Science Journal of University of Zakho
DOI: https://doi.org/10.25271/sjuoz

Original Article

FACE DETECTION USING REFINED-RETINAFACE MODEL

Sagvan J. M. Ameen[1, *] **and Ahmed AK. Tahir**[1]

[1]Department of Computer Science, College of Science, University of Duhok, Kurdistan Region, Iraq

[*] Corresponding author. E-mail: sagvan.jm3722@stu.uod.ac (Tel.: +964-7507924600)

Received: 28, Sep, 2025
Accepted: 08, Des, 2025
Published: 01,Jan,2026

ABSTRACT

RetinaFace is a multi-task and single-stage face detection model that detects faces and landmarks. However, it has limitations in detecting non-face content in output bounding boxes and mislocalizes facial landmarks for profile faces. To address these issues, Refined-RetinaFace (R-RetinaFace) is proposed. R-RetinaFace adds a post-optimization module that resizes bounding boxes and ensures all landmarks are within them. R-RetinaFace outperforms RetinaFace on SDUMLA-HMT and CASIA-3D-FaceV1 databases. On SDUMLA-HMT, R-RetinaFace achieves an ideal detection rate of 98.02%, a moderate detection rate of 1.32%, and a poor detection rate of 0.66%. On CASIA-3D-FaceV1, R-RetinaFace achieves ideal detection rates of 92.2%, moderate detection rates of 7%, and poor detection rates of 0.8%. In contrast, RetinaFace did not achieve ideal detection on both databases. It achieved only moderate and poor detection rates. On SDUMLA-HMT, RetinaFace achieves a moderate detection rate of 96.32% and a poor detection rate of 3.68%. On CASIA-3D-FaceV1, RetinaFace achieves a moderate detection rate of 83.9% and a poor detection rate of 16.1%. These results put R-RetinaFace a state-of-the-art method for face detection.

KEYWORDS: Face detection, RetinaFace, ResNet50, Deep Learning, Machine Learning, Skin Color Detection.

1. INTRODUCTION

Face detection is the process of identifying human faces in images or videos and is a fundamental task in computer vision. It is considered a necessary step for many face-related applications, including biometric-based security, face modeling, head-posture tracking, age and gender recognition, face expression recognition, and human-computer interaction (Kumar *et al*., 2019; Hasan *et al.,* 2021; Hasan & Mstafa, 2022; Tahir & Anghelus, 2024).

There are many face detection methods; some are useful for gray-scale images, some for colored images, some for-crowd images, and some for single-person images. In other words, one method may produce better results when applied to gray-scale images rather than colored images and vice versa. Succinctly, a low-accuracy model of face detection, which is an initial step for a face recognition system, will lead the whole face recognition system to a higher error rate and vice versa (Kumar *et al*., 2019; Hasan *et al*., 2021; Minaee *et al*., 2021).

The first recorded face detection method was developed by Sakai *et al.* (1972). They developed an algorithm to detect face and localize facial features such as eyes, nose, and mouth. It was a rule-based system designed to detect faces in gray-scale images. Then in the early 1990s, the researchers began to focus on handcrafted features, including edge detection, template matching, and skin color models (Dengi *et al*., 2024).

In general, face detection can be implemented by either a traditional approach or a learning-based approach. The traditional approach is early, using handcrafted features and rule-based algorithms to detect faces in images. This approach does not require pre-learning of the dataset, but it detects face area using predetermined patterns, statistical models, or geometrical relationships (Dengi *et al*., 2024; Zhang *et al*., 2019). A learning-based approach, including machine and deep learning, trains models on datasets to automatically locate faces in images. Machine learning techniques usually use predetermined features by handcrafted methods to train the model for classification. Examples of machine learning methods are; Support Vector Machine (SVM), Decision trees, Random-Forest, KNN, etc. While, deep learning especially, the Convolutional Neural Networks (CNN) learns features directly from images for classification. Learning-based approach provides more success rate and flexibility to challenging environments such as various illuminations or

Access this article online
https://doi.org/10.25271/sjuoz.2026.14.1.1662
Science Journal of University of Zakho
Vol. 14, No. 01, pp. 1 –15 January-2026
Printed ISSN 2663-628X;
Electronic ISSN 2663-6298

lightning, different poses, and different face expressions (Kumar *et al*., 2019; Dengi *et al*., 2024).

Despite the developments that have taken place in the field of face detection, there are still some challenges facing the process of face detection, including pose variation, odd expression, face occlusion, and variation in illumination level (Minaee *et al*., 2021; Soni *et al*., 2023; Dengi *et al*., 2024; Thaher *et al*., 2025). In this paper, the RetinaFace model, which is a type of deep learning approach, is refined and employed for face detection. The RetinaFace model has demonstrated efficiency in overcoming the aforementioned challenges. However, one of its limitations is that it may include non-face contents within bounding boxes and exclude crucial facial landmarks, such as the nose tip, when dealing with profile faces. To mitigate this issue in this paper, a new module named Refined-RetinaFace (hereafter, R-RetinaFace) is integrated into the conventional RetinaFace model. The aim of adding this module is to enhance face detection accuracy by resizing the bounding boxes generated by RetinaFace with precise localization of the face boundary and facial landmarks.

The main contribution of this work can be summarized as:

1. Refining RetinaFace with a post-optimizer module to enhance face detection.
2. Utilizing facial landmarks provided by the landmark localization loss-function of the RetinaFace eliminates the need for extra information.
3. Demonstrating the model's flexibility through successful application on two different databases, 2D (SDUMLA-HMT) and 3D (CASIA-3D-FaceV1).

The remaining parts of this paper include: Approaches and methods of face detection are introduced in section 2, the section on related works is provided in section 3, the proposed method is explained in section 4, databases are given in section 5, results and discussion are highlighted in section 6, and finally, the concluding remarks are provided in section 7.

Approaches of Face Detection:

Face detection can be implemented using two primary approaches: traditional and learning approaches. The traditional approach relies on handcrafted feature methods and a simple classifier to detect faces. It is based on extracting invariant facial features such as eyes, nose, mouth, eyebrows, and skin color. However, the major problem of feature-based algorithms is that they are less adaptable to changes in pose variation, changes in lighting conditions, and the image features can be severely affected by noise, shadows, and occlusion. While learning-based approaches (machine and deep learning) can automatically acquire features from data (images), hence providing more accuracy and reliability. Therefore, learning-based approaches are more resistant to face detection challenges such as pose variation, change in lighting conditions, and occlusion, and have shown superior performance and robustness in face detection tasks, especially in challenging conditions (Dengi *et al*., 2024). The following subsections provide the most commonly used methods in each approach.

Methods of Traditional Approach:

Traditional methods are based on extracting specific features from faces, such as edges, color, corners, and texture patterns. Generally, they are less efficient compared to machine and deep learning methods and sensitive to variations in lighting, pose and facial expression (Thaher *et al*., 2025). Several methods for traditional approach have already been developed. The most commonly used of them include, skin color detection, eigenfaces, template matching, etc., (Maw *et al*., 2018; Yousif *et al*., 2024).

1) Skin color detection: The skin color detection is a technique of finding and isolating areas in images based on normal ranges of human skin color in a particular color format such as RGB, HSV, or YCbCr. It helps by ignoring pixels in image that does not contain human skin. It is usually used with another face detection method to improve the performance of face detection process. This method is simple, fast, and works well under specific (controlled) lighting conditions. While in some circumstances such as illumination variation and the existence of objects and background near to skin color will lead the detection process to failure (Kumar, 2014).

2) Eigenfaces: Eigenfaces is a face detection method that uses PCA to locate faces by projecting image patches into lower-dimensional space made-up of eigenfaces, which are the main components of a set of face images. These eigenfaces represent the principal components of the variation within a set of training face images. The system projects new face images onto this "face space" and compares them to known individuals based on their projection vectors (Yazdani and Shojaeifard, 2023). The main advantage of this approach is that it has low sensitivity to noise and the information needed to identify the person is reduced by a great percentage, so it is efficient for real-time applications. However, this method is not efficient when there are variations in pose, expression, illumination, scale, occlusion and it provides good results only for front-view images (Çarıkçı and Özen, 2012). In addition, eigenfaces heavily relies on the assumption that facial images lie in a linear subspace, which may not always hold in real-world scenarios (HO *et al*., 2024).

3) Template matching: Template matching is a conventional method for object detection especially facial features detection. It is used to measure the similarities between two images. The matching can be based on either features, or area or any similarity measure between the template and the pre-defined pattern. When used for face detection, the template contains facial features such as mouth, eyes, and nose (Boss *et al*., 2020). Template of the facial features has also been used with skin color information for face detection (Yuen *et al*., 2009).

Methods of Machine Learning:

Machine learning methods learn features and patterns from features extracted from dataset. Examples of machine learning techniques include the Viola-Jones algorithm (Haar cascades), Support Vector Machine (SVM), Random Forest (RF), and K-Nearest Neighbor (KNN), etc. (Minaee *et al*., 2021; Dengi & Patil, 2024; Hassen & Naser, 2024). In general, machine-learning methods can be effective for face detection, but they may not perform as well as deep learning methods in some instances, especially with pose

DOI: https://doi.org/10.25271/sjuoz.2026.14.1.1662

variation. Descriptions of the most common methods of machine learning are given below:

1) Viola Jones: The Viola Jones is a face detection method introduced by Viola and Jones (2001), which can be used for real time systems. It is considered as one of the hybrid approaches combining traditional method (Haar-like features and integral images) with machine-learning methods (AdaBoost and cascade classifier). It scans the image quickly by moving a window over it and uses an integrated image to speed up feature detection. It has several benefits, including working in real time, ease of use, and being quite accurate for detecting faces in frontal face images. However, it may not be efficient to detect faces in cases when the image is a non-frontal face, occlusion, poor lighting, and small faces (Viola & Jones, 2001; Viola & Jones, 2004; Pal, 2020; Saputra *et al*., 2025). In addition, it deals with gray images.

