


Article

UFaceNet: Research on Multi-Task Face Recognition Algorithm Based on CNN

Huoyou Li ^{1,*} , Jianshiun Hu ¹, Jingwen Yu ², Ning Yu ² and Qingqiang Wu ^{2,*}

¹ School of Mathematics and Information Engineering, Longyan University, Longyan 364012, China; hjx@lyun.edu.cn

² Information School, Xiamen University, Xiamen 361005, China; 24320161152624@stu.xmu.edu.cn (J.Y.); 24320181153615@stu.xmu.edu.cn (N.Y.)

* Correspondence: lhy@lyun.edu.cn (H.L.); wuqq@xmu.edu.cn (Q.W.)

Abstract: With the application of deep convolutional neural networks, the performance of computer vision tasks has been improved to a new level. The construction of a deeper and more complex network allows the face recognition algorithm to obtain a higher accuracy. However, the disadvantages of large computation and storage costs of neural networks limit the further popularization of the algorithm. To solve this problem, we have studied the unified and efficient neural network face recognition algorithm under the condition of a single camera; we propose that the complete face recognition process consists of four tasks: face detection, in vivo detection, keypoint detection, and face verification; combining the key algorithms of these four tasks, we propose a unified network model based on a deep separable convolutional structure—UFaceNet. The model uses multisource data to carry out multitask joint training and uses the keypoint detection results to aid the learning of other tasks. It further introduces the attention mechanism through feature level clipping and alignment to ensure the accuracy of the model, using the shared convolutional layer network among tasks to reduce model calculations amount and realize network acceleration. The learning goal of multi-tasking implicitly increases the amount of training data and different data distribution, making it easier to learn the characteristics with generalization. The experimental results show that the UFaceNet model is better than other models in terms of calculation amount and number of parameters with higher efficiency, and some potential areas to be used.

Keywords: face recognition; UFaceNet; multi-task; CNN



Citation: Li, H.; Hu, J.; Yu, J.; Yu, N.; Wu, Q. UFaceNet: Research on Multi-Task Face Recognition Algorithm Based on CNN. *Algorithms* **2021**, *14*, 268. <https://doi.org/10.3390/a14090268>

Academic Editor: Tom Burr

Received: 23 August 2021

Accepted: 13 September 2021

Published: 15 September 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Face recognition is a kind of biometric identification technology based on the facial features of people. A series of technologies relate to face recognition by using cameras to collect images or video streams containing faces, and automatically detect and track faces in the images. Face verification is a subfield of face recognition, which refers to several images containing faces to judge whether these faces belong to the same identity. The method is to extract facial features from the target image to be verified, and then traverse the database of the known identity of facial features.

Due to the popularity of camera technology and the upsurge of machine learning, the basic research of face recognition has been relatively mature. According to a 2020 study by MarketsandMarkets, the global biometric systems market is expected to be worth USD 36.6 billion by 2021, and grow to USD 68.6 billion by 2025, with a CAGR of 13.4% during the forecast period (2021–2025).

The application of neural networks improves the performance of face recognition to a new level. The development trend is to build a deeper and more complex network to achieve higher accuracy, but the storage space and speed causes difficulties when attempting to meet the requirements of universal application. Face recognition includes

multiple subtasks, which is often inefficient when the neural network-based face recognition algorithm is run locally on common devices. There are disadvantages such as high delay and poor user experience, and to date no good solution has been found. To solve this problem, we researched the unified and efficient neural network algorithm for face recognition under the condition of a single camera, and propose a fast and efficient unified network: UFaceNet.

The paper [1] points out that there is a certain quantifiable correlation between computer vision tasks, and that the reasonable use of the relationship between individual tasks can improve the performance of said tasks. There are many starting points for introducing multitasking in the network. From a biological point of view, multitasking is similar to the simulation of human learning processes; humans use the knowledge of related tasks to learn new tasks. From an educational point of view, learning a simple task can help people master more complex tasks faster. As for the neural network model, multi-task learning avoids bias to meet the requirement of its hypotheses which provide sparse solutions, and can learn from the solutions that can explain more tasks at the same time. The model has better generalization [2].

UFaceNet uses Deep Detectable Convolutional Network as its model infrastructure to jointly learn, using this unified network, four subtasks: face detection, body detection, keypoint detection, and face verification. We hope to reduce the network time complexity while learning more advanced features with more generalizability. By sharing the shallow network between simple tasks and the deep network between complex tasks, the characteristics between tasks can be shared to reduce the amount of computing required by the network, and the model can be accelerated. At the same time, the attention mechanism is used to cut the feature level by using the basic face keypoints output in the network, so that the advanced tasks in the network (such as the accurate face keypoint detection and face verification) can focus on a specific space, making the model more robust and accurate.

Our main contributions are as follows:

- (1) We proposed a unified model network integrating face recognition related subtasks (UFaceNet), which can effectively use multi-task information for supervised learning of the network and improve the network generalization ability.
- (2) We have made use of the association between multiple subtasks in face recognition, designed inter-task dependencies to ensure network accuracy, used task fusion to improve network efficiency, and realized the acceleration of unified network learning and reasoning by sharing shallow convolution features among tasks.
- (3) We have designed and completed the common training process of the multisource multitask dataset with the unified model for the case where there is no single dataset covering all face recognition-related subtask tags.

2. Related Work

Face detection works to find the position of all faces in an image. Generally, the input is an image, and the output is the coordinates of any rectangular frames that detect the faces. Face detection is the basis of various face image analyses, and it is also the first step in the overall face verification algorithm. The main difficulties in face detection include: the diversity of facial gestures and angles, the influence of illumination intensity and angle, the possibility of partial occlusion of the face, the proximity of the face position affects the size of the face, and the face may appear in various positions of the image. At present, face detection algorithms can be divided into three categories: VJ framework-based, DPM model-based, and convolutional neural network-based.

In [3], the VJ framework was proposed by Viola and Jones in 2001. They used the integral graph to quickly calculate the Haar features of an image. The Haar features can reflect the contrast between parts of an image. The algorithm uses AdaBoost as a classifier and adopts a cascading structure, which greatly improves the detection efficiency. Under the hardware conditions at that time, the processing speed can reach 15 fps. However, the VJ framework also has some shortcomings. Haar features are too simple and have

insufficient stability. Moreover, the use of decision trees as weak classifiers can easily lead to overfitting. It also does not work well when the face is partially obscured or has an exaggerated expression.

In [4], Deformable Part Model (DPM) is a variable component model. At the time of detection, the model first calculates the DPM feature map of the input image, and then the input image is unsampled by Gaussian pyramid to obtain an image twice as large as the original, and the DPM feature map is calculated. The model then uses the root filter to obtain a response map for the DPM feature map. The model uses the part filter for the feature map of the upsampled image, and then performs Gaussian pyramid downsampling on the obtained response graph. Thus, the response maps obtained by the root filter and the part filter have the same resolution. Finally, the response graph is weighted and averaged to obtain the final response graph. The DPM-based method can achieve better face detection than the VJ-based method in complex scenes such as outdoors, but the DPM model is still hard to realize real-time detection due to the computational complexity of the model.

