IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 13, 2019, accepted February 4, 2019, date of publication February 12, 2019, date of current version March 5, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2898705

Face Depth Estimation With Conditional Generative Adversarial Networks

ABDULLAH TAHA ARSLAN AND EROL SEKE
Department of Electrical and Electronics Engineering, Eskisehir Osmangazi University, 26480 Eskisehir, Turkey
Corresponding author: Abdullah Taha Arslan (atarslan@ogu.edu.tr)

ABSTRACT Depth map estimation and 3-D reconstruction from a single or a few face images is an important research field in computer vision. Many approaches have been proposed and developed over the last decade. However, issues like robustness are still to be resolved through additional research. With the advent of the GPU computational methods, convolutional neural networks are being applied to many computer vision problems. Later, conditional generative adversarial networks (CGAN) have attracted attention for its easy adaptation for many picture-to-picture problems. CGANs have been applied for a wide variety of tasks, such as background masking, segmentation, medical image processing, and superresolution. In this work, we developed a GAN-based method for depth map estimation from any given single face image. Many variants of GANs have been tested for the depth estimation task for this work. We conclude that conditional Wasserstein GAN structure offers the most robust approach. We have also compared the method with other two state-of-the-art methods based on deep learning and traditional approaches and experimentally shown that the proposed method offers great opportunities for estimation of face depth maps from face images.

INDEX TERMS 3D face reconstruction, generative adversarial networks, deep learning.

I. INTRODUCTION

Human face depth estimation and 3D reconstruction is an important field of research in computer vision. 3D information provides additional benefits in overcoming hurdles related with 2D images in vision tasks such as detection and recognition, especially under varying pose, illumination, and expression (PIE) [1]. Pose variations, estimating lighting conditions and occlusions are some examples to these problems. However, constructing 3D models or reconstructing from 2D images have been a major challenge for the researchers. Many approaches have been proposed and developed over the last few decades to this end, with each one having its own complications. With the advent of the GPU computational methods, convolutional neural networks are being applied to many computer vision problems. Lately, a specific kind of network structure, called conditional generative adversarial networks, has attracted attention for its easy adaptation for many picture-to-picture problems. Segmentation, superresolution, background masking are some of the examples to these solutions among many. In this work, we developed a CNN-based method for depth estimation from any single face image.

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaochun Cheng.

3D reconstruction and depth estimation techniques developed in the last decades fall into several categories. One prominent one is called *shape-from-shading* [2], [3] falling under other *shape-from-* umbrella techniques utilizing different vision cues, and related approaches developed over the years, e.g. with a statistical approach [4]–[6], with an approach of symmetry [7], with a reference model input and spherical harmonics [8]. Another major category is model based algorithms [9]. Vast amount of work in depth estimation from single or multiple images accumulated over the years can be seen in [10] and in its references section. If more than one image is provided, a separate set of techniques may be implemented, such as photometric stereo [11], where many images of the same scene are needed to be taken under varying lighting conditions; geometry based methods, such as structure from motion where camera and few number of point positions are calculated followed by outlier-detection and intensive bundle-adjustment process [12], [13], and stereo correspondence algorithm [14].

With the advent of convolutional neural networks and GPU based computation approaches, major problems of computer vision field have been adapted to this new realm. While the previous methods always had to get involved with shape and image characteristics such as reflectance, albedo, or distribution of light sources, deep learning methods leave these

details to be learned by the filters located inside the networks. Shape recovery becomes a mere choice of appropriate network structure, optimization technique, loss function, and data set. Also, the previous techniques almost always had to incorporate or develop one (or many) regularization or optimization algorithm(s) in order to impose additional constraints to this ill-posed problem. Again, deep neural networks handle this work internally, or through a parameter in loss function equation.

Lately, a specific deep learning structure, called Generative Adversarial Networks (GANs) [15], [16], has emerged and attracted wide attention. These structures are constructed of two separate networks, generators and discriminators (or called critics) and have been proved to be able to produce realistic images. One variant of GANs is named as Conditional Generative Adversarial Networks (cGAN) [17], [18]. Motivation behind cGANs is to provide a general-purpose solution to image-to-image transformation problems. The difference of a cGAN from an unconditional network is that input images are fed to both discriminator and generator networks. Some of the application areas of cGANs have been background masking, segmentation [19], and interesting implementations such as edges-to-objects [18].

In this work, we propose a solution for depth estimation and reconstructing 3D models from single 2D face images. We utilize conditional generative adversarial networks, and examine variants and training techniques. Following, we also compare the results against the results of other deep learning based and successful traditional techniques. For the former, the proposed technique suggested recently in [20] has been chosen in order to compare Generative Networks with Autoencoders, and for the latter, the SfS-derived method in [8] has been chosen for its efficiency and fast implementation.

II. RELATED WORK

Zhang *et al.* [20] implemented a deep learning approach for learning 3D faces from 2D images. They utilized an autoencoder structure, called Stacked Contractive Autoencoders (SCAE), to learn low-dimensional features of both input images and corresponding 3D models, and connect these two with another network. Therefore, their system consists of three components.

Generative Adversarial Nets (GANs) [15] provided an effective way to train a generative network (*generator*), by constructing a zero-sum game between this network and another one, named *discriminator*, whose objective is to differentiate produced (*fake*) images from real ones. During this training process, the zero sum game between these networks convolve towards a maxmin solution, theoretical Nash equilibrium, where neither of the networks could learn and adapt a little bit more. Radford *et al.* [16] introduced a class of CNNs called deep convolutional generative adversarial networks (DCGANs) for unsupervised learning, and they also outline certain rules and architecture guidelines for stabilizing the GAN training processes.

GANs have been extended to conditional GAN models (cGANs) by feeding extra information to the networks [17], [18]. In [17], one-to-one mappings for output categories are extended to one-to-many labels with the input taken to be the conditioning variable. Isola *et al.* [18] search for a general framework of networks to image-to-image problems. This generalized network architecture produces wide range of translations, such as aerial photos to maps, BW to color, labels to street scenes and building facades and interesting outcomes such as sketches to hand-bags, shoes and even cats. In this architecture, the image producing generator network is a modified version of U-Net [21], which was developed earlier for image segmentation and consists of an encoder-decoder type of shape with addition of some skip layers connecting corresponding resolution blocks.

