Article

Improving Accuracy of Face Recognition in the Era of Mask-Wearing: An Evaluation of a Pareto-Optimized FaceNet Model with Data Preprocessing Techniques

Damilola Akingbesote [1], Ying Zhan [1], Rytis Maskeliūnas [2] and Robertas Damaševičius [2,*]

1 Department of Multimedia Engineering, Faculty of Informatics, Kaunas University of Technology, 51368 Kaunas, Lithuania; damilola.akingbesote@ktu.edu (D.A.); ying.zhan@ktu.edu (Y.Z.)
2 Center of Excellence Forest 4.0, Faculty of Informatics, Kaunas University of Technology, 51368 Kaunas, Lithuania; rytis.maskeliunas@ktu.lt
* Correspondence: robertas.damasevicius@ktu.lt

Abstract: The paper presents an evaluation of a Pareto-optimized FaceNet model with data pre-processing techniques to improve the accuracy of face recognition in the era of mask-wearing. The COVID-19 pandemic has led to an increase in mask-wearing, which poses a challenge for face recognition systems. The proposed model uses Pareto optimization to balance accuracy and computation time, and data preprocessing techniques to address the issue of masked faces. The evaluation results demonstrate that the model achieves high accuracy on both masked and unmasked faces, outperforming existing models in the literature. The findings of this study have implications for improving the performance of face recognition systems in real-world scenarios where mask-wearing is prevalent. The results of this study show that the Pareto optimization allowed improving the overall accuracy over the 94% achieved by the original FaceNet variant, which also performed similarly to the ArcFace model during testing. Furthermore, a Pareto-optimized model no longer has a limitation of the model size and is much smaller and more efficient version than the original FaceNet and derivatives, helping to reduce its inference time and making it more practical for use in real-life applications.

Keywords: Pareto optimization; face recognition; mask-wearing; methods for segmentation of faces; synthetic datasets; FaceNet; face GAN; face CNN

1. Introduction

Facial recognition biometrics is one of the most popular uses of AI today [1]. First-generation uses include unlocking and mobile phone payment [2]; second-generation uses include camera monitoring and security systems [3]. However, as the COVID-19 pandemic spread, mask use became commonplace, and many people put on masks to stay safe and avoid infection. As the world fought the heinous COVID-19 pandemic, traditional face-to-face engagements such as board meetings, press conferences, product introductions, and even family reunions have shifted to a more digital presence through video conversations based on the Internet. However, the use of masks has negatively affected the performance of existing face-based identification systems, as these systems have not been trained on masked face images [4].

Facial recognition has become increasingly used by police in their investigations and responses. The Detroit Police Department used "facial recognition" to make 42 arrests; of these, only eight were valid [5]. This rate is quite low, and some individuals may even be questioned as a result of the police force's errors. There are, however, two primary causes: First, the cameras captured tens of millions of faces across the city of Detroit over a period of time, which is an enormous order of magnitude. Second, the main cause of this low accuracy rate is that in some environments, surveillance can record the criminal content of the offender, but routine crime is generally not recorded [6]. The reason for the incredibly

poor angle is that criminals frequently bow their heads, avoid looking directly into any camera equipment, and cover their face and facial characteristics, as well as the light and darkness. Therefore, light, angle, and causes of visual occlusion are the main limitations of face recognition technology. Finding ways to more effectively detect faces in specific scenarios is also the study focus of our project (wearing a mask).

Several fascinating studies have pointed out the limitations of currently available face recognition software. In one set of experiments, for example, it was discovered that most data packets tend to be more accurate for white male faces than for faces of people of color or females. In particular, there was a 10–100 times higher rate of false positives for Asian and African American faces than white ones in the database. Additionally, women are more likely than men to be misidentified [7]. Although deep learning-based algorithms have reached great performance, large-scale annotated data are difficult to obtain, and the enormous parameters make model implementation in embedded systems problematic. To overcome this, a dissimilarity-based strategy [8] that selects few but representative samples while taking data variety into account might be utilized.

Obviously, recent advances in this area have focused on trying to recognize an obscured face [9]. The technical requirements for these two cases could not be more dissimilar [10]. High requirements for AI recognition accuracy, typically beginning at four nines, are necessary in consumer scenarios with a focus on technical accuracy, such as unlocking mobile phones that involve financial payments [11]. For example, the Ruyi Pay PAD is a face-swiping payment device that has a cloud slave enhanced liveness detection module that has achieved 99.99% precision in the anti-living attack, as certified by the Bank Card Testing Centre [12]. In terms of security, the scope of available tech is now more of a priority [13]. To avoid being tracked by CCTV, for example, when captured by law enforcement and subsequently escaping from their custody, the vast majority of criminal suspects choose to cover some of their faces with hats or masks [14]. Several researchers have studied face occlusion technology for quite some time [15] with an eye on meeting the needs of real-world security scenarios [16,17], and have made several different attempts to make the technology more user-friendly, as the ethical use of face recognition in areas such as law enforcement investigations requires a set of clear criteria to ensure that this technology is trustworthy and safe [18]. Deep forgery detection techniques are learning-based systems that rely on data to a certain degree. Enhancing facial anti-spoofing databases is an excellent way to address the aforementioned issue. Yang et al. [19] proposed a face swapping system based on StyleGAN based on a feature pyramid network to obtain facial features and map them to the latent space of StyleGAN for whole face swapping while offering accurate information for deep forgery detection for ensuring the security of audiovisual systems. Their alternative strategy [20] included post-processing to improve the image's authenticity. To demonstrate the advantages of our proposed technique. Experiments demonstrated the usefulness of identity latent space and controllability, and the suggested network was able to deliver photo-level results while outperforming previous face swapping approaches.

As part of our study, we investigated the following scientific complexities. Preprocessing data and its effect on facial recognition systems: although many face recognition algorithms are "squealed down" on various evaluation lists [21], Google's brilliant FaceNet maintains a high accuracy rate in this area [22]. With FaceNet, the precision was 99.63% in the LFW dataset and 95.12% in the YouTube Faces DB dataset. Therefore, today due to COVID-19, everyone is required to wear protective masks. In this case, we are making changes to the structure and parameters, and adapting the dataset and image preprocessing of the FaceNet method, which previously had a very high accuracy rate in face recognition. The masked face dataset should be just as successful and find use in a wide range of contexts. It helps law enforcement identify criminals hiding behind masks. Since we are attempting to recognize faces while they are obscured by masks, we need to be able to do so with less information than is available in the standard face dataset. Since there are fewer data to learn from, accurate subject recognition becomes more difficult when the faces of the

subjects being studied are obscured or masked [23]. Therefore, it is necessary to identify an appropriate model and approach for this. Evaluations of recognition algorithms conducted after the pandemic reveal that the vast majority suffer from a decline in performance when faces are concealed. To this end, we plan to work on better masked-face models. There are numerous varieties of masks, each with its own level of occlusion. Another factor is the question of how to make better use of the data gathered from non-occlusion regions. Furthermore, when both the training and test images are masked, recognition performance decreases. This issue will be resolved soon.

The novelty of this paper is that it addresses a new challenge in face recognition technology caused by the widespread use of masks during the COVID-19 pandemic. With an increasing demand for technology that can identify individuals while wearing masks, this paper provides an overview of various standard face recognition technologies and the latest models for masked and unmasked face recognition.

The contribution of this paper is that it evaluates and compares the accuracy of these technologies and models and concludes that the best model to recognize individuals while wearing masks is FaceNet. The paper also presents a novel approach of using data preprocessing techniques such as 'CutMix' and 'mixup' and making changes to the model's parameters and structure to improve the accuracy of FaceNet. This research is a significant step forward in the field of face recognition technology and its application in the era of mask wear.

To further guide our research, we introduce the following *research questions*.

1. How can facial recognition algorithms, particularly FaceNet, be improved to accurately recognize faces that are partially or fully obscured by masks?
2. What techniques can be used to effectively utilize data from non-occluded regions of masked faces to improve recognition performance?
3. What approaches and models are most appropriate for recognizing masked faces?
4. How can recognition performance be improved in masked face datasets, including in scenarios where both training and test images are masked?
5. What solutions can be implemented to overcome the decline in recognition performance when faces are concealed?

The remainder of the paper is organized as follows. Section 2 presents an overview of the state-of-the-art. Section 3 discusses the methods for masked face detection, describes the construction of the datasets, explains the image augmentation methods, and describes the proposed method. Section 4 presents the results of experiments and presents the results of the ablation study. Section 5 provides answers to research questions and discusses the limitations of the study. Finally, Section 6 presents the conclusions.

2. State-of-the-Art Overview

This section first introduces some of the popular methods proposed for regular face recognition. Then, the state-of-the-art overview continues with masked face recognition.

In general, the techniques to be applied in face recognition are quite simple [24]. Face characteristics can be extracted using principal component analysis or linear discriminant analysis, and then for example basic Euclidean distance with a backpropagated neural network can be employed to categorize face subjects [25]. Often, such facial recognition algorithms are subject to face-presenting assaults (face-PA), including print, video playback, and rubber masks [26]. To address the aforementioned issues, Shekel et al. [27] built a unique deep neural network to deep-encode face areas. Others used PCA to minimize the dimensionality of feature representation while eliminating redundant and contaminated visual information [28]. Damer et al. [17] investigated the accuracy of face recognition and proposed the use of evolutionary algorithms to maximize the selection and prioritization of test cases, while machine learning guided the search for successful test cases. Yu et al. [29] built a face detection and recognition system based on neural computing paradigms and artificial neural methods. The research findings indicated that the approach had a greater detection accuracy and a faster computation. Tavakolian's team [30] suggested a

technique based on multiscale facial components and the characteristics of the Eigen/Fisher artificial neural network, aiming to reduce the components of the face of various resolutions, such as eyes, nose, mouth, and the complete face, according to their saliency, and then apply the principal component analysis of the subspace or linear analysis to generate a vector of facial characteristics. Soni et al. [31] suggested using preprocessing, cascade feature extraction, optimal feature extraction, and recognition as the four basic phases in convolutional processing. Deotale et al. [32] suggested an unsupervised neural network for the analysis of human activity as well as capturing faces. Gao et al. [33] established the idea of candidate areas for faces. Thilapi et al. [34] used the Ada boost face recognition system to scrutinize and retain all candidate regions. In [35], the candidate area was then classified using a small-scale CNN to determine whether it is a face and a medium-scale CNN to complete the categorization of all candidate regions. Moghadam et al. [36] introduced a new deep dynamic neural network to assess and extract three key aspects of facial expression movies. The suggested model of [37] had recurrent network benefits and can be used to assess the sequence and dynamics of information in moving faces.

