PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
UNIVERSITY MOHAMED BOUDIAF - M'SILA

FACULTY : Mathematics And Computer Science **DEPARTMENT : Computer Science**
DOMAIN: Mathematics And Computer Science **BRANCH : Computer Science**

N°:.....…..... **OPTION: SIGL & RTIC**

A Thesis Submitted To Obtain the Master Degree

Prepared by: ALI ATALLAOUI & IDRIS MADJIDI

SUBJECT

Generating 3D Avatar from 2D images

Supervised by:

- **Dr. ABDESSATTAR GHEMOUGUI**
- **Dr. MOHAMED BENOUIS**

Academic year: 2020/2021

ACKNOWLEDGEMENT

In the name of Allah, the Most Gracious and the Most Merciful,And prayers and peace be upon the holy Prophet Mohammed.

Firstly, we would like to thank God for giving us the power to complete this work, and because it illuminated our path until we achieved the goal that we aspired to.

We faced a lot of difficulties and pressures, , But despite all that, with the help of God first, and with the help of our supervisor ***Dr. Mohamed Benouis****, who offered us the opportunity to work on this project and had enough patience to review and correct our work over and over. So, thank you very much for your guidance, understanding and patience, it was pleasure to work with you. It has been our great pleasure and honor to have a supervisor like you.*

Deepest gratitude is also due to ***Dr. Ghemougi****, our supervisor. Thank you for everything.*

Thank you to our parents who have been supportive during our educational career for many years. They always believed in us despite everything.

Last but not least, deepest thanks go to all people who took part in making this thesis real.

Ali Atallaoui Idris Madjidi

DEDCATION

The sake of Allah, our Creator and our Master, our great teacher and messenger, Mohammed (MayAllah blessand grant him), who taught us the purpose of life.

Our ***Mothers*** *and* ***Fathers*** *for their love, endless support and encouragement; to our beloved brothers****(«Atallaoui Yahya****,* ***Atallaoui brahim****,* ***Atallaoui Amar****,* ***Atallaoui Hamza****,* ***Atallaoui Youcef» ; «Madjidi Abdelhak , Madjidi Ilyes»),*** *and sisters****(«Atallaoui Fatna»; « Madjidi Douaa, Madjidi Kenza»)****; and to our friends And in particular "* ***Dokmane Mohamed Anes , Ossama Zroug, Bisker Hamada , Koudier Zazzgad, Mazouz Mohamed , chnaikher Mohamed , Islam Boukraa, Redhoune Boutchicha , Thamer Lkhdari ,*** *and also we don't forget our family scout badr specifcly* ***Said Dokman*** *And All The leader in our Scout and for everyone who knows* ***Ali*** *and* ***Idris****.*

Our colleagues and profs in the faculty of Mathematics and Computer Science and the university ***MOHAMED BOUDIAF – M'SILA****.*

Table of Content

List of Figures and Tables

List of figures:

List of tables:

Table of equations:

General introduction

GENERAL INTRODUCTION

Deep learning methods have attracted many researchers in the computer vision field to solve computer vision problems such as image segmentation and object recognition. This success also pointed to the implementation of deep learning techniques in 3D reconstruction. Thus, is considered as classical problem in computer vision in which has been tackled by many techniques.

Computer vision is a field of artificial intelligence that trains computers to interpret and understand the visual world. Using digital images from cameras and videos and deep learning models, machines can accurately identify and classify objects — and then react to what they "see."

Early experiments in computer vision took place in the 1950s, using some of the first neural networks to detect the edges of an object and to sort simple objects into categories like circles and squares. In the 1970s, the first commercial use of computer vision interpreted typed or handwritten text using optical character recognition. This advancement was used to interpret written text for the blind.

A variety of stereo matching techniques have been developed to reconstruct high quality 3D models from two or more images. However, stereo is just one of the many potential cues that can be used to infer shape from images, which include not only visual cues such as shading and focus, but also techniques for merging multiple range or depth images into 3D models, as well as techniques for reconstructing specialized models, such as heads, bodies, or architecture. In addition, this kind of this application required a huge of computation algorithms and high of resource of hardware as CPU and GPU modules. To deal with this issues, deep learning techniques is coming to handle the 3D image vision requirements.

To deal with this issue, our thesis aims to investigate on the promising of deep learning algorithms to produce an efficient and realistic face reconstruction and generating a 3D avatar from a 2D image. Meanwhile, we address many core boundaries of the state of the art in facial reconstruction and editing and focus on facial reconstruction. Moreover, we study the state-of-the-art deep learning-based approaches for 3d reconstruction that mainly were carrying out to learn a specific parameter from a single image enabled to produce a 3D image with low error reconstruction as well low consuming time.

Motivation

High-quality face reconstruction and editing frameworks have facilitated a professional level of visual effects on 3D face models in the post-production method of filmmaking. Modern

improvements that enhance the usability of face editing technologies have also led non-expert users to create digital content in home video and mobile applications with much less effort. In the traditional computer graphics pipeline, face modeling and editing are produced through various stages (i.e., face reconstruction, editing, and compositing). Firstly, the face reconstruction step recovers the high-quality 3D face models of actors in professional studios. In the second step, trained artists add visual effects to the 3D face model by re-writing the facial appearance and geometry. Finally, the adjusted 3D face model is distributed back to the image frame by compositing with other layers such as background and illumination.

Statement of the problem

Over the last years, with the birth of Deep learning algorithms, image vision applications for 2D image have been successfully performed in real time application, for example identifying a face of human being among a million of face "in-the-wild" or producing a new image that is not exited in the real world. However, the deep learning algorithms in 3D image is still remaining an open issue: (a) how to generate the ground truth data (i.e., poverty of available data for training) and (b) Ho to find a suitable general loss function that can effectively englobe all the parameters used to generate 3D image within low error reconstruction

Report outline

Our work was divided into three chapters:

The First chapter: This chapter provides an overview of the technical background of the thesis. Each section discusses the details about an advanced camera model with an optical lens, face-specific geometric models represented; we took a deeper dive into 3D reconstruction from image requirements and the algorithm used to converse 2D to 3D, – the main mathematical tool for the chapter.

The Second chapter: In this chapter, we provided a thorough overview of the fundamental of deep learning and deep neural networks. We have discussed the CNN architecture by providing a different architecture and clarifying their limits. We have also spoken their weaknesses, e.g., getting stuck in the local minima, overfitting, and training time for large problem sets. We mentioned a several state-of-the-art algorithms to overcome these challenges with different optimization methods.

The Third chapter: tackling other deep learning architectures, and more specifically paying attention on 3D Reconstruction how it does incorporate in the convolutional neural network.

CHAPTER 1:
Study Background

1. Introduction

Generating a 3D model from 2D images that is realistic as possible is one of the primary issues of image-based modeling and computer vision. The best option is an automatic reconstruction of the scene with short or no user intervention. Currently, there are numerous methods of 3D reconstruction from 2D images; each algorithm has its conditions of execution, its strengths as well as its weak points.

The 3D reconstruction problem in computer vision is to get the 3D information of a target object from its multiple 2D images. It is a commonly ill-posed problem. As a result, although intensive researching works have been done in this area, 3D reconstruction is still an open problem.

The 3D reconstruction problem normally contains a series of steps including camera calibration, image matching, and reconstruction. There are specific problems in every step; however, the results of the early step will also have a strong impact on the ones in the late stages. For example, the entire reconstruction system accuracy performance is relying on the step with the worst error. This ends in the concept of thinking about the stairs jointly. Moreover, maximum of current photo matching and reconstruction strategies paintings on 2D snapshots due to the fact they're the most effective statistics available. However, this method could have trouble coping with the versions.

2. Camera Model

The camera is defined with the aid of using a fashionable projective pinhole camera model. A 3D point in area X is mapped to the point at the image plane wherein a line joining.

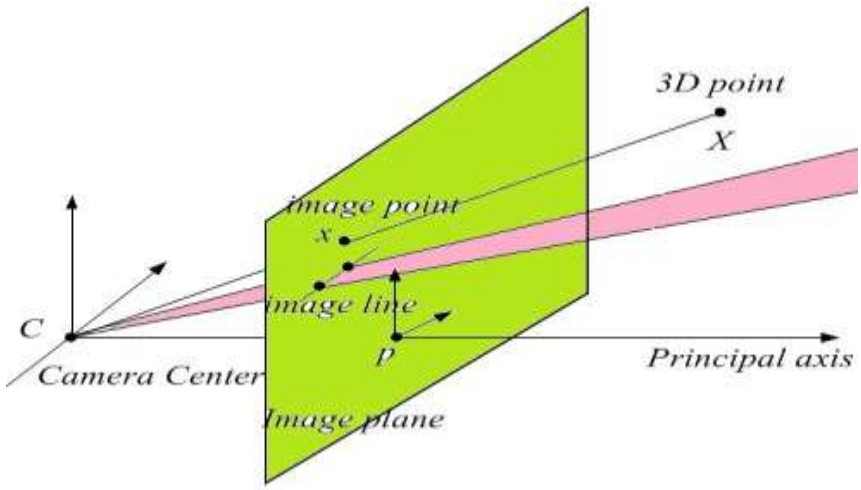


Figure 1: C is the camera center and p the important point. The digital digicam center is positioned on the coordinate origin. Note that the photo aircraft is positioned in the front of the digital digicam center

The point X to the center of the projection meets the image plane, as proven in Fig. 1. The center of projection is referred to as the camera center. The camera center is located on the coordinate foundation and the image plane is located in the front of the camera. The line from the camera center perpendicular to the image plane is referred to as the important axis or important ray of the camera, and the factor in which the important axis meets the image plane is referred to as the important factor. As proven in Fig. 1, p is the important factor.

The perspective projection may be interpreted through the usage of projective geometry. Identifying factors alongside a ray via the projection center approach the 3D international may be interpreted because the projective area P3, wherein the image plane is interpreted because the projective aircraft P2.

Suppose a 3D point.

$$X = \left[A_x\, A_y\, A_z\right]^T \tag{1}$$

Is found at

$$x = [u\ v]^T \tag{2}$$

At the image plane of a camera of focal period f. Then, assuming the imaging geometry as depicted in Figure 1, in which the arena coordinate machine-starting place coincides with the center of the camera of projection, the arena axes are aligned with the image plane and the

camera faces alongside the bad intensity axis. The angle projection equations for photo formation are given by:

$$\frac{u}{A_x} = \frac{v}{A_y} = \frac{-f}{A_z} \tag{3}$$

3. Face Reconstruction

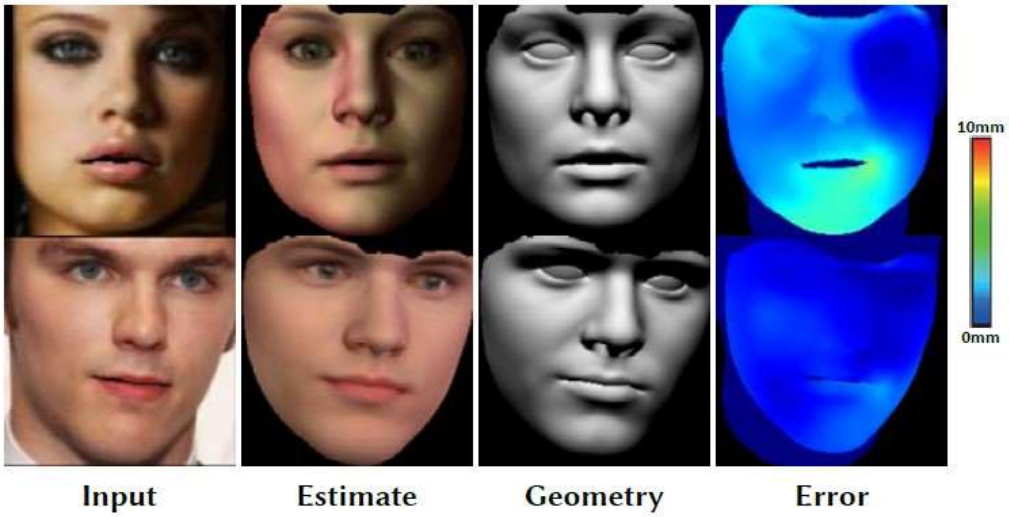


Figure 2 : The single-shot deep inverse face renderer InverseFaceNet obtains a high-quality geometry, reflectance and illumination estimate from just a single input image. InverseFaceNet jointly recover the facial pose, shape, expression, reflectance and incident scene illumination.

The literature on reconstructing face geometry, often with appearance, but without any illumination, is much more extensive compared to inverse rendering. We focus on single-view techniques and do not further discuss multi-view or multi-image approaches [83; 84; 85; 86; 87]. Recent techniques approach monocular face reconstruction by fitting active appearance models [88; 89], blend shape models [90; 91; 92; 93], affine face models [94; 95; 96; 97; 98; 99; 100; 101], mesh geometry [102; 103; 104; 105; 106], or volumetric geometry [107] to input images or videos. Shading-based surface refinement can extract even fine-scale geometric surface detail [108; 109; 110; 111; 112; 113]. Many techniques use facial landmark detectors for more robustness to changes in the head pose and expression,

4. Projective Geometry and Homogeneous Coordinates

Projective geometry is an essential device for representing SfM troubles in computer vision. In projective geometry, the picture formation procedure seems as a projective transformation from a

3D to a 2D projective area. Homogeneous coordinates offer a mathematical approach for computations and theorem proofs in projective geometry. Points in the n-dimensional projective area are represented via way of means of n + 1 issue column vectors; for example:

$$X = \left[A_x\, A_y\, A_z\, A_w\right]^T \tag{4}$$

Is the homogeneous illustration of an arbitrary factor X in 3D space? A one-to-one correspondence exists among the factors below Euclidean coordinates and the homogeneous coordinates of projective geometry. When

$$A_w \neq 0, \left[A_x\, A_y\, A_z\, A_w\right]^T \tag{5}$$

Are the homogeneous coordinates for the Euclidean 3D point?

$$\left[A_{xE}\, A_{yE}\, A_{zE}\,\right]^T \tag{6}$$

Where in the connection among Euclidean coordinates and the homogeneous coordinates is:

$$\left[A_{xE}\, A_{yE}\, A_{zE}\,\right]^T \sim \left[\frac{A_x}{A_w} = \frac{A_y}{A_w} = \frac{A_z}{A_w}\right]^T ; A_{xE} = \frac{A_x}{A_w}, A_{yE} = \frac{A_y}{A_w}, A_{zE} = \frac{A_z}{A_w} \tag{7}$$

As proven in (7), the 3 axes of Ax, Ay and Aware supplied for brevity to demonstrate the 4D vector.

$$X = [A_x\, A_y\, A_z\, A_w]^T \tag{8}$$

The department with the aid of using Aw shows that the conversion of a homogeneous factor to its Euclidean equal is inherently a projection of the homogenous factor onto the Aw = 1 plane; a factor at infinity may be represented with the aid of using Aw = zero in homogeneous coordinates. Furthermore

$$\left[A_x\, A_y\, A_z\, 0\right]^T \tag{9}$$

And

$$\left[-A_x - A_y - A_z\, 0\right]^T \tag{10}$$

Constitute the identical factor at infinity.

