biomimetics

MDPI

Article

A 3D Face Recognition Algorithm Directly Applied to Point Clouds

Xingyi You [1,2] and Xiaohu Zhao [1,2,*]

[1] National and Local Joint Engineering Laboratory of Internet Applied Technology on Mines, China University of Mining and Technology, Xuzhou 221008, China; xingyiyou@cumt.edu.cn
[2] School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221008, China
* Correspondence: xiaohuzhao@cumt.edu.cn

Abstract: Face recognition technology, despite its widespread use in various applications, still faces challenges related to occlusions, pose variations, and expression changes. Three-dimensional face recognition with depth information, particularly using point cloud-based networks, has shown effectiveness in overcoming these challenges. However, due to the limited extent of extensive 3D facial data and the non-rigid nature of facial structures, extracting distinct facial representations directly from point clouds remains challenging. To address this, our research proposes two key approaches. Firstly, we introduce a learning framework guided by a small amount of real face data based on morphable models with Gaussian processes. This system uses a novel method for generating large-scale virtual face scans, addressing the scarcity of 3D data. Secondly, we present a dual-branch network that directly extracts non-rigid facial features from point clouds, using kernel point convolution (KPConv) as its foundation. A local neighborhood adaptive feature learning module is introduced and employs context sampling technology, hierarchically downsampling feature-sensitive points critical for deep transfer and aggregation of discriminative facial features, to enhance the extraction of discriminative facial features. Notably, our training strategy combines large-scale face scanning data with 967 real face data from the FRGC v2.0 subset, demonstrating the effectiveness of guiding with a small amount of real face data. Experiments on the FRGC v2.0 dataset and the Bosphorus dataset demonstrate the effectiveness and potential of our method.

Keywords: 3D face recognition; deep learning; point clouds

Academic Editor: Qian Jiang

Received: 31 December 2024
Revised: 21 January 2025
Accepted: 22 January 2025
Published: 23 January 2025

Citation: You, X.; Zhao, X. A 3D Face Recognition Algorithm Directly Applied to Point Clouds. *Biomimetics* **2025**, *10*, 70. https://doi.org/10.3390/biomimetics10020070

1. Introduction

Face recognition, including 2D and 3D recognition, has attracted a lot of interest recently because of its unique features. It is being used extensively in a variety of sectors, including affective computing, border control, criminal detection, mobile device user identification, and video surveillance [1–3]. Ongoing research is being carried out to develop new theories and methods with the aim of enhancing the accuracy of face recognition while ensuring its availability. Current techniques fall into two categories: deep learning-based and classic methods. Classic techniques frequently use feature extractors that were artificially created, like EigenFace [4], Fisher-Face [5], and LBP Face [6], which are well known among the well-established solutions. In contrast, deep learning techniques that naturally learn features during the training process without the need for an artificially constructed feature extractor are more popular.

Despite the success of 2D face recognition, it is still affected by occlusion, light changes, camera type, and resolution. Consequently, 3D face recognition has received more and more attention, though it is still in its infancy [3]. The problems of 3D face recognition based on deep learning are mainly reflected in the following two aspects:

(1) Deep learning methods mostly rely on data, necessitating large-scale training datasets for optimal outcomes. However, the largest available 3D face dataset currently contains only tens of thousands of training images, paling in comparison to the nearly one million images in 2D face recognition datasets like ArcFace [7], MS-Celeb-1M [8], and FaceNet [9].

(2) Developing effective network models is the foundation of deep learning methods. Existing models often operate on 2D images, neglecting the characteristics of 3D data. Point clouds, often representing 3D face data, exhibit rich geometric information and an unstructured data format. While some researchers have endeavored to employ existing deep learning networks directly on point cloud data [10–15], the efficacy of such models has primarily been demonstrated on rigid objects like chairs and tables. The uncertainties associated with face data, stemming from its unique expression and posture changes, pose significant challenges for feature extraction. This complexity underscores the need for specific deep learning network models when applied to 3D face point cloud data.

In response to the challenges encountered by 2D face recognition, such as inaccuracies due to occlusion, pose, and expression transformations, we have opted to explore 3D face recognition methods grounded in deep learning techniques. While these methods are promising, the field still has insufficient 3D face data and a lack of a network model specifically designed for 3D face data. While existing approaches have shown promise, many rely on limited training data or extensive manual preprocessing. For instance, notable methods like the one proposed by Cai et al. [16] have demonstrated significant efficacy but heavily rely on manual data preprocessing, which constrains the model's adaptability to diverse facial structures and feature distributions, resulting in diminished performance on other datasets. Similarly, Zhang et al.'s [17] approach employs transfer learning but lacks a specifically designed network model for 3D face data, making it difficult to achieve the desired effect in real-world applications due to sample uncertainty. To address these issues, we propose a novel network model tailored to the characteristics of facial data, enabling direct processing of original 3D face point cloud data. Leveraging the distinct features of 3D face data, including varied expressions and complex postural changes, we extract 3D information representations efficiently without information loss [18–23]. To overcome data scarcity problems, we leverage synthetic methods [24], employing a Gaussian Process Morphable Model (GPMM) to generate large-scale [25], diverse face scans. With the GPMM, we create faces with random shapes, expression coefficients, and pose transformations, facilitating effective network training with minimal real-world data. Our approach, inspired by KPConv [10], presents a dual-branch network structure; these branches are tailored to handle positive neutral faces and non-neutral faces showing expression and posture changes, leveraging the advantages of dual-branch feature fusion to enhance face recognition performance. Additionally, our method incorporates a local adaptive feature learning module and employs context sampling technology to address the unique challenges posed by 3D face recognition. This approach establishes a comprehensive framework for 3D face recognition tasks, beginning with point clouds. The custom network model based on KPConv is designed to tackle the intricacies of 3D face recognition, presenting a novel solution in this evolving field.

The main contributions of this work are as follows:

(1) To diversify the training data for 3D face recognition, encompassing various identities, expressions, and poses, we introduce a data-enhanced learning framework guided by a Gaussian Process Morphable Model (GPMM). This framework enables effective network training, even with a limited amount of real data.
(2) We propose a dual-branch network structure based on KPConv, adding a local neighborhood adaptive feature learning module designed for direct facial feature extraction from point clouds.
(3) We conduct extensive experiments on established 3D face recognition benchmarks. The results show the competitiveness of our 3D face recognition method and its efficacy in addressing challenging face identification tasks in 3D space.

The rest of this paper is organized as follows:

Section 2 provides an overview of related work, while Section 3 outlines our proposed methodology. Specifically, Section 3.1 details the synthesis of a substantial volume of 3D facial scans through the GPMM in the data generator module. Our approach uses a data-augmented learning framework guided by real data, enhancing the realism of synthetic face scans. Distinguished by its efficiency in both memory and time, our method stands out among other data creation techniques. In Section 3.2, we elaborate on our strategy to utilize a KPConv-based network for extracting 3D facial representations. KPConv learns the local geometric mode of the point cloud by designing the weight of the moving kernel point, but the low-dimensional spatial coordinate relationship does not have enough ability to describe the association of adjacent points. For instance, points with the same relative position may have different semantic relations. In order to better capture facial representations, we designed a novel dual-branch network structure and added an adaptive feature learning module to replace the radius neighborhood sampling strategy of KPConv, which is used to calculate the similarity between inputs and find points and interactions between points, thereby improving the discriminability of the face recognition model in the eigenvector space and the recognition accuracy. Section 4 presents an ablation study and reports identification results on the Bosphorus [26] and FRGC v2.0 [27] datasets. The paper concludes in Section 5.

2. Related Work

2.1. On 3D Face Recognition

Li et al. [28], Guha et al. [29], and Zhang et al. [17] have contributed extensive insights into diverse 3D face recognition (3DFR) techniques. The field of 3DFR has evolved into a useful tool for facial feature identification over recent decades. These approaches are divided into two categories, classical and modern, depending on the technological procedures utilized in recognition. The classical methods focus on extracting distinct facial features—global, local, and hybrid—to facilitate matching. The global feature extraction method seeks to match all the surface features sensitive to facial expressions, such as the baseline algorithm Iterative Closest Point (ICP) introduced by Besl and McKay [30]. Yu et al., through the integration of resampling and denoising procedures into the sparse ICP algorithm, enhanced the accuracy and robustness of facial verification [31]. Local feature-based approaches, as opposed to global features, usually capture unique, compact features that reflect 3D local face information [32,33]. Guo and Da [34] focus on the investigation of a method centered on local descriptors that aims to strengthen systems' resistance to changes in local descriptors. Hybrid feature techniques combine both global point cloud registration and local feature matching [35]. These methods represent the diversity in classical approaches for 3D face recognition, providing valuable insights into global and local feature extraction methods for matching facial features.