The proposed framework in [22] is called SAGAN and the authors introduce self-attention to GANs in order to combine features from separated regions of the images. Their network structure is claimed to be the best performer in Inception and FID scores. They also conduct training with separate learning rates and different update steps for generator and discriminator networks, as suggested in [23]. In [24], both the generator and discriminator are grown progressively, adding new layers from low resolutions as the training progresses. Relativistic GANs (RGANs) and Relativistic average GANs(RaGANs) change the behavior of discriminator networks such that they estimate the probability of the given data is more realistic than the fake data [25].

Training GANs have been reported to be not an easy task as it is unstable and very sensitive to parameter choices [26]. The outcome of vanishing or exploding gradients is just one of the problems. In order to improve convergence, many approaches and alternatives of GANs have been proposed in the last couple of years [22], [23], [25], [27]–[33]. Arjovsky *et al.* [27] investigate several distance measures between distributions and propose Earth-Mover(EM) or Wasserstein distance for their GAN structure, named Wasserstein GAN (WGAN) with improved stability. In their algorithm, a new loss metric is presented and claimed to be a better alternative than the original one, also the discriminator network becomes a *critic*. WGAN-GP improves performance of WGAN by introducing a gradient penalty rather than clipping weights as in WGAN [28]. To overcome vanishing gradients, Least Squares GANs (LS-GANs) change the discriminator loss function with a LS approach [29], and claim to be more stable and producing higher quality images. Another generative model is named Stacked GAN and decomposes the training process into multiple layers, consisting of a top-down stack of GANs [30]. It is claimed to be the best performer on Inception Score on CIFAR-10 database [31]. With BEGAN, Berthelot *et al.* propose a new method for balancing the generator and discriminator during training, and also a measure of convergence derived from the Wasserstein distance. They use an auto-encoder as the discriminator network, previously proposed with Energy-Based

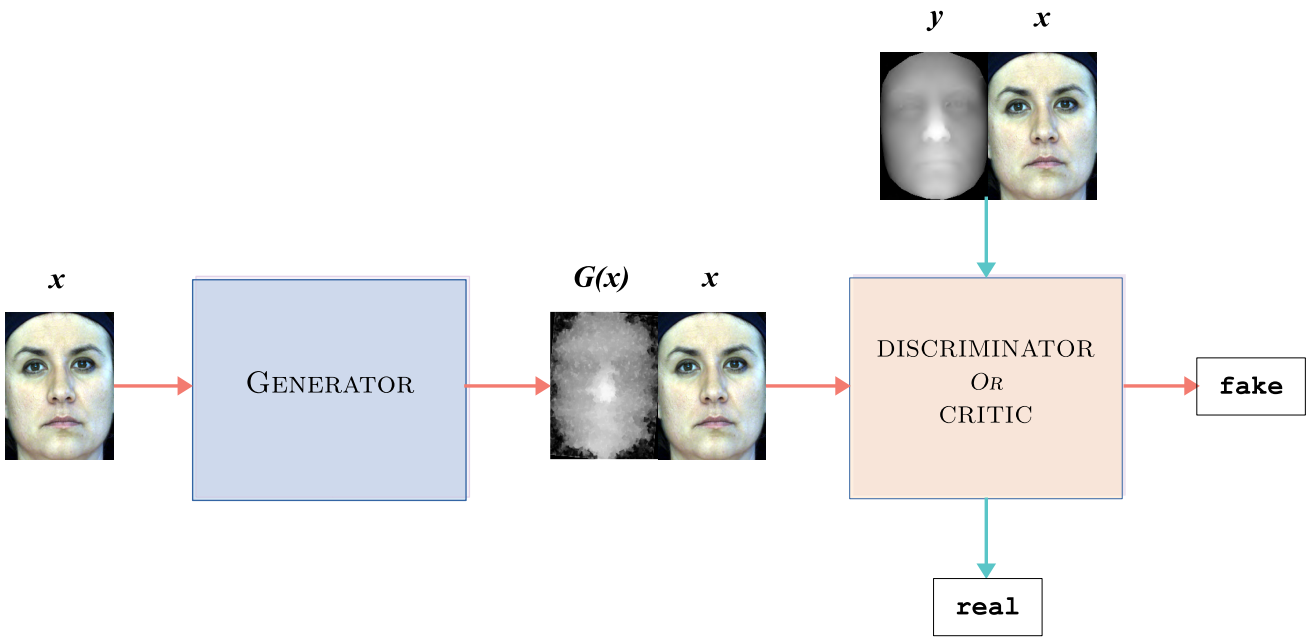

FIGURE 1. General structure of a conditional generative adversarial network with inputs of 2D images and depth maps.

GAN (EBGAN) in [33]. EBGAN model considers the discriminator as an energy function.

The proposed framework in [22] is called SAGAN and the authors introduce self-attention to GANs in order to combine features from separated regions of the images. Their network structure is claimed to be the best performer in Inception and FID scores. They also conduct training with separate learning rates and different update steps for generator and discriminator networks, as suggested in [23]. In [24], both the generator and discriminator are growed progressively, adding new layers from low resolutions as the training progresses. Relativistic GANs (RGANs) and Relativistic average GANs(RaGANs) change the behavior of the discriminator network such that they estimate the probability of the given data to be more realistic than the fake data [25].

Researches from Google Brain conducted a study to measure several of the aforementioned models. They conclude that more important task in training lies in hyperparameter optimization rather than a network structure and many models can reach similar results [34].

III. NETWORK STRUCTURE

While the original Generative Adversarial Network used a noise variable as input [15], conditional GANs (*CGAN*) incorporate input data as a conditioning variable. This conditioning has been applied in many applications previously, such as labels [17], text [35], images and videos [36]–[40], and also in general non application-specific structure [18].

Conditional GAN structure that is going to be implemented for the depth estimation goal can be defined as follows: Let G and D represent two networks, generator and discriminator, respectively. G maps a random Gaussian noise z under the condition of observed image x to depth map d:

$$G : \{x, z\} \rightarrow d$$

In training the generator network, our aim is to maximize the objective function

$$L_G(G, D) = \sum \log D(x, G(x, z)) \tag{1}$$

where G tries to force D to accept generated depth maps as true outputs. At the same time D is trained to discriminate fake maps from real ones, maximizing the objective function:

$$L_D(G, D) = \sum \log D(x, d) + \log(1 - D(x, G(x, z))) \tag{2}$$

The first part of the last equation represent the training with real images to real depth maps, while second part covers the output maps of the generator network, labeled as fake. An additional distance loss term can be added in Equation 1 to prevent the generator from moving too far away from the ground-truth data during the training process. This term can be an L2 distance loss, or an L1 distance loss as suggested in [18]. Figure 1 illustrates the general structure of the approach. The final objective function for the generator can be written as

$$G^* = \arg \min_G \max_D L_G(G, D) + \lambda d_{L1|L2}(G) \tag{3}$$

where L_G is the loss function given in Equation 1. The last term is a $L1$- or a $L2$-norm distance function.