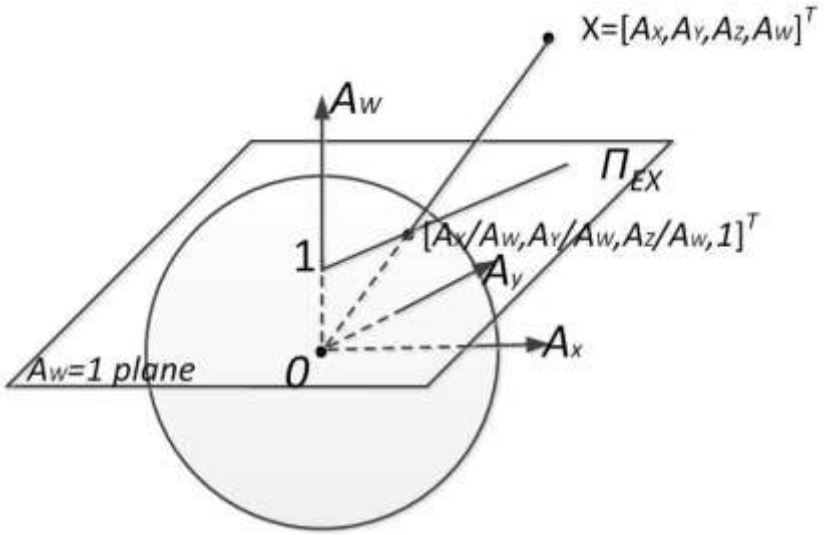


Figure 3: The conversion of a homogeneous factor to its Euclidean equal is inherently a projection of the homogeneous factor onto Aw = 1 plane, in which Aw = zero is the countless factor.[1]

If factors are, signal reversed and Aw ≠ 0, the point

$$\left[\frac{-A_x}{-A_w} = \frac{-A_y}{-A_w} = \frac{-A_z}{-A_w}\right]^T \tag{11}$$

Is equal to the point

$$\left[\frac{A_x}{A_w} = \frac{A_y}{A_w} = \frac{A_z}{A_w}\right]^T \tag{12}$$

Wrapped around infinity getting back from the other direction [2]. The point

$$\left[-A_x\ -A_y\ -A_z\ -A_w\right]^T \tag{13}$$

Does now no longer present

$$\left[A_x\ A_y\ A_z\ A_w\right]^T \tag{14}$$

Which is known as the antipode of X. In this thesis, the sign: is used to indicate the antipode, accordingly, the antipode of X is supplied as X

4.1. Facial Geometry

Facial geometry is formed of shape (also referred to as identity) and expression. To efficiently model the variations of each component, it is represented by a linear combination of an average

face model and displacement vectors mathematically. We render backgrounds about parametric face representation and blend shapes that are used to model facial geometry in the thesis.

- **Parametric Face Representation:** We realize facial shapes with a low-dimensional parametric model [114]. In this model, the geometric deformation of a 3D face model is produced over an affine model $v \in R^{3N}$ that stacks the per-vertex deformations of the underlying template mesh with N vertices, as follows:

$$v(\theta^{[s]} = a^{[g]} + \sum_{k=1}^{N_s} \theta_k^{[s]} b_k^{[s]}. \quad (15)$$

Here, $\boldsymbol{a}^{[g]} \in R^{3N}$ and $\{ b_k^{[s]} \}_{k=1}^{N_s}$ represent the average facial geometry and the basis that is computed by applying principal part analysis (PCA) to 200 high-quality face scans, respectively.

- **Blendshapes:** We select a blend shape model [119] to describe facial expression, which has been widely used in the 3D facial animation literature due to its flexibility in face editing. Each blend shape is a static face geometry that refers to a semantically meaningful deformation such as blink, smile, and frown. To create in-between facial expressions and animation, these deformations are linearly blended with the weights that represent the strength of each deformation. Consequently, the blend shape model provides intuitive control of facial expression, allowing part-based 3D face editing.

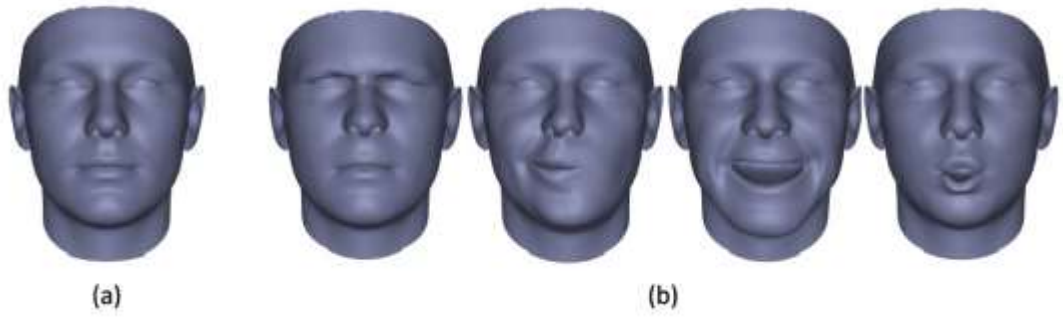


Figure 4: Example of a blendshape model. (a) Neutral face. (b) Semantic shapes. From left to right: Frown, mouth to the right, smile and "O"-like mouth shape. Images from

Mathematically, the blend shape model is expressed with additive shape deformations on top of neutral face geometry, as Figure 4 shows. Let $a_0^{[e]}$ be the neutral face and $B = \{\boldsymbol{b}_1^{[e]}, \dots \boldsymbol{b}_n^{[e]}\}$ be the set of n blend shapes. Here $\boldsymbol{b}_1^{[e]} \in R^{3k}, \forall i$ represents column vectors

of k vertices of 3D face geometry. Facial expression e is then represented by a linear combination of the neutral face shape and its per-vertex 3D displacements to each blend shape:

$$e = a_0^{[e]} + \sum_{i=1}^{n} \theta_i^{[e]} \left(b_i^{[e]} - b_0^{[e]} \right) = b_0^{[e]} + \sum_{i=1}^{n} \theta_i^{[e]} d_i^{[e]} = b_0^{[e]} + B\theta^{[e]} \quad (16)$$

Where $0 \leq \theta_i^{[e]} \leq 1, \forall i = 1{:}n$ denote the linear weights. With $\theta^{[e]} = \left[\theta_1^{[e]}, \ldots, \theta_n^{[e]}\right]^T \in R^n$ and $B = \left[d_1^{[e]} | \ldots | d_n^{[e]}\right] \in R^{3k \times n}$ that are a stack of the linear weights and per-vertex 3D displacements respectively, the linear combination can be expressed in the matrix form.

5. What is 3d Reconstruction from images

3D reconstruction from more than one picture is the advent of 3D models from a hard and fast of pictures. It is the opposite technique of acquiring 2D pictures from 3D scenes.

A photo does now no longer provides us sufficient facts to reconstruct a 3D scene. This is because of the character of the photo forming technique that includes projecting a three-dimensional scene onto a two-dimensional photo. During this technique, the intensity is lost. The factors seen in the pictures are the projections of the actual factors in the image.

6. Fields of Application of 3D Reconstruction from Images

3D reconstruction has packages in lots of fields. Such as: Medicine, Free-perspective video reconstruction, robot mapping, town planning, Gaming, digital environments, and digital tourism, landslide stock mapping, robotic navigation, archaeology, augmented reality, opposite engineering, movement capture, Gesture reputation, and hand tracking.

In medicine, 3D reconstruction from 2D pics may be used for each healing and diagnostic function by using the camera to take more than one pics at more than one angle. Even if there are clinical imaging strategies like MRI and CT scanning, they are nonetheless pricey and might result in excessive radiation doses, which is a danger for sufferers with certain diseases, and they are now no longer appropriate for sufferers with ferromagnetic metal implants. Both the strategies may be performed best while the affected person is in a mendacity role wherein the worldwide

shape of the bone changes. There are a few strategies of 3D reconstruction like Stereo Corresponding Point-Based Technique or Non-Stereo corresponding contour method (NCSS) which may be finished at the same time as status and require low radiation dose via way of means of the use of X-ray pics. [3]

In the sector of robot navigation, a self-sufficient robotic can use the pics taken via way of means of its camera to create a 3D map of its surroundings and use it to carry out real-time processing on the way to locate its manner or to keep away from barriers which could rise up at any second in its path, in addition to making measurements on the distance wherein it's miles located. [4].

The estimation of landslides size is a widespread undertaking at the same time as getting ready the landslide stock map, for which satellite TV for pc aerial/statistics images is required, which could be very pricey. An opportunity is the usage of drones for such mapping. The result of 3D reconstruction from 2D pics could be very correct and offers the opportunity of measurements as much as cm stage or even small items can be effortlessly identified. By the use of pics taking via way of means of a drone in aggregate with 3D scene reconstruction algorithms, we will offer powerful and bendy gear to display and map landslide. [5].

7. 3D Reconstruction from Images Requirements

To obtain, as desired, the coordinates of the factors of the scene, it is miles important to clear up a sure variety of problems:

7.1. Calibration trouble

Calibration trouble or how the factors of the scene are projected at the image. For this, the pinhole version is used and its miles then vital to understand so-referred to as intrinsic parameters of the camera (focal length, middle of the image ...). Then, it's miles vital to understand the relative role of the cameras for you to decide the coordinates of the factors of the distance in a reference of the scene now no longer related to the camera. These parameters, referred to as extrinsic, are the placement and orientation of the camera in space.

7.2. Matching problem

Is the capacity to understand and accomplice the factors that seem on numerous pictures..

7.3. The density of the reconstruction

Once the coordinates of a positive quantity of factors in the area were obtained, it's far important to discover the floor to which those factors belong to acquire a mesh, a dense model. Otherwise, in a few cases, whilst we acquire a huge quantity of factors, the cloud of factors shaped is sufficient to visually outline the form of the item however the reconstruction is then sparse.

8. Active and Passive Reconstruction Methods

The depths recovery method of seen factors in the picture may be accomplished through lively or passive methods.

8.1. Active methods

In order to accumulate an intensity map, energetic strategies actively intervene with the item to be reconstructed thru radiometric or mechanical techniques (laser rangefinder, based on mild and different energetic detection techniques). For example, an intensity map may be reconstructed the usage of an intensity gauge to degree the intensity relative to an item positioned on a rotating plate or the usage of radiometric strategies thru transferring mild sources, colored seen mild, time-of-flight lasers, microwaves, or ultrasounds that emit radiance in the direction of the item after which degree its pondered part.

8.2. Passive methods

Passive techniques of 3D reconstruction do now no longer intrude with items to be rebuilt; they best use a sensor to degree the luminance pondered or emitted via way of means of the floor of the item with the intention to deduce its 3D shape via photograph processing. The sensor used with inside the camera is a photosensor in touchy to seen light. The enter factors for this method are a hard and fast of virtual images (one, or more) or video. For this case, we are telling approximately photograph-primarily based totally reconstruction and the output detail is a 3D model [6].

9. 2D TO 3D CONVERTION ALGORITHMS

Quantity of entering pix, we will categorize the present conversion algorithms into groups: algorithms primarily based totally on or extra pix and algorithms primarily based totally on an unmarried nonetheless image. In the primary case, the 2 or extra enter pix might be taken through a couple of constant cameras positioned both at exclusive viewing angles or through an unmarried camera with shifting gadgets withinside the scenes. We name the intensity cues utilized by the primary organization the multi-ocular intensity cues. The 2nd organization of intensity cues operates on an unmarried nonetheless image, and they're called the monocular intensity cues.

Number of Input Images	Depth Cues
Two or More Images	Binocular disparity
	Motion parallax
	Image blur
	Silhouette
	Structure from motion
One Single Image	Linear perspective
	Atmosphere scattering
	Shape from shading

Table 1: Depth cue used in 2D to 3D conversion algorithms.

9.1. Binocular disparity

By using the photos of the equal scene captured from barely exceptional factors of view, we are able to manipulate to get better the intensity of a factor gift on the 2 photos. First, a corresponding set of factors in each photo are found. Then, with the use of the technique of triangulation, we are able to get to decide the intensity of a factor at the photos [7].

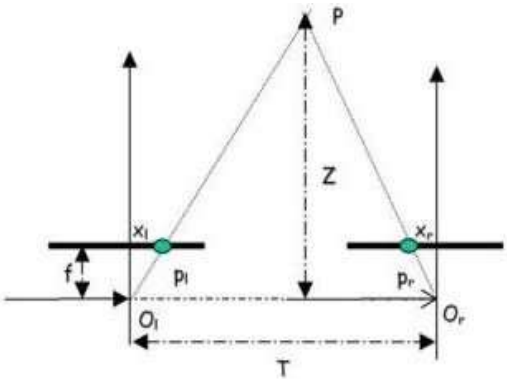


Figure 5: Binocular disparity. [8]

We anticipate that Pl and Pr are the 2 projections of the factors P on the 2 pics and O1 and Or are the origins of the coordinate structures of the 2 cameras. Based on the connection among the triangles (P P1 pr) and (P O1 Or) the intensity Z of the factor P may be acquired in which $D = xr - xl$.

$$z = f\frac{T}{D} \tag{17}$$

9.2. Motion Parallax

The relative movement between the camera and the scene affords essential clues with inside the notion of intensity. Objects, which can be near the camera, circulate quicker than the items, which can be near. The extraction of 3D structure and the camera is named as shape from movement. The movement may be visible as a shape of disparity over time, represented with the aid of using the idea of moving subject. The movement subject is the 2D pace vectors of the picture factors and the determined scene. The fundamental assumptions for shape-from-movement are that do not deform object and their actions are linear. This truth has been exploited in numerous applications, inclusive of wiggle stereoscopy [9] in which movement parallax is used as a metaphor for stereoscopic images or parallax scrolling [10] utilized in video games in which, with the aid of using transferring foreground and history at special speeds, an intensity sensation is evoked. The energy of this cue is particularly excessive while in comparison to other monocular cues and additionally while in comparison to binocular disparity.

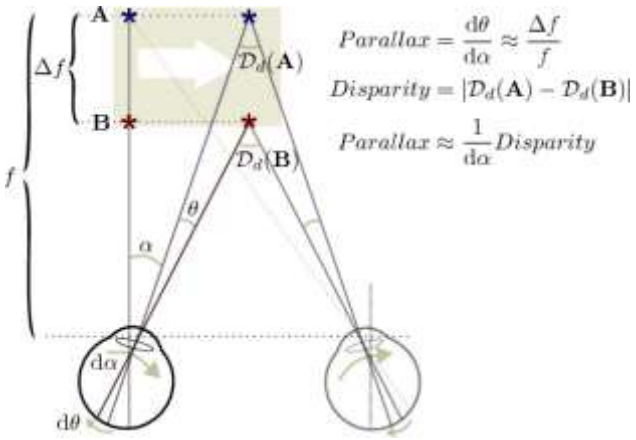


Figure 6: Motion Parallax Mechanics.

9.3. Image Blur

Evidence for using photograph blur intensive belief has been said with the aid of using Mather [11] and with the aid of using Marshall et al [12]. Their papers defined experiments on ambiguous figure-floor stimuli, containing areas of texture separated with the aid of using a wavy boundary. Objects, which are in-consciousness, are certainly pictured at the same time, as gadgets at different distances are defocused.

The following preferred expression relates the space of a factor from a lens to the radius s of its blurred photograph (Pentland 1987): Where F is focal length, r is lens aperture radius, and v is the space of the image plane from the lens. If we recognize the values of F, r, and v and a degree of picture blur are available, and then absolute distance may be calculated. Eq. (1) may be used to expect retinal blur as a feature of distance, on assuming ordinary values for the optical parameters of the human eye (r = 1.five mm, v = sixteen mm).

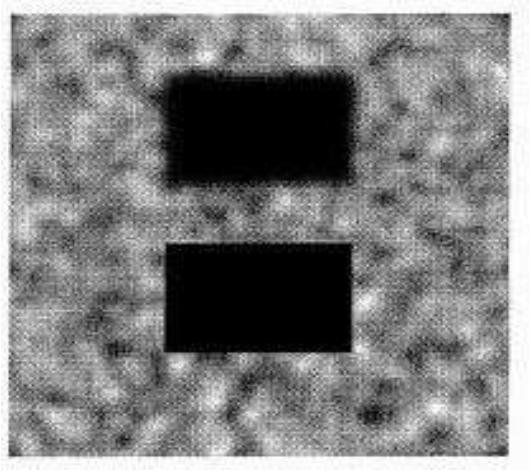


Figure 7: Blur as a depth cue in random patterns.

The higher rectangle is greater some distance than the decrease black rectangle due to the fact the higher rectangle has sharply described edges.

9.4. Silhouette

A silhouette of an item in a picture refers back to the contour keeping apart the item from the heritage. Shape-from-silhouette techniques require more than one perspective of the scene taken via way of means of cameras from exceptional viewpoints. Such a technique collectively with correct texturing generates a complete 3D version of the gadgets inside the scene, permitting visitors to study a stay scene from an arbitrary viewpoint. Shape-from-silhouette calls for correct camera calibration.