The deep learning approach uses complex network architectures and extensive training datasets to derive high-level, meaningful facial features from low-level information. This approach can be divided into three main types: conversion of data from three dimensions to two dimensions, advancement in network architectures, and techniques for facial reconstruction. When converting three-dimensional data to two-dimensional data, it is common to employ depth images for recognition [36,37]. Network performance has been improved by designing deep loss functions with attribute-aware loss functions, such as the one proposed by Jiang et al. [38], and incorporating face attributes like age, gender, and ethnicity into the training process. Additionally, a novel deep learning network with 3D voxel representation methods has also been used for 3D shape recognition [22], which has high memory requirements. Some approaches integrate 3D face reconstruction with deep learning, identifying features from a 3D Morphable Model (3DMM), generating dense deformable models, or locating face landmarks [39,40]. Liu et al. [41] use a cascaded regression approach to reconstruct 2D landmark location estimates along with 3D shapes. Despite their utility, these methods tend to lose 3D geometric structure information. Therefore, we design a 3D face recognition network based on point clouds using deep neural networks with 3D geometric data.

2.2. Deep Learning on Point Clouds

In recent years, there has been a significant surge in the application of deep learning techniques to process point clouds, addressing various challenges in this field [11–15]. Despite these advancements, there remains a substantial need for further exploration in the realm of deep learning on point clouds given the unique complexities associated with using deep neural networks for point cloud processing. Deep learning networks were first used for 3D point cloud processing by PointNet [11]. Using an asymmetric function, PointNet aggregated point-wise features, ensuring that they were unaffected by input point permutations. Building upon PointNet, PointNet++ [12] uses a recursive and hierarchical approach to extract both global and local characteristics from point clouds. The integration of graph CNN into point cloud processing was brought about by techniques like DGCNN [14] and SpecGCN [42]. The EdgeConv technique, which is similar to a convolution, was used in DGCNN to generate a local neighborhood graph and extract relevant local geometric information. Each layer expanded its corresponding region on the graph by considering the nearest neighbors in the feature space. Thomas et al. [10] established stiff and deformable kernel point convolution (KPConv) operators designed specifically for 3D point clouds. Without explicit optimization for facial structures, Bhople et al. [43,44] introduced identification using PointNet, which was originally designed for 3D objects (such as an airplane, chair, and desk). Obviously, it did not work out so well. Our approach builds upon the strengths of KPConv, demonstrating superior effectiveness. We devise a dual-branch network model, utilizing KPConv as the foundation, and introduce a local neighborhood adaptive feature learning module to enhance the extraction of intricate facial details, thus improving the accuracy of 3D face recognition.

2.3. On 3D Face Generation

The efficacy of deep learning outcomes is heavily reliant on the quality of training data. Due to the difficulty of 3D data acquisition, it is largely dependent on hardware devices, which brings challenges to the direct application of deep learning to 3D data. Existing approaches either utilize pre-existing face models or reconstruct 3D faces from images to construct training datasets [45–47]. For instance, Blanz and Vetter [46] enabled the generation of diverse facial models by adjusting parameters, named 3DMM, and developing a flexible and precise method to achieve fine control of face shape and appearance

during synthesis. Deng et al. [45] presented a weakly supervised learning method for accurate 3D face reconstruction, while Guo et al. [47] presented an optimization technique. Gilani and Mian [48] presented a method for simultaneous interpolation across face identity and expression spaces. To address limitations in accuracy and diversity, Bhople et al. [43,44] proposed point cloud-level augmentation methods. However, these methods have constraints such as reconstruction accuracy and dense correspondence establishment. Yu et al. [49] used the GPMM method to generate training data, but they only considered shape and expression coefficients in the process of data generation and did not consider the influence of recognition pose variations. In our work, we adopt a learn-from-synthesize methodology, inspired by Yu et al. [49], aiming to automatically generate a substantial collection of annotated 3D facial scans with diverse identities. Unlike Yu et al. [49], who solely consider shape and expression coefficients in their data generation process, we extend the GPMM method by incorporating a rotation matrix during training data generation. This enhancement ensures that our training dataset encompasses not only variations in facial identity and expression but also accounts for different postural transformations, thereby facilitating effective 3D facial recognition training. As a result, our proposed 3D face recognition model demonstrates improved robustness to changes in position and posture, thereby enhancing its generalization capabilities.

3. Method

In this section, we first prepare the synthetic data for training, then introduce our network architecture. Subsequently, we delve into the specifics of the methodology, illustrated in Figure 1, which shows the framework of the proposed approach.

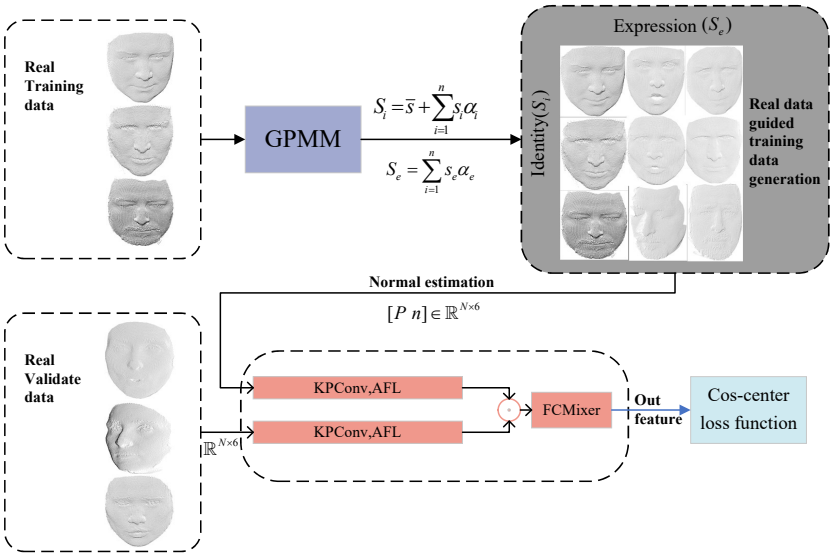

Figure 1. Framework for 3D face point cloud recognition.

3.1. Generation of 3D Face Data

Considering the poor performance of 3DMM's facial details and its limited flexibility, we opted for GPMM, which offers enhanced deformation expression capabilities, to generate 3D face data. We use GPMM to learn a Gaussian distribution based on PCA, encompassing shape, texture, expression, and other attribute features, as expressed in Equation (1):

$$S(p) = \bar{s} + U_s(\alpha_s) + U_e(\alpha_e) \tag{1}$$

where p represents a particular shape formed through a linear combination of the mean vector and each principal component (eigenvector). The mean of all shape vectors is

represented by $\bar{s}$, where α_s is the principal component coefficient, U_s is the eigenvector matrix, U_e is the noise eigenvector matrix, and α_e stands for the noise coefficient.

In this research, we introduced data augmentation techniques, incorporating algorithms for facial expression alterations and face pose variations, to address challenges related to pose and noise in real-world applications. This approach aims to tackle the limitation of limited samples in existing 3D face data and enhance data diversity. Specifically, the noise in Equation (1) is modeled as facial expression changes and face pose variations, as shown in Equation (2):

$$S(p) = \bar{s} + U_s(\alpha_s) + U_e(\alpha_e) + RSLG(\gamma) \tag{2}$$

Equation (2) introduces a noise matrix set $RSLG(\gamma)$ based on Equation (1), encompassing components such as the rotation matrix R, scaling matrix S, displacement matrix L, and a parametric function G to manipulate face pose through pose parameters. As the coefficients α_s, α_e, and γ follow independent normal distributions, a diverse set of 3D face models is generated by randomly sampling from this distribution, incorporating various expressions and poses.

In order to bring our training data closer to real data, we introduce a strategy to generate training data using real data as the principal component coefficients. This involves calculating the expression coefficient α_e and noise parameters γ of each face in the FRGC v2.0 subset of the real-face dataset. During face generation, we randomly select α_e and γ from their respective distributions, resulting in $\alpha_e^{'}$ and $\gamma^{'}$.

$$\alpha_e^{'} = \lambda(\alpha_e) - (1-\lambda)\mu_1 \tag{3}$$

$$\gamma^{'} = \lambda(\gamma) - (1-\lambda)\mu_2 \tag{4}$$

where λ is (0,1), μ follows a normal distribution, and is a random value between $\mu_1 \sim N(0, {\sigma_{\mu_1}}^2)$, $\mu_2 \sim N(0, {\sigma_{\mu_2}}^2)$.

Subsequently, by substituting the values from Equations (3) and (4) into Equation (2), a diverse range of facial expressions resembling those found in the actual dataset can be achieved during face generation. The experimental results demonstrate a substantial improvement in the recognition rate of non-neutral faces using this approach.

In the end, we generated a large-scale face dataset of 10,000 identities (α_s), each containing 200 expressions (α_e) and 120 poses (γ). Figure 2 shows the synthesized data samples, where the first column represents data without additional noise and real-face constraints, while the last three columns include real-face constraints and noise changes. Upon analysis, it is evident that our improved GPMM face synthesis data appear more authentic and reliable. Notably, in Figure 2, we can see that the facial point cloud is different from other point clouds (chairs, tables) and exhibits distinctive features concentrated in local areas such as the eyes, nose, and mouth, undergoing changes based on posture and expression variations. According to this characteristic, we propose a novel dual-branch network structure based on KPConv, featuring a local neighborhood adaptive feature learning module for the extraction of 3D facial features.

