VOL. 24, NO. 1, 2025, 35-40
elektrika.utm.my
ISSN 0128-4428

ELEKTRIKA
Journal of Electrical Engineering

Profile Face Detection using SSD MobileNetV2 with Feature Pyramid

Aifian Adi Sufian Chan*, **M.F.L Abdullah**, **Saizal Md Mustam**, **Farhana Ahmad Po'ad** and **Ariffuddin Joret**

[1]Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia (UTHM), Johor, Malaysia

*Corresponding author: he200022@siswa.uthm.edu.my

Abstract: A profile face is an extreme face pose. This makes the facial poses harder to detect compared to normal facial poses due to a lack of facial feature information. Hence, this paper proposes the combination of three different networks to detect the profile faces in an image. The CNN architecture proposed to be used is SSD MobileNetV2 with FPN, where it is divided into three different networks. MobileNetV2 acts as the backbone of the overall architecture. The main function is to extract feature maps from the input image. The feature map is fed into a feature extractor layer that consists of FPN and generates a feature pyramid that consists of feature maps in different scales. The SSD is the detection network that will detect profile faces in the image and produce candidate bounding boxes. The NMS is applied as a post-processing step to remove duplicate detection and low confidence level bounding boxes. As a result, the final output is the highest confidence level of the detected profile face's bounding box in the image. Based on the experimental results, the proposed work has achieved good accuracy in detecting the profile face view.

Keywords: Profile face, convolutional neural network, SSD MobileNetV2, FPN, NMS

Article History: received 2 November 2024; accepted 12 February 2025; published 30 April 2025

1. INTRODUCTION

Face detection is a computer vision application that involves various techniques to detect faces in any digital resource, such as video or images. Face detection has become an integral part of modern human life as it has been used in various applications for security and entertainment purposes. However, face detection faces various challenges that might affect its performance in detecting human faces. The challenges that have been identified are in areas such as occlusion [1]– [3], variation in facial poses [3], variation in illumination [1], [3], and the scale factor of the human face [4].

In this research, the challenge that will be focused on is the variation of facial poses, especially the extreme face pose of the profile face view (PFV). PFV is a face pose when the face is taken from the side, showing just the profile of their nose, lips, eye, and mouth, where only one side of the face of the image is visible. Most face detection algorithms fail to detect PFV as it does not provide enough facial data or visual cues to be accurately identified as a human face. Figure 1 shows an image of a PFV. This is the main motivation of this research: to develop a profile face detection method that is capable of detecting PFV. It can be used in various applications related to security. One example is that profile face detection is used in forensic science as an additional resource to detect wanted criminals from the profile view of the CCTV. It can also be used in automated gate security that allows user access by detecting and recognizing the user's profile face.

The main contribution of this paper is the development of a profile face detection method that helps detect PFV in an image. The development of the method used a convolutional neural network (CNN) as the main architecture for detecting the PFV.

Figure 1. Sample image of PFV

2. RELATED WORKS

Face detection has been one of the most studied areas of computer vision in the past few decades due to its demands and usage in a wide range of applications that include security systems, social media platforms, and even virtual reality. The ability to detect the human face accurately and efficiently has become essential for many modern applications, and this has caused a significant amount of research and development in this area. The face detection technique can be classified into two different categories: the traditional face detection method and the CNN-based face detection method [4].

2.1 Traditional Face Detection Method

Traditional face detection methods use handcrafted features as the main way to detect the face in an image. A popular example is the Viola-Jones algorithm [5], which uses Haar-like features to detect features and a cascaded classifier to quickly detect faces in an image. Local Binary Pattern (LBP) is another common technique where it acts as a feature descriptor to capture unique characteristics of a human face [6]. By comparing the patterns of different regions within an image to a pre-defined pattern, the LBP can classify the face region in an image. Histogram of Oriented Gradients (HOG) [7] is a feature descriptor technique that captures the local appearance of a face by analysing the distribution of gradient orientations in an image. It used sliding windows and a trained HOG classifier to detect the face region in an image. Another example is skin colour segmentation [8], [9], which relies on segmentation of human skin within the range of colour in the YCbCr colour space. The segmentation allows the algorithm to detect the human face in the image. The Active Appearance Model (AAM) [10] is a statistical model that is trained to learn the appearance and shape parameters of the face in different variations. This model can be used to detect faces by matching the learned parameters to detect faces in an unseen image.