For every picture, the silhouette of the goal gadgets has segmented the use of heritage subtraction. The retrieved silhouettes are returned projected to a not unusual place 3D area with projection facilities same to the camera locations. Back-projecting a silhouette produces a cone-like volume. The intersection of all of the cones paperwork the visible hull of the goal 3D item, that is regularly processed inside the voxel representation. This 3D reconstruction technique is known as shape-from-silhouette [13].

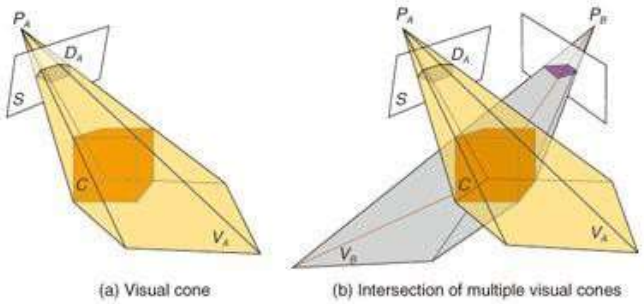


Figure 8: Silhouette volume intersection.

C denotes a cube, which is an instance of a 3D object, S denotes a 2D screen, PA denotes a perspective in 3D space, DA denotes a 2D polygon at the screen, that is the silhouette of the cube, and VA denotes the visible cone back-projected from the perspective PA. PB, DB, and VB denote the corresponding meanings to PA, DA, and VA. [14].

9.5. Linear Perspective

Linear attitude refers to parallel traces together with roads or pathways that converge with distance. The factors of the line of those traces are much less seen than the ones of the closest ones. The technique proposed through Battiato, Curti, and al. [15] works for photos containing surfaces with inflexible geometry. The intersection with the maximum intersection factors with inside the community is taken into consideration to be the vanishing factor. The essential traces near the vanishing factor are marked because of the vanishing traces. Between every pair of neighboring vanishing traces, a fixed of gradient planes are assigned, every similar to an unmarried intensity level. The pixels towards the vanishing factors are assigned a bigger intensity cost and the density of the gradient planes is higher [16].

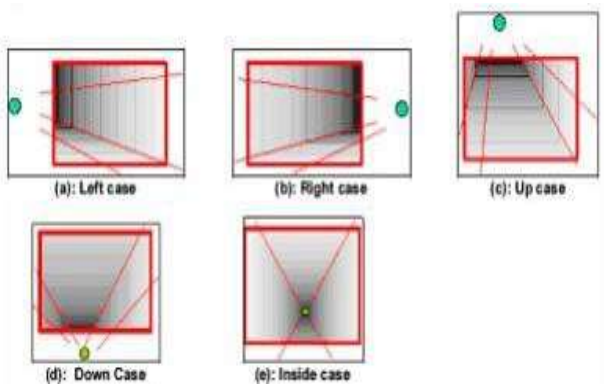


Figure 9: Blur as a depth cue in random patterns. [17]

9.6. Atmosphere Scattering

The atmosphere scattering method is primarily based totally on the truth that the strength and route of mild are modified whilst the mild passes via the surroundings due to small particles found in it. The items, which might be near the camera, seem clearer whilst the ones close to be extra blurred. [15].In 1997, Krotkov and Cosman and [18] supplied an evaluation of this conversion. It became primarily based totally on Lord Rayleigh's 1871 bodily scattering model. Their set of rules is appropriate for estimating the intensity of out of doors snapshots containing a part of the sky.

Figure 10: Depth map from atmosphere scattering. [14]

9.7. Shape from Shading

Shape from shading is a method that provides an understanding of the surface normal of an object by knowing the reflectance of the light on this object. The amount of light reflected by the surface of the object depends on the orientation of the object. The quantity of mild pondered via way of means of the floor of the item relies upon the orientation of the item. Woodham delivered this approach in 1980. When the information is an unmarried image, we name it to

form from shading, and it became analyzed via way of means of B. K. Horn P. In 1989. Photometric stereo has in view that been generalized to many different situations, like non-Lambertian surface finishes and prolonged mild sources.

Multiple photos of an item beneath neath one-of-a-kind lights are analyzed to provide a predicted everyday course at every pixel [19].

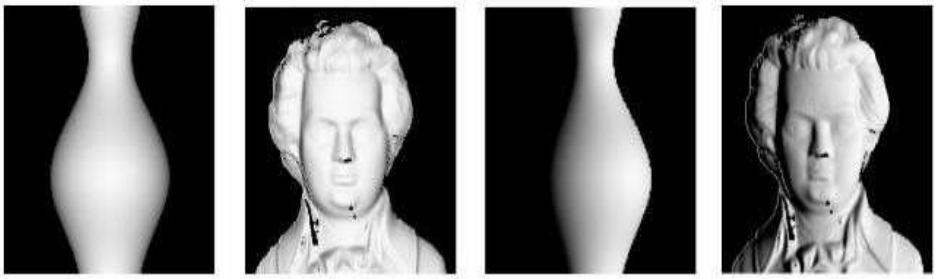


Figure 11: Explanatory figure of shape from shading [20].

9.8. Structure from Motion

Structure from Motion (SfM) is a way that makes use of a chain of dimensional pictures of a scene or item to reconstruct its 3D shape. SfM can produce 3D fashions primarily based totally on excessive-decision factor clouds.

SfM is primarily based totally on the identical ideas as stereoscopic photogrammetry. In stereo photogrammetry, triangulation is used to calculate the relative 3D positions (x, y, z,) of items from stereo pairs. Traditionally those strategies require highly priced specialized systems and software.

To create a 3D reconstruction one virtually desires many pictures of a place or an object with an excessive diploma overlap, taken from distinctive angles. The camera does not want to be specialized, well-known consumer-grade cameras paintings properly for SfM methods. The pictures are regularly be taken from a shifting sensor or via way of means of one or more than one humans at distinctive places and angles. SfM entails the three major steps:

- Step 1: Match corresponding capabilities and degree distances among them at the camera image plane d, d'. Scale Invariant Feature Transform (SIFT) [21] al- lows corresponding capabilities to be matched despite massive versions in scale and standpoint and below situations of partial occlusion and converting illumination.

- Step 2: When we have the matching places of more than one factor on or greater pics, there's generally simply one mathematical answer for wherein the pics had been taken. Therefore, we will calculate character camera positions (x, y, z), (x', y', z'), orientations i,

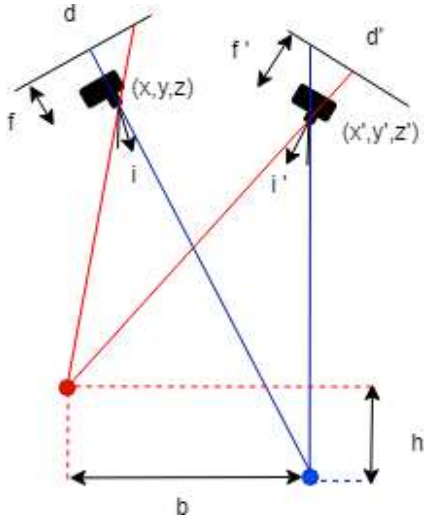


Figure 12: Explanatory figure of structure from motion.

i', focal lengths f, f', and relative positions of corresponding capabilities b, h, in an unmarried step regarded as "package deal adjustment". This is wherein the period Structure from movement comes from. Scene shape refers to these types of parameters; movement refers to the motion of the camera.

- Step 3: Next, a dense factor cloud and 3D floor are decided by the use of the camera parameters and the use of the SfM factors. This step is called "multiview stereo matching" (VMS)

9.9. Passive methods

Passive techniques of 3D reconstruction do now no longer intervene with gadgets to be rebuilt; they most effectively use a sensor to degree the luminance pondered or emitted with the aid of using the floor of the item for you to deduce its 3D shape via picture processing. The sensor used within side the camera is a picture sensor in touchy to seen light. The enter factors for this technique are fixed of virtual images (one, or more) or video. For this case, we are speaking approximately picture primarily based reconstruction and the output detail is a 3D model [6].

10.CONCLUSION

A huge range of 2D to 3D conversion algorithms are devoted to the healing of 3D form from an item in a scene. Each of those algorithms has its very own requirements. These algorithms may be higher utilized in distinctive domain names inclusive of monitoring, robotic navigation, etc. No intensity cue is higher or quintessential than a different intensity cue. Each cue has its very own blessings and disadvantages.

It is important to mix a couple of intensity cues a good way to reap a sturdy conversion algorithm. Some intensity cues produce much less precise floor facts because of motives inclusive of smoothness constraints different intensity cues give a higher precise sur-face, combing them can also additionally ends in a higher result.

CHAPTER 2:
Deep Learning

1. Introduction

In recent years, Deep Learning has earned huge success in many domains. It has been increasing rapidly and applied to the most traditional application domains, this achievement led to the implementation of deep learning techniques in 3D reconstruction as well as some new areas that present more opportunities. Different approaches have been proposed based on different categories of learning, including supervised, semi-supervised, and unsupervised learning. Experimental results give state-of-the-art production using deep learning when compared to classical methods in the domain of face reconstruction and recognition, computer vision, 3D reconstruction, and several others. However, deep learning techniques for 3D reconstruction are still in the early phase, this thesis studies deep learning-based methods in 3D reconstruction from a single image. Several methods and their significance are discussed, also some challenges and research opportunities are proposed for further research directions.

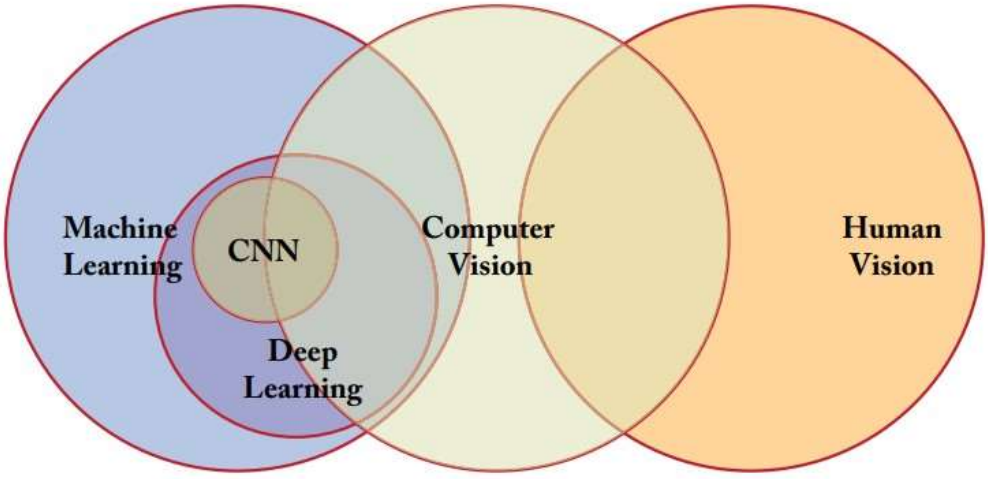


Figure 13: The relation between human vision, computer vision, machine learning, deep learning, and CNNs.

2. Related work

The problem of reconstructing the 3D face from a single image has recently received renewed attention from the field with the use of classical computer vision methods [39]. The first method to reconstruct 3D faces from In-The-Wild images are DECA Flame which learns an animatable displacement model from in-the-wild images without 2D-to-3D supervision [37], another popular 3D face model is the multilinear tensor model [46, 47, 48, 49]. A few model-free single-image reconstruction methods have been proposed [50, 51, 52]; numerous methods also are proposed

which employ CNNs to achieve efficient face reconstruction. Some of them apply CNNs to regress 3DMM coefficients directly [52, 53, 54, 55, 56, 57],

2.1. Coarse reconstruction

Numerous monocular 3D face reconstruction methods support Vetter and Blanz [1998] by computing coefficients of pre-computed statistical models in an analysis-by-synthesis fashion. Such methods can be classified into optimization-based [120; 121; 122; 123; 124; 125; 126], or learning-based methods [127; 128; 129; 130; 131; 132; 133; 134; 135; 136]. These methods estimate parameters of a statistical face model with a fixed linear shape space, which captures only low-frequency shape information. This results in overly smooth reconstructions.

2.2. Single-view Method

Current CNN methods [59, 60, 61, 62, 63, 64, 65, 67, 68, 69, 70, 71, 72] train the CNN network supervised by 3D face scan ground truth and deliver impressive results. [60, 63, 66] generate synthetic presented face images with real 3D scans. [61, 64, 65, 67, 69, 70] suggest their deep neural networks trained using fitted 3D shapes by old methods as substitute labels. The necessity of realistic training data is still a great hindrance.

3. Fundamentals of Deep Learning

3.1. What is Deep Learning?

Deep Learning (DL) is a subdomain of machine learning that enables computers to learn from experience and understand the world in terms of a hierarchy of concepts and it is the new subfield generation of the IA domain (see figure1.1). Deep learning models can be stronger than shallow machine learning models in feature representation. The performance of traditional machine learning methods usually rely on users' experiences, while deep learning approaches rely on the data. Therefore, we can find out that deep learning approaches have reduced the demands for users. Deep learning models usually adopt hierarchical structures to connect their layers. The output of a lower layer can be regarded as the input of a higher via simple linear or nonlinear calculations. These models can transform low-level features of the data into high-level abstract features.

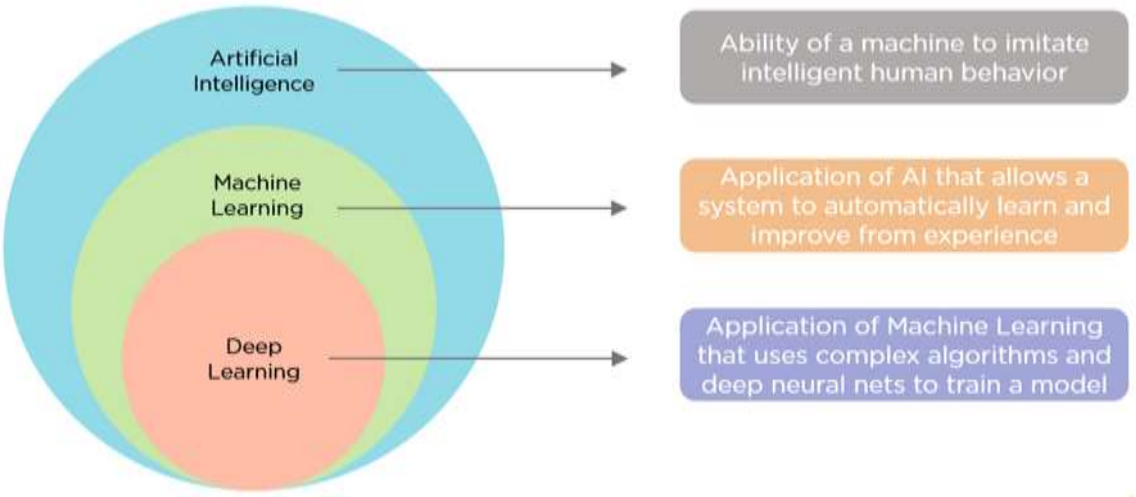


Figure 14: The relation between AI, ML and deep learning

3.2. Why deep learning?

There are three key benefits, which are granted by deep learning.

- **Simplicity**: Rather than of problem specific tweaks and tailored point detectors, deep networks give essential architectural blocks, network layers, which are repeated many times to create large networks.
- **Scalability**: Deep learning standards are easily scalable to large datasets. Other bidding methods, e.g., kernel machines, encounter serious computational difficulties if the datasets are huge.
- **Domain transfer**: A model learned on one task connects to other related tasks and the learned features are common enough to serve on a variety of tasks which may become scarce data available.