Training GANs have been reported to be problematic for many reasons [23], [32], [41], including non-convergence where the model oscillates or never converges, vanishing or exploding gradients where the discriminator network overwhelms over the generator in the zero-sum game, mode

FIGURE 2. Augmented images for an indiviudal in the dataset. From left to right: a) White-balanced, b) Rotated clockwise, c) Rotated counterclockwise, d) Gaussian blurred, e) Original images.

collapses where the generator network does not learn and generates small number of outputs, and overfitting issues. In general, the models are highly sensitive to *hyperparameters*. A lot of effort has been spent to remedy these problems to train stable and robust networks [26]–[29], [31]. Gradient descent and related algorithms to train deep convolutional neural networks can easily collapse in training GANs where the sought solution is a Nash equilibrium, rather than a minimum.

Lately, Wasserstein GAN structure have been reported to be overcoming some of these hurdles in training GANs [27]. The *Earth-Mover* (EM) or Wasserstein-1 distance is the distance between two probability distributions over a region and can be formulated between two distributions μ and ν as:

$$W_p(\mu, \nu) = inf\, \mathbf{E}[d(X, Y)] \tag{4}$$

where $\mathbf{E}$ is the expected value taken over all joint distributions of the random variables X and Y, and $d(\cdot)$ is the absolute-value distance function. Arjovsky *et al.* [27], in order to tract the infimum in Equation 4, introduce K-Lipschitz continuity constraint, and apply weight clipping to enforce this condition roughly.

In the next section, our experimental setup for depth estimation and several alternative network structures are presented. Conditional GAN and Wasserstein GAN cost functions with the same generator and discriminator networks are compared quantitatively. Although the Wasserstein GAN is an unsupervised learning method, by conditioning the probability distribution with input images, the proposed method becomes a Conditional Wasserstein GAN.

IV. EXPERIMENTS

In this section we introduce our setup for training and testing, as well as the datasets used in the process. Real ground-truth depth maps measured with laser scanners are needed for any dataset that would be utilized in developing depth estimation algorithms. A small percentage of images in any dataset is left aside for testing step conducted after the training process finishes.

A. DATABASES

Two separate 3D face databases have been used in our experiments: The Texas 3D Face Recognition Database [42] and Bosphorus Database for 3D Face Analysis [43]. We will refer to these databases as Database I and II, respectively. Database I consists of 116 individuals, while II has 105 individuals' varying poses and emotional expressions. Neutral and near-neutral posed images from the latter one were consolidated with the former. Ground truth depth maps are provided with the databases. Each 2D portrait image in the databases comes with the depth map showing corresponding depth values for each pixel of the portrait image. Depth maps were normalized to [0,255], where 255 represents the near clipping plane while 0 denotes the background. All input 2D images and output depth maps were scaled to 256×256 resolution and faces (and corresponding depth images) in Database I were zoomed in in order to fill the most of the frame since there are large background regions in the original images. 10 individuals from each database are randomly set aside for testing purposes, and training data set is constructed from the remaining image-depth pairs.

In order to increase variance of the data set at the hand, and improve robustness of the trained networks, augmentation was applied to the training data. The following transformations were applied to randomly selected one-third of the original data set: Gaussian blur, slight rotation clockwise and counterclockwise separately, and histogram equalization. The last one was chosen because of the fact that majority of the original database images were taken under low contrast and dim lighting conditions. Therefore, the total number of images in the data set were increased from 1901 to 4442. A set of augmented images belonging to an individual can be seen in Figure-2. Table 1 summarizes information about the databases.

TABLE 1. Databases.

Database	Number of individuals	Number of images for training	Augmented images
I	116	843	1979
II	105	1058	2463

B. SETUP

With a choice of a generator network, a discriminator network and a loss function among many options, several number of different structures for a GAN can be constructed. Three of them have been tested and are listed in Table 2. Note that configuration I is the same network structure as described in [18], named *pix2pix*. It uses a modified version of U-Net [21]

