Paper Review: Large Pose 3D Face Reconstruction from a Single Image via Direct Volumetric CNN Regression

Antoni Mauricio
Research and Innovation Center
in Computer Science
Universidad Católica San Pablo
manasses.mauricio@ucsp.edu.pe

Jorge López
Research and Innovation Center
in Computer Science
Universidad Católica San Pablo
jorge.lopez.caceres@usp.pe

***Abstract*—3D reconstruction is one of the most recurrent problems in computer vision, its complexity goes through problems from objetc segmentation in a 3D environment to matching of points, most applications assume the availability of multiple images from different points of view as input and address a lot of propper issues. In 3D facial reconstruction, this issues include non-uniform illumination, expression and facial poses recognition in multiple facial images. The presented work proposes a nobel method based on Convolutional Neuronal Networks (CNN) using an 2D images as input dataset and voxel binary values (from 3D images and scans) as target dataset, but just a single 2D facial image in 3D reconstruction (using a trained model). The 3D output model works for arbitrary facial poses and expressions, and can be used to reconstruct whole 3D geometry, even non-visible parts of the face, bypassing the geometry model (in training) and fitting of a 3D Morphable Model - 3DMM (in testing). The method proposed include facial landmark localization task in CNN pipeline to improve reconstruction quality, especially for the cases of large poses and facial expressions, then CNN performs 2D input volumetric representation by direct regression of the 3D facial geometry output.**

Keywords: 3D facial reconstruction, CNN pipeline, 3D Morphable Model.

1. Introduction

3D face reconstruction implies to build a 3D geometry model from 2D images. Many solutions and approaches have been used through years to solve it, depending on some assumptions. One of most studied assumptions is high-quality facial reconstruction from a single image, considering non-uniform illumination, hidden areas, expression and facial poses recognition. In this paper, a CCN architecture is used to correlate statistically a mapping from 2D pixels to voxels in 3D coordinates.

The most popular approach in one image techniques for 3D face reconstruction is the 3D Morphable Model (3DMM) [1], [4]. 3DMM describes the 3D face space with a Principal Component Analisys (PCA), but applied over an iterative flow procedure for dense image correspondence becomes prone to failures. Also, its application requires a careful parameters initialisation in order to solve a difficult highly non-convex optimization problem (too slow).

3D face reconstruction. The work of [4] describes a multi-feature based approach to 3DMM fitting using non-linear least-squares optimization (Levenberg-Marquardt), which given appropriate initialisation produces results of good accuracy. More recent work has proposed to estimate the update for the 3DMM parameters using CNN regression, as opposed to non-linear optimization. In [12], the 3DMM parameters are estimated in six steps each of which employs a differentCNN. Notably, [12] estimates the 3DMM parameters on a sparse set of landmarks, i.e. the purpose of [12] is 3D face alignment rather than face reconstruction. The method of [2] is currently considered the state-of-the-art in 3DMM fitting. It is based on a single CNN that is iteratively applied to estimate the model parameters using as input the 2D image and a 3D-based representation produced at the previous iteration. Finally, a state-of-the-art cascaded regression landmark-based 3DMM fitting method is proposed in [13].

CNN-based depth estimation. [14], [15] showed that a CNN can be directly trained to regress from pixels to depth values using as input a single image. The article works from [14], [15] results, but it differs in 3 important respects: Firstly, the article focus on faces (i.e. deformable objects) whereas [5, 6] on general scenes containing mainly rigid objects. Secondly, [5, 6] learn a mapping from 2D images to 2D depth maps, while the article try to demonstrate that one can actually learn a mapping from 2D to the full 3D facial structure including the non-visible part of the face. Thirdly, [5, 6] use a multi-scale approach by processing images from low to high resolution. In contrast, process faces at fixed scale (assuming that this is provided by a face detector) is treaty is the article. Finally the article use a CNN-based approach on a state-of-the-art bottom-up top-down module [16] that allows analysing and combining CNN features at different resolutions for eventually making predictions at voxel level.

