Efficient Face Detector Using Spatial Attention Module in Real-Time Application on an Edge Device

Muhamad Dwisnanto Putro, Duy-Linh Nguyen, and Kang-Hyun Jo(✉)

Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Ulsan 44616, South Korea
{dputro,ndlinh301}@mail.ulsan.ac.kr, acejo@ulsan.ac.kr

Abstract. The practical application requires a vision-based face detector to work in real-time. The robot application uses the face detection method as the initial process for the face analysis system. During its development, it utilizes an edge device to be used to process sensor information. Jetson Nano is a mini portable computer that is easily synchronized with sensors and actuators. However, traditional detectors can work fast on this device but have low performance for occlusion cases, multiple poses, and small faces. On the other hand, CNN-based detectors that implement deep layers are slow to run on low memory GPU devices. In this work, an efficient real-time face detector using a simple spatial attention module was developed to localize faces rapidly. The proposed architecture consists of the backbone module to efficiently extract features, the light connection module to reduce the size of the detection layer, and multi-scale detection to perform prediction of faces on various scales. As a result, the proposed detector achieves competitive performance from state-of-the-art fast detectors on several benchmark datasets. In addition, this efficient detector can run at 55 frames per second in video graphics array resolution on a Jetson Nano.

Keywords: Face detector · Efficient network · Jetson Nano · Real-time application

1 Introduction

Face detection is a popular field in computer vision for localizing facial areas in an image. This method is the initial process used for advanced vision systems such as face recognition, emotion, gender, and landmarks [1–4]. Face analysis is widely used in intelligent video technology. It encodes data from specific facial components to determine the characteristics and models of that information. Besides, face detection is a current technology required in a portable device to unlock mobile phones and log in payment accounts.

Nowadays, a security system is needed in every aspect to monitor the environment. It is used to prevent crime that occurs in public areas [5]. Even developed countries use it to find missing people. Intelligent video surveillance not only applies this monitoring

© Springer Nature Switzerland AG 2021
D.-S. Huang et al. (Eds.): ICIC 2021, LNCS 12836, pp. 829–841, 2021.
https://doi.org/10.1007/978-3-030-84522-3_67

application, but it can also explore the control system to optimize its performance. The system needs to be supported by massive process memory availability because the face detectors work with heavy compressing data. It requires expensive hardware for the detector to work optimally. Another application is for Human-robot Interaction (HRI) to implement a face detector on a robot supporting the facial recognition system's capabilities [6]. Even emotion recognition is applied to a robot service to predict the expressions of the customer. Both methods require a face detector as an initial step to localize the face area.

A face has a distinctive texture and color against the background. It is unique because every human being has a different characteristic [7]. Information from facial organs is important feature data to identify them. The eye has an ellipse shape and a characteristic color that is different from the nose. The lips contain a reddish color that is different from the eyebrows and chin. Although each organ's shape is almost the same as the other background features, the relationship between these features generates a face model strengthening the distinguisher against the background. Therefore, the extraction of faces and the relation between features is essential information to identify a face in the image.

Several works have presented conventional methods for detecting faces in an image. The Viola-Jones discovered important facial features by applying Haar-like features that were moved around using a sliding window [8]. The difference between light and dark areas is used to identify interest facial features. Combining an integral image and AdaBoost Learning generates a learning model that can work quickly to classify facial and background features. Although this detector can work in real-time, it has low accuracy in occlusion cases, multiple poses, and small faces. The Haar-like feature has limitations in capturing facial feature information that is blocked and small sizes. The rotation-invariant drops the performance of this method in a real-case application for finding parallel facial features.

The modern method introduced the Convolution Neural Network (CNN) as a robust features extractor feature [9]. It employs convolutional operations on input features and kernel shape. It adopts a neural network approach that updates the kernel weights to improve network performance and minimize error rates. This method obtains a high degree of accuracy for image classification work [10]. It can distinguish categories from images by extracting specific information.

On the other hand, CNN capabilities have been implemented to distinguish facial and background features to localize facial areas in an image. Several studies [11–13] achieved high accuracy for the complicated challenge, but practical applications prevent these methods from working in real-time. Moreover, robotic applications require a vision system to work quickly on low-cost devices. The deep backbone tends to employ huge filter layers, resulting in over a million total parameters. VGG-16 [14] and ResNet [15] are benchmark backbones that have successfully filtered out important object features, but these models also generate large parameter weights.