Here are some examples of traditional methods that have been used. The traditional method is much simpler and requires less computational power. However, the traditional method is more susceptible to many challenges that affect the detection rate, such as illumination, occlusion, and face pose. In addition, the traditional face detection model works poorly when detecting PFV, as the traditional method focuses on frontal face poses.

2.2 Convolutional Neural Network (CNN) Based Face Detection Method

With the rapid development and breakthrough of CNN in recent years, more researchers have focused their attention on it. The main reason is that CNN provides a more robust solution for face detection. Multi-Task Cascaded Convolutional Neural Network (MTCNN) [11] is a popular face detection algorithm that uses a three-stage cascaded CNN architecture to detect faces at different scales and orientations. MTCNN used the three different networks to perform face detection, facial landmarking, and alignment. Another example is TinaFace [12]. Here it used ResNet50 as the backbone and the FPN network as the feature extractor, the network allowing it to detect faces in various scales and sizes. IRNet is an improved version of the RetinaNet model that is customized for face detection [13]. It introduces the use of the feature fusion model N-FPN, which helps to improve the detection accuracy of the model. RefineFace is based on RetinaNet and integrated with five additional modules to improve its capability of detecting faces in an unconstrained environment. YOLO5Face is a face detector that is based on the YOLOv5 object detector and has achieved state-of-the-art performance in the Wider Face dataset [14].

The current face detection methods use diverse CNN architectures and have attained state-of-the-art performance on various face databases, such as Wider Face. The models are able to detect faces in various face poses, but their effectiveness in detecting extreme face poses such as PFV is limited due to the lack of facial features. Hence, this limitation has become the motivation for developing a CNN- based face detection model that is specifically aimed at detecting PFV.

3. PROPOSED METHODOLOGY

3.1 SSD MobileNetV2 Architecture

The convolutional network that is used in this proposed method is SSD MobileNetV2 with FPN. It combines three different networks into a singular network. This combination will provide a high- accuracy and high-efficiency network capable of detecting objects of various sizes while maintaining real-time performance on limited computational hardware.

The network can be divided into three parts. The backbone of the network is MobileNetV2 [15], whose main function is to extract features from the input image. The main reason why MobileNetV2 is used instead of other well-established networks such as VGG16 [16] or ResNet [17] is its lightweight nature. The network is capable of delivering high-accuracy detection while remaining lightweight due to three important key design choices. The key designs that were highlighted are inverted residual, linear bottleneck, and depthwise separable convolution [15]. The inverted residuals reduce the dimensionality of the input feature maps, which helps reduce the number of parameters and speed up computation. In addition, linear bottlenecks further reduce the computational cost by reducing the dimensionality of the input feature maps before applying them to depth-wise separable convolutions. The MobileNetV2 architecture used depthwise separable convolutions instead of traditional convolutional layer. This approach significantly reduces the number of parameters used for each convolutional operation while maintaining the ability to extract important features from the input data.

The next part of the network is the feature extractor using the Feature Pyramid Network (FPN) [18]. The incorporation of FPN is to help the model detect objects of different sizes and scales, especially in this case, a human profile face, which can easily differ in size depending on the situation. FPN is a technique that generates feature pyramids with multi-scale feature maps. FPN takes the output feature maps from the backbone network and generates a set of feature maps with different spatial resolutions. The FPN consists of two pathways: the bottom-up pathway and the top-down pathway. The bottom-up pathway increases the resolution of feature maps while reducing their channel dimension. The top-down pathway gradually reduces the resolution of feature maps and merges them with the corresponding feature maps from the bottom-up pathway. The resulting set of feature maps has a pyramid-like structure with different scales, as shown in Figure 2.

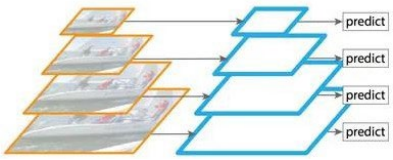

Figure 2. The feature pyramid generated by FPN [18].

The final part of the proposed network architecture is the detection network. Single Shot Detector (SSD) [19] is used as the detection network, where it uses the feature pyramid generated by FPN as the input. It uses a set of anchor boxes with different aspect ratios and scales to predict the locations of the PFV in the image. For each anchor box, the detection network predicts the probability of a PFV being present in that box and the offsets needed to adjust the bounding box to the detected PFV. The detection network uses convolutional and pooling operations to process the generated feature pyramid and produces two outputs: the PFV confidence level and the bounding box at each anchor box in each spatial location in the feature maps. However, the number of candidates generated can be huge, so the non- maximum suppression (NMS) algorithm is applied. NMS will remove duplicate detections and select the most confident predictions of PFV in the image as the final output. The overall parameters used in the network are less than 12 million.