The development of deep learning was designed to achieve automatic feature extraction from raw data without any handy art features tool and can be made in huge data space, as it represented in **Figure 15**

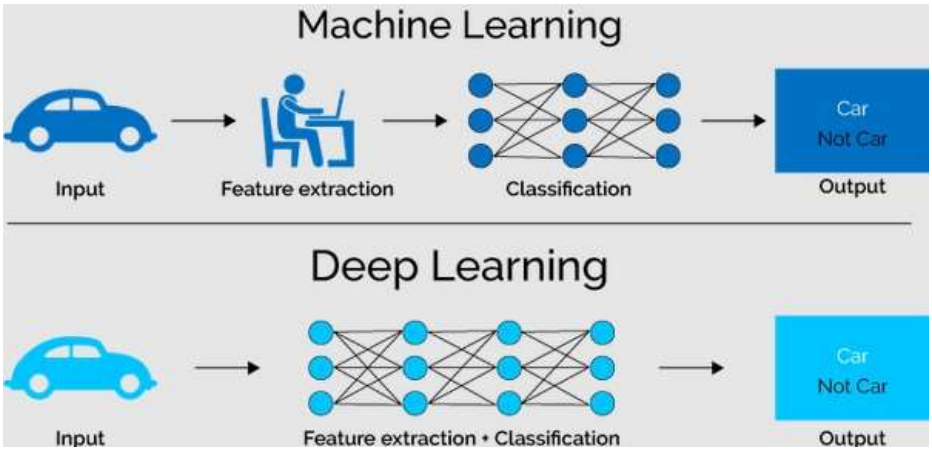


Figure 16 The process of ML classic compared to that of Deep Learning

3.3. Applications of Deep Learning

There are numerous commitments done which shows us the achievement of Deep Learning in the various application fields [29]. Applications domain like object detection, natural image classification, biological data classification like biological X-ray, MRI images, biological signals, natural Language Processing , genomes expression recognition, robotics, speech recognition and detection, Facial Expression Recognition [22][25].All three Deep Learning Architectures are useful in different application areas.

3.4. Deep Learning Frameworks

Figure 17: Different DL frameworks

3.5. Different architectures of Deep Learning

Here, we can distinguish the most popular deep learning architectures are proposed for different real world applications.

Table 1: Different architectures of Deep Learning

Architecture	Application
RNN	Speech recognition, handwriting recognition
LSTM/GRU network	Natural language text compression, handwriting Recognition, speech recognition, image captioning
CNN	Image recognition, video analysis, natural language processing
DBN	Image recognition, information retrieval, natural language understanding, failure prediction
DSN	Information retrieval, continuous speech recognition

Table 1: Different architectures of Deep Learning

4. Convolutional Neural Networks

4.1. Introduction

Convolutional Neural Network is the widely used deep learning framework [26] that was inspired by the visual cortex of animals [23]. Initially, it had been widely used for object recognition tasks but now it is being examined in other domains as well like object tracking [24], pose estimation [25], text detection and recognition [28], visual saliency detection [28], action recognition [32], scene labeling [31] and many more [32].

The neocognitron in 1980 [35] is granted the predecessor of ConvNets. LeNet was the pioneering work in Convolutional Neural Networks by LeCun et al. in 1990 [32] and later improved on it [36]. It was specifically designed to classify handwritten digits and was successful in recognizing visual patterns directly from the input image without any preprocessing. But, due to a lack of sufficient training data and computing power, this architecture failed to perform well in complex problems. Later in 2012, Krizhevsky et al. [37] had come up with a CNN model that succeeded in making down the error rate on ILSVRC competition [38]. Over the years later, their job has grown and become one of the most important ones in the domain of computer vision and used by many for trying out variations in CNN architecture. AlexNet was able to achieve remarkable results compared to the previous model of learning and without any unsupervised pre-training to keep the net simple. The architecture can be considered as a major variant of LeNet having five convolutional layers

supported by three fully-connected layers. There have been numerous variations of AlexNet since its a big success in ILSVRC-2012 competitions

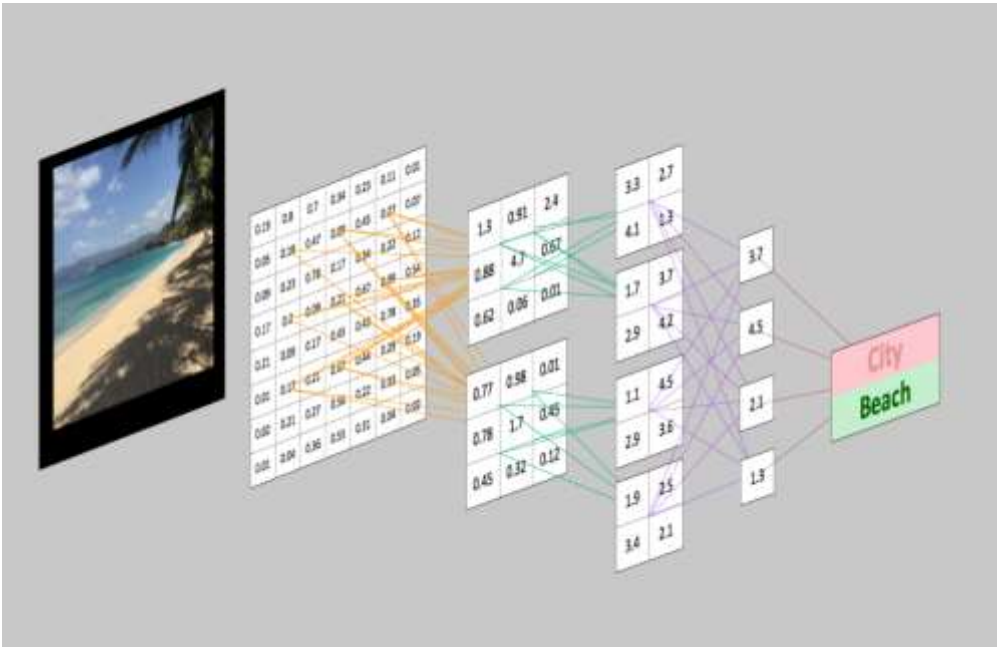


Figure 18: CNNs and computer vision

4.2. CNN Layers

4.3.1. Linear or Fully Connected

Mathematically, we can imagine a linear layer as a function that utilizes a linear transformation on a vectorial input of dimension I and output a vector of dimension O. Usually the layer has a bias parameter.

$$y = A \cdot x + \tag{18}$$

$$y_i = \sum_{j=1}^{I} (A_{i,j} x_j) + b \tag{19}$$

The basic computational unit of the brain called a neuron moves the linear layer. Around 86 billion neurons can be found in the human nervous system and they are related with nearly 1014 - 1015 synapses. Each neuron accepts input signals from its dendrites and delivers output signals beside its axon. The linear layer is a simplification of a group of neurons having their dendrites connected to the same inputs. Normally, an activation function, such as sigmoid, is used to mimic the 1-0 impulse carried away from the cell body

and to add non-linearity. However, we examine here that the activation function is the identity function that output real values

4.3.2. Activation functions or Non Linearity

Every activation function (or *non-linearity*) uses a single number and delivers a certain fixed mathematical operation on it. You may face many activation functions in practice:

- **Sigmoid**: takes a real-valued input and squashes it to range between 0 and 1

$$\mathrm{f}(x) = \frac{1}{1 + e^{-x}} \tag{20}$$

Equation 4.1: Sigmoid

- **tanh**: takes a real-valued input and squashes it to the range [-1, 1]

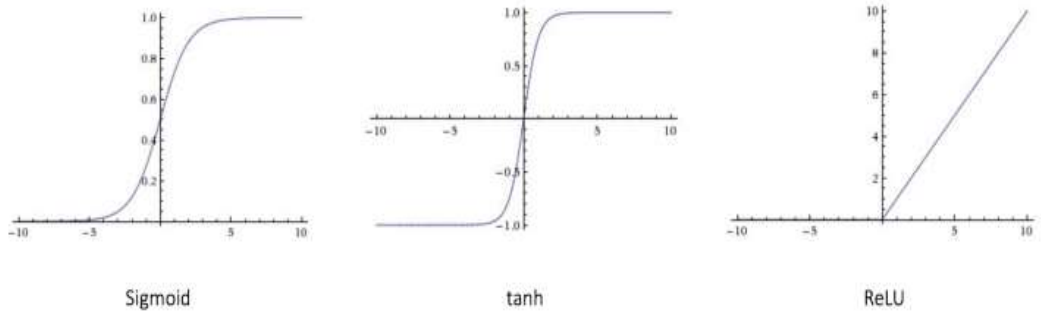
$$\tanh(x) = 2f(2x) - 1 \tag{21}$$

Equation 4.2: Tanh

- **ReLU**: stands for Rectified Linear Unit. It takes a real-valued input and thresholds it at zero (replaces negative values with zero)

$$f(x) = \max(0, x) \tag{22}$$

Equation 4.3: ReLu



4.3.3. Pooling Layer

Fundamental ConvNet design has to exchange Conv layers and pooling layers and the last capacities to diminish the spatial component of the actuation maps (without loss of data) and

the number of boundaries in the net and in this manner lessening the by and large computational intricacy. This controls the issue of overfitting. A portion of the regular pooling activities is max pooling, normal pooling, stochastic pooling [45], unearthly pooling [46], spatial pyramid pooling [26], and multiscale orderless pooling [48]. Fig. 2 shows the activity of max pooling.

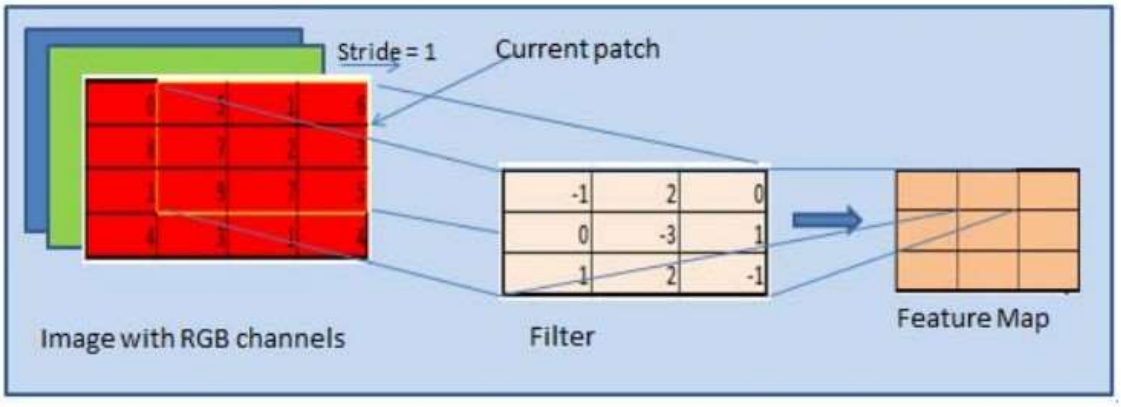


Figure 19: Convolution Operation. The input image with Red (R), Green (G), and Blue (B) color channels, whose current receptive region is highlighted as a yellow box. The convolution operation involves computing dot products with corresponding elements of the receptive region (of R, G, B channels) and filter. The receptive field window slides through the image, spatially computing inner products and resulting in a feature map [2].

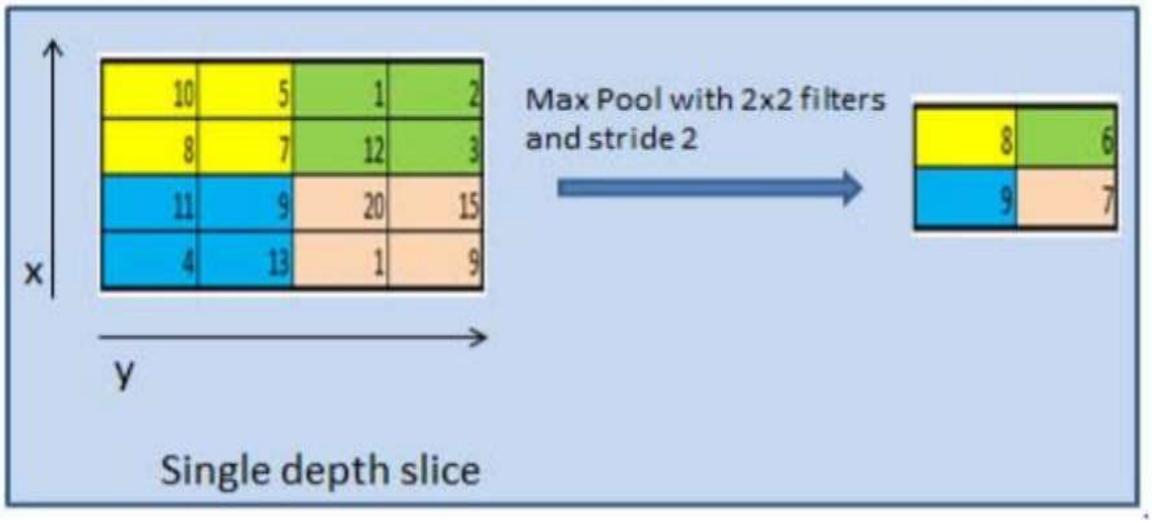


Figure 20: Max-Pooling Operation

4.3.4. Spatial Convolution

Regular Neural Networks, particularly produced of linear and activation layers, do not estimate well to full images. For instance, images of size 3 × 224 × 224 (3 color channels, 224 wide, 224 high) would require a first linear layer having $3 * 224 * 224 + 1 = 150,129$ parameters for a individual neurone (e.g. output). Spatial convolution layers get the advantage of the fact that their input (e.g. images or feature maps) presents many spatial relationships. Neighboring pixels should not be touched by their location within the image.

Thus, a convolutional layer learns a set of N_k filters $F = f_1, \ldots, f_{N_k}$, which are convolved spatially with input image x to produce a set of N_k 2D features maps z:

$$zk = fk * x \tag{23}$$

Where $*$ is the convolution operator. When the filter correlates well with a region of the input image, the response in the corresponding feature map location is strong. Unlike conventional linear layers, weights are shared over the entire image reducing the number of parameters per response and equivariance is learned (i.e. an object shifted in the input image will simply shift the corresponding responses similarly). In addition, a fully connected layer can be seen as a convolutional layer with a filter of sizes 1 × 1 × input-Size. It is important to highlight that a spatial convolution is not defined by the spatial size of the input feature maps (e.g. wide and high), neither by the size of the output feature maps, but by the number of filters (e.g. number of output channels), the properties of its filters (e.g. number of input channels, wide, high) and the properties of the convolution (e.g. padding, stride). Animations showing different kinds of convolution can be viewed on line 1.

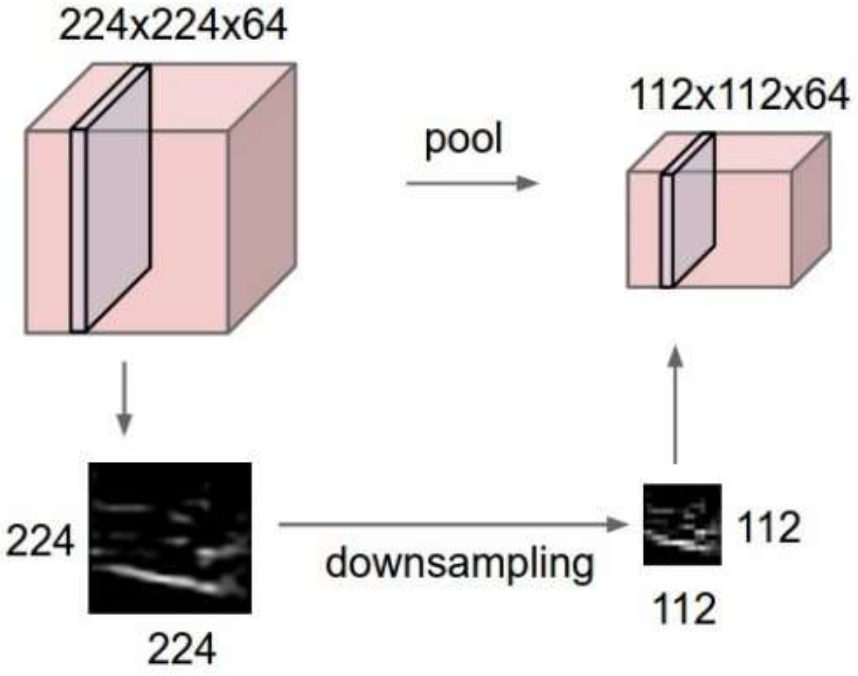


Figure 21: The example of a spatial pooling operation in 2 × 2 areas by a stride of 2 in a high way, and 2 in the width way, without padding.

4.3.5. Spatial Pooling

In CNN, a pooling layer is typically present to produce invariance to somewhat different input images and to decrease the dimension of the feature maps (e.g. wide, high):

$$P_R = P_{i \in R}(zi) \quad (24)$$

Where P is a pooling function above the region of pixels R. Max, pooling is favored as it avoids removal of negative elements and prevents blurring of the activations and gradients throughout the network since the gradient is placed in a single location during backpropagation. Its aggregation function, the high and width dimensions of the space where it is applied, and the properties of the convolution (e.g. padding, stride) mark the spatial pooling layer.