Additionally, MobileNet [16] and ShuffleNet [17] have been introduced as lightweight CNN backbones, but these models have stagnated in real-time work when implemented a low-cost device. Jetson Nano is an edge device generally used in IoT (Internet of Things) and robotics application [18]. The CNN method is relatively implemented in this hardware as a visual approach to sense the object and environment.

The lightweight architecture employs a few slim layers of convolution and delivers efficient computation power. Apart from preventing premature saturation, this network produces a small number of parameters. However, superficial networks do not produce high accuracy. Several methods apply an attention mechanism to improve the feature extraction performance [19]. This module summarizes the essential features and generates attention weights to update the feature input. The spatial attention module captures specific information based on the feature map's size representing valuable information from each cell element [20]. This block employs sigmoid activation to generate probability weights and produces low computation costs and parameters. It is very efficient to increase the performance of a shallow backbone implemented in a real-time detector.

Based on the above issues, this study proposes a lightweight CNN with simple spatial attention to rapidly localize facial areas. The contributions of this work are as follows:

1. A new efficient CNN architecture is used to build a face detector that is fast works in a real-time application.
2. A simple spatial attention module was introduced as a reinforcing module for shallow backbones to support the network's efficiency and effectiveness.

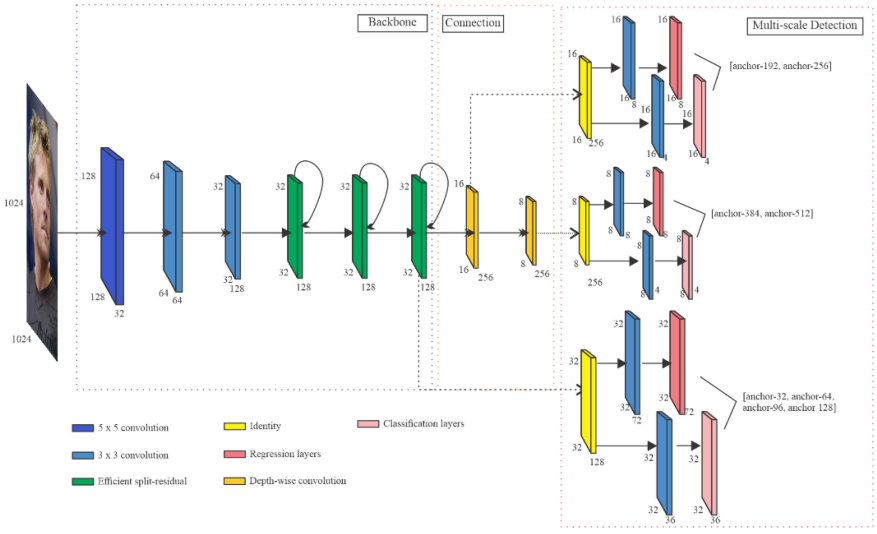

Fig. 1. The proposed architecture of face detector. It uses a combination of 1×1 and 3×3 convolution as extractor features. Multilayer detection with anchors assignment plays a role in predicting faces at various scales.

2 Proposed Architecture

In this section, the detail of our proposed architecture is explained. The proposed method applies a shallow layer of Convolutional Neural Network and combines each module

to produce an efficient architecture. Instead of significantly reducing the detector performance, the main module consists of backbone, connection, and multi-scale detection layers, as shown in Fig. 1.

2.1 Backbone Module

CNN-based architectures tend to use multiple layers in the backbone to extract essential information. This layer has an impact on a large number of parameters and computations. The proposed detector employs nine layers of convolution by combining 3×3 and 1×1 filters. Specifically, the backbone module consists of a shrink layer for reducing the feature map size and a stem module that sequentially discriminates against facial and background features. The shrink module employs a 7×7 filter at the beginning of the stage to significantly reduce the feature map's size [9]. It is followed by a 3×3 convolution with a stride of 2, generating the 32×32 with 128 channel feature map. This filter effectively and efficiently captures facial features of various sizes [14]. Additionally, in order to prevent saturation and vanishing gradients, ReLU and Batch-Normalization are used after the convolution process at each layer. ReLU selects positive values and ignores negative values from feature input, while Batch-Normalization maintains the average distribution is close to 0 and the output standard deviation close to 1 for each mini-batch.

