Expression removal in 3D faces for recognition purposes

Lucas Amparo Barbosa*†, Gabriel Dahia, Maurício Pamplona Segundo
Intelligent Vision Research Lab, Federal University of Bahia, Brazil

***Abstract*—We present an encoder-decoder neural network to remove deformations caused by expressions from 3D face images. It receives a 3D face with or without expressions as input and outputs its neutral form. Our objective is not to obtain the most realistic results but to enhance the accuracy of 3D face recognition systems. To this end, we propose using a recognition-based loss function during training so that our network can learn to maintain important identity cues in the output. Our experiments using the Bosphorus 3D Face Database show that our approach successfully reduces the difference between face images from the same subject affected by different expressions and increases the gap between intraclass and interclass difference values. They also show that our synthetic neutral images improved the results of four different well-known face recognition methods.**

***Index Terms*—Deep Learning, Facial Recognition, 3D Images**

I. Introduction

Faces are a widely used source of information for recognizing individuals, either in a real or virtual context. This process is non-intrusive and can be used in a variety of ways, from a local system that uses controlled acquisition to large-scale security systems [1]. Recognition can be done by many different methods [2]–[5] and could use 2D and/or 3D images. Regardless of advantages and disadvantages, all these image modalities share a common problem: humans use facial expressions to communicate, which changes the facial shape making it different from its respective neutral version [6]. This can cause a recognition system to consider two face images from different individuals more similar than two images of the same person with different expressions.

The most straightforward way to address this problem is to focus on face regions less affected by expressions, such as the area around the eyes in 2D images [7] and around the nose in 3D images [8], [9]. A generalization of this idea assumes that a small neighborhood around any point of the face is locally rigid and less prone to expression variations, which led to the creation of part-based matching approaches [10], [11]. Creating expression invariant representations for 3D images is an alternative approach which found some success [12], [13].

Dealing with expressions means modeling deformations to an extent that allows imitating and/or removing them. Expression simulation is better fitted to controlled systems that ensure enrolled faces are neutral. This way, an enrolled face can be deformed according to a precomputed model to match an input face with expression variations [14]. Meanwhile, expression removal does not have such requirement as it can eliminate deformations caused by facial expressions from all images before matching them to each other. This is a huge advantage over the former option when considering an identification scenario, in which a probe image is matched against multiple enrolled images, because a gallery can be preprocessed for expression removal but not for expression simulation.

As expressions are mostly shape changes, 3D images pioneered removal studies. Pan *et al.* [15] learned how to infer the expression residue of a non-neutral face using Radial Basis Function regression model. With this approach, they can later subtract this residue from the input face to reconstruct its neutral form. This second step can be eliminated by modeling both identity shape and expression residue together with a deformable model based on Active Shape Models [16].

With the rise of Convolutional Neural Networks (CNN), recent studies have successfully used generative adversarial learning to obtain realistic results for this task using 2D images as well [17]–[19]. Despite that, latest CNN-based works using 3D images focus on the expression recognition task [20], [21] but not on face recognition under expression variations.

In this work, we aim to use CNNs to remove expressions from 3D faces specifically for recognition purposes. We do not require realistic images as long as recognition accuracy improves. To achieve this objective, we:

- describe a neural network model - adapted from Badrinarayanan *et al.* [22] - to map non-neutral 3D face images to their correspondent neutral versions (Section II-C);
- propose, as our main contribution, using a recognition-based loss function to regularize the training into maintaining or improving discriminability after expression removal (Section II-C);
- compare the effect of our expression removal on shape similarity with the state-of-the-art (Section III-A);
- test our method with established face recognition methods to show its potential to improve them (Section III-B).

II. Methodology

We propose to use a CNN to, given a 3D face image with or without expressions, output the individual's neutral face. Considering the great advances in machine learning for computer vision using 2D representations [22]–[25], we decided to avoid volumetric representations and preferred to use 2D orthogonal projection images of 3D faces. This choice also brings a second benefit: a considerable reduction in the number of model parameters.

* *Corresponding author:* lucasamparo.ti@gmail.com

† This research was funded by FAPESB (http://www.fapesb.ba.gov.br). The Titan Xp used for this research was donated by the NVIDIA Corporation.

To accomplish the aforementioned objective, we first normalize the pose of the 3D face by aligning it to an average face model. Then, we project it to 2D space and use our neural network to remove any deformations caused by facial expressions. The output 2D projection can be easily reverted to a 3D mesh if necessary. This process is illustrated in Figure 1.