3.2. Network Architecture

In real-life scenarios, facial data often involve non-neutral expressions and varying poses. While existing point cloud deep learning networks exhibit strong recognition performance for neutral faces [18,48], their effectiveness diminishes when dealing with natural non-neutral faces. To address this challenge, we propose a dual-branch network structure based on KPConv. This design segregates neutral face data with robust recognition performance and non-neutral face data with natural expressions into two branches for dedicated

processing. Additionally, a local neighborhood feature learning module is incorporated into one branch to selectively extract pertinent information. Subsequently, by merging the information from both branches, we achieve a more comprehensive and enriched feature representation, leading to improved accuracy in face recognition results.

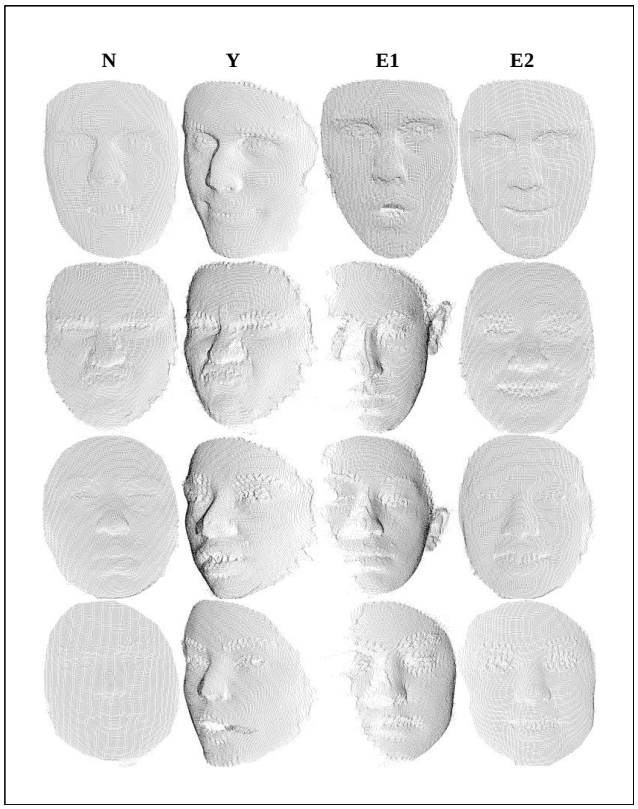

Figure 2. Column 1 is the face we generated, and columns 2, 3, and 4 are the noisy data after adding real-face guidance.

The forward process of our network can be represented as follows:

$$f = 3DRecNet(P_1, P_2) \tag{5}$$

where $P_i = \{x_{p_1}, x_{p_2}, \cdots, x_{p_n}\} \in \mathbb{R}^{N\times 3}$ is the unordered input point cloud pair; N is the number of points; and $f \in \mathbb{R}^{1024}$ is the output feature.

Our proposed network architecture is shown in Figure 3, with subsequent subsections providing detailed introductions to each submodule.

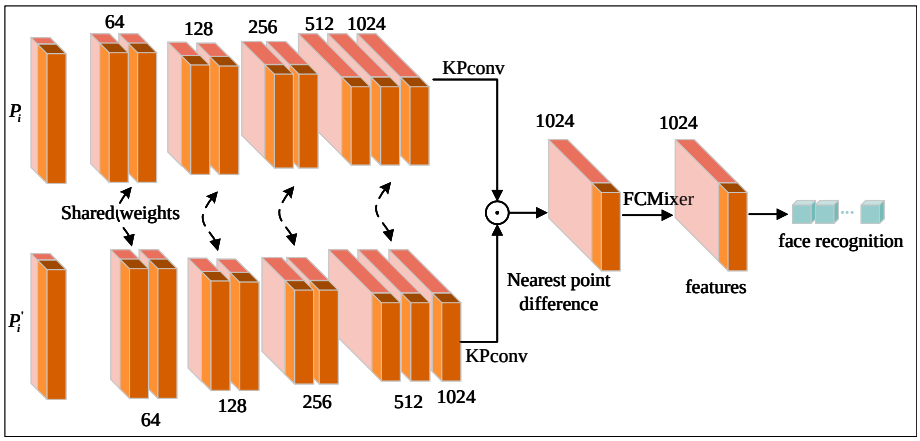

Figure 3. Our KPConv-based dual-branch network architecture for 3D face recognition.

3.2.1. KPConv-Based Dual-Branch Network Structure

Our objective is to enhance the accuracy of face recognition tasks through the utilization of a dual-branch network architecture based on KPConv. This enables improved capturing of similarities across input point clouds, especially in the presence of variations in expression and posture. KPConv introduces a learnable convolution kernel, represented as a kernel point, akin to a spherical area with an adjustable radius. This kernel is employed to calculate features for each point. In Equation (6), the convolution operation is executed on the domain set surrounding each point i, and a weight ϖ is computed for point j within each neighborhood, representing the influence of point j on point i. Subsequently, the eigenvectors of the points in these neighborhoods are weighted and summed based on the corresponding weights to derive the eigenvector θ of point i:

$$f_\theta(x_i) = \sum_{j \in N(x_i)} \varpi(\| s_j - x_i \|) s_j \theta(\| s_j - x_i \|) \tag{6}$$

where $f_\theta(x_i)$ represents the feature of the i-th point, s_j represents the position of the j-th point, θ is the feature vector, ϖ is the weight parameter of the convolution kernel, and $N(x_i)$ is the neighborhood set of the i-th point.

In Figure 4, each point on the point cloud is associated with a convolution kernel. Refer to Section 3.2.2 for details on the determination method of the convolution kernel.

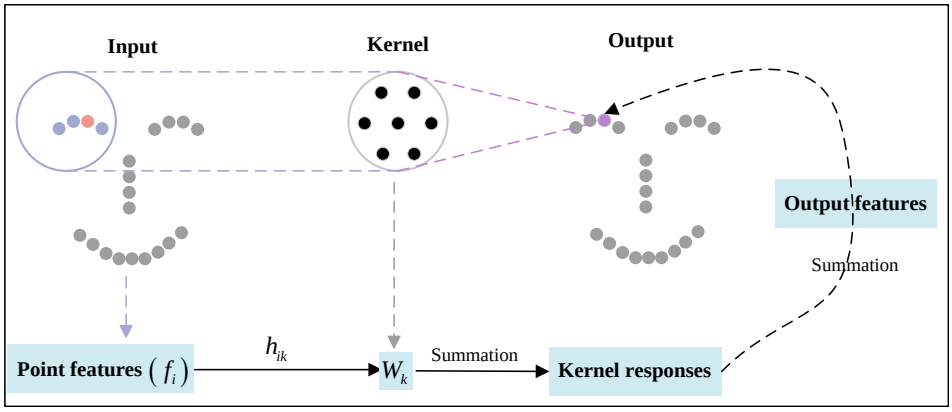

Figure 4. The core weight matrix Ω_θ multiplies each input point feature f_i, and the correlation coefficient $h_{i\theta}$ is determined by the spatial relationship of the point with respect to the core point.

Our dual-branch network takes a set of point clouds (P_1, P_2) as input and processes them through two encoders, each comprising 4 layers and 2 convolutional blocks using KPConv, as shown in Figure 3. To fuse the encoding information from the two encoders, we concatenate the feature differences corresponding to the encoding ratio. To calculate this feature difference, we designed the FCMixer function at the top of the dual-branch network. This function compares each point in P_2 with its nearest spatial point in P_1, obtaining features $(x_i, x_i^{'})$. A linear transformation is applied through the learnable weight matrix (ω_1, ω_2), calculating the difference between the features of each point in P_2 and the features of the nearest point in P_1. The fused feature vector $f(v_i)$ is then obtained as a network output, indicative of the likelihood that two input faces correspond to the same individual. The decoder part of the network consists of a 4-layer stack containing recent upsampling and concatenation stages and a single convolution.

$$f(v_i) = FCMixer(\omega_1 \cdot x_i + \omega_2 \cdot x_i^{'}) \tag{7}$$

where $f(v_i)$ are the fused feature vectors, x_i and $x_i^{'}$ represent the feature vectors of the dual-branch networks, respectively, and ω_1 and ω_2 are the learnable weight matrices.

3.2.2. Adaptive Feature Learning Module for Local Neighborhood

KPConv uses space to generate fixed convolution kernel points, computes the weight matrix through a kernel function, and processes points within a spherical neighborhood. By addressing the challenges posed by the direct and intricate regular grid convolution problem, KPConv determines the points of the spherical neighborhood through methods such as Poisson disks or random points. This approach effectively reduces operational complexity in the convolution process.

However, in 3D face recognition, traditional convolutional approaches, such as the rigid convolution kernel used in KPConv, face challenges due to the non-rigid nature of three-dimensional faces. These faces exhibit strong geometric irregularities caused by variations in expression and posture, resulting in uneven sides and fronts and clustering in specific dense areas. Unlike flat surfaces like chairs and tables, 3D faces demand a more adaptive convolutional approach. The rigid convolution kernel of KPConv, employing a system of attraction and repulsion, indiscriminately considers features from local neighborhood points, limiting its effectiveness in recognizing faces from point cloud data. For instance, attributing equal importance to the densest points in the forehead center and the regions around the eyes and nose may overlook crucial information related to facial shape and expression. Therefore, integrating contextual information from various facial areas, such as the connection between the forehead, eyebrows, and eyes, proves essential for effective 3D facial recognition.