The overall architecture is shown in Figure 3, where the input image will be fed into the trained network. The base network will extract the feature map from the input image. The feature map is then used as the input for the FPN network. The FPN network will generate a feature pyramid that is fed into the detection network, where it will produce candidates for bounding boxes of the detected PFV in the images. The resulting detections are post-processed using NMS to remove duplicate detections and low confidence detections. This will give an accurate result in the final output of the bounding box of PFV.

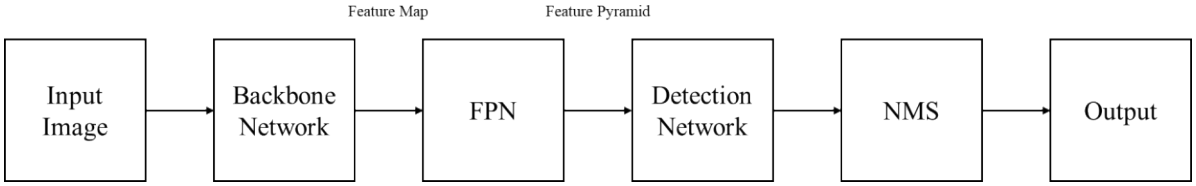

Figure 3. Overall architecture of the proposed network

3.2 Training Process of the SSD MobileNetV2

In this subsection, the training process for SSD MobileNetV2 will be explained in detail. The training process is divided into three parts: dataset preparation, data augmentation, and hyperparameter tuning.

3.2.1 Dataset Preparation

The Profile Face Database (PFD) was created by combining several datasets into a single dataset. The datasets that were used are the FEI database [20], UK Research Lab London Set [21], Iranian women face dataset [22], and Sibling databases [23]. In each database, only the profile face view images were selected for the Profile Face Database. The total number of PFV images collected is 2025. The images consist of participants from various age groups, skin colours, genders, and ethnicities. This provides huge diversity for the network to learn from. To get the region of interest (ROI) from each image, LabelImg software was used to label and extract the bounding boxes from the PFD database. The data was split into two sets: the training set and the test set. The ratio of the training dataset to the testing dataset is 4:1. The training set is used to train the network, while the testing set is used to evaluate its performance. Figure 4 shows a sample of a labelled image using LabelImg software.

Figure 4. Sample labelled image in LabelImg software.

3.2.2 Data Augmentation

Since the PFD has only 2025 images and can be classified as a small dataset, data augmentation is used to artificially increase the training data by creating new data from existing data in the PFD. This creates even more variation for the network to learn from. In addition, data augmentation also helps prevent overfitting in networks [19–21]. There are many techniques available in data augmentation, such as geometric transformations, colour space transformations, and rotation. The images in the dataset were augmented using four different augmentation techniques that involve rotation, shift, scaling, and colour transformation. Figure 5 shows the sample images that have undergone data augmentation.

Figure 5. Sample augmented images.

3.2.3 Hyperparameter Tuning

Hyperparameter tuning is one of the major steps in the training process. According to researchers [27], the optimization of the hyperparameter is not the same for all problems. There are many hyperparameters that can be tuned. However, the hyperparameter tuning in this network will involve the batch size, learning rate (LR), optimization algorithm, and regularization technique used.

During the training process, batch size and LR have drawn the most attention as they determine the speed of convergence of the trained model. Batch size refers to the amount of training data used in a single iteration. The batch size used in the proposed work is eight; the main reason is due to the insufficient memory of the hardware available. Higher batch sizes required more memory to store intermediate activations and gradients. The next hyperparameter is the optimizer algorithm. Today, there are many optimization algorithms that are available, but the momentum optimizer was used. The reason is because of the fast convergence and better generalization of data.

The learning rate of the network is used based on the cosine decay learning rate schedule. The algorithm was used as it provided a smoother and continuous decrease in learning rate, which improved the stability and convergence of the training process. The base learning rate is set to 0.09667, while the warmup learning rate is set at 0.02666. The final hyperparameter is the regularization technique, and in this paper, the L2 regularization (Ridge regularization) was used as it helps in avoiding overfitting and encourages the use of small weights, which helps in reducing the complexity of the model. The training process was done using TensorFlow and Keras and was carried out using a RTX3060 Nvidia GPU and 32GB of RAM.