Table 1 presents the recognition rate of different methods, tested on a regular face (without mask) dataset.

Table 1. Comparison of methods by recognition rates.

Method	Recognition Rate
Principal component analysis with ANN face recognition system [38]	95.45%
Deep Dense Face Detector [39]	91.79%
Radial Basis Neural Network [40]	97.56%
Convolutional Neural Network [41]	85.1%
Branch Convolutional Neural Network [42]	97.2%
Radial Basis Function Network [43]	97.75%
Region-based Convolutional Neural Networks [44]	99%
Rotations Invariant Neural Network [45]	90.6%
Gabor Wavelet [46]	99.94%
Weightless neural networks [47]	89.22%

Face recognition algorithms have evolved rapidly over the years due to a variety of causes [48]. Researchers have researched and created a variety of algorithms for occluded face identification in response to the unexpected aspects encountered in real world circumstances [49]. Zhao suggested a consistent subdecision network to obtain subdecisions that correspond to different facial areas and constraining subdecisions using weighted bidirectional KL divergence to focus the network on the upper faces without occlusion [50]. Fine-tuning current face recognition models on a dataset of masked faces is one of the most prevalent ways for masked face identification [51]. This strategy has been shown to improve the accuracy of masked face recognition, but it depends on the availability of a large and diverse collection of masked faces [52]. In [53], for example, scientists fine-tuned a face recognition model using a dataset of masked faces and reached a recognition accuracy of 92.5%.

A multitask learning architecture, in which a single network is trained to perform mask classification and face recognition tasks, is another technique successfully applied to mask recognition [54]. This strategy has also been found to increase the precision of mask face recognition by using mask information to aid in the recognition process [55]. For example, in [56], the authors suggested a multitask learning architecture and attained a recognition accuracy of 96.2%. Another approach is to employ generative models, such as Generative Adversarial Networks (GANs) [57] or Variational Autoencoders (VAEs) [58], to generate a varied set of masked faces for fine-tuning or training new models. A similar VAE-based method has also been shown to improve the accuracy of masked face recognition, although it depends on the availability of a large and diverse collection of unmasked faces [59]. For example, in [60], the authors used a GAN to generate masked faces and attained a recognition accuracy of 94.5%. A popular strategy is to remove the mask from

the face before applying a facial recognition model to the unmasked face [61]. This is accomplished by training the deep learning model to create an unmasked face from the input masked face. For example, Liu et al. [62] explored facial action recognition and face identification applications and discovered that both benefit from the encoding of face photos using Gabor wavelets. They performed dimensionality reduction and a linear discriminant analysis on the down-sampled data. Gabor wavelet faces can help to enhance discrimination. The closest feature space is expanded using several similarity measures. Hao et al. [63] proposed a uniform framework to identify both masked and unmasked faces. They proposed rectification blocks to correct features extracted by a cutting-edge classification method in both the spatial and channel dimensions to reduce the distance in the corrected feature space between a masked face and its mask-free equivalent.

Other approaches, such as Region-based CNN, Two-stream CNN, and 3D CNN, have been proven to increase the recognition accuracy of masked faces. Ref. [64] used a Region-based CNN to extract characteristics from the masked face, which were subsequently put into a fully connected layer for classification. On a masked face dataset, their approach achieved an accuracy of 96.2%. In [65], a 3D CNN was used to learn spatial–temporal information from the masked face, which was subsequently input into a fully connected layer for classification. On a masked face dataset, the approach attained an accuracy of 98.5%.

Although these approaches have increased masked face recognition accuracy, they still depend on the availability of a large and diverse collection of masked faces. In [66], to recognize faces of persons in mines, avalanches, under water, or other hazardous settings where their face may not be highly visible over the surrounding background, a lightweight CNN architecture was presented. The created model supports mobile devices as easily as possible. A box is displayed on the device's screen as the processing output at the face location. The findings demonstrate that the proposed lightweight CNN recognized human faces over a range of textures with an accuracy of more than 95%. In [67], face verification was performed using a hybrid method based on SURF and a neural network classifier. The entire system can be applied in real time to confirm individuals' IDs in congested areas such as airports. To boost overall performance, Fadi presented the Embedding Unmasking Model, which works on top of current face recognition algorithms [68]. The authors of [69] presented a dual-branch training technique to direct the model's attention to the top half of the face in order to extract strong features for masked face recognition. During training, the characteristics gained at the global branch's intermediate layers are supplied into the suggested attention module, which functions as a local branch and aids in resilience. The Masked Face Detection Dataset (MFDD), the Real-World Masked Face Recognition Dataset (RMFRD), and the Synthetic Masked Face Recognition Dataset (SMFRD) are the three types of masked face dataset proposed by Huang et al. [70], allowing for a more realistic evaluation of face classification algorithms. Cao et al. [71] proposed a new dataset called Diverse Masked Faces and advised that the YOLOX model be modified with a new composite loss that combines CIoU and alpha-IoU losses and retains both benefits. Wang's mask creation module [72], on the other hand, used facial landmarks to generate more realistic and reliable masked faces for training in addition to using existing datasets. The loss function search module aimed to find the best loss function for face recognition. Boutros et al. [68] presented an Embedding Unmasking Model (EUM) that would work over current face recognition methods. They also provided an innovative loss function, the Self-Restrained Triplet (SRT), which allowed the EUM to generate embeddings that resembled those of unmasked faces of the same individuals.

Table 2 presents the comparison of recognition accuracy of different methods, tested on a masked face dataset.

Table 2. Comparison of methods by recognition rates.

Method	Recognition Rate
MTArcFace [73]	99.78%
MTCNN + FaceNet [74]	64.23%
MaskNet [75]	93.80%
HSNet-61 [76]	91.20%
OSF-DNS [77]	99.46%
Attention-based [78]	95.00%
Cropping-based [79]	92.61%
FaceNet [52]	97.25%
LPD [80]	97.94%
MTCNN [81]	98.50%
Convolutional Neural Networks [82]	90.40%

In summary, biometric face recognition systems have gained widespread use in various applications such as security, access control, and identification [83]. However, there are several challenges and limitations that affect their performance and accuracy. One major challenge is the variability of lighting conditions, which can cause shadows, reflections, and other distortions that can affect the quality of captured images. This can lead to poor recognition performance, especially in outdoor environments [84]. Another challenge is the change in facial appearance over time, such as aging, hairstyles, glasses, and makeup. These variations can cause problems for systems that are trained on a single image of a person, leading to poor recognition accuracy [85]. Additionally, facial recognition systems can be affected by the presence of occlusions such as masks, hats, and scarfs, which can make it difficult to accurately identify a person [86].

3. Materials and Methods

3.1. Datasets Characteristics

This research conducted its experiments on a combination of two original datasets. The first dataset is CASIA [87,88], which contains 492,832 face images with 10,585 identities. Figure 1 illustrates samples of masked faces in the CASIA dataset.

Figure 1. CASIA dataset images samples.

The second dataset is a VGG-Face [87] dataset. It contains 2,024,897 images of 8631 identities. Examples for images of the VGG-Face are presented in Figure 2. The VGG-Face dataset was combined with the CASIA dataset used for the training, validation, and testing phases.

Figure 2. VGG-FACE dataset images samples.

This combined dataset as shown in Figure 3 is used for the training, validation, and testing phases.

Figure 3. Combined dataset images samples.

3.2. Image Augmentation

Image augmentation is a technique used to increase the diversity of a dataset by applying various types of image transformation to existing images [89]. This technique is particularly useful in the face recognition task, as it helps to improve the robustness and generalization of a model by exposing it to a wider range of variations in the input data. The benefits of image enhancement for the face recognition task are as follows. Brightness and contrast adjustments can help a model handle variations in lighting conditions, which can be a major challenge in face recognition. Rotation, scaling, and flipping can help a model handle variations in pose, which can make it difficult for a model to recognize a face from different angles. Image warping can help a model to handle variations in facial expressions,

which can make it difficult for a model to recognize a face with different emotions. Adding masks or glasses can help a model to handle variations in occlusion, which can make it difficult for a model to recognize a face with masks or glasses on. By exposing a model to a wider range of variations in the input data, image augmentation can help to improve the generalization of the model and make it more robust to unseen variations.

First, we removed the data without masks from the dataset. Image augmentation is a very important step in our image preprocessing step. Unlike other projects, the purpose is to enhance the recognition ability of the image, such as strengthening the contrast and strengthening the light to make the image clearer. We aimed at making the image difficult to recognize, such as flipping the image, reducing the light intensity, etc., so that the image is not easy to recognize. Only in this way could we verify whether our face recognition algorithm can actually detect the part of the face and mask.

Image enhancement was performed using the Albumentation library [90] using functions such as transpose, horizontal flip, vertical flip, shift scale rotation, change in hue and saturation, and random adjustment of brightness and contrast. Figure 4 illustrates sample training images after augmentation.
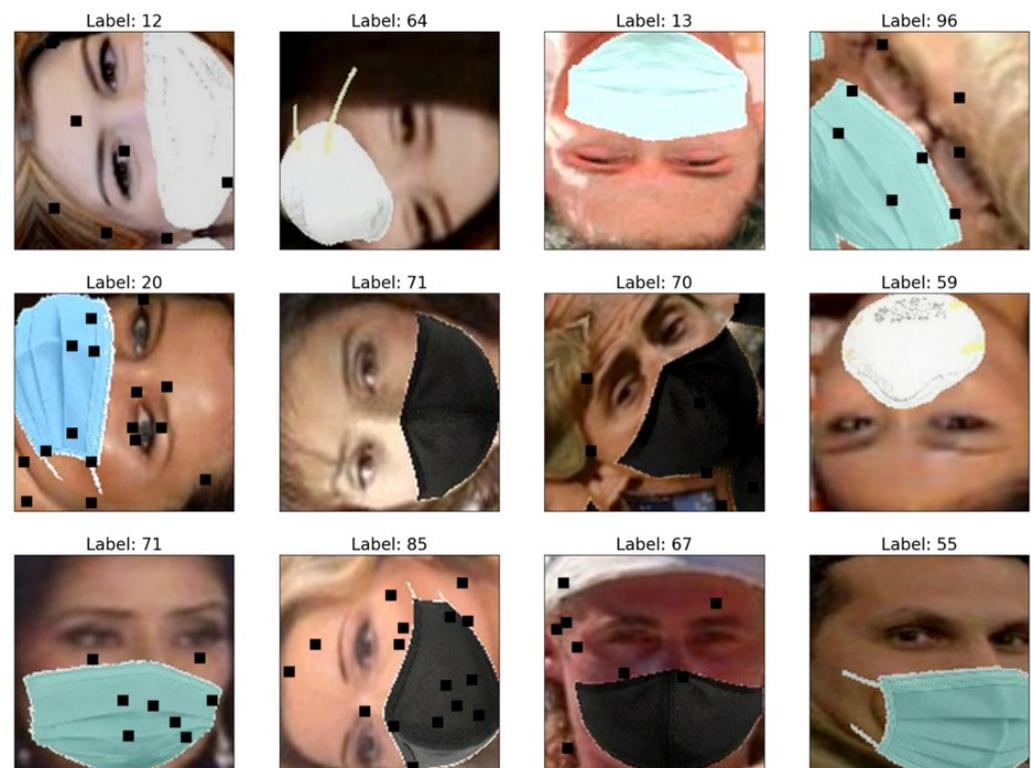

Figure 4. Sample of training images after augmentation.