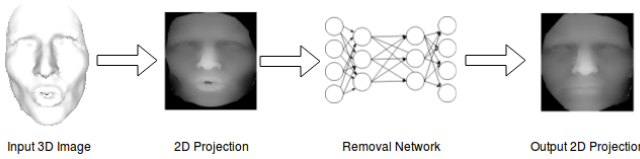

Fig. 1. Proposed pipeline for facial expression removal. Input 3D faces are converted to 2D projections, which are then fed to an encoder-decoder network to produce their correspondent neutral images. The output image can be mapped back to 3D space by using the inverse projection transformation.

A. Acquisition

We used the Bosphorus 3D Face Database [26] (hereon Bosphorus) as a source of 3D faces in this work. It contains 3D images of faces from 105 individuals and an average of 36,000 points, with 3D coordinates in milimeters. We used nearly frontal faces only, totaling 299 neutral faces and 2189 faces with expression variations.

To perform a fair evaluation of our approach, we split this dataset into training, validation and test subsets in a subject-independent manner (*i.e.* no individual has images in more than one subset). To do so, we randomly selected half of the available individuals for training (53 people), one-fourth for validation (26 people) and one-fourth for test (26 people).

B. Pose normalization and 2D projection

Our normalization process requires an average face model, which is created using all neutral faces in the training set. These images are aligned to each other using the Iterative Closest Points (ICP) algorithm [27], and corresponding points are averaged to create the final model (Figure 2). Lastly, the center of mass of the resulting average model is shifted to the origin.

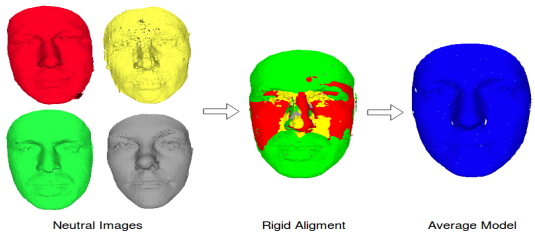

Fig. 2. Average face model created by aligning all neutral faces in the training subset to each other and averaging corresponding points.

To normalize the pose of an input 3D face, we align it to the average face model with ICP and then project it to a 128×128 orthogonal projection image by applying the following equations to every 3D point x_i, y_i, z_i:

$$r = \lfloor 64 - y_i \rceil \tag{1}$$

$$c = \lfloor 64 + x_i \rceil \tag{2}$$

$$I(r, c) = 127 + 3z_i \tag{3}$$

where r and c are the row and column of the pixel where the point is being projected, $\lfloor a \rceil$ is the nearest integer to a, and I is the projection image. Finally, we fill holes with the values of neighboring pixels. Figure 3 shows an example of the intermediate steps of this process and its result.

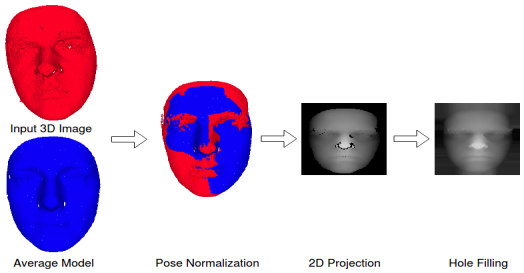

Fig. 3. Orthogonal projection of a 3D face into a 2D image. An input 3D face is aligned to a precomputed average model. After that, y and x coordinates become row and column while the z axis becomes the pixel intensity.

C. Neural Network Model

Our network is an adaptation of the architecture proposed by Badrinarayanan *et al.* [22], consisting of a convolutional encoder-decoder to regress, from a 2D orthogonal projection of the input 3D face, a 2D orthogonal projection of the same face without facial expressions.

The encoder's input is the 2D orthogonal projection image created by the previous step (Section II-B). Our encoder consists of two blocks of two convolutional layers followed by max-pooling, and then three convolutional layers, all using the ReLU activation function. The encoder output is a feature vector in latent space of size $4 \times 4 \times 64$. When compared to Badrinarayanan *et al.*'s [22] architecture, the last convolutional layer in our encoder was originally a max-pooling layer.

The decoder is symmetrical to the encoder, replacing convolutions with transposed convolutions and max-pooling with up-sampling. The other difference to Badrinarayanan *et al.*'s architecture [22] is the absence of an activation function in the last layer of our network. The complete description of our architecture can be seen in Table I.

TABLE I
NEURAL NETWORK ARCHITECTURE FOR FACIAL EXPRESSION REMOVAL.

