Journal of Computer Networks, Architecture and High Performance Computing
Volume 7, Number 3, July 2025
https://doi.org/10.47709/cnahpc.v7i3.6161

Submitted : Jun 19, 2025
Accepted : Jul 7, 2025
Published : Jul 10, 2025

Multiscale Facial Detection using RetinaFace Architecture with Loss Function

Irma Amelia Dewi*[1)], Nadhiva Adzra Tsania Maryadi[2)]
[1)2)]Department of Informatics, Institut Teknologi Nasional Bandung, Indonesia
[1)]irma_amelia@itenas.ac.id, [2)] nadhiva.adzra123@gmail.com

ABSTRACT

Facial recognition technology has become increasingly prevalent in modern applications, from security systems to social media platforms. However, one of the most significant challenges in this field remains the accurate detection of faces across varying scales, orientations, and image qualities. Traditional face detection methods often struggle when faces appear at different sizes within the same image or when dealing with low-resolution imagery, leading to inconsistent performance that can compromise system reliability. The RetinaFace architecture emerges as a promising solution to address these multiscale detection challenges. By incorporating a Feature Pyramid Network (FPN), the system creates a hierarchical representation of features that enables effective detection of faces regardless of their size in the image. The FPN works by combining low-resolution, semantically strong features with high-resolution, semantically weak features, creating a robust feature pyramid that captures facial characteristics at multiple scales simultaneously. Context modules within RetinaFace further enhance detection capabilities by providing additional contextual information that helps distinguish faces from background noise and other objects. This comprehensive approach allows the system to maintain high accuracy even in challenging scenarios where faces appear small, partially occluded, or at unusual angles. The comparative analysis between ArcFace and SphereFace loss functions reveals important insights into optimization strategies for facial recognition systems. The experimental results on the WIDERFACE dataset demonstrate exceptional performance, with the RetinaFace-ResNet152-SphereFace combination achieving 94% accuracy. These findings highlight the importance of architectural choices and loss function selection in developing robust facial recognition systems capable of handling real-world deployment challenges.

Keywords: ArcFace loss; Facial Recognition; RetinaFace; SphereFace loss; Widerface

INTRODUCTION

Facial recognition detection is a biometric capability and one of the most critical fields in the development of artificial intelligence technology, especially in applications such as security, surveillance, and authentication. Facial recognition works on the principle of comparing an image of a face with a database of faces, yielding a match or similarity score. Facial recognition systems typically involve two stages: face detection and face recognition (Kortli, Jridi, Al Falou, & Atri, 2020). As technology advances, the demand for accurate facial recognition systems has increased. These systems are used in a variety of applications, from access control to automated surveillance and consumer devices like smartphones and smart cameras. However, one of the main challenges in facial recognition detection is handling faces at different scales, also known as multiscale detection.

Multiscale facial detection refers to the system's ability to detect faces of various sizes within a single image. In practice, faces can appear at different sizes due to the relative distance from the camera or varying image resolutions. Accurate multiscale facial detection is crucial before the face recognition stage. Without the ability to accurately detect faces of different sizes, the face recognition process can be compromised by inaccurate detections or missed detections of faces of different sizes. Strong multiscale detection capabilities ensure that all faces in an image are accurately detected, thereby improving the overall performance and accuracy of facial recognition systems. This importance is highlighted by research from (Li, Guo, Ye, Fan, & Tang, 2020), which shows that multiscale detection capability is critical for real-world applications.

RetinaFace is a face detection architecture designed to achieve high precision in various conditions. RetinaFace uses a convolutional neural network (CNN) to perform end-to-end face detection. One of RetinaFace's main advantages is its ability to handle multiscale face detection effectively, which is crucial for enhancing detection accuracy across different sizes and scales. According to (Deng, Guo, Ververas, Kotsia, & Zafeiriou, 2020), RetinaFace employs techniques like landmark localization and dense regression to achieve superior detection performance. Loss functions are used to enhance facial recognition performance. ArcFace Loss and SphereFace Loss are types of loss functions that can be used with RetinaFace. ArcFace (Additive Angular Margin Loss) is designed to improve inter-

Journal of Computer Networks, Architecture and High Performance Computing
Volume 7, Number 3, July 2025
https://doi.org/10.47709/cnahpc.v7i3.6161

Submitted : Jun 19, 2025
Accepted : Jul 7, 2025
Published : Jul 10, 2025

class discrimination by adding an angular margin to classification decisions, helping the model better distinguish between similar facial features, thereby increasing facial recognition accuracy. SphereFace (A-Softmax Loss) uses an angular margin to maximize inter-class margins, directing facial features to a unit hypersphere manifold, which helps enhance feature generalization and discrimination.

Research by (Deng, Guo, Zhou, et al., 2020) titled "RetinaFace: Single-stage Dense Face Localisation in the Wild" used RetinaFace and the ArcFace loss function. This study demonstrated that RetinaFace achieved an accuracy rate of 99.86% on the LFW dataset in detecting faces across various poses and lighting conditions. Another study by (Das et al., 2022) titled "ArcFace: Additive Angular Margin Loss for Deep Face Recognition" used the ResNet-50 architecture with the ArcFace loss function for facial recognition based on features, showing that ResNet-50 and ArcFace produced highly discriminative feature representations with an accuracy rate of 99.82% on the LFW dataset for face recognition. Additionally, research by (Liu et al., n.d.) titled "SphereFace: Deep Hypersphere Embedding for Face Recognition" used the SphereFace loss function with MTCNN, achieving an accuracy rate of 99.42% on the LFW dataset. Based on previous research, this study focuses on handling multiscale facial detection using the RetinaFace architecture. After detecting faces, the next stage involves facial recognition using two different loss functions. The loss functions compared in this study are ArcFace Loss and SphereFace Loss. This study aims to compare the performance of these two loss functions in the context of facial recognition following multiscale detection using RetinaFace.