In order to better improve the performance of the model, we also added the CutMix and mixup augmentation so as to perform some additional data preprocessing steps before actually training the model. We clipped and pasted random patches between the training images while using the CutMix augmentation. Depending on the size of the patches in the photos, the ground truth labels were blended. By forcing the model to concentrate on less discriminative aspects of the object being classified, CutMix improves the localization ability and is thus also well suited for tasks like object identification.

In Mixup augmentation, the pictures and labels of two samples are linearly interpolated to combine them. Mixup samples are poor at tasks such as image localization and object detection due to their unrealistic output and label ambiguity. Furthermore, a random patch from an image is zeroed out in a localized dropout technique known as "cutout augmentation" (replaced with black pixels). Reduced information and regularization capacity

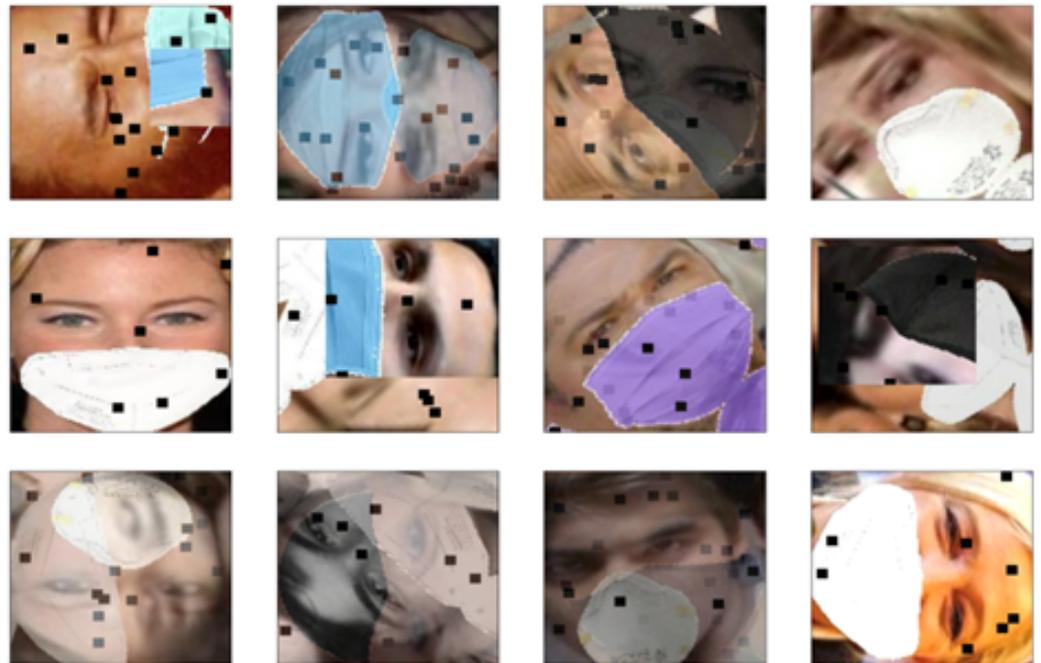affect cutout samples. Figure 5 shows a sample of images from our dataset after CutMix and Mixup augmentation.

Figure 5. Illustration of Cutmix and MixUp image augmentations.

3.3. Denoising of Images

The goal of denoising is to remove the noise from the image and recover the original image. This can be achieved by using a denoising model, which is trained to remove the noise added during the noise injection step. Image noise, which is usually electrical noise, is a random variation in the brightness or color information in photographs. It can be produced by the image sensor and circuitry of a scanner or digital camera [90]. Noise invariably reduces the quality of images, resulting in a decrease in visual image quality [91]. It should be noted that the impact of image noise manipulation on gaze distribution was mainly determined by noise intensity rather than noise type [92].

There are several types of noise models that can be used to add noise to images in deep learning, such as Gaussian noise, salt and pepper noise, and Poisson noise.

Gaussian noise is the most used noise model, which is characterized by a normal distribution with a mean of zero and a standard deviation of σ. It can be mathematically represented as follows:

$$y = x + N(0, \sigma^2) \tag{1}$$

where x is the original image and y is the noisy image.

Salt and pepper noise is a type of noise that randomly sets certain pixels to the minimum or maximum value. The mathematical representation of salt and pepper noise can be represented as:

$$y = x + Bernoulli(p) \times (rand - 0.5) \tag{2}$$

where x is the original image, y is the noisy image, p is the probability of noise, and $rand$ is a random number between 0 and 1.

Poisson noise is a type of noise that is typically added to images taken by sensors such as cameras. Poisson noise can be mathematically represented as:

$$y = x + Poisson(\lambda x) \tag{3}$$

where x is the original image, y is the noisy image, and λ is the intensity of the noise.

To test for model robustness, we added Gaussian noise from the Albumentations library to the training dataset to reduce the image features and see how well our model performs. Noise was added to the images during the training phase and this was usually performed by applying a noise model to each image in the training dataset. The noise model was usually applied to each pixel of the image, with the goal of simulating the noise that would be present in real-world scenarios.

A loss function was used to measure the difference between the denoised image and the original image. The most commonly used loss function for image denoising is the mean squared error (MSE), which is defined as:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{4}$$

where y is the original image, $\hat{y}$ is the image denoised, and n is the number of pixels in the image.

Finally, the denoising model was optimized by minimizing the loss function using gradient descent, a method for finding the minimum of a function by iteratively moving in the direction of steepest descent as defined by the negative of the gradient:

$$\theta_i = \theta_{i-1} - \alpha \frac{\partial J(\theta)}{\partial \theta} \tag{5}$$

where θ is the parameter to optimize, $J(\theta)$ is the loss function, α is the learning rate, and $\frac{\partial J(\theta)}{\partial \theta}$ is the loss function gradient with respect to the parameters. The negative of the gradient is used to find the steepest descent direction, and the learning rate α determines the step size to take in that direction. The process is repeated until the function reaches a minimum.

In general, the goal of adding noise to images in deep learning is to improve the robustness of the model by making it more resistant to the noise present in real-world scenarios. This is achieved by training the model on noisy images and by using denoising techniques to remove the noise during the testing phase. The mathematical formulas and optimization techniques used in this process help minimize the difference between the denoised image and the original image, thus recovering the original image.

3.4. Up-Scaling of the Resolution

We have used a model called a "super-resolution generative adversarial network" (SRGAN) to evaluate the effects of higher resolution. The loss function in this model consists of two components: adversarial loss and content loss. Adversarial loss aims to produce realistic images that resemble the original, while content loss makes sure that the generated image has the same features as the low-resolution original. The loss function incorporates both adversarial and content loss using a perceptual loss function. A discriminator network, trained to differentiate between high-resolution images generated and true photorealistic images, drives the solution towards the manifold of natural images through adversarial loss [92].

In SR-GAN, a generator network is trained to learn the mapping between LR and HR images. The generator network is a convolutional neural network (CNN) that takes an LR image as input and produces an SR image as output. A discriminator network is also trained to distinguish the SR image generated by the generator from the HR image. The generator and discriminator networks are trained simultaneously in an adversarial manner. The generator network is trained to minimize the difference between the SR image and the HR image, while the discriminator network is trained to maximize this difference.

The adversarial loss function used for this purpose is typically the binary cross-entropy loss function, given by:

$$L_{adv} = -(y \times log(D(x_{hr})) + (1-y) \times log(1 - D(G(x_{lr}))))$$ (6)

where $G(x_{lr})$ represents the generated SR image, $D(x_{hr})$ represents the output of the discriminator network for the HR image, and y is the label (1 for real images and 0 for generated images).

Additionally, a Mean Square Error (MSE) loss function is also used to measure the difference between the generated image and the original image:

$$L_{mse} = \frac{1}{n} \times \sum (x_{hr} - G(x_{lr}))^2$$ (7)

where x_{hr} represents the original HR image and x_{lr} represents the LR input image. These two loss functions are combined to create a total loss function, which is used to train the generator network. An optimizer algorithm such as Adam, SGD, or RMSprop is used to adjust the parameters of the generator and discriminator networks during training.

The hyperparameters of SR-GAN are summarized in Table 3. The performance of SR-GAN was evaluated on several publicly available datasets. The results show that SR-GAN was able to produce SR images with a significant improvement in quality compared to We implemented the SRGAN tensorflow model used in [92] and trained it on our commercial dataset of VGG-masking face and the CASIA masking face dataset to further improve the resolution of the images and obtain a better generalization and performance of the model.

Table 3. Hyperparameters of SR-GAN.

Hyper Parameter	Value
Batch size	8
Epoch	5001
Seed	2020
Optimizer	Adam
Loss function	Perpetual loss

3.5. FaceNet Architecture

The original FaceNet architecture is a convolutional neural network (CNN), which has been trained to map face images to a compact and meaningful representation in Euclidean space, where the distances between points indicate the similarity between faces [60].

The architecture of FaceNet can be divided into three main parts: The first part of the model is the convolutional neural network (CNN) that is used to extract features from the input face image. This CNN is typically based on an architecture called Inception, which is a variant of GoogleNet. The Inception architecture uses a combination of 1×1, 3×3, and 5×5 convolutional filters, as well as max pooling layers, to extract features from the input image. The output of the Inception CNN is a 512-dimensional feature vector. The second part of the model is the embedding layer, which is a fully connected (FC) layer that maps the 512-dimensional feature vector to a 128-dimensional embedding vector. The embedding vector is used to calculate the distance between faces. The embedding layer is defined as:

$$E(x) = W_2 \times max(0, W_1 \times x + b_1) + b_2$$ (8)

where x is the 512-dimensional feature vector, W_1 and W_2 are the weight matrices, and b_1 and b_2 are the bias vectors.