In [5–10], Cascade CNN is a convolutional neural network implementation of the VJ framework. Cascade CNN uses three CNN cascading structures. Cascade CNN first constructs a detection image pyramid, uses the primary network scan to remove most of the windows, and then adjusts the window position and size through a correction network, and uses the non-maximum suppression to merge the height coincidence window as the next level network input for further detection. Cascade CNN solves the problem of illumination and angle better than traditional methods. However, the performance of the method is affected to some extent by the fact that the first-level network still uses the dense sliding window. DenseBox uses a convolutional neural network to train images of different sizes, and finally directly predicts the position and confidence information of the face frame. DenseBox splices different convolutional layer outputs through upsampling and linear interpolation to achieve a multiscale fusion strategy and simultaneously locate keypoints, which improves the accuracy of detection. Faceness-Net inputs the images into five CNN networks, each of which outputs the position information of five different parts of the face, scores the information, and analyzes the scores of each part to obtain the face candidate frame. Face R-CNN is based on Faster R-CNN; it adds center loss to the last two classifications of the network to increase cohesion and adds the N largest samples of loss in each batch as a difficult case to the next training, which improves the classification ability of the entire network. This paper proposes an iterative algorithm for solving SDDLSp, which is suitable for training and testing images which are polluted by a large amount of noise. FHEDN is an end-to-end depth convolutional neural network, which uses a multiscale hierarchical feature pyramid fused with context prior-based information to detect faces in unconstrained scenes.

In face detection algorithms: (1) The VJ framework improves the detection performance by cascading, but the face detection effect for complex expressions and angles is not good. (2) The DPM model is less affected by noise, but is limited by the complex structure of the model. This leads to large calculations and poor real-time performance. (3) Neural network-based detection methods can obtain better facial features, but these algorithms generally have the disadvantages of low interpretation, difficult parameter adjustment, and long operation time.

Liveness detection refers to determining whether biological information comes from a legitimate user of living organism when obtaining biological information. The method of liveness detection mainly distinguishes biometrics forged by nonliving substances, such as photographs, silica gel, and plastic, by identifying physiological information on the organism as a feature. The paper [11] divides the liveness detection technology into texture information analysis, motion information analysis, and living part analysis. The performance gap of the classifier based on the nonrigid motion of the authentic image, the noise difference, and the face background dependence are discussed. According to the input, liveness detection can be divided into single-frame input and continuous multi-frame input.

The paper [12] proposes a liveness detection method that uses image distortion and color to construct feature vectors and uses SVM to perform two classifications. This method is not effective in the case where the forged facial image distortion is not serious. Color Texture [13] believes that the living and nonliving are indistinguishable in RGB space, but there are significant texture differences in other color spaces. A method for obtaining a facial feature, by converting a facial image from RGB space to YCbCr space, and then using the SVM classifier for two classifications, is proposed. The method is simple and efficient. The paper [14] implements liveness detection based on Lambertian Reflection and believes that a true facial living body and a nonliving body from a video or a photo are differently reflected under the same lighting conditions.

The paper [15] enhances facial micromotion by inputting continuous multi-frame facial images, and then extracting dynamic texture features and histograms of oriented optical flow. Santosh Tirunagari et al. [16] used the dynamic mode decomposition DMD to obtain the subspace map of the maximum motion energy, then perform texture analysis, and finally, input to the SVM classifier for binary classification. This method has a poor effect on forging the video or shaking a printed image. The paper [17] introduces the liveness detection dataset PHOTO-ATTACK, which is extended based on PRINT-ATTACK, adding high-resolution screen images and mobile phone images. Furthermore, an optical flow-based analysis method is proposed to distinguish the authenticity of the image, which achieves a better performance.

These traditional methods are mainly based on feature engineering. They rely on image quality evaluation, illumination, smoothness, and moiré, and then obtain the test results through two classifications. In addition to traditional detection methods, some literature has begun to apply deep learning to liveness detection in recent years but, limited by the small number of samples, the performance struggles to exceed traditional methods. CNN-LSTM [18] introduces deep learning to liveness detection earlier, and simulates traditional liveness detection through CNN, but the effect is not good. Yousef Atom et al. [19] designed a depth framework to replace the two-category problem with a targeted feature monitoring problem. Song Xiao et al. [20] used VGG16 as the basic network to directly add liveness detection to the face detection network. The detected bounding box includes three categories of confidence: background, living face, and nonliving face.

Liveness detection can be divided into traditional methods based on image quality, texture information, and deep learning-based methods. The former is not effective in some special cases, while the latter is limited by the small number of data samples, which makes the network training difficult to fit, and the performance struggles to surpass the traditional method.

Facial point detection refers to the image of a given face, finding the position and contour information of the key areas of the face. Facial point detection is roughly divided into three types: model-based, ASM and AAM methods; cascading shape regression; and deep learning-based methods.

Active Shape Model (ASM) [21] first aligns the training images so that the images are rotated, scaled, and translated as close as possible to a selected reference image, and local features are constructed for each keypoint. When searching for shapes, ASM first calculates the position of the eye, aligns the faces with a simple scale and rotation change, and then searches for the vicinity of the aligned points to match the local keypoint, obtains the preliminary shape, and then corrects it using the average face model. Active Appearance Models (AAM) [22] make improvements to ASM, which not only uses shape constraints but also adds texture features throughout the face area. Such linear models struggle to obtain better results under occlusion, special expressions, poses, and illumination changes, and their methods of searching for keypoints to exhaustive iterations similarly limit the computational efficiency of the method.

Cascaded waveform regression (CPR) [23] specifies the initial prediction values and gradually refines them through a series of linear models. Each regression relies on the output of the previous regression to perform simple image operations. The entire system

can automatically learn from the training samples. This algorithm is similar to random forest regression. It is a clear and simple regression algorithm. It can train a good model with a small amount of training data, but the model only detects three keypoints of the face. Dong Chen et al. [24] inherited the idea of CPR using simple features and cascading tree structure to complete the classification and regression. It can do face detection and facial point detection at the same time. The calculation speed is fast, and the memory is small, but the model parameters are too numerous, and difficult to adjust. Local Binary Feature (LBF) [25] is a tree-based method that learns the local binary features of each keypoint and then uses linear regression to detect keypoints by combining the features. The algorithm can be divided into three processes: feature extraction, LBF coding, and acquisition of shape increments. The model is fast and accurate, but the model only detects five keypoints on the face. Ensemble of Regression Trees (EERT) [26] uses the GBDT algorithm to build a cascaded residual tree, and then gradually returns to the key point in the iteration. The model occupies less memory, and the calculation is fast, but the model is larger.

