Learning Streamed Attention Network from Descriptor Images for Cross-resolution 3D Face Recognition

JOÃO BAPTISTA CARDIA NETO, São Paulo State Technological College (FATEC), Brazil
CLAUDIO FERRARI, Dept. of Architecture and Engineering, University of Parma & University of Florence, Italy
APARECIDO NILCEU MARANA, RECOGNA Laboratory, São Paulo State University (UNESP), Brazil
STEFANO BERRETTI, MICC, University of Florence, Italy
ALBERTO DEL BIMBO, MICC, University of Florence, Italy

In this paper, we propose a hybrid framework for cross-resolution 3D face recognition which utilizes a Streamed Attention Network (SAN) that combines hand-crafted features with Convolutional Neural Networks (CNNs). It consists of two main stages: first, we process the depth images to extract low-level surface descriptors and derive the corresponding Descriptor Images (DIs), represented as 4-channel images. To build the DIs, we propose a variation of the 3D Local Binary Pattern (3DLBP) operator that encodes depth differences using a sigmoid function. Then, we design a CNN that learns from these DIs. The peculiarity of our solution consists in processing each channel of the input image separately, and fusing the contribution of each channel by means of both self- and cross-attention mechanisms. This strategy showed two main advantages over the direct application of Deep-CNN to depth images of the face; on the one hand, the DIs can reduce the diversity between high- and low-resolution data by encoding surface properties that are robust to resolution differences. On the other, it allows a better exploitation of the richer information provided by low-level features, resulting in improved recognition. We evaluated the proposed architecture in a challenging cross-dataset, cross-resolution scenario. To this aim, we first train the network on scanner-resolution 3D data. Next, we utilize the pre-trained network as feature extractor on low-resolution data, where the output of the last fully connected layer is used as face descriptor. Other than standard benchmarks, we also perform experiments on a newly collected dataset of paired high- and low-resolution 3D faces. We use the high-resolution data as gallery, while low-resolution faces are used as probe, allowing us to assess the real gap existing between these two types of data. Extensive experiments on low-resolution 3D face benchmarks show promising results with respect to state-of-the-art methods.

CCS Concepts: • **Computing methodologies → Computer vision**.

Additional Key Words and Phrases: 3D Face Recognition, Convolutional Neural Networks, Feature Descriptors, Self- and Cross-Attention

Authors' addresses: João Baptista Cardia Neto, joao.cardia@fatec.sp.gov.br, São Paulo State Technological College (FATEC), Rua Maranhão, 898, Catanduva, São Paulo, Brazil, 15800-020; Claudio Ferrari, Dept. of Architecture and Engineering, University of Parma & University of Florence, Florence, Italy; Aparecido Nilceu Marana, RECOGNA Laboratory, São Paulo State University (UNESP), Av. Eng. Luís Edmundo Carrijo Coube, 14-01, Bauru, Brazil, nilceu.marana@unesp.br; Stefano Berretti, MICC, University of Florence, Florence, Italy; Alberto Del Bimbo, MICC, University of Florence, Florence, Italy.

1551-6857/2022/1-ART1 $15.00
https://doi.org/10.1145/3527158

1 INTRODUCTION

Many works in the literature have demonstrated the potential of applying Deep Convolutional Neural Networks (DCNNs) for face recognition [Parkhi et al. 2015; Schroff et al. 2015; Yin and Liu 2018], which have reached astonishing performance even in extremely challenging scenarios. One tendency in recent face recognition works is to consider more difficult data in terms of resolution, quantity, and source [Al-Obaydy and Suandi 2020; Ferrari et al. 2018; Singh et al. 2018]. Extending the imaging modes to less conventional ones, including 3D data represented as depth images is a further research direction that aims at improving face recognition in specific scenarios [He et al. 2020; Xiong et al. 2019]. Several low-cost/low-resolution devices can capture RGB-D data (*e.g.*, Kinect camera) and, spite not being comparable with high-resolution 3D scanners, the depth maps they capture may provide additional cues that improve recognition in many cases. Compared to standard RGB imagery, performing recognition with 3D data can be advantageous because of its invariance to image nuisances such as illumination or pose changes. Another key aspect of using depth data is that it can be viewed as a sort of basic *feature map*, where each pixel encodes the distance from the camera, overall representing the 3D structure of the face. Thus, differently from RGB images, several other surface properties can be extracted from such data and used as additional channels when training the network. Some previous works exploited this intuition to enrich the data representation for different tasks by adding, for example, normal orientation maps [Gilani et al. 2017], or curvature maps [Galteri et al. 2019]. On the other hand, learning from this *augmented* data can be difficult. In fact, unlike natural RGB imagery, the information might be significantly uncorrelated from one channel to another. It is also possible that different surface properties or features have their discriminative information located at different image regions. Thus, processing the data as a whole with standard convolutional layers could lead to sub-optimal performance. Finally, the burdensome process of collecting a sufficient amount of good quality 3D faces to effectively train DNNs makes dealing with 3D data difficult, and resorting to low-cost sensors like Kinect seems the only viable solution in most practical scenarios. However, this poses an additional difficulty in dealing with the existing differences between high- and low-resolution 3D scans.

In this work, we address the problem of face recognition across 3D data of different resolutions. The founding idea of our solution is that of encoding hand-crafted, low-level features extracted from the depth data into color images, and train a CNN with them. First, given an input depth image a variation of the 3D Local Binary Pattern (3DLBP) [Huang et al. 2006] is computed on it, and encoded into a 4-channel image, referred to as *Descriptor Image* (DI). Then, a CNN is designed that learns on top of these DIs. Our novel architecture processes each channel separately to account for the different information encoded in each of them. In addition, self- and cross-attention mechanisms are applied to each stream so as to effectively fuse the information of the different streams. We have named our architecture as Streamed Attention Network (SAN) since it utilizes different streams of data as input coupled at an attention mechanism.

We will show that using low-level features as input significantly reduces the gap between high- and low-resolution 3D data, making it possible to train a CNN with high-resolution data and successfully apply it to low-resolution one.

In summary, the main contributions of this work are:

- We propose a hybrid face recognition solution that combines hand-crafted features and the SAN network architecture specifically designed to learn from the DIs. Our proposed solution aims at generating a representation that is more robust to resolution differences. Also, to the best of our knowledge, there are no previous works explicitly addressing the problem of cross-resolution 3D face recognition;
- We propose a novel solution to deal with uncorrelated input data. Our network architecture processes each input channel separately, and fuses the data streams using self- and cross-attention layers, so as to maximize the mutual information from each channel. In this way, we can benefit from a richer data representation;

- In a comprehensive experimental validation, we demonstrate that the proposed approach can effectively bridge the gap between high- and low-resolution 3D data. We show that the proposed DIs encode complementary information, and improve the performance upon that obtained with depth data.

The founding ideas of the method proposed in this paper have been introduced in our previous conference contribution [Cardia Neto et al. 2019]. We extended our contribution as follows: *(i)* we added a deepened discussion about the sigmoid encoding, which allowed us to better identify the optimal shape of the sigmoid; *(ii)* we significantly modified the network architecture by adding self- and cross-attention layers to better exploit the information carried by the DIs; *(iii)* we experimented using the proposed network as a feature extractor, so without any prior knowledge of the test data. This is a common setup in real-world scenarios, where the gallery of known subjects, and more in general the testing data, might not be available for learning; *(iv)* we also report a comprehensive evaluation that spans several datasets including low- and high-resolution data. In particular, the IIIT-D RGB-D and the MICC-3D face dataset have been added to evaluate our method on a large set of very low-resolution depth images. Overall, we propose and evaluate a novel network architecture that takes a combination of DIs and depth information as input. The network utilizes an attention mechanism to fuse and enhance the intermediate features, learned separately from each channel. In an ablation study, we investigate the effects of combining the input data, and the way the attention mechanism influences the performance of our approach. We also analyze the face recognition performance of our method in a cross-resolution scenario, and compare it with other well-established architectures from the literature.

The rest of the paper is organized as follows: In Section 2, we summarize related works in the literature; In Section 3, we describe the preprocessing operations applied to the depth images and introduce the computation of the Descriptor Images from the depth; The network architecture that operates on the DIs is proposed in Section 4; An extensive experimental evaluation on several datasets is reported in Section 5; Finally, discussion and conclusions are given in Section 6.

2 RELATED WORK

In this section, we summarize some works in the literature that focused on 3D face recognition and, more specifically, on low-resolution depth data. Given the lack of methods in the literature that explicitly deal with cross-resolution 3D face recognition, we review methods that either used *hand-crafted features* or *CNN architectures* for recognition.