Although previous studies have applied ArcFace or SphereFace loss functions independently for face recognition tasks, there is still a lack of comparative analysis between these two loss functions in the context of multiscale face detection using the RetinaFace architecture. Most existing research focuses on improving recognition accuracy under specific conditions or on benchmarking single loss functions, without evaluating their combined effect on face recognition following multiscale detection. Therefore, this study aims to fill this gap by systematically comparing the performance of ArcFace and SphereFace loss functions after facial detection using RetinaFace, especially in scenarios with varying face scales, poses, and resolutions.

LITERATURE REVIEW

This research refers to state-of-the-art research that discusses RetinaFace in recognizing faces. Research by (Deng, Guo, Zhou, et al., 2020) with the title "RetinaFace: Single-stage Dense Face Localization in the Wild" uses RetinaFace with the ResNet-152 backbone. The results of this study show that RetinaFace provides AP values of 96.3% (easy), 95.6% (medium), and 91.4% (hard) on the WIDER FACE dataset. Research by (Nanni, Brahnam, & Lumini, 2023) with the title "Coupling RetinaFace and Depth Information to Filter False Positives", this research uses RetinaFace backbone MobileNetV2, with a focus on reducing face detection errors, especially "False Positives". Tests show that this method is effective in reducing false positives without sacrificing overall accuracy. Research by (Ma & Long, 2023)with the title "A Face Recognition Method Using ResNet34 and RetinaFace", this research uses RetinaFace for face detection by comparing it with OpenCV. This research shows that with the same image OpenCV detects 20 faces, while RetinaFace detects 50 faces. Research by (Zaki, 2023) with the title "Detection of Mask Use in Images Using RetinaFace with MobileNetV2" uses RetinaFace as face detection. RetinaFace takes part of the face in the image. The accuracy result obtained by RetinaFace and MobileNetV2 is 99.3%.

Research by (Alhanaee, Alhammadi, Almenhali, & Shatnawi, 2021) with the title "Face Recognition for Smart Attendance System Using Deep Learning" compares two methods for face detection, namely RetinaFace and MTCNN. The results of this study show that RetinaFace gets mAP values of 94.20% easy, 93.24% medium, and 83.55% hard, better than MTCNN with mAP 83.31% easy, 80.32% medium, 83.55% hard on the WIDER FACE dataset. Research by (Filian, Istianto, & Kusuma, 2024)with the title "Image Enhancement using Convolutional Neural Network for Flow Light Face Detection" uses RetinaFace for face detection. The focus of this research is to overcome the problem of face detection in dark conditions. This study obtained an AP result of 52.94% using the DARKFACE dataset.

Previous research shows that ResNet-152 on RetinaFace excels in face detection accuracy, especially under harsh conditions, but requires longer training time and greater computation. In contrast, MobileNetV1 (0.25) offers speed and efficiency, making it ideal for resource-constrained applications, albeit with reduced accuracy. This research will discuss how RetinaFace detects faces under various scale conditions, as well as evaluate to what extent RetinaFace is able to maintain detection accuracy at various scales and distances in the image. This comparison aims to assess the advantages and disadvantages of each backbone in multiscale face detection applications.

Journal of Computer Networks, Architecture and High Performance Computing
Volume 7, Number 3, July 2025
https://doi.org/10.47709/cnahpc.v7i3.6161

Submitted : Jun 19, 2025
Accepted : Jul 7, 2025
Published : Jul 10, 2025

METHOD

The whole system process begins with preparing the dataset first, then preprocessing and cropping the face using RetinaFace using either ResNet-152 or MobileNet V1 (0.25). Because the number of datasets is quite small, we perform dataset augmentation after which it will be split into training, testing and validation datasets. The training and validation datasets will be used for training ArcFace and SphereFace models with the ResNet-50 backbone as a feature extractor and then the model will be saved when it reaches the highest validation accuracy during training. The model will then be used for face recognition testing.

In Fig 1 the diagram illustrates the entire facial recognition process using two loss functions: ArcFace Loss and SphereFace Loss. It begins with dataset collection, where raw face images are gathered. The next step is preprocessing, which involves enhancing and standardizing image quality, followed by cropping using the RetinaFace model to focus on facial areas. The images then undergo augmentation to increase data variety and model robustness. After preprocessing and augmentation, the dataset is divided into training, validation, and testing subsets to ensure comprehensive evaluation. During training, the model is trained with ArcFace Loss and SphereFace Loss to minimize loss and improve facial recognition accuracy. The trained models are saved as Model ArcFace Loss and Model SphereFace Loss. In the testing phase, the trained models are evaluated using the testing dataset. The performance of ArcFace Loss and SphereFace Loss is measured to assess model performance on unseen data. The results are then used for facial recognition. Based on Figure 1, the research to be conducted has several stages, including:

Collect the Dataset

The research utilizes both public and proprietary datasets. For face detection, the public WIDER FACE dataset is used, containing 32,203 images and 393,703 face labels under various conditions such as scale, pose, and occlusion. This dataset is accessible at WIDER FACE. The proprietary dataset, used for face recognition, consists of 50 identities, each with 18 images, totaling 400 images. In this study, the WIDER FACE dataset is employed for training the RetinaFace architecture, divided into two parts: 80% for training and 20% for testing. For face recognition, the proprietary dataset is used and divided into three parts for training with a loss function: 70% for training, 20% for validation, and 10% for testing. This division is essential for fine-tuning the model and accurately evaluating its performance.