We propose an adaptive feature learning (AFL) module designed to merge the local neighborhood features of each point in the point cloud with the contextual features of the local environment. In this process, the AFL module effectively modulates the contribution of each point to the target feature through calculating the weighted influence based on their relative positions and importance. Although traditional normalization methods are not employed during feature fusion, the adaptive calculation of each point's influence, considering its relative relationships and significance in the local region, achieves a similar effect. This mechanism ensures balanced feature fusion while mitigating potential over-smoothing issues that can arise from conventional normalization techniques. The AFL module dynamically adjusts the radius neighborhood range, identifies neighboring points, learns the impact of each point on others to refine features, and concurrently reduces the computational burden of convolution. This adaptive adjustment enables AFL to effectively integrate inter-neighborhood contextual information into point features, significantly enhancing its capacity to characterize local neighborhoods. Figure 5 provides an overview of the AFL module, a method adept at extracting local neighborhood context through dense interconnections among points.

Given a region R and its feature set $P_i = \{x_{p_1}, x_{p_2}, \cdots, x_{p_n}\}$, we introduce the adaptive feature learning (AFL) module. The AFL module is designed to augment point features within P by acquiring contextual information from local neighborhoods.

$$\begin{aligned} P_i^{'} &= P_i + \Delta P_i \\ \Delta P_i &= F(P_i, P), \forall P_i \end{aligned} \tag{8}$$

where $P_i^{'}$ is the feature enhancement of P_i, and P is the feature set after the feature mechanism F is aggregated.

The feature mechanism F efficiently facilitates the exchange and aggregation of information within a local region P by adaptively learning the influence of each feature in P on each P_i. It is mathematically expressed as follows:

$$\xi_{ij} = F(P_i, P) = \sum_{j=1}^{n} M(g(P_i, P_j)) \cdot p_{rel}(P_i, P_j) \tag{9}$$

Here, $M(g(P_i, P_j))$ computes the influence of P_j on P_i, denoted as ξ_{ij}, and p_{rel} represents the relationship between P_j and P_i. Notably, we account for P_i's self-influence on F.

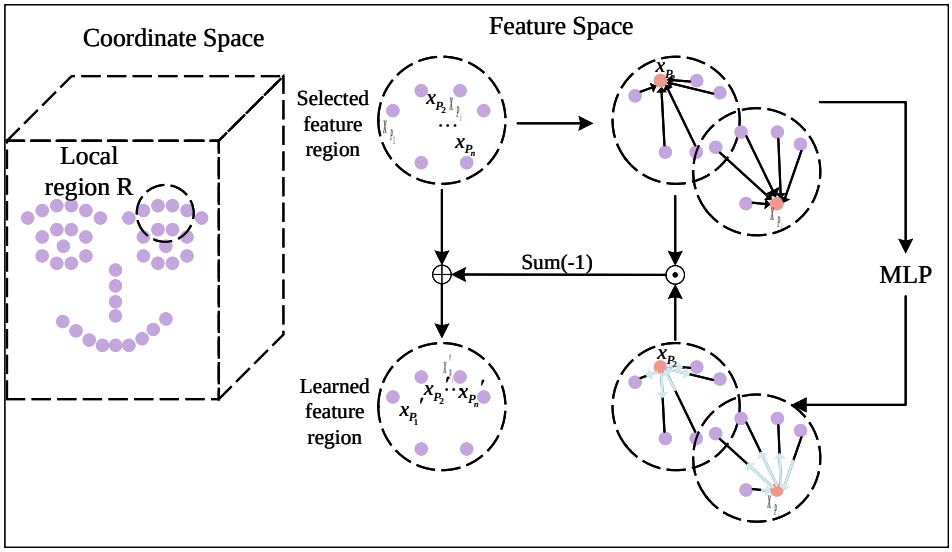

Figure 5. In our adaptive feature learning (AFL) module in the local region R, each feature P_i experiences the influence of other features, and the strength and direction of this influence are dynamically determined by the coefficients (i) and j based on the differences in feature vectors. This adaptive learning process aims to enhance the descriptive power of the output P_i' to better capture the characteristics of the entire region.

The influence function $g(P_i, P_j)$ between features P_i and P_j is calculated using the M network to obtain the influence index ξ_{ij} between P_i and P_j. Here, the function g combines P_i and P_j, and ξ_{ij} serves as an indicator of the influence of P_j on P_i. There are four simple ways to model a function g. These include no combination (AFL-non), combination by feature summation (AFL-sum), feature subtraction (AFL-sub), and feature concatenation (AFL-Con). The effectiveness of these methods is demonstrated in the following experiments. Throughout this process, the M network learns to calculate ξ_{ij} and represent the influence of each P_j on P_i.

The purpose of the relation function p_{rel} is to ascertain how the influence indicator ξ_{ij} affects P_i.

$$p_{rel}(P_i, P_j) = \begin{cases} P_i - P_j, & if\ i \neq j \\ P_i, & if\ i = j \end{cases} \tag{10}$$

where when $i = j$, p_{rel} is P_i, and when $i \neq j$, p_{rel} is $P_i - P_j$.

After the enhancement of each feature P_i in the local region R by the feature mechanism, the final result is as follows:

$$P_i' = \alpha_i^{(i)} \cdot P_i + \sum_{j=1, j\neq i}^{n} \alpha_j^{(i)} \cdot (P_j - P_i) \tag{11}$$

$$\alpha_j^{(i)} = \begin{cases} -p_{imp}(P_i, P_j), & if\ i \neq j \\ 1 + p_{imp}(P_i, P_j), & if\ i = j \end{cases} \tag{12}$$

In accordance with Equation (12), each feature P_i within the local neighborhood R undergoes the influence of a force field-like effect from the constructed adaptive feature learning (AFL) module. P_i is subjected to forces exerted by each feature in the feature space, resulting in either attraction or repulsion. The adaptive learning coefficient ξ_{ij}, influenced by the difference between the two feature vectors, determines the magnitude and direction of the force. Consequently, the output $P_i^{'}$ furnishes a more comprehensive characterization of the region by incorporating contextual information from the entire region.

3.3. Loss Function

In addressing the significant inter-class variations and intra-class similarities observed in face data, our network optimization incorporates an enhanced central loss function. This function calculates the distance between each feature vector in the point cloud and the associated category center, utilizing cosine similarity as a replacement for the original Euclidean distance. The adoption of cosine similarity aims to assess the angle between two feature vectors, emphasizing the direction of the face data. This approach proves more effective for face learning, especially when dealing with feature vectors of varying lengths. We refer to several existing works, such as SphereFace [50] and ArcFace [7], which adopt cosine similarity to measure the similarity between facial feature vectors and achieve significant performance improvement. In addition, the application of cosine similarity has also been verified in other computer vision tasks, especially in the case of dealing with high dimensionality, sparse features, and inconsistent feature lengths. Compared with Euclidean distance, cosine similarity can provide a more stable learning process and is more effective for face learning.

$$L_c = \frac{1}{2N}\sum_{i=1}^{N}\|d(x_i, x_i^{'}) - c_{y_i}\|_2^2 + \lambda\sum_{j=1}^{m}\|c_j\|_2^2 \tag{13}$$

where N is the total number of point clouds, d(·) calculates the cosine distance between two vectors, c_{y_i} represents the category center of the category to which the i-th sample belongs y_i, m is the total number of categories, c_j is the category center of the j-th category, and λ is the regularization term coefficient.

3.4. Implementation Details

Trained using facial scans comprising N = 24,000 points, the network is utilized for classification training. Each point in the dimensional space is characterized by Euclidean coordinates (x, y, z) and associated normal vectors (n_x, n_y, n_z). We train the proposed network using PyTorch. With the exception of the final classification layer, all layers are batch-normalized using the Adam optimizer. Every 20 epochs, the initial learning is lowered by a factor of 10 before being reset to 10^{-3}. Additionally, weight decay begins at 0.5 and reduces by 0.5 each time it reaches 0.99. The network is trained on a single NVIDIA GeForce GTX 3060TI GPU for a total of 150 epochs with a batch size of 10 scans.

4. Experiments

First, the dataset utilized in this study and the data pretreatment processes before the experiments are described in this section. Then, for 3D face recognition, we assess the performance of the suggested dual-branch network structure using synthetic training data. Finally, we use two open-source 3D face benchmarks to evaluate our methodology.

4.1. Datasets

In this research, we evaluate our proposed 3D face recognition network using two publicly available datasets: FRGC v2.0 and Bosphorus.

FRGC v2.0 (Face Recognition Grand Challenge version 2.0) is a facial recognition dataset released by NIST in 2006. It contains about 1432 facial images of 466 people taken under different lighting and expressions, with individual still images, video sequences, and 3D reconstructed models, as well as gender, ethnicity, and age information. The composition of the dataset consists of two parts: Gallery and Probe. The Gallery set contains static frontal images of subjects, while the Probe set consists of query images seeking the most similar image in the Gallery, evaluating retrieval performance. In experiments, a subset of FRGC v2.0 serves as real data for network training, with 443 faces allocated for the real-data validation set and 524 faces for guiding the generation of real data.