The third part of the model is the triplet loss function, which is used to train the model. The triplet loss function is defined as:

$$L(A, P, N) = max(\|E(A) - E(P)\|^2 - \|E(A) - E(N)\|^2 + \alpha, 0)$$ (9)

where A, P, and N are anchor, positive, and negative images, respectively, $E(A)$, $E(P)$, and $E(N)$ are embedding vectors of anchor, positive, and negative images, respectively, and α is a margin constant. The triplet loss function is used to ensure that the embedding vectors of the same person are closer to each other than the embedding vectors of different people.

3.6. Pareto-Optimized FaceNet Architecture

The Pareto-optimized FaceNet architecture is a variant of the original FaceNet model, which seeks to balance multiple objectives, such as accuracy, computational complexity, and memory requirements. To achieve this balance, the Pareto-optimized FaceNet architecture is designed based on a Pareto frontier, which is the set of solutions that cannot be further improved in one objective without degrading another objective.

Let $\mathcal{A}$ be the set of all possible architectures for FaceNet, and let $f_1(a)$, $f_2(a)$, and $f_3(a)$ be the accuracy, computational complexity, and memory requirements of the architecture $a \in \mathcal{A}$, respectively. An architecture a_1 is said to dominate another architecture a_2 if $f_1(a_1) \geq f_1(a_2)$, $f_2(a_1) \leq f_2(a_2)$, and $f_3(a_1) \leq f_3(a_2)$, with at least one inequality being strict. The Pareto-optimized FaceNet architecture is obtained from the set of Pareto optimal solutions, defined as:

$$\mathcal{P} = a \in \mathcal{A} : | : \nexists a' \in \mathcal{A} : \text{s.t.} : a' : \text{dominates} : a \tag{10}$$

The algorithm for Pareto optimization of the FaceNet architecture (see Algorithm 1) is based on the concept of multi-objective optimization using genetic algorithms. Note that this is just one possible approach to Pareto optimization, and other optimization algorithms may be used as well.

Algorithm 1 Pareto optimization of FaceNet architecture

Require: *population_size*, *max_generations*, *mutation_rate*, *crossover_rate*
  1: Initialize *population* $\leftarrow$ RANDOMPOPULATION(*population_size*)
  2: *generation* $\leftarrow$ 0
  3: **while** *generation* < *max_generations* **do**
  4:     Evaluate the objectives $f_1(a)$, $f_2(a)$, and $f_3(a)$ for all $a \in$ *population*
  5:     *pareto_front* $\leftarrow$ PARETOFRONT(*population*)
  6:     *offspring* $\leftarrow \varnothing$
  7:     **while** |*offspring*| < |*population*| **do**
  8:         $parent_1$, $parent_2 \leftarrow$ SELECTION(*population*, *pareto_front*)
  9:         $child_1$, $child_2 \leftarrow$ CROSSOVER($parent_1$, $parent_2$, *crossover_rate*)
10:         $child_1 \leftarrow$ MUTATION($child_1$, *mutation_rate*)
11:         $child_2 \leftarrow$ MUTATION($child_2$, *mutation_rate*)
12:         Add $child_1$, $child_2$ to *offspring*
13:     **end while**
14:     *population* $\leftarrow$ *offspring*
15:     *generation* $\leftarrow$ *generation* $+ 1$
16: **end while**
17: **return** $pareto_front$

The algorithm starts by initializing a random population of FaceNet architectures (line 2). Then, for a predefined number of generations (lines 3–14), the algorithm evaluates the objectives for each architecture (line 4) and computes the Pareto front (line 5). It generates offspring through selection, crossover, and mutation operations (lines 7–12), and the offspring become the new population for the next generation (line 13). Once the algorithm reaches the maximum number of generations, it returns the final Pareto front (line 15).

The Pareto-optimized FaceNet architecture includes the components from the original FaceNet architecture, such as the Inception CNN, the embedding layer, and the triplet

loss function. However, the specific structure of the Inception CNN and the embedding layer may be altered to achieve a balance between the objectives. For example, a Pareto-optimized FaceNet architecture (see Figure 6) might have a reduced number of layers, filters, or neurons in the Inception CNN and the embedding layer. This would result in a trade-off between accuracy, computational complexity, and memory requirements, achieving a balance that is optimal according to the Pareto frontier.
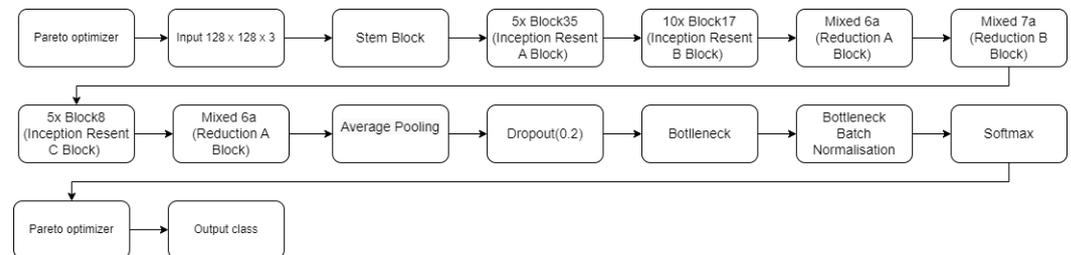

Figure 6. Pareto-optimized FaceNet architecture.

4. Results

4.1. Experimental Results

All experimental trials have been conducted on an Apple macbook pro M1 device equipped with the M1 8 core processor and 8 GB of RAM. The Jupyter Notebook software was chosen for conducting experiments and implementing them in this research. In addition, the models were trained and tested using the tensorflow and keras python packages. As described in Section 3, the datasets used are a combination of the CASIA-WebFace+masks image dataset, which contains 492,832 face images with 10,585 identities (10,585 different classes), and the VGG-Face dataset, which included a total of 16,903 masked facial images (2622 identity classes).

The models were split into training, validation, and test sets using stratified K-folds cross-validation and the 60%-20%-20% rule. The training set was first preprocessed using Albumentations and then CutMix and MixUp to further improve the accuracy of the model.

To evaluate the performance, size, and computation time of the different algorithms, performance metrics had to be investigated through this research. The performance metrics used in this research are accuracy, flopping, and model size.

Classification Report: A classification algorithm's predictive accuracy can be evaluated with the use of a classification report. It shows the ratio of accurate to inaccurate forecasts. In particular, the metrics of a categorization report can be predicted using True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN).

Accuracy: The ratio of correctly classified data instances over all data instances is known as accuracy.

$$Accuracy = \frac{TN + TP}{TN + FP + TP + FN} \tag{11}$$

FLOPS, or floating-point operations per second, is a metric that determines a microprocessor's capability to carry out floating-point calculations in one second.

Model size: Model size (number of parameters) is related to performance, and it is the size of the model after training. In this research, our model was measured in megabytes (MB).

4.2. Model Analysis and Comparison

The initial experiments performed involved six different models, namely: Arc-Face ResNet50, Inception ResnetV1, tensorflow densenet, vision transformer, and FaceNet keras models pre-trained on the imagenet dataset, and, finally, the Pareto-optimized FaceNet model.

All of the above models have been implemented and tested in detail. After training the models over 30 epochs, the results in Table 4 show the performance of the models in

all the different performance metrics. After comparing the different models, the Inception ResnetV1 model and the Densenet model had a precision of 60% and 66% in the test set, respectively, which is quite low. This shows an underperforming model compared to the rest. The vision transformer that had the worst performance, highest model size and flops after 30 epochs required a lot of computation power to train the model and increasing the number of epochs might have resulted in a higher accuracy.

Table 4. Results using different models.

Model	Training Accuracy	Validation Accuracy	Testing Accuracy	Model Size	Flops
ArcFace ResNet50	92%	86%	77%	59.2 MB	8.93 G
Inception ResnetV1	90%	64%	60%	350 MB	3.25 G
Densenet	94%	69%	66%	95 MB	2.81 G
Vision Transformers	92%	92%	87%	300 MB	10.8 G
FaceNet	95%	86%	85%	257 MB	2.84 G
Pareto-optimized FaceNet	97%	93%	91%	187 MB	3.74 G

The ArcFace ResNet50 and FaceNet models were more accurate in the test dataset, although they had more computation time compared to the rest. However, the best performance was obtained by the Pareto-optimized FaceNet model.

4.2.1. GradCam HeatMap

One of the most popular methods for computer vision interpretability is Grad-CAM. A saliency map weight feature map, resembling a heat map, can be created by multiplying the target feature map obtained by forward propagation by the gradient of the fully connected layer obtained by backward propagation on the target feature map and then passing a ReLU activation function. This identifies the critical receptive field for task execution. Grad-CAM is a classic method of CNN interpretability. Compared to CAM (class activation map), it can generate a heatmap without changing the model, which is very convenient and flexible [93]. The implementation of this Grad-Cam heatmap to recognize our masked face dataset could help us increase the accuracy. The images in Figure 7 are examples of the test results.
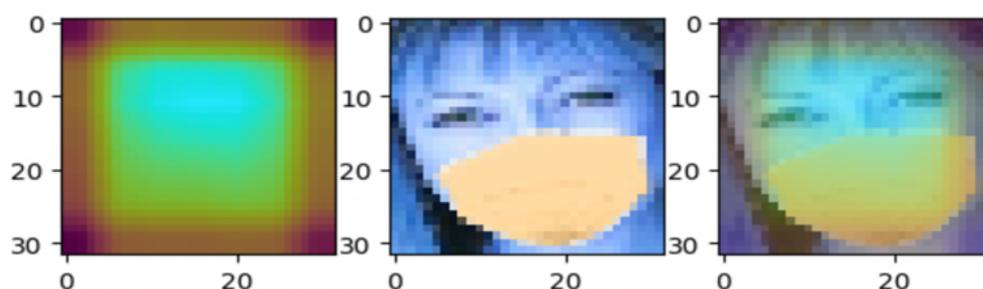

Figure 7. Grad-Cam example.

4.2.2. Accuracy and Loss Curves

As seen in Figure 8 below, after training the FaceNet model over 30 epochs, the validation accuracy of the model increased along with the training accuracy. This means that our model made better prediction increases as the epochs increased.