	#	Type	Input	Filter	Stride	Output
Encoder	1	Convolutional + ReLU	128×128×1	7×7×1×64	2	64×64×64
	2	Convolutional + ReLU	64×64×64	7×7×64×64	1	64×64×64
	3	Max Pooling	64×64×64	2×2	2	32×32×64
	4	Convolutional + ReLU	32×32×64	7×7×64×64	2	16×16×64
	5	Convolutional + ReLU	16×16×64	7×7×64×64	1	16×16×64
	6	Max Pooling	16×16×64	2×2	2	8×8×64
	7	Convolutional + ReLU	8×8×64	7×7×64×64	2	4×4×64
	8	Convolutional + ReLU	4×4×64	7×7×64×64	1	4×4×64
	9	Convolutional + ReLU	4×4×64	7×7×64×64	1	4×4×64
Decoder	10	Deconvolutional + ReLU	4×4×64	7×7×64×64	1	4×4×64
	11	Deconvolutional + ReLU	4×4×64	7×7×64×64	1	4×4×64
	12	Deconvolutional + ReLU	4×4×64	7×7×64×64	2	8×8×64
	13	Upsampling	8×8×64	2×2	2	16×16×64
	14	Deconvolutional + ReLU	16×16×64	7×7×64×64	1	16×16×64
	15	Deconvolutional + ReLU	16×16×64	7×7×64×64	2	32×32×64
	16	Upsampling	32×32×64	2×2	2	64×64×64
	17	Deconvolutional + ReLU	64×64×64	7×7×64×64	1	64×64×64
	18	Deconvolutional	64×64×64	7×7×64×1	2	128×128×1

With that in mind, we state our learning problem as optimizing the network's parameters by minimizing the L_2 loss between the network output $\hat{\mathbf{y}}$ and the ground truth $\mathbf{y}$. We also considered an alternative loss function to help preserving the identity of an input image. To this end, we used the Euclidean distance between CNN descriptors extracted by a state-of-the-art 2D face recognition approach, a publicly available implementation inspired by Facenet [4]. For simplicity, we hereon refer to this CNN as Color-Facenet and to this loss scheme as recognition loss. The recognition loss is defined as:

$$\mathcal{L}_r = \gamma \|h(\hat{\mathbf{y}}) - h(\mathbf{y})\| + \|h(\hat{\mathbf{y}}) - h(\mathbf{x})\| \quad (4)$$

where $h(\mathbf{a})$ is the Color-Facenet descriptor for $\mathbf{a}$. To measure the benefits of the recognition loss, we evaluated two versions of the proposed expression removal network:

- **Net A**: optimized using L_2 loss only;
- **Net B**: optimized alternating L_2 loss with the $\mathcal{L}_r$ loss.

Figure 4 illustrates the difference between **Net A** and **Net B**.

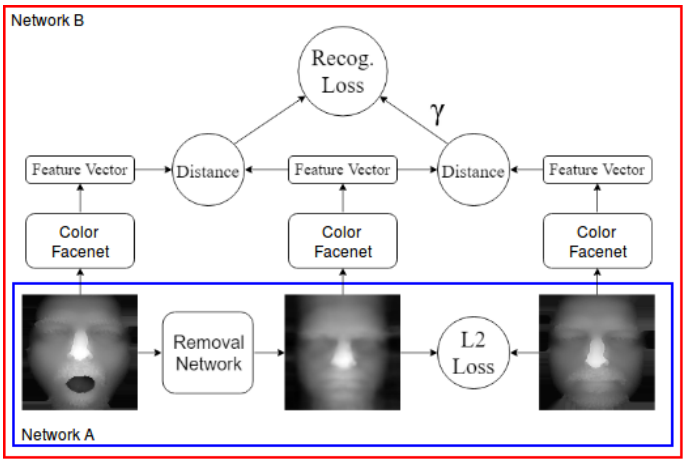

Fig. 4. Optimization strategies for **Net A** and **Net B**.

It is important to mention that the weights of Color-Facenet are not updated during training. We are interested in using its gradient as a surrogate for the gradient of face recognition in general, aiming with this to regularize our objective into keeping or improving face recognition when using the outputs of our method. Although Color-Facenet was trained only on 2D images, it is known that such models generalize reasonably well to other face recognition modalities [28].