3D face recognition based on hand-crafted features. Methods that follow this approach describe surfaces by specifically capturing geometric properties of the face [Berretti et al. 2010; Drira et al. 2013; Faltemier et al. 2008; Spreeuwers 2011], and most of them use high-resolution data acquired in controlled environments. Works that instead utilize Kinect-like devices exploit the temporal redundancy of frames or deformable models to increase the resolution and remove scanner-induced noise [Bondi et al. 2016; Drosou et al. 2013; Ferrari et al. 2021; Hernandez et al. 2012]. The main problem of those approaches is the increased demand for computational power, and the fact that a sequence or additional data is required to build the super-resolved models.

Few methods performed face recognition directly from low-resolution data. In the work by Min *et al.* [Min et al. 2012], a real-time 3D face identification system that receives a depth sequence as input is proposed. The face is detected and segmented utilizing a threshold on depth values. Next, the faces are reduced to common resolutions and the matching is obtained by registering a probe with several intermediate references in the gallery with the Expectation Maximization Iterative Closest Point (EM-ICP) algorithm. Despite interesting results were reported for this method, few subjects were included in the dataset used in the evaluation, and no comparison with other approaches was reported. Also, the method was evaluated only on low-resolution data, with no test provided on a cross-resolution scenario. In [Mantecón et al. 2016], Mantecon *et al.* proposed an algorithm for face recognition based on an image descriptor called bag of dense derivative depth patterns. Dense spatial derivatives were first computed and quantized in a face-adaptive fashion to encode the 3D local structure. Then, a multi-bag of words created a compact vector description from the quantized derivatives. One limitation of this approach is the size of the used descriptor, *i.e.*, a 98304-dimensional vector, whilst our learned feature has a size of 2048. In all the cases, differently from our proposal, the gallery set of the testing dataset is used for learning the features to train the classifier.

CNN-based 3D face recognition. Deep architectures that deal with 3D data had a slower expansion than the image-based counterpart, mainly because of the data representation problem. The wide variety of modalities that exist to represent 3D data (*e.g.*, point-clouds, triangular meshes) makes it difficult to work in the same standardized way without making significant modifications to the whole framework. Also, the lack of large-scale datasets for training contributed significantly to this problem. Given the above, a possible workaround to make use of existing DCNNs for 3D face recognition was proposed in the work by Kim *et al.* [Kim et al. 2017], where the authors utilized a pre-trained version of the VGG-Face network and fine-tuned it for depth data. To train such a deep architecture, a very large amount of data is needed, and several datasets were joined together. To further increase the amount of data for training, synthetic expressions and occlusions were generated. Still, the approach struggles when tested on low-resolution data. To address the need for large amounts of data, Gilani *et al.* [Gilani and Mian 2018] proposed a synthetic data generation technique that they used to build a dataset of ≈3M scans. Such data was utilized to train a deep architecture consisting of 13 convolutional layers, 3 fully connected layers, and a softmax layer. Among other observations, authors concluded that because of the smooth nature of the face surface, there is the need for larger kernels for the convolutional filters. A hybrid solution exploiting both the RGB and depth information was presented in [Jiang et al. 2019]. In that solution, a CNN was trained guided by the supervision of an additional loss, called "attribute-aware" loss, that attempts to cluster the face images based on attribute information such as gender or age. Along these lines of investigations, Mu *et al.* [Mu et al. 2019] employed a lightweight CNN equipped with a multi-scale feature fusion layer to fill in the gap between high- and low-resolution depth scans. However, all the aforementioned methods do not really investigate a cross-resolution scenario, since the gallery set of the testing dataset is always used for training a classifier. Differently, we explicitly separate the two and train the network on high-resolution data and perform tests on low-resolution without any further learning process on such data. Overall, these design and evaluation aspects have no parallel in the existing literature on 3D face recognition.

3 BUILDING THE DESCRIPTOR IMAGES

Training a DCNN from scratch on depth data is difficult due to the large quantity of acquisition noise, especially with low-resolution, and the difficulty to acquire large volumes of labeled data due to the limited operating range of such devices (*e.g.*, few meters for Kinect-like depth cameras). Since, in the case of depth, the Web is not a viable source of additional instances the most practiced workaround in the literature is that of taking a DCNN pre-trained on RGB data and fine-tune it with a small set of depth images.

We propose a different approach, where the learning tools are applied on top of intermediate images generated from the original data by applying a low-level feature extractor. In this work, we use the 3DLBP [Huang et al. 2006] feature because of its computational efficiency and effectiveness in describing depth images of the face [Cardia Neto and Marana 2018]. The 3DLBP definition and its modified version used in this work are introduced in Section 3.2. Before computing 3DLBP, depth images are enhanced using the operations illustrated below in Section 3.1.

3.1 Depth image pre-processing

To diminish noise effects, a pre-processing pipeline has been used. This pipeline is the same as proposed in [Cardia Neto and Marana 2014] for the case of data acquired with the Kinect v1 camera. In the case of high-resolution

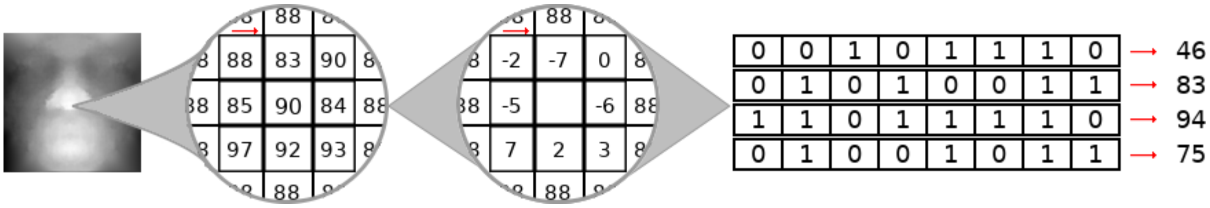


Fig. 1. 3DLBP computation on a depth image of the face. A 3 × 3 neighborhood region is shown; First, the difference between the neighborhood and the central pixel is computed, clockwise, starting from the top-left pixel; Then, each column of four bits encodes the difference value using the first bit for the sign (0 for negative, 1 for zero or positive values), and the three subsequent bits for the absolute value.

data, the pipeline is applied without the symmetric filling step. The pre-processing pipeline has three main steps: *(i)* Segmentation, *(ii)* symmetric filling, and *(iii)* generation of new, *i.e.*, pre-processed, depth maps. Each step of this pipeline is detailed in the following. We first segment a circle with radius $r = 70mm$ centered at the nose-tip and mirror the original face. A point from the original face is selected and its nearest neighbor is found on the mirrored face. If the Euclidean distance between the point from the mirrored face and the point from the original face is greater than a threshold δ, the mirrored point is added to the original face. In this work, we set $\delta = 0.5$. This was originally proposed in [Li et al. 2013]. After the whole procedure, for each face we employ the Iterative Closest Point (ICP) algorithm [Besl and McKay 1992] to perform fine fitting between the set of points. This is needed to deal with small pose rotations. Finally, new depth maps are generated using a ridge estimator[1], to derive a surface on a 2D grid starting from a sparse set of points. We use the depth maps generated by applying the above steps for CNN training or to compute the 3DLBP operator or its sigmoid version.

3.2 3DLBP

The 3DLBP is a variation of the traditional LBP proposed by Ojala *et al.* [Ojala et al. 1996]. Its computation starts in a 3×3 region defined around a center pixel, or more generally a region with radius R in which P points are sampled. The depth value of the central pixel is subtracted from its neighbors and each of those values is truncated in the range $[-7, +7]$. This is motivated by the fact the face is a smooth surface and most of those differences fall in that range [Huang et al. 2006].

With the $[-7, +7]$ range, 15 different values are encoded, which results in a four-bit representation. Each bit is regarded as a separate channel: the first channel encodes the sign of the difference, *i.e.*, 0 if the difference is negative, 1 otherwise; The other channels encode the absolute value of the difference transformed in a binary code of three bits. Figure 1 shows the generation of a 3DLBP descriptor in a 3×3 region. Each one of the four bits of the 3DLBP is regarded as a separate channel of the final *Descriptor Image* (DI), forming an image with four channels. We used an RGBA image to this end.