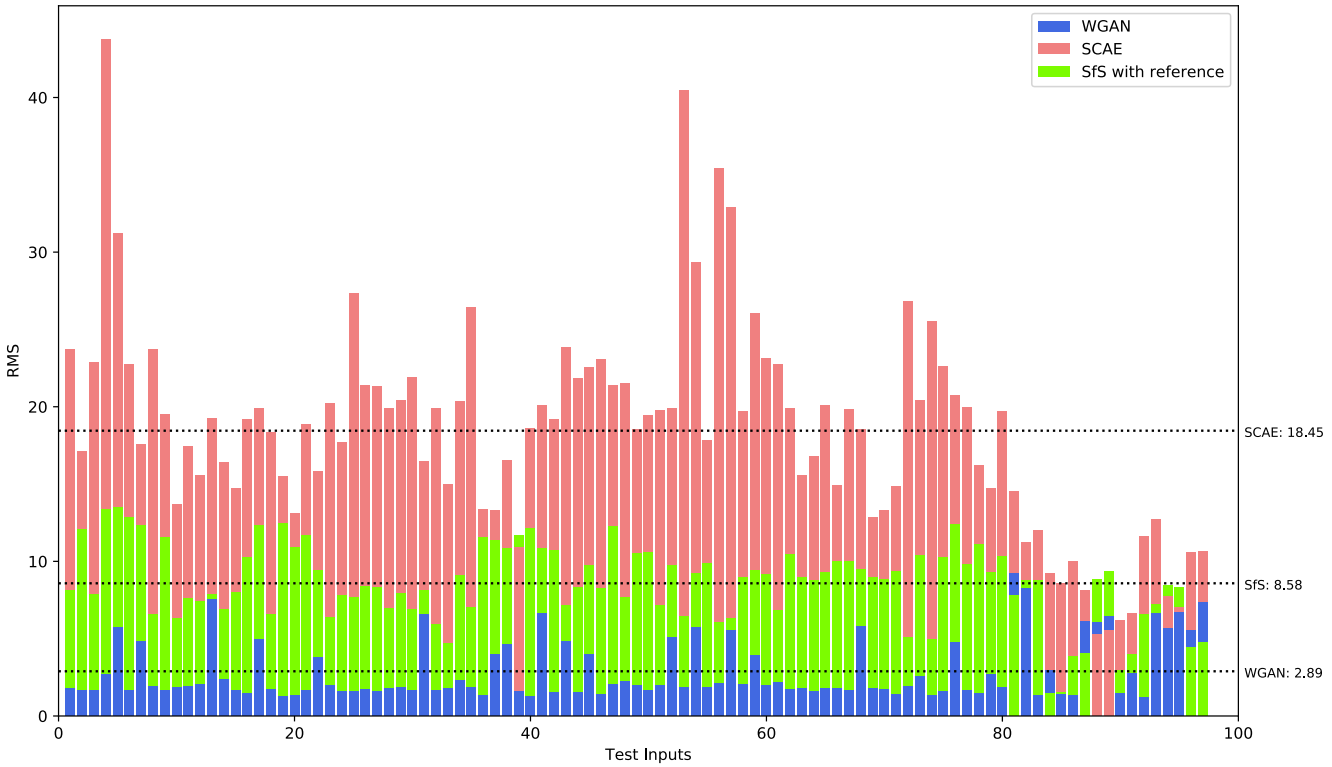

FIGURE 3. RMS error comparisons for a) WGAN, b) SCAE [19] and c) SfS with a reference image [8] on tested 97 face images. Average RMS values are also shown by horizontal lines.

TABLE 2. Configuration options.

Configuration	Generator	Disciriminator	Loss Function
I	U-Net	PatchGAN	CGAN + L1 (L2)
II	U-Net	DCGAN	WGAN + L1 (L2)
III	SHG	DCGAN	WGAN + L1 (L2)

as the generator network. Configuration II has conditional version of Wasserstein GAN loss function with DCGAN [16] as a critic. In configuration III, a different network for the generator developed for pose estimation earlier and named as *Stacked Hourglass* (SHG) [44], is applied. While the details can be seen in [44], implementation in our research omits the first layer of the structure in order to preserve the input image dimensions. Apart from this, a stack of 8-hourglass 1-residual modules as suggested in [44], is applied.

All of the code development has been done with Apache MxNet and PyTorch deep learning frameworks on Ubuntu operating system, and the training were conducted on a single GPU (*NVIDIA Tesla V100*).

In order to evaluate and compare the success of overall reconstruction outputs, pixel-wise average Mean Absolute Error in percentage is calculated for each image in the testing dataset. Then, average for all test images is calculated in a similar way implemented in [8]. Let $h_{out}(x, y)$ and $h_{gt}(x, y)$ be the pixel depth value estimated for an input image and corresponding ground-truth value, respectively.

Then, MAE in percentage can be calculated using:

$$e(x, y) = \left| \frac{h_{out}(x, y) - h_{gt}(x, y)}{255.0} \right| * 100 \tag{5}$$

and overall error rate in percentage for an image using:

$$E_i = \left(\sum_{x,y \in \Omega} e(x, y) \right) \times \frac{1}{m \times n} \tag{6}$$

where the size of the image is $m \times n$.

The test results for various configurations can be seen in Table-3.

TABLE 3. Comparison of depth estimator networks (100 minus average error rate %).

Network	Loss	Success Rate
U-Net / PatchGAN	CGAN+L1	96.210 ±06.397
U-Net / PatchGAN	CGAN+L2	95.705 ±06.588
SHG / DCGAN	WGAN+L1	94.394 ±05.072
U-Net / DCGAN	WGAN+L1	**97.970 ±03.403**

C. CONVERGENCE ISSUES AND LEARNING RATE OPTIMIZATION

GANs have been reported to be very hard to train. In order to optimize the training process, we explored several optimization algorithms including Stochastic Gradient