We use the distance between input and output image descriptors, respectively $h(\mathbf{x})$ and $h(\hat{\mathbf{y}})$, as well as between output and ground truth image descriptors, respectively $h(\hat{\mathbf{y}})$ and $h(\mathbf{y})$, to take identity cues from input and ground truth images into account. γ is a hyperparameter that controls to which of the descriptors, input or ground truth, we would like the output descriptor to be closer. Increasing γ prioritizes expressionless face recognition, and we empirically set $\gamma = 3$.

For both versions, the training phase was divided into two parts: autoencoder-based pre-training and encoder-decoder training for expression removal.

1) Autoencoder-based pre-training: learns an identity map for neutral human faces in orthogonal projection images. It uses the same image as input and expected output, so it learns how to perform dimensionality reduction on the input space and then to reconstruct this input minimizing the error. It is an easier problem when compared to the expression removal problem, but its domain is close enough to serve as a good initialization for the next training part [29]. We optimized the L_2 loss with the Adam optimizer [30], learning rate with exponential decay from 10^{-2} to 10^{-5} and batches of 256 samples for 50 epochs. These parameters were empirically defined based on the performance in the validation set. The same initialization process was used for **Net A** and **Net B**.

2) Encoder-decoder training: uses every non-neutral, neutral pair of faces in the training set to train our expression removal net. For **Net A** we use the same training configuration of the pre-training stage, except by the number of epochs, which is now 3000. For **Net B**, we intercalate training batches using the L_2 and $\mathcal{L}_r$ losses in a proportion of four to one. Besides also using 3000 epochs, we use a learning rate between 10^{-5} to 10^{-7} in batches that use the $\mathcal{L}_r$ loss.

III. Experiments

We ran our experiments in an Intel i7-6700k 4GHz machine with 32GB of RAM and a single NVIDIA Titan X Pascal 12GB GPU. Our implementation uses the Point Cloud [31], OpenCV [32], and Tensorflow [33] libraries.

We conducted two experiments in our empirical evaluation of the proposed approach. First, we provide a quantitative analysis of the stability of our approach, comparing with the state-of-the-art, in Section III-A. Finally, we compare the performance of four face recognition methods before and after using the proposed work for expression removal in Section III-B. By doing that, we were able to establish a baseline performance for each method and then to quantitatively assess our method's effect on face recognition.

A. Evaluation of Expression Removal Stability

A stable expression removal approach should consistently increase intraclass similarity and create a larger separation between intraclass and interclass similarity distributions. To quantify this stability, we used the Root Mean Squared Error (RMSE) between pairs of raw images. The closer to zero the RMSE is, more similar these images will be. We computed the RMSE for every pair of neutral and non-neutral faces in the test set and computed the average RMSE value for intraclass and interclass combinations. This procedure was repeated for the original images and their processed versions using **Net A** and **Net B**, and the obtained values are presented in Table II. As expected, the average RMSE for intraclass combinations is smaller than interclass combinations in all cases. After expression removal, both interclass and interclass RMSE values are reduced, meaning that images from the same person are getting more similar but images from different individuals are getting closer too. However, the gap between intraclass and interclass values increases, which shows that

removing expressions helps discriminating them. Figure 5 visually illustrates this effect.

TABLE II
AVERAGE RMSE VALUES AND THEIR STANDARD DEVIATION

	Intraclass	**Interclass**
Original	0.046176 ± 0.011230	0.072338 ± 0.011539
Net A	0.024199 ± 0.008934	0.059927 ± 0.014833
Net B	0.024082 ± 0.008785	0.059696 ± 0.014278

When compared to the state-of-the-art, our approach was able to reduce the average intraclass RMSE by 47.8%, similar to the reduction of 43.6% achieved by Pan *et al.* [15]. Although they use a different database, BU-3DFE [34], it is very similar to the subset from BOSPHORUS that was used in this work (BU-3DFE vs. our BOSPHORUS subset: 100 vs. 105 individuals; 1 vs. 2.84 neutral faces per person; 24 vs. 20.8 non-neutral faces per person; different expression intensities in both). The main difference is that we use fixed training and test sets, while Pan *et al.* follow a leave-one-out protocol and use all but one person for training and the remaining person for testing. They repeat this experiment (leaving one person out at a time) and compute the average result. This means we achieve comparable results using a 2:1 train:test ratio while Pan *et al.* used a 99:1 ratio.

These results suggest that a face recognition system could benefit from our approach, so we continued our experiments with an evaluation of our method's impact on different face recognition systems in Section III-B.