Table 1. Dataset

No	Split Dataset	Image WiderFace	Primary
1	Training	12880	6930
2	Validation	3226	990
3	Testing	16097	1980

From Table 1. shows how the datasets are divided. The dataset for detection uses the WiderFace dataset with a division of 12,880 training data, 3226 validation data and 16097 testing data. As for face recognition, we use the primary dataset taken using the Samsung A52 cellphone. In evaluating the complexity of the training process and resource requirements, all models were trained and tested on a system with an NVIDIA GeForce RTX 4090 GPU, 32GB RAM and an Intel Core i7-1165G7 CPU.

Pre-processing

Preprocessing involves preparing the raw dataset by applying various operations like cropping, resizing, and normalizing images. This ensures the consistency and standardization of the input data. The cropping process utilizes a pre-trained model called RetinaFace, a deep learning-based facial detection and alignment framework for accurate and efficient facial location and detection. During the cropping phase, RetinaFace identifies face coordinates, eyes, nose, and mouth, along with the bounding box surrounding the face. The images are cropped to focus solely on the face, ensuring the model learns face-specific features. Data augmentation increases dataset diversity through various transformations. These augmentations include rotation (0-10°), horizontal flipping, random brightness and contrast adjustments, random scaling (-0.1 to 0.1), x & y shifting (up to 0.05), adding Gaussian noise (10-50), and blurring (0-3). These transformations help the model learn to recognize faces under various conditions, improving its accuracy in real-world face recognition scenarios.

Journal of Computer Networks, Architecture and High Performance Computing
Volume 7, Number 3, July 2025
https://doi.org/10.47709/cnahpc.v7i3.6161

Submitted : Jun 19, 2025
Accepted : Jul 7, 2025
Published : Jul 10, 2025

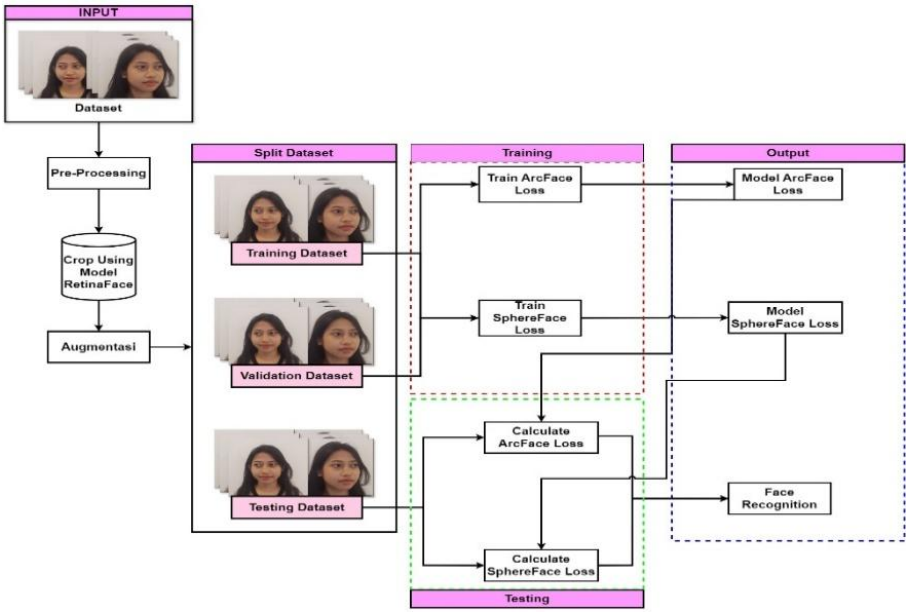

Fig. 1. Sytem process of multiscale face recognition

RetinaFace Architecture

RetinaFace (Deng, Guo, Zhou, et al., 2020) is a state-of-the-art face detection model designed to achieve high accuracy in detecting faces across various scales and challenging conditions. It is built upon the principles of a single-stage dense face localization approach and leverages feature pyramids with independent context modules. The architecture of RetinaFace integrates multiple advanced techniques to ensure robust and precise face detection. A key feature of RetinaFace (Deng, Guo, Ververas, et al., 2020) is its use of the Feature Pyramid Network (FPN), which allows it to handle faces of different sizes effectively. The FPN extracts features at multiple scales, forming a pyramid of feature maps (P2, P3, P4, P5, and P6) that capture different levels of detail, essential for detecting faces of varying sizes within a single image. RetinaFace also incorporates context modules, which significantly enhance its performance. These modules use deformable convolutional layers that replace standard convolutions, allowing the network to dynamically adjust its receptive fields. This flexibility is crucial for accurately detecting faces with different shapes and sizes (Zhu, Hu, Lin, & Dai, 2019). Additionally, RetinaFace captures surrounding contextual information, which helps in distinguishing faces from complex backgrounds and improves detection accuracy.(Qin, Bai, & Zhao, 2021)

A standout feature of RetinaFace (Deng, Guo, Zhou, et al., 2020) is its facial landmark localization capability. The model predicts key facial points such as the eyes, nose, and mouth, providing additional supervision signals that enhance detection accuracy, especially under occlusion or varying poses. This is complemented by dense regression, which predicts precise bounding box coordinates for face detection, ensuring high precision. RetinaFace employs two different backbone networks in this study: ResNet-152 and MobileNetV1 0.25. ResNet-152 is a deep convolutional neural network with 152 layers, known for its ability to capture complex patterns and achieve high accuracy. MobileNetV1 0.25, (Howard et al., 2019) on the other hand, is a lightweight network designed for efficiency, making it suitable for real-time applications on resource-constrained devices.