Bosphorus, a collaborative project from Bogazici University, Turkey, focuses on collecting facial expression and shape information for 3D face modeling. It includes data from 105 individuals, with two to four facial expressions per individual, resulting in 4665 images. The dataset offers high-resolution 3D facial models with detailed information for accurate facial analysis. The Bosphorus dataset contributes to the evaluation of the proposed methodology.

4.2. Data Preprocessing

We conduct preprocessing on the acquired 3D face dataset obtained in Section 3.1, involving operations like point unification, normal estimation, and coordinate transformation.

The normal vector, a crucial characteristic of a point cloud, $n \in \mathbb{R}^{N\times 3}$ is computed for the input point cloud P using principal component analysis. The output of the normal estimation submodule is denoted as $[P_n] \in \mathbb{R}^{N\times 6}$.

In the context of 3D face recognition, the tip of the nose serves as the reference point for normalization. The coordinate transformation involves setting the coordinates of the nose tip as the origin $(0, 0, 0)$ for all points in the 3D point cloud data. To mitigate potential interference from non-face areas, a sphere with a radius of 90 mm is constructed based on the nose tip. Points outside this sphere are removed, ensuring the retention of only face-related point cloud data. Additionally, potential outliers or noise are eliminated by setting a threshold for the nearest neighbor, excluding points that are excessively distant or deviate from the facial structure.

Through these preprocessing steps, non-face regions and outliers in the 3D face point cloud data are effectively addressed, providing normal estimation and a refined and consistent representation of the data with the nose tip as the coordinate origin.

4.3. Ablation Study

Training the network on our hardware takes approximately 80 h with a dataset comprising 5000 identities, each identity has 200 different expressions, unless otherwise specified. For the ablation investigation, the evaluation primarily focuses on the accuracy on the Bosphorus dataset, with emphasis on a subset of neutral scans identified by the filename convention "N_N_0" from the Bosphorus dataset. This approach allows for expedited and efficient comparisons.

4.3.1. Effectiveness of the Local Neighborhood Feature Learning Module (AFL)

We conducted ablation studies on the Bosphorus dataset, categorizing experiments into those without the AFL module (baseline) and those with the AFL module. Following the principles outlined in Section 3.2.2 for the AFL module, we explored four different styles of the combination function g in prel within each local group. These styles included no combination (AFL-non), combination by feature summation (AFL-sum), feature subtraction (AFL-sub), and feature concatenation (AFL-Con). Quantitative results are presented in Table 1, where "rank-1 identification rate" signifies the recognition rate at which the match-

ing item returned by the recognition algorithm ranks first among all possible matching items after comparing the query image with all items in the database.

Table 1. Whether the AFL module is added in the ablation study.

Method	Rank-1 Identification Rate (%)
Baseline	95.51
AFL-non	96.01
AFL-sum	94.37
AFL-sub	97.19
AFL-Con	96.34

Learning only the adjustment of individual features without interaction with other features hampers the exploitation of contextual information in local regions. Summation operations, which involve summing pairwise features, may diminish the discriminative ability for local area features, impacting recognition capability. Conversely, feature concatenation renders some feature representations nearly identical, falling short of the discriminative power achieved by the subtraction operation. The subtraction operation ensures each feature is unique to the combination of P_i and P_j, enhancing classification ability. Consequently, we selected the local feature subtraction operation as the adjustment method due to its superior discriminative and representative characteristics compared to other alternatives. On the Bosphorus dataset, the rank-1 recognition rate increased by 1.68, demonstrating the notable effectiveness of the local neighborhood feature learning module.

As shown in Figure 6, the inclusion of the local neighborhood feature learning module yields a substantial enhancement in facial feature extraction compared to scenarios where the module is absent. This improvement allows for a more effective capture of prominent feature variations among facial regions, thereby better reflecting individual differences in faces.

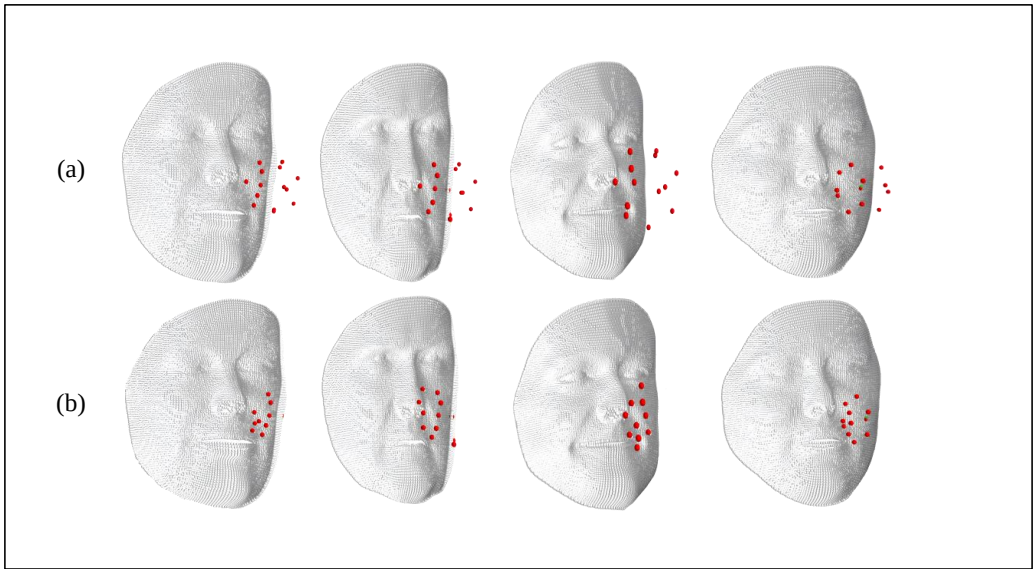

Figure 6. Local neighborhood feature selection, where (**a**) is without adding the AFL module and (**b**) is the change after adding the AFL module.

4.3.2. Effectiveness of the Dual-Branch Network Structure

Compared with the KPConv network, the introduction of a dual-branch network structure aims to process two inputs through the same network, facilitate focused feature extraction at different scales in each branch, calculate input similarities, and share weights to reduce parameters. So that the feature advantage in positive neutral faces can be effectively

applied to non-neutral faces, the subsequent merging of branch outputs achieves the fusion of information, leading to improved model training efficiency and generalization.

Table 2 outlines the configuration and training parameters of various networks, maintaining a consistent external structure for comparative experiments.

Table 2. A list of the deep learning training parameters.

	Optimizer	Initial Learning Rate	Learning Rate Scheduler	Dropout	Loss	Batch Size
PointNet	Adam	0.001	Step	No	NLL	32
PointNet++	Adam	0.001	Step	No	NLL	24
KPConv	SGD	0.001	Exponential	Yes	NLL	10
Dual-branch KPConv	SGD	0.001	Exponential	No	NLL	10

Table 3 demonstrates the consistent superiority of the dual-branch network structure, regardless of the backbone architecture (PointNet, PointNet++, and KPConv) and input form (Points, Points + Normals). Specifically, the rank-1 recognition rate of the KPConv network surpasses that of the PointNet and PointNet++ networks by 7.94 and 3.44, respectively. Moreover, the KPConv network employing the dual-branch topology achieves a higher rank-1 recognition rate (1.82) compared to the standalone KPConv network. These findings validate the significant contribution of the proposed dual-branch network and the input form using the "Points+Normals" (x, y, z, n_x, n_y, n_z) layout to the improvement of 3D face recognition accuracy. Subsequent tests utilize both "Points+Normals" (x, y, z, n_x, n_y, n_z) from our suggested 3D facial scans as input modalities.

Table 3. The effectiveness of dual-branch network structure on the Bosphorus database.

Method	Modality	Rank-1 Identification Rate (%)
PointNet	Points	87.53
PointNet++	Points	92.03
KPConv	Points	95.47
Dual-branch KPConv	Points	97.29
PointNet	Points+Normals	91.60
PointNet++	Points+Normals	94.10
KPConv	Points+Normals	96.75
Dual-branch KPConv	Points+Normals	98.83

4.3.3. Effectiveness of the 3DRecNet Network Architecture

Here, we provide visualization results that validate the performance of the KPConv-based 3DRecNet network that uses a dual-branch network topology together with a local neighborhood feature learning module customized for 3D face recognition.

Figure 7 presents a t-SNE visualization of depth features obtained from various network models projected into a two-dimensional embedded space [51]. While KPConv exhibits tightly grouped depth features compared to PointNet and PointNet++, occasional clustering errors are observed. In contrast, the 3DRecNet model designed in this study demonstrates a slightly superior performance to the KPConv model, showcasing its ability to extract more discriminative features.