B. Expression Removal's Effect on Face Recognition

To evaluate the effect of our approach for expression removal over face recognition, we investigated the results of four different methods: Eigenfaces [2], Fisherfaces [3], Local Binary Pattern Histograms (LBPH) [5] and Color-Facenet [4]. Eigenfaces and Fisherfaces were trained using only neutral faces from the training set. By doing so, they represent a recognition system that cannot handle expression variations. The validation set was used to find the number of eigenvectors with the best recognition performance, 30. Thus, each face in the test set was later described by a 30-dimensional vector in both cases. LBPH and Color-Facenet did not require any training, and were used to extract 4096- and 128-dimensional descriptors from a test face, respectively. LBPH is known for being robust against expression variations [35]. The Color-Facenet model was trained using 2D face images, but could be used to extract features on 3D images as well [28] and helps us checking if our method improves a CNN's performance.

Each recognition experiment was repeated three times, as in Section III-A, one using the original images and the other two using these images after being processed by **Net A** and **Net B**. We remark that even neutral faces need to be processed by the nets in order to obtain the results presented in Figure 5.

For each combination of recognition approach and input data, we performed two experiments: an All versus All matching of images in the test subset, whose results were reported as Receiver Operating Characteristic (ROC) curves in Figure 6 and Equal Error Rate (EER) values in Table III; and a Rank-N identification using one or two neutral images for each individual to form a gallery and all non-neutral images as probe samples, whose results were reported as Cumulative Match Characteristic (CMC) curves in Figure 7. The former experiment evaluates how well genuine and impostor matches can be distinguished, while the latter tells how easily the identity of a probe can be retrieved in a controlled scenario in which facial expressions are not allowed during enrollment.

Overall, the benefit of removing expressions with either

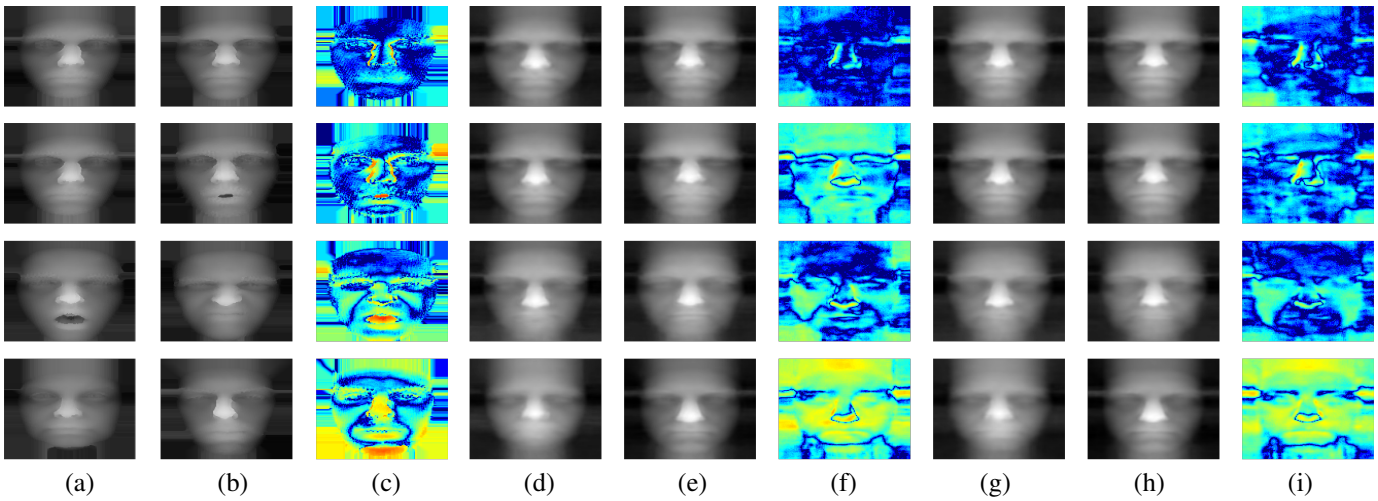

Fig. 5. Results to illustrate the applicability of our expression removal in face recognition. The first three rows show the comparison of pairs images from the same person in different scenarios: neutral vs. neutral, neutral vs. non-neutral and non-neutral vs. non-neutral. The last row shows a neutral vs. neutral comparison between images from different subjects. The original images are shown in (a) and (b), and their difference is (c). (d) and (e) shown results of **Net A** for (a) and (b), and their difference is (f). Finally, outputs from **Net B** for (a) and (b) are shown in (g) and (h) and their difference in (i). Areas with large difference are red and areas with small difference are blue. A logarithmic scale was used to magnify those differences.