Based on Fig 2, training the RetinaFace model for face detection, the first step divides the dataset into training, testing, and validation sets. Then, the data will go into the RetinaFace model. During training, we will use the training and validation sets. The model will extract facial features and then use the Feature Pyramid Network (FPN) to obtain features at different scales. After that, the context module will enhance contextual information to improve face detection. The model will also have a multi-task head to detect faces and predict their locations and landmarks. The loss value will be used to train the model for all these tasks, and during testing, the model will detect faces and predict their locations and landmarks. This predicts the positions of key facial landmarks, such as the eyes, nose, and mouth. These landmarks provide additional supervision signals, improving detection accuracy, especially under occlusion or varying poses (Zhang, Zhang, Li, & Qiao, 2016). The loss value will be used to train the model for all these tasks, and during testing, the model will detect faces and predict their locations and landmarks, which measures how well it

Journal of Computer Networks, Architecture and High Performance Computing
Volume 7, Number 3, July 2025
https://doi.org/10.47709/cnahpc.v7i3.6161

Submitted : Jun 19, 2025
Accepted : Jul 7, 2025
Published : Jul 10, 2025

performs in face classification, landmark regression, and bounding box regression. Lower loss indicates better model performance, and the model is updated to minimize this loss.

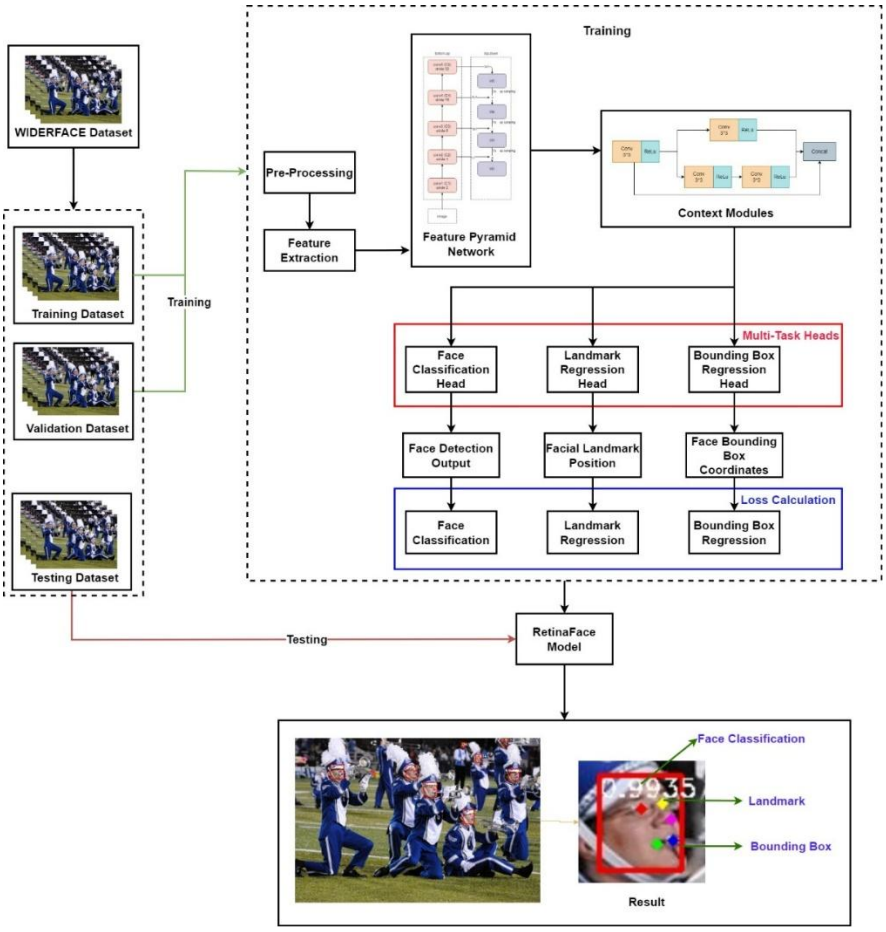

Fig. 2. Flow process training and testing model

The testing dataset is used to assess the model's performance after training. This evaluation helps determine how well the model can detect faces under various conditions that might not have been present in the training dataset. RetinaFace stands out due to its use of FPN for handling faces of various sizes, context modules for understanding the surrounding context, and precise facial landmark localization, which enhances detection accuracy. The study compares two backbone networks: ResNet-152, which is deep and capable of capturing complex patterns for high accuracy, and MobileNetV1 0.25, which is lightweight and efficient, suitable for real-time applications on devices with limited resources. In the training phase, images are preprocessed by cropping the face regions using RetinaFace, followed by feature extraction using a CNN. The extracted features are normalized and passed through fully connected layers. A critical step involves calculating cosine similarity, where the main difference between ArcFace and SphereFace emerges. ArcFace adds an angular margin to the cosine similarity calculation, increasing the separation between classes, while SphereFace scales the angle by a factor, also enhancing class separation. Both methods use a softmax function to convert the results into class probabilities, and a loss function guides the model's learning. Finally, the model outputs class predictions, identifying the faces in the input images.

RESULT

Training Process

The training process is carried out by implementing various training scenarios to find the best hyperparameters that produce the most optimal model. From Fig 3 presents the training performance of RetinaFace models using two different backbone architectures, MobileNet (Mnet) and ResNet. The represented models (Mnet 1-7 and ResNet 1-7)

Journal of Computer Networks, Architecture and High Performance Computing
Volume 7, Number 3, July 2025
https://doi.org/10.47709/cnahpc.v7i3.6161

Submitted : Jun 19, 2025
Accepted : Jul 7, 2025
Published : Jul 10, 2025

are the best performing models from each backbone based on different hyperparameter settings.