In addition, the training and validation loss value decreased, which means that the model was constantly learning. However, since the loss decreased in both the training set and the validation set, but there was a noticeable difference, the model could be improved in this case, which is the reason why we used GAN.
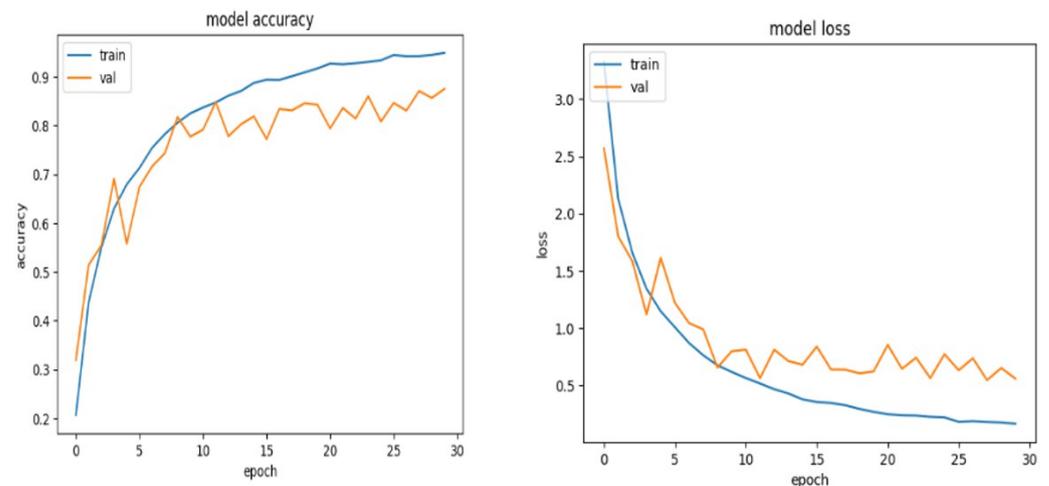
Figure 8. Accuracy and loss curves.

4.2.3. Classification Report

Class-by-class, the report in Table 5 below displays the precision, recall, and F1 score of the primary classification metrics. True positives, false negatives, and everything in between were used to determine these measures. The predicted classes are simply referred to here as "positive" or "negative".

Table 5. Classification results.

Class	Precision	Recall	F1-Score	Support
0	0.75	1	0.8571	6
1	1	0.8695	0.9302	23
2	0.9487	1	0.9736	74
3	1	0.7692	0.8695	13
4	1	1	1	4
5	0.6667	0.5714	0.6153	7
…	…	…	…	…
95	0.9230	0.9230	0.9230	26
96	0.9682	0.9682	0.9682	4 63
97	0.78	1	0.8764	39
98	0.8409	1	0.9135	37
99	0.9565	0.9778	0.9670	45
accuracy	0.9396	0.9396	0.9396	0.8796
macro avg	0.9084	0.9394	0.8986	3381
weighted avg	0.9070	0.9396	0.9387	3381

4.2.4. Model Robustness

The robustness of the model is very important for a model. A good model should not cause huge deviations in results due to changes in values or data. Basically, programmers check the performance of the model by using new versus training data to determine the robustness of the model [94].

The simplest way to check whether our model is robust is to add noise to the test data. In this case, we added noise to our dataset to test our model.

A method to determine whether models are change-resistant is to introduce noise into the test data. When we vary the amplitude of the noise, we can infer the model's performance with new data and other noise sources.

In our test set, we have introduced Gaussian noise from the Albumentations library discussed in Section 3.1 above. Means ranging from 10 to 50 were used, in order to see the performance of the mean changes.

Table 6 shows the performance of the model; as the mean of the Gaussian noise increased from 10 to 50, the model performance on the test set reduced from 89% to 82%.

Table 6. Gaussian noise (GN) results.

Model	Training Accuracy	Validation Accuracy	Test Accuracy
GN Mean 10	97%	92%	89%
GN Mean 30	96%	88%	85%
GN Mean 50	97%	84%	82%

After adding different intensities of noise, through step-by-step testing, we obtained the results in Table 6. The data in the results can be stabilized to more than 80%, indicating that the stability of our model is quite good.

4.2.5. SR-GAN Result

Super-Resolution Generative Adversarial Networks (SRGANs) are a class of deep learning models that are used to increase the resolution of images. The goal of an SRGAN implementation is to generate high-resolution images from low-resolution inputs while maintaining the visual quality and realism of the output. In this paper, we present an SRGAN implementation that achieved an accuracy of 94% in generating high-resolution images.

The SRGAN model architecture consists of two main components: a generator and a discriminator. The generator is responsible for generating high-resolution images, while the discriminator is used to evaluate the realism of the generated images. Both the generator and the discriminator are deep neural networks that are trained using a variant of the Generative Adversarial Networks (GAN) training algorithm.

The generator network is based on a U-Net architecture, which is a type of convolutional neural network that uses skip connections to propagate information from the contracting path to the expanding path. This allows for the preservation of fine details in the generated images. Additionally, the generator uses a Residual-in-Residual Dense Block (RRDB) architecture to increase the capacity of the network, which improves the quality of the generated images.

The discriminator network is a PatchGAN, which classifies whether each NxN patch in an image is real or fake. This allows for the evaluation of the entire image, rather than just a single output. The discriminator uses a multi-scale discriminator architecture, which evaluates the image at multiple scales to improve the realism of the generated images.

During training, the generator and discriminator networks are optimized in an adversarial manner. The generator aims to generate high-resolution images that are indistinguishable from real images, while the discriminator aims to correctly classify the generated images as fake. The two networks are trained together, with the generator being updated to improve the realism of its output and the discriminator being updated to better classify the generated images.

The SRGAN implementation was trained on a dataset of low-resolution images and their corresponding high-resolution versions. The model was trained for 200 epochs, with a batch size of 16. The Adam optimizer was used for optimization, with a learning rate of 0.0001 and a beta value of 0.9.

The results of the SRGAN implementation show that it is capable of generating high-resolution images with a high degree of visual realism. The generated images have a resolution that is four times higher than the input images, and the accuracy of the model in generating high-resolution images was 95%. These results demonstrate the effectiveness of SRGANs in image super-resolution tasks and the potential for their use in various applications such as medical imaging, surveillance, and video compression.

In conclusion, in our model, the Super Resolution GAN was used to denoise and also increase the resolution of images in the training dataset, as seen in Figure 9 below. After the implementation of the SR-GAN, the performance of the model was improved to 94% in the

test set. Table 7 shows the result of the model after training the model with the SR-GAN generated images.

Figure 9. SR-GAN.

Table 7. Performance of the model.

Model	Training Accuracy	Validation Accuracy	Test Accuracy
Pareto-optimized FaceNet with SR-GAN generated images	96%	95%	94%

4.3. Ablation Study

FaceNet's InceptionResnetV2 backbone is used for its quick learning time. This course serves as an ablation test bed for us to determine which components are most effective.

The effects of various convolutional blocks were investigated in this study by omitting or altering individual blocks progressively, and the results are summarized in Table 8. We can see from the table that removing the bottleneck block resulted in a 6.4% performance drop, that Mixed 7a (Reduction-B block) decreased accuracy by 15% to 72%, and that all other techniques improved performance by 0.2% to 0.3%, with the exception of removing the dropout and bottleneck simultaneously, resulting in a 1% performance drop. This resulted in a 67.8% accuracy rate and highlights the vital role these components play in the overall structure of the model. The results of the evaluations are consistent with each other and do not indicate any significant progress.

Table 8. Ablation study of the InceptionResnetV2 backbone of the FaceNet Model architecture. Time is the typical length of the model inference process.

Technique	Test Accuracy	Evaluation Time
FaceNet model	85.67%	30.5 s
Bottleneck	79.3%	28.8 s
Block 8 (Inception Resnet C Block)	76.21%	29.3 s
Block 17 (Inception Resnet C Block)	76.03%	30.9 s
Mixed 7a (Reduction-B block)	72.57%	31.2 s
Dropout and Bottleneck	67.80%	28.8 s

We also investigated our approach on unmasked faces (see Table 9. The baseline FaceNet model achieved a test accuracy of 92.25% with an evaluation time of 36.2 s, showcasing its strong performance. Various ablations were conducted to examine the effects of specific components on model performance. Notably, removing the bottleneck and certain Inception Resnet C Blocks (Block 8, Block 17, and Block 35) led to decreased test accuracies ranging from 46.61% to 83.26% and slightly affected the evaluation time. The inclusion of Dropout in conjunction with the bottleneck resulted in a lower test accuracy of 54.50%, indicating that this combination did not contribute positively to smoke detection. Furthermore, the Mixed 7a (Reduction-B block) ablation achieved a test accuracy of 74.25%, showing a moderate impact on model performance.

Table 9. Performance analysis of unmasked faces.

Technique	Test Accuracy	Evaluation Time
Facenet model	92.25%	36.2 s
- Bottleneck	83.26%	31.4 s
- Block 8 (Inception Resnet C Block)	81.14%	33.2 s
- Block 17 (Inception Resnet C Block)	75.73%	35.7 s
- Mixed 7a (Reduction-B block)	74.25%	34.6 s
- Dropout and Bottleneck	54.50%	33.5 s
- Block 35 (Inception Resnet C Block)	46.61%	35.1 s

5. Discussion

5.1. Answers to Research Questions

5.1.1. Research Question 1: How Does Preprocessing Data Affect Facial Recognition Systems, Specifically in the Context of Individuals Wearing Masks?

Pre-processing data plays a crucial role in the performance of facial recognition systems, particularly when dealing with individuals wearing masks. Preprocessing techniques such as CutMix and MixUp can be used to augment the training data, resulting in improved accuracy. These techniques involve applying random cropping and mixing of images to the training dataset, which helps the model generalize better and reduces overfitting. Additionally, increasing image resolution and de-noising can be used to enhance the quality of input images, which can also improve the performance of the recognition system.

5.1.2. Research Question 2: How Can We Improve the Performance of Face Recognition Systems When Individuals Wear Masks?

To improve the performance of face recognition systems when people wear masks, we need to identify an appropriate model and approach. One approach is to modify the structure and parameters of existing models, such as Google's FaceNet, which has a high rate of accuracy in face recognition. Additionally, we can adapt the dataset and image pre-processing techniques to better handle masked faces. For example, we can use a masked face dataset and train the model with it. Furthermore, we can make better use of data gathered from non-occlusion regions, such as the eyes and mouth, which are less likely to be obscured by a mask. Additionally, we can explore different types of masks and the level of occlusion they provide to improve the performance of the recognition system.