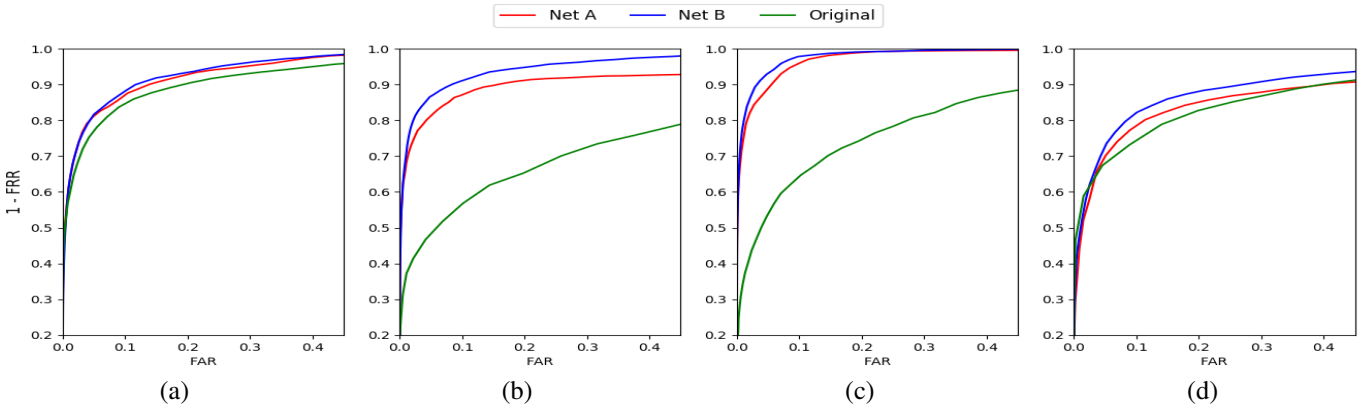

Fig. 6. ROC curves for verification results using (a) Eigenfaces, (b) Fisherfaces, (c) LBPH, and (d) Color-Facenet. FRR stands for False Rejection Rate and FAR stands for False Acceptance Rate.

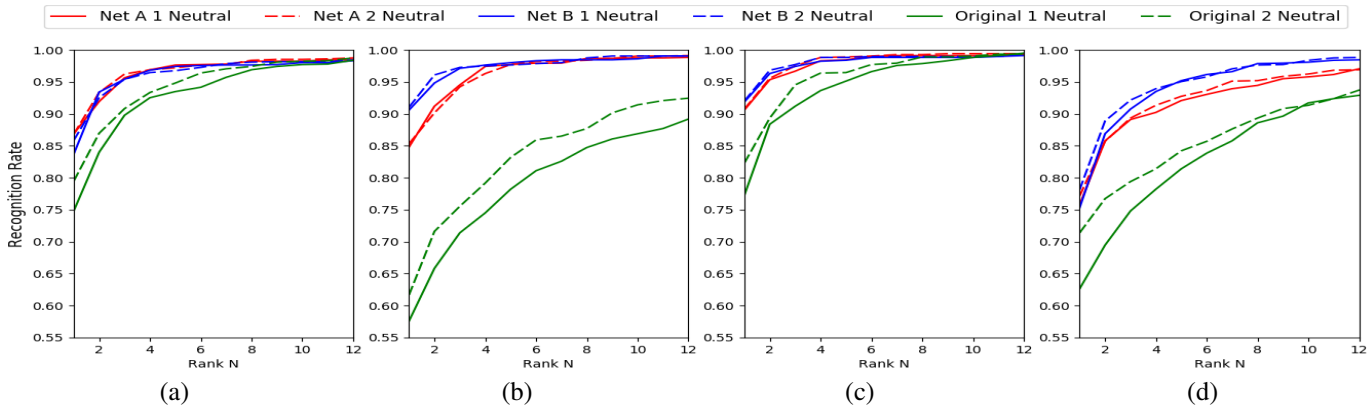

Fig. 7. CMC curves for identification results using (a) Eigenfaces, (b) Fisherfaces, (c) LBPH, and (d) Color-Facenet. Solid lines are obtained when there is one image per person in the gallery, and dashed lines when there are two images per person.