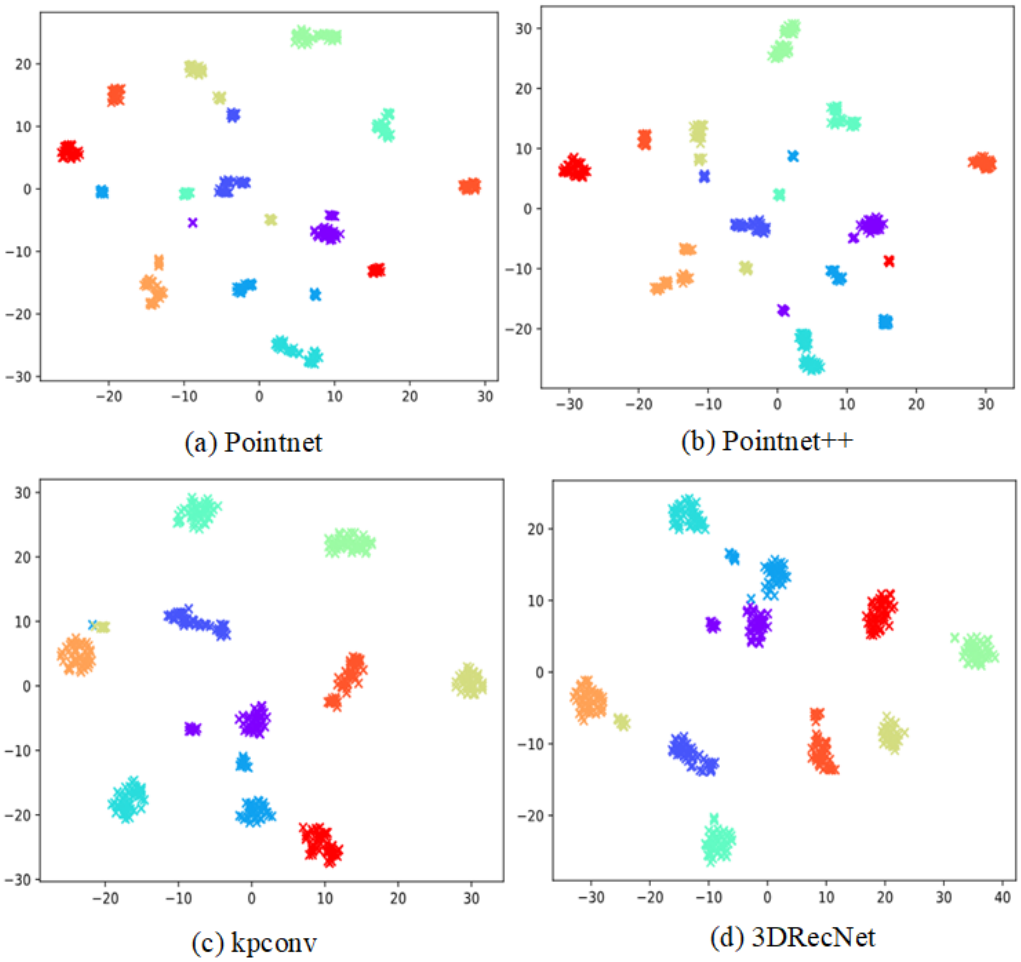

Figure 7. The facial features that were recovered from the different models—PointNet, PointNet++, and KPConv—as well as the suggested model, 3DRecNet, are displayed in the t-SNE visualization. Each color in the visualization corresponds to a different identity.

4.3.4. Effectiveness of the Real-Data-Guided Generation

While achieving a peak accuracy of 98.83% with the initially generated dataset, our subsequent experiments revealed the potential presence of overfitting phenomena. To address this concern and validate the training efficacy, we propose a novel data generation approach guided by real data. Further experiments are conducted on constrained subsets of real data to mitigate overfitting. In these experiments, approximately half of the faces in the FRGC v2.0 sample subset are randomly selected for real-data-guided generation, forming the real-data validation set with the remaining half. The results indicate that the integration of the real-data-guided generation strategy with the initial generation method significantly enhances the rank-1 recognition rate on the Bosphorus dataset and effectively mitigates overfitting. This strategy not only sustains high accuracy but also improves the model's generalization ability, offering a robust solution to the overfitting challenge.

Figure 8 presents accuracy curves for both the training and test sets, contrasting the baseline model with the inclusion of real data. Despite an escalation in disturbances under real-data guidance, there is a gradual rise in the accuracy rate, enhancing the generalization ability of the model. This underscores the efficacy of genuine 3D face data augmentation in addressing overfitting challenges.

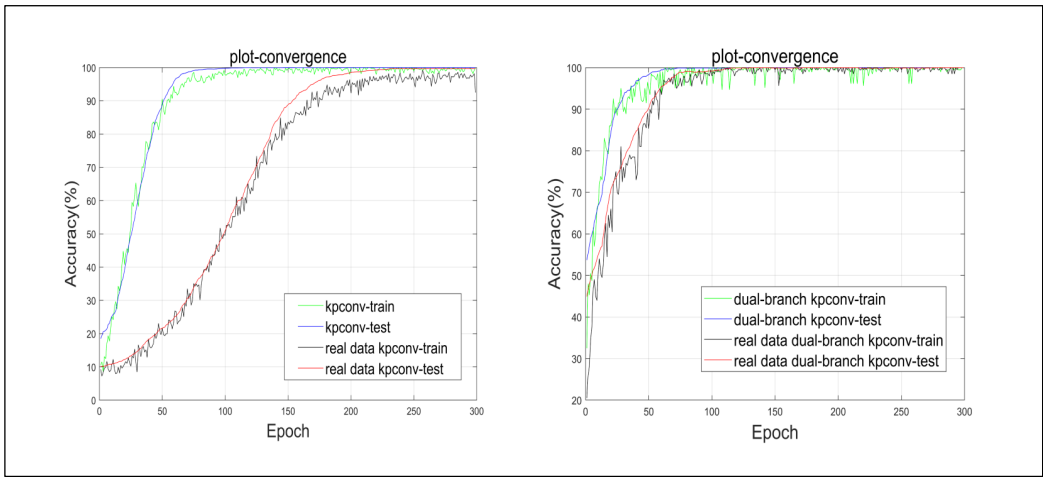

Figure 8. A detailed demonstration of the performance of the KPConv and dual-branch KPConv backbones is presented, considering their effectiveness on both training and test sets and whether they use real 3D face data.

4.3.5. Effectiveness of the Training Data Volume

While generating training data is needed, an excessive amount of data may lead to inefficient use of time and resources, resulting in reduced recognition efficiency. Striking a balance between recognition efficiency and the volume of training data is crucial. Through extensive experimentation, we determined that using 10,000 training samples, each comprising 200 expressions, achieved an optimal rank-1 recognition rate of 99.72% on the Bosphorus dataset, as depicted in Figure 9. Although a slightly higher recognition rate may be achievable with a larger dataset, the associated increase in time and resource costs is considered unnecessary.

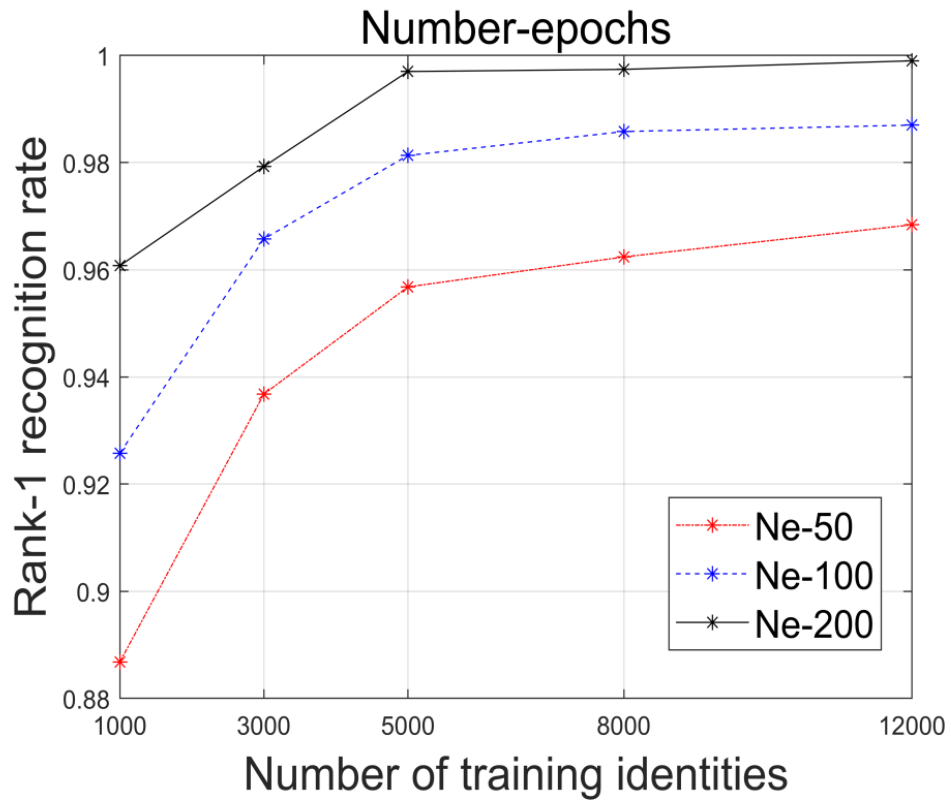

Figure 9. Contrasts based on various training volumes. The number of created expressions for each identity is shown by N_e.

4.4. Comparison with Other Methods

After conducting the ablation investigation in Section 4.3, we selected the 3DRecNet network architecture with the "Point+Normal" input modality as our final model. In this section, we compare our finalized model with earlier state-of-the-art techniques on two widely known public 3D face datasets: Bosphorus and FRGC v2.