2. Method

This section describes the main mathematical ideas and deep learning architectures including the proposed data representation used in the paper and in our implementation. Figure 1 shows the general pipeline, Since an initial photo, the face detector algorithm (2D-FAN-300W) runs in order to crop the face from image (if there are much more than 1 face or non-face at all, then the rest of the model does not run). A resize/ normalize algorithm runs over cropped image. From this image, we start the training process. To do so, first we define the architecture (described on section 2.4), then we apply our defined cost functions (equation 1) using the conversion algorithm (2D RGB to 3D mesh) described on [20] as the error metric for back-propagation training stage. This is possible, even when the algorithm proposed on the paper works only to engage 2D/3D voxel conversion algorithm, due the pipeline transform 3D voxel to 3D mesh using *isosurface smoothing function* from ***MatLab***.

2.1. Dataset

Given the article objectives, required data are 2D images and 3D facial scans. According to the article, 2D images for training and testing datasets are obteined from [2], while 3D facial scans dataset (pre-trained on VRN-guided) has been produced by fitting a 3DMM built from the combination of the Basel [4] and FaceWarehouse [5] models to the unconstrained images of the 300W dataset [6] using the multi-feature fitting approach of [4], careful initialisation and by constraining the solution using a sparse set of landmarks. Face profiling is then used to render each image to 10-15 different poses resulting in a large scale dataset (more than 60,000 2D facial images and 3D meshes) called 300W-LP. Note that because each mesh is produced by a 3DMM, the vertices of all produced meshes are in dense correspondence; however this is not a prerequisite for our method and unregistered raw facial scans could be also used if available.

2.2. Hourglass Network

Hourglass architecture is motivated by the need to capture information at every scale. While local evidence is essential for identifying features like faces and hands, a final pose estimate requires a coherent understanding of the full body. The persons orientation, the arrangement of their limbs, and the relationships of adjacent joints are among the many cues that are best recognized at different scales in the image. The hourglass is a simple, minimal design that has the capacity to capture all of these features and bring them together to output pixel-wise predictions.

The network must have some mechanism to effectively process and consolidate features across scales. Some approaches tackle this with the use of separate pipelines that process the image independently at multiple resolutions and combine features later on in the network [7], [9]. Instead, we choose to use a single pipeline with skip layers to preserve spatial information at each resolution. The network reaches its lowest resolution at 4x4 pixels allowing smaller spatial filters to be applied that compare features across the entire space of the image.

The hourglass is set up as follows: Convolutional and max pooling layers are used to process features down to a very low resolution. At each max pooling step, the network branches off and applies more convolutions at the original pre-pooled resolution. After reaching the lowest resolution, the network begins the top-down sequence of upsampling and combination of features across scales. To bring together information across two adjacent resolutions, we follow the process described by Tompson et al. [9] and do nearest neighbor upsampling of the lower resolution followed by an elementwise addition of the two sets of features. The topology of the hourglass is symmetric, so for every layer present on the way down there is a corresponding layer going up.

After reaching the output resolution of the network, two consecutive rounds of 1x1 convolutions are applied to produce the final network predictions. The output of the network is a set of heatmaps where for a given heatmap the network predicts the probability of a joints presence at each and every pixel. The full module (excluding the final 1x1 layers) is illustrated in Figure 2.

2.3. Residual Module

Due backpropagation learning impact reduction, some feedback from original source is needed, then a residual signal from original is added. According to He et al. [8], an increase in network performance is obteined after switching from standard convolutional layers with large filters and no reduction steps to newer methods like the residual learning modules and inception-based designs [11].

After the initial performance improvement with these types of designs, various additional explorations and modifications to the layers did little to further boost performance or training time.

2.4. Volumetric Regression Networks

The main idea is do a probabilistic match between 2D image input and 3D volume $f : I \longrightarrow V$. For preprocessing, the paper converts each 3D facial scan into a 3D binary volume V_{whd} by discretizing the 3D space into voxels w, h, d, assigning a value of 1 to all points enclosed by the 3D facial scan, and 0 otherwise. That is to say V_{whd} is the ground truth for voxel w, h, d and is equal to 1, if voxel w, h, d belongs to the 3D volumetric representation of the face and 0 otherwise (i.e. it belongs to the background).Notice that the process creates a volume fully aligned with the 2D image. Given that the error of state-of-the-art methods [18], [19] is of the order of a few mms, the paper concludes that discretization by 192 x 192 x 200 produces negligible error.