In the DI, each channel behaves differently. The first one encodes the sign of the difference and describes if the local neighborhood is increasing or decreasing with respect to the central point (*e.g.*, a local minimum would be encoded as 255, that is to say, all the bits are 1). The last three channels encode the absolute depth difference between each center point and its neighbors. The first channel encodes the sign of the difference, and changes in its values appear to occur smoothly. This happens mainly because faces are smooth surfaces and, locally, shifts in values do not occur abruptly. The second channel receives the encoding of the most significant bit of the absolute depth difference. Thus, values of this channel are 1 for differences bigger or equal to 4. This does not happen so frequently on the face because of its smooth surface, but it can occur in the nose and periocular region. The last

[1] https://www.mathworks.com/matlabcentral/fileexchange/8998-surface-fitting-using-gridfit

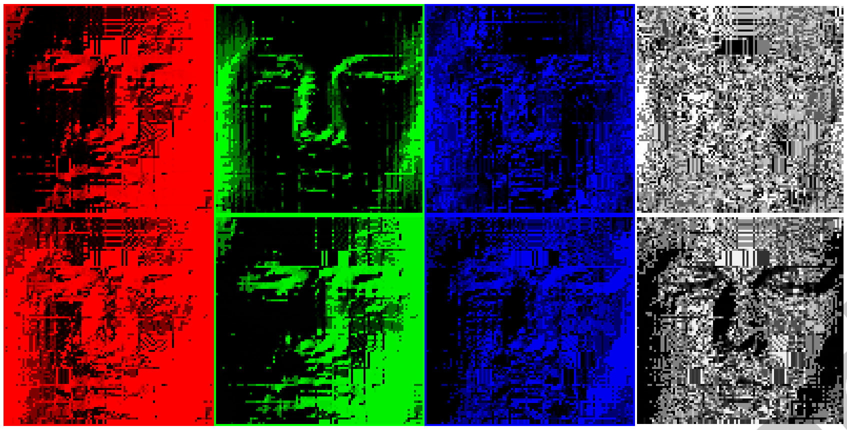


Fig. 2. The channels of a DI are shown separately for the 3DLBP DI (top) and the Sigmoid DI (bottom). The red, green, and blue channels are shown in the first three columns, respectively, followed by the alpha channel in the fourth column.

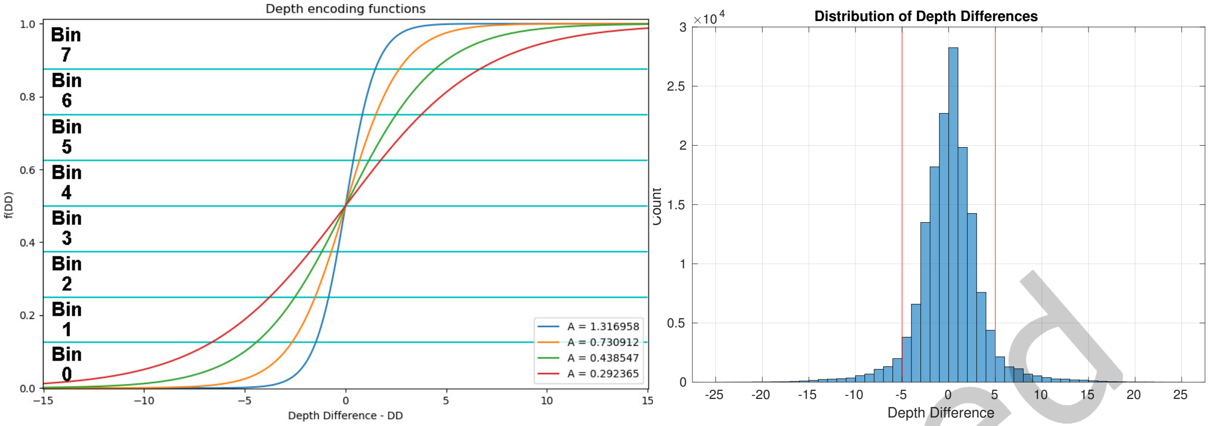
two channels, instead, are noisier because they encode less significant bits, so that changes occur more frequently. The third channel (second bit) changes for differences of two, while the fourth one (least significant bit) changes for depth differences of one. This generates high frequency information. Figure 2, top row, shows an example of the four channels resulting from the 3DLBP encoding.

3.3 Sigmoid encoding

A possible limitation of the standard 3DLBP approach is that, within the $[-7, +7]$ interval, negative or positive difference values share the same binary code, except for the sign bit. This implies that some regions of the resulting DI might have the same values on three out of four channels, ultimately resulting in redundant and noisy information. Furthermore, the range of depth differences needs to be kept fixed to fit the 4-bits representation. One way to account for these limitations is to incorporate a sigmoid function in the computation of the 3DLBP operator. In this case, instead of truncating the exceeding values in the $[-7, +7]$ range, the sigmoid can be utilized to map an arbitrary interval $[-\delta, +\delta]$ to four bits. To control the range of the encoded values, a parameter A is used, leading to the following:

$$f(x) = \frac{1}{1 + \exp(-Ax)} \, , \tag{1}$$

where x is the depth difference between a point in the neighborhood and its center, and A is a scalar value that stretches the sigmoid function. To encode the sigmoid values, eight bins are defined in the interval between 0 and 1. Then, each $f(x)$ is mapped to its closest bin, in a histogram-like fashion. Note that, even though the sign channel is used to build the four-channel image, $f(x)$ is computed considering the depth difference along with the sign, so that same values with opposite sign are put into different bins. Figure 3 (a) shows the function encoding for different values of A. This has the advantage of letting us choose the proper range of depth differences that are encoded into each bin. Indeed, from Figure 3 (a), it is possible to observe that the larger the range, the coarser the encoding resolution. So, it is fundamental to find the right balance between these parameters. To choose a proper threshold to truncate the exceeding values, we estimated the distribution of depth differences from a sample set of 3D scans. Figure 3 (b) shows that the majority of differences fall in the range $[-4, +4]$. We thus chose the value of A to truncate values exceeding ±4, that is $A = 1.31$. Among the possible choices of A, we empirically found



(a) Sigmoid encoding for different values of A in Eq. (1)

(b) Distribution of 3DLBP differences

Fig. 3. In (a), the sigmoid functions utilized in the encoding of the depth differences and obtained for different values of the A parameter in Eq. (1) are reported. In (b), the distribution of the differences obtained in the 3DLBP computation from a set of samples from the FRGC dataset is reported.

that this is the best performing one, which is consistent with the depth differences distribution. The smaller range with respect to the standard 3DLBP, also allows for a finer encoding of each depth difference value.

3.4 Correlation between channels in the DIs

One main idea in our approach is that of encoding 3D information in 2D images, where each channel encodes local differences between depth values rather than photometric information like in *standard* RGB images. In such photometric RGB images, the color information carried by the three channels is heavily correlated, allowing us to process them through convolutions.

However, we expect this correlation to be less significant for the DIs due to the different nature of the information encoded in the input channels. Similar investigations were performed in the Computer Vision literature for several different tasks [Abbass et al. 2021; Parchami et al. 2017; Wang et al. 2020a, 2021]. We verified this aspect by measuring the correlation between channels in the DIs. In Figure 4, it is possible to note the difference between the correlation of the red layer and the other layers in the Sigmoid and 3DLBP DIs, compared to photometric RGB images. As expected, the channels in the DIs show almost no correlation.

CNN architectures normally rely on correlation between channels in the input images to learn meaningful features through hierarchical sets of convolutions with shared weights. Given the observations above, processing the DIs as a whole could lead to sub-optimal results. As explained in detail in Section 4, our idea is that of processing each input channel separately through dedicated branches.

4 NETWORK ARCHITECTURE

Before going into the details of our proposed architecture, we first motivate the proposed design, and refer to the process of DIs generation as described in Section 3.2. From Figure 2, it turns out clearly that each channel encodes information of different granularity. On the one hand, this has the advantage of providing richer information. On the other hand, it gives rise to some issues: depending on the channel, the important information could be located in different image regions. Moreover, as shown in Figure 2, the last channels, *i.e.*, least significant bits,

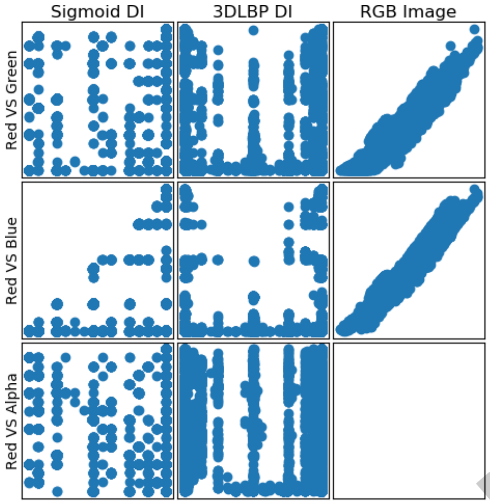


Fig. 4. Correlation between the red and the other channels in the DIs is shown. The plot is obtained by reporting the value from a position in the red channel against the value in the same position of another channel. This is performed for the DIs obtained with Sigmoid, and 3DLBP, and for the RGB image corresponding to the same depth image from which the DIs were computed.