(a) (b)

Fig. 3. the RetinaFace training results using (a) the ResNet-152 (Resnet), (b)MobileNet V1 (0.25) (Mnet)

Fig 3. shows the RetinaFace training results using the MobileNet V1 (0.25) (Mnet) and ResNet-152 (Resnet) backbones. Overall, in the process of building the RetinaFace model with the MobileNet V1 (0.25) backbone, the best model was obtained by using the optimizer = SGD parameters, learning_rate = 0.001, batch_size = 2 and epoch = 120 with a result of 88.20% (see Fig 3(b)). While RetinaFace with the ResNet-152 backbone managed to obtain the best model by using optimizer parameters = SGD, learning_rate = 0.001, batch_size = 2 and epoch = 120 obtained a result of 92.35%. (see Fig 3(a)) Factors that affect the training process are seen from the use of parameters during the process of each training. The results of this study show that the epoch and batch_size parameters affect the accuracy of the model. This shows that increasing the number of epochs with a small batch size can provide more optimal results. There is a decrease in accuracy when the epoch is decreased to 60 or the batch_size is increased to 4. This indicates that for the ResNet-152 backbone, increasing the batch_size or reducing the number of epochs can reduce the effectiveness of the model in detecting faces.

Table 1 show the results of training. For the MobileNet V1 0.25 backbone (Mnet 1 to Mnet 7), the validation accuracy ranged from 80% to 82%, with a training time between 10.137 to 26.099 seconds. The highest validation accuracy was achieved by MNet 5, which had a validation accuracy of 82.88% and a validation loss of 5.00. The MobileNet models generally showed lower training times compared to the ResNet-152 models, making them more efficient in terms of training duration. However, their validation accuracy is slightly lower than that of the ResNet-152 model. The ResNet-152 backbone (ResNet 1 to ResNet 7) shows a wider range of validation accuracy, from 80% to 89%, with a much higher training time, ranging from 41.317 to 91.872 seconds. The highest validation accuracy was achieved by ResNet 7, which had a validation accuracy of 89.04% and a validation loss of 3.78, despite taking 42.249 seconds for training. Despite the longer training time, the ResNet-152 model generally outperformed the MobileNet V1 0.25 model in terms of validation accuracy.

Comparative analysis of the performance of RetinaFace models using the ResNet152 backbone and MobileNet v1 0.25, there are some significant differences. In the model with the ResNet152 backbone, the graph shows that the training and validation accuracy reaches a high level quickly and then stabilizes, with the validation accuracy reaching 89,04% and the validation loss of 3,78. In contrast, the model with the MobileNet v1 0.25 backbone also shows a rapid increase in training and validation accuracy, but the level of accuracy achieved is slightly lower with validation accuracy reaching 82,88% and validation loss of 5,00. The accuracy difference between the two models is 6,16%, while the loss difference is 1.22, indicating that the model with the ResNet152 backbone is superior in terms of accuracy and more effective in minimizing errors. Overall, the ResNet152 backbone is more suitable for scenarios that require high accuracy and sufficient computing resources, while the MobileNet v1 0.25 backbone is more efficient and suitable for devices with limited computing power, while still providing sufficient accuracy.

Table 2
Comparison Result of training model between Mnet and Resnet

Scenarios	RetinaFace + Backbone	Opt	Epoch	Val Acc	Vall Loss	Time
1	Mnet 1	Adam	100	82,20%	5,25	17666
2	Mnet 2	Adam	60	82,31%	5,23	10617
3	Mnet 3	Adam	80	81,22%	5,12	14262
4	Mnet 4	SGD	60	81,25%	5,42	11023
5	**Mnet 5**	**SGD**	**120**	**82,88%**	**5,00**	**26099**

Journal of Computer Networks, Architecture and High Performance Computing
Volume 7, Number 3, July 2025
https://doi.org/10.47709/cnahpc.v7i3.6161

Submitted : Jun 19, 2025
Accepted : Jul 7, 2025
Published : Jul 10, 2025

6	Mnet 6	SGD	100	81,97%	5,20	20996
7	Mnet 7	SGD	80	82,77%	5,26	16805
1	Resnet 1	Adam	100	83,09%	5,37	76286
2	Resnet 2	Adam	60	81,16%	5,63	45816
3	Resnet 3	Adam	80	82,47%	5,51	61394
4	Resnet 4	SGD	60	84,81%	4,93	43173
5	Resnet 5	SGD	120	88,07%	3,77	91872
6	Resnet 6	SGD	100	88,39%	3,79	54005
7	**Resnet 7**	**SGD**	**80**	**89,04%**	**3,78**	**42249**

MobileNet V1 is superior in training speed at 4 minutes 36 seconds, compared to ResNet-152 requiring 10 minutes 22 seconds. This study concludes that where ResNet-152 is more applicable for accuracy-prioritizing conditions, while MobileNetV1 (0.25) is more efficient for speed-driven applications.

Testing process

After the training process is complete, testing is performed using the best model obtained to obtain the Mean Average Precision (mAP) value. This best model is selected based on the optimal performance shown during training with various combinations of hyperparameters and different backbone architectures. This testing process is important to evaluate how well the model can detect and recognize objects in the test data that were not seen before, and the resulting mAP value will give an idea of the accuracy and effectiveness of the model in the object detection task.

Table 3
Testing RetinaFace

Backbone	Easy	Medium	Hard
ResNet	92,78%	89,95%	74,24%
MobileNet	89,47%	86,04%	67,30%

Table 3 shows the test results performed on the two best models, where the Widerface test dataset is divided into three classes: easy, medium, and hard. The Mean Average Precision (mAP) value obtained from this test shows the difference in model performance in each class. In the easy class, the model with the ResNet152 backbone achieved an mAP of 92.78%, while MobileNet v1 0.25 achieved an mAP of 89.47%, with a difference of 3.31%, indicating that both models are very effective in detecting faces in simple conditions, although ResNet152 is slightly superior. In the medium class, ResNet152 obtained an mAP of 89.95%, while MobileNet v1 0.25 achieved 86.04%, with a difference of 3.91%, indicating that both models are still quite accurate in more challenging detection conditions, with ResNet152 still performing better. In the hard class, ResNet152 achieved a mAP of 74.24%, while MobileNet v1 0.25 achieved 67.30%, with a difference of 6.94%, reflecting the great challenge of detecting faces in highly complex and diverse conditions. ResNet152 shows better performance in all classes compared to MobileNet v1 0.25, with the mAP difference getting larger as the detection difficulty level increases, confirming the superiority of ResNet152 in various conditions, especially in more complex and challenging situations.