4.3.6. Batch Normalization

This layer immediately became very common mostly because it serves to converge faster [14]. It attaches a normalization step (shifting inputs to zero-mean and unit variance) to make the inputs of each trainable layer comparable beyond features. By doing this it assures a high learning rate while preserving network learning. Also, it allows activations functions such as TanH and Sigmoid to not get lost in the saturation mode (e.g. gradient equal to 0).

CNNs are one of the most successful classes of neural networks. They are dominant in several computer vision tasks and attracting care across a difference of domains, including radiology [23].

One of the variants of the traditional CNN is "Network In-Network"(NIN) proposed by Lin et al. [43], where the 1*1 convolution filter used is a Multi-Layer Perceptron(mlp) instead of the conventional linear filters and the fully connected layers are replaced by a global average pooling layer[23]. The resultant structure is called the mlpconv layer since the micro-network consists of a stack of mlp conv layers. Unlike CNN, NIN is capable of enhancing the abstraction ability of the latent concepts. They were successful in proving that the last mlpconv layer of NIN were confidence maps of the categories which leads to the possibility of performing object [2].

4.3. Training Methods

4.3.1. From Scratch

- **Initialization:** All the network parameters are generally initialized with Layer-sequential unit-variance (LSUV) (e.g. each parameter as Gaussian random variables with mean 0 and standard deviation $\frac{1}{n_{inputs}}$ and biases are initialized to zero). Since the LSUV initialization works under the assumption of preserving unit variance of the input, pixel intensities are given after subtracting the mean and dividing by the standard deviation. More information can be found in chapter 3 of Michael Nielsen's book [30]. In the case of pre-train networks, the mean and std of the original dataset are kept.

- **Loss function:** To measure the capacity of the network and approximate the ground truth labels for all training inputs, we represent a loss function that accepts as inputs the weights, biases, and examples from the training set. For case, the loss could be the number 12 of images correctly classified. Nevertheless, the most efficient way to find the weights and biases, regarding the number of parameters, is to use an algorithm similar to the Stochastic Gradient Descent (SGD). To do so, if our preferred loss function is not smooth, we have to choose a surrogate loss (e.g. derivable function) such as Mean Square Error or Cross-Entropy.

- **Backpropagation:** For each example, we calculate the prediction and its connected loss. We sum up all the losses to measure the final error. Then we use the backpropagation algorithm to propagate the error in order to count the partial derivatives $\frac{\delta E}{\delta w}$ and $\frac{\delta E}{\delta b}$ of the cost function E for all weights w and bias b [81].

- **Optimization:** Once all the derivatives are counted, we update our parameters using a chosen optimization technique such as SGD. We then iterate the prediction (e.g. forward pass), the backpropagation of errors (e.g. backward pass), and the optimization until convergence hoping to find a local minimum low sufficient to ensure good predictions. Even if the chosen surrogate loss function of a neural network is non-convex, SGD works well in practice.

- **Grid search:** It is common to examine manually the area of hyperparameters such as learning rate, weight decay, learning rate decay, amount of dropout, not to consider

the architectures hyperparameters, to achieve the best performance in terms of both accuracy and training time. [82] evaluated the influence of architecture options and optimization hyperparameters on ImageNet. While there are very rare theoretical studies, technical studies of this kind can help the reader to reduce the space of hyperparameters to explore.

4.3.2. Transfer Learning

- **Features Extraction:** It consists of obtaining features from the network by forwarding examples. Changes to the examples are possible such as horizontal flip. Then the associated features to the example are aggregated whether by equalizing or piling them. Finally, a classifier is trained and tested on the features. Typically, the latter is a Support Vector Machine with a linear kernel.
- **Fine Tuning:** It consists of training a pre-trained network on a shorter dataset. Typically, the ultimate fully connected layers, which can be viewed as classification layers, are reset and a smaller learning rate is applied to the pre-trained layers. By doing so, the purpose is to adapt the features to the new dataset. More different is the latter from the primary dataset, more parameters/layers must be reset.

4.3.3. Loss Functions

In this subsection, we present the three most used loss functions to train deep neural networks for classification.

- **Mean Square Error (MSE):** It is a multi-class loss formerly used to train neural networks.

$$\text{Loss}(\text{x},\text{y}) = \frac{1}{n}\sum_{i} |x_i - y_i|^2 \tag{25}$$

With x a vector of n predictions, and y a binary vector full of 0 besides a 1 in the similar class dimension.

- **Cross Entropy:** It is a multi-class loss, which is nearly a better choice than MSE.

$$\text{Loss}(\text{x},\text{y}) = -\sum_{i} y_i * \log\left(\frac{\exp(x_i)}{(\sum_j \exp(x_j))}\right) \tag{26}$$

With x a vector of n predictions, and y a binary vector full of 0 besides a 1 in the similar class dimension. The initialization of the parameters may result in the network being decisively wrong for some training input (an output neuron will have saturated near 1 when it should be 0, or vice versa). The MSE loss will regularly slow down learning, but the Cross-Entropy loss will not.

- **Loss Multi Label:** It is the adaptation of the Cross-Entropy loss for multi-label classification. It is a multi-label one-versus-all loss based on max-entropy.

$$\text{Loss}(\text{x},\text{y}) = -\sum_{i}\left(y_i * \log(\frac{\exp(x_i)}{(\sum_j \exp(x_j))}) + (1-y_i) * \log\left(\frac{1}{1+\exp(x_i)}\right)\right) \quad (27)$$

4.3.4. Optimization Algorithms

The loss function of a CNN is highly non-convex. Confidently, the latter is also fully derivable, so that gradient-based optimization algorithms can be applied. However, CNN's are usually made of tens of millions of parameters. Thus, only the first-order derivatives are used in practice. The second derivatives are costly in terms of memory and computational effort.

- **Stochastic Gradient Descent (SGD):** It is the main optimization algorithm. It consists of using a few examples to measure the gradient of the parameters regarding the loss function:

$$\theta_{t+1} = \theta_t - \lambda \cdot \nabla_{0_t} \text{L}(\text{f}_{0_t}(\text{x}_i), \text{y}_i) \quad (28)$$

There is no proof of good convergence. However, this algorithm achieves good local minima in practice, even when the parameters are randomly initialized. One of the causes could be the stochastic characteristic of this algorithm, allowing the latter to optimize another loss function and thus to get out of bad minima. The other causes could be that a lot of local minima are almost as accurate as global minima. Answers to this question are still under active research.

- **Approximation of Second Order Derivatives:** Other optimization algorithms rely on more advanced techniques such as momentum, second-order approximation, and adaptive learning rates [42, 17]. They are known to converge faster and their parameters are sometimes easier to tune by grid search. However, they take a bit more processing time to compute, but also much more memory (2 to 3 more).

- **Distributed SGD:** It is the kind of optimization used in parallel computing environments. Several computers train the same architecture with almost the same parameter values. It provides more investigation of the parameters space, which can lead to improved performance [50].

4.3.5. Regularization Approaches

A success of layers is trained to extract the features from complex data. However, the missing training data may lead towards the imperfection generalization. This phenomenon is called overfitting. To avoid this phenomenon, a series of optimisations rules have been proposed to leverage the training process, among of theses well-known strategies are listed below:

- **Regularization L2:** The primary main approach to overcome overfitting is the classical weight decay, which adds a term to the cost function to chasten the parameters in each dimension, checking the network from exactly modeling the training data and therefore help generalize to new examples:

$$\mathrm{Err}(\mathrm{x},\mathrm{y}) = \mathrm{Loss}(\mathrm{x},\mathrm{y}) + \sum_{i} \theta_i^2 \tag{29}$$

With θ a vector including all the network parameters.

- **Data augmentation:** It is a method of raising the size of the training set so that the model cannot record all of it. This can take various forms depending on the dataset. For instance, if the objects are assumed to be invariant to rotation such as galaxies or planktons, it is well suited to apply different kinds of rotations to the original images.

- **Dropout:** Finally, recent success has been shown with a regularization procedure called Dropout [41]. The concept is to randomly set a specific percentage of the activations in each layer to 0. During the training, neurons must learn better representations without

co-adapting to each other being active. During the testing, all the neurons are used to compute the prediction and Dropout acts like a form of model averaging over all possible instantiations of the model.

- **Early stopping:** It consists of ending the training before the model begins to overfit the training set. In practice, it is used a lot during the training of neural networks.

4.4. The popular CNN architectures

	LeNet	**AlexNet**	**ZFNet**	**GoogLe-Net**	**VGGNet**	**ResNet**
Input size	32 x 32	128 x 128	128X128	224x224	128 x 128	224 X 224
Kernel size	5 x 5	11 x11	7,11	1, 3,5,7	3 ,5,11	1,3,7
Stride	1	1	1	1,2	1	1,2
Developer	Yann LeCun et al [15], [14]	Alex Krizhevsky, Geoffrey Hinton, Ilya Sutskever [16]	Matthew Zeiler and Rob Fergus [16], [14]	Google [16]	Simonyan, Zisserman [16], [14]	Kaiming He [16]
Data set	MNIST	ImageNet	ImageNet	ImageNet	ImageNet	ImageNet CIFAR-10
Nb of parameters	0.060 M	60 M	60 M	4 M	138 M	6.8 M 1.7 M
Nb. Of Layers	7	8	8	22	19	152 110
Error rate	0.8	16.4	11.7	6.7	7.3	3.6 6.43

References	[15], [14]	[16], [14]	[16]	[16], [14]	[16]	[16], [14]

Table 2: The Popular CNN architectures

5. Deep Learning approach for 3D reconstruction

5.1. Introduction

Deep learning methods have attracted many researchers in the computer vision field to solve computer vision problems such as image segmentation and object recognition. This success also pointed to the implementation of deep learning techniques in 3D reconstruction. The 3D reconstruction itself is a classical problem in computer vision that has been approached by many techniques.

As far as input type and the number of pictures, there are numerous varieties of the 3D reconstruction issue, i.e. 3D reconstruction from a single image (shape-from-X), various images (sound system), or 3D reconstruction from RGB-D information. This paper just examinations the main kind of issue, for example, 3D reconstruction from a single image.

5.2. Deep Learning Techniques

Currently, several widely known architectures have made important additions to the computer vision domain, e.g. AlexNet, VGG-16, GoogLeNet, LeN , and ResNet.

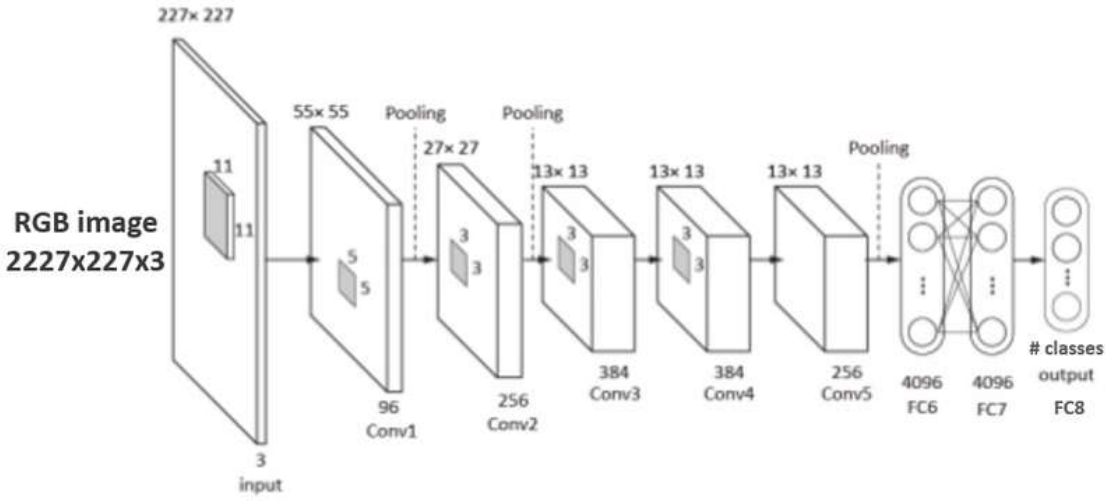


Figure 22: AlexNet Architecture

5.2.1 AlexNet

AlexNet [8] is the first deep Convolutional Neural Network (CNN) that won ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012 with an accuracy of 84.6%, compared to the traditional techniques as the next winner with an accuracy of 73.8%. AlexNet consists of five convolutional layers, max-pooling, Rectified Linear Units (ReLUs) as non-linearities, three fully-connected layers, and dropout. Fig. 10 shows the AlexNet architecture [29].

Meanwhile, University of Oxford's Visual Geometry Group (VGG) created VGG model called VGG-16 submitted to the ILSVRC in 2013 and achieved 92.7% accuracy [30]. Fig. 11 illustrates the VGG-16 architecture. VGG-16 increases the performance of the model by employing smaller receptive fields in its first layers.

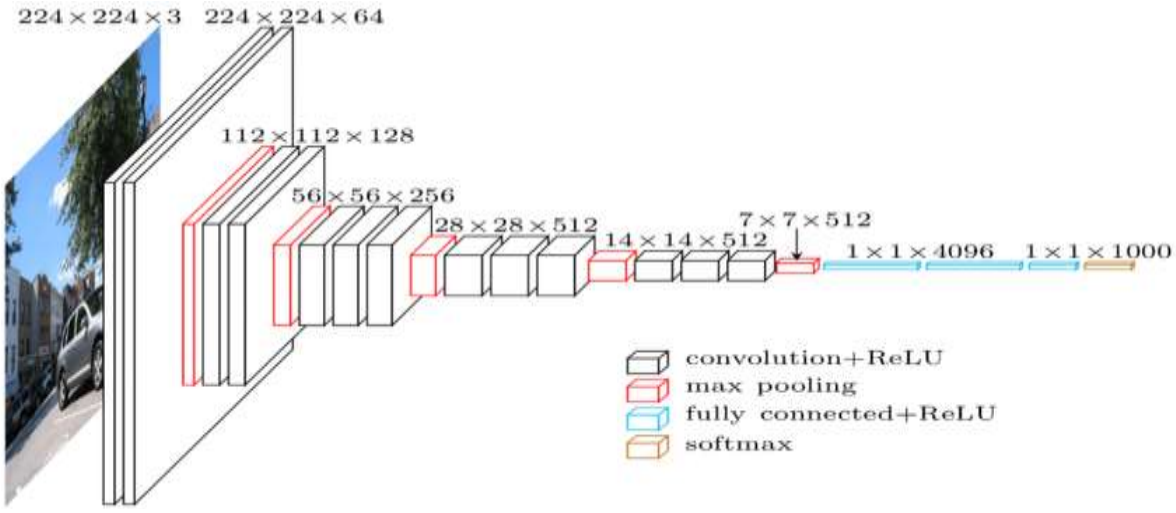


Figure 23: VGG-16 Architecture

5.2.2 GoogLeNet

GoogLeNet was developed by Szegedy et al. [31], won the ILSVRC in 2014 with an accuracy of 93.3%. The architecture of GoogLeNet consists of 22 layers and inception modules, computed in parallel, consisting of a NiN layer, a pool operation, and convolution layers. All have lxl convolution operations to reduce dimensionality. Fig. 12 shows the inception module from the GoogLeNet architecture [10].