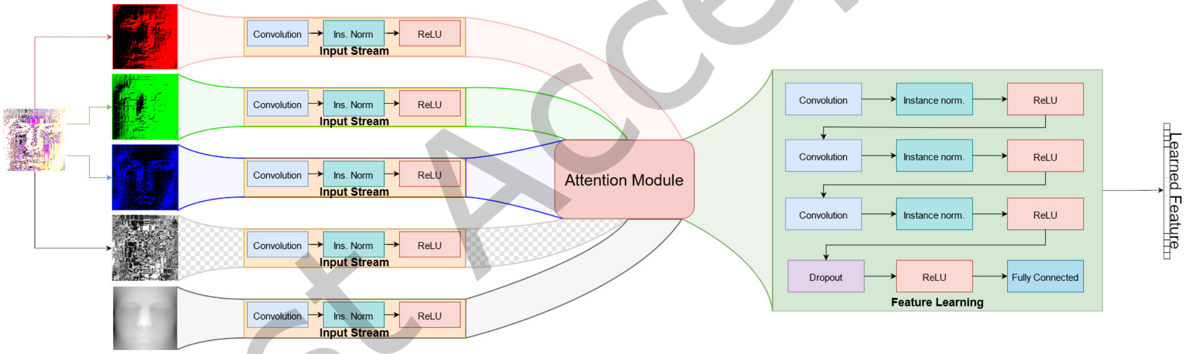


Fig. 5. Our proposed network architecture. The input is divided into five streams, *i.e.*, four for the DI and one for the given depth image. First, each stream is processed through a block with different kernel and padding sizes; Then, self- and cross-attention are used to enhance the features; Finally, the enhanced features are concatenated and processed by a block that learns the resulting feature representation.

contain higher-frequency information with respect to the first ones. Processing such different data with shared sets of convolutions with fixed kernel size could lead to loss of information. In addition, differently from RGB images, the DIs are characterized by a low cross-channel correlation. Hence, processing all the channels together could result in sub-optimal representations that are learned by the network.

Given the above observations, we designed a multi-branch network architecture, in which each input channel is processed by separate sets of convolutional layers. The overall architecture is shown in Figure 5. Initially, the channels of the DI and the pre-processed depth go separately through different *input stream* blocks. These blocks are composed of a convolutional layer, an instance normalization layer, and a ReLU. Each stream outputs 64 feature maps, but the kernel and padding have some variations in order to adapt to the specific channel. The stream that receives the red channel has a kernel size of 5 × 5 and a padding of 2, the green stream has a kernel of 4 × 4 and a padding of 1, the blue stream has a kernel of 3 × 3 and a padding of 1, the alpha stream has a kernel of 2 × 2 and padding of 1. Finally, following [Gilani et al. 2017], we set the kernel size for the depth stream as 5 × 5, with padding of 2.

The outputs of the above streams need to be fused together for further processing. The simplest choice consists in concatenating the feature maps resulting from the input streams, and feeding them to an additional module, so to obtain a compact descriptor. We refer to this module as *Feature Learning*, as shown in Figure 5 (right). This is composed of three convolutional layers outputting 128, 256, 256 feature maps, with kernel size of 5, 3, 3, respectively. Each convolution is followed by instance normalization and ReLu activation. Finally, a dropout layer and a fully connected layer of size 2048 are stacked to obtain the face descriptor.

We observe though that processing the input channels independently, then concatenating the outputs could still lead to inconsistency in the data and so difficulties in processing the concatenated features. We find a possibly effective solution to address this issue is using an attention mechanism. So, before concatenating the feature maps, an *Attention Module* enhances the features by means of self- and cross-attention layers, before sending them to the feature learning module. The structure of this part is illustrated in Figure 6 and described in detail in Section 4.1.

To summarize, our proposed framework, that we refer to as *Streamed Attention Network* (SAN) is composed of three main modules: *(i)* the input streams that process each input channel independently; *(ii)* the attention module that fuses the feature maps together, and *(iii)* a feature learning module that finally produces the face descriptor. The code of our approach is available at https://github.com/jbcnrlz/san.

4.1 Self- and Cross-Attention Mechanism

Attention mechanisms as originally proposed in [Vaswani et al. 2017] have been recently utilized in a variety of tasks, including face recognition among them [Liao et al. 2020; Wang et al. 2020b]. Such interest can be explained by the fact that attention maps highlight important and discriminative facial regions within the image [Wang et al. 2020b], and several works started to include such mechanism in their framework. An attention function is defined as a mapping between an input composed of query and a set composed of key and value, to an output [Vaswani et al. 2017]. One possible way to implement an attention mechanism is to utilize the concept of self-attention, which builds a representation for a sequence combining different weighted positions from it [Vaswani et al. 2017]. For example, in [Liao et al. 2020] a generative adversarial architecture was used to generate realistic frontal face images from faces with different poses. This generative adversarial architecture has an attention mechanism integrated into it, being named Attention Selective Network (ASN). The role of attention, in the original work, is to increase the quality of the segmentation of the face from the background. In [Wang et al. 2020b], a Hierarchical Pyramid Diverse Attention (HPDA) is used to account for hierarchical multi-scale local features in the face recognition process. With this approach, it is possible to learn a multi-scale diverse local representation in an automatic and adaptive way, taking into consideration the diversity in the data. In the end, the authors used information from several levels on the proposed model.

From the aforementioned works, since different face areas might impact differently on recognition, it appears appropriate to apply this approach for such a task. We believe such a mechanism can be useful also in our context, making our architecture focus on relevant information from the input channels. As a result, the learned features would include richer information. This hypothesis derives from how hand-crafted features are built. Both the DIs we examined describe low-level shape features that should help in maintaining the identity information despite other nuisances such as resolution differences. Nonetheless, given that the input of our network lacks correlation, we believe that processing the streams separately and learning how to fuse the information from the

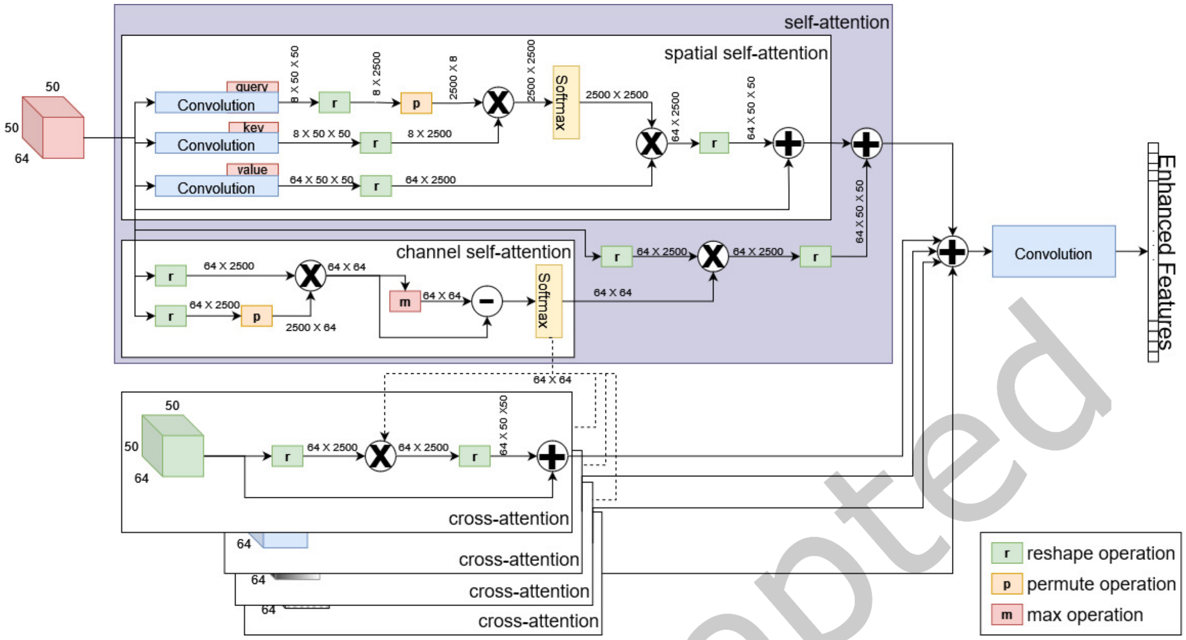


Fig. 6. The attention module used in our approach, inspired by [Yu et al. 2020]. For each data stream, the attention module combines the self-attention maps with cross-attention from the other data streams. The streams are the red, green, blue, and alpha channels from the DI, and the pre-processed depth data.

streams in a collaborative way, would lead to even more robust features. To this aim, we further identified the cross-attention mechanism proposed by [Yu et al. 2020] as a way to increase the collaboration from different data in the feature learning process. It was designed to be used into a Siamese architecture, more specifically to combine inter-dependencies in an image between context and object template for visual object tracking. The attention module proposed by [Yu et al. 2020], which combines self- and cross-attention, inspired the attention module utilized in this work. Differently from the original one, we do not employ deformable convolutions, and we combine several cross-attention maps with one self-attention map to enhance the learned features. For each stream, this combination is performed by summing its self-attention map with the cross-attention from the other streams.