4. ANALYSIS OF RESULTS

An experiment was carried out using the testing dataset of PFD to evaluate the performance of the profile face detection model. The testing dataset consists of 405 PFV images. The performance metrics that are used for the evaluation are accuracy, precision, and f1-score, as shown for equations (1), (2), and (4), respectively. Accuracy is the basic performance metric that evaluates the model's accuracy in terms of detecting the PFV correctly. Precision is used to measure the model's accuracy in detecting the PFV as the positive sample. Meanwhile, f1-score is an evaluation metric that measures the model's accuracy on the testing dataset.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (1)$$

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

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

$$F1_{score} = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (4)$$

Based on the test result, the proposed method has managed to achieve a good detection accuracy of 98.52% on the testing dataset, while the precision and f1-score are 100% and 99.25%, respectively. Furthermore, the total time required to process an image has an average duration of 0.30 seconds. This is a good indicator that the proposed model has the capability to detect the PFV accurately and efficiently without any major issues. Figure 6 shows the sample result of the detected PFV for four different participants. Furthermore, even with minor occlusions, such as wearing glasses or a slight occlusion due to hair, the proposed model is capable of detecting the PFV.

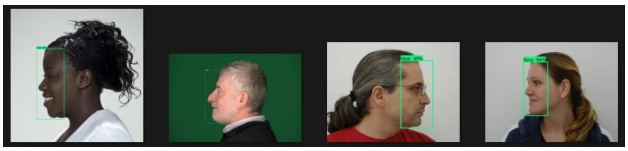

Figure 6. Sample detected PFV image in the testing dataset.

Table 1 shows the comparative analysis of four prominent models in the field with the proposed works. The three models are MTCNN, the OpenCV Viola-Jones algorithm, the DLIB HOG face detection model [28], and the DLIB face detection CNN model [29]. It is evident that the proposed model emerges as one of the best performers. Using the performance metrics, the proposed model consistently outperforms other models, establishing itself as the best solution for detecting profile faces. In addition, based on Table 4, it shows that the traditional method that uses HOG and the Viola-Jones algorithm has the worst performances, as the hand-crafted features are extremely weak against extreme face poses. This further proves the point made in the literature review, as the traditional method focuses more on frontal face detection. MTCNN and DLIB CNN models perform well, showing the robustness of CNN architecture in detecting different face poses, but do not perform as well as the proposed model. The average time taken for MTCNN to detect PFV images is approximately 35 milliseconds, while the average time taken for DLIB CNN models is approximately 3.5 seconds to detect PFV images. Not only is the proposed method fast, but it also produces better accuracy in detecting PFV images.

Table 1. Comparative analysis of five different face detection models

	Proposed Model	MTCNN	DLIB CNN model	DLIB HOG model	OpenCV Viola Jones
Accuracy	98.52%	91.89%	92.86%	21.73%	2.16%
Precision	100.00%	96.49%	96.30%	100.00%	0.2%
F1-Score	0.9925	0.9577	0.9630	0.3570	0.0423

5. CONCLUSION

In this paper, it is highlighted that a profile face is an extreme face pose that is hard to detect using a normal face detection algorithm. This is due to a lack of facial feature data for the algorithm to recognize it as a face. Therefore, to solve this problem, this paper has proposed the use of SSD MobileNetV2 with FPN in the profile face detection model. The proposed model used the combination of three different models as a strategy to solve the issue of detecting the profile face view. The results of the experiment show that the proposed model has achieved good performance in terms of accuracy and precision. The model has achieved an accuracy of 98.52% against the testing dataset while having a precision of 100%. The experiment finds that the SSD MobileNetV2 in the proposed face detection model can effectively detect profile faces in images. For future works, the capability of the proposed model needed to be expanded to cover other related face detection challenges such as occlusion.

ACKNOWLEDGMENT

The authors would like to thank University Tun Hussein Onn Malaysia for providing grant RE-GG (H884) funding and Advanced Telecommunication Centre (ATRC) for supporting this research.