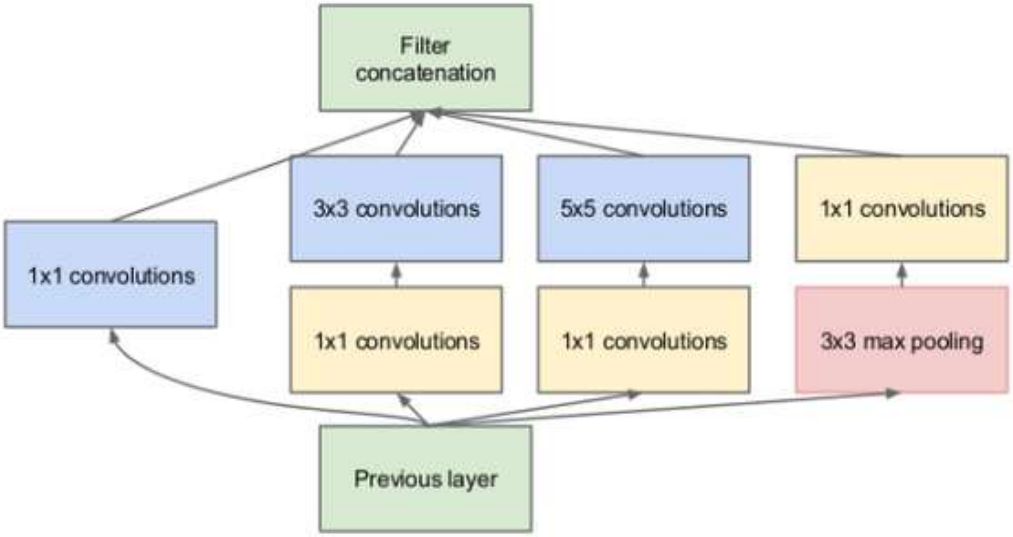


Figure 24: Inception module from the GoogLeNet architecture

5.2.3 ResNet

ResNet [32] by Microsoft won the ILSVRC in 2016 with 96.4% accuracy. The ResNet architecture is well-known because of its 152 layers and its residual blocks. The residual blocks introduced identity skip connections such that layers can copy their inputs to the next layer. Fig. 13 shows the residual block in a ResNet [32].

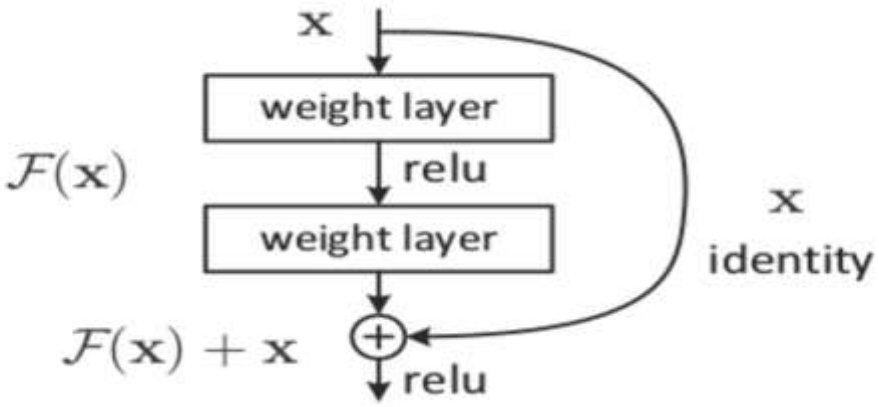


Figure 25: The Architecture ResNet

5.3. Datasets used in 3d Reconstruction and face reconstruction

Dataset Name and Reference	Year	Categories	Images	3D shapes	3D annotation type
ShapeNetCore[21]	2016	55	NA	51,300	2D-3D alignment
3DFace[12]	2019	Na	80,000	187,860	NA
ObjectNet3D[22]	2016	100	90,127	44,161	2D-3D alignment
Pascal3D[23]	2014	12	30,899	79	2D-3D alignment
KITTI[24]	2012	2	14,999	N/A	3D point

Table 2: THE MOST POPULAR DATASETS USED IN 3D RECONSTRUCTION

1.1. **VoxCeleb2**

is a large-scale speaker recognition dataset obtained automatically from open-source media. VoxCeleb2 consists of over a million utterances from over 6k speakers. Since the dataset is collected 'in the wild', the speech segments are corrupted with real world noise including laughter, cross-talk, channel effects, music and other sounds. The dataset is also multilingual, with speech from speakers of 145 different nationalities, covering a wide range of accents, ages, ethnicities and languages. The dataset is audio-visual, so is also useful for a

Figure 26: VoxCelab A large scale audio-visual dataset

number of other applications, for example – visual speech synthesis, speech separation, cross-modal transfer from face to voice or vice versa and training face recognition from video to complement existing face recognition datasets.

1.2. NoW Benchmark

The goal of this benchmark is to introduce a standard evaluation metric to measure the accuracy and robustness of 3D face reconstruction methods under variations in viewing angle, lighting, and common occlusions. The dataset contains 2054 2D images of 100 subjects, captured with an iPhone X, and a separate 3D head scan for each subject. This head scan serves as ground truth for the evaluation. The subjects are selected to contain variations in age, BMI, and sex (55 female, 45 male).

Figure 27: now benchmark dataset

1.3. VGGFace2

The **VGGFace2** dataset is made of around 3.31 million images divided into 9131 classes, each representing a different person identity. The dataset is divided into two splits, one for the training and one for test. The latter contains around 170000 images divided into 500 identities while all the other images belong to the remaining 8631 classes available for training. While constructing the datasets, the authors focused their efforts on reaching a very low label noise and a high pose and age diversity thus, making the VGGFace2 dataset a suitable choice to train state-of-the-art deep learning models on face-related tasks. The images of the training set have an average resolution of 137x180 pixels, with less than 1% at a resolution below 32 pixels (considering the shortest side).

Figure 28: VGGFace2 dataset is made of around 3.31 million images divided ino 9131 classes.

6. Conclusion

In this chapter, we have introduced the deep learning algorithms. Then, we payed attention on the CNN architecture by explaining main functions required (CNN filters, training functions, optimization methods and so on) to design it. After that, we highlighted their weaknesses, e.g., getting stuck in the local minima, overfitting, and consuming time while training a huge data set. Finally, we presented the well-known data based used for 3D reconstructions task. We have also discussed the most CNN architectures made for multiple tasks.

CHAPTER 3: Experiments and Results

1. Introduction

The aim of our work is to provide a comparison between two methods which they employed the deep learning model to generate 3D face from one single image. Each of them is making a principal common feature linked how to construct a prior model that able to generate 3D face with low error reconstruction. In this work, we will discuss the most well-known data generation used in the 3D reconstruction, which are either synthetic or real. Thus, are trained through supervised, semi-supervised and unsupervised models.

For instance [138] introduced a model-based encoder-decoder architecture, which replaces the trainable decoder with an expert-designed fixed decoder. This expert-designed decoder takes the 3DMM parameters (latent code) predicted by an encoder as an input and transforms them into a 3D reconstruction using the 3DMM.

In [129; 115] they developed a new unsupervised training model data that can generate 3D model from synthetic data. This model has made higher-level loss functions like identity preservation.

Motivated by their results in DECA [58] and [115], we will discuss in details both the weakness and performance of their approaches by giving how the dataset and their model was made and the important results.

2. DECA: Detailed Expression Capture and Animation

DECA reconstructs a 3D head model with detailed facial geometry from a single input image. The resulting 3D head model can be easily animated.

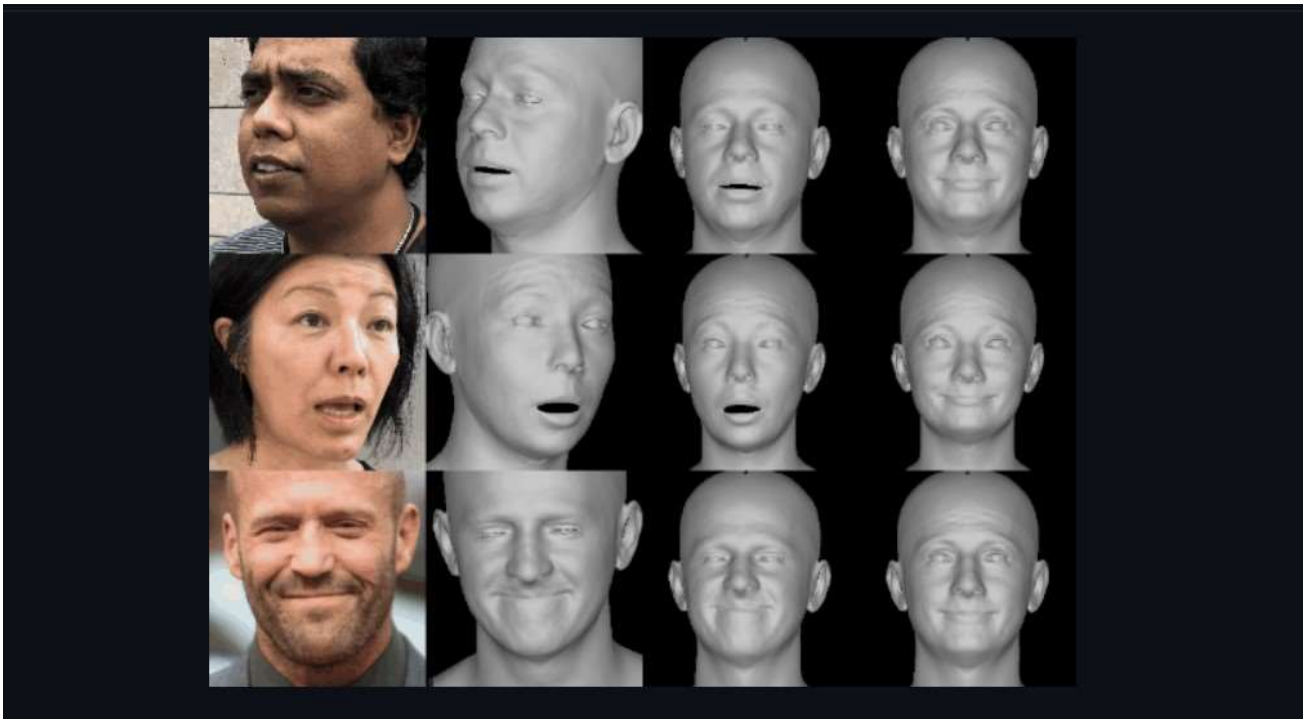


Figure 29: input image, aligned reconstruction, animation with various poses & expressions

2.1. Model Description

Here, we present the first approach that regresses 3D face shape and animatable details that are specific to an individual but it changes with expression settings [58]. In this model, DECA (Detailed Expression Capture and Animation), is trained to robustly produce a UV displacement map from a low-dimensional latent representation that consists of person-specific detail parameters and generic expression parameters, while a regressor is trained to predict detail, shape, albedo, expression, pose and illumination parameters from a single image.

DECA can learn an animatable displacement model from in-the-wild images without 2D-to-3D supervision. In contrast to prior work, these animatable expression-dependent wrinkles are specific to an individual and are regressed from a single image. Specifically, DECA jointly learns

- A geometric detail model that generates a UV displacement map from a low-dimensional representation that consists of subject-specific detail parameters and expression parameters.
- a regressor that predicts subject-specific detail, albedo, shape, expression, pose, and lighting parameters from an image.

2.2. PRELIMINARIES

2.2.1. Geometry prior

FLAME [58] is a statistical 3D head model that combines separate linear identity shape and expression spaces with linear blend skinning (LBS) and pose-dependent corrective blend shapes to articulate the neck, jaw, and eyeballs. Given parameters of facial identity $\boldsymbol{\beta} \in \mathrm{R}|\boldsymbol{\beta}|$, pose $\boldsymbol{\theta} \in \mathrm{R}3k+3$ (with $k = 4$ joints for neck, jaw, and eyeballs), and expression $\boldsymbol{\psi} \in \mathrm{R}|\boldsymbol{\psi}|$, FLAME outputs a mesh with $n = 5023$ vertices. The model is defined as

$$(\beta, \theta, \psi) = W(TP(\beta, \theta, \psi), J(\beta), \theta, W), \tag{30}$$

With the blend skinning function W (T, J, $\boldsymbol{\theta}$,W) that rotates the vertices in T $\in$ R3n around joints J $\in$ R3k , linearly smoothed by blend weights W $\in$ R$k \times n$. The joint locations J are defined as a function of the identity $\boldsymbol{\beta}$. Further,

$$TP(\beta, \theta, \psi) = T + BS(\beta; S) + BP(\theta; P) + BE(\psi; E) \tag{31}$$

Denotes the mean template T in "zero pose" with added shape blendshapes BS ($\boldsymbol{\beta}$;S) : R|$\boldsymbol{\beta}$| $\rightarrow$ R3n, pose correctives BP ($\boldsymbol{\theta}$; P) : R3k+3 $\rightarrow$ R3n, and expression blendshapes BE ($\boldsymbol{\psi}$; E) : R|$\boldsymbol{\psi}$| $\rightarrow$ R3n, with the learned identity, pose, and expression bases (i.e. linear subspaces) S, P and E. See [Li et al. 2017] for details.

2.2.2. Appearance model

FLAME does not have an appearance model; hence, we convert the Basel Face Model's linear albedo subspace [137] into the FLAME UV layout to make it compatible with FLAME. The appearance model outputs a UV albedo map ($\boldsymbol{\alpha}$) $\in$ R$d \times d \times 3$ for albedo parameters $\boldsymbol{\alpha} \in$ R|$\boldsymbol{\alpha}$|.

2.2.3. Camera model

Photographs in existing in-the-wild face datasets are often taken from a distance. We, therefore, use an orthographic camera model c to project the 3D mesh into image space. Face vertices are projected into the image as v = $s\Pi(Mi)$ + t, where $Mi \in$ R3 is a vertex in M, $\Pi \in$ R2×3 is the orthographic 3D-2D projection matrix, and $s \in$ R and t ∈ R2 denote isotropic scale and 2D translation, respectively. The parameters, and t are summarized as $\boldsymbol{c}$.

2.2.4. Illumination model

For face reconstruction, the most commonly employed illumination model is based on Spherical Harmonics (SH) [Ramamoorthi and Hanrahan 2001]. By considering that the light source is distant and the face's surface reflectance is Lambertian, the shaded face image is computed as:

$$B(\alpha, I, N_{uv})ij = A(a)_{ij} \odot + \sum_{k=1}^{9} I_k H_k\left(N_{i,j}\right), \quad (32)$$

Where the albedo, A, surface normal, N, and shaded texture, B, are represented in UV coordinates and where $Bi, j \in R^3$, $A_{i,j} \in R^3$, and $Ni, j \in R^3$ denote pixel (i, j) in the UV coordinate system. The SH basis and coefficients are defined as : R3 ⟶ R and l = [lT 1, … , lT9]T , with lk ∈ R3, and ⊙ denotes the Hadamard product.

2.2.5. Texture rendering

Given the geometry parameters ($\boldsymbol{\beta}$, $\boldsymbol{\theta}$, $\boldsymbol{\psi}$), albedo ($\boldsymbol{\alpha}$), lighting (l) and camera information $\boldsymbol{c}$, we can generate the 2D image Ir by rendering as Ir = R(M, B, c), where R denotes the rendering function. FLAME can create face geometry with different poses, shapes, and expressions from a low-dimensional latent space. However, the representational power of the model is limited by the low mesh resolution and consequently, mid-frequency details are mostly missing from FLAME's surface.

2.3. Main Features

- **Reconstruction:** provides head pose, shape, detailed face geometry, and lighting information of a single image.

- **Animation:** animate the face with realistic wrinkle deformations.
- **Robustness:** tested on facial images in unconstrained conditions. Our method is robust to various poses, illuminations, and occlusions.
- **Accurate:** state-of-the-art 3D face shape reconstruction on the "NoW Challenge" benchmark dataset.

2.4. Results

Figure 30: the predicted 2D landmanks, 3D landmarks (red means non-visible points), coarse geometry, detailed geometry, and depth.