As proposed in the original work, the self-attention is the combination of spatial self-attention and channel self-attention. A block diagram with our attention mechanism is shown in Figure 6.

Given the inputs $S_r, S_g, S_b, S_a, S_d \in \mathbb{R}^{C\times H\times W}$, with S_r being the red stream, S_g the green stream, S_b the blue stream, S_a the alpha stream, S_d the depth stream, C the channels, H the height, and W the width for the feature maps, the idea is to enhance each input stream with its self- and cross-attention. To this end, we initially calculate the self-attention for one of those inputs, and the cross-attention for the others.

For the self-attention, the spatial self-attention and channel self-attention are computed. To generate the Query, Q, and Set, S, we first reduce the input feature maps using 1×1 convolution. This operation results in $Q, S \in \mathbb{R}^{C'\times H\times W}$, with $C' = \frac{C}{8}$. Next, both are flattened in the width and height dimensions, resulting in $\overline{Q}, \overline{S} \in \mathbb{R}^{C'\times N}$ with $N = H \times W$. To generate the spatial self-attention A_s^s, we compute the softmax with Q and S as follows:

$$A_s^s = softmax(\overline{Q}^T\overline{S}) \in \mathbb{R}^{N\times N} \; . \tag{2}$$

For the Value V, the input stream goes trough a 1 × 1 convolution without reducing the number of channels so that $V \in \mathbb{R}^{C\times H\times W}$. Similarly to query and key features, V is flattened in the height and width dimensions, resulting in $\overline{V} \in \mathbb{R}^{C\times N}$. After this, the spatial self-attention X_s^s feature can be computed as:

$$X_s^s = \alpha\overline{V}A_s^s + I \in \mathbb{R}^{C\times N} \ , \tag{3}$$

with I being the input to the self-attention and α a scalar parameter.

Once this operation is computed, X_s^s is reshaped back to its original shape. After computing the spatial self-attention, the channel self-attention X_x^s is evaluated in a similar manner. Initially, the input I is reshaped to $\overline{I} \in \mathbb{R}^{C\times N}$, then A_c^s is computed as:

$$A_c^s = \overline{I}\overline{I}^T \in \mathbb{R}^{C\times C} \ . \tag{4}$$

Finally, X_c^s is calculated with the column softmax of the difference between the maximum value of A_c^s and it original values:

$$X_c^s = softmax(A_c^s - \max(A_c^s)) \in \mathbb{R}^{C\times C} \ . \tag{5}$$

Once X_c^s is calculated, we perform matrix multiplication with $\overline{I}$ and reshape the result to the original shape of the input. With both X_c^s and X_s^s being calculated and having the same shape, it is possible to define the value of X^{sa} as the output of the self-attention $X^{sa} = X_s^s + X_c^s$.

For the cross-attention part of the approach, our idea is to calculate it on different inputs than those used for the self-attention part (*e.g.*, while the S_r is processed for the self-attention, the inputs $S_g, S_b, S_a, andS_d$ are processed for the cross-attention). In the cross-attention, the input I_c is reshaped, resulting in $\overline{I_c} \in \mathbb{R}^{C\times N}$. The reshaped input goes trough a matrix multiplication, adding information from the channel self-attention data, as given by:

$$A^c = X_c^s I_c \in \mathbb{R}^{C\times N} \ . \tag{6}$$

The variable A^c is reshaped to the same shape as I_c. Finally, the output from the cross-attention module is given by:

$$X^{ca} = \gamma A^c + I_c \in \mathbb{R}^{C\times H\times W} \ , \tag{7}$$

with γ being a scalar parameter.

The self- and cross-attention are then combined by summing the outputs from each module. After this, the data is utilized as an input to a convolutional layer with an output size of 64 and kernel of 3. Our enhanced feature is the output from this layer.

As said previously, all the channels from the DIs (red, green, blue, and alpha) are used as inputs to the attention module, plus the pre-processed depth. Every input to the attention module is enhanced with the self-attention and cross-attention: for example, while the self-attention for the red channel is extracted, the cross-attention for the other input data (green, blue, alpha, and depth) is also computed. These result of the self- and cross-attention are then used in the feature enhancing process.

In Figure 7 the input and output from the attention module are displayed. The heatmaps show that the attention for the depth data mostly focuses on the nose and mouth regions, whilst the other inputs distribute the attention across the whole face, complementing each other. The high granularity of the DIs describes finer details of flat face regions that are not captured by the depth maps, thus carrying richer information. The attention module helps to capture and maximize such complementary information, which we will show improve the recognition performance.

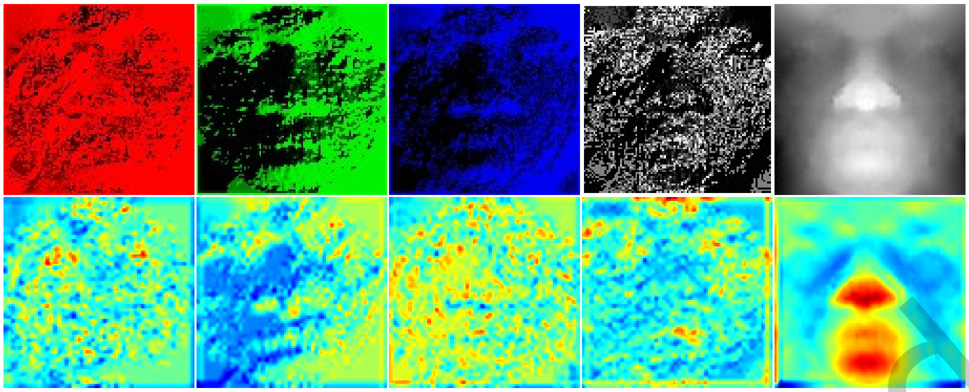


Fig. 7. Class Activation Maps (CAM) based on [Zhou et al. 2016] from the output of the attention module. The first line shows the input data for the attention module, while the second line illustrates the resulting output with both the self- and cross-attention. For generating the maps we do a matrix multiplication with the weights from the last convolutional layer from the attention module with the enhanced features, this will generate an image with the shape of $C \times C$, with C being the size of feature maps from the enhance features (on our case, 64). This is then reshaped to the same size as the input stream.

5 EXPERIMENTAL RESULTS

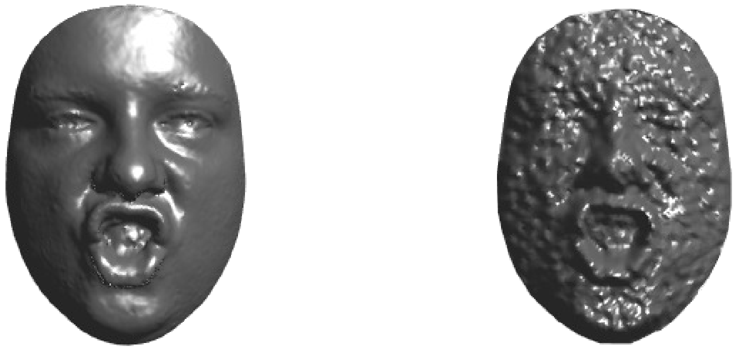
In the following, we report the results of a comprehensive evaluation of our approach. To this end, in Section 5.1, we first present the five face datasets we have used in the experiments, two for training and the remaining three for test. Then, we report details about the training of our network architecture in Section 5.2. The evaluation is performed by initially reporting on an ablation study aiming to investigate the contribution of the different parts of the proposed architecture to the final accuracy and also to investigate the effect of different design choices and parameters (Section 5.3). Results on the three test datasets also in comparison to baseline solutions and state-of-the-art methods in the literature are finally reported and discussed in Section 5.4.

5.1 Datasets

In this work, we used the following face datasets: *(i)* The FRGCv2.0 [Phillips et al. 2005] dataset, *(ii)* the Bosphorus [Savran et al. 2008] 3D face database, the *(iii)* EURECOM Kinect face dataset [Min et al. 2014], the *(iv)* IIIT-D RGB-D face database [Goswami et al. 2013], and *(v)* the newly collected MICC-HR/LR 3D face dataset [Ferrari et al. 2022]. The high-resolution FRGC and Bosphorus datasets were used to train our network from scratch, while tests were conducted on the others.

FRGC – The FRGC dataset includes 4,007 high-resolution scans of 466 different individuals acquired in two separated sessions. About 60% of the scans have neutral expression, while the rest show slight spontaneous expressions.