Net A or **Net B** is very clear in comparison to the results for unprocessed images. Rank-1 and EER were improved in all cases, with a reasonable advantage of **Net B** over **Net A** in most cases. This shows that removing expressions improves the accuracy of a recognition system. We observe this even when the recognition method for which the network was optimized is different from the one being used, showing that our approach of using a state-of-the-art neural network as a surrogate gradient for face recognition is promising. Nevertheless, we can make other observations based on the results presented in Figures 6 and 7 and Table III.

1) We can see that Eigenfaces outperforms Fisherfaces for the original image set but the opposite happens when facial expressions are removed by **Net B**. Both methods were trained with neutral images only. Fisherfaces probably became too specialized to this domain to handle expression variations. Meanwhile, Eigenfaces seeks to represent the main components of the training set, which helped it generalize better to a different domain. Fisherfaces' discriminative power is restored on expression removed images, which indicates that its training domain (neutral faces only) was consistently

TABLE III
EER CONSIDERING DIFFERENT GROUPS OF GENUINE COMPARISONS: ALL VERSUS ALL, NEUTRAL VERSUS NON-NEUTRAL AND NON-NEUTRAL VERSUS NON-NEUTRAL IMAGES. THE SET OF IMPOSTOR COMPARISONS IS ALWAYS THE SAME (ALL VERSUS ALL).

Method	All vs. All			Neutral vs. Non-Neutral			Non-Neutral vs. Non-Neutral		
	Net A	Net B	Original	Net A	Net B	Original	Net A	Net B	Original
PCA	0.116708	**0.109266**	0.133388	0.121929	**0.116014**	0.123910	0.128130	**0.120554**	0.192649
LDA	0.118026	**0.094660**	0.280848	0.122854	**0.099432**	0.280226	0.112079	**0.098947**	0.377257
LBPH	0.072148	**0.058066**	0.229458	0.075262	**0.061903**	0.230442	0.081411	**0.066287**	0.278878
Color-Facenet	0.169345	**0.146602**	0.188003	0.178849	**0.152614**	0.187543	0.174687	**0.158206**	0.255858

reinstated by the use of our method.

2) In Figure 7, we see that the Rank-N increases for original images when the gallery has two images per person, but not for images processed by **Net A** or **Net B**. Adding multiple images of the same person to the gallery is only useful when there is variation between them [36]. This suggests that even neutral faces get closer to each other when processed by our networks and that our method can possibly reduce the enrollment effort in face recognition systems.
3) The small difference between non-neutral versus neutral and non-neutral versus non-neutral matches in Table III for **Net A** and **Net B** indicates that there is no need to control user expression during enrollment in recognition systems that use our method.
4) There is an increase in accuracy even when methods that are robust to expressions variations are evaluated, showing that the benefits of expression removal are not limited to expressionless methods.
5) 3D recognition results using Color-Facenet were not as high as other methods, as expected from a model trained with color images. They show, however, that our method can enhance the accuracy of CNN-based systems.

IV. Conclusion and Future Works

We used an encoder-decoder neural network for facial expression removal in 3D images aiming to improve the accuracy of recognition systems. Our main contribution is the use of a recognition system to guide the training process to maintain identity cues in the output neutral image, and we show the advantages of the proposed approach through qualitative and quantitative evaluations. Our approach was able to reduce the RMSE between a neutral and a non-neutral image roughly by half, which is comparable to the state-of-the-art while requiring much less training data. It also increases the separation between intraclass and interclass RMSE values. When used for recognition purposes, images processed by our approach improved the accuracy of four different face recognition systems, showing that it is system agnostic. Finally, we observed that our approach reduces the enrollment effort in a recognition system, as it requires fewer gallery samples and removes the need to impose neutral expressions in enrollment.

As a future work, we intend to combine multiple 3D face datasets in order to train a 3D recognition CNN to be used as a better estimate for the recognition loss. We also intend to investigate if more realistic images obtained by GANs [37] can further improve the recognition results.

References

[1] W.-Y. Zhao *et al.*, "Face recognition: A literature survey," *ACM Computing Surveys*, vol. 35, pp. 399–458, 12 2003.

[2] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[3] P. N. Belhumeur *et al.*, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *IEEE TPAMI*, vol. 19, no. 7, pp. 711–720, 1997.

[4] F. Schroff *et al.*, "Facenet: A unified embedding for face recognition and clustering," *CoRR*, 2015.

[5] T. Ahonen *et al.*, "Face recognition with local binary patterns," in *ECCV*, 2004, pp. 469–481.