Sun et al. [27] first applied CNN to facial point detection and proposed a cascaded CNN network DCNN (Deep Convolutional Network). This method belongs to the cascade regression method. It detects five facial keypoints by designing a three-layer convolutional neural network. It focuses on the depth of the first-level network. It believes that the deeper network structure can better extract the global features and improve the problem of local optimality caused by the initial inaccuracy, but the network does not detect well when the face is occluded. Erjin Zhou et al. [28] made improvements on DCNN, proposing a four-level cascade network from coarse to fine facial point detection. It uses the image of the face area predicted by CNN as the input of the network, which improves the positioning accuracy of the latter stages. Similar to DCNN, this method also has the problem of complex network structure. Kaipeng Zhang et al. [29] proposed the MTCNN (Multi-task Cascaded Convolutional Network), which can perform face detection and facial point detection simultaneously, making full use of the potential links between the two tasks. MTCNN has a certain improvement in speed and accuracy, but the network structure is complex and only detects five keypoints on the face. DAN, Deep Alignment Network [30], is also a method based on cascaded neural networks. It introduces a keypoint heat map, and each level of the network uses the entire image as input. Model positioning is accurate, but the calculation speed still needs to be improved.

The main three algorithms of facial point detection are: (1) Model-based ASM and AAM models are simple, the architecture is clear and easy to understand and apply, but its exhaustive iterative search limits the performance of the method. (2) The calculation speed based on the cascading shape regression model is fast, but such models have problems that the parameters are difficult to adjust. (3) The method based on deep learning has a strong feature extraction ability of the convolutional neural network, and the detection is more accurate, but the network is more complicated, and the cascade structure limits the performance of the model.

The purpose of face recognition is to extract feature information from the face image and identify the identity based on the feature. The general face recognition process is divided into two steps. The first step is facial feature extraction and feature selection, and the second step is object classification.

Popular traditional recognition algorithms include principal component analysis using feature faces, linear discriminant analysis, the Fisherface algorithm, the hidden Markov model, etc. The method of identifying feature faces is developed by Sirovich and Kirby and used by Matthew Turk and Alex Pentland for face classification [31]. The method takes the pixel points of the image as the original dimension unit and attempts to transform to another target space through one transformation, in which each face can be best distinguished. However, the performance of this method will decrease when the face is occluded. At the same time, the image analysis speed still needs to be optimized. The paper [32] first applies the hidden Markov model to the face recognition algorithm. The method trains the HMM on the spatial sequence of multiple sample images and obtains the

two-dimensional hidden Markov model based on the top-down and left-to-right structural features of the face according to the natural characteristics that the facial features are fixed. The algorithm of the Cove model uses DCT as the observation vector to obtain a good recognition effect, but its disadvantage is that the structure is complicated and the calculation amount is large.

In the past decade, Convolutional Neural Network (CNN) [33] has become one of the most popular techniques for solving computer vision problems. Many visual tasks, such as image classification, object detection, and face recognition, benefit from CNN's powerful learning and discriminative characterization. The CNN-based face recognition method usually regards CNN as a powerful feature extractor. DeepFace [34] uses CNN as a feature extractor for the face to train on 4 million facial images, and obtains 67 base points to transform the triangulated face into a 3D model to depth information, and then turns the face back. Finally, it uses the 4096-dimensional feature vector output to find the classification result. It achieves an accuracy of 97.35% on the LFW dataset. DeepID [35] uses CNN as a feature extractor to learn a 160-dimensional feature vector and finally uses various classifiers to obtain classification results. The main task of the DeepId network is to learn features, and its classification error rate is high. The input to FaceNet [36] is a triple image with two identical identity images and one different identity. The network directly learns the separability between features: the feature distance between different identities should be as large as possible and the feature distance between the same identities should be as small as possible. FaceNet does not consider the face alignment problem. It only relies on a large amount of training data and a special objective function to obtain an accuracy of 99.63% on the LFW. Xiang Wu et al. [37] proposed a Max-Feature-Map operation and used an MFM-based CNN model to learn facial information. MFM is an extension of the largest pooling. It suppresses the features with lower activation values by maximizing the characteristics of the same site on the adjacent two feature maps, and can effectively distinguish the noise data to make the model more robust. However, the samples used in the model training process are all aligned with facial patterns.

It can be seen that the traditional algorithms in face verification have a simple structure, but the ability to deal with occlusion and illumination changes is not good; the methods based on deep learning obtain good accuracy, but there is a widespread problem of a large amount of network computation.

3. Methods

A complete face recognition algorithm includes several related subtasks: face detection, liveness detection, facial point detection, and face verification. The design of the network structure, corresponding to each subtask and ensuring the accuracy of the algorithm, is the basis for building a unified multi-tasking network. We hope to use the intertask dependencies to achieve feature level-based clipping and alignment to increase model focus. At the same time, by sharing the shallow network, the amount of calculation required to complete all tasks is reduced, and the shared parameters reduce the model space. The multitask-based model structure enables the network-learning features to have better generalization capabilities and enhance the robustness of the model.

The network structure of UFaceNet is shown in Figure 1: Input the original image into the network, into the full connection layer network in Part ① and output the coordinates of five keypoints of the face in the image. Five keypoint coordinates are used for the shear feature mapping and the final output of the convolutional layer Conv_net_1 is the feature of the candidate region of the face, which is the input of the complete connection layer network in Part ②. The final output is whether the candidate regions contain faces. If the detection result is no face, the task is terminated; Continuing the living body detection task, the network clipped the output feature map of the second layer of the convolutional layer in Conv_net_1. The specific step is to divide an area closer to the surface according to the coordinates of the five keypoints. In this area, the rectangular area is randomly designated to cut the feature map to obtain the texture features of the corresponding facial skin area.

In vivo detection results were obtained through the full connection layer network in Part ③; when it was confirmed that the image contains the facial region and belongs to the legitimate living object, the original feature map F output by the convolutional layer in $Conv_net_1$ is cut out after the local facial features are cut, and then four feature maps, $O1\sim O4$, are stitched and input into the convolutional network at $Conv_net_2$ stage. The feature map output by the convolutional layer in $Conv_net_2$ stage includes five parts: F' and $O1'\sim O4'$. Here, $O1'\sim O4'$ are input into the corresponding fully connected layer in Part ⑤, and the final output is the position of each keypoint corresponding to the edges of each five senses, a total of 41 keypoints. According to the positions of the 41 keypoints, face clipping is performed on the feature map F' and rotated to face level according to the positions of both eyes to provide the feature aligned facial feature map AF . Finally, the feature map AF is input to the convolutional layer in the $Conv_net_3$ stage, and the final face verification task was achieved through the fully connected network in Part 77. In the test, the output of the second-lowest layer of the full connection layer will be used; a one-dimensional vector with a length of 256 is used as the input image to finally extract the expression of advanced facial features. By calculating and comparing the similarity between facial features, identity verification can be realized.