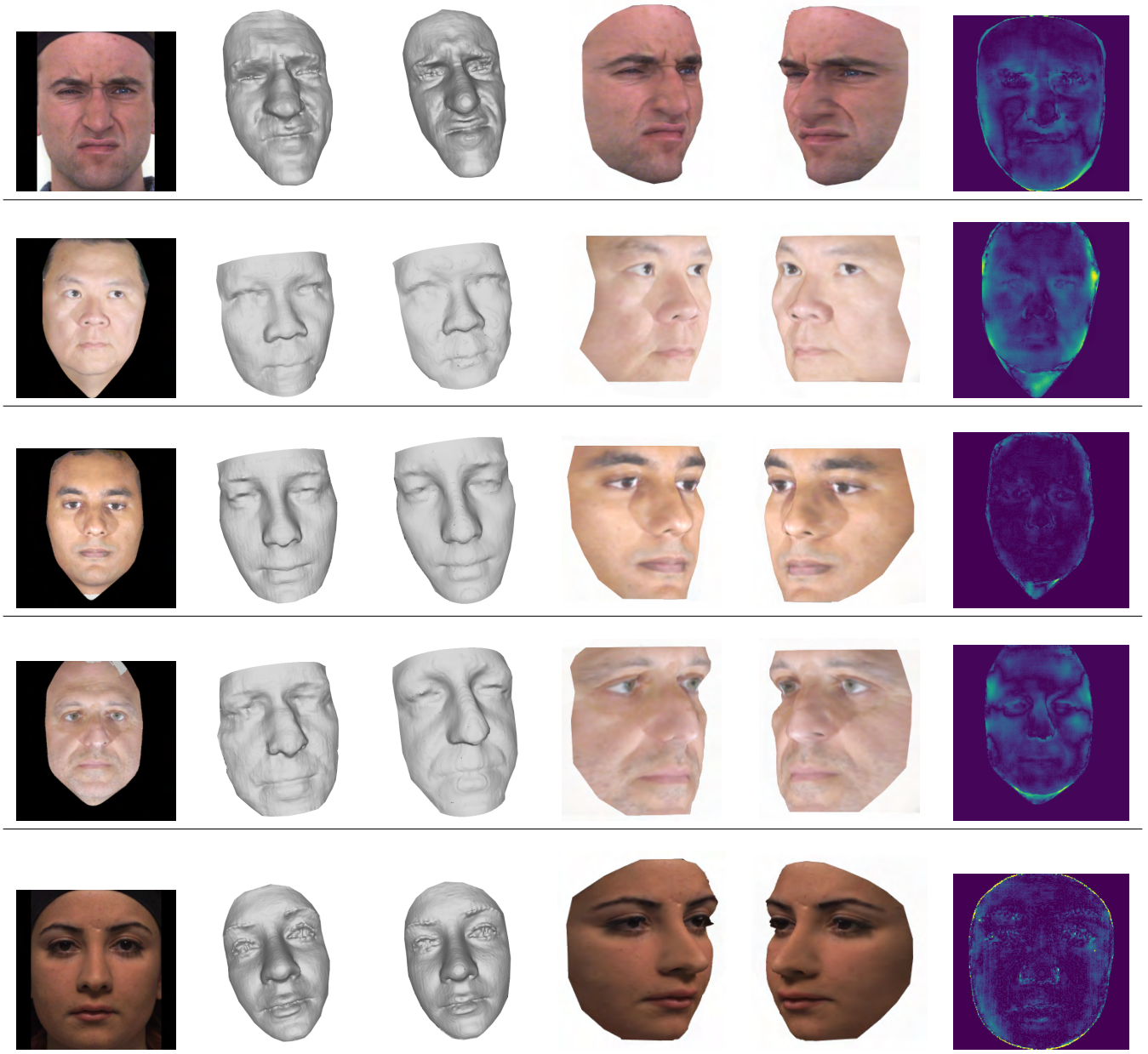

FIGURE 4. Test outputs for the proposed WGAN depth estimation network. Test images come from databases, but they are not included in the training stage. From left to right: a) Input image, b) Test output reconstructed snapshot image, c) Ground truth depth map reconstructed snapshot image, e) and f) Snapshots of reconstructed outputs from two different views with textures mapped, f) Difference heat map image between ground truth and test output depth maps. The pixel-wise error rates from top-to-bottom are %5.385 ± 6.713, %3.108 ± 3.074, %0.649 ± 0.844, %2.307 ± 1.937 and %0.356 ± 0.882.

Descent, RMSProp, Adam and Adamax [45], AdaGrad [46], ADADELTA [47] and Nadam. In overall, Adam algorithm provided the most robust training process for the depth estimation task. A monotonically decreasing learning rate has been found to be a much better alternative to a constant rate in training processes [48], in terms of updating layers' weights with a gradually decreasing learning rate as the training epochs progress. The degree of change for this decreasing should be adjusted carefully. There are several learning rate scheduling functions suggested, including exponential decaying, step or multi-step scheduling, cosine annealing, or learning rate reducing functions when a metric stops improving. In this work, these algorithms have been tested extensively, and a cosine scheduling was observed to be the most effective learning rate adjustment algorithm, where a learning rate is adjusted at just before any optimizer step moment according to the following equation [49]:

$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min})(1 + \cos(\frac{T_{cur}}{T_{max}}\pi)). \quad (7)$$

Here η_{max} is the initial learning rate, η_{min} is the final rate, T_{cur} is the current weight updating step, T_{max} is the total

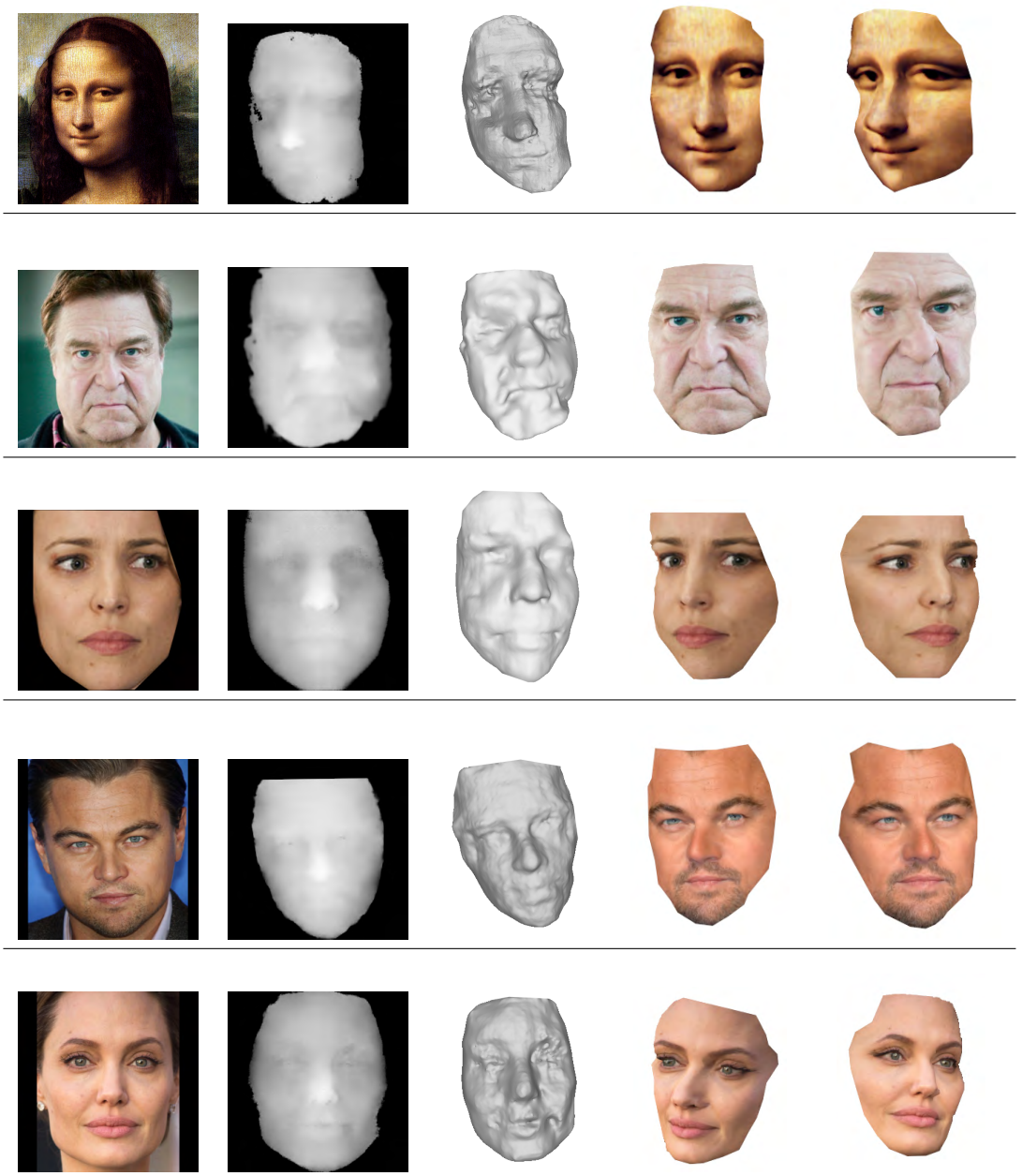

FIGURE 5. Reconstructed outputs of test inputs which were downloaded from internet for visual inspection. Since ground-truth depth-maps are not available for these images, objective performance measurements can not be done. In each row, from left to right, a) input 2D image, b) output depth map, c) surface reconstructed snapshots, d) and e) reconstructed and texture mapped 3D data viewed from two different angles.

number of updating steps. η_{min} is set as 0 in most of the applications, however, in our experiments we have observed that setting η_{min} to a fraction of η_{max} (like 1/20th) yields a better convergence outcome.