The CNN architecture for 3D segmentation is based on the hourglass network, an extension of the fully con-

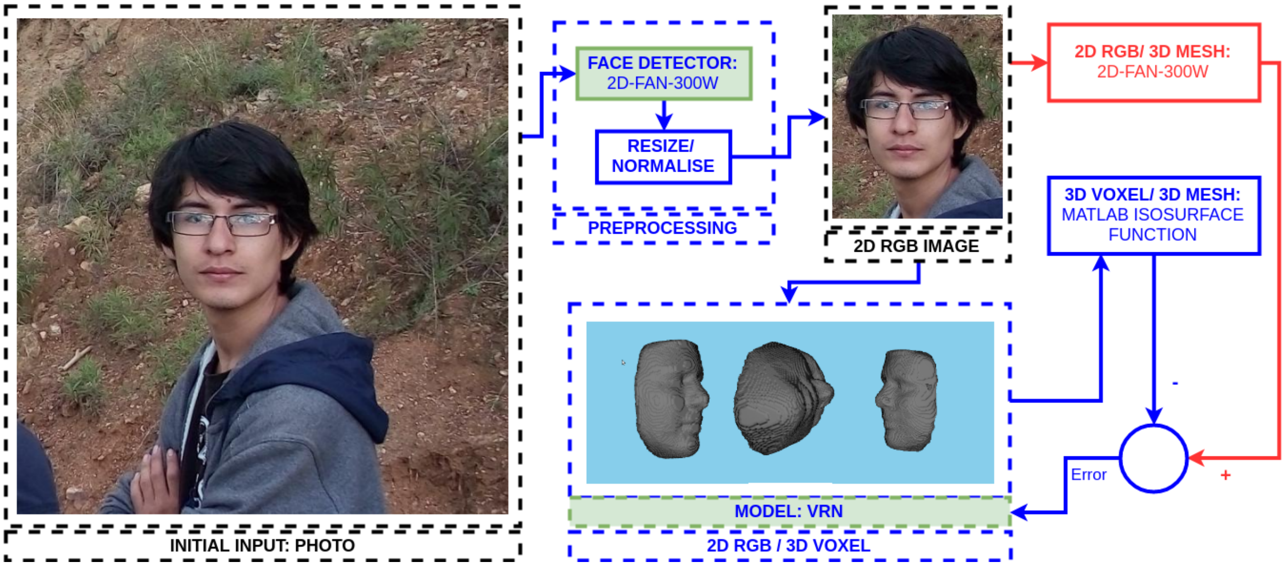

Figure 1: Basic idea of the *Volumetric Regression Network* (VRN) accepts as input an RGB input and directly regresses a 3D volume completely bypassing the fitting of a 3DMM. Each rectangle is a residual module of 256 features.

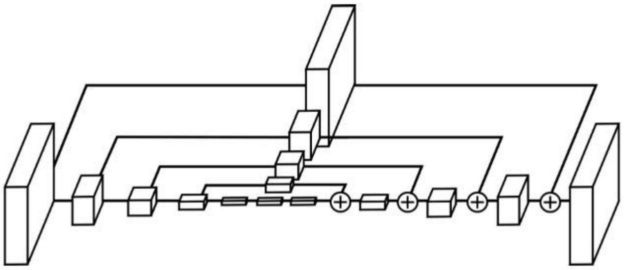

Figure 2: Hourglass module. Each box corresponds to a residual module as seen in Figure 3. The number of features is consistent across the whole hourglass. [Image from [16]]

volutional network of [17], using skip connections and residual learning. The volumetric architecture consists of two hourglass modules which are stacked together without intermediate supervision.

The input is an RGB image and the output is a volume of 192 x 192 x 200 of real values. This architecture is shown in Figure 4 as it can be observed, the network has an encoding/decoding structure where a set of convolutional layers are firstly used to compute a feature representation of fixed dimension. This representation is further processed back to the spatial domain, re-establishing spatial correspondence between the input image and the output volume. Features are hierarchically combined from different resolutions to make per-pixel predictions. The second hourglass is used to refine this output, and has an identical structure to that of the first one.