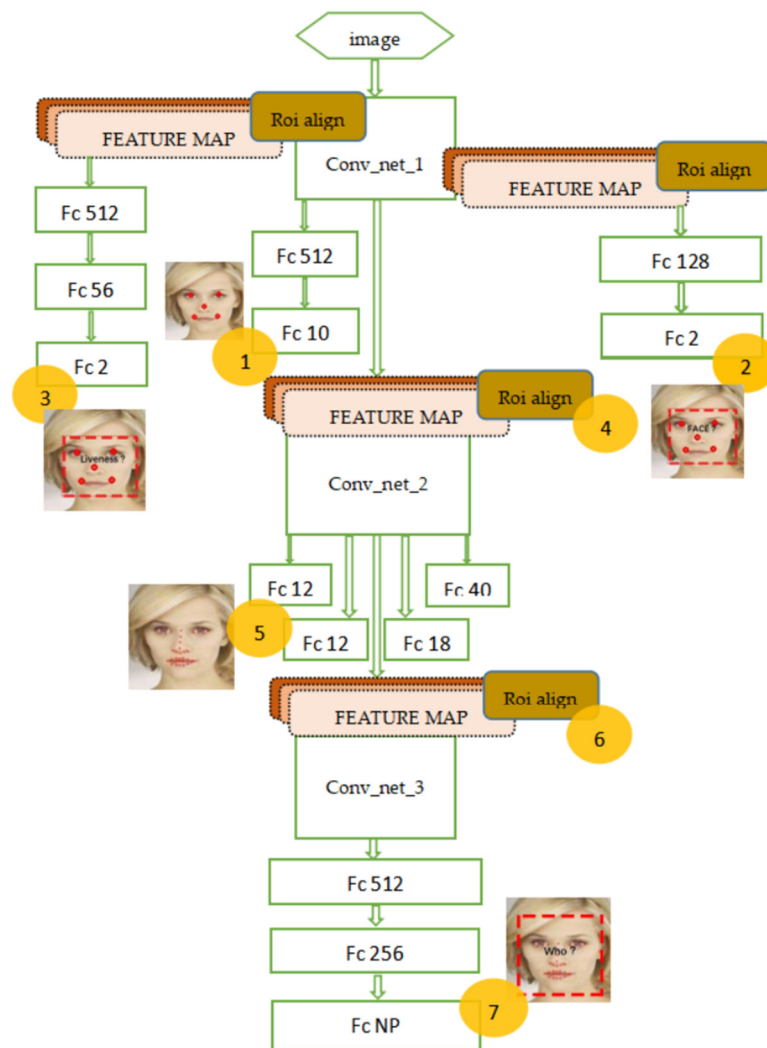


Figure 1. UFaceNet Network Structure.

3.1. Face Detection Based on Basic Facial Points

Considering the user scene of UFaceNet-face recognition, the user does not have a far distance from the imaging device, and the face area will occupy most of the image. Therefore, UFaceNet directly performs basic facial point detection on the entire picture to obtain a candidate face area, and then determines whether it is a face in the candidate area. The basic facial points are shown in Figure 2. It reduces the increase in structure and calculation caused by face detection, making the overall network more concise and focused. Combined with the final face recognition target, the model simplifies the facial point detection task in the algorithm. When there are multiple faces in the picture, the main user should be close to the center of the picture, so the model only performs facial point detection and subsequent detection, face verification, etc. on the face closest to the center of the picture. The objective functions for basic facial detection and face detection are as follows:

$$Basic_Loss = \alpha \frac{1}{N} \sum_{i=0}^N \sum_{l=0}^5 \sqrt{(x_{il} - X_{il})^2 + (y_{il} - Y_{il})^2} \quad (1)$$

$$Face_Detect_Loss = \gamma \frac{1}{N} \sum_{i=0}^N \sqrt{(c_i - C_i)^2} \quad (2)$$

where N is the number of samples of a batch in training, (x_{il}, y_{il}) is the predicted coordinates of the basic facial point l of the sample i , (X_{il}, Y_{il}) is the true coordinates of the basic facial point l of the sample i , α is the weight of *Bacic_Loss*, and γ is the weight of *Face_Detect_Loss*.

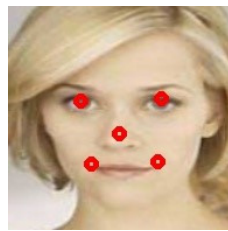


Figure 2. Basic facial points of the eyes, nose, and mouth.

3.2. Accurate Facial Point Detection

UFaceNet uses basic facial points to find accurate facial points. A total of 41 accurate facial points are shown in Figure 3. UFaceNet uses the ROI align [38] method to perform local area clipping on the feature map output by Conv_net_1. As shown in Figure 4, the clipping is based on five basic points, and the model selects the rectangular area around the eyes, the tip of the nose, and the mouth. The corresponding receptive field of the clipping local feature map contains the target area information that the model hopes to obtain. Additionally, the scope is more precise. Therefore, the corresponding network of the accurate facial point detection task pays more attention to a specific area and improves the accuracy of detection. The objective function of accurate facial point detection is as follows:

$$Accuarte_Loss = \beta \frac{1}{N} \sum_i^N \sum_l^{41} \sqrt{(m_{il} - M_{il})^2 + (n_{il} - N_{il})^2} \quad (3)$$

where N is the number of samples of a batch in training, (m_{il}, n_{il}) is the predicted coordinates of the facial point l of the sample i , (M_{il}, N_{il}) is the true coordinates of the facial point l of the sample i , and β is the weight of *Accurate_Loss*.

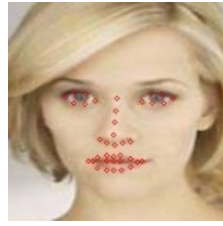


Figure 3. 41 points on the face.

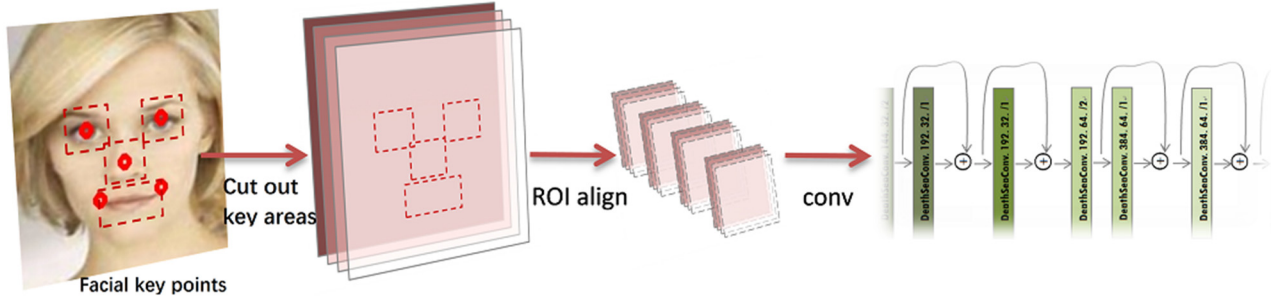


Figure 4. The process of clipping 41 facial points using ROI align.

3.3. Liveness Detection

Data play an important role in liveness detection. However, most of the existing liveness detection datasets were collected about 4 to 8 years ago. The resolution and image quality of current devices and devices used in previous data is very different. This will result in the model being less able to distinguish between existing fraudulent means. At the same time, the number of samples, the number of people, and the age group of the existing datasets are relatively limited. Detailed dataset information is shown in Table 1. These problems limit the performance of deep learning, and it is difficult to exceed the traditional methods in the case of too few samples based on deep learning.