2.5. METHOD

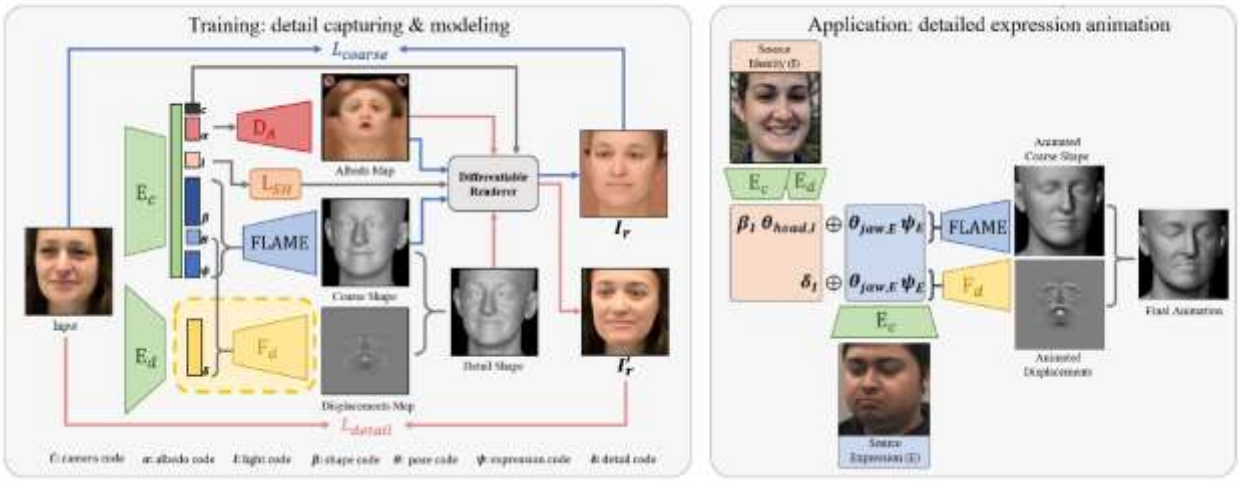


Figure 31: DECA training and animation

DECA Using Flame Model

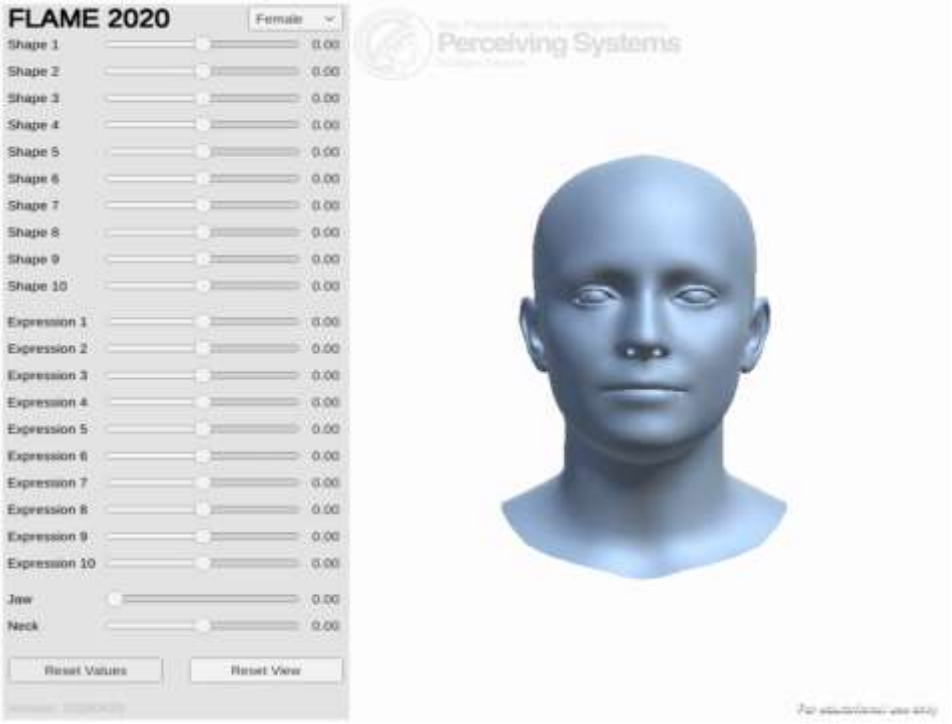


Figure 32: Flame Model

2.6. Training

2.6.1. Dataset Using for Training

DECA use three publicly available datasets:

- **VGGFace2**: contains images of over 8k subjects, with an average of more than 350 images per subject
- **BUPT-Balancedface**: offers 7k subjects per ethnicity (i.e. Caucasian, Indian, Asian and African),
- **Vox-Celeb2:** contains 145k videos of 6k subjects. In total, DECA is trained on 2 Million images.
- **FAN**: to predict 68 2D landmarks $\boldsymbol{k_i}$ on each face. To improve the robustness of the predicted landmarks, we use FAN for each image twice with different face crops, and discard all images with non-matching landmarks.

2.6.2. Pseudo Code

- **Decoder Model**

```
def __init__(self, latent_dim=100, out_channels=1, out_scale=0.01, sample_mode = 'bilinear'):
        super(Generator, self).__init__()
        self.out_scale = out_scale
        self.init_size = 32 // 4  # Initial size before upsampling
        self.l1 = nn.Sequential(nn.Linear(latent_dim, 128 * self.init_size ** 2))
        self.conv_blocks = nn.Sequential(
            nn.BatchNorm2d(128),
            nn.Upsample(scale_factor=2, mode=sample_mode), #16
            nn.Conv2d(128, 128, 3, stride=1, padding=1),
            nn.BatchNorm2d(128, 0.8),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Upsample(scale_factor=2, mode=sample_mode), #32
            nn.Conv2d(128, 64, 3, stride=1, padding=1),
            nn.BatchNorm2d(64, 0.8),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Upsample(scale_factor=2, mode=sample_mode), #64
            nn.Conv2d(64, 64, 3, stride=1, padding=1),
            nn.BatchNorm2d(64, 0.8),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Upsample(scale_factor=2, mode=sample_mode), #128
            nn.Conv2d(64, 32, 3, stride=1, padding=1),
            nn.BatchNorm2d(32, 0.8),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Upsample(scale_factor=2, mode=sample_mode), #256
            nn.Conv2d(32, 16, 3, stride=1, padding=1),
            nn.BatchNorm2d(16, 0.8),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Conv2d(16, out_channels, 3, stride=1, padding=1),
            nn.Tanh(),
        )
```

- **Options for Face Model**

```
# Options for Face model
cfg.model = CN()
cfg.model.topology_path = os.path.join(cfg.deca_dir, 'data', 'head_template.obj')
# texture data original from
http://files.is.tue.mpg.de/tbolkart/FLAME/FLAME_texture_data.zip
cfg.model.dense_template_path = os.path.join(cfg.deca_dir, 'data',
'texture_data_256.npy')
cfg.model.fixed_displacement_path = os.path.join(cfg.deca_dir, 'data',
'fixed_displacement_256.npy')
cfg.model.flame_model_path = os.path.join(cfg.deca_dir, 'data', 'generic_model.pkl')
cfg.model.flame_lmk_embedding_path = os.path.join(cfg.deca_dir, 'data',
'landmark_embedding.npy')
cfg.model.face_mask_path = os.path.join(cfg.deca_dir, 'data', 'uv_face_mask.png')
cfg.model.face_eye_mask_path = os.path.join(cfg.deca_dir, 'data',
'uv_face_eye_mask.png')
cfg.model.mean_tex_path = os.path.join(cfg.deca_dir, 'data', 'mean_texture.jpg')
cfg.model.tex_path = os.path.join(cfg.deca_dir, 'data',
'FLAME_albedo_from_BFM.npz')
cfg.model.tex_type = 'BFM' # BFM, FLAME, albedoMM
cfg.model.uv_size = 256
cfg.model.param_list = ['shape', 'tex', 'exp', 'pose', 'cam', 'light']
cfg.model.n_shape = 100
cfg.model.n_tex = 50
cfg.model.n_exp = 50
cfg.model.n_cam = 3
cfg.model.n_pose = 6
cfg.model.n_light = 27
cfg.model.use_tex = False
cfg.model.jaw_type = 'aa' # default use axis angle, another option: euler
```

2.7. Evaluation

DECA achieved 9% lower mean shape reconstruction error on the NoW Challenge dataset compared to the previous state-of-the-art method [139].

The left figure compares the cumulative error of our approach and other recent methods (RingNet and Deng et al. have nearly identical performance, so their curves overlap each other). Here we use point-to-surface distance as the error metric, following the NoW Challenge.

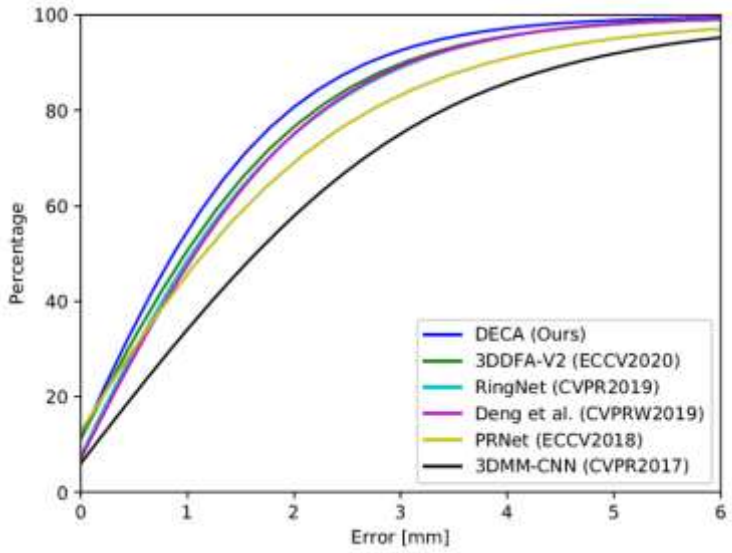


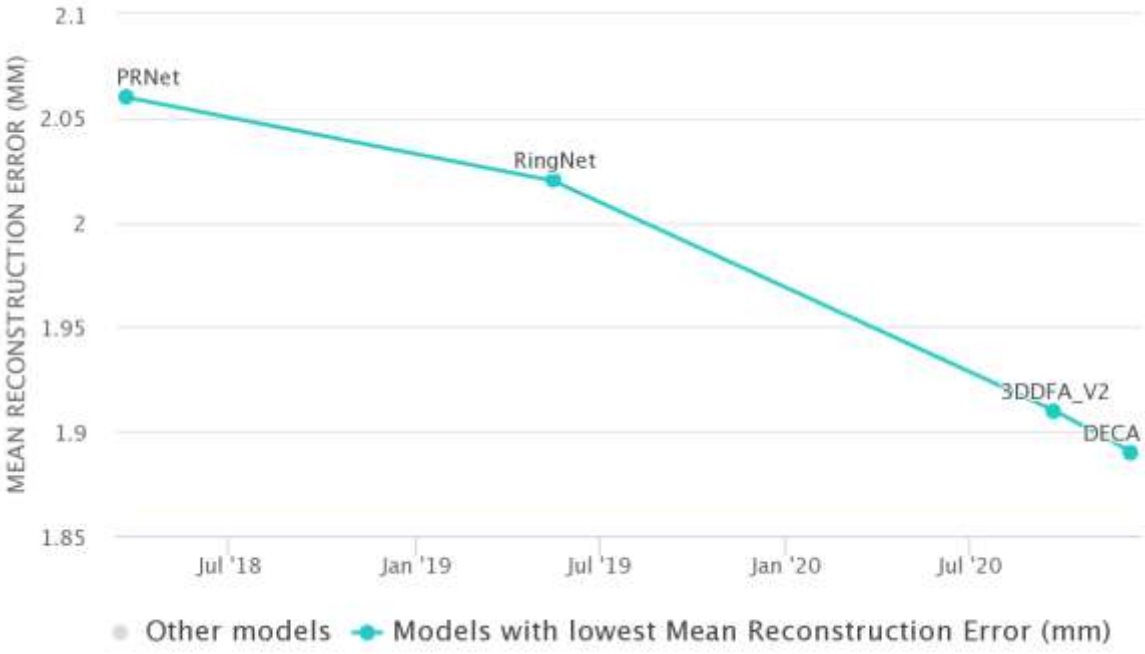
Figure 33: the comparison of the cumulative error of this approach and other recent methods (RingNet and Deng et al. have nearly identical performances

Method	Median(mm)	Mean(mm)	Std(mm)
Ours	**1.09**	**1.38**	**1.18**
3DDFA-V2	1.23	1.57	1.39
RingNet	1.21	1.54	1.31
Deng et al.	1.23	1.54	1.29
PRNet	1.50	1.98	1.88
3DMM-CNN	1.84	2.33	2.05

Table 3: the comparison of the cumulative error of deca approach

Task	Dataset	Model	Metric Name	Metric Value	Global Rank
3D Face Reconstruction	NoW Benchmark	DECA	Mean Reconstruction Error (mm)	1.38	# 1
3D Face Reconstruction	Stirling-HQ (FG2018 3D face reconstruction challenge)	DECA	Mean Reconstruction Error (mm)	1.89	# 1
3D Face Reconstruction	Stirling-LQ (FG2018 3D face reconstruction challenge)	DECA	Mean Reconstruction Error (mm)	1.91	# 1

Table 4: FG2018 3D face reconstruction challenge



2.8. Limitations of DECA

- The albedo model mainly limits the rendering quality for DECA detailed meshes.
- Existing methods, like DECA, do not explicitly model facial hair. This pushes skin tone into the lighting model and causes facial hair to be explained by shape deformations. A different approach is needed to properly model this.

- While robust, our method can still fail due to extreme head pose and lighting.
- the limited size of high-resolution datasets makes it difficult to disentangle expression- and identity-dependent details
- DECA uses a weak perspective camera model.

3. Accurate 3D Face Reconstruction with Weakly-Supervised Learning

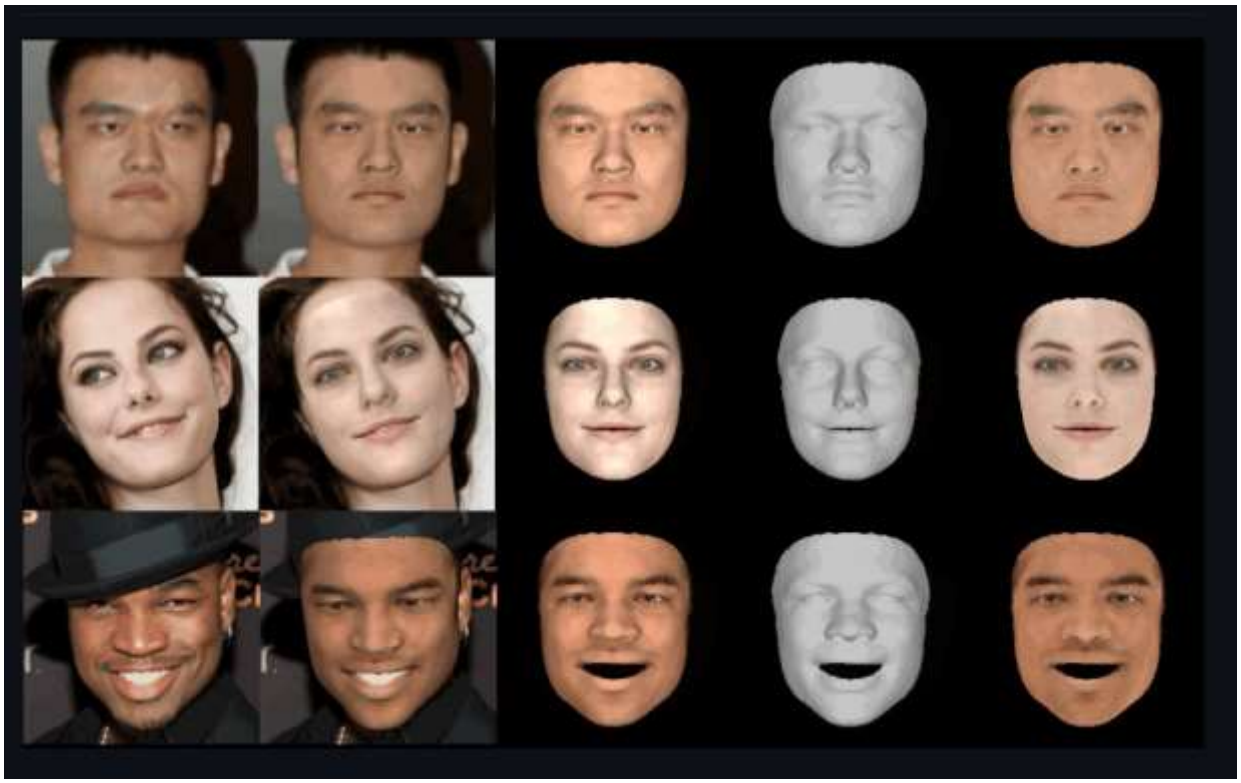


Figure 34: Accurate 3D Face Reconstruction with Weakly Supervised Learning

Accurate, and robust to pose and occlusions. It achieves state-of-the-art performance on multiple datasets such as Face Warehouse, MICC Florence and BU-3DFE.