In training, the 3D image is represented as 3D binary matrix, which means that classification is done using logistic regression, in paper's case is sigmoid cross entropy loss function, which could be represented as the probability to choose a class while no the other. This cost function is presented in equation 1:

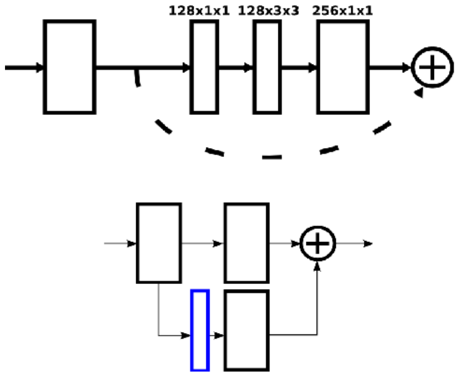

Figure 3: **Up**: Residual Module [8] that we use throughout our network. **Down**: Illustration of the intermediate supervision process. The network splits and produces a set of heatmaps (outlined in blue) where a loss can be applied. A 1x1 convolution remaps the heatmaps to match the number of channels of the intermediate features. These are added together along with the features from the preceding hourglass. [Image from [16]]

$$l_1 = \sum_{w=1}^{W}\sum_{h=1}^{H}\sum_{d=1}^{D} W_{whd} log(W^{'}_{whd}) + (1 - W_{whd}) log(1 - W^{'}_{whd}) \tag{1}$$

Where $W^{'}_{whd}$ is the corresponding sigmoid output at voxel w, d, h of the regressed volume.

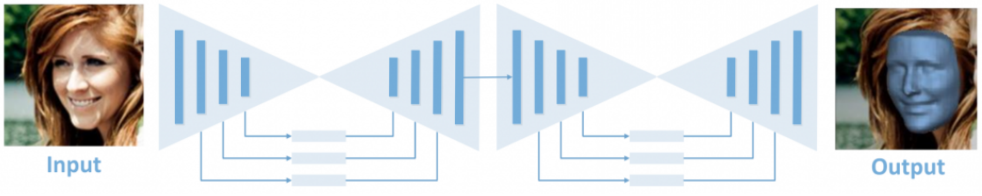

Figure 4: Basic idea of the *Volumetric Regression Network* (VRN) accepts as input an RGB input and directly regresses a 3D volume completely bypassing the fitting of a 3DMM. Each rectangle is a residual module of 256 features. [Image from the original paper]

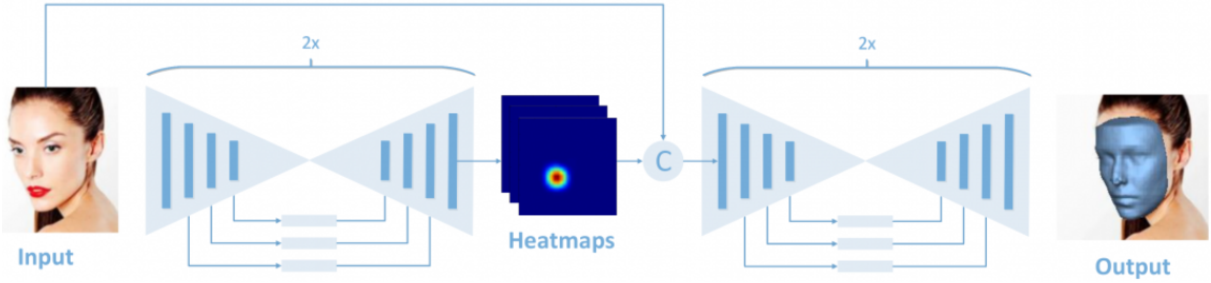

Figure 5: Proposed *VRN - Guided* architecture firsts detects the 2D projection of the 3D landmarks, and stacks these with the original image. This stack is fed into the reconstruction network, which directly regresses the volume. [Image from the original paper]