5.1.3. Research Question 3: How Does the Use of Both Masked Training and Test Images Affect Recognition Performance?

The use of both masked training and test images can significantly affect recognition performance. When the faces of the subjects studied are obscured or masked, there are less data from which to learn, making accurate subject recognition more challenging. Furthermore, when both the training and test images are masked, recognition performance decreases. This is because the model has not seen enough unmasked faces during training and, therefore, struggles to recognize masked faces during testing. To overcome this, we can use a combination of masked and unmasked faces in the training dataset and also use data augmentation techniques to increase the diversity of the data.

5.1.4. Research Question 4: How Can We Improve the Performance of Face Recognition Systems When Faces Are Obscured or Masked?

One potential approach to improving the performance of face recognition systems when faces are obscured or masked is to focus on preprocessing the data. This could involve techniques such as data augmentation, denoising, and resolution increase, which can help to improve the quality and diversity of the data used to train the model. Additionally, other techniques, such as adding CutMix and mixing augmentation, can also be used to improve the performance of the model. Another approach could be to change the parameters and

structure of the model, such as adjusting the number of layers or the size of the filters, to improve its performance on masked data.

5.1.5. Research Question 5: How Can We Make Better Use of the Data Gathered from Non-Occlusion Regions?

One potential approach to making better use of the data gathered from non-occlusion regions is to focus on feature extraction and selection techniques. This could involve techniques such as principal component analysis (PCA) or linear discriminant analysis (LDA) to identify the most informative features in the data. Additionally, techniques such as deep learning can also be used to extract features from non-occlusion regions of the face and use them to improve the performance of the model. Another approach could be to use data from both occluded and non-occluded regions to train the model, which can help to improve its generalization performance.

5.1.6. Research Question 6: How Can We Evaluate the Performance of Face Recognition Systems When Faces Are Obscured or Masked?

One potential approach to evaluating the performance of face recognition systems when faces are obscured or masked is to use a dataset of masked faces. This dataset could be used to train and test the model, and its performance could be evaluated using metrics such as accuracy, precision, recall, and F1-score. Additionally, it can be evaluated using ROC curve analysis. Another approach could be to use a dataset of both occluded and non-occluded faces, which can help to provide a more comprehensive evaluation of the model's performance. Additionally, real-world evaluation can be carried out by using the models on a real-time scenario, so as to check its accuracy and efficiency in identifying people with masks.

5.1.7. Research Question 7: How Can We Improve Recognition Performance When Both the Training and Test Images Are Masked?

One way to improve recognition performance when both the training and test images are masked is to use data augmentation techniques, such as CutMix and MixUp, on the training dataset. This can help increase the diversity of the data and make the model more robust to variations in mask types and levels of occlusion. Additionally, using a deep CNN, such as FaceNet, to convert the input face image into a vector and calculate the Euclidean distance between the two vectors with the vectors of each face in the dataset can help improve the accuracy of the model. Finally, changing the model's parameters and structure to better suit the masked face dataset can also improve recognition performance.

5.1.8. Research Question 8: How Can We Make Better Use of the Data Gathered from Non-Occlusion Regions?

One way to make better use of the data gathered from non-occlusion regions is to use a deep CNN, such as FaceNet, to convert the input face image into a vector and calculate the Euclidean distance between the two vectors with the vectors of each face in the dataset. This can help the model focus on the non-occluded regions of the face, such as the eyes and mouth, and improve recognition performance. Additionally, data preprocessing techniques such as resolution increase and denoising can help make the non-occluded regions more distinguishable. Finally, using a large, diverse dataset that includes a variety of mask types and levels of occlusion can also help the model better utilize the non-occluded regions.

5.1.9. Research Question 9: How Can We Improve the Performance of Facial Recognition Systems When Faces Are Obscured by Masks?

One approach to improve the performance of facial recognition systems when faces are obscured by masks is to use data preprocessing techniques such as 'CutMix' and 'MixUp', which have been found to improve the accuracy of the FaceNet model. Another approach is to make changes to the model's parameters and structure, such as increasing the resolution and denoising the input image, in order to better capture the features of the

face that are visible despite the mask. Additionally, it may be beneficial to use a different dataset that includes more examples of masked faces to train the model to better recognize obscured faces.

5.1.10. Research Question 10: What Are Some Potential Uses of Improved Facial Recognition Systems for Masked Faces?

Improved facial recognition systems for masked faces have the potential to be used in a variety of contexts, including law enforcement, security, and surveillance. For example, it could aid law enforcement in identifying criminals hiding behind masks and to detect individuals who are trying to evade detection by wearing masks. It could also be used in security settings, such as airports and train stations, to identify people trying to enter a restricted area while wearing a mask. Additionally, it could be used in surveillance settings, such as public areas and retail stores, to monitor individuals who wear masks in order to comply with local regulations.

5.2. Limitations

One limitation of this study is that it focussed solely on the use of FaceNet as a facial recognition model. While FaceNet has been shown to have high accuracy rates, there may be other models that are better suited for recognizing people while wearing masks. Additionally, the study was limited to the use of the CASIA and VGG-Face datasets, which may not fully represent the diversity of faces and masks in the real world. Another limitation is that the study focussed on the effect of data preprocessing techniques on the accuracy of the facial recognition model. However, other factors such as lighting and camera angle may also play a role in the accuracy of facial recognition when masks are worn, and these factors were not considered in the study. The study also did not consider the potential ethical and privacy implications of using facial recognition technology to identify people while wearing masks. This may raise concerns about surveillance and the collection of personal data. The study considered only the faces of healthy people; however, it is known that some diseases such as facial palsy can significantly distort the characteristics of the face [95] while distorting its symmetry features [96], which can negatively affect biometric recognition using face. Additionally, the study did not consider adversarial attacks or attempts at face forgery with the aim of concealing identity or performing impersonation, which can decrease the performance of face recognition [97], Finally, the study did not address the fact that the use of masks to conceal faces may be done for legitimate reasons and not only by criminals. Therefore, the study did not consider the impact of increasing facial recognition accuracy in individuals who wear masks for safety or medical reasons.

6. Conclusions

This research paper presents a novel hybrid model for the purpose of recognizing masked faces, which combines deep learning techniques with traditional machine learning methods. Our own Pareto-optimized FaceNet model was proposed as the main model for this task. This model is widely used in deep learning for facial recognition and has proven to be effective. The study utilized a mixture of two datasets consisting of 100 labels and various training and testing procedures. The dataset was divided into training, validation, and test sets using stratified K-Fold cross-validation, and the proposed model was trained and tested on these sets. The results of this study show that the Pareto optimization allowed improving the overall accuracy, over the 94% achieved by the original FaceNet variant, which also performed similarly to the arcface model during testing. Furthermore, the Pareto-optimized model no longer has a limitation of the model size and is a much smaller and more efficient version than the original FaceNet and its derivatives, helping to reduce its inference time and making it more practical for use in real-life applications.

Future work on FaceNet and facial recognition technology, in general, will focus on several key areas to address current limitations, improve performance, and explore new

applications. We will continue exploring alternative network architectures that provide better trade-offs between accuracy, computational complexity, and memory requirements. This may involve designing novel layers, activation functions, or loss functions, as well as utilizing techniques such as network pruning, quantization, or knowledge distillation.

References

1. Sharma, R.; Ross, A. Periocular biometrics and its relevance to partially masked faces: A survey. *Comput. Vis. Image Underst.* **2023**, *226*, 103583. [CrossRef]
2. Liu, Q.; Albina, E.M. Application of Face Recognition Technology in Mobile Payment. In Proceedings of the 2022 IEEE 12th International Conference on RFID Technology and Applications, RFID-TA 2022, Cagliari, Italy, 12–14 September 2022; pp. 217–219.
3. Elharrouss, O.; Almaadeed, N.; Al-Maadeed, S. A review of video surveillance systems. *J. Vis. Commun. Image Represent.* **2021**, *77*, 103116. [CrossRef]
4. Guo, Y. Impact on Biometric Identification Systems of COVID-19. *Sci. Program.* **2021**, *2021*, 3225687. [CrossRef]
5. Yan, S. Algorithms are not bias-free: Four mini-cases. *Hum. Behav. Emerg. Technol.* **2021**, *3*, 1180–1184. [CrossRef]
6. Rehman, A.; Saba, T.; Khan, M.Z.; Damaševičius, R.; Bahaj, S.A. Internet-of-Things-Based Suspicious Activity Recognition Using Multimodalities of Computer Vision for Smart City Security. *Secur. Commun. Netw.* **2022**, *2022*, 8383461. [CrossRef]
7. Grother, P.; Ngan, M.; Hanaoka, K. *Face Recognition Vendor Test Part 3: Demographic Effects*; Technical Report; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2019. [CrossRef]
8. Yang, Y.; Li, Y.; Yang, J.; Wen, J. Dissimilarity-based active learning for embedded weed identification. *Turk. J. Agric. For.* **2022**, *46*, 390–401. [CrossRef]
9. Meena, M.K.; Meena, H.K. A Literature Survey of Face Recognition Under Different Occlusion Conditions. In Proceedings of the 2022 IEEE Region 10 Symposium (TENSYMP), Mumbai, India, 1–3 July 2022; pp. 1–6. [CrossRef]
10. Kumar, G.; Zaveri, M.A.; Bakshi, S.; Sa, P.K. Who is behind the Mask: Periocular Biometrics when Face Recognition Fails. In Proceedings of the 2022 Second International Conference on Power, Control and Computing Technologies (ICPC2T), Raipur, India, 1–3 March 2022; pp. 1–6. [CrossRef]
11. Alonso-Fernandez, F.; Hernandez-Diaz, K.; Ramis, S.; Perales, F.J.; Bigun, J. Facial masks and soft-biometrics: Leveraging face recognition CNNs for age and gender prediction on mobile ocular images. *IET Biom.* **2021**, *10*, 562–580. [CrossRef]
12. Cloudwalk Technology Co., Ltd. Face Scan Payment Terminal. Available online: https://www.cloudwalk.com/en/Product?status=1&id=4 (accessed on 13 January 2023).
13. Fang, M.; Damer, N.; Kirchbuchner, F.; Kuijper, A. Real masks and spoof faces: On the masked face presentation attack detection. *Pattern Recognit.* **2022**, *123*, 108398. [CrossRef]
14. Rankin, J.C. *That Angry Darkness: An Ex-Unitarian Meets Satan Face-to-Face*; Independently Published: Traverse City, MI, USA, 2019; p. 262.
15. Sghaier, S.M.; Elfaki, A.O. Efficient Techniques For Human Face Occlusions Detection and Extraction. In Proceedings of the 2021 International Conference of Women in Data Science at Taif University (WiDSTaif), Taif, Saudi Arabia, 30–31 March 2021; pp. 1–5. [CrossRef]
16. Akhtar, Z.; Rattani, A. A Face in any Form: New Challenges and Opportunities for Face Recognition Technology. *Computer* **2017**, *50*, 80–90. [CrossRef]
17. Damer, N.; Grebe, J.H.; Chen, C.; Boutros, F.; Kirchbuchner, F.; Kuijper, A. The Effect of Wearing a Mask on Face Recognition Performance: An Exploratory Study. In Proceedings of the 2020 International Conference of the Biometrics Special Interest Group (BIOSIG), Darmstadt, Germany, 16–18 September 2020; pp. 1–6.
18. World Economic Forum. A Policy Framework for Responsible Limits on Facial Recognition Use Case: Law Enforcement Investigations (Revised 2022). White Paper. Available online: https://www.weforum.org/whitepapers/a-policy-framework-for-responsible-limits-on-facial-recognition-use-case-law-enforcement-investigations/ (accessed on 13 January 2023).
19. Yang, J.; Lan, G.; Xiao, S.; Li, Y.; Wen, J.; Zhu, Y. Enriching Facial Anti-Spoofing Datasets via an Effective Face Swapping Framework. *Sensors* **2022**, *22*, 4697. [CrossRef]