Besides the lack of detection from using MobileNet V1 (0.25) which detects 4 less (~8%) faces compared to using ResNet-152, we can see that the inference time from using MobileNet V1 (0.25) outperforms the speed from using ResNet-152. As we can see in the Table 4, that the inference time shown in ms, using MobileNet V1 (0.25) can be 25x faster than ResNet-152 which is shown from testing using a CPU (12th Generation Intel® Core™ i9-12900K 24 Core) at HD resolution only. The smallest inference time difference still shows significant results, with MobileNet V1 (0.25) about 2.5x faster than ResNet-152 when using a single GPU (NVIDIA RTX 4090 24GB) at VGA resolution.

Journal of Computer Networks, Architecture and High Performance Computing
Volume 7, Number 3, July 2025
https://doi.org/10.47709/cnahpc.v7i3.6161

Submitted : Jun 19, 2025
Accepted : Jul 7, 2025
Published : Jul 10, 2025

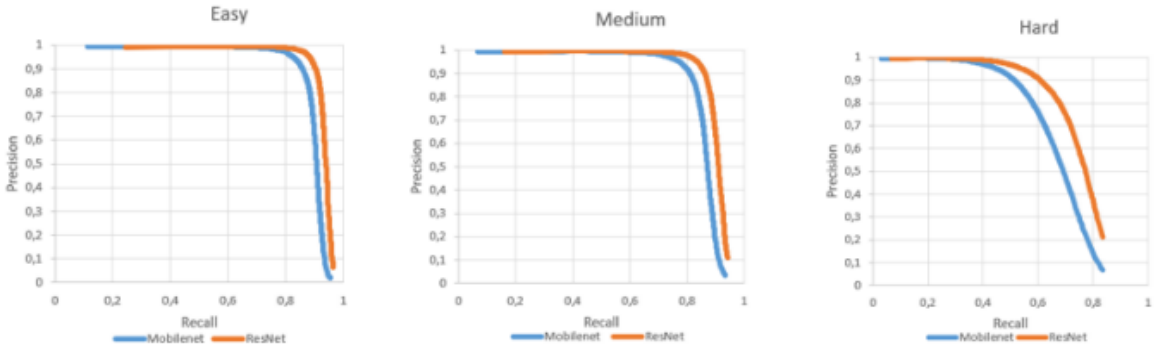

Fig. 4. Comparison of mAP in Easy, Medium, and Hard scenarios on RetinaFace using ResNet-152 backbone and MobileNet V1 (0.25)

Table 4
Comparison inference time in difference resolution

Backbones	FHD	HD	VGA	nHD
ResNet-152 (GPU)	915,2	69,3	38,5	33,8
ResNet-152 (CPU)	6195,6	4905,6	1228,8	666,6
MobileNet-0.25 (GPU)	113,9	12,1	15,5	8
MobileNet-0.25 (CPU)	1069,8	190,4	74,8	40,5

In this evaluation, face detection is additionally executed under a range of conditions. The scale is obtained based on two things, namely the distance of the object to the camera (1-24m) and the image resolution (240x320, 480x640, 720x960, 1080x1440, 1440x1920, 2160x2880). Following the evaluation, the quantity of faces identified is 163 when utilizing the ResNet-152 backbone and 162 when employing the MobileNetV1 (0.25) backbone. These results show that the ResNet-152 backbone detects more faces. RetinaFace's ability to detect faces in multiscale conditions is well-established. However, it's important to note that the quality of images captured by cameras and influenced by lighting can also impact RetinaFace's performance in face detection. In addition, tests were conducted to analyze the detected facial landmarks with various scale conditions. Testing uses datasets with front-facing, left-side and right-side face images with landmark facial features of the left eye, right eye, nose, left mouth corner and right mouth corner. The test results show that RetinaFace with ResNet-152 backbone is able to detect all five facial landmarks (left eye, right eye, nose, and two corners of the mouth) accurately for front-facing faces up to 17 meters away with a minimum image resolution of 720×960. At distances above 18 meters, landmark detection gradually degrades and fails completely at distances ≥ 23 meters. For faces with left and right side orientations, the system was consistently only able to detect three landmarks, which were most likely the eyes, nose, and one corner of the visible mouth. This shows that the pose of the face affects the system's ability to recognize landmarks thoroughly. In conclusion, increasing the image resolution significantly aided detection at longer distances, indicating the model's high dependency on the quality of the input visuals. MobileNetV1, on the other hand, showed random detection at distances of up to 17 meters, despite using high resolution. For left and right side poses, both models showed similar limitations with only three landmarks detected, which is most likely due to some facial features not being visible. Overall, ResNet-152 excelled in terms of consistency and resilience to image quality degradation, although MobileNetV1 showed higher training and inference time efficiency.

Testing the Loss Function

Table 5 summarizes the experiments conducted to identify the best-performing models for each combination of loss function and backbone network. For the ArcFace loss with the MobileNet (MNet) backbone, various hyperparameter settings were tested, including optimizers (SGD), epochs (60, 80, 100, and 120), learning rates (ranging from 0.0001 to 0.1), and batch sizes (2 and 4). Among these, the best-performing model was *ArcFace 2*, which used a batch size of 32 and a learning rate of 0.01, achieving a validation accuracy of 89.79% and a validation loss of 0.99.