2) Support Vector Machine (SVM): This classifier learns to categorize image characteristics (pixel values, texture) into face and non-face classes. These characteristics are usually extracted using techniques like HOG or pixel intensity patterns to extract feature from the image to classify them as face or none-face using a pre-defined hyperplane (Hasan, 2022; Kokare and Ghisare, 2025). It is effective when there is a small dataset with low dimension (Kukenys and Mccane, 2008).

3) Random Forest (RF): This is an ensemble learning method that can be used for face detection by training multiple decision trees on features extracted from face and non-face images (Kremic and Subasi, 2015; Mady and Hilles, 2018). First, features such as HoG, LBP, Gabor filter, etc., which can differentiate between faces and other image parts, are extracted from the images. Second, these features are divided into subsets and each tree in the forest is trained on a random subset of the training data and learns to classify image as either "face" or "not face. Finally, face detection is done in the testing phase by a moving window over the test image and the random classifier predicts whether the window contains face or non-face parts. The efficiency of random forest depends on the robustness of feature extraction method. It is somehow robust to variations in lighting, pose, and expression. The testing phase of random forest can be fast but computationally intensive during the training phase (Mala and Mohammad, 2022).

4) K-Nearest Neighbors (KNN): This classifier is based on extracting feature vectors from face and none-face images such as pixel intensity, Haar-like features, or HOG features and store them as enrolled database. KNN model works by extracting feature vector from the testing image and calculating Euclidean or Manhattan distance between the testing image and those of the enrolled images. The label of the testing image whether it is a face image or none-face image is determined by the majority vote of the K nearest neighbors. KNN is conceptually easy, does not need training and it is adaptable whenever new images are added. However, it takes long computation time, especially for large database and not efficient for detecting profile faces (Guo, 2021; Wirdiani *et al*., 2019).

5) OpenCV: A popular computer vision library with pre-trained Haar cascade classifiers for face detection (Madan, 2021). It is lightweight but less accurate than deep learning-based models.

Methods of Deep Learning:

Methods of deep learning approach are mainly based on Convolutional Neural Network (CNN), which learn features and patterns with different scales from relatively huge dataset. A comprehensive study by (Thaher *et al*., 2025) revealed that deep learning methods dominate recent studies, benefiting from their ability to extract detailed features and handle complex patterns, specifically, after the emergence of the transfer-learning architectures. Examples of deep learning models for face detection are RetinaFace, You-Only-Look-Once (YOLO), Multi-Task Cascaded Convolutional Neural Network (MTCNN), Single-Shot Detector (SSD), etc. (Minaee *et al*., 2021; Dengi & Patil, 2024). Descriptions of the most common models for deep learning are given below:

1) RetinaFace: Is a deep learning model that works for face detection and facial landmarks localization in a single forward pass. It is a modified version of the RetinaNet framework, which was developed by Facebook AI Research in 2017 to detect objects in images (Yousif *et al*., 2024). In the RetinaFace model, some face-specific features were added, including more supervision for landmarks, dense regression for both bounding boxes and landmarks, and context modules to better capture facial characteristics. It uses ResNet50 as a backbone for extracting features from the image and creates feature maps to be used by Feature Pyramid Network (FPN) to create Multiscale feature maps (Deng *et al*., 2019; Deng *et al*., 2020; Liu and Yu, 2023). Then, the Context Module (CM) takes these multiscale feature maps and adds the surrounding information of the image to make the face detection process more robust against face detection challenges. Using bounding boxes and classification confidence scores on the outputs of the previous step, a face can be detected (Ponnmoli & Pandian, 2025). Compared to other deep learning models, the RetinaFace model produces accurate results even under some face detection challenges such as pose variation, expression variation, occlusions, complex background, and illumination variation. RetinaFace can also use MobileNet as a backbone. However, with ResNet50, it achieves a better balance between accuracy and computational cost.

2) Multi-Task Cascaded Convolutional Neural Network (MTCNN): Is a deep learning-based face detector that uses a cascade structure with more than one network to detect faces and facial landmarks. MTCNN is popular for its state-of-the-art performance on benchmark datasets and its ability to recognize facial features such as eyes and mouths (Zhang *et al*., 2020; Hassan *et al*., 2025). It is a robust face detection challenge, such as variations in face size, lighting, and rotation. Like RetinaFace, it detects faces and localizes face landmarks simultaneously, but may not be as accurate as RetinaFace with large pose variations or occlusions, despite being computationally less intensive than RetinaFace.

3) You-Only-Look-Once (YOLO): YOLO is a deep learning-based model used to detect faces in real time by dividing an image into grids and using neural network to predict bounding boxes and confidence score for each grid

DOI: https://doi.org/10.25271/sjuoz.2026.14.1.1662

in a single forward pass. YOLO was primarily designed for object detection, but it can be used for face detection after fine-tuning. It performs well for normal size images, but has limitations when it deals with images of small size. In addition, it is not robust to variation in image scale and does not localize facial landmarks. It uses image data to learn spatial patterns and features in order to recognize faces (Ponnmoli & Pandian, 2025). Several versions of YOLO have been developed, such as YOLOv3, to improve the performance of the model (Chen et al., 2020). In general, YOLO is not as accurate as RetinaFace and MTCNN for face detection.

4) Single-Shot Detector (SSD): A single-shot detector is mainly an object detector method, but can be used for face detection after some modifications. The main reasons of using SSD for face detection are that the SSD provides Real-time performance and it is a multi-scale face detector. SSD detects face from a single input image. The model consists of Siamese networks that learn similarity metrics between faces (Thakurdesai *et al*., 2018; Ye *et al*., 2021). Using only one input image for each face in the training represents challenges for traditional CNNs, which require a large number of training examples for each face and as a result, it is less robust to variations in face-expression, lighting and pose. It provides good detection performance but average alignment scores. However, computationally, it is faster than RetinaFace and MTCNN.

RELATED WORKS:

Numerous studies have already been conducted on face detection using both approaches, traditional and learning-based. The most impactful of these studies are given in the following subsections.

Traditional Approach:

Paul and Garvilova (2011) proposed Principal Component Analysis (PCA) and Skin Color Modeling (SCM), in which the output of SCM is used as input to PCA in the face detection process. The goal of using a fusion of both methods is to overcome illumination variation and reduce noise. The system was implemented on four different databases: CIT, BaoFace, Essex, and Georgia Tech. The achieved accuracies were 98.7% for CIT, 97.1% for BaoFace, 97.1% for Essex, and 96.7% for Georgia Tech. (Tripathi *et al* 2011) Combined skin color detector based on YCbCr color model with template matching method. First, a skin color detector was used to detect face and non-face regions, then a template matching method was used to remove non-face areas and detect faces more accurately. Experimental results show that the proposed method outperforms the skin color detector alone. Jabbar *et al*., (2018) used a combination of color segmentation and template matching for detecting faces in multi-face images. The achieved accuracy was between 70% and 80% depending on the combined methods. Hajraoui and Sabri (2014) proposed a model for face detection based on skin color. The model consisted of two main modules. The first was for image segmentation to retrieve a significant region, and the second was for classification of skin regions into face and non-face regions. In the segmentation module, image pixels were classified into two classes (skin / non-skin), producing a binary image, which was segmented by the watershed technique to produce a connected and consistent region, followed by extracting the significant area for the skin region to be classified via the classification module using a cascade Gabor filter. Hajraoui and Sabri claimed that their model achieved good performance when tested on two databases: Caltech_10k_webfaces and 200 webfaces. However, their model did not localize the face landmarks, which are very important for reducing the false face detection rate and for implementing face recognition.

Zhang *et al*. (2017) introduced a real-time face detection and recognition model. They used a combination of Ada-Boost, cascade classifier, Local Binary Pattern (LBP), Haar-Like features, facial image processing and PCA. The Ada boost algorithm was used to train the face and eye detection in the cascade classifier. While LBP was used to extract facial features and finally PCA was used for face recognition. Despite its operation in real-time and achieving true positive rate of 98.8%, only the eyes among the face landmarks were detected. In addition, the results of face detection, as mentioned in the paper, shows that the detected face include some of non-face areas. Maw *et al*., (2018) used a combination of skin color detector using YCbCr color format and Viola-jones face detector to detect faces. Skin color was used to reduce the false positive rate and to speed up the model, while Viola-Jones algorithm was used to improve the speed of the processes and to detect faces. They achieved an accuracy of 86.55% on an in-house database consisting of 30 images of varying lighting conditions and complex background. However, Maw and others' model work good only for frontal images. Ochango, (2023) extracted eigenfaces from the components of PCA for face detection, using 104 face images, 60 images for training and 44 images for testing. They generated the eigenvectors of the covariance matrix of all images during the training set and sorted them by descending magnitude. Then, the top eight were selected as resembling faces, since they show the unchanged features of the face better than the trailing components. For testing, they normalized the testing image by subtracting the mean face and compared with the PCA components in the training dataset. They achieved an accuracy of 86.36%.