Table 1. Descriptions of liveness detection datasets.

Dataset	Time	Number of People	Age	Image
MSU MFSD	2014	55	20–60	-
MSU USSAD	-	1000	-	9000
NAAA	2010	15	20–30	12,614
CASIA FASD	2012	50	20–35	-
IDIAP:THE Replay-Attack Database	2012	50	20–40	1300
IDIAP:3DMAD	-	17	-	76,500
IDIAP:Multispectral-Spoof Database	-	21	-	200
IDIAP:Replay-Mobile	-	40	-	1190

We have created a liveness detection dataset, the HWLD dataset that is more in line with deep network learning. First, the HWLD dataset uses all portrait images (unaligned) in the CelebA dataset as positive samples, including about 10,000 different identities, for a total of 202,599 images. These positive sample pictures are then made into video clips of approximately 10 hours (5 frames per second). We played videos on different devices, recorded them on the screen with other devices, and finally sourced the negative samples in HWLD by cutting frames of the recorded videos. In this way, a liveness detection dataset consisting of about 400,000 images and 10,000 different identities is obtained. Negative sample images are shown in Figure 5.



Figure 5. Negative sample images in the HWLD dataset.

The model imposes special restrictions on the image during training to improve accuracy. The network uses basic facial points to obtain a face area as a candidate area. This part of the candidate area contrasts the range of candidate areas in face detection to be more closely attached to the face, eliminating background interference outside the face as much as possible, and letting the network focus on learning facial texture features. Then, a rectangular frame is randomly obtained in the candidate region as an input feature map of the fully connected layer; that is, a random skin region in the face, as shown in Figure 6. This allows the model to focus more on texture features. When using the model for prediction, we randomly take multiple skin regions on the face and finally, use the voting mechanism to see the final classification results. The objective of liveness detection is as follows:

$$Spoof_Detection_Loss = \mu \frac{1}{N} \sum_i^N D_i \ln(\text{Softmax}(d_i)) \quad (4)$$

where N is the number of samples of a batch in training, d_i is the confidence of sample i predicted by the model, D_i is the real label of the sample i , and μ is the weight of $Spoof_Detection_Loss$.

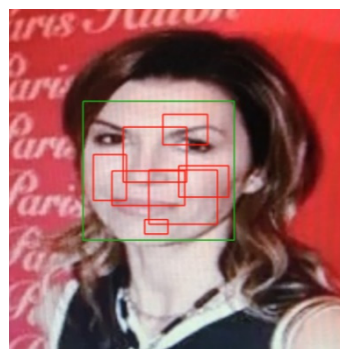


Figure 6. Random clipping of skin areas in liveness detection.

3.4. Face Verification

Face verification is based on the given two facial images, to determine whether the image belongs to the same person, which is the last task in UFaceNet. First, the output image of the converting layer is rotated. The rotation angle is calculated from the angle between the outer corners of the eye and the horizontal edge of the image in the key points. Then, a cut is made according to the center of the eyes and the center of the mouth. The clipping width is 1.66 times the distance between the eyes, and the cutting length is twice the eye-to-mouth spacing. Finally, as the size of the feature maps cropped by each sample is inconsistent, the model will implement the resize of the feature map by the ROI

align method. This results in a feature map of the key areas of the face that are spatially aligned. The process is shown in Figure 7. The model uses the central loss function and the cross-entropy loss function as the loss function for this part of the network. Their definitions are as follows:

$$Cross_Entropy_Loss = \varphi \frac{1}{N} \sum_i^N F_i \ln(Softmax(f_i)) \tag{5}$$

$$Center_Loss = \omega \frac{1}{N} \sum_i^N \|fv_i - c_i\|_2^2 \tag{6}$$

where N is the number of samples of a batch in training, f_i is the predicted probability vector of sample i , F_i is the true label of sample i , fv_i is the predicted facial feature vector, c_i is the mean of the corresponding category of sample i , φ is the weight of $Cross_Entropy_Loss$, and ω is the weight of $Center_Loss$.

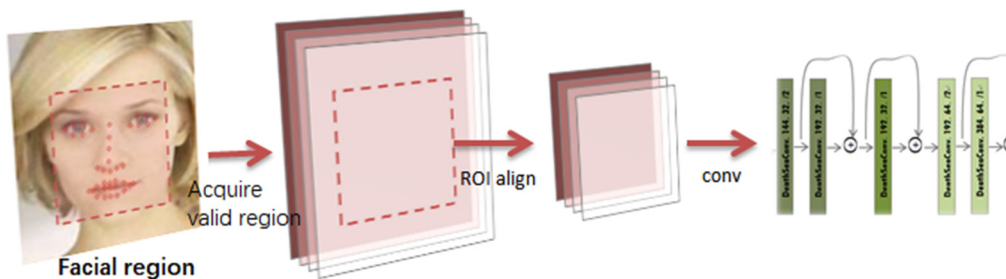


Figure 7. Face verification process for clipping face regions based on ROI align.

In the process of face verification, the model will discard the last layer of the fully connected layer for a specific dataset, and use the penultimate layer to obtain a one-dimensional feature vector fv of length 256 as the feature vector of the face image. We calculate the cosine similarity between fv_1 and fv_2 corresponding to the two images to determine whether it is the same identity. The formula for calculating the cosine similarity is:

$$cos_similarity = \frac{\sum_{i=1}^{256} fv_{1i}fv_{2i}}{\sqrt{\sum_{i=1}^{256} fv_{1i}^2} \sqrt{\sum_{i=1}^{256} fv_{2i}^2}} \tag{7}$$

When the similarity $cos_similarity$ is greater than the threshold, the two images are considered to belong to the same identity. The threshold is calculated from the validation set.

4. Experiment

4.1. Datasets

UFaceNet is a unified multi-tasking network. As there is no unified dataset with the tags needed in the network, multiple datasets are used in the training process and verification and testing process of the network. The tasks and corresponding datasets are shown in Table 2.

Table 2. Tasks and datasets.