V. DISCUSSION

In Table 3, quantitative comparison of methods are given. According to this, the most successful network is a U-Net generator with a DCGAN discriminator, and the loss function is Wasserstein metric with a L1 distance term. In Figure 4, output results for some of the images in the testing group can be seen. Each output depth map has been converted to a point cloud file, and surface reconstruction has been done with MeshLab software [50]. Poisson surface reconstruction approach is applied after point normals are calculated. In order to handle some irregularities and to improve smoothness Laplace surface smoothing algoritm is also run in the next step. It should be noted that there was no particular reason for opting any smoothing algorithm against others here. In addition to the face images in test databases whose

depth values are available, some portrait images from internet were downloaded and processed for visual inspection. These results can be seen in Figure 5. Outputs for test images from databases are found to be better and smoother due to the fact that these images come from the same source with the training images, taken under similar conditions, although test images are never included in the training process.

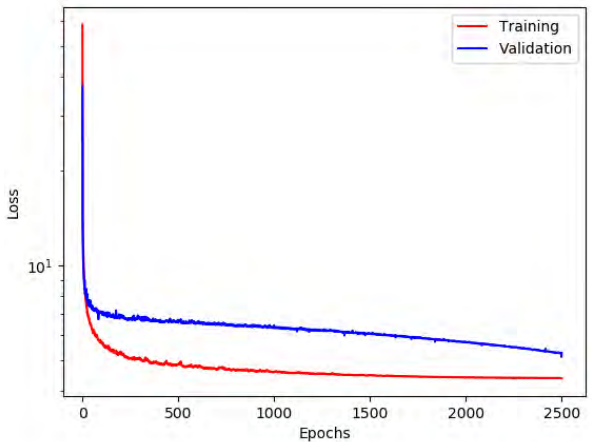

FIGURE 6. In order to investigate any possible over-fitting issues, the training data set were divided into two parts randomly: 80% for training and 20% for validation. The generator network's loss value is plotted in this figure for each epoch over 2500 epochs.

In order to investigate any potential over-fitting issues in training the proposed Generative method, the training set is divided into two parts; training and validation. After each epoch the loss function of the generator network is recorded and the results are given in Figure V. The validation loss is observed to be decreasing monotonically by tracing the training set's curve with a certain amount of difference between the two.

A. COMPARISON WITH AUTOENCODERS

In order to compare GAN structure with other deep networks in terms of depth estimation task, we have implemented the Stacked Contractive Autoencoder (*SCAE*) network, a recent algorithm proposed in [20]. Contractive autoencoders have proved themselves effective in representing input data in lower dimensions. The proposed method's [20] layout is as follows: Initially, a stacked AE structure is trained for 2D input images in order to represent these images in lower dimension. Another similar stacked AE network is also trained for 3D models (point clouds). These two networks are brought together with a third network, consisting of only a dense layer. Given a 2D image, the aim is to predict 3D representation of the input via whole network top-to-bottom. In our experiments, we have followed the researchers' suggestion of 3-layer structure for autoencoders. For comparison reasons we resized images from both databases to 70x70 pixel gray-scale images. Without any convolutional layers, it is obvious that training with original images at larger resolutions with only linear layers would be a gigantic workload (For example, a layer of 10,000 neurons with a 256x256x3 input will result in approximately 2 billion weights in a single layer!) The exact layer structure of the SCAE is 4900-500-100-10 and 14900-1000-100-10 for 2D images and 3D points, respectively.

Following the generation of depth maps, RMS errors [20] are calculated using

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\tilde{x}_{3D}(i) - x_{3D}(i))^2}. \quad (8)$$

RMS results of SCAE and the generative methods can be comparatively seen in Figure 3. The superiority of generative models is very obvious. We believe that there are mainly two reasons for lower than expected performance of SCAE. 1) The research conducted in [20] uses synthesized face models and images of perfect design for the task at hand. When it comes to real images with real textures, albedos and shading the proposed method results in poor outcomes. 2) Autoencoders naturally act as lossy compressors. At each layer down the pipeline, a considerable amount of detail information of the input is naturally eliminated. We have observed this phenomena by reconstructing the back propagated images, decoding at each layer's output and comparing with the original image. Although at [20, Table 2] the reconstruction error increases significantly starting with layer number 4, in our experience after the third layer a sudden drop of detail information occurs due to diminishing number of features. This might be something to look for when distinguishing human faces against other objects, but certainly not desirable for separating one face from another. By contrast, U-Net network [21] with its skip layers is a robust structure for reconstruction of large and real images (256×256 vs. 70×70).

B. COMPARISON WITH SFS METHODS

We have also implemented the shape from shading algorithm described in [8] in order to compare generative models with one of the last state-of-the-art methods of *pre* deep-learning era. In this work the authors conduct depth estimation with an 2D image as input as usual, but also use another average image of all individuals in the dataset, of which depth information is known. They also utilize spherical harmonics expansion to capture the lighting component of the Lambertian reflectance model. The algorithm pipeline is a three step procedure as follows: 1) Four lighting coefficients are estimated with a linear expansion utilizing reference image's depth values and albedo map, 2) Pixel-wise depth values are estimated via a sparse matrix solution of a large set of linear equations with regularization terms using the lighting coefficients estimated in step 1 and reference image's albedo map, 3) Albedos are estimated in a similar fashion with the step 2, this time using estimated values from step 1 and 2. For the reference input image, we took average of all images in both databases separately, and used this reference 2D image and corresponding depth map for test images coming from that database. For estimating reference albedo