Machine Learning Approach:

Machine learning approach is a powerful approach in object detection in general and face detection in specific. Tsai *et al*., (2006) proposed a face detection system by cascading Eigenfaces, Back-propagation neural network and a simple face verification scheme. First, PCA was used to extract eigenfaces from which the candidates of the face region were extracted. Then, a neural network examined these candidates for face or non-face region. Finally, a template-based face verification method is used to confirm each face region from the output of the neural network. The role of Back-propagation algorithm was to process the output of Eigenfaces to improve the accuracy of face detection. The network was trained on face blocks and non-face blocks including frontal and profile faces chosen

DOI: https://doi.org/10.25271/sjuoz.2026.14.1.1662

from the ORL, the MIT CMU and the Wide World Web face datasets. For testing the system, three databases were used, the BiolD, the ORL and the World Wide Web. They achieved an accuracy of 96.38%. Cerna *et al*., (2013) propose an efficient approach to discriminate face from non-face images using a combination of HOG, vector quantization and SVM with a linear kernel as classifier. Their approach was robust to challenges of face detection such as variations in pose, illumination, and occlusion. First, HOG descriptors were extracted within regular grids in the image. Then, the extracted descriptors were vector quantized to generate codebooks using Bag-of-Feature method. Finally, these codebooks were driven to SVM classifier for learning a model to classify the image as face or non-face. Cerna and others' approach was trained on 2385 face images and 7025 non-face images, collected from AT&T Databases. The obtained accuracies were subject to the codebook size and the number of images. However, the approach was restricted to frontal images and the testing results showed that the detected face contained some parts of non-face areas.

Kremic & Subasi (2015) compared the performance of Random Forest with that of SVM for face detection using International Burch University (IBU) image databases, which consists of 20 single image face per person of size 205 x 274 with different facial expression. The SVM achieved accuracy of 93.20% versus 97.17% for Random forest. However, when optimizers with different kernel were used, the achieved accuracy of SVM increased to 95.89%, 96.92%, 97.94%. They used, skin color detection, RGB to gray and image histogram for feature extraction. Wirdiani, *et al*., (2019) developed face identification system using a combination of PCA and KNN. First, contrast stretch method was applied to enhance the images then, a Haar cascade segmentation was used for segmentation followed by PCA for feature extraction. The developed system applied to a database containing 150 images from 30 subjects with 60% for training and 40% for testing. The result obtained from several tests of K value gave 81% as the best F1-score with K = 1. However, using only 150 images is not wise when KNN is used as a classifier because KNN can provide better results for large database. Al-Dabbas *et al*., (2023) Tested three machine learning classifiers, J48, OneR, and JRip on MUCT database for face detection. They applied Linear Discriminant Analysis (LDA) for feature extraction. Their results indicated that the J48 classifier with LDA achieves the highest performance with 96.0001% accuracy. Komlavi *et al*., (2024) compared Support Vector Machines (SVM), Artificial Neural Networks (ANN), KNN, Random Forests (RF), Logistic Regression (LR) and Naive Bayesian Classification using ORL and YALE image databases. The best accuracy amongst the machine learning models was achieved by SVM with an accuracy of 98.19%. Saputra *et al*., (2025) used Viola-Jones method for face detection to be used in a graphical user interface (GUI) system created with Matlab. They tested the system on fifteen single-face and multi-face images randomly chosen from the websites. They obtained an average accuracy of 89.86%. However, they noted that the system was not efficient for occluded faces, non-frontal faces. They suggested using advanced preprocessing techniques or algorithms of machine learning approach.

Deep Learning Approach:

The use of deep learning methods that are based on convolutional neural networks has greatly improved the performance of face detection. Jiang and Learned-Miller, (2017) proposed a face detection system based on Faster Region Convolutional Neural Network (Faster R-CNN). Despite R-CNN's use for object detection, it can also be adapted for face detection. In addition to its speed, Faster R-CNN can produce good results for occluded face, small faces and tilted faces. Jiang and Miller's system achieved true positive rate of 95.2% when implemented on WIDERFACE dataset. Wang *et al*., (2017) used Face Attention Network (FAN) algorithm to detect faces with a partial occlusion caused by hat, glasses, mask, and hair. They selected only 16% of WIDERFACE database, which represents the occluded face. They divided the database into three subsets, easy to detect, medium to detect, and hard to detect. They obtained accuracies of 94.6% for easy subset, 93.6% for medium subset, and 88.5% for hard subset. Ye *et al*., (2021) proposed Single Shot multi-box Detector (SSD) system for detecting tiny faces in images. Their use of SSD was based on two arguments: first, SSD provides Real-time performance, second, it is a multi-scale face detector. The proposed system was implemented on two datasets including FDDB and WIDERFACE achieving accuracy of 93.7% and 82.6%. The clear difference between the two accuracies is because WIDERFACE contains single-face and multi-face images with variation in lighting and poses, occlusions, while FDDB images are mostly frontal face images. Recently, the use of RetinaFace model in face detection brought the attention of many researchers. Deng *et al*., (2020) used RetinaFace, which is a single-stage face detector model for detecting faces and localizing face landmarks (eyes, nose tip and mouth corners). Their use of RetinaFace model was based on its efficiency in handling face images with variations in pose, scale, and illumination with complex background. In addition, RetinaFace can be employed to detect 3D faces. After extensive experiments, they concluded that RetinaFace could detect faces and localize the face landmarks efficiently for images with different pose and illumination.

Xue *et al*. (2020) also used RetinaFace model for both, masked-face detection and recognition. The system was implemented on WIDERFACE and MAFA datasets. The authors claimed that the system achieved good performance without mentioning any of the numerical results. Liu and Yu, (2023) used RetinaFace with MobileNetV3, instead of ResNet50 as a backbone for feature extraction. The model was tested on WIDERFACE dataset using three image subsets, easy, medium, and hard. The average accuracy for easy subset was (94.1%, for medium subset was 92.2% and for hard subset was 82.1%. Ren *et al*., (2025) proposed a combination of RetinaFace and AdaFace for face detection to overcome some face detection challenges such as occlusion, low-resolution images, and small faces. The proposed system achieved 96.12% accuracy when tested on the WIDERFACE dataset. Hangaragi *et al*., (2023) proposed face detection and recognition system based on face mesh and deep

DOI: https://doi.org/10.25271/sjuoz.2026.14.1.1662

neural network. Their system was designed to operate under varying illumination and background and to handle non-front images. Hangaragi and others' system achieved an accuracy of 94.23% on Labeled WILD-Face (LWF) database. Xiong *et al.,* (2023) used a RetinaNet to overcome the problem of rotation variation in face detection. They incorporated a contextual module and added a new head module to the multi-task head for the regression task of facial landmarks. Xiong and others' system consist of four parts instead of three parts as in RetinaFace. The first three parts act exactly as the RetinaFace. The fourth part, the Multi-Task Head module performs the subsequent classification and regression task to acquire the final output. The system achieved a face detection accuracy of 92.15% when experimented on FDDB database. The Multi-Task Cascade CNN (MTCNN) is one of the competitive model for face detection.

Zhang *et al*. (2020) used MTCNN for face detection and facial landmarks localization in multi-scale and occluded face images. In MTCNN, a cascade structure with more than one network are used. The system and achieved an accuracy of 85.7% on WIDERFACE dataset. Hassan *et al*., (2025) used almost the same system with DenseNet, but it was tested on Labeled Faces in the Wild (LFW) dataset, and achieved an accuracy of 96.64%. Qi *et al*., (2022) have used You-Only-Look-Once (YOLOv5) for face detection in real time for mobile application. YOLO is a single-stage face detector that treats the detection as a regression problem. They added a five-point landmark regression head with Wing loss function. They designed models with different size, large to super small to cope with database of small, medium and large subsets. The application of their system on the WIDERFACE dataset achieved state-of-the-art performance in all subsets. Gao *et al*., (2024) introduced a face detection model named Deep and Compact Face Detection (DCFD), which adopts an improved lightweight EfficientNetV2 network to replace the backbone network of RetinaFace. In addition, they used the focus loss function to replace the traditional cross-entropy loss function to balance the training process of positive and negative samples. They tested their system on used WiderFace dataset and LFW dataset using subsets, easy subset, medium subset and hard subset. With WiderFace database, the achieved averaged precisions were 96.64 for easy subset, 96.3% for medium subset and 96.73% for hard subset. With LFW dataset, the achieved averaged precisions were 97.04% for easy subset, 96.43% for medium subset and 87.01% for hard subset. Praveen *et al*., (2025) deigned an automated attendance marking system based on face detection for a classroom. They employed a collection of Convolutional Neural Networks (CNN), ResNet, and Histogram of Oriented Gradients (HOG). HOG was used for image enhancement. While CNN was used for feature extraction and ResNet was used for face detection. They tested the system under three different conditions, control lighting, low lighting and varying angles. They achieved accuracies 93%, 91% and 92% respectively for the three conditions.

THE PROPOSED MODEL FOR FACE DETECTION (R-RETINAFACE)

In this paper, a deep learning model is adopted for face detection, since deep learning proven to perform better compared to traditional and machine learning approaches (Ponnmoli and Pandian 2025). The blockdiagram of the proposed model is shown in Figure (1).

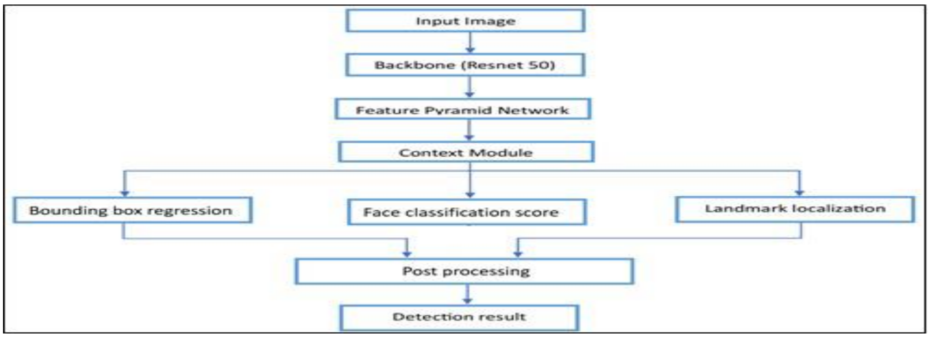

Figure 1: The Blockdiagram of the proposed model

.