The proposed detector is designed as a low weight efficiency detector by applying a partial transfer structure at the stem module. An efficient split-residual block divides input features map into two parts and unites them at the end of the module. Figure 2 (a) shows that the split approach reduces computation at the start of the module without removing other parts' information. Half of the feature map is processed in feature extraction, sequentially applying convolution. At the same time, the attention mechanism is applied to other chunks. Finally, the efficient module concatenates the representation of the essential elements and other extraction parts.

2.2 Simple Spatial Attention

Attention mechanism increases the interest features intensity by eliminating distinctive features and reducing trivial information [20]. The feature location on each feature map is valuable information. Thus, spatial information tends to show cues of extracted facial features. The spatial attention module is proposed as an enhanced module without producing excessive computation. Figure 2 (b) shows that average pooling for each cell of the input features is applied to summarize the channel array information. The simple spatial attention is defined as follows:

$$S_{att} = \sigma(W_{c1} AVG(x_i)) \tag{1}$$

where σ is sigmoid activation to generate probability weights from a single feature map representation of the simple convolution ($c1$). This module updates each element of the input features and implements element-wise multiplication with weighted maps. Therefore, this module reduces non-facial features and enhances distinctive facial features to strengthen the discrimination process.

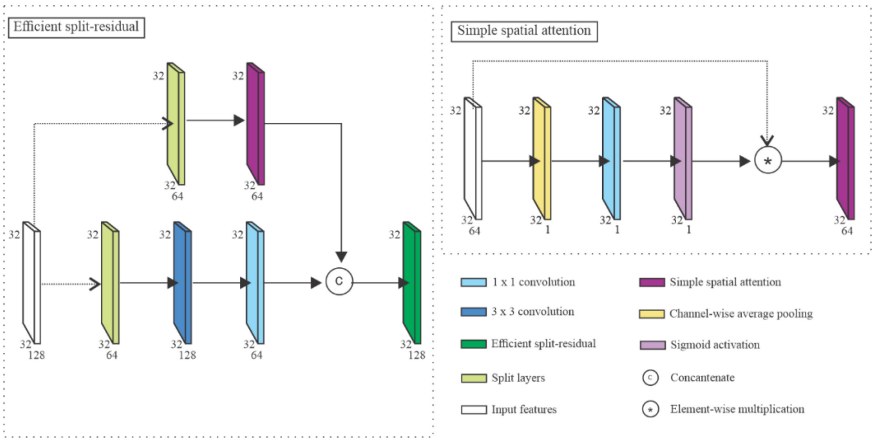

Fig. 2. The efficient split-residual as a stem module that sequentially extracts features and produces a few parameters and low-cost computation power (a). Simple spatial attention is employed at part of the input feature map to capture the representation of the valuable elements (b).

2.3 Connection Module

CNN-based detectors use the connection module to link between the prediction layers. Multilayer detection uses prediction layers with various scales and employs shrink blocks to create feature maps of different sizes. Instead of using standard convolution, it employs Depth-wise convolution with a stride of two to save the parameters. This convolution does not apply a multi convolutional filter to each input channel, but it only uses a single convolutional filter. This block emphasizes computational efficiency with fewer multiplication and addition operations on each channel. Besides, this approach also specifies that the number of relations between each kernel and input elements is equal to the number of channels. A 3×3 kernel with linear operations is used to filter specific object features. Furthermore, this is also followed by ReLU and BatchNormaliszation to maintain network performance and to avoid accuracy dropped in the training process.

2.4 Multi-scale Detection and Anchor Assignment

The proposed detector is a one-stage architecture that predicts classes and face bounding box location. It also assigns anchors of various sizes as initial bounding boxes. This feature map is generated by a 3×3 convolution, which predicts the category and regression layers. Category prediction is face and none, while regression determines the location coordinates (x, y) and fits the scale of bounding boxes, namely height (h) and width (w). The variation in face size emphasizes that the detector needs to assign predictions with various scales. A three-layered detection handles small, medium, and large faces, with feature map sizes including 8, 16, and 32, respectively. In addition, anchor assignments of various sizes are placed on each prediction layer. Anchors 32, 64, 96, and 128 were assigned to predict small faces, while 192 and 256 for medium and large faces were applied 384 and 512.