[6] C. Frith, "Role of facial expressions in social interactions," *Philos. Trans. Royal Soc. B*, vol. 364, no. 1535, pp. 3453–3458, 2009.

[7] T. E. Campos *et al.*, "Eigenfaces versus eigeneyes: First steps toward performance assessment of representations for face recognition," in *MICAI*, 2000, pp. 193–201.

[8] K. I. Chang *et al.*, "Multiple nose region matching for 3d face recognition under varying facial expression," *IEEE TPAMI*, vol. 28, no. 10, pp. 1695–1700, 2006.

[9] M. Emambakhsh and A. Evans, "Nasal patches and curves for expression-robust 3d face recognition," *IEEE TPAMI*, vol. 39, no. 5, pp. 995–1007, 2017.

[10] H. Li *et al.*, "Eigen-pep for video face recognition," in *ACCV*, 2015, pp. 17–33.

[11] S. Elaiwat *et al.*, "3-d face recognition using curvelet local features," *IEEE Signal Processing Letters*, vol. 21, no. 2, pp. 172–175, 2014.

[12] S. Berretti *et al.*, "3d face recognition using isogeodesic stripes," *IEEE TPAMI*, vol. 32, no. 12, pp. 2162–2177, 2010.

[13] I. A. Kakadiaris *et al.*, "Three-dimensional face recognition in the presence of facial expressions: An annotated deformable model approach," *IEEE TPAMI*, vol. 29, no. 4, pp. 640–649, 2007.

[14] X. Lu and A. Jain, "Deformation modeling for robust 3d face matching," *IEEE TPAMI*, vol. 30, no. 8, pp. 1346–1357, 2008.

[15] G. Pan *et al.*, "Removal of 3d facial expressions: A learning-based approach," in *CVPR*, June 2010, pp. 2614–2621.

[16] A. S. Agianpuye and J. L. Minoi, "Synthesizing neutral facial expression on 3d faces using active shape models," in *IEEE REGION 10 SYMPOSIUM*, April 2014, pp. 600–605.

[17] H. Ding *et al.*, "Exprgan: Facial expression editing with controllable expression intensity," in *AAAI*, 2018.

[18] L. Song *et al.*, "Geometry guided adversarial facial expression synthesis," *CoRR*, 2017.

[19] H. Ding *et al.*, "Exprgan: Facial expression editing with controllable expression intensity," *CoRR*, 2017.

[20] A. Jan *et al.*, "Accurate facial parts localization and deep learning for 3d facial expression recognition," in *IEEE FG*, 2018.

[21] H. Yang and L. Yin, "Cnn based 3d facial expression recognition using masking and landmark features," in *ACII*, 2017, pp. 556–560.

[22] V. Badrinarayanan *et al.*, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *CoRR*, 2015.

[23] A. Radford *et al.*, "Unsupervised representation learning with deep convolutional generative adversarial networks," *CoRR*, 2015.

[24] A. Krizhevsky *et al.*, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.

[25] G. Liu *et al.*, "Image Inpainting for Irregular Holes Using Partial Convolutions," *ArXiv e-prints*, Apr. 2018.

[26] A. Savran *et al.*, "Bosphorus database for 3d face analysis," in *The First COST 2101 Workshop on BIOID*, May 2008.

[27] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE TPAMI*, vol. 14, no. 2, pp. 239–256, 1992.

[28] G. Dahia *et al.*, "A study of cnn outside of training conditions," in *IEEE ICIP*, 2017, pp. 3820–3824.

[29] A. Shrivastava *et al.*, "Learning from simulated and unsupervised images through adversarial training," *CoRR*, 2016.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[31] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *ICRA*, 2011, pp. 1–4.

[32] G. Bradski, "Opencv - open computer vision library," *Dr. Dobb's Journal of Software Tools*, 2000.

[33] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015.

[34] L. Yin *et al.*, "A 3d facial expression database for facial behavior research," in *IEEE FG*, 2006, pp. 211–216.

[35] J. A. Khorsheed and K. Yurtkan, "Analysis of local binary patterns for face recognition under varying facial expressions," in *SIU*, 2016, pp. 2085–2088.

[36] K. W. Bowyer *et al.*, "Face recognition using 2-d, 3-d, and infrared: Is multimodal better than multisample?" *Proceedings of the IEEE*, vol. 94, no. 11, pp. 2000–2012, 2006.

[37] I. Goodfellow *et al.*, "Generative adversarial nets," in *NIPS*, 2014, pp. 2672–2680.