Given that the proposed model of face detection is based on post-refining the output of the RetinaFace model, it is worthwhile first to provide a detailed description of RetinaFace and R-RetinaFace models in the following subsections.

RetinaFace Model Architecture:

RetinaFace is a multi-task deep learning-based face detector that performs face detection and facial landmark localization in one forward pass. It provides good results for face detection in challenging conditions such as occlusions and variation in pose. Its unique property is that it can detect small faces in crowded environments (Ponnmoli and Pandian 2025). In addition, RetinaFace is a pre-trained model on WIDERFACE dataset, which means that it can benefit from an old training and does not require to train again (Deng *et al*., 2019; Xue *et al*., 2020). WIDERFACE dataset is a well-known universal dataset containing over 32,203 images with single and multi-face and a total of 393,703 marked faces with variations in pose, illumination and accessories (Yang *et al*., 2016). This makes RetinaFace model flexible and suitable for

DOI: https://doi.org/10.25271/sjuoz.2026.14.1.1662

detecting faces under different conditions. Figure (2) shows the structure of RetinaFace model and description of its components are given below:

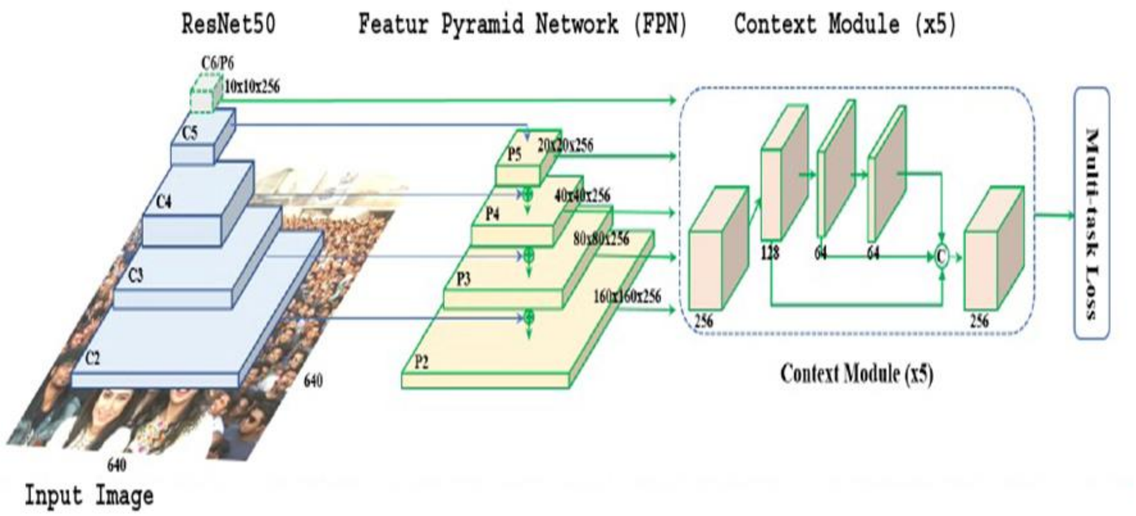

Figure 2: The Structure of RetinaFace Model (Deng *et al*., 2019).

Input image: A digital image of any size, either colored or gray images.

Backbone (ResNet50) is the main feature extractor that takes an input image and makes hierarchical feature maps from it. RetinaFace employs the ResNet50 as its base. ResNet50 is a deep CNN with 50 layers where the last layer is not used in RetinaFace, because it is related to image classification tasks. ResNet50 uses residual learning through skip connections to make it easier to train very deep networks. In this paper, RetinaFace library of python is used. This library provides a pre-trained ResNet50 on ImageNet dataset. For some applications, MobileNetV1 is used instead of RESNet50.

Feature Pyramid Network (FPN): This component operates on the hierarchical feature maps generated by backbone. FPN is a crucial part in making RetinaFace more robust to scale variation by improving multi-scale feature representation. It builds high-level semantic feature maps at multiple scales by making a top-down structure. This lets the detector find faces of different sizes by mixing high-resolution features from earlier layers with semantically strong features from later layers

Context Module: Designed to improve the feature representation by getting information about the context around each pixel in the feature maps, both locally and globally. It does this by using a number of convolutional layers with varying receptive fields. This helps the network to capture the spatial information around and makes the face detection more accurate in case of small or occluded faces. Therefore, the module helps the network to focus on important information and block out background noise by combining multi-scale context. This step is essential for accurate face detection and landmark localization.

Multi-Task Loss: includes three loss functions for bounding-box regression, face classification score, and landmark localization.

1) Face classification score: it is actually the level of confidence given to each detected area, which shows how likely this area contains a face. Using a binary classification system, this score helps to tell the difference between real faces and background or non-face areas.

2) Bounding-box regression: The process of estimating the exact coordinates of the face bounding-box by adjusting default anchor boxes in order to match the actual position of the face in the image. This is done by figuring out the difference between the anchor boxes, which are used by the retina to detect face, and the actual face boxes.

3) Landmark localization: this is used to figure out the coordinates of important facial landmarks such as eyes, nose tips, and mouth corners. It provides five points: two points in the middle of each eye, one point on the nose tip, and two points at mouth corners. As mentioned before, RetinaFace is pre-trained on WIDERFACE dataset so it is able to localize landmarks due to its previous training process.

Refined-RetinaFace (R-RetinaFace):

Despite the fact that RetinaFace model is not a real time model, it possesses several advantages over other face detection models such as YOLO, MTCNN, SSD, etc. RetinaFace is a specialized face detection model and designed to detect face and localize facial landmarks (eyes, nose tip, and mouth corners) In addition, RetinaFace is the most robust model to face detection challenges such as variation in pose, illumination, facial expression, etc., and can detect small faces. Ponnmoli and Pandian, (2025) compared seven face detector models, (MTCNN, YOLO, Dlib CNN, SSD, SSH, Tiny Face Detector and Haar cascade) all of which were pre-trained on WIDERFACE database. They found that Dlib CNN is ranked the first by achieving the highest accuracy (92%) and Haar cascade wase ranked the last with an accuracy of 45%. In a similar

DOI: https://doi.org/10.25271/sjuoz.2026.14.1.1662

work, Ren *et al.,* (2025) compared RetinaFace with MTCNN, Fast-RCNN, DSFD and YOLOv8. They found that RetinaFace achieved the highest accuracies with easy subset (94.76%), Medium subset (93.22%) and hard subset (84.92).

Despite the aforementioned features of RetinaFace, sometimes it shows limitations in detecting non-face areas surrounding the actual face as part of the bounding-box and excluding crucial facial landmarks, such as the nose tip, when dealing with profile faces. Deng *et al*., (2019), attributed this limitation to one of the following reasons:

1) Large size of the receptive field leading the ResNet50 to capture features from a larger area than the actual face.
2) Misalignment between anchor boxes and the face, which may include the surrounding areas
3) When training data includes faces with varying amount of surrounding context.
4) When the threshold value to match the priori box with ground-truth box is low, more surrounding areas are detected as a part of actual face region. The default threshold value using in training RetinaFace is 0.5. Increasing this threshold value may filter out the small faces.

The aim of this paper is to reduce these limitations. Based on the aforementioned reasons, this can be done using one of the following scenarios:

1) Adjusting the detection threshold: Experiment with different threshold values to find the optimal balance between detection accuracy and reducing surrounding area inclusion.
2) Using a more precise face alignment model to refine the detected face landmarks and reduce surrounding area inclusion.
3) Post-refinement of the bounding box to minimize the non-face context.

The first and second scenarios may demand some changes in the parameters of ResNet50, FPN and context module, which is a difficult task, since changing parameters, may affect the overall performance of the model. Therefore, in this paper, the scenario of post-refinement on the output bounding-boxes is adopted. The refinement includes adding a post-optimizer to the multi-task unit of the original RetinaFace model. This post-optimizer uses the locations of the landmark's features (eyes, not tip and mouth corners) to resize the bounding box such that the non-face contexts are minimized and at the same time all landmarks lie within the box. In doing so, a rule-based method (IF-THEN) is used which is based on the locations of the landmarks taking into consideration that the x and y-coordinates of the landmarks vary according to the pose and orientation of the face (front, Left profile, right profile, upward downward). In front, upward and downward faces, the right eye is projected near to the left side of the bounding box and left eye is projected near to right side of the bounding box, Figure (3). For left profile faces, either the nose tip or the right mouth corner or the right eye, depending on the value of view angle, are projected near to the left side of the bounding box. For right profile faces, either the nose-tip or the left mouth corner or the left eye, depending on the value of view angle, are projected near to the right side of the bounding-box. For upward and downward faces, eyes are projected near to the top side of the bounding box and the mouth corners are near to the bottom side of the bounding-box. In addition, in some cases where there is face orientation, one eye is located nearer to the top side than the other.