2.5 Multi Boxes Loss Function

The CNN-based detector assigns a loss function to measure each prediction error compared to the ground-truth label and location. Backpropagation exploits the performance of this function to optimize the weight neurons and minimize prediction errors. The end of the detection layer predicts regression (x, y, h, w) and label classes. Each prediction variable has calculated the difference with the ground-truth value. Then the total loss applies two objectives with the imbalanced parameter. The multi boxes loss is assigned to each predicted anchor (i-th), which is defined as

$$Loss\left(p_i, r_i^*\right) = \frac{2\sum_i L_{cat}\left(p_i, p_i^*\right)}{N} + \frac{\sum_i L_{reg}\left(r_i, r_i^*\right)}{N}, \tag{2}$$

where p_i, p_i^*, r_i, r_i^* are the prediction category of classes, ground-truth label, four coordinate vectors of predicted location, and ground-truth scale and location box, respectively. $L_{cat}\left(p_i, p_i^*\right)$ applies Softmax-loss [12] to calculate losses from predictive class classes, while L1-smooth loss [11] is used to calculate regression losses $L_{reg}\left(r, r_i^*\right)$. It gives greater weight to the loss classification side, which tends to produce lower scores at the default training stage. Therefore, this manipulation balances the updating weight's performance in neurons to work fairly on both sides of the function.

3 Dataset and Implementation Setup

Proposed models are trained on a WIDER dataset containing 32,203 total images, with only 12,800 of the training set is used by detector as knowledge to learn the characteristics of facial features. Additionally, PASCAL face, AFW (Annotated Faces in the Wild), and FDDB (face Detection Data Set and Benchmark) are test datasets for evaluating training models. In order to enrich training data variation, random cropping, scale transformation, color distortion, and horizontal flipping are applied as augmentation methods. The end of this process produces 1024×1024 RGB as the input image size of training.

The training process divides all images dataset into 32 batches, shortening the time from the network for learning data on small partitions. Proposed models are trained through the end-to-end stage with random weights at the beginning of the epoch. The Stochastic Gradient Descent (SGD) was used to optimize the neuron weights in the backpropagation process with $5 \cdot 10^{-4}$ weight decay and 0.9 momentum. It assigns different learning rate weights for the variation in the number of epochs. The initial stage uses a 10^{-3} learning rate for 200 epochs, followed by a 10^{-4} learning rate for 100 epochs, the next 10^{-5} learning rate for 50 epochs, and the last 20 epochs at a 10^{-6} learning rate. Intersection over Union (IoU) of 0.5 is used to select predicted anchors that overlap in the evaluation process. Finally, the training, evaluation, and real-time testing processes of this detector are implemented in the PyTorch framework.

4 Experiments and Results

In this section, the proposed detector is tested for the performance of each module and evaluates on benchmark datasets. It also compares Average Precision (AP) with various competitors. Besides, the efficiency of the face detector is also tested in real-time applications on the Jetson Nano device.

4.1 Ablative Study

This ablative study comprehensively shows the strength of each proposed module, including shrink, stem with attention, connection, and multi-scale detection module. Each proposed module gradually is removed, it analyzes the accuracy and number of parameters with the same training configuration. Table 1 shows that each module increases the accuracy and number of parameters of the detector. The stem module increases the accuracy of this detector by 4.12% and adds 220K parameters. Additionally, the proposed attention module also increases the accuracy by 1.08%, but this only adds a few parameters.

Table 1. Ablative study of the proposed modules on FDDB dataset

Modules	Proposed detector				
Simple spatial attention	√				
Stem	√	√			
Connection	√	√	√		
Multi-scale detection	√	√	√	√	
Shrink	√	√	√	√	√
Number of parameter	433.363	405.700	186.448	152.656	152.656
Average precision (%)	96.46	95.38	91.26	90.50	82.71

