lei, Z.; Li, S.Z.; Zou, X. Selective Refinement Network for High Performance Face Detection. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 8231–8238.
24. Liu, W.; Liao, S.; Ren, W.; Hu, W.; Yu, Y. High-Level Semantic Feature Detection: A New Perspective for Pedestrian Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5187–5196.
25. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
26. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into High Quality Object Detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6154–6162.
27. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
28. Zhang, S.; Wen, L.; Bian, X.; Lei, Z.; Li, S.Z. Single-Shot Refinement Neural Network for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4203–4212.
29. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
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; pp. 1–9.
31. Yu, S. *libfacedetection.train*; GitHub: San Francisco, CA, USA, 2020; Available online: https://github.com/ShiqiYu/libfacedetection.train (accessed on 6 February 2022).
32. Linzaer. *Ultra-Light-Fast-Generic-Face-Detector-1MB*; GitHub: San Francisco, CA, USA, 2019; Available online: https://github.com/Linzaer/Ultra-Light-Fast-Generic-Face-Detector-1MB (accessed on 6 February 2022).
33. Jin, H.; Zhang, S.; Zhu, X.; Tang, Y.; Lei, Z.; Li, S.Z. Learning Lightweight Face Detector with Knowledge Distillation. In Proceedings of the 2019 International Conference on Biometrics (ICB), Crete, Greece, 4–7 June 2019; pp. 1–7.
34. 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.
35. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
36. Lin, T.-Y.; 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, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
37. Dvornik, N.; Mairal, J.; Schmid, C. Modeling Visual Context Is Key to Augmenting Object Detection Datasets. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 364–380.
38. Ghiasi, G.; Cui, Y.; Srinivas, A.; Qian, R.; Lin, T.-Y.; Cubuk, E.D.; Le, Q.V.; Zoph, B. Simple Copy–Paste Is a Strong Data Augmentation Method for Instance Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 2918–2928.
39. Dwibedi, D.; Misra, I.; Hebert, M. Cut, Paste and Learn: Surprisingly Easy Synthesis for Instance Detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1301–1310.

40. Kisantal, M.; Wojna, Z.; Murawski, J.; Naruniec, J.; Cho, K. Augmentation for small object detection. *arXiv* **2019**, arXiv:1902.07296.
41. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
42. Zhang, H.; Cisse, M.; Dauphin, Y.N.; Lopez-Paz, D. Mixup: Beyond Empirical Risk Minimization. *arXiv* **2017**, arXiv:1710.09412.
43. Zitnick, C.L.; Dollár, P. Edge boxes: Locating Object Proposals from Edges. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 5–12 September 2014; pp. 391–405.
44. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 2778–2788.
45. Arthur, D.; Vassilvitskii, S. *K-Means++: The Advantages of Careful Seeding*; Stanford: Stanford, CA, USA, 2006.