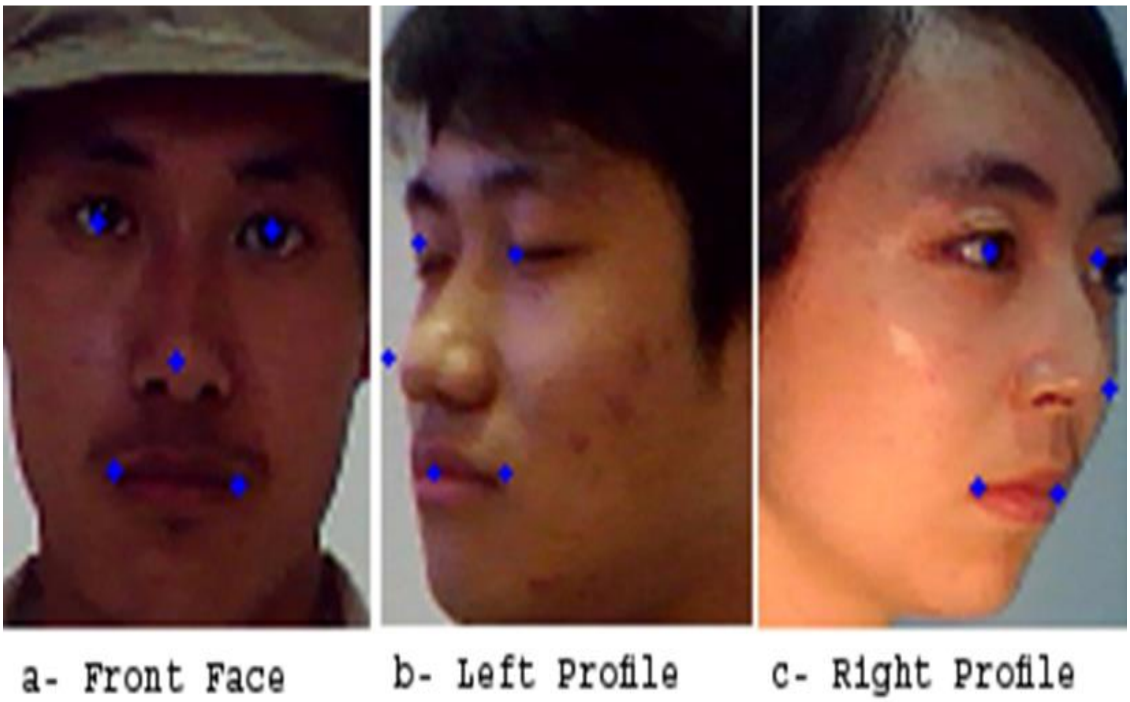

Figure 3: Front, left profile and right profile faces as detected by RetinaFace.

Taking all these cases in the consideration, the rule-based algorithm is designed to perform post-optimization module, as shown in the following algorithm:

DOI: https://doi.org/10.25271/sjuoz.2026.14.1.1662

Algorithm: Rule-based algorithm for Post-optimization Module

Step 1: *Record the* x and $y-coordinates$ *for the five landmarks w.r.t. the top-left corner of the bounding-box.*

Let:

$x1$ and $x2$ *: represent the centers of the right and left eyes,*

$x3$ *: represents the nose tip,*

$x4$ and $x5$ *: represent the right and left mouth corners.*

Step 2: *Select the minimum and maximum of these* $x-coordinates$ *to represent the initial starting x and initial ending x of the bounding box. Use the following functions:*

$$Min\_x = armin(x1, x2, x3, x4, x5)$$
$$Max\_x = armax(x1, x2, x3, x4, x5)$$

Step 3: *Shift* Min_x *to the left and* Max_x *to the right by a number of pixels (here the shifting distance is chosen as 15 pixels):*

$$Min\_x = Min\_x - 15)$$
$$Max\_x = Max\_x + 15)$$

The shifting distance (15 pixels) is chosen such that to minimize the non-face context and to make sure that all landmarks are within the refined bounding-box.

Step 4: *Repeat steps (1-3) for* $y-coordinates$ *of the landmarks:*

$$Min\_y = armin(y1, y2, y3, y4, y5)$$
$$Maxy\_ = armax(y1, y2, y3, y4, y5)$$
$$Min\_y = (Min\_y - 15)$$
$$Max\_y = (Max\_y + 15)$$

Step 5: *Draw new bounding-box with the following points:*

$$Top - left\ coordinates\ (Min\_x, Min\_y)$$
$$Bottom - right\ coordinates\ (Max\_x, Max\_y)$$

The algorithm is based on resizing the bounding box by shifting the nearest left, furthest right, the top most and bottom-most landmarks by a threshold distance The algorithm is conceptually simple and does not need any extra information, except the facial landmarks which are provided by the facial landmark loss-function in the multi-task block of Figure (2).

Figure (4) shows the multi-task loss unit after refinement. Figure (5) shows the faces of Figure (2) after the application of post-optimizer. It can be noticed that the non-face contexts are reduced to a great extent and all the landmarks lie within the bounding boxes.

However, the selection of the threshold value (15) may represent one limitation of the algorithm. Decreasing this value may lead to cut facial area, while increasing this value may lead to detect non-facial areas. In some face detection applications such as face recognition, cutting facial features may lead to the loss of other important features such as the curve of the face and the corners of the eyes. While increasing the threshold value may lead to degrade the performance of recognition. However, to guarantee that all the facial area is detected, the left least, right least, top and bottom landmarks are used as the basis of the rule-based algorithm. In the paper, the value of 15 was determined empirically through experimentation.

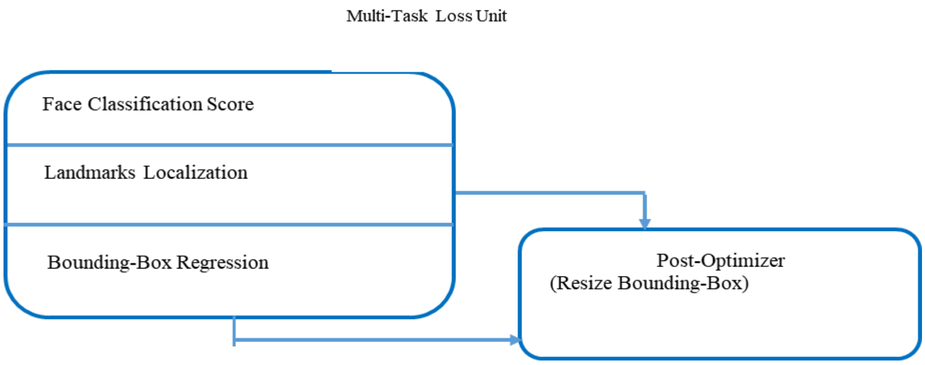

Figure 4: The multi-task loss unit after post-refinement.

DOI: https://doi.org/10.25271/sjuoz.2026.14.1.1662

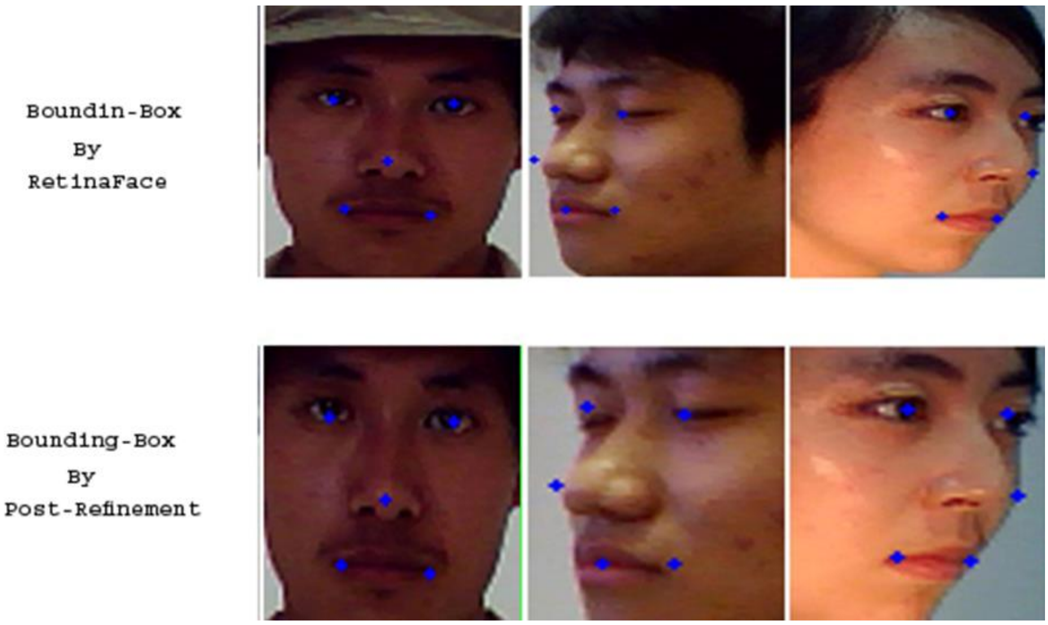

Figure 5: Faces Bounded by RetinaFace and R-RetinaFace.

Databases:

The performance of the proposed model is evaluated on two databases, SDUMLA-HMT and CASIA-3D-FaceV1. SDUMLA-HMT is a homologous multimodal biometric database collected by a group of Machine Learning Applications at Shandon University (SDUMLA) in 2010. It consists of 8904 face images for 106 subjects taken with different poses (upward view, downward view, and forward view, left and right), four different facial expressions (smiling, frowning, surprised, and closed eyes), different illuminations and two accessories, including glasses and hat (Yin *et al*, 2011). CASIA-3D-FaceV1 was collected by the Chinese Academy of Sciences' Institute of Automation (CASIA) in 2004. It consists of 4624 face 3D images for 123 subjects with five different poses (frontal, left, right, upward, and downward), six different facial expressions (normal, happy, sad, angry, surprised, and disgusted) and different illuminations and only glasses as accessory with one image per user with glass (CASIA-3D FaceV1, 2004; Zhong *et al*., 2007; Zhong *et al*., 2008). (Table 1) shows the details of these databases.