20. Yang, J.; Zhu, Y.; Xiao, S.; Lan, G.; Li, Y. A controllable face forgery framework to enrich face-privacy-protection datasets. *Image Vis. Comput.* **2022**, *127*, 104566. [CrossRef]

21. Ali, W.; Tian, W.; Din, S.U.; Iradukunda, D.; Khan, A.A. Classical and modern face recognition approaches: A complete review. *Multimed. Tools Appl.* **2020**, *80*, 4825–4880. [CrossRef]

22. Golwalkar, R.; Mehendale, N. Masked-face recognition using deep metric learning and FaceMaskNet-21. *Appl. Intell.* **2022**, *52*, 13268–13279. [CrossRef] [PubMed]

23. Queiroz, L.; Lai, K.; Yanushkevich, S.; Shmerko, V. Biometrics in the Time of Pandemic: 40% Masked Face Recognition Degradation can be Reduced to 2%. *arXiv* **2022**, arXiv:2201.00461. [CrossRef]

24. Guo, G.; Zhang, N. A survey on deep learning based face recognition. *Comput. Vis. Image Underst.* **2019**, *189*, 102805. [CrossRef]

25. Aggarwal, R.; Bhardwaj, S.; Sharma, K. Face Recognition System Using Image Enhancement with PCA and LDA. In Proceedings of the 2022 6th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 29–31 March 2022; pp. 1322–1327. [CrossRef]

26. Jia, S.; Guo, G.; Xu, Z. A survey on 3D mask presentation attack detection and countermeasures. *Pattern Recognit.* **2020**, *98*, 107032. [CrossRef]

27. Shakeel, M.S.; Lam, K.M. Deep-feature encoding-based discriminative model for age-invariant face recognition. *Pattern Recognit.* **2019**, *93*, 442–457. [CrossRef]

28. Ma, J.; Yuan, Y. Dimension reduction of image deep feature using PCA. *J. Vis. Commun. Image Represent.* **2019**, *63*, 102578. [CrossRef]

29. Yu, C.; Pei, H. Face recognition framework based on effective computing and adversarial neural network and its implementation in machine vision for social robots. *Comput. Electr. Eng.* **2021**, *92*, 107128. [CrossRef]

30. Tavakolian, N.; Nazemi, A.; Azimifar, Z.; Murray, I. Face recognition under occlusion for user authentication and invigilation in remotely distributed online assessments. *Int. J. Intell. Def. Support Syst.* **2018**, *5*, 277. [CrossRef]

31. Soni, N.; Sharma, E.K.; Kapoor, A. Hybrid meta-heuristic algorithm based deep neural network for face recognition. *J. Comput. Sci.* **2021**, *51*, 101352. [CrossRef]

32. Deotale, D.G.; Verma, M.; Suresh, P.; Srivastava, D.; Kumar, M.; Jangir, S.K. Analysis of Human Activity Recognition Algorithms Using Trimmed Video Datasets. In *Machine Learning and Data Science: Fundamentals and Applications*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2022. [CrossRef]

33. Gao, P.; Lu, K.; Xue, J.; Shao, L.; Lyu, J. A Coarse-to-Fine Facial Landmark Detection Method Based on Self-attention Mechanism. *IEEE Trans. Multimed.* **2021**, *23*, 926–938. [CrossRef]

34. Thilagavathi, B.; Suthendran, K.; Srujanraju, K. Evaluating the AdaBoost Algorithm for Biometric-Based Face Recognition. In *Data Engineering and Communication Technology*; Springer: Singapore, 2021; pp. 669–678. [CrossRef]

35. Tang, C.; Chen, S.; Zhou, X.; Ruan, S.; Wen, H. Small-Scale Face Detection Based on Improved R-FCN. *Appl. Sci.* **2020**, *10*, 4177. [CrossRef]

36. Moghadam, S.M.; Seyyedsalehi, S.A. Nonlinear analysis and synthesis of video images using deep dynamic bottleneck neural networks for face recognition. *Neural Netw.* **2018**, *105*, 304–315. [CrossRef] [PubMed]

37. Zhang, J.; Zi, L.; Hou, Y.; Wang, M.; Jiang, W.; Deng, D. A Deep Learning-Based Approach to Enable Action Recognition for Construction Equipment. *Adv. Civ. Eng.* **2020**, *2020*, 8812928. [CrossRef]

38. Mazloom, M.; Ayat, S. Combinational Method for Face Recognition: Wavelet, PCA and ANN. In Proceedings of the 2008 Digital Image Computing: Techniques and Applications, Canberra, Australia, 1–3 December 2008. [CrossRef]

39. Farfade, S.S.; Saberian, M.; Li, L.J. Multi-view Face Detection Using Deep Convolutional Neural Networks. *arXiv* **2015**, arXiv:1502.02766. [CrossRef]

40. Er, M.J.; Wu, S.; Lu, J.; Toh, H.L. Face recognition with radial basis function (RBF) neural networks. *IEEE Trans. Neural Netw.* **2002**, *13*, 697–710. [CrossRef]

41. Farfade, S.S.; Saberian, M.J.; Li, L.J. Multi-view Face Detection Using Deep Convolutional Neural Networks. In Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, New York, NY, USA, 23–26 June 2015. [CrossRef]

42. Zangeneh, E.; Rahmati, M.; Mohsenzadeh, Y. Low resolution face recognition using a two-branch deep convolutional neural network architecture. *Expert Syst. Appl.* **2020**, *139*, 112854. [CrossRef]

43. Agarwal, V.; Bhanot, S. Radial basis function neural network-based face recognition using firefly algorithm. *Neural Comput. Appl.* **2017**, *30*, 2643–2660. [CrossRef]

44. Cho, S.; Baek, N.; Kim, M.; Koo, J.; Kim, J.; Park, K. Face Detection in Nighttime Images Using Visible-Light Camera Sensors with Two-Step Faster Region-Based Convolutional Neural Network. *Sensors* **2018**, *18*, 2995. [CrossRef]

45. Shi, X.; Shan, S.; Kan, M.; Wu, S.; Chen, X. Real-Time Rotation-Invariant Face Detection with Progressive Calibration Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.

46. Li, C.; Huang, Y.; Huang, W.; Qin, F. Learning features from covariance matrix of gabor wavelet for face recognition under adverse conditions. *Pattern Recognit.* **2021**, *119*, 108085. [CrossRef]

47. Zarkasi, A.; Nurmaini, S.; Stiawan, D.; Suprapto, B.Y. Weightless Neural Networks Face Recognition Learning Process for Binary Facial Pattern. *Indones. J. Electr. Eng. Inform. (IJEEI)* **2022**, *10*, 955–969. [CrossRef]

48. Damer, N.; Boutros, F.; Süßmilch, M.; Fang, M.; Kirchbuchner, F.; Kuijper, A. Masked face recognition: Human versus machine. *IET Biom.* **2022**, *11*, 512–528. [CrossRef]

49. Neto, P.C.; Pinto, J.R.; Boutros, F.; Damer, N.; Sequeira, A.F.; Cardoso, J.S. Beyond Masks: On the Generalization of Masked Face Recognition Models to Occluded Face Recognition. *IEEE Access* **2022**, *10*, 86222–86233. [CrossRef]

50. Zhao, W.; Zhu, X.; Shi, H.; Zhang, X.Y.; Lei, Z. Consistent Sub-Decision Network for Low-Quality Masked Face Recognition. *IEEE Signal Process. Lett.* **2022**, *29*, 1147–1151. [CrossRef]

51. Martínez-Díaz, Y.; Méndez-Vázquez, H.; Luevano, L.S.; Nicolás-Díaz, M.; Chang, L.; González-Mendoza, M. Towards Accurate and Lightweight Masked Face Recognition: An Experimental Evaluation. *IEEE Access* **2022**, *10*, 7341–7353. [CrossRef]

52. Anwar, A.; Raychowdhury, A. Masked Face Recognition for Secure Authentication. *arXiv* **2020**, arXiv:2008.11104. [CrossRef]

53. Banati, U.; Prakash, V.; Verma, R.; Srivast, S. Soft Biometrics and Deep Learning: Detecting Facial Soft Biometrics Features Using Ocular and Forehead Region for Masked Face Images. *Res. Sq.* **2022**. [CrossRef]

54. Savchenko, A.V. Facial expression and attributes recognition based on multi-task learning of lightweight neural networks. In Proceedings of the 2021 IEEE 19th International Symposium on Intelligent Systems and Informatics (SISY), Subotica, Serbia, 16–18 September 2021; pp. 119–124. [CrossRef]

55. Neto, P.C.; Boutros, F.; Pinto, J.R.; Damer, N.; Sequeira, A.F.; Cardoso, J.S. FocusFace: Multi-task Contrastive Learning for Masked Face Recognition. In Proceedings of the 2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021), Jodhpur, India, 15–18 December 2021; pp. 01–08. [CrossRef]