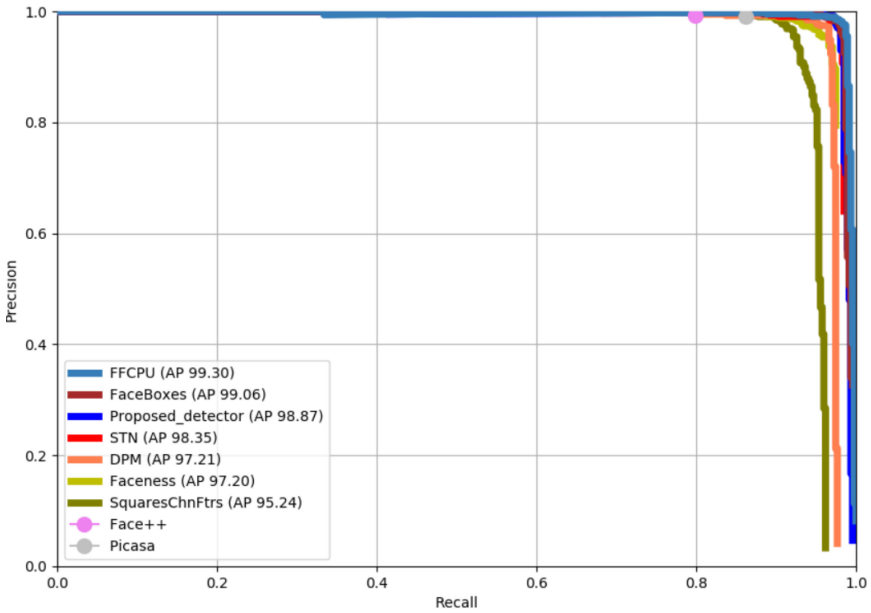

Fig. 3. Evaluation of proposed detector on AFW dataset

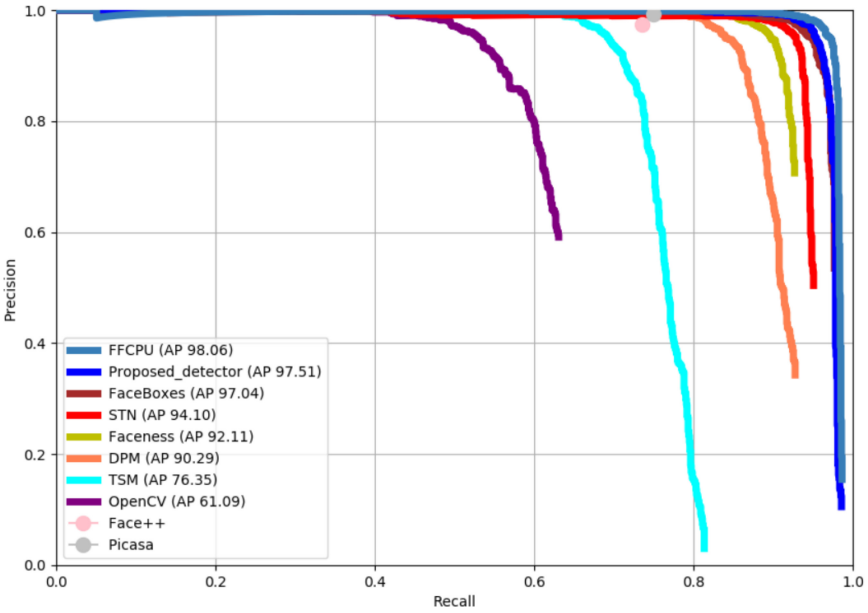

Fig. 4. Evaluation of proposed detector on PASCAL face dataset

4.2 Evaluation on Datasets

AFW Dataset. This dataset is obtained from Flickr images and has 205 images with 473 faces. It contains various background and viewpoint faces (e.g., ages, glasses, skin colors, expression, etc.). Figure 3 shows that the proposed detector achieves 98.87% of AP, which outperforms the STN detector. However, this detector is below the accuracy of FaceBoxes [11] and FFCPU [12]. It has a more robust performance than commercial detectors (Face++ and Picasa). Figure 6 (a) shows the prediction results for face locations on several images. However, the proposed detector generates a false positive for the texture of the background and the human component. It is successful in detecting faces of various poses, occluded and of various sizes.

PASCAL Face Dataset. This dataset contains 1,335 faces from 851 images obtained from the test set of PASCAL person dataset. It provides a variety of face appearances and poses with indoor and outdoor backgrounds. Figure 4 shows that the proposed detector outperformed FaceBoxes in this dataset by achieving 97.51% of AP. The qualitative results in Fig. 6 (b) show that the detector can detect faces at various brackets and produce error prediction on textures and colors similar to faces.