Table 1: Descriptions of the Databases

Database	No. of Users	No. of Images	Images per user	Image size	Color format	Accessories
SDUMLA-HMT	106	8904	84	640 X 480	RGB	Eye-glasses+ Hat
CASIA-3D-FaceV1	123	4624	37-38	640 X 480	RGB	Eye-glasses

RESULTS, DISCUSSION AND EVALUATION

The proposed (R-RetinaFace) model and the original RetinaFace model were implemented on SDUMLA-HMT and CASIA-3D-FaceV1 datasets. Images were subsampled to quarter size to accelerate face detection. However, the evaluation of the model performance cannot be done directly, since both databases lack the ground-truth annotations for bounding boxes. In such cases, one of these scenarios can make evaluation. First scenario, by creating testing images by annotating a subset of the database manually. Second scenario, by evaluating the model on a public database holding annotations of the bounding boxes and at the same time being similar to the given database. Third scenario, by visual inspection of the model's outputs for a subset of database and evaluate in a qualitatively sense the position of the bounding boxes and percentage of non-face features within each bounding box. The first method can provide a good estimate of accuracy only if the created testing subset is well representing the global database. Otherwise, it can only provide a rough estimate of accuracy. The second method demands the public dataset and the given dataset to be similar. The third method depends on visual inspection and seems to be more flexibly since it depends on qualitative sense and may give reasonable results if a subset of the output images is selected carefully.

In this paper, third scenario of visual inspection of the model outputs is adopted for testing the model. In doing so, two subsets of images each representing 12% of the two databases are selected. For SDUMLA-HMT, 1060 images were selected in total, ten images per user. For CASIA-3D-FaceV1, 615 images were selected in total, five images per

DOI: https://doi.org/10.25271/sjuoz.2026.14.1.1662

user. For each subset, images with different pose, different illumination, different expression and accessories are selected. During the visual evaluation of the outputs, the focus was placed on assessing the accuracy of face area and landmark localization. Three level of detection were considered in the assessment, ideal detection, moderate detection, and poor detection. The ideal detection is considered, if the bounding box precisely encloses all five key landmarks (both eyes, nose tip, and mouth corners) with no or minimal non-face contents. The moderate detection is considered, if the bounding box encloses all five key landmarks with small amount of surrounding context. Note that, the achievement of face detection with moderate level has been considered as a successful result in most of the previous works. The poor detection is considered, if the bounding box does not enclose all five key landmarks with medium to large amount of non-face contents. The evaluation results for both models, RetinaFace and R-RetinaFace using both databases are shown in (Tables 2 and 3). According to these tables, R-RetinaFace model outperforms RetinaFace model. (Table 2) shows that the detection level for most images with RetinaFace model was moderate, giving moderate detection rate of 96.32% and 83.9% for SDUMLA-HMT and CASIA-3D-FaceV1 databases respectively. While, the poor detection rates are 3.68% and 16.1% for the two databases and the ideal detection rate is zero for both databases. On the other hand, (Table 3) shows that the R-RetinaFace model achieved ideal detection rates of 98.02 and 92.2, moderate detection rates of 1.32% and 7%, and poor detection rate of 0.66% and 0.8% for both databases.

Table 2: Accuracies for the Three Detection Levels Achieved by RetinaFace Model

Database	No. of Test Images	Poor Detection	Moderate Detection	Ideal Detection
SDUMLA-HMT	1060	3.68%	96.32%	0
CASIA-3D-FaceV1	615	16.1%	83.9%	0

Table 3: Accuracies for the Three Detection Levels Achieved By R-RetinaFace Model

Database	No. of Test Images	Poor Detection	Moderate Detection	Ideal Detection
SDUMLA-HMT	1060	0.66%	1.32%	98.02%
CASIA-3D-FaceV1	615	0.8%	7%	92.2%

Figure (6) shows some face images, including frontal and profile poses, with superimposed bounding boxes produced by both models, RetinaFace and R-RetinaFace models for both databases SDUMLA-HMT and CASIA-3D-FaceV1. The figure shows that the bounding boxes produced by RetinaFace model contain non-face contents such as hair, hat and background and categorized as moderate-detection. While, the bounding boxes produced by R-RetinaFace contain almost nothing of the non-face contents and categorized as Ideal-Detection. In specific, the profile face and faces with accessories are well detected by R-RetinaFace compared to the original RetinaFace. In addition, in one of the profile faces, the face landmark feature, specifically the nose tip, is detected by RetinaFace outside the box, while in R-RetinaFace, the box bounded all the facial landmarks.

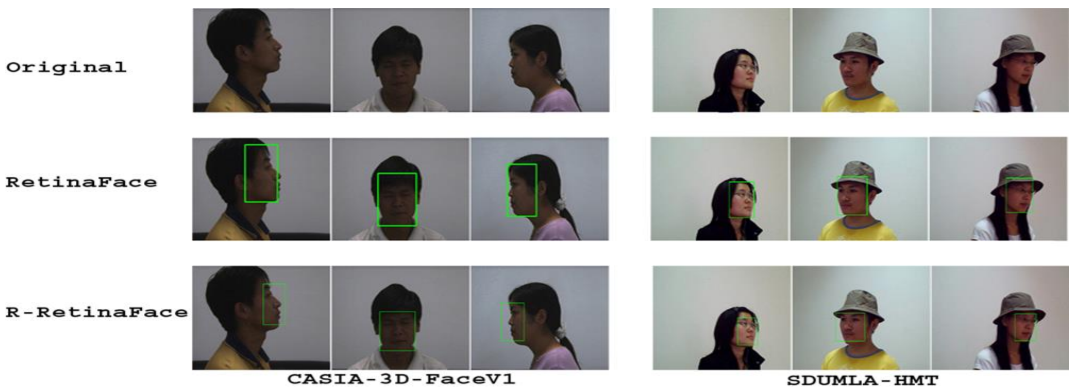

Figure 6: Samples of original images from CASIA-3D-FaceV1 and SDUMLA-HMT database with detected face enclosed in the bounding boxes using RetinaFace and R-RetinaFace.

Figures (7 and 8) show faces detected by both RetinaFace and R-RetinaFace models for both databases. Figure (7) shows samples from SDUMLA-HMT database with bounding boxes and facial landmarks as detected by

DOI: https://doi.org/10.25271/sjuoz.2026.14.1.1662

RetinaFace and R-RetinaFace models. The hat, hair and other background, which are detected by RetinaFace as actual face, are excluded by R-RetinaFace. Figure (8) shows samples from CASIA-3D-FaceV1 database as detected by RetinaFace and R-RetinaFace models. This database does not possess face images with hat, but still it can be seen that the hair and background contents, which are detected by RetinaFace are almost excluded by R-RetinaFace. These results ensure that R-RetinaFace outperforms RetinaFace. In particular, these results ensure the effectiveness of the post-optimizer component for improving face detection level to ideal detection. In addition, they ensure the effectiveness of transfer-learning process as the RetinaFace is a pre-trained on WIDERFACE dataset and tested on SDUMLA-HMT and CASIA-3D-FaceV1.

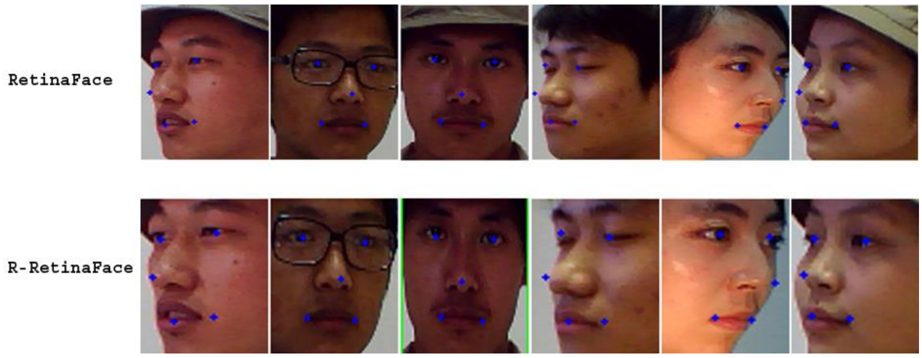

Figure 7: Detected face by RetinaFace and R-RetinaFace for SDUMLA-HMT Database.

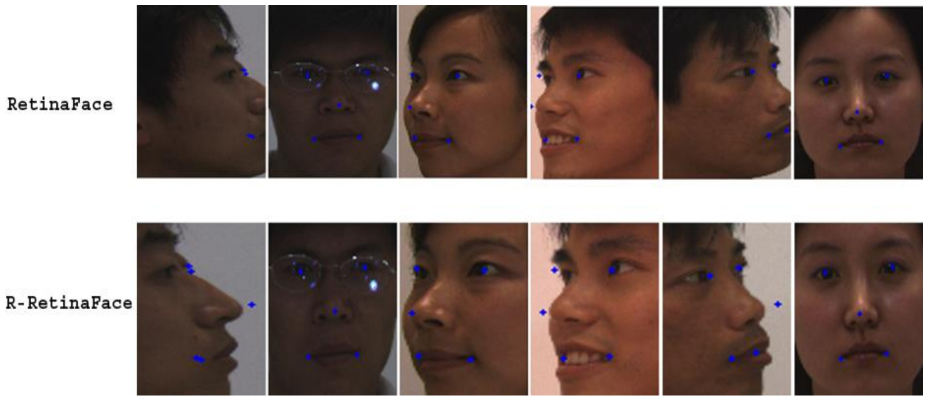

Figure 8: Detected face by RetinaFace and R-RetinaFace for CASIA-3D-FaceV1 Database.