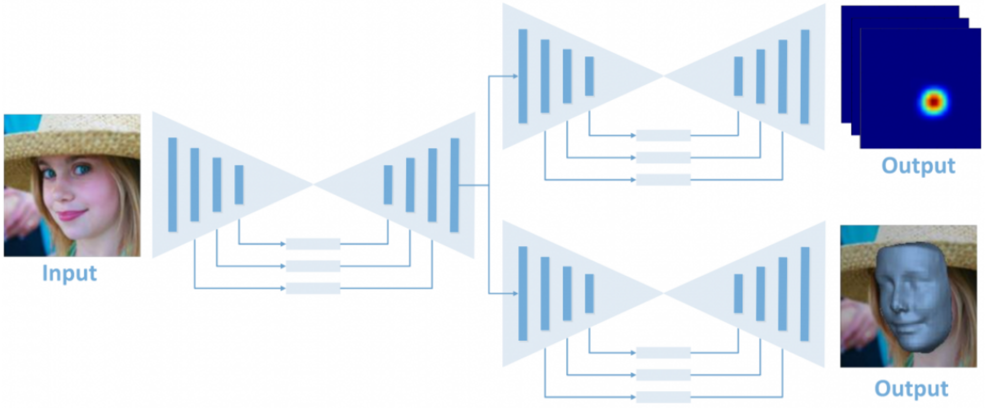

Figure 6: The proposed *VRN - Multitask* architecture regresses both the 3D facial volume and a set of sparse facial landmarks. [Image from the original paper]

Error metric. To measure the accuracy of reconstruction for each face, the Normalised Mean Error (NME) is used defined as the average per vertex Euclidean distance between the estimated and ground truth reconstruction (using 2D/3D mesh transformation model as target) normalised by the outer 3D interocular distance:

$$NME = \frac{1}{N}\sum_{k=1}^{N}\frac{||x_k - y_k||_2}{d} \tag{2}$$

where N is the number of vertices per facial mesh, d is the 3D interocular distance and x_k ,y_k are vertices of the grouthtruth and predicted meshes. The error is calculated on the face region only on approximately 19,000 vertices per facial mesh.

VRN - Guided. The reconstruction should benefit from firstly performing a simpler face analysis task; in particular the paper propose an architecture for volumetric regression guided by facial landmarks. To this end, a stacked hourglass network is trained, which accepts guidance from landmarks during training and inference. This network has a similar architecture to the unguided volumetric regression method, however the input to this architecture is an RGB image stacked with 68 channels, each containing a Gaussian ($\sigma = 1$, approximate diameter of 6 pixels) centred on each of the 68 landmarks. This stacked representation and architecture is demonstrated in Fig. 5. During training we used the ground truth landmarks while during testing we used a stacked hourglass network trained for facial landmark localisation.

VRN - Multitask. A Multitask VRN is proposed, shown in Fig. 5, consisting of three hourglass modules The first hourglass provides features to a fork of two hourglasses. The first of this fork regresses the 68 iBUG landmarks [6] as 2D Gaussians, each on a separate channel. The second hourglass of this fork directly regresses the 3D structure of the face as a volume, as in the aforementioned unguided volumetric regression method. The goal of this multitask network is to learn more reliable features which are better suited to the two tasks.

2.5. Training

Each of the architectures was trained end-to-end using RMSProp with an initial learning rate of 10^{-4} , which was lowered after 40 epochs to 10^{-5} . During training, random augmentation should be applied to each input sample (face image) and its corresponding target (3D volume): we applied in-plane rotation $r \in [-45°, ..., 45°]$, translation $t_z, t_y \in [15, ..., 15]$ and scale $s \in [0.85, ..., 1.15]$ jitter. In 20% of cases, the input and target were flipped horizontally. Finally, the input samples are adjusted with some colour scaling on each RGB channel.

According to the article. For the VRN - Guided case, the landmark detection module was trained to regress Gaussians with standard deviation of approximately 3 pixels ($\sigma = 1$).

3. Implementation Details

Even when the article presents 3 architectures and shows its comparison in efficiency, which gives a clear winner (in section 4). The authors does not leave the source code of the best architecture available instead they delivers a model already trained in Torch, which according to the code documentation, contains only weights from the first architecture (slightly above the average of state of art and of the proposed architectures).