Bosphorus – The Bosphorus 3D face database comprises 4, 666 high-resolution scans of 105 individuals; There are up to 54 scans per subject, which include expression variations, facial action units activation, rotations and occlusions. **EURECOM** – The EURECOM Kinect face dataset collects RGB-D images of 52 subjects acquired with a Kinect sensor in two separate sessions (*Session* 1 and *Session* 2) with 7 variations each: neutral, smile, illumination, paper occlusion, mouth occlusion, eyes occlusion and open mouth. This dataset is employed for evaluating our approach with low-resolution data with three different protocols, as defined in [Min et al. 2014]. For each protocol, the gallery is composed of all the seven variations listed above from *Session* 1, while the probe set varies as follows:



(a) High-resolution

(b) Low-resolution

Fig. 8. Examples of a high- and a low-resolution face of the same subject from the MICC-HR/LR 3D face dataset.

(i) Probe composed of all the seven variations from *Session* 2;
(ii) Probe composed of three variations (neutral, smile, illumination) from *Session* 2;
(iii) Probe composed of one variation (neutral) from *Session* 2.

IIIT-D RGB-D – This dataset includes 106 subjects with 4, 605 RGB-D images comprising neutral as well as spontaneously expressive faces. For data acquisition, the Kinect V1 was used.

MICC-HR/LR 3D face dataset – For the specific purpose of face recognition across scans with different resolution, we collected a dataset of paired high- and low-resolution scans of 11 people performing 18 complex and non-standard expressions, plus the neutral pose. The collection of this data was motivated by the lack of publicly available datasets explicitly including both high- and low-resolution scans of the same subject. The dataset has been collected during Summer 2019. The participants were students and staff members with age ranging from 20 to 50, 10 men and 1 woman. To collect aligned pairs of high- and low-resolution 3D scans, a KinectV2 sensor was placed in front of a high-resolution 3dMD scanner. The 3dMD scanner provides very accurate meshes with an average RMS reconstruction error of about 0.2mm or better, depending on the exact pre-calibration and configuration of the device. The scans have an average of about 40,000 vertices and 80,000 facets. The Kinect sequences start from a neutral expression and reach a peak expression. The high-resolution scan is captured at the peak expression. In this work, we use only the peak depth frame from the Kinect sequence. This is intended to validate our network in the worst case scenario, without considering depth aggregation for increasing the depth resolution. We did not consider the RGB video as we focus specifically on 3D recognition. An example of the paired high- and-low resolution face scans is shown in Figure 8.

To perform recognition, we defined some specific protocols. The gallery set is always composed of the high-resolution scans, while the Kinect depth frames compose the probe set. Given that one neutral plus 18 expressions are included in the data of each subject, we define three different experiments:

(1) Probe composed of the neutral sequence only;
(2) Probe composed of the 18 expressive sequences;
(3) Probe composed of all the 19 sequences.

The gallery, instead, can be either composed of the neutral scan only, or all the 19 scans. This provides a total of 6 different protocols. In the following, we will refer to each protocol as *Gallery*-vs-*Probe*, where both can be Neutral (*N*), Non-Neutral (*NN*), and All (*A*). We will report identification results in terms of rank-1 recognition.

Table 1. EURECOM dataset: rank-1 results using depth data, and the combination between 3DLBP DI and Sigmoid DI. The gallery is from Session 1, while the probe set is composed of: *(i)* Session 2, *(ii)* Session 2 without occlusions, *(iii)* neutral scans from Session 2.

Data Type	*(i)* Session 2	*(ii)* Session 2 w/o occlusion	*(iii)* Session 2 neutral
Pre-processed depth	66.20%	74.51%	73.07%
3DLBP DI	63.19%	73.08%	73.08%
Sigmoid DI	64.56%	73.08%	80.77%
3DLBP DI + Depth	67.31%	76.92%	75.00%
Sigmoid DI + Depth	**72.25**%	**80.29**%	**84.62**%

5.2 Training Details

For training our network, we used together the face scans in the FRGC and Bosphorus datasets. The network was trained from scratch for 200 epochs. We augmented the training set by applying a 3D rotation around the Y (pitch angle) and X (yaw angle) axis from -30 to $+30$ degrees to each training face. The final number of training images was 98, 173, of 571 individuals. The images were normalized utilizing their average and standard deviation. We used the Adam optimizer with a learning rate of 0.0005, and for the moment estimates (β_1 and β_2) we utilized the exponential decay rates of 0.9 and 0.999, respectively. Our batch size was 25.

5.3 Ablation study

An ablation study was carried out to determine the best type of data to be used as input and so validate our proposed Sigmoid DI. To this end, we used the pre-processed depth, the DI originated from the original 3DLBP operator, our proposed sigmoid encoding, and combinations of DIs plus pre-processed depth. The above experiments were performed on the EURECOM dataset. For evaluation, we used our complete architecture comprising self-and-cross attention, except when using depth data only as input. In this latter case, there are no extra channels with which compute the cross-attention. So, only self-attention is applied.

Table 1 shows the results obtained with different input data. The training was carried out as described in Section 5.2, and the matching was performed by computing the cosine similarity between a probe and all the faces in the gallery. The probe is finally associated with the identity of the most similar face in the gallery. Results are reported in terms of rank-1 recognition rate. Looking at the table, it is possible to see that the Sigmoid DI plus the pre-processed depth outperforms all the other data, with a performance increase of about 5% from the second-best performing input. One key aspect to notice is that, while performance are comparable when separately using the depth or the Sigmoid DI, combining them results in a noticeable accuracy improvement. This evidences the potentiality of 3D data over standard RGB imagery, which can be exploited to extract richer information. In what follows though, we will also be showing that, in order to effectively exploit this additional information, a careful network design is required.

5.4 Results

In this section, we report results obtained on the EURECOM Kinect face dataset, the IIIT-D RGB-D face database, and the MICC-HR/LR 3D face dataset, also in comparison to other methods in the literature. Given the very few approaches that reported results using these datasets with a comparable protocol, *i.e.*, cross-database and cross-resolution, we mainly compare our proposed architecture against other state-of-the-art network architectures. The results reported for the ResNet50 [He et al. 2015] and MobileNet V3 [Howard et al. 2019] architectures were obtained by training the models from scratch with the Sigmoid plus depth data using the same parameters and augmentation as for our approach. For the VGG approach, we utilized the publicly available code in [Kim

Table 2. EURECOM dataset: Rank-1 results. The gallery is from Session 1, while the probe set is composed of: *(i)* Session 2, *(ii)* the three variants without occlusions from Session 2, *(iii)* neutral scans from Session 2. The results marked with * are from the network published in [Cardia Neto et al. 2019] utilized as feature extractor. PPD stands for pre-processed depth.

Method	*(i)* Session 2	*(ii)* Session 2 w/o occlusion	*(iii)* Session 2 neutral
Sigmoid*	55.49%	59.13%	63.46%
3DLBP*	57.41%	65.38%	69.23%
Fusion*	58.52%	62.98%	59.62%
VGG [Kim et al. 2017]	13.9%	14.8%	13.5%
ResNet50	58.52%	66.35%	69.23%
Mobilenet v3	59.34%	74.04%	78.85%
SAN - No attention + Sigmoid + PPD	69.78%	78.37%	80.77%
SAN - Self-attention + Sigmoid + PPD	**73.63%**	81.25%	80.77%
SAN - Cross-attention + Sigmoid + PPD	73.35%	**82.21%**	**86.54%**
SAN - Self- and cross-attention + Sigmoid + PPD	72.25%	80.29%	84.62%

et al. 2017] to process the depth images, and pre-trained weights. We also considered the network architecture described in [Cardia Neto et al. 2019], using it as feature extractor instead of fine-tuning the classifier on the test dataset as done in [Cardia Neto et al. 2019].

EURECOM dataset. Table 2 shows the results of the experiments performed with the EURECOM dataset, where the matching is performed between low- and low-resolution data. Remembering that the training is carried out only with high-resolution data, the fact that our approach is the best performing among several different architectures suggests us that the learned features maintained discriminative information, which is also robust to changes in resolution. The considerable gap between our results and those obtained with the VGG model [Kim et al. 2017], which only uses depth as input is a further evidence of the resolution robustness obtained with our solution, and how the DIs contribute to it. The improved accuracy with respect to other state-of-the-art architectures, instead, demonstrates the advantage of our architectural choices. On the opposite, the drop in performance when occluded faces are also considered (experiment *(i)*) evidences some lacks of our method in this respect. In the EURECOM dataset, the occlusions are originated by subjects putting a hand or a sheet of paper in front of the face, making the face surface flat on those regions. Given the fact that our DIs describe low-level shape information in terms of depth differences, it is possible to assume that those flat regions affect the performance of our method. However, this limitation is related to the operator rather than to the network architecture itself. Another important aspect to consider is that the best performing approaches use the proposed attention mechanism. The difference between the best performing method with any attention, and the one without attention can vary from 4% to 6% depending on the protocol. The results point towards the importance of utilizing any type of attention mechanism, even if it does not combine self- and cross-attention.