The accuracies achieved in this paper by RetinaFace and R-RetinaFace are compared with the accuracies of seven deep learning models achieved previously by (Ponnmoli and Pandian, 2025) and three versions of RetinaFace models by (Ren *et al*., 2025). (Table 4) shows that the accuracy provided by R-RetinaFace for both databases are better than others.

Table 4: Comparison with previous Results achieved by (Ponnmoli and Pandian, 2025; Ren *et al*., 2025)

Reference	Model	Accuracy
(Ponnmoli & Pandian, 2025)	Haar Cascade	45%
	MTCNN	70%
	SSH	80%
	Tiny Face Detector	82%
	YOLO	88%
	Dlib CNN	92%
	OpenCV SSD ResNet	85%
(Ren *et al*., 2025)	RetinaFace with easy subset	94.76%
	RetinaFace with moderate subset	93.22%
	RetinaFace with hard subset	84.92%
The Proposed R-RetinaFace Model	R-RetinAFace with SDUMLA-HMT Database (Proposed)	98.02%
	R-RetinAFace with with CASIA-3D-FaceV1 Database (Proposed)	92.2%

DOI: https://doi.org/10.25271/sjuoz.2026.14.1.1665

CONCLUSIONS

R-RetinaFace improves face detection rates. It achieves ideal detection with minimal non-face content and accurate landmark localization. In contrast, the original RetinaFace achieves only moderate detection. Its bounding boxes contain non-face areas, and landmarks are often incorrectly localized. The post-optimization module enhances RetinaFace's efficiency. R-RetinaFace's success on 2D (SDUMLA-HMT) and 3D (CASIA-3D-FaceV1) databases proves its flexibility and robustness. The model's ability to detect facial landmarks is crucial for face recognition. The post-cropping module reduces unwanted areas, increasing ideal face detection accuracy. R-RetinaFace's performance confirms its versatility in handling diverse image sources. However, incorrect threshold value selection may lead to negative results, therefore, this value should be chosen with some pre-cautions.

Acknowledgement:

The authors would like to thank the University of Duhok, Kurdistan Region of Iraq, for its support. They would also like to express their deep thanks to Shandong University in Jinan / China, the producer of SDUMLA-HMT multimodal biometrics datasets. Also, deep thanks go to Chinese Academy of Sciences Institute of Automation, the developer of CASIA-3D-FaceV1database.

Ethical statement:

The Ethical Committee of the University of Duhok approved the current experiment.

Author Contributions:

The MSc student Sagvan J. M. Ameen under the supervision of Prof.Dr Ahmed AK. Tahir conducts this paper as a part of MSC research.

Both authors have reviewed the final version to be published and agreed to be accountable for all aspects of the work.

REFERENCES

Al-Dabbas, H. M., Azeez, R. A., Ali, A. E. (2023). Machine Learning Approach for Facial Image Detection System, Iraqi *Journal of Science*, 64(10), 5428- 5441. DOI: 10.24996/ijs.2023.64.10.44.

Boss, P., Bandyopadhyay, S. K. (2020). Human Face and Facial Parts Detection using Template Matching Technique. International *Journal of Engineering and Advanced Technology (IJEAT)*, 9(4), 2296-2299. DOI: 10.35940/ijeat.D6689.049420.

CASIA-3D FaceV1, 2004, http://biometrics.idealtest.org/.

Chen, W., Huang, H., Peng, S., & Zhou, C. (2020). YOLO-face: a real-time face detector. The Visual Computer, 37(4), 805-813. DOI:10.1007/s00371-020-01831-7.

Çarıkçı, M., & Özen, F. (2012). A face recognition system based on eigenfaces method. *Elsevier*, Vol. 1, 118–123. DOI:10.1016/j.protcy.2012.02.023.

Cerna, L. R., Cámara-Chávez, G., & Menotti, D. (2013). Face detection: Histogram of oriented gradients and bag of feature method. In Proceedings of the 2013 International Conference on Image Processing, Computer Vision, and Pattern Recognition, IPCV, Vol. 2, 657-661.

Deng, J., Guo, J., Zhou, Y., Yu, J., Kotsia, I., & Zafeiriou1, S. (2019). RetinaFace: Single-stage Dense Face Localisation in the Wild, 1-10. DOI: 10.48550/arXiv.1905.00641.

Deng, J., Guo, J., Ververas, E., Kotsia, I. & Zafeiriou, S. (2020). RetinaFace: Single-Shot Multi-Level Face Localisation in the Wild," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 5202-5211. DOI: 10.1109/CVPR42600.2020.00525.

Dengi, O., & Patil, D. Y. (2024). A Comparative Study of Classical and Modern Face Detection and Recognition Methods: Accuracy, Challenges, Efficiency and Performance Analysis. International Research *Journal of Engineering and Technology*, 11(5), 704-709.

Gao, Q., Ding, B., Jia, X., Xie, Y., & Han, X. (2024). Dense pedestrian face detection in complex environments, Scientific Report, 14(1), 1-17. DOI:10.1038/s41598-024-72523-8.

Guo, X. (2021). A KNN Classifier for Face Recognition. International Conference on Communications, Information System and Computer Engineering (CISCE), Beijing, China, 292-297. DOI: 10.1109/CISCE52179.2021.9445908.

Hajraoui, A., Slimane, M., Mellal, B., & Sabri, M. M. (2014). Face Detection Algorithm based on Skin Detection, Watershed Method and Gabor Filters. *International Journal of Computer Applications*, 94(6), 33-39. DOI:10.5120/16349-5695.

Hangaragi, S., Tripty, S., & Neelima, N. (2023). Face Detection and Recognition Using Face Mesh and Deep Neural Network, International Conference on Machine Learning and Data Engineering, Perocedia Computer Science, Vol. 218, 741-749. DOI:10.1016/j.procs.2023.01.054.

Hasan, B. M. S., & Mstafa, R.J. (2022). A Study of Gender Classification Techniques Based on Iris Images: A Deep Survey and Analysis. *Science Journal of University of Zakho*, 10(4), 222 –234. DOI:10.25271/sjuoz.2022.10.4.1039.

Hassan, M., Biswas, A., Al Hakim, N., Mumtaj, M., Samin, A. A., Hossen, M. S., Hossain, M. L., Soumo, S. P., Dey, A. (2025). Face Detection with MTCNN Using Densenet for Enhance Security. *Journal of Networking and Communication Systems*, 8(1), 33-49. DOI: 10.46253/jnacs.v8i1.a4.

Hasan, M. K., Ahsan, M. S., Abdullah-Al-Mamun, Newaz, S. H. S., & Lee, G. M. (2021). Human Face Detection Techniques: A Comprehensive Review and Future Research Directions. Electronics, 10(19), 1-46. DOI:10.3390/electronics10192354.

Hasan, Z. F. (2022). An Improved Facial Expression Recognition Method Using Combined Hog and Gabor Features. Scientific *Journal of University of Zakho*,10(2), 54-59. DOI:10.25271/sjuoz.2022.10.2.897.

Hassen, F., & Naser, M. A. (2024).A Face Detection System: A Comprehensive Survey. (2024). *Journal*

DOI: https://doi.org/10.25271/sjuoz.2026.14.1.1665

Of University Of Babylon For Pure And Applied Sciences, 32(2), 45-61. DOI:10.29196/jubpas.v32i2.5266.

Ho, H. T., Nguyen, L. V., Le, T. H. T., & Lee, O. J. (2024). Face Detection Using Eigenfaces: A Comprehensive Review, in *IEEE Access*, vol. 12, 118406-118426. DOI: 10.1109/ACCESS.2024.3435964.

Jabbar, M. K., Hussain, M. A., & Kareem, T. A. (2018). Face Detection and Recognition using Color Segmentation, Template Matching and Gabor Neural Network with Fuzzy System, Wasit *Journal of Engineering Sciences*, 6(3), 29-38. DOI:10.31185/ejuow.Vol6.Iss3.102.

Jiang, H., & Learned-Miller, E. (2017). Face Detection with the Faster R-CNN. 12th IEEE International Conference on Automatic Face & Gesture Recognition, Washington, DC, USA, 650-657. DOI: 10.1109/FG.2017.82.

Kokare, S. & Ghisare, V. (2025). SVM-Based Approach for Human Face Detection and Recognition, 16(2), 1-5. DOI:10.71097/IJSAT.v16.i2.3306**.**

Komlavi, A. A., Chaibou, K., & Naroua, H. (2024). Comparative study of machine learning algorithms for face recognition, Revue Africaine de Recherche en Informatique et Mathématiques Appliquées, Vol. 40, 1-27. DOI:10.46298/arima.9291.

Kremic E., & Subasi, A. (2015). Performance of random forest and SVM in face recognition. *Int. Arab J. Inf. Technol*., 3(2), 287–293.

Kukenys I., & Mccane, B. (2008). Support Vector Machines for Human Face Detection. Proceedings of the New Zealand Computer Science Research Student Conference [Online]. Available:

Kumar, A. (2014). An Empirical Study of Selection of the appropriate Color Space for Skin Detection: A Case of Face Detection in Color images. *International Conference on Issues and Challenges in Intelligent Computing Techniques* (ICICT), Ghaziabad, India, 725-730, DOI: 10.1109/ICICICT.2014.6781370.

Kumar, A., Kaur, A., & Kumar, M. (2019). Face detection techniques: A review. *Artificial Intelligence Review*, 52(2), 927–948. DOI: 10.1007/s10462-018-9650-2.

Liu, B., & Yu, H. (2023). A Lightweight and Accurate Face Detection Algorithm Based on RetinaFace. 1-14. DOI:10.48550/arXiv.2308.04340.