FDDB Dataset. This dataset obtains from news articles on Yahoo websites, which contains 5,171 faces annotated in 2,845 images. It provides a variety of challenges, such as occlusions, large poses, and low image resolutions. Proposed detectors are evaluated on discrete criteria, as shown in Fig. 5. The AP on this graph means the true-positive rate at 1,000 false positives. The performance of proposed detector is below LFFD [13], FFCPU, and FaceBoxes. The shallow layer of the detector cannot correctly discriminate

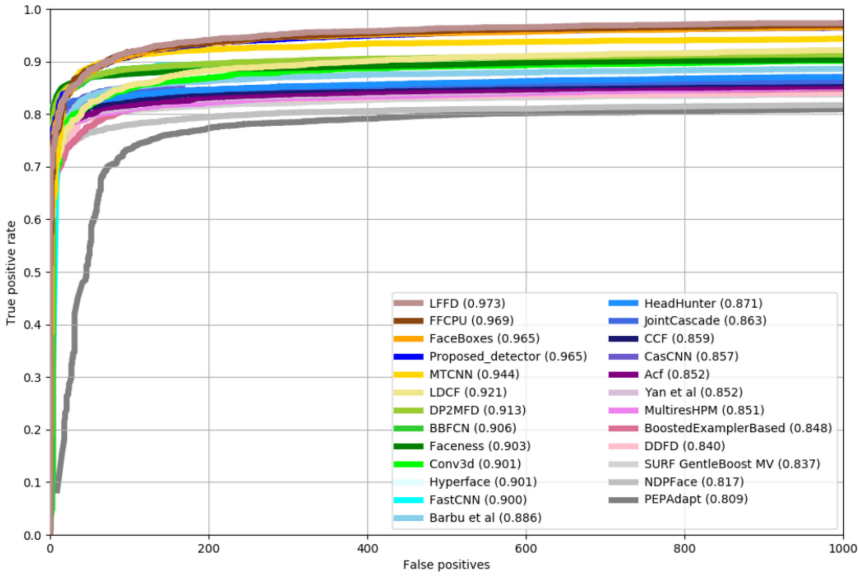

Fig. 5. Evaluation of proposed detector on FDDB dataset at 1000 false positives

facial and non-facial features, as shown in Fig. 6 (c). Hand features are predicted as facial features. However, the proposed detector produces a lower number of parameters, and it can work faster on the Jetson Nano device.

4.3 Runtime Efficiency

The CNN-based detector is useful if it can work quickly on low-cost devices. The CNN models require large and expensive GPUs to work in real-time. In general, robotics applications use a portable computer that can acquire sensor and actuator data. Jetson Nano is an edge device that is easily connected and synchronized with robotic devices. However, heavyweights detectors are slow to work in real-time on this device. The proposed detector generates 433,363 parameters which are lower than the other competitors. The LFFD detector achieved the best performance. However, this produces 2M trainable parameters, as shown in Fig. 7. Therefore, this detector works slowly in real-time applications.

Testing of the real-time application using a webcam as an input device, this data is directly processed on each detector. The speed of each detector at different video input sizes is shown in Fig. 8. The proposed detector outperformed competitors' speed by achieving 54.87 FPS at VGA resolution. It differs 13 FPS from the slower FFCPU detector. The implementation at Full HD resolution shows that the proposed detector can work in real-time by reaching a speed of 25.89 FPS, while other detectors work slowly with speeds below 20 FPS. The superficial model of the proposed detector emphasizes