Task	Dataset	Images
Basic Facial Point Detection	CelebFaces Attributes Dataset [39–41]	202,599
	WIDER Face [42]	32,203
	CelebFaces Attributes Dataset	202,599
	300 Faces In-the-Wild Challenge [43,44]	600
	LFPW	1132
	HELEN	348
Liveness Detection	AFW	205
	IBUG	135
	HWLD dataset	400,000
	MS-Celeb-1M [45]	5,000,000
	Labeled Faces in the Wild [46]	5000

4.2. Evaluation Criteria

We used the calculation and the number of parameters to measure the complexity of the model. The number of parameters determines the size of the model space, and the calculation measures the computational time of the model. We use accuracy (*ACC*), false acceptance rate (*FAR*), false rejection rate (*FRR*), *err_o*, and *err_c* as evaluation criteria. Their definitions are as follows:

$$ACC = \frac{T}{ALL} \quad (8)$$

$$FAR = \frac{NFA}{NIRA} \quad (9)$$

$$FRR = \frac{NFR}{NGRA} \quad (10)$$

$$err_o = \frac{1}{N} \sum_i \frac{\frac{1}{L} \sum_j |p_{ij} - y_{ij}|_2}{|l_{oij} - r_{oij}|_2} \quad (11)$$

$$err_c = \frac{1}{N} \sum_i \frac{\frac{1}{L} \sum_j |p_{ij} - y_{ij}|_2}{|l_{cij} - r_{cij}|_2} \quad (12)$$

where *T* is the correct number of tests, *ALL* is the total number of tests, *NFA* is the number of errors accepted, *NIRA* is the number of tests for different classes, *NFR* is the number of correct rejections, *NGRA* is the number of tests of the same class, *N* is the total number of samples tested, *L* is the number of points on the face, *p* is the coordinate of the predicted point, *y* is the true coordinate of this point, *lo* and *ro* correspond to the coordinates of the outer corners of the left and right eyes, respectively, and *lc* and *rc* correspond to the coordinates of the center position of the left and right eyes, respectively.

4.3. Results

(1) Face Verification: We calculated the facial features on the LFW dataset and found the similarity between the images. As shown in Figure 8, the left half of the area is the test group of the same identity, and the right half of the area is the test group of different identities. We compare the accuracy of the model when calculating the face center vector with different numbers of images. As shown in Table 3, the accuracy can be 99.9% when using three image calculation centers for verification, and the accuracy can reach 100% when using more than four image calculation centers. We compared the accuracy of different algorithms on the LFW dataset. The result is shown in Table 4. We compared the calculations of different models. As shown in Table 5, UFaceNet is far superior in the calculation to other models. It can be seen that UFaceNet guarantees the accuracy of face verification while achieving network acceleration.



Figure 8. UFaceNet faces the verification test results and similarity calculation with LFW.

Table 3. Comparison of the Accuracy of Different Image Numbers.

Number of Faces	Accuracy
1	0.987
2	0.992
3	0.999
4	1
5	1

Table 4. Comparison of Face Verification Accuracy Between Different Models with LFW.

Mode	Dimension	Accuracy	FAR = 0
DeepFace [34]	4096	0.973	0.463
DeepID [35]	160	0.99	0.693
WebFace [47]	320	0.977	-
FaceNet [36]	512	0.996	-
VGG	4096	0.97	0.61
LightCNN-4 [37]	256	0.979	0.79
LightCNN-9 [37]	256	0.988	0.95
LightCNN-29 [37]	256	0.993	0.975
UFaceNet	256	0.986	0.965

Table 5. Comparison of Calculation and Parameters of Different Models.

Model	Calculation	Parameter	Number of Network Layers
FaceNet	1600	140	11
LightCNN-29	2300	12.6	29
LightCNN-9	1900	5.5	9
LightCNN-4	1300	4.1	4
WebFace	774	5	10
UfaceNet	589	300	19
UfaceNet(face)	297	8.2	19

(2) Accurate Facial Point Detection: We compared different facial point detection algorithms on the 300-W test data. As shown in Tables 6 and 7, common corresponds to the test set portion of the LFPW and HELEN datasets, challenge corresponds to the IBUG dataset, and fully corresponds to all test sets. It can be seen that the facial point detection method of the UFaceNet model has achieved better results than other existing facial point

detection models. We also perform facial point detection tests on the LFW dataset. The result is shown in Figure 9.

Table 6. Comparison of err_o of Different Algorithms 500 Images Randomly Selected on 300-W Test Data.

Mode	Number of Key Points	Err_o
Intraface [46]	37	0.046
Face++	3	0.075
Lambda	3	0.097
Model [28]	51	0.043
UFaceNet	41	0.033

Table 7. Comparison of error_o of Different Algorithms on 300-W Test Data.

Mode	Common	Challenging	Full
MDM [48]	-	-	0.059
Kowalski et al. [49]	0.049	0.096	0.058
LBF [25]	0.072	0.176	0.093
Cgprt [50]	-	-	0.084
CFSS [51]	0.069	0.146	0.085
Kowalski et al. [52]	0.068	0.139	0.082
RAR [53]	0.060	0.122	0.072
DAN [30]	0.065	0.111	0.074
DAN-Menpo [30]	0.063	0.103	0.071
UFaceNet	0.051	0.107	0.062

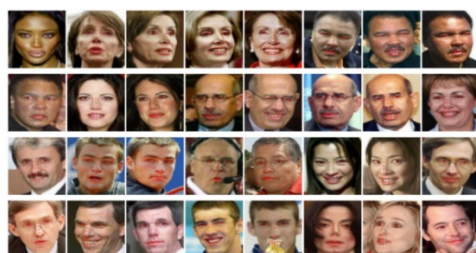


Figure 9. UFaceNet facial point detection results on LFW.

(3) Liveness Detection: We conducted training and testing of liveness detection on the HWND dataset constructed in this paper. We tested the effect of different numbers of random skin clipping regions on 3000 images on the HWLD dataset test set. As shown in Table 8, the accuracy can be improved as the number of cropping areas increases.

Table 8. Liveness Detection Results for the Number of Different Clipping Areas.

Number	FAR	FRR	HTER
1	3.3	6.7	5.00
3	2.8	5.4	4.10
7	1.2	5.5	3.35
15	0.9	4.6	2.75

(4) Face Detection: Intersection-over-Union (IoU) is an evaluation parameter in the target detection, which is the overlap ratio between the predicted bounding box and the real bounding box. When the IoU is greater than 0.5, it is considered that the target is detected. We tested the accuracy of the algorithm for different IoUs with LFPW and WIDEF Faces. The results are shown in Table 9. As our model learns the face closest to the center during training, the training data entered must contain the facial points. In a real application

scenario, if the input image does not contain a face, the model will also check out the facial point location somewhere on the image. Therefore, the face detection task of the model needs to classify the candidate regions after obtaining the candidate region to determine whether it is a human face. We perform experiments on classification accuracy on the CelebA data test set. We enter 250 original images of CelebA and cut out the background sections as negative samples. A total of 500 test images are input into the network for facial point detection and classification. The final detection accuracy rate was 97.8%, of which the negative samples were all classified correctly, and 4.4% of the positive samples are misclassified.

Table 9. Accuracy of Different IoU.

Dataset	IoU	Accuracy
LFPW	0.5	0.9917
LFPW	0.6	0.9629
LFPW	0.7	0.9094
WIDER Face	0.5	0.9705
WIDER Face	0.6	0.9314
WIDER Face	0.7	0.8922

4.4. Model Validation

We use different types of models to compare computing speeds between networks. The result is shown in Figure 10. As can be seen in Figure 10, the UFaceNet model is far superior to other models in terms of calculation amount and number of parameters. The UFaceNet model has a detection time of $0.51 \text{ ms} \pm$ (including all tasks) per image in an experimental environment.