In the code documentation. Full error calculation and expected results showed on the article are on ***MatLab***. A working installation of Torch7 is required (due pre-trained model is included here).

To improve and test our own data. We first fix bumps problem, using a metaheuristic based on dynamic labeling of bodies on 3D voxel space. Bumps problem happens when 3d mesh target has hidden surfaces like cheekbones or cheeks in a profile face. The face detection algorithm for its part has been trained with front faces in which all the characteristics of the face are seen, and although it distinguishes a possible angle of horientacin of each face also generates meshes in areas that do not correspond to the edge of the face , these may be only small errors in 3D mesh that could have vanished in backpropagation but when returned to an autoencoder with residual networks this error remains almost uniform in epochs, that is, the failure of the network could be worked by the side of targets as an input from external algorithm or **can also be worked on a posprocessing algorithm**. In order to test our data and have more code handling, we started using the Torch pre-trained model to obtain voxelized 3D bodies and not in mesh as it was previously, but chancing the rest of the models. To do that, we just have to replace weights in .t7 format with another with same shape (or tensor size). Then we observe results in diferent cases and test error function planted on equation 2.

3.1. Dynamic labeling

To eliminate excess bodies or bumps in a post-processing algorithm, we must first identify all existing bodies, this task is complex and heavy in its voxelized model, however in 3D-mesh model the problem is reduced to analyzing neighborhoods, which requires a series of passes over the entire mesh (which has the fragments properties but not of neighborhood directly) in order to identify sectors that are connected. Finally bodies are eliminated based on the relative size to be left with only one body that is assumed to be the face.

```
Algorithm Object Detection Algorithm (Mesh):
Mesh fragments → graph point.
for each fragment do
  if at least one vertex has a label: then
    Search all labels related to the vertices labels.
    From all labels look for the smallest number → m.
    Previous labels are now connected with m.
    label the vertices with m
  else
    ¡text¿
  end if
  for $F = 1$ to $N_{frames}$ do
    Evaluate density with Iframes[F]
    if Pass evaluation then
      Add frame to OFrames
    end if
  end for
end for
return  Vector of OFrames
```

4. Results

This section presents the results of the paper and also, results from our own compilations (using papers datasets and own 2D images).

The paper performed cross-database experiments only, on 3 different databases, namely AFLW2000-3D, BU-4DFE, and Florence reporting the performance of all the proposed architectures along with the performance of two state-of-the-art methods, namely 3DDFA [2] and EOS [13]. Both methods perform 3DMM fitting (3DDFA uses a CNN), a process completely bypassed by VRN.

Table 1 shows that *VRN - Guided* is better in all datasets with less error values in testing [Equation 2].

Figure 7 present visual results, as was mentioned in section 3, mixing papers results with our own images to processing. Many more images were tested but face detector algorithm [5] rejects them, due they does not meets the minimum requirements for the face recognition.

Method	AFLW2000-3D	BU-4DFE	Florence
VRN	0.0676	0.0600	0.0568
VRN - Multitask	0.0698	0.0625	0.0542
VRN - Guided	**0.0637**	**0.0555**	**0.0509**
3DDFA [2]	0.1012	0.1227	0.0975
EOS [13]	0.0971	0.1560	0.1253

TABLE 1: Reconstruction accuracy on AFLW2000-3D, BU4DFE and Florence in terms of NME. Lower is better.

Acknowledgments

This work was supported by grant 234-2015-FONDECYT (Master Program) from Cienciactiva of the National Council for Science,Technology and Technological Innovation (CONCYTEC-PERU).

References

[1] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In Computer graphics and interactive techniques, 1999.

[2] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li. Face alignment across large poses: A 3d solution. 2016.

[3] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, pages 34763483. IEEE, 2013.

[4] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3d face model for pose and illumination invariant face recognition. In AVSS, 2009.

[5] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou. Faceware-house: A 3d facial expression database for visual computing. IEEE TVCG, 20(3), 2014.

[6] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. A semi-automatic methodology for facial landmark annotation. In CVPR-W, 2013.