Journal of Computer Networks, Architecture and High Performance Computing
Volume 7, Number 3, July 2025
https://doi.org/10.47709/cnahpc.v7i3.6161

Submitted : Jun 19, 2025
Accepted : Jul 7, 2025
Published : Jul 10, 2025

Table 5.
Loss function testing on the RetinaFace model

Loss Function	Retina+	Batch Size	Learning Rate	Val Acc	Val Loss
ArcFace 1	Mnet	32	0,1	85,41%	4,16
ArcFace 2	**Mnet**	**32**	**0,01**	**89,79%**	**0,99**
ArcFace 3	Mnet	32	0,001	31,87%	11,02
ArcFace 4	Mnet	64	0,1	89,95%	2,93
ArcFace 5	Mnet	64	0,01	82,14%	1,50
ArcFace 6	Mnet	64	0,001	25,44%	14,16
ArcFace 7	**Resnet**	**32**	**0,1**	**86,04%**	**4,00**
ArcFace 8	Resnet	32	0,01	78,75%	2,93
ArcFace 9	Resnet	32	0,001	32,29%	11,44
ArcFace 10	Resnet	64	0,1	85,04%	4,39
ArcFace 11	Resnet	64	0,01	62,94%	4,72
ArcFace 12	Resnet	64	0,001	19,41%	14,97
SphereFace 1	**Mnet**	**32**	**0,1**	**90,83%**	**0,13**
SphereFace 2	Mnet	32	0,01	84,58%	0,21
SphereFace 3	Mnet	32	0,001	44,16%	1,45
SphereFace 4	Mnet	64	0,1	87,72%	0,06
SphereFace 5	Mnet	64	0,01	71,20%	0,87
SphereFace 6	Mnet	64	0,001	57,81%	0,42
SphereFace 7	**Resnet**	**32**	**0,1**	**88,75%**	**0,16**
SphereFace 8	Resnet	32	0,01	82,91%	0,23
SphereFace 9	Resnet	32	0,001	42,91%	1,47
SphereFace 10	Resnet	64	0,1	89,06%	0,06
SphereFace 11	Resnet	64	0,01	69,86%	0,41
SphereFace 12	Resnet	64	0,001	51,11%	1,00

Meanwhile, RetinaFace model with ArcFace 2 loss shows the model effectively balanced learning and regularization, capturing essential features for accurate face recognition without overfitting. For ArcFace with ResNet152, the best model was ArcFace 7, with a batch size of 32 and a learning rate of 0.1, resulting in a validation accuracy of 86.04% and a validation loss of 4.00. The higher learning rate likely helped the deeper architecture of ResNet152 converge more effectively, although the higher validation loss suggests some instability or overfitting. SphereFace with MobileNet (SphereFace 1) yielded the best overall performance, achieving the highest validation accuracy of 90.83% and the lowest validation loss of 0.03.

SphereFace is particularly effective in distinguishing facial features when paired with MobileNet, benefiting from MobileNet's efficiency and SphereFace's discriminative power. SphereFace with ResNet152 (SphereFace 7) also performed well, with a validation accuracy of 83.75% and a validation loss of 0.21, indicating that ResNet152's deep feature extraction capabilities combined with SphereFace's loss function could capture intricate facial features, though slightly less effectively than MobileNet.

Overall, SphereFace with MobileNet (SphereFace 1) emerged as the best model, achieving the highest accuracy and lowest loss. ArcFace also performed well, particularly with MobileNet, but did not match the top performance of SphereFace. The experiments highlight the significant impact of backbone and loss function choices on model performance, with MobileNet generally outperforming ResNet152. This could be attributed to MobileNet's efficient and effective feature extraction combined with appropriate learning rates, whereas ResNet152's deeper architecture may require more careful tuning to avoid overfitting.

Table 6 show the results of testing different face recognition models: ArcFace and SphereFace, with configurations ArcFace 2, ArcFace 7, SphereFace 1, and SphereFace 7. The models are evaluated based on their performance in face recognition tasks using several metrics: precision (prec), recall (rec), accuracy (acc), F1-score (F1), and the area under the receiver operating characteristic curve (ROC).

Table 6.
Loss function preformances

Model	prec	rec	acc	F1	ROC
ArcFace 2	0,90	0,88	0,88	0,88	99

Journal of Computer Networks, Architecture and High Performance Computing
Volume 7, Number 3, July 2025
https://doi.org/10.47709/cnahpc.v7i3.6161

Submitted : Jun 19, 2025
Accepted : Jul 7, 2025
Published : Jul 10, 2025

ArcFace 7	0,88	0,84	0,84	0,84	96
SphereFace 1	0,97	0,96	0,96	0,96	100
SphereFace 7	0,95	0,94	0,94	0,94	100

ArcFace 2 shows strong performance with a precision of 0.90, recall of 0.88, accuracy of 0.88, F1-score of 0.88, and a high ROC score of 99. ArcFace 7, while still performing well, has slightly lower metrics with precision at 0.88, recall at 0.84, accuracy at 0.84, F1-score at 0.84, and a ROC score of 96. SphereFace 1 demonstrates the highest performance among all the models, with a precision of 0.97, recall of 0.96, accuracy of 0.96, F1-score of 0.96, and a perfect ROC score of 100. SphereFace 7 also performs exceptionally well, with a precision of 0.95, recall of 0.94, accuracy of 0.94, F1-score of 0.94, and a perfect ROC score of 100. Table 6 supports this conclusion by showing that SphereFace with datasets 1 and 7 yields higher overall evaluation metrics (precision, recall, accuracy, F1 score, and ROC) compared to ArcFace. Although MobileNet achieves higher evaluation scores on some datasets, the images illustrate that ResNet with SphereFace or ArcFace delivers better and more reliable face recognition results. In conclusion, the combination of ResNet with ArcFace or SphereFace is more dependable for face recognition, although multiscale techniques are necessary for more challenging conditions.