IIIT-D dataset. To evaluate our results in a less constrained setting, we have performed experiments on the IIIT-D dataset. To this end, we followed the protocol described in [Goswami et al. 2013]. In particular, this protocol used a 5-fold cross validation, where each fold contains 464 faces from the 106 subjects and the results from the Rank-5 accuracy is utilized. The validation is instead performed on 4,181 face images from different subjects. We underline that the depth images contained in this dataset are highly noisy, and captured at a very low-resolution. For this reason, most works employing this dataset make use of both RGB and depth data. A very recent work also analyzed this fact, reporting a detailed discussion regarding the difficulty of using such data [Hu et al. 2019]. In this context, results show that our proposed approach still performs effectively, even using only very low-resolution depth information. Table 3 reports the results of our method and for the same approaches we used in the comparison on the EURECOM dataset, plus the method in [Hu et al. 2019]. Our approach with sigmoid

Table 3. IIIT-D face dataset: rank-5 accuracy. The results marked with * are from the network published in [Cardia Neto et al. 2019] utilized as feature extractor. PPD stands for pre-processed depth.

Method	Rank-5 Accuracy
Sigmoid*	25.02% ± 0.01
3DLBP*	32.83% ± 0.01
Fusion*	37.44% ± 0.01
VGG [Kim et al. 2017]	58.29% ± 0.009
ResNet50	53.05% ± 0.006
Mobilenet v3	31.75% ± 0.01
Hu, Zhao, and Liu [Hu et al. 2019]	26.8%
SAN - No-attention + Sigmoid + PPD	43.77% ± 0.01
SAN - Self-attention + Sigmoid + PPD	45.82% ± 0.006
SAN - Cross-attention + Sigmoid + PPD	45.22% ± 0.01
SAN - Self- and cross-attention + Sigmoid + PPD	**59.31% ± 0.01**

+ pre-processed depth (PPD in the Table) still outperformed all the other works we are comparing with. Since the data in this dataset is very noisy, the process of encoding at a coarser-resolution but with a larger range of variations, which is what the Sigmoid encoding process does, leads to a significant robustness. In this experiment, it is also possible to observe how joining self- and cross-attention significantly increases the performance for our approach. The performance of the full attention mechanism proposed is about 15% better than utilizing only one of the parts of the mechanism (either self- or cross-attention). Given the quality of data, it is reasonable that using more information from different aspects of the attention mechanism helps the method in reducing the negative impact of noise.

MICC-HR/LR 3D face dataset. Results obtained on the MICC-HR/LR 3D face dataset are summarized in Table 4. In these experiments. the network is trained with high-resolution data, and the matching is performed between high- and low-resolution. This resembles a real-world scenario, where the gallery is collected in a controlled setting, while the probe comes from surveillance devices. Our approach has shown to be more effective than other works in comparison, making it suitable for real applications, where a certain degree of robustness to noisy conditions is required.

Finally, we note that both aspects, *i.e.*, architecture and data, are important. The VGG uses only depth data, whilst the ResNet and the Mobilenet v3 were trained with the combination of Sigmoid DI + pre-processed depth. There are some cases where the former is more accurate (see Table 3), while others where using the DIs result significantly better (see Table 2 and Table 4). Overall, our solution comprising the DIs and a dedicated architecture, reports consistent results across all the datasets, which span a variety of challenging conditions, including very noisy data, occlusions, expressions and resolution differences. In these cross-resolution scenarios, the approaches that utilize our DI as its input data have performed better. This is because our pre-processing stages help to reduce the gap between resolutions, hence learning a set of features more robust to resolution differences. Combining this with a low-level feature representation, which is more robust to acquisition noise, makes the learned features perform better in such scenarios.

Similar to previous datasets, we observe an increase in performance when utilizing attention in the proposed architecture. Every time any attention mechanism is used, our proposed method increases its performance and, in most cases, joining self- and cross-attention increases it even further. This is mostly evident for the protocols that include more samples in the gallery.

Table 4. MICC face dataset. Gallery is composed of high-resolution scans, while probe is composed from Kinect sequences. The results marked with * are from the network published in [Cardia Neto et al. 2019] utilized as feature extractor. PPD stands for pre-processed depth.

Method	N vs N	N vs NN	N vs A	A vs N	A vs NN	A vs A
Sigmoid*	63.64%	54.87%	55.34%	**81.82%**	73.85%	74.27%
3DLBP*	63.64%	61.54%	61.65%	63.64%	67.18%	66.99%
Fusion*	**81.82%**	61.54%	62.62%	72.73%	75.38%	75.24%
VGG [Kim et al. 2017]	18.18%	18.27%	18.26%	18.18%	18.27%	18.26%
ResNet50	**81.82%**	68.21%	68.23%	63.64%	76.41%	75.73%
Mobilenet v3	72.73%	65.13%	65.53%	**81.82%**	75.90%	76.21%
SAN - No-attention + Sigmoid + PPD	**81.82%**	67.69%	68.45%	72.73%	75.38%	75.24%
SAN - Self-attention + Sigmoid + PPD	72.73%	**71.79%**	**71.84%**	72.73%	80.00%	79.61%
SAN - Cross-attention + Sigmoid + PPD	**81.82%**	68.21%	68.96%	72.73%	77.95%	77.67%
SAN - Self- and cross-attention + Sigmoid + PPD	**81.82%**	70.77%	71.36%	**81.82%**	**86.67%**	**86.41%**

6 CONCLUSIONS

This work proposed a new low-resolution 3D face recognition approach, which proved effective and highly robust to resolution differences. Our proposed method develops on the assumption that a "hybrid" approach can be built, composed of a CNN that learns from multi-channel images made up of a combination of DIs, generated from handcrafted low-level features and depth data. The proposed sigmoid DI revealed more effective than the standard 3DLBP counterpart in encoding discriminative traits of the face from the depth images, while being sufficiently robust to the acquisition noise. However, we found the visual information in the generated images being significantly uncorrelated, which required processing each channel separately, and a strategy to fuse the information from each stream. This process resulted decisive to learn robust features. We achieved this by introducing self- and cross-attention mechanisms in our architecture. Both contributed to help the CNN focus on relevant parts of the face, generating better results. The cross-attention part of the attention module plays an important role in the performance of our approach, highlighting the importance of increasing collaboration from the different input streams. Another important aspect shown by our solution is the strong resolution invariance. The comparison with the deep VGG16 architecture evidences quite well this aspect; in fact, spite its impressive results on high-resolution datasets that can be found in the literature, when tested on low-resolution, VGG16 performance drops dramatically.

From the reported results, we can also conclude that, while the DIs and the depth resulted in comparable performance if used separately, joining them in our architecture led to a significantly improved accuracy. This suggests our solution comprising the attention mechanism allows us to effectively exploit all the information carried by each stream, being the consistently improved results on all the tested datasets a clear evidence of this. Hence, our solution would likely result effective if additional input data, *e.g.*, normal or curvature maps, are used. We also concluded that our architecture has yet some limitations that need to be further investigated. However, the higher performance compared to other solutions based on deep networks suggests us interesting future perspectives, that can open the way to the development of smaller yet effective networks for 3D face recognition systems.

ACKNOWLEDGMENTS

The TitanXp used for this research was donated by the NVIDIA Corporation.

CAPES 1–Brazil (PDSE grant number 88881.188744/2018-01) and Petrobras/Fundunesp (Process 2662/2017).

REFERENCES

Mohammed Y Abbass, Ki-Chul Kwon, Nam Kim, Safey A Abdelwahab, Fathi E Abd El-Samie, and Ashraf AM Khalaf. 2021. Efficient object tracking using hierarchical convolutional features model and correlation filters. *The Visual Computer* 37, 4 (2021), 831–842.

Wasseem N Ibrahem Al-Obaydy and Shahrel Azmin Suandi. 2020. Open-set single-sample face recognition in video surveillance using fuzzy ARTMAP. *Neural Computing and Applications* 32, 5 (2020), 1405–1412.

S. Berretti, A. Del Bimbo, and P. Pala. 2010. 3D Face Recognition Using Isogeodesic Stripes. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 32, 12 (Dec 2010), 2162–2177.

P. J. Besl and N. D. McKay. 1992. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14, 2 (Feb 1992), 239–256. https://doi.org/10.1109/34.121791

E. Bondi, P. Pala, S. Berretti, and A. Del Bimbo. 2016. Reconstructing High-Resolution Face Models From Kinect Depth Sequences. *IEEE Trans. on Information Forensics and Security* 11, 12 (Dec 2016), 2843–2853.