[7] Toshev, A., Szegedy, C.: Deeppose: Human pose estimation via deep neural networks. In: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE (2014) 16531660

[8] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. Computer Vision and Pattern Recognition, 2016. CVPR 2016. IEEE Conference on (2015)

[9] Tompson, J.J., Jain, A., LeCun, Y., Bregler, C.: Joint training of a convolutional network and a graphical model for human pose estimation. In: Advances in Neural Information Processing Systems. (2014) 17991807

[10] Newell, Alejandro, Kaiyu Yang, and Jia Deng. ”Stacked hourglass networks for human pose estimation.” European Conference on Computer Vision. Springer, Cham, 2016.

[11] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 19

[12] A. Jourabloo and X. Liu. Large-pose face alignment via cnn-based dense 3d model fitting. In CVPR, 2016.

[13] P. Huber, G. Hu, R. Tena, P. Mortazavian, W. P. Koppen, W. Christmas, M. Ratsch, and J. Kittler. A multiresolution 3d morphable face model and fitting framework.

[14] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In ICCV, 2015.

[15] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In NIPS, 2014.

[16] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In ECCV, 2016.

[17] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In CVPR, 2015.

[18] J. Roth, Y. Tong, and X. Liu. Adaptive 3d face reconstruction from unconstrained photo collections. In CVPR, 2016.

[19] F. Liu, D. Zeng, Q. Zhao, and X. Liu. Joint face alignment and 3d face reconstruction. In ECCV, 2016.

[20] Bulat, Adrian, and Georgios Tzimiropoulos. ”How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks).” International Conference on Computer Vision. Vol. 1. No. 6. 2017.

[21] Peng, Xi, et al. ”Reconstruction-based disentanglement for pose-invariant face recognition.” intervals 20 (2017): 12.

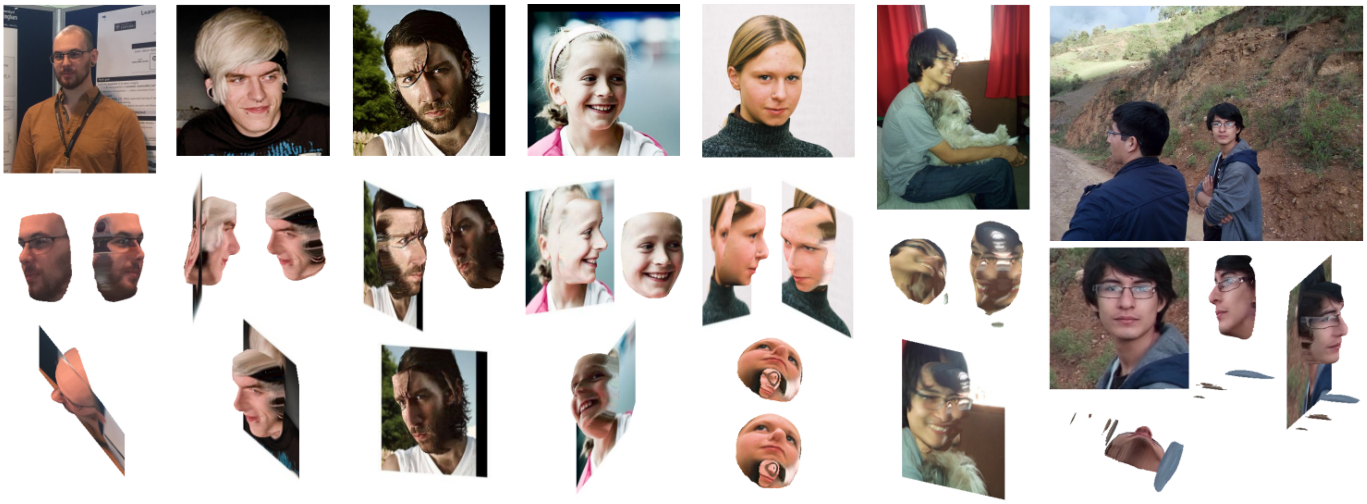

Figure 7: Compilation results with the paper datasets and owns. The first five (left to right) are from the paper. In the fifth image, some texture issues are observed. Then figure 6 and 7, show morphological errors (presence of strange objects from the face).