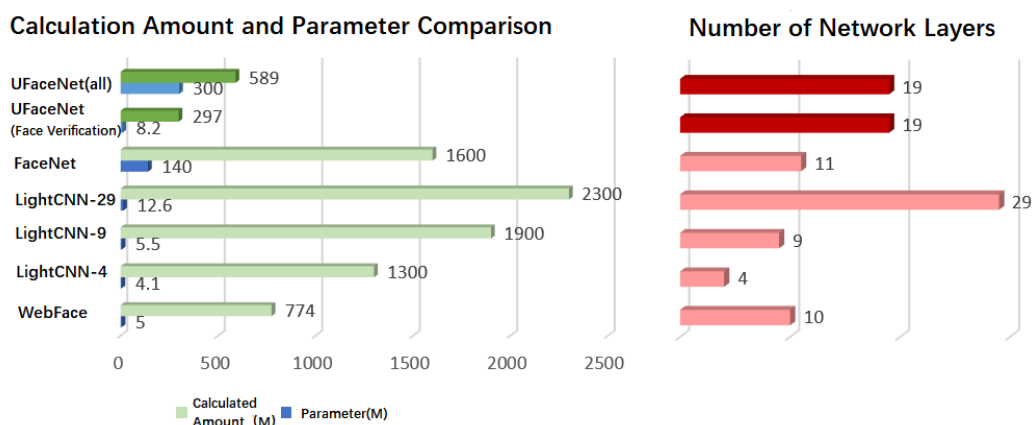


Figure 10. Comparison of calculations amount and parameter among different models.

5. Conclusions

This paper is based on the problem that the face recognition neural network model struggles to meet the requirements of mobile devices in terms of storage space and speed. We propose a complete and feasible real-time face recognition scheme for 2D images to promote the progress of the 2D face recognition scheme, reduce the hardware requirements and hardware cost of equipment, and improve the popularity of the face recognition scheme. We implicitly increase the amount of training data and the distribution of different data through multitask learning objectives. This makes it easier for the network to learn a generalized identification feature that is less susceptible to noise in the data and reduces the risk of the network overfitting a single task. The network structure, based on deep wise separable convolution and the high degree of multiplexing of features in the multi-tasking network, avoids the repeated calculation of the same feature by different tasks, reduces the

need for multi-tasking model calculation, reduces the overall network calculation cost, and achieves efficient network forward- and backpropagation. The experimental results show that the UFaceNet model is better than other models in terms of calculation amount and number of parameters, higher efficiency, and ease to be widely used.

UFaceNet has some problems that need to be improved: In the training process, the weight allocation of the multi-tasking loss function was set manually according to prior knowledge. In the future, we can try to build an intelligent weight allocation algorithm to dynamically allocate the weight according to the current multi-tasking learning situation. The self-built database HWLD for in vivo detection tasks has improved in terms of the number of people with different identities compared with other existing relevant databases; due to the limited time and equipment resources, it failed to collect more abundant deception scenes, so it can continue to expand and improve the data in the future. The face verification algorithm only carries out the matching verification of user identity, so the security aspect needs to be improved. In the follow-up work, we will try to add the fixation tracking model, based on the original network, to prevent users from using facial information without knowing it, to enhance the reliability and security of the network.

Author Contributions: Writing—original draft: H.L. Writing—review & editing: H.L., J.H., J.Y., N.Y. and Q.W. All authors have read and agreed to the published version of the manuscript.

Funding: The works are supported by the Longyan University's Qi Mai Science and Technology Innovation Fund Project of Longyan City (2017SHQM07).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: There is no conflict of interest regarding the publication of this paper.

References

1. Zamir, A.R.; Sax, A.; Shen, W.; Guibas, L.; Malik, J.; Savarese, S. Taskonomy: Disentangling Task Transfer Learning. In Proceedings of the IEEE/Cvf Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3712–3722.
2. Hsieh, Y.-H.; Leung, F.W. An overview of deep learning algorithms and water exchange in colonoscopy in improving adenoma detection. *Expert Rev. Gastroenterol. Hepatol.* **2019**, *13*, 1153–1160. [[CrossRef](#)]
3. Viola, P.A.; Jones, M.J.; Snow, D. Detecting Pedestrians Using Patterns of Motion and Appearance. In Proceedings of the IEEE International Conference on Computer Vision, Nice, France, 13–16 October 2003.
4. Felzenszwalb, P.F.; Mcallester, D.A.; Ramanan, D. A discriminatively trained, multiscale, deformable part model. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008.
5. Li, H.; Lin, Z.; Shen, X.; Brandt, J.; Hua, G. A convolutional neural network cascade for face detection. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
6. Huang, L.; Yi, Y.; Deng, Y.; Yu, Y. DenseBox: Unifying Landmark Localization with End to End Object Detection. *arXiv* **2015**, arXiv:1509.04874.
7. Yang, S.; Luo, P.; Loy, C.-C.; Tang, X. From Facial Parts Responses to Face Detection: A Deep Learning Approach. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015.
8. Jiang, H.; Learned-Miller, E. Face Detection with the Faster R-CNN. In Proceedings of the 2017 12th IEEE International Conference on Automatic Face and Gesture Recognition, Washington, DC, USA, 30 May–3 June 2017; pp. 650–657.
9. Chang, H.; Zhang, F.; Gao, G.; Zheng, H.; Chen, Y. Structure-constrained discriminative dictionary learning based on Schatten p-norm for face recognition. *Digit. Signal Process.* **2019**, *95*, 102573. [[CrossRef](#)]
10. Zhou, Z.; He, Z.; Jia, Y.; Du, J.; Wang, L.; Chen, Z. Context prior-based with residual learning for face detection: A deep convolutional encoder-decoder network. *Signal Process.-Image Commun.* **2020**, *88*, 115948. [[CrossRef](#)]
11. Kahm, O.; Damer, N. 2D face liveness detection: An overview. In Proceedings of the International Conference of Biometrics Special Interest Group (BIOSIG), Darmstadt, Germany, 6–7 September 2012.
12. Di, W.; Hu, H.; Jain, A.K. Face Spoof Detection with Image Distortion Analysis. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 746–761.
13. Boulkenafet, Z.; Komulainen, J.; Hadid, A. Face Spoofing Detection Using Color Texture Analysis. *IEEE Trans. Inf. Forensics Secur.* **2017**, *11*, 1818–1830. [[CrossRef](#)]