J. B. Cardia Neto and A. N. Marana. 2014. *3D Face Recognition Using Kinect.* Master's thesis. São Paulo State University (UNESP), Bauru SP 17033-360, Brazil.

J. B. Cardia Neto and A. N. Marana. 2018. Utilizing Deep Learning and 3DLBP for 3D Face Recognition. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications (CIARP).* 135–142.

J. B. Cardia Neto, A. N. Marana, C. Ferrari, S. Berretti, and A. Del Bimbo. 2019. Depth Based Face Recognition by Learning from 3D-LBP Images. In *Eurographics Workshop on 3D Object Retrieval (3DOR).* –.

H. Drira, B. Ben Amor, A. Srivastava, M. Daoudi, and R. Slama. 2013. 3D Face Recognition under Expressions, Occlusions, and Pose Variations. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 35, 9 (Sept 2013), 2270–2283.

A. Drosou, P. Moschonas, and D. Tzovaras. 2013. Robust 3D face recognition from low resolution images. In *Int. Conf. of the BIOSIG Special Interest Group.* 1–8.

T. C. Faltemier, K. W. Bowyer, and P. J. Flynn. 2008. A Region Ensemble for 3-D Face Recognition. *IEEE Trans. on Information Forensics and Security* 3, 1 (March 2008), 62–73.

Claudio Ferrari, Stefano Berretti, Pietro Pala, and Alberto Del Bimbo. 2021. A Sparse and Locally Coherent Morphable Face Model for Dense Semantic Correspondence Across Heterogeneous 3D Faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence* to appear (2021).

Claudio Ferrari, Stefano Berretti, Pietro Pala, and Alberto Del Bimbo. 2022. The MICC-3D Face Dataset. *Sensors* to appear (2022).

Claudio Ferrari, Giuseppe Lisanti, Stefano Berretti, and Alberto Del Bimbo. 2018. Investigating nuisances in DCNN-based face recognition. *IEEE Transactions on Image Processing* 27, 11 (2018), 5638–5651.

Leonardo Galteri, Claudio Ferrari, Giuseppe Lisanti, Stefano Berretti, and Alberto Del Bimbo. 2019. Deep 3D morphable model refinement via progressive growing of conditional Generative Adversarial Networks. *Computer Vision and Image Understanding* 185 (2019), 31–42.

Syed Zulqarnain Gilani and Ajmal Mian. 2018. Learning From Millions of 3D Scans for Large-Scale 3D Face Recognition. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR).* 1896–1905.

Syed Zulqarnain Gilani, Ajmal Mian, and Peter Eastwood. 2017. Deep, dense and accurate 3D face correspondence for generating population specific deformable models. *Pattern Recognition* 69 (2017), 238–250.

G. Goswami, S. Bharadwaj, M. Vatsa, and R. Singh. 2013. On RGB-D face recognition using Kinect. In *IEEE Int. Conf. on Biometrics: Theory, Applications and Systems (BTAS).* 1–6. https://doi.org/10.1109/BTAS.2013.6712717

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. arXiv:1512.03385 [cs.CV]

R. He, J. Cao, L. Song, Z. Sun, and T. Tan. 2020. Adversarial Cross-Spectral Face Completion for NIR-VIS Face Recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 42, 5 (2020), 1025–1037.

M. Hernandez, J. Choi, and G. Medioni. 2012. Laser scan quality 3-D face modeling using a low-cost depth camera. In *European Signal Processing Conf. (EUSIPCO).* 1995–1999.

Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. 2019. Searching for MobileNetV3. arXiv:1905.02244 [cs.CV]

Zhenguo Hu, Qijun Zhao, and Feng Liu. 2019. Revisiting Depth-Based Face Recognition from a Quality Perspective. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) Workshops.* 0–0.

Y. Huang, Y. Wang, and T. Tan. 2006. Combining Statistics of Geometrical and Correlative Features for 3D Face Recognition. In *British Machine Vision Conf. (BMVC).* 90.1–90.10. doi:10.5244/C.20.90.

Luo Jiang, Juyong Zhang, and Bailin Deng. 2019. Robust RGB-D Face Recognition Using Attribute-Aware Loss. *IEEE transactions on pattern analysis and machine intelligence* (2019).

D. Kim, M. Hernandez, J. Choi, and G. Medioni. 2017. Deep 3D face identification. In *IEEE Int. Joint Conf. on Biometrics (IJCB).* 133–142. https://doi.org/10.1109/BTAS.2017.8272691

B.Y.L. Li, A.S. Mian, Wanquan Liu, and A. Krishna. 2013. Using Kinect for face recognition under varying poses, expressions, illumination and disguise. In *Applications of Computer Vision (WACV), 2013 IEEE Workshop on.* 186–192. https://doi.org/10.1109/WACV.2013.6475017

Jiashu Liao, Alex Kot, Tanaya Guha, and Victor Sanchez. 2020. Attention Selective Network For Face Synthesis And Pose-Invariant Face Recognition. In *2020 IEEE International Conference on Image Processing (ICIP)*. 748–752. https://doi.org/10.1109/ICIP40778.2020.9190677

T. Mantecón, C. R. del-Blanco, F. Jaureguizar, and N. García. 2016. Visual Face Recognition Using Bag of Dense Derivative Depth Patterns. *IEEE Signal Processing Letters* 23, 6 (June 2016), 771–775.

R. Min, J. Choi, G. Medioni, and J. Dugelay. 2012. Real-time 3D face identification from a depth camera. In *Int. Conf. on Pattern Recognition (ICPR)*. 1739–1742.

Rui Min, Neslihan Kose, and Jean-Luc Dugelay. 2014. KinectFaceDB: A Kinect Database for Face Recognition. *IEEE Trans. on Systems, Man, and Cybernetics: Systems* 44, 11 (Nov 2014), 1534–1548. https://doi.org/10.1109/TSMC.2014.2331215

Guodong Mu, Di Huang, Guosheng Hu, Jia Sun, and Yunhong Wang. 2019. Led3D: A Lightweight and Efficient Deep Approach to Recognizing Low-Quality 3D Faces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5773–5782.

Timo Ojala, Matti Pietikäinen, and David Harwood. 1996. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition* 29, 1 (jan 1996), 51–59.

Mostafa Parchami, Saman Bashbaghi, and Eric Granger. 2017. Cnns with cross-correlation matching for face recognition in video surveillance using a single training sample per person. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 1–6.

Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. 2015. Deep Face Recognition. In *British Machine Vision Conf. (BMVC)*. 6.

P Jonathon Phillips, Patrick J Flynn, Todd Scruggs, Kevin W Bowyer, Jin Chang, Kevin Hoffman, Joe Marques, Jaesik Min, and William Worek. 2005. Overview of the face recognition grand challenge. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Vol. 1. 947–954.

Arman Savran, Neşe Alyüz, Hamdi Dibeklioğlu, Oya Çeliktutan, Berk Gökberk, Bülent Sankur, and Lale Akarun. 2008. Bosphorus database for 3D face analysis. In *European Workshop on Biometrics and Identity Management*. Springer, 47–56.

F. Schroff, D. Kalenichenko, and J. Philbin. 2015. FaceNet: A unified embedding for face recognition and clustering. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 815–823.

Maneet Singh, Shruti Nagpal, Mayank Vatsa, Richa Singh, and Angshul Majumdar. 2018. Identity Aware Synthesis for Cross Resolution Face Recognition. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) Workshops*. 592–59209.

Luuk Spreeuwers. 2011. Fast and Accurate 3D Face Recognition. *Int. Journal of Computer Vision* 93, 3 (July 2011), 389–414.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

Heng Wang, Du Tran, Lorenzo Torresani, and Matt Feiszli. 2020a. Video Modeling With Correlation Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Qiangchang Wang, Tianyi Wu, He Zheng, and Guodong Guo. 2020b. Hierarchical Pyramid Diverse Attention Networks for Face Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Qiang Wang, Yun Zheng, Pan Pan, and Yinghui Xu. 2021. Multiple Object Tracking With Correlation Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 3876–3886.

Xingwang Xiong, Xu Wen, and Cheng Huang. 2019. Improving RGB-D face recognition via transfer learning from a pretrained 2D network. In *Int. Symposium on Benchmarking, Measuring and Optimization*. Springer, 141–148.

Xi Yin and Xiaoming Liu. 2018. Multi-Task Convolutional Neural Network for Pose-Invariant Face Recognition. *IEEE Trans. on Image Processing* 27, 2 (Feb 2018), 964–975.

Yuechen Yu, Yilei Xiong, Weilin Huang, and Matthew R Scott. 2020. Deformable siamese attention networks for visual object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6728–6737.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2921–2929.