REFERENCES

[1] Kumar, A., Kumar, M. & Kaur, A. Face detection in still images under occlusion and non-uniform illumination. Multimed Tools Appl, 2021. 80: 14565–14590.
[2] Minaee, S., Luo, P., Lin, Z. and Bowyer, K., 2021. Going deeper into face detection: A survey. arXiv:2103.14983.
[3] Mohammed OA, Al-Tuwaijari JM, Analysis of challenges and methods for face detection systems: A survey. International Journal of Nonlinear Analysis and Applications. 2022.
[4] 13(1):3997-4015.
[5] Hao Z, Shi H. 2021 Small Face Detection Based On Feature Fusion. J. of Phys: Conf. Ser. 1871
[6] 012086
[7] Viola P, Jones M., Rapid object detection using a boosted cascade of simple features. IEEE Computer Society Conference on Computer Vision and Pattern Recognition 1: pp. I-I.
[8] Junaidy D, Wulandari M, Tanudjaja H Real time face detection using haar-like feature method and
[9] local binary pattern method," IOP Conf Ser Mater Sci Eng 508, 012076
[10] Dalal, N. and Triggs, B., Histograms of oriented gradients for human detection. IEEE computer society conference on computer vision and pattern recognition 1:886-893
[11] Thakur, S., Paul, S., Mondal, A., Das, S. and Abraham, A., December. Face detection using skin tone segmentation. World Congress on Information and Communication Technologies, 2011. 53-6
[12] Ning, Z. and Xutao, G., Face detection based on skin color extraction scheme. IOP Conf Ser Mater Sci Eng. 569(3) 032006
[13] Cootes, T.F., Edwards, G.J. and Taylor, C.J., 2001. Active appearance models. IEEE Transactions on pattern analysis and machine intelligence, 23(6) 681-685.
[14] Zhang, K., Zhang, Z., Li, Z. and Qiao, Y., Joint face detection and alignment using multitask cascaded convolutional networks. IEEE signal processing letters, 2016. 23(10): 1499-1503.
[15] Zhu, Y., Cai, H., Zhang, S., Wang, C. and Xiong, Y. 2020. Tinaface: Strong but simple baseline for face detection. arXiv:2011.13183.
[16] Jiang, C., Ma, H. and Li, L., July. IRNet: An Improved RetinaNet Model for Face Detection. 7th International Conference on Image, Vision, and Computing (ICIVC), 2022. 129-134.
[17] Qi, D., Tan, W., Yao, Q. and Liu, J., YOLO5Face: Why reinventing a face detector. Computer Vision–ECCV 2022 Workshops, 2022. 228-244
[18] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L.C., Mobilenetv2: Inverted residuals and linear bottlenecks. Proceedings of the IEEE conference on computer vision and pattern recognition, 2018. 4510-4520.
[19] Simonyan, K. and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556.
[20] He, K., Zhang, X., Ren, S. and Sun, J., Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016. 770-778.
[21] Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B. and Belongie, S., Feature pyramid networks for object detection. Proceedings of the IEEE conference on computer vision and pattern recognition, 2017. 2117-2125.
[22] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y. and Berg, A.C., 2016. SSD: Single shot multibox detector. Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 2016. 14: 21-37.
[23] Thomaz, C.E. and Giraldi, G.A., A new ranking method for principal components analysis and its application to face image analysis. Image and vision computing, 2010. 28(6): 902-913.
[24] DeBruine, Lisa Marie and Benedict C. Jones. Face Research Lab London Set. 2017.

[25] Khan, G., Samyan, S., Khan, M.U.G., Shahid, M., and Wahla, S.Q., A survey on analysis of human faces and facial expressions datasets. International Journal of Machine Learning and Cybernetics, 2020. 11: 553-571.
[26] Vieira, T.F., Bottino, A., Laurentini, A. and De Simone, M., Detecting siblings in image pairs. The Visual Computer, 2014. 30: 1333-1345.
[27] Shorten, C. and Khoshgoftaar, T.M., A survey on image data augmentation for deep learning. Journal of big data, 2019. 6(1): 1-48.
[28] Yang, S., Xiao, W., Zhang, M., Guo, S., Zhao, J. and Shen, F. 2022. Image data augmentation for deep learning: A survey. arXiv:2204.08610.
[29] Shorten, C., Khoshgoftaar, T.M. and Furth, B., Text data augmentation for deep learning. Journal of big Data, 2021. 8: 1-34.
[30] Yu, T. and Zhu, H., 2020. Hyper-parameter optimization: A review of algorithms and applications. arXiv:2003.05689.
[31] King, D.E., Dlib-ml: A machine learning toolkit. The Journal of Machine Learning Research, 2009. 10:1755-1758.
[32] Alvarez Casado, C. and Bordallo Lopez, M., Real-time face alignment: evaluation methods, training strategies and implementation optimization. Journal of Real-Time Image Processing, 2021. 18(6): 2239-2267.