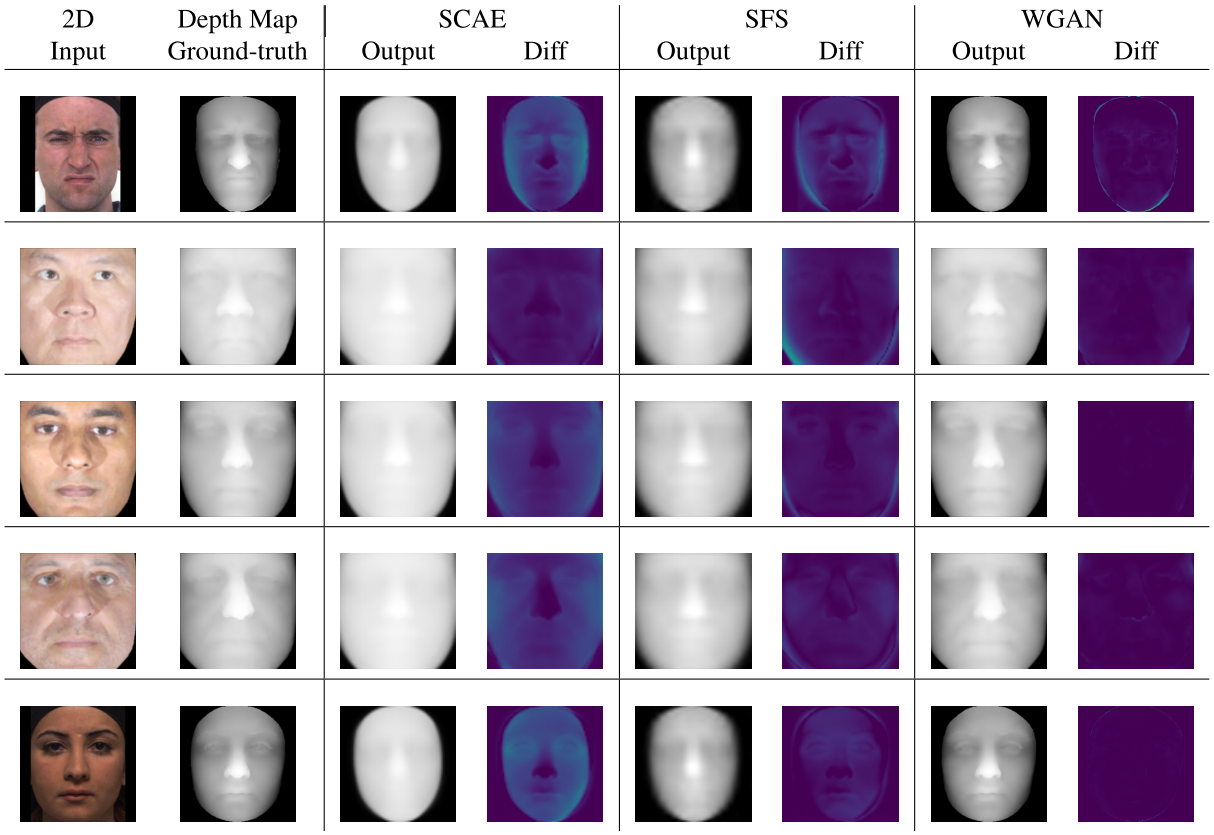

FIGURE 7. Comparison of the three methods discussed in Section V. The test images come from the test data set and they are also the same input images presented in Figure 4. The RMSE errors of three methods for each row is shown above the difference images. as: 1- (32.52, 12.20, 5.10); 2- (31.71, 2.66, 3.10); 3- (23.82, 11.46, 1.43); 4- (27.61, 11.33, 2.19); 5- (33.75, 10.09, 4.62).

map, the point light source estimation algorithm in [51] was deployed and the unknown albedo map for the reference inputs were obtained. The results of this overall procedure can be seen in Figure 3. Although this is a very fast algorithm not requiring any training process, the generative models still perform much better in general in the depth estimation task.

VI. CONCLUSION

In this work, we have implemented a generative adversarial network solution for depth estimation of 2D images for 3D reconstruction. This is an ill-posed, however a well researched problem for which many algorithms and approaches have been proposed over the decades. We have constructed several GAN structures and concluded that Wasserstein GAN is a robust and well-performing solution for the task. We have also compared this method with two other methods, one of which is another deep network approach based on autoencoders [20], and the other one is a variant of a conventional SfS algorithm based on spherical harmonics expansion [8]. After rigorous tests of all of the mentioned methods, we conclude that WGAN approach outperforms other methods in depth estimation of a face from a single 2D image. Although this is an exciting and promising approach, generative models are found to be very difficult to be trained fully and future work is needed to optimize the methods and develop new algorithms to circumvent possible convergence issues. Traditional approaches, such as 3D Morphable Models [9], may be combined with Deep Learning techniques, especially with Generative Networks in order to facilitate the complex process of depth estimation with new computational opportunities.

REFERENCES

[1] V. Blanz, S. Romdhani, and T. Vetter, "Face identification across different poses and illuminations with a 3D morphable model," in *Proc. 5th IEEE Int. Conf. Autom. Face Gesture Recognit.*, May 2002, pp. 202–207.
[2] B. K. P. Horn, *Obtaining Shape from Shading Information*. Cambridge, MA, USA: MIT Press, 1989.
[3] R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah, "Shape-from-shading: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 8, pp. 690–706, Aug. 1999.
[4] J. J. Atick, P. A. Griffin, and A. N. Redlich, "Statistical approach to shape from shading: Reconstruction of three-dimensional face surfaces from single two-dimensional images," *Neural Comput.*, vol. 8, no. 6, pp. 1321–1340, Aug. 1996.
[5] R. Dovgard and R. Basri, "Statistical symmetric shape from shading for 3D structure recovery of faces," in *Computer Vision—ECCV*. Berlin, Germany: Springer, 2004, pp. 99–113.
[6] W. A. P. Smith and E. R. Hancock, "Recovering facial shape using a statistical model of surface normal direction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 1914–1930, Dec. 2006.
[7] W. Y. Zhao and R. Chellappa, "Symmetric shape-from-shading using self-ratio image," *Int. J. Comput. Vis.*, vol. 45, no. 1, pp. 55–75, Oct. 2001.
[8] I. Kemelmacher-Shlizerman and R. Basri, "3D face reconstruction from a single image using a single reference face shape," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 2, pp. 394–405, Feb. 2011.
[9] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," in *Proc. 26th Annu. Conf. Comput. Graph. Interact. Techn.*, 1999, pp. 187–194.
[10] R. Szeliski, *Computer Vision: Algorithms and Applications*. Springer, 2010.