14. Tan, X.; Yi, L.; Liu, J.; Jiang, L. Face Liveness Detection from a Single Image with Sparse Low Rank Bilinear Discriminative Model. In *European Conference on Computer Vision, Proceedings of the 11th European Conference on Computer Vision, Heraklion, Crete, Greece, 5–11 September 2010*; Springer: Berlin/Heidelberg, Germany, 2010.
15. Bharadwaj, S.; Dhamecha, T.I.; Vatsa, M.; Singh, R. Face Anti-spoofing via Motion Magnification and Multifeature Videolet Aggregation. *IEEE Trans. Inf. Forensics Secur.* **2016**, *3*, 49–60.
16. Tirunagaris, S.; Poh, N.; Windridge, D.; Iorliam, A.; Suki, N.; Ho, A.T.S. Detection of Face Spoofing Using Visual Dynamics. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 762–777. [[CrossRef](#)]
17. Anjos, A.; Chakka, M.M.; Marcel, S. Motion-Based Counter-Measures to Photo Attacks in Face Recognition. *IET Biom.* **2014**, *3*, 147–158. [[CrossRef](#)]
18. Xu, Z.; Shan, L.; Deng, W. Learning temporal features using LSTM-CNN architecture for face anti-spoofing. In *Proceedings of the 3rd IAPR Asian Conference on Pattern Recognition (ACPR), Kuala Lumpur, Malaysia, 3–6 November 2015*.
19. Atoum, Y.; Liu, Y.; Jourabloo, A.; Liu, X. Face Anti-Spoofing Using Patch and Depth-Based CNNs. In *Proceedings of the IEEE International Joint Conference on Biometrics, Denver, CO, USA, 1–4 October 2017*.
20. Song, X.; Zhao, X.; Fang, L.; Lin, T. Discriminative Representation Combinations for Accurate Face Spoofing Detection. *Pattern Recognit.* **2018**, *85*, 182–191. [[CrossRef](#)]
21. Valstar, M.; Martinez, B.; Binefa, X.; Pantic, M. Facial point detection using boosted regression and graph models. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010*.
22. Cootes, T.F.; Taylor, C.J. Statistical Models of Appearance for computer vision. *Proc. SPIE—Int. Soc. Opt. Eng.* **2004**, *4322*, 236–248.
23. Dollar, P.; Welinder, P.; Perona, P. Cascaded pose regression. *IEEE* **2010**, *238*, 1078–1085.
24. Dong, C.; Ren, S.; Wei, Y.; Cao, X.; Sun, J. Joint Cascade Face Detection and Alignment. In *European Conference on Computer Vision, Proceedings of the 13th European Conference, Zurich, Switzerland, 6–12 September 2014*; Springer: Cham, Switzerland, 2014.
25. Ren, S.; Cao, X.; Wei, Y.; Sun, J. Face Alignment at 3000 FPS via Regressing Local Binary Features. In *Proceedings of the Computer Vision & Pattern Recognition, Columbus, OH, USA, 23–28 June 2014*.
26. Kazemi, V.; Sullivan, J. One Millisecond Face Alignment with an Ensemble of Regression Trees. In *Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition, Columbus, OH, USA, 23–28 June 2014*.
27. Yi, S.; Wang, X.; Tang, X. Deep Convolutional Network Cascade for Facial Point Detection. In *Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition, Portland, OR, USA, 23–28 June 2013*.
28. Zhou, E.; Fan, H.; Cao, Z.; Jiang, Y.; Yin, Q. Extensive Facial Landmark Localization with Coarse-to-Fine Convolutional Network Cascade. In *Proceedings of the IEEE International Conference on Computer Vision Workshops, Sydney, Australia, 2–8 December 2013*.
29. 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](#)]
30. Kowalski, M.; Naruniec, J.; Trzcinski, T. Deep Alignment Network: A Convolutional Neural Network for Robust Face Alignment. In *Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017*.
31. Turk, M.A.; Pentland, A.P. Face recognition using eigenfaces. In *Proceedings of the International Conference on Computer Research & Development, Maui, HI, USA, 3–6 June 2011*.
32. Samaria, F.; Young, S. HMM-based architecture for face identification. *Image Vis. Comput.* **1994**, *12*, 537–543. [[CrossRef](#)]
33. Lecun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Comput.* **2014**, *1*, 541–551. [[CrossRef](#)]
34. Taigman, Y.; Ming, Y.; Ranzato, M.A.; Wolf, L. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. In *Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition, Columbus, OH, USA, 23–28 June 2014*.
35. Yi, S.; Wang, X.; Tang, X. Deeply learned face representations are sparse, selective, and robust. In *Proceedings of the Computer Vision & Pattern Recognition, Boston, MA, USA, 7–12 June 2015*.
36. 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*.
37. Xiang, W.; Ran, H.; Sun, Z.; Tan, T. A Light CNN for Deep Face Representation with Noisy Labels. *IEEE Trans. Inf. Forensics Secur.* **2015**, *99*, 120–125.
38. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *99*, 20–24.
39. Liu, Z.; Ping, L.; Wang, X.; Tang, X. Deep Learning Face Attributes in the Wild. *arXiv* **2014**, arXiv:1411.7766.
40. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
41. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv* **2015**, arXiv:1502.03167.
42. Yang, S.; Ping, L.; Loy, C.C.; Tang, X. WIDER FACE: A Face Detection Benchmark. In *Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016*.
43. Sagonas, C.; Antonakos, E.; Tzimiropoulos, G.; Zafeiriou, S.; Pantic, M. 300 Faces In-The-Wild Challenge: Database and results. *Image Vis. Comput.* **2016**, *47*, 3–18. [[CrossRef](#)]

44. Sagonas, C.; Tzimiropoulos, G.; Zafeiriou, S.; Pantic, M. 300 Faces in-the-Wild Challenge: The First Facial Landmark Localization Challenge. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Sydney, Australia, 2–8 December 2013.
45. Guo, Y.; Lei, Z.; Hu, Y.; He, X.; Gao, J. MS-Celeb-1M: A Dataset and Benchmark for Large-Scale Face Recognition. *arXiv* **2016**, arXiv:1607.08221.
46. Belhumeur, P.N.; Jacobs, D.W.; Kriegman, D.J.; Kumar, N. Localizing parts of faces using a consensus of exemplars. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 2930–2940. [[CrossRef](#)] [[PubMed](#)]
47. Dong, Y.; Zhen, L.; Liao, S.; Li, S.Z. Learning Face Representation from Scratch. *arXiv* **2014**, arXiv:1411.7923.
48. Trigergeris, G.; Snape, P.; Nicolaou, M.A.; Antonakos, E.; Zafeiriou, S. Mnemonic Descent Method: A Recurrent Process Applied for End-to-End Face Alignment. In Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
49. Kowalski, M.; Naruniec, J. Face Alignment Using K-Cluster Regression Forests with Weighted Splitting. *IEEE Signal Process. Lett.* **2016**, *23*, 1567–1571. [[CrossRef](#)]
50. Lee, D.; Park, H.; Chang, D.Y. Face alignment using cascade Gaussian process regression trees. In Proceedings of the Computer Vision & Pattern Recognition, Boston, MA, USA, 7–12 June 2015.
51. Cheng, L. Face Alignment by Coarse-to-Fine Shape Searching. In Proceedings of the Computer Vision & Pattern Recognition, Boston, MA, USA, 7–12 June 2015.
52. Xiong, X.; Torre, F.D.L. Supervised Descent Method and Its Applications to Face Alignment. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013.
53. Xiao, S.; Feng, J.; Xing, J.; Lai, H.; Yan, S.; Kassim, A. Robust Facial Landmark Detection via Recurrent Attentive-Refinement Networks. In *Computer Vision—ECCV 2016; Lecture Notes in Computer Science*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer: Cham, Switzerland, 2016; pp. 57–72.