CONCLUSION

This research focuses on addressing the challenge of multiscale face detection by evaluating RetinaFace using two different backbone architectures: ResNet-152 and MobileNet V1 0.25. The results indicate that both backbones are effective in handling multiscale face detection, with each achieving the best accuracy for their respective architectures. The ResNet-152 model achieved a validation accuracy of 89.04% and mAP scores of 92.78%, 89.95%, and 74.24% for easy, medium, and difficult classes, respectively. On the other hand, the MobileNet V1 0.25 model achieved a validation accuracy of 82.88% and mAP scores of 89.47%, 86.04%, and 67.30%. Overall, the results indicate that despite the higher computational demands and longer training time, the ResNet-152 model, with its deeper network architecture, is better able to capture complex features, resulting in superior performance. In contrast, the MobileNet V1 0.25 model offers a more efficient training process and lower resource consumption, making it suitable for applications in resource-constrained environments. RetinaFace demonstrates proficiency in multiscale detection, which is crucial for improving the performance of face recognition models such as ArcFace and SphereFace. While RetinaFace effectively handles multiscale detection, ArcFace and SphereFace still face challenges in recognizing faces directly under multiscale conditions. Evaluation of different loss functions reveals that SphereFace's loss function performs better than ArcFace's loss function in our experiments. SphereFace achieved 94% accuracy, demonstrating its superior ability in handling face recognition tasks, while ArcFace achieved 88% accuracy. This research has evaluated RetinaFace's ability to detect faces at various scales and orientations, including tests on side-facing faces (varied poses) and at different distances, which partially reflect real-world conditions. However, this study has not covered other real-world scenarios such as poor lighting, masked faces, or extreme expressions. Therefore, further studies are recommended to test the performance of the model using datasets such as DARKFACE (low-light), MAFA (masked face), or AffectNet (extreme expressions) to ensure broader generalizability of the model in real environments.

REFERENCES

Alhanaee, K., Alhammadi, M., Almenhali, N., & Shatnawi, M. (2021). Face Recognition Smart Attendance System using Deep Transfer Learning. *Procedia Computer Science*, *192*, 4093–4102. https://doi.org/10.1016/j.procs.2021.09.184

Das, P., Asif, N. A., Hasan, M. M., Abhi, S. H., Jahin Tatha, M., & Bristi, S. D. (2022). Intelligent Door Controller Using Deep Learning-Based Network Pruned Face Recognition. *Proceedings of 2022 25th International Conference on Computer and Information Technology, ICCIT 2022*, (December), 120–124. https://doi.org/10.1109/ICCIT57492.2022.10056094

Deng, J., Guo, J., Ververas, E., Kotsia, I., & Zafeiriou, S. (2020). Retinaface: Single-shot multi-level face localisation in the wild. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 5202–5211. https://doi.org/10.1109/CVPR42600.2020.00525

Deng, J., Guo, J., Zhou, Y., Yu, J., 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: IEEE. https://doi.org/10.1109/CVPR42600.2020.00525

Filian, A., Istianto, B., & Kusuma, G. P. (2024). Image Enhancement using Convolutional Neural Network for Low

Journal of Computer Networks, Architecture and High Performance Computing
Volume 7, Number 3, July 2025
https://doi.org/10.47709/cnahpc.v7i3.6161

Submitted : Jun 19, 2025
Accepted : Jul 7, 2025
Published : Jul 10, 2025

Light Face Detection. *KESATRIA: Jurnal Penerapan Sistem Informasi (Komputer & Manajemen)*, *5*(1), 71–85.

Howard, A., Wang, W., Chu, G., Chen, L., Chen, B., & Tan, M. (2019). Searching for MobileNetV3 Accuracy vs MADDs vs model size. *International Conference on Computer Vision*, 1314–1324.

Kortli, Y., Jridi, M., Al Falou, A., & Atri, M. (2020). Face recognition systems: A survey. *Sensors (Switzerland)*, *20*(2). https://doi.org/10.3390/s20020342

Li, Q., Guo, N., Ye, X., Fan, D., & Tang, Z. (2020). *Video Face Recognition System: RetinaFace-mnet-faster and Secondary Search*.

Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., & Song, L. (n.d.). *SphereFace: Deep Hypersphere Embedding for Face Recognition*.

Ma, L., & Long, Z. (2023). A Face Recognition Method Using ResNet34 and RetinaFace. *Francis-Press*, *6*(10), 18–23. https://doi.org/10.25236/AJCIS.2023.061003

Nanni, L., Brahnam, S., & Lumini, A. (2023). Coupling RetinaFace and Depth Information to Filter False Positives. *Applied Sciences (Switzerland)*, *13*(5). Retrieved from https://www.mdpi.com/2076-3417/13/5/2987

Qin, J., Bai, H., & Zhao, Y. (2021). Multi-scale attention network for image inpainting. *Computer Vision and Image Understanding*, *204*(December 2019), 103155. https://doi.org/10.1016/j.cviu.2020.103155

Zaki, M. (2023). Deteksi Penggunaan Masker Pada Citra Menggunakan RetinaFace dengan MobileNetV2. *E-Proceeding of Engineering*, *10*(5), 4896–4902.

Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal Processing Letters*, *23*(10), 1499–1503. https://doi.org/10.1109/LSP.2016.2603342

Zhu, X., Hu, H., Lin, S., & Dai, J. (2019). Deformable convnets V2: More deformable, better results. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2019-June*, 9300–9308. https://doi.org/10.1109/CVPR.2019.00953