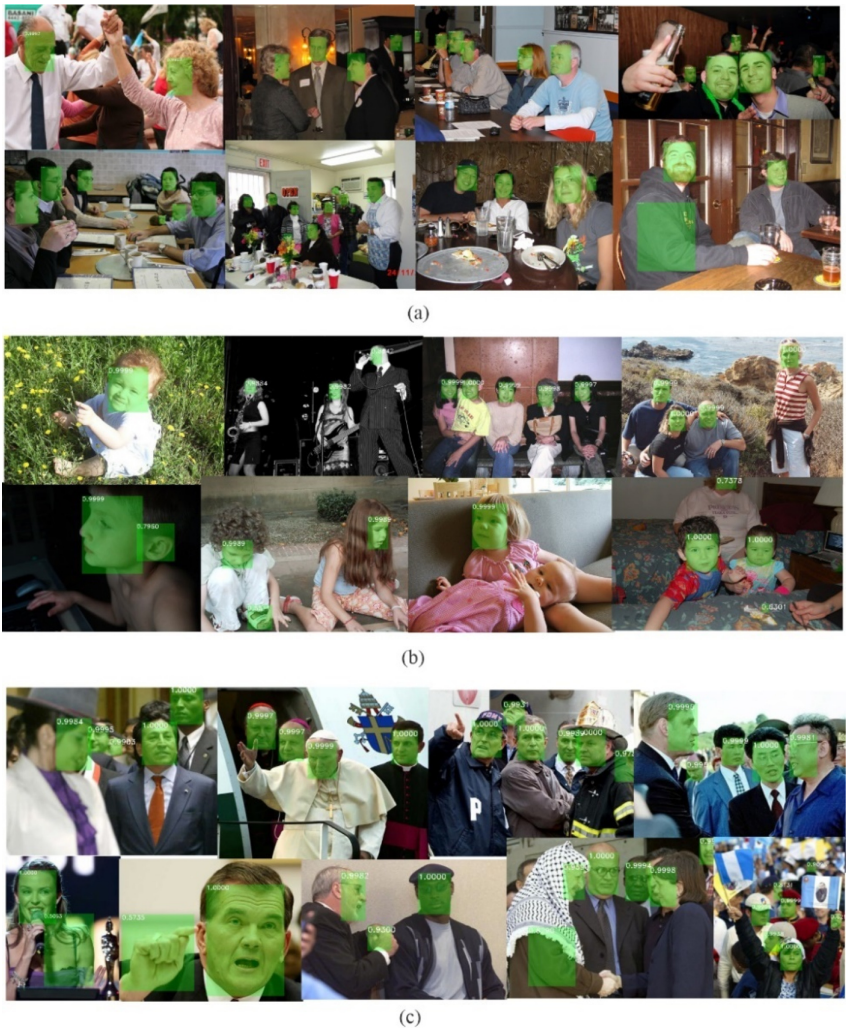

Fig. 6. Visualization of result from proposed detector on AFW (a), PASCAL face (b), and FDDB datasets (c)

the computational efficiency and the number of parameters. The backbone module produces a small number of parameters, but it maintains the quality of feature extraction. Furthermore, the simple spatial attention module improves detector performance without significantly slowing down the real-time detector speed.

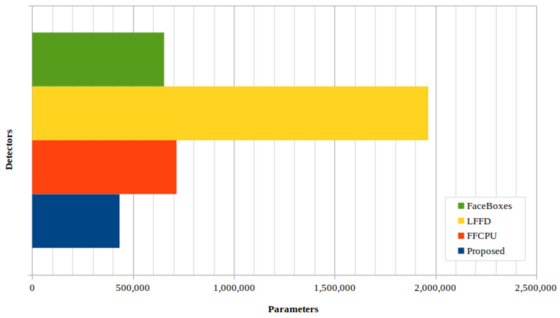

Fig. 7. Comparison of trainable parameters detector with other competitors

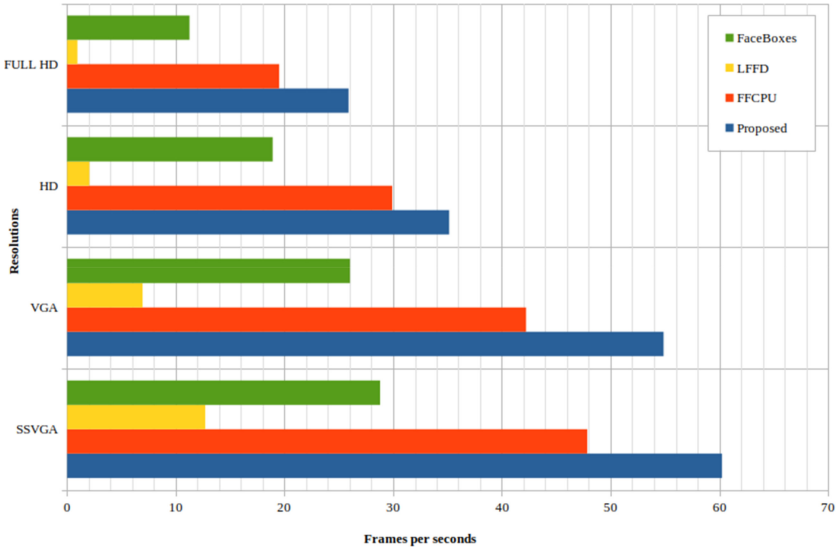

Fig. 8. Comparison of data processing speed in real-time application with other competitors at different video input sizes.

5 Conclusions

This work presented an efficient real-time face detector using a simple spatial attention module implemented on a Jetson Nano. The proposed architecture consists of three main modules, a backbone, a connection, and multi-scale detection. The backbone plays an essential role in discriminating facial and background features by applying a simple spatial attention block. This module effectively improves backbone performance without adding significant number of parameters. Proposed detectors produce trainable parameters that are lower than CNN-based fast detectors. Finally, the results showed that the proposed detector achieved competitive performance with state-of-the-art fast detectors and outperformed their speed by 55 FPS in real-time on a Jetson Nano. In the future,

the quality of model training can be improved by implementing and exploring IoU loss and Focal Loss without reducing speed in real-time applications.