56. Robinson, P.L. Automorphisms of Liouville Structures. *arXiv* **2015**, arXiv:1503.00383. [CrossRef]

57. Iranmanesh, S.M.; Riggan, B.; Hu, S.; Nasrabadi, N.M. Coupled generative adversarial network for heterogeneous face recognition. *Image Vis. Comput.* **2020**, *94*, 103861. [CrossRef]

58. Sharma, S.; Kumar, V. 3D landmark-based face restoration for recognition using variational autoencoder and triplet loss. *IET Biom.* **2020**, *10*, 87–98. [CrossRef]

59. Silabela, M.; Bogdandy, B.; Toth, Z. Automatic Mask Detecion using Convolutional Neural Networks and Variational Autoencoder. In Proceedings of the 2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI), Timisoara, Romania, 19–21 May 2021; pp. 461–466. [CrossRef]

60. Schroff, F.; Kalenichenko, D.; Philbin, J. FaceNet: A unified embedding for face recognition and clustering. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 815–823. [CrossRef]

61. Mishra, S.; Reza, H. A Face Recognition Method Using Deep Learning to Identify Mask and Unmask Objects. In Proceedings of the 2022 IEEE World AI IoT Congress (AIIoT), Seattle, WA, USA, 6–9 June 2022; pp. 91–99. [CrossRef]

62. Liu, J.; Zhao, S.; Xie, Y.; Gui, W.; Tang, Z.; Ma, T.; Niyoyita, J.P. Learning Local Gabor Pattern-Based Discriminative Dictionary of Froth Images for Flotation Process Working Condition Monitoring. *IEEE Trans. Ind. Inform.* **2021**, *17*, 4437–4448. [CrossRef]

63. Hao, S.; Chen, C.; Chen, Z.; Wong, K.Y.K. A Unified Framework for Masked and Mask-Free Face Recognition Via Feature Rectification. In Proceedings of the 2022 IEEE International Conference on Image Processing (ICIP), Bordeaux, France, 16–19 October 2022; pp. 726–730. [CrossRef]

64. Ledig, C.; Theis, L.; Huszar, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. *arXiv* **2016**, arXiv:1609.04802. [CrossRef]

65. Anwar, S.; Khan, S.; Barnes, N. A Deep Journey into Super-resolution: A survey. *arXiv* **2019**, arXiv:1904.07523. [CrossRef]

66. Wieczorek, M.; Silka, J.; Wozniak, M.; Garg, S.; Hassan, M.M. Lightweight Convolutional Neural Network Model for Human Face Detection in Risk Situations. *IEEE Trans. Ind. Inform.* **2022**, *18*, 4820–4829. [CrossRef]

67. Winnicka, A.; Kęsik, K.; Połap, D.; Woźniak, M. *SURF Algorithm with Convolutional Neural Network as Face Recognition Technique*; Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer International Publishing: Cham, Switzerland, 2020; Volume 12416 LNAI, pp. 95–102.

68. Boutros, F.; Damer, N.; Kirchbuchner, F.; Kuijper, A. Self-restrained triplet loss for accurate masked face recognition. *Pattern Recognit.* **2022**, *124*, 108473. [CrossRef]

69. Zhang, Y.; Wang, X.; Shakeel, M.S.; Wan, H.; Kang, W. Learning upper patch attention using dual-branch training strategy for masked face recognition. *Pattern Recognit.* **2022**, *126*, 108522. [CrossRef]

70. Huang, B.; Wang, Z.; Wang, G.; Jiang, K.; He, Z.; Zou, H.; Zou, Q. Masked Face Recognition Datasets and Validation. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, QC, Canada, 11–17 October 2021; pp. 1487–1491. [CrossRef]

71. Cao, Z.; Li, W.; Zhao, H.; Pang, L. YoloMask: An Enhanced YOLO Model for Detection of Face Mask Wearing Normality, Irregularity and Spoofing. In *Biometric Recognition, Proceedings of the 16th Chinese Conference, CCBR 2022, Beijing, China, 11–13 November 2022*; Deng, W., Feng, J., Huang, D., Kan, M., Sun, Z., Zheng, F., Wang, W., He, Z., Eds.; Springer Nature Switzerland: Cham, Switzerland, 2022; pp. 205–213.

72. Wang, K.; Wang, S.; Yang, J.; Wang, X.; Sun, B.; Li, H.; You, Y. Mask Aware Network for Masked Face Recognition in the Wild. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, QC, Canada, 11–17 October 2021; pp. 1456–1461. [CrossRef]

73. Montero, D.; Nieto, M.; Leskovsky, P.; Aginako, N. Boosting Masked Face Recognition with Multi-Task ArcFace. *arXiv* **2021**, arXiv:2104.09874. [CrossRef]

74. Hong, Q.; Wang, Z.; He, Z.; Wang, N.; Tian, X.; Lu, T. Masked Face Recognition with Identification Association. In Proceedings of the 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI), Baltimore, MD, USA, 9–11 November 2020; pp. 731–735. [CrossRef]

75. Song, L.; Gong, D.; Li, Z.; Liu, C.; Liu, W. Occlusion Robust Face Recognition Based on Mask Learning With Pairwise Differential Siamese Network. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 773–782. [CrossRef]

76. Wang, Q.; Guo, G. DSA-Face: Diverse and Sparse Attentions for Face Recognition Robust to Pose Variation and Occlusion. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 4534–4543. [CrossRef]

77. Biswas, R.; González-Castro, V.; Fidalgo, E.; Alegre, E. A new perceptual hashing method for verification and identity classification of occluded faces. *Image Vis. Comput.* **2021**, *113*, 104245. [CrossRef]

78. Wu, G. Masked Face Recognition Algorithm for a Contactless Distribution Cabinet. *Math. Probl. Eng.* **2021**, *2021*, 5591020. [CrossRef]

79. Li, Y.; Guo, K.; Lu, Y.; Liu, L. Cropping and attention based approach for masked face recognition. *Appl. Intell.* **2021**, *51*, 3012–3025. [CrossRef] [PubMed]

80. Ding, F.; Peng, P.; Huang, Y.; Geng, M.; Tian, Y. Masked Face Recognition with Latent Part Detection. In Proceedings of the 28th ACM International Conference on Multimedia, New York, NY, USA, 12–16 October 2020. [CrossRef]

81. Zhang, K.; Zhang, Z.; Li, Z.; Qiao, Y. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal Process. Lett.* **2016**, *23*, 1499–1503. [CrossRef]

82. Kocacinar, B.; Tas, B.; Akbulut, F.P.; Catal, C.; Mishra, D. A Real-Time CNN-Based Lightweight Mobile Masked Face Recognition System. *IEEE Access* **2022**, *10*, 63496–63507. [CrossRef]

83. Zulfiqar, M.; Syed, F.; Khan, M.J.; Khurshid, K. Deep Face Recognition for Biometric Authentication. In Proceedings of the 2019 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), Swat, Pakistan, 24–25 July 2019; pp. 1–6. [CrossRef]

84. Adjabi, I.; Ouahabi, A.; Benzaoui, A.; Taleb-Ahmed, A. Past, Present, and Future of Face Recognition: A Review. *Electronics* **2020**, *9*, 1188. [CrossRef]

85. Meng, Q.; Zhao, S.; Huang, Z.; Zhou, F. MagFace: A Universal Representation for Face Recognition and Quality Assessment. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 14225–14234.

86. Alfattama, S.; Kanungo, P.; Bisoy, S.K. Face Recognition from Partial Face Data. In Proceedings of the 2021 International Conference in Advances in Power, Signal, and Information Technology (APSIT), Bhubaneswar, India, 8–10 October 2021; pp. 1–5. [CrossRef]

87. Liu, A.; Tan, Z.; Wan, J.; Escalera, S.; Guo, G.; Li, S.Z. CASIA-SURF CeFA: A Benchmark for Multi-modal Cross-ethnicity Face Anti-spoofing. In Proceedings of the 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 3–8 January 2021; pp. 1178–1186. [CrossRef]

88. Mare, T.; Duta, G.; Georgescu, M.I.; Sandru, A.; Alexe, B.; Popescu, M.; Ionescu, R.T. A realistic approach to generate masked faces applied on two novel masked face recognition data sets. *arXiv* **2021**, arXiv:2109.01745. [CrossRef]

89. Abayomi-Alli, O.O.; Damaševičius, R.; Qazi, A.; Adedoyin-Olowe, M.; Misra, S. Data Augmentation and Deep Learning Methods in Sound Classification: A Systematic Review. *Electronics* **2022**, *11*, 3795. [CrossRef]

90. Buslaev, A.; Iglovikov, V.I.; Khvedchenya, E.; Parinov, A.; Druzhinin, M.; Kalinin, A.A. Albumentations: Fast and Flexible Image Augmentations. *Information* **2020**, *11*, 125. [CrossRef]

91. Fan, L.; Zhang, F.; Fan, H.; Zhang, C. Brief review of image denoising techniques. *Vis. Comput. Ind. Biomed. Art* **2019**, *2*, 7. [CrossRef]

92. Röhrbein, F.; Goddard, P.; Schneider, M.; James, G.; Guo, K. How does image noise affect actual and predicted human gaze allocation in assessing image quality? *Vis. Res.* **2015**, *112*, 11–25. [CrossRef]

93. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. *arXiv* **2016**, arXiv:1610.02391. [CrossRef]

94. Manfren, M.; James, P.A.; Tronchin, L. Data-driven building energy modelling – An analysis of the potential for generalization through interpretable machine learning. *Renew. Sustain. Energy Rev.* **2022**, *167*, 112686. [CrossRef]

95. Abayomi-alli, O.O.; Damaševicius, R.; Maskeliunas, R.; Misra, S. Few-shot learning with a novel voronoi tessellation-based image augmentation method for facial palsy detection. *Electronics* **2021**, *10*, 978. [CrossRef]

96. Wei, W.; Ho, E.S.L.; McCay, K.D.; Damaševičius, R.; Maskeliūnas, R.; Esposito, A. Assessing Facial Symmetry and Attractiveness using Augmented Reality. *Pattern Anal. Appl.* **2022**, *25*, 635–651. [CrossRef]

97. Arunkumar, P.M.; Sangeetha, Y.; Raja, P.V.; Sangeetha, S.N. Deep Learning for Forgery Face Detection Using Fuzzy Fisher Capsule Dual Graph. *Inf. Technol. Control* **2022**, *51*, 563–574. [CrossRef]