Madan, A. (2021). Face Recognition Using Haar Cascade Classifier. Modern Trends in Science and technology, 7(10), 85-87. DOI: 10.46501/IJMTST070119.

Mady, H., & Hilles,S. M. S. (2018). Face recognition and detection using Random forest and combination of LBP and HOG features. International Conference on Smart Computing and Electronic Enterprise. (ICSCEE2018) ©2018 IEEE.

Mala, Y. H., & Mohammad, M. A. (2022). Brain Waves Signal Modeling for Object Classification Using Random Forest Method. *Science Journal of University of Zakho*,10(1), 16-23. DOI:10.25271/sjuoz.2022.10.1.876.

Maw, H. M., Lin, Z., & Mon, T. (2018). Face Detection using Fusion of Skin Detection and Viola-Jones Face Detection. 2nd International Conference on Advanced Information Technologies (ICAIT), Nov. 1-2, Yangon, Mynamar, 68-73.

Minaee, S., Luo, P., Lin, Z., & Bowyer, K. (2021). Going Deeper Into Face Detection: A Survey, 1-7. https://arxiv.org/abs/2103.14983.

Ochango, V. A. (2023). A Model for Face Recognition using EigenFace Algorithm. International *Journal of Formal Sciences: Current and Future Research Trends (IJFSCFRT)*, 18(1), 12-21.

Pal, S. (2020). Human Face Detection Technique using Haar-like Features. International *Journal of Computer Applications*, 175(32), 56-60. DOI: 10.5120/ijca2020920883.

Paul, P. P., & Gavrilova, M. (2011). PCA based geometric modeling for automatic face detection. 2011 International Conference on Computational Science and Its Applications, *Santander, Spain, 33-38,* DOI: 10.1109/ICCSA.2011.69.

Ponnmoli, K. M., & Pandian, A. (2025). Comparative Evaluation of Face Detection Algorithms: Accuracy, Efficiency, and Robustness." Indica *Journal*, 6(3), 1-8. DOI: 10.5281/zenodo.15023420.

Praveen, K., Rahman, H. K., Nandakumar, D., Shervin S., & Jyotsna. A. (2025). Face Detection Attendance Marking Using AI: A Deep Learning Approach for Classroom Environment, *International Research Journal of Innovations in Engineering and Technology* (IRJIET), 9(1), 11-16. DOI:10.47001/IRJIET/2025.901002.

Qi, D., Tan, W., Yao, Q., & Liu, J. (2022). YOLO5Face: Why Reinventing a Face Detector. 1-10. http://arxiv.org/abs/2105.12931.

Ren, Z., Liu, X., Xu, J., Zhang, Y., & Fang, M. (2025). LittleFaceNet: A Small-Sized Face Recognition Method Based on RetinaFace and AdaFace. *Journal of Imaging*, 11(1), 1-20. DOI: 10.3390/jimaging11010024.

Sakai, T., Nagao, M. &Kanade, T. (1972). Computer Analysis and Classification of Photographs of Human Faces. First USA-JAPAN Computer Conference 55-62.

Saputra, S., Akbar, M., & Hikmatiar, H. (2025). Graphical User Interface (GUI) for Face Detection Using Viola-Jones Algorithm. Bincang Sains dan Teknologi, 4(1), 1–9. DOI: 10.56741/bst.v4i01.768.

Soni L. N., & Waoo, A. A. (2023). A Review of Recent Advances Methodologies for Face Detection. 86-International *Journal of Current Engineering and Technology*, 13(2), 86-92. DOI.org/10.14741/ijcet/v.13.2.6.

Tahir, A. AK. & Anghelus, S. 2024. Biometric Based Recognition Systems - An Overview. International *Journal of Open Information Technologies*, 12(7), 110-118. http://injoit.org/index.php/j1/article/viewFile/1858/1734.

Thaher, T., Mafarja, M., Saffarini, M., Abdulhakim. M. Mohamed, A. M., & Ayman A. El-Saleh, A. A. (2025). A Comprehensive Review of Face Detection Techniques for Occluded Faces:

DOI: https://doi.org/10.25271/sjuoz.2026.14.1.1665

Methods, Datasets, and Open Challenges, Computer Modeling in Engineering Science, 43(3), 2615-2673. DOI: 10.32604/cmes.2025.064857.

Thakurdesai, N., Raut, N., Tripathi, A. (2018). Face Recognition using One-shot Learning. *International Journal of Computer Applications*, 182(23, 35-39. DOI: 10.5120/ijca2018918032.

Tripathi, S., Sharma, V., & Sharma, S. (2011). Face Detection using Combined Skin Color Detector and Template Matching Method. International Journal of Computer Applications, 26(7), 5-8. DOI: 10.5120/3119-4290.

Tsai, C. C., Cheng, W. C., Taur, J. S., and Tao, C. W. (2006). Face Detection Using Eigenface and Neural Network. 2006 IEEE International Conference on Systems, Man, and Cybernetics, October 8-11, 2006, Taipei, Taiwan, Vol. 5, 4343-4347. DOI: 10.1109/ICSMC.2006.384817.

Viola, P.; Jones, M. (2001). Rapid Object Detection Using A Boosted Cascade of Simple Features. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Kauai, HI, USA Vol. 1, 1-9. DOI: 10.1109/CVPR.2001.990517.

Viola, P., Jones, M.J. (2004). Robust Real-Time Face Detection. *International Journal of Computer Vision*, 57, 137–154. DOI:10.1023/B:VISI.0000013087.49260.fb.

Wang, J., Yuan, Y., & Yu, G. (2017). Face Attention Network: An Effective Face Detector for the Occluded Faces. 1-10. DOI:10.48550/arXiv.1711.07246.

Wirdiani, N. K. A., Hridayami, P., Widiari, N. P. A., Rismawan, K. D., Putu Bagus Candradinatha, P. B., & I Putu Deva Jayantha, I. P. D. (2019). Face Identification Based on K-Nearest Neighbor. *Scientific Journal of Informatics*, 6(2), 150–159. DOI: 10.15294/sji.v6i2.19503.

Xiong, Y., Meng, W., Yan, J., & Yang, J. (2023). A Rotation-Invariance Face Detector Based on RetinaNet. *Journal of Physics: Institute of Physics*, 1-7. DOI: 10.1088/1742-6596/2562/1/012066.

Xue, X., Hu, J., & Zhang, P. (2020). Intelligent detection and recognition system for mask wearing based on improved RetinaFace algorithm. Proceedings - 2020 2nd International Conference on Machine Learning, Big Data and Business Intelligence, MLBDBI 2020, Institute of Electrical and Electronics Engineers Inc., 474–479. DOI: 10.1109/MLBDBI51377.2020.00100.

Yang, S., Luo, P., Loy C. C., & Tang, X. (2016). WIDER FACE: A Face Detection Benchmark. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 5525-5533, DOI: 10.1109/CVPR.2016.596.

Yazdani, H. R., & Shojaeifard, A. R. (2023). Facial recognition system using eigenfaces and PCA. *Mathematics and Computational Sciences*, 4(1), 29-35. DOI: 10.30511/mcs.2023.562662.1085.

Ye, B., Shi, Y., Li, H., Li, L., & Tong, S. (2021). Face SSD: A Real-time Face Detector based on SSD. 40th Chinese Control Conference (CCC), Shanghai, China, 8445-8450. DOI: 10.23919/CCC52363.2021.9550294.

Yin, Y., Liu, L. & Sun, X. (2011). SDUMLA-HMT: A Multimodal Biometric Database. Proceedings of the 6th Chinese conference on Biometric recognition, 260–268. https://dl.acm.org/doi/proceedings/10.5555/2074627.

Yousif, R. Z., Abdulrahman Hamad, S. A., & S. Mohammed Jihad Abdalwahid, S. M. J. (2024). A Comprehensive Review of Face Detection Using Machine Learning and Deep Learning Approaches. 5th International Conference On Communication Engineering and Computer Science (CIC-COCOS'24), Cihan University-Erbil, 369–374. DOI: 10.24086/cocos2024/paper.1499.

Yuen, C. T., Rizon, M., San, W. S., & Seong, T. C. (2009). Facial Features for Template Matching Based Face Recognition. *American Journal of Applied Sciences* 6 (11), 1897-1901. DOI: 10.3844/ajassp.2009.1897.1901.

Zhang, F., Fan, X., Ai, G., Song, J., Qin, Y., & Wu, J. (2019). Accurate Face Detection for High Performance. 1-9. https://arxiv.org/abs/1905.01585.

Zhang, N., Luo, J., & Gao, W. (2020). Research on face detection technology based on MTCNN. Proceedings - 2020 International Conference on Computer Network, Electronic and Automation (ICCNEA), Institute of Electrical and Electronics Engineers Inc., 154–158. DOI: 10.1109/ICCNEA50255.2020.00040.

Zhang, X., Gonnot, T., & Saniie, J. (2017). Real-Time Face Detection and Recognition in Complex Background. *Journal of Signal and Information Processing*, 8(2) 99-112. DOI: 10.4236/jsip.2017.82007.

Zhong, C., Sun, Z., & Tan, T. (2007). Robust 3D Face Recognition Using Learned Visual Codebook. 2007 IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, 1-6. DOI: 10.1109/CVPR.2007.383279.

Zhong, C., Sun, Z., & Tan, T. (2008). Learning Efficient Codes for 3D Face recognition. 15th IEEE International Conference on Image Processing, San Diego, CA, 1928-1931. DOI: 10.1109/ICIP.2008.4712158.

DOI: https://doi.org/10.25271/sjuoz.2026.14.1.1665