Acknowledgement. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the government (MSIT) (No. 2020R1A2C200897212).

References

1. Choi, J.Y., Lee, B.: Ensemble of deep convolutional neural networks with gabor face representations for face recognition. IEEE Trans. Image Process. **29**, 3270–3281 (2020). https://doi.org/10.1109/TIP.2019.2958404
2. Putro, M.D., Nguyen, D.-L., Jo, K.-H.: A dual attention module for real-time facial expression recognition. In: IECON 2020 the 46th Annual Conference of the IEEE Industrial Electronics Society, Singapore, pp. 411–416 (2020). https://doi.org/10.1109/IECON43393.2020.9254805
3. Zhou, Y., Ni, H., Ren, F., Kang, X.: Face and gender recognition system based on convolutional neural networks. In: 2019 IEEE International Conference on Mechatronics and Automation (ICMA), Tianjin, China, pp. 1091–1095 (2019). https://doi.org/10.1109/ICMA.2019.8816192
4. Hoang, V.-T., Huang, D.-S., Jo, K.-H.: 3-D facial landmarks detection for intelligent video systems. IEEE Trans. Industr. Inf. **17**(1), 578–586 (2021). https://doi.org/10.1109/TII.2020.2966513
5. Awais, M., et al.: Real-time surveillance through face recognition using HOG and feedforward neural networks. IEEE Access **7**, 121236–121244 (2019). https://doi.org/10.1109/ACCESS.2019.2937810
6. Putro, M.D., Jo, K.: Real-time face tracking for human-robot interaction. In: 2018 International Conference on Information and Communication Technology Robotics (ICT-ROBOT), Busan, Korea (South), pp. 1–4 (2018). https://doi.org/10.1109/ICT-ROBOT.2018.8549902
7. Li, X., Yang, Z., Wu, H.: Face detection based on receptive field enhanced multi-task cascaded convolutional neural networks. IEEE Access **8**, 174922–174930 (2020). https://doi.org/10.1109/ACCESS.2020.3023782
8. Paul, V., Michael, J.: Robust real-time face detection. Int. J. Comput. Vision **57**(2), 137–154 (2004)
9. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 818–833. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10590-1_53
10. Lei, X., Pan, H., Huang, X.: A dilated CNN model for image classification. IEEE Access **7**, 124087–124095 (2019). https://doi.org/10.1109/ACCESS.2019.2927169
11. Zhang, S., Wang, X., Lei, Z., Li, S.Z.: Faceboxes: a CPU real-time and accurate unconstrained face detector. Neurocomputing **364**, 297–309 (2019). ISSN 0925-2312
12. Putro, M.D., Jo, K.-H.: Fast face-CPU: a real-time fast face detector on CPU using deep learning. In: 2020 IEEE 29th International Symposium on Industrial Electronics (ISIE), Delft, Netherlands, pp. 55–60 (2020). https://doi.org/10.1109/ISIE45063.2020.9152400
13. He, Y., Xu, D., Wu, L., Jian, M., Xiang, S., Pan, C.: LFFD: A Light and Fast Face Detector for Edge Devices (2019). arXiv:1904.10633
14. Szegedy, C., et al.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (Jun 2015). https://doi.org/10.1109/CVPR.2015.7298594

15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
16. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.: MobileNetV2: inverted residuals and linear bottlenecks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, pp. 4510–4520 (2018). https://doi.org/10.1109/CVPR.2018.00474
17. Zhang, X., Zhou, X., Lin, M., Sun, J.: ShuffleNet: an extremely efficient convolutional neural network for mobile devices. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, pp. 6848–6856 (2018). https://doi.org/10.1109/CVPR.2018.00716
18. Süzen, A.A., Duman, B., Şen, B.: Benchmark analysis of Jetson TX2, Jetson Nano and Raspberry PI using deep-CNN. In: 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, pp. 1–5 (2020). https://doi.org/10.1109/HORA49412.2020.9152915
19. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, pp. 7132–7141 (2018). https://doi.org/10.1109/CVPR.2018.00745
20. Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.): ECCV 2018. LNCS, vol. 11210. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01231-1