[11] A. S. Georghiades, P. N. Belhumeur, and D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 643–660, Jun. 2001.
[12] M. I. Lourakis and A. A. Argyros, "SBA: A software package for generic sparse bundle adjustment," *ACM Trans. Math. Softw.*, vol. 36, no. 1, pp. 1–30, Mar. 2009.
[13] Y. Shan, Z. Liu, and Z. Zhang, "Model-based bundle adjustment with application to face modeling," in *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 2, Jul. 2001, pp. 644–651.
[14] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Comput. Vis.*, vol. 47, nos. 1–3, pp. 7–42, Apr. 2002.
[15] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
[16] A. Radford, L. Metz, and S. Chintala. (2015). "Unsupervised representation learning with deep convolutional generative adversarial networks." [Online]. Available: https://arxiv.org/abs/1511.06434
[17] M. Mirza and S. Osindero. (2014). "Conditional generative adversarial nets." [Online]. Available: https://arxiv.org/abs/1411.1784
[18] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. (2017). "Image-to-image translation with conditional adversarial networks." [Online]. Available: https://arxiv.org/abs/1611.07004
[19] Y. Li and L. Shen, "cC-GAN: A robust transfer-learning framework for HEp-2 specimen image segmentation," *IEEE Access*, vol. 6, pp. 14048–14058, 2018.
[20] J. Zhang, K. Li, Y. Liang, and N. Li, "Learning 3D faces from 2D images via stacked contractive autoencoder," *Neurocomputing*, vol. 257, pp. 67–78, Sep. 2017.
[21] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, Nov. 2015, pp. 234–241.
[22] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena. (2018). "Self-attention generative adversarial networks." [Online]. Available: https://arxiv.org/abs/1805.08318
[23] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local nash equilibrium," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6626–6637.
[24] T. Karras, T. Aila, S. Laine, and J. Lehtinen. (2017). "Progressive growing of GANs for improved quality, stability, and variation." [Online]. Available: https://arxiv.org/abs/1710.10196
[25] A. Jolicoeur-Martineau. (2018). "The relativistic discriminator: A key element missing from standard GAN." [Online]. Available: https://arxiv.org/abs/1807.00734
[26] M. Arjovsky and L. Bottou. (2017). "Towards principled methods for training generative adversarial networks." [Online]. Available: https://arxiv.org/abs/1701.04862
[27] M. Arjovsky, S. Chintala, and L. Bottou. (2017). "Wasserstein GAN." [Online]. Available: https://arxiv.org/abs/1701.07875
[28] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5767–5777.
[29] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2813–2821.
[30] X. Huang, Y. Li, O. Poursaeed, J. Hopcroft, and S. Belongie, "Stacked generative adversarial networks," in *Proc. CVPR*, vol. 2, Jul. 2017, pp. 5077–5086.
[31] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2234–2242.
[32] D. Berthelot, T. Schumm, and L. Metz. (2017). "Began: Boundary equilibrium generative adversarial networks." [Online]. Available: https://arxiv.org/abs/1703.10717
[33] J. Zhao, M. Mathieu, and Y. LeCun. (2016). "Energy-based generative adversarial network." [Online]. Available: https://arxiv.org/abs/1609.03126
[34] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet. (2017). "Are gans created equal? A large-scale study." [Online]. Available: https://arxiv.org/abs/1711.10337
[35] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. (2016). "Generative adversarial text to image synthesis." [Online]. Available: https://arxiv.org/abs/1605.05396
[36] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2536–2544.
[37] X. Wang and A. Gupta, "Generative image modeling using style and structure adversarial networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Sep. 2016, pp. 318–335.
[38] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, "Generative visual manipulation on the natural image manifold," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 597–613.
[39] M. Mathieu, C. Couprie, and Y. LeCun. (2015). "Deep multi-scale video prediction beyond mean square error." [Online]. Available: https://arxiv.org/abs/1511.05440
[40] C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Sep. 2016, pp. 702–716.
[41] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein. (2016). "Unrolled generative adversarial networks." [Online]. Available: https://arxiv.org/abs/1611.02163
[42] S. Gupta, K. R. Castleman, M. K. Markey, and A. C. Bovik, "Texas 3D face recognition database," in *Proc. IEEE Southwest Symp. Image Anal. Interpretation (SSIAI)*, May 2010, pp. 97–100.
[43] A. Savran, N. Alyüz, H. Dibeklioğlu, O. Çeliktutan, B. Gökberk, B. Sankur, and L. Akarun, "Bosphorus database for 3D face analysis," in *Proc. Eur. Workshop Biometrics Identity Manage.* Berlin, Germany: Springer, 2008, pp. 47–56.
[44] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proc. Eur. Conf. Comput. Vis.* Springer, Sep. 2016, pp. 483–499.
[45] D. P. Kingma and J. Ba. (2014). "Adam: A method for stochastic optimization." [Online]. Available: https://arxiv.org/abs/1412.6980
[46] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Jul. 2011.
[47] M. D. Zeiler. (2012). "Adadelta: An adaptive learning rate method." [Online]. Available: https://arxiv.org/abs/1212.5701
[48] Tensorflow Documentation, *Exponential Decay*. Accessed: Nov. 17, 2018. [Online]. Available: https://www.tensorflow.org/api_docs/python/tf/train/exponential_decay
[49] PyTorch Documentation, *How to Adjust Learning Rate*. Accessed: Nov. 17, 2018. [Online]. Available: https://pytorch.org/docs/stable/optim.html#how-to-adjust-learning-rate
[50] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia, "MeshLab: An open-source mesh processing tool," in *Proc. Eurograph. Italian Chapter Conf.*, V. Scarano, R. D. Chiara, and U. Erra, Eds. Salerno, Italy: The Eurographics Association, 2008.
[51] M. J. Brooks and B. K. Horn, "Shape and source from shading," in *Proc. 9th Int. Joint Conf. Artif. Intell.*, Burlington, MA, USA, Morgan Kaufmann, Aug. 1985, pp. 932–936.

ABDULLAH TAHA ARSLAN received the B.S. degree in electronics and communications engineering from Istanbul Technical University and the MBA degree from California State University, East Bay. He is currently pursuing the Ph.D. degree with the Department of Electrical and Electronics Engineering, Eskisehir Osmangazi University, Turkey. He is also a Co-Founder of a technology research and development firm, specialized in software development, computer vision, and medical imaging.

EROL SEKE received the B.S. degree in electrical and electronics engineering from Anadolu University, Turkey, in 1985, the M.S. degree in electrical and electronics engineering from FBE, Anadolu University, in 1987, and the Ph.D. degree in electrical and electronics engineering from Lehigh University, USA, in 1995. He is experienced in programming and digital design with HDL. He is currently with the Department of Electrical And Electronics Engineering, Eskisehir Osmangazi University, Turkey. His research interests include communications, image processing (denoising and superresolution), and compression.