4.4.1. Results on FRGC v2.0

In Table 4, we present a comparison between our proposed method and other face recognition algorithms using the FRGC v2.0 dataset.

Table 4. Rank-1 recognition rate (RR1) on FRGC v2.0 dataset.

Method	Rank-1 Identification Rate (%)	Time Cost (s)
Huang et al. [52]	97.60	3.28
Liu et al. [53]	96.94	4.4
Elaiwat et al. [54]	97.10	5.3
Lei et al. [55]	96.30	3.16
Gilani and Mian [48]	97.06	4.02
Gilani et al. [56]	98.50	3.8
Cai et al. [16]	100	3.57
Zhang et al. [17]	99.46	2.6
Yu et al. [49]	98.85	4.43
Ours	99.37	2.35

The presented table illustrates that some advancements in deep learning-based methods have demonstrated impressive recognition rates. Particularly, approaches artificial feature extraction and transfer learning, such as Zhang et al. [17] and Cai et al. [16], exhibit substantial improvements in recognition accuracy. However, the effectiveness of these methods is constrained by challenges such as the uncertainty of test samples and the need for privacy protection of sample data. Our method's dual-branch network model designed for face-data features is still competitive among methods without transfer learning, achieving a notable 99.37% rank-1 recognition rate on the FRGC v2.0 dataset.

We evaluate the time complexity between the proposed method and the compared methods. Specifically, we analyze the computation time of preprocessing and recognition matching for each probe, since in 3D face recognition systems, the time consumption is usually due to the fact that the probe face needs to be matched with the entire gallery set. In this experiment, preprocessing includes the time to process raw 3D data and extract features. From Table 4, we can see that the time consumed by our proposed method is 2.35 s, making it the least time-consuming among all methods.

In contrast to approaches that necessitate larger datasets for marginal gains in recognition rates, our method excels in training with only the 967 faces from the FRGC v2.0 sample. Despite the limited real data, our methodology surpasses some specific approaches, underscoring its efficacy and potential.

4.4.2. Results on Bosphorus

An additional experiment was carried out on the Bosphorus dataset to validate the effectiveness of our proposed method. Table 5 provides a comparative analysis of our methodology with other techniques using the Bosphorus dataset. The recognition rates are reported for an identical subset of the Bosphorus dataset to ensure a fair and consistent evaluation.

Table 5. Rank-1 recognition rate (RR1) on Bosphorus dataset.

Method	Rank-1 Identification Rate (%)	Time Cost (s)
Huang et al. [52]	97.00	3.16
Liu et al. [53]	95.63	4.08
Berretti et al. [57]	95.67	5.25
Lei et al. [55]	98.90	2.9
Gilani et al. [56]	98.50	3.55
Cai et al. [16]	99.75	3.3
Zhang et al. [17]	99.68	1.82
Yu et al. [49]	99.33	5.46
Ours	99.72	2.06

Notably, our results closely align with those of Cai et al. [16], achieving a rank-1 recognition rate of 99.72%. The key distinction lies in our method's utilization of cosine similarity between feature embeddings as a classifier for matching scores. Furthermore, our method is nearly 0.3 percentage points higher than the recognition rate of Yu et al. [49], who also used GPMM to generate training data with a small amount of real data. This is primarily attributed to the inclusion of a rotation matrix in our GPMM method, enabling us to capture more training data reflecting changes in attitude, enhancing the generalization capability of our model. In terms of time complexity, our proposed method takes 2.06 s, second only to Zhang et al.'s [17] method, but their recognition rate is lower than ours. Therefore, compared with the existing methods, the proposed method has higher computational efficiency and can perform face recognition faster.

5. Conclusions

In this research, we propose 3DRecNet, an innovative end-to-end deep learning network tailored for 3D facial recognition using point clouds. To address the challenge of limited training data, our approach leverages the Gaussian Process Morphable Model (GPMM) learning-from-synthesis technique, generating diverse 3D face scans with various identities and expressions. Unlike previous methods that reconstruct 3D faces from photos or interpolate between them, our approach excels in creating realistic face scans in terms of both achieving this at larger scales and in shorter times.

Additionally, we introduce a novel point cloud network specifically designed for 3D facial recognition, addressing performance constraints in face recognition compared to generic object-based point cloud networks. Our local neighborhood adaptive feature learning module focuses on utilizing contextual cues from nearby areas to enhance face feature representation. This learning-based technique outperforms traditional 3D face recognition algorithms by capturing more abstract and high-level characteristics, providing resilience against various changes without relying on human-defined feature descriptions. In contrast to methods that incorporate depth information into 2D images to simulate or reconstruct 3D structures, our approach directly operates on point cloud data, eliminating the need for a laborious face registration phase. Employing a dual-branch network structure and various data augmentation approaches enhances training efficacy by capturing feature changes before and after processing. Comprehensive tests and comparisons on the FRGC v2.0 and Bosphorus datasets validate the superiority of our 3D face recognition system over other techniques, showcasing its resilience and efficiency in tasks such as face recognition and verification.

In future research, we aim to (1) explore face recognition with added temporal changes, including aging effects, to broaden the applicability of our method; (2) integrate generative adversarial networks (GANs) to adapt to point cloud data, address fraud concerns,

and achieve improved 3D face recognition results. By combining the data diversity and realism of GAN-generated face data with the fine-grained control and robustness of the GPMM, we aim to further enhance recognition accuracy, especially in challenging conditions such as extreme facial expressions or occlusions; and (3) investigate the incorporation of meta-learning into our training framework to enhance network performance.

Author Contributions: Conceptualization, X.Y. and X.Z.; methodology, X.Y.; validation, X.Z.; investigation, X.Y.; writing—original draft preparation, X.Y. and X.Z.; writing—review and editing, X.Y., supervision, X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Fundamental Research Funds for the Central Universities under Grant 2020ZDPY0223.

Institutional Review Board Statement: No applicable.

Informed Consent Statement: No applicable.

Data Availability Statement: The data presented in this study are openly available at https://www.nist.gov/programs-projects/face-recognition-grand-challenge-frgc (FRGC V2.0) accessed on 5 March 2024 and https://github.com/huyhieupham/3D-Face-Recognition?tab=readme-ov-file (Bosphorus) accessed on 10 April 2024.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Chadha, A.R.; Vaidya, P.P.; Roja, M.M. Face recognition using discrete cosine transform for global and local features. In Proceedings of the 2011 International Conference on Recent Advancements in Electrical, Electronics and Control Engineering, Sivakasi, India, 15–17 December 2011; pp. 502–505. [CrossRef]
2. Akhtar, Z.; Rattani, A. A Face in any Form: New Challenges and Opportunities for Face Recognition Technology. *Computer* **2017**, *50*, 80–90. [CrossRef]
3. Zhalehpour, S.; Akhtar, Z.; Eroglu Erdem, C. Multimodal emotion recognition with automatic peak frame selection. In Proceedings of the 2014 IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA) Proceedings, Alberobello, Italy, 23–25 June 2014; pp. 116–121. [CrossRef]
4. Turk, M.; Pentland, A. Face recognition using eigenfaces. In Proceedings of the 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Maui, HI, USA, 3–6 June 1991; pp. 586–591. [CrossRef]
5. Belhumeur, P.; Hespanha, J.; Kriegman, D. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1997**, *19*, 711–720. [CrossRef]
6. Ahonen, T.; Hadid, A.; Pietikäinen, M. Face Recognition with Local Binary Patterns. In Proceedings of the Computer Vision-ECCV 2004, Prague, Czech Republic, 11–14 May 2004; Springer: Berlin/Heidelberg, Germany, 2004; pp. 469–481.
7. Deng, J.; Guo, J.; Xue, N.; Zafeiriou, S. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 4685–4694. [CrossRef]
8. Guo, Y.; Zhang, L.; Hu, Y.; He, X.; Gao, J. MS-celeb-1M: A dataset and benchmark for large-scale face recognition. In Proceedings of the Computer Vision-ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Springer International Publishing: Cham, Switzerland, 2016; pp. 87–102. [CrossRef]
9. 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]
10. Thomas, H.; Qi, C.R.; Deschaud, J.E.; Marcotegui, B.; Goulette, F.; Guibas, L. KPConv: Flexible and Deformable Convolution for Point Clouds. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6410–6419. [CrossRef]
11. Charles, R.Q.; Su, H.; Kaichun, M.; Guibas, L.J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 77–85. [CrossRef]
12. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; pp. 5100–5109.

13. Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; Chen, B. PointCNN: Convolution on X-transformed points. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; pp. 820–830.
14. Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M. Dynamic graph Cnn for learning on point clouds. *ACM Trans. Graph.* **2019**, *38*, 146. [CrossRef]
15. Liu, J.; Ni, B.; Li, C.; Yang, J.; Tian, Q. Dynamic Points Agglomeration for Hierarchical Point Sets Learning. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 7545–7554. [CrossRef]
16. Cai, Y.; Lei, Y.; Yang, M.; You, Z.; Shan, S. A fast and robust 3D face recognition approach based on deeply learned face representation. *Neurocomputing* **2019**, *363*, 375–397. [CrossRef]
17. Zhang, Z.; Da, F.; Yu, Y. Learning directly from synthetic point clouds for "in-the-wild" 3D face recognition. *Pattern Recognit.* **2022**, *123*, 108394. [CrossRef]
18. Kim, D.; Hernandez, M.; Choi, J.; Medioni, G. Deep 3D face identification. In Proceedings of the 2017 IEEE International Joint Conference on Biometrics (IJCB), Denver, CO, USA, 1–4 October 2017; pp. 133–142. [CrossRef]
19. Soltani, A.A.; Huang, H.; Wu, J.; Kulkarni, T.D.; Tenenbaum, J.B. Synthesizing 3D Shapes via Modeling Multi-view Depth Maps and Silhouettes with Deep Generative Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2511–2519. [CrossRef]
20. Zhang, J.; Hou, Z.; Wu, Z.; Chen, Y.; Li, W. Research of 3D face recognition algorithm based on deep learning stacked denoising autoencoder theory. In Proceedings of the 2016 8th IEEE International Conference on Communication Software and Networks (ICCSN), Beijing, China, 4–6 June 2016; pp. 663–667. [CrossRef]
21. Feng, Y.; Zhang, Z.; Zhao, X.; Ji, R.; Gao, Y. GVCNN: Group-View Convolutional Neural Networks for 3D Shape Recognition. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 264–272. [CrossRef]
22. Qi, C.R.; Su, H.; Nießner, M.; Dai, A.; Yan, M.; Guibas, L.J. Volumetric and Multi-view CNNs for Object Classification on 3D Data. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 5648–5656. [CrossRef]
23. Jackson, A.S.; Bulat, A.; Argyriou, V.; Tzimiropoulos, G. Large Pose 3D Face Reconstruction from a Single Image via Direct Volumetric CNN Regression. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 1031–1039. [CrossRef]
24. Richardson, E.; Sela, M.; Kimmel, R. 3D Face Reconstruction by Learning from Synthetic Data. In Proceedings of the 2016 Fourth International Conference on 3D Vision (3DV), Stanford, CA, USA, 25–28 October 2016; pp. 460–469. [CrossRef]
25. Lüthi, M.; Gerig, T.; Jud, C.; Vetter, T. Gaussian Process Morphable Models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 1860–1873. [CrossRef]
26. Savran, A.; Alyuz, N.; Dibeklioglu, H.; Celiktutan, O.; Gokberk, B.; Sankur, B.; Akarun, L. Bosphorus database for 3D face analysis. In Proceedings of the Biometrics and Identity Management: First European Workshop, Roskilde, Denmark, 7–9 May 2008; pp. 47–56.
27. Phillips, P.; Flynn, P.; Scruggs, T.; Bowyer, K.; Chang, J.; Hoffman, K.; Marques, J.; Min, J.; Worek, W. Overview of the face recognition grand challenge. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 947–954. [CrossRef]
28. Li, M.; Huang, B.; Tian, G. A comprehensive survey on 3D face recognition methods. *Eng. Appl. Artif. Intell.* **2022**, *110*, 104669. [CrossRef]
29. Guha, R. A Report on Automatic Face Recognition: Traditional to Modern Deep Learning Techniques. In Proceedings of the 2021 6th International Conference for Convergence in Technology (I2CT), Pune, India, 2–4 April 2021.
30. Besl, P.; McKay, N.D. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **1992**, *14*, 239–256. [CrossRef]
31. Yu, Y.; Da, F.; Guo, Y. Sparse ICP With Resampling and Denoising for 3D Face Verification. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 1917–1927. [CrossRef]
32. Yu, C.; Zhang, Z.; Li, H.; Sun, J.; Xu, Z. Meta-learning-based adversarial training for deep 3D face recognition on point clouds. *Pattern Recognit.* **2023**, *134*, 109065. [CrossRef]
33. Kong, W.; You, Z.; Lv, X. 3D face recognition algorithm based on deep Laplacian pyramid under the normalization of epidemic control. *Comput. Commun.* **2023**, *199*, 30–41. [CrossRef] [PubMed]
34. Guo, B.; Da, F. Expression-Invariant 3D Face Recognition Based on Local Descriptors. *Jisuanji Fuzhu Sheji Yu Tuxingxue Xuebao/J. Comput.-Aided Des. Comput. Graph.* **2019**, *31*, 1086–1094. [CrossRef]
35. Guo, Y.; Lei, Y.; Liu, L.; Wang, Y.; Bennamoun, M.; Sohel, F. EI3D: Expression-invariant 3D face recognition based on feature and shape matching. *Pattern Recognit. Lett.* **2016**, *83*, 403–412. [CrossRef]
36. Liang, B.; Wang, Z.; Huang, B.; Zou, Q.; Wang, Q.; Liang, J. Depth map guided triplet network for deepfake face detection. *Neural Netw.* **2023**, *159*, 34–42. [CrossRef]

37. Jin, B.; Cruz, L.; Gonçalves, N. Pseudo RGB-D Face Recognition. *IEEE Sens. J.* **2022**, *22*, 21780–21794. [CrossRef]
38. Jiang, L.; Zhang, J.; Deng, B. Robust RGB-D Face Recognition Using Attribute-Aware Loss. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2552–2566. [CrossRef]
39. Zhao, S.; Wang, X.; Zhang, D.; Zhang, G.; Wang, Z.; Liu, H. FM-3DFR: Facial Manipulation-Based 3-D Face Reconstruction. *IEEE Trans. Cybern.* **2023**, *54*, 209–218. [CrossRef]
40. Zhu, X.; Yu, C.; Huang, D.; Lei, Z.; Wang, H.; Li, S.Z. Beyond 3DMM: Learning to Capture High-Fidelity 3D Face Shape. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 1442–1457. [CrossRef]
41. Liu, F.; Zhao, Q.; Liu, X.; Zeng, D. Joint Face Alignment and 3D Face Reconstruction with Application to Face Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 664–678. [CrossRef]
42. Wang, C.; Samari, B.; Siddiqi, K. Local Spectral Graph Convolution for Point Set Feature Learning. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; Volume 11208 , pp. 56–71.
43. Bhople, A.R.; Shrivastava, A.M.; Prakash, S. Point cloud based deep convolutional neural network for 3D face recognition. *Multimed. Tools Appl.* **2021**, *80*, 30237–30259. [CrossRef]
44. Bhople, A.R.; Prakash, S. Learning similarity and dissimilarity in 3D faces with triplet network. *Multimed. Tools Appl.* **2021**, *80*, 35973–35991. [CrossRef]
45. Deng, Y.; Yang, J.; Xu, S.; Chen, D.; Jia, Y.; Tong, X. Accurate 3D Face Reconstruction With Weakly-Supervised Learning: From Single Image to Image Set. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Long Beach, CA, USA, 16–17 June 2019; pp. 285–295. [CrossRef]
46. Blanz, V.; Vetter, T. A morphable model for the synthesis of 3D faces. In *Seminal Graphics Papers: Pushing the Boundaries*; Association for Computing Machinery: New York, NY, USA, 2023; Volume 2, pp. 157–164.
47. Guo, J.; Zhu, X.; Yang, Y.; Yang, F.; Lei, Z.; Li, S.Z. Towards Fast, Accurate and Stable 3D Dense Face Alignment. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Volume 12364 , pp. 152–168.
48. Zulqarnain Gilani, S.; Mian, A. Learning from Millions of 3D Scans for Large-Scale 3D Face Recognition. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1896–1905. [CrossRef]
49. Yu, Y.; Da, F.; Zhang, Z. Few-data guided learning upon end-to-end point cloud network for 3D face recognition. *Multimed. Tools Appl.* **2022**, *81*, 12795–12814. [CrossRef]
50. Liu, W.; Wen, Y.; Yu, Z.; Li, M.; Raj, B.; Song, L. Sphereface: Deep hypersphere embedding for face recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 212–220.
51. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
52. Huang, D.; Ardabilian, M.; Wang, Y.; Chen, L. 3-D Face Recognition Using eLBP-Based Facial Description and Local Feature Hybrid Matching. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 1551–1565. [CrossRef]
53. Liu, P.; Wang, Y.; Huang, D.; Zhang, Z.; Chen, L. Learning the Spherical Harmonic Features for 3-D Face Recognition. *IEEE Trans. Image Process.* **2013**, *22*, 914–925. [CrossRef]
54. Elaiwat, S.; Bennamoun, M.; Boussaid, F.; El-Sallam, A. A Curvelet-based approach for textured 3D face recognition. *Pattern Recognit.* **2015**, *48*, 1235–1246. [CrossRef]
55. Lei, Y.; Guo, Y.; Hayat, M.; Bennamoun, M.; Zhou, X. A Two-Phase Weighted Collaborative Representation for 3D partial face recognition with single sample. *Pattern Recognit.* **2016**, *52*, 218–237. [CrossRef]
56. Gilani, S.Z.; Mian, A.; Shafait, F.; Reid, I. Dense 3D Face Correspondence. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 1584–1598. [CrossRef]
57. Berretti, S.; Werghi, N.; Del Bimbo, A.; Pala, P. Matching 3D face scans using interest points and local histogram descriptors. *Comput. Graph.* **2013**, *37*, 509–525. [CrossRef]