3.1 Model Description

we propose a novel deep 3D face reconstruction approach that 1) leverages a robust, hybrid loss function for weakly-supervised learning which takes into account both low-level and perception-level information for supervision, and 2) performs multi-image face reconstruction by exploiting complementary information from different images for shape aggregation [115].

We propose a novel shape confidence-learning scheme for multi-image face reconstruction aggregation. Our confidence prediction subnet is also trained in a weakly supervised fashion

without ground-truth label. We show that our method clearly outperforms naive aggregation (e.g., shape averaging) and some heuristic strategies [34]. To our knowledge, this is the first attempt towards CNN-based 3D face reconstruction and aggregation from an unconstrained image set.

3.2 Result of Model

Figure 35: Results on in-the-wild image sets. The left-most bar chart displays the sorted value of the confidence vector summation of each image in the set. Five images sampled from a set are shown in the center with their confidence vector summations presented in the top left corner. The last two columns are our final results.

3.3 Comparison with other methods

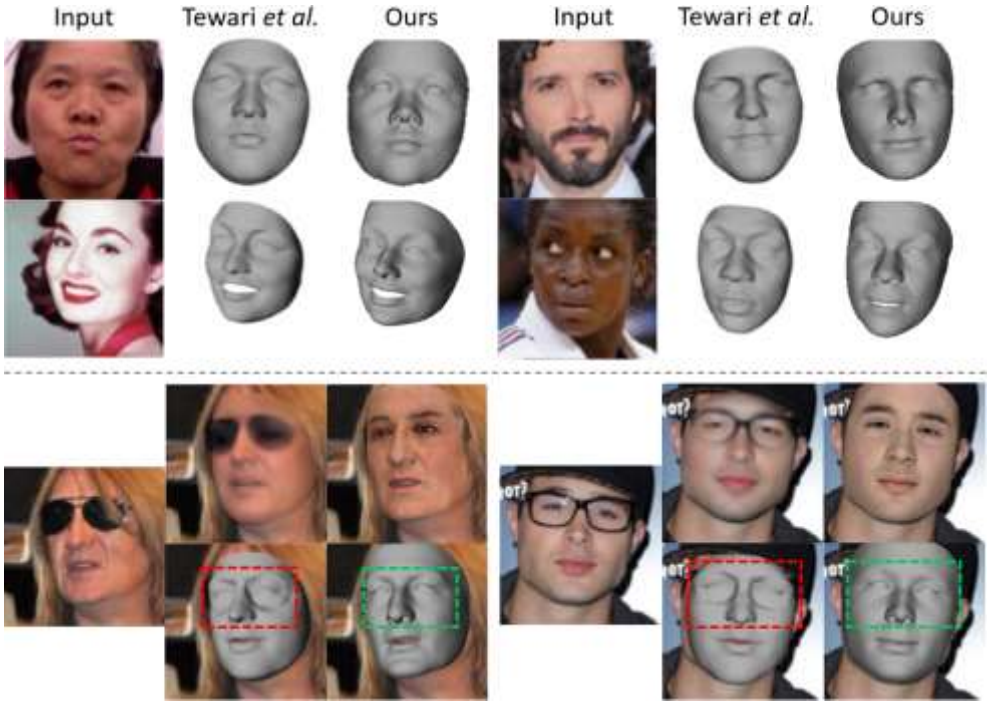


Figure 36: Comprison with Tewari et al. [78] (fine results). Top: results on different races. Bottom: results under occlusion. The images are from [78].

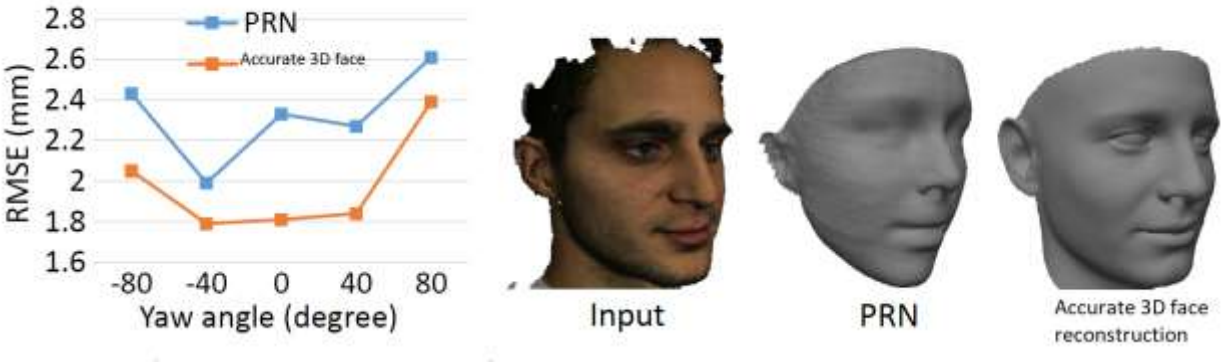


Figure 37: Comparison with PRN [79] on MICC. Leftmost: Mean RMSE of different yaw angles. Accurate 3D face Reconstruction method excels at all views. Right three images: qualitative result comparison.

3.4 PRELIMINARIES

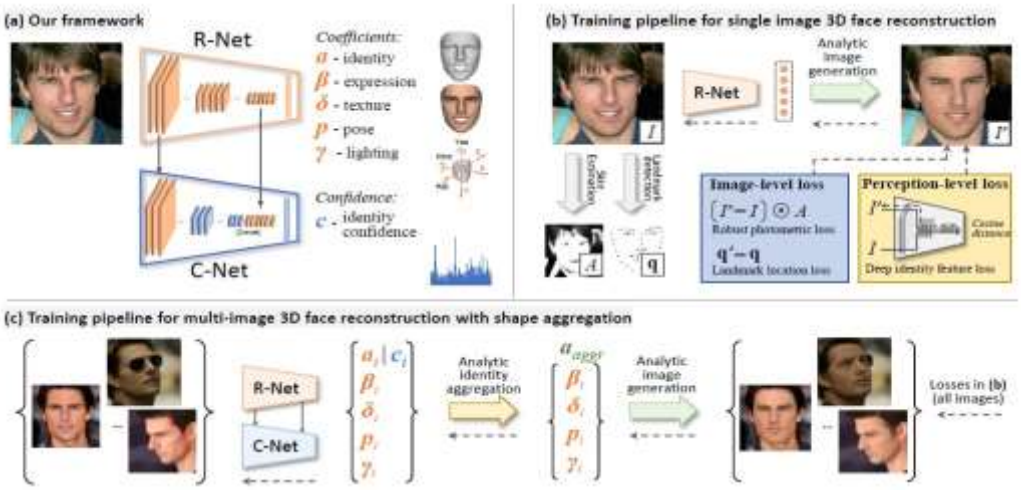


Figure 38: overview of the approach using on Accurate 3d Face Reconstruction

3.4.1. 3D Face Model

With a 3DMM, the face shape S and the texture T can be represented by an affine model

$$S = S(\alpha, \beta) = \bar{S} + B_{id}\alpha + B_{exp}\beta \tag{33}$$

$$T = T(\delta) = \bar{T} + B_t\delta \tag{34}$$

where $\bar{S}$ and $\bar{T}$ are the average face shape and texture; B_{id}, B_{exp} and B_t are the PCA bases of identity, expression, and texture respectively, which are all scaled with standard deviations; α, β , and δ , are the corresponding coefficient vectors for generating a 3D face We

adopt the popular 2009 Basel Face Model [116] for $\bar{S}$, B_{id}, $\bar{T}$ and B_t, and use the expression bases $B_{exp}$of [117] which are built from Face-Warehouse[118]

3.5 Illumination Model

We assume a Lambertian surface for face and approximate the scene illumination with Spherical Harmonics (SH) [35, 36]. The radiosity of a vertex s_i with surface normal n_i and skin texture t_i can then be computed as $C(n_i; n_i|\gamma) = \sum_{b=1}^{B^2} \gamma_b \Phi_b(n_i)$ where$\Phi_b : R^3 \rightarrow R$ are SH, basis functions and b are the corresponding SH coefficients. We choose $B = 3$ bands following [49, 48] and assume monochromatic lights such that $\gamma \in R^9$.

3.6 Camera Model

We use the perspective camera model with an empirically selected focal length for the 3D-2D projection geometry. The 3D face pose p is represented by rotation R ∈ SO (3) and translation t $\in R^3$. In summary, the unknowns to be predicted can be represented by a vector $x = (\alpha, \beta, \delta, \gamma, p) \in R^{239}$.

This model use ResNet-50 network [119] to regress the coefficients by modifying the last fully connected layer to 239 neurons. For brevity, we denote this modified ResNet50 network for single image reconstruction as R-Net. We present how we train it in the following section.

3.7 Hybrid-level Weak-supervision for Single-Image Reconstruction

Figure 39: Comparison of the results without (top row) and with (bottom row) using our skin attention mask for training.

3.5.1. Image-Level Losses

We first introduce our loss functions on low-level information including per-pixel color and sparse 2D landmarks.

3.5.2. Robust Photometric Loss

First, it is straightforward to measure the dense photometric discrepancy between the raw image and the reconstructed one [5, 50, 49, 48]. In this paper, we propose a robust, skinaware photometric loss instead of a naive one, defined as:

$$L_{Photo^{(X)}} = \frac{\sum_{\mathrm{i}\in\mathrm{M}} \mathrm{A_i} \cdot ||I_i - I_i'(x)||_2}{\sum_{\mathrm{i}\in\mathrm{M}} \mathrm{A_i}} \quad (355)$$

Where I denotes pixel index, M is projected face region which can be readily obtained, k·k denotes the l2 norm, and A is a skin color based attention mask for the training image which is described as follows.

3.5.3. Skin Attention

To gain robustness to occlusions and other challenging appearance variations such as beard and heavy make-up, we compute a skin-color probability Pi for each pixel. We train a naive Bayes classifier with Gaussian Mixture Models on a skin image dataset from [26] . For each pixel I, we set

$$A_i = \begin{cases} 1, if\ p_i > 0.5 \\ p_i, \text{otherwise} \end{cases} \quad (366)$$

3.8 Landmark Loss

Figure 40: Comparison of the results with only image-level losses (top row) and with both image-level and perceptual losses (bottom row) for training.

We also use landmark locations on the 2D image domain as weak supervision to train the network. We run the state-of-the art 3D face alignment method of [8] to detect 68 landmarks {qn} of the training images during training, we Project the 3D landmark vertices of our reconstructed shape onto the image obtaining $\{q_n\}$,and compute the loss as:

$$\mathrm{L_{Lan}(x)}\ \frac{1}{\mathrm{N}}\sum_{\mathrm{n=1}}^{\mathrm{N}}\omega_n\ \|q_n-\ q_n'(X)\|^2 \tag{377}$$

3.9 Perception-Level Loss

While using the low-level information to measure image discrepancy can generally yields decent results [5, 50, 49, 48], we find using them alone can lead to local minimum issue for CNN-based 3D face reconstruction. Figure 3 shows that our R-net trained with only image-level losses generates smoother textures and lower photometric errors than the compared opponents, but the resultant 3D shapes are less accurate by visual inspection.

$$L_p(x)=1-\frac{<f(I),f(I'(x)>}{\left||f(I)|\right|\ ||f(I'(x)||} \tag{388}$$

3.10 Limitation of Accurate 3d Face Reconstruction

- While robust, our method can still fail due to extreme head pose and lighting.
- the limited size of high-resolution datasets makes it difficult to disentangle expression- and identity-dependent details
- Do not explicitly model facial hair. This pushes skin tone into the lighting model and causes facial hair to be explained by shape deformations. A different approach is needed to properly model this.

3.11 Our Implementaion result

Figure 41: Our result of implementing the deep 3d face model

3.12 Training

3.12.1. Dataset

To train R-Net, we need to collect images from several dataset such as :

- CelebA
- 300W-LP
- I-JBA
- LFW
- LS3D

3.12.2. Pseudo Code

- **Model for Single image face reconstruction**

```
# Initialization

   def __init__(self,opt):
      self.Face3D = face_decoder.Face3D () #analytic 3D face object
      self.opt = opt # training options
      self.Optimizer = tf.train.AdamOptimizer (learning_rate = opt.lr) # optimizer
   # Load input data from queue
      def set_input(self,input_iterator):
      self.imgs,self.lm_labels,self.attention_masks = input_iterator.get_next()
process of the model
   def forward(self,is_train = True):

        with tf.variable_scope(tf.get_variable_scope(), reuse = tf.AUTO_REUSE):
        self.coeff = networks.R_Net(self.imgs,is_training = is_train) self.Face3D.Reconstruc-
tion_Block(self.coeff,self.opt)

        self.id_labels = networks.Perceptual_Net(self.imgs)
        self.id_features = networks.Perceptual_Net(self.Face3D.render_imgs)

        self.photo_loss = losses.Photo_loss(self.imgs,self.Face3D.ren-
der_imgs,self.Face3D.img_mask_crop*self.attention_masks)

        self.landmark_loss  =  osses.Landmark_loss(self.Face3D.landmark_p,self.lm_labels)
        self.perceptual_loss = losses.Perceptual_loss(self.id_features,self.id_labels)

        self.reg_loss      =   losses.Regulation_loss(self.Face3D.id_coeff,self.Face3D.ex_co-
        eff , self .Face3D.tex_coeff,self.opt)
        self.reflect_loss  =   losses.Reflectance_loss(self.Face3D.face_tex-
        ture,self.Face3D.facemodel)
        self.gamma_loss =   losses.Gamma_loss(self.Face3D.gamma)

           self.loss = self.opt.w_photo*self.photo_loss + self.opt.w_lm*self.landmark_loss +
self.opt.w_id * self.perceptual_loss +  self.opt.w_reg * self.reg_loss + self.opt.w_ref *
self.reflect_loss + self.opt.w_gamma * self.gamma_loss
```

- **Training Parameters**

```
# Training parameters
        self.w_photo = 1.92
        self.w_lm = 1.6e-3
        self.w_id = 0.2
        self.w_reg = 3.0e-4
        self.w_ref = 5.0
        self.w_gamma = 10.0
        self.w_ex = 0.8
        self.w_tex = 1.7e-2
        self.batch_size = 16
        self.boundaries = [100000]
        lr = [1e-4,2e-5]
        self.global_step = tf.Variable(0,name='global_step',trainable = False)
        self.lr = tf.train.piecewise_constant(self.global_step,self.boundaries,lr)
        self.augment = True
        self.train_maxiter = 200000
        self.train_summary_iter = 50
        self.image_summary_iter = 200
        self.val_summary_iter = 1000
        self.save_iter = 10000
```

3.12.3. Loss Functions

- **Photo Loss**

```
def Photo_loss(input_imgs,render_imgs,img_mask):
    input_imgs = tf.cast(input_imgs,tf.float32)
    # img_mask = tf.squeeze(img_mask,3)
    img_mask = tf.stop_gradient(img_mask[:,:,:,0])
    # photo loss with skin attention
    photo_loss = tf.sqrt(tf.reduce_sum(tf.square(input_imgs - render_imgs),axis = 3))*img_mask/255
    photo_loss = tf.reduce_sum(photo_loss) / tf.maximum(tf.reduce_sum(img_mask),1.0)
    return photo_loss
```

- **Landmark Loss**

```
# landmark loss
# landmark_p and landmark_label are [batchsize, 68, 2] landmark projections for reconstruction images and input images
def Landmark_loss(landmark_p,landmark_label):
    # we set higher weights for landmarks around the mouth and nose regions
    landmark_weight = tf.concat([tf.ones([1,28]),20*tf.ones([1,3]),
tf.ones([1,29]),20*tf.ones([1,8])],axis = 1)
    landmark_weight = tf.tile(landmark_weight,[tf.shape(landmark_p)[0],1])
    landmark_loss = tf.reduce_sum(tf.reduce_sum(tf.square(landmark_p-
        landmark_label),2)*landmark_weight)/(68.0*tf.cast(tf.shape(landmark_
        p)[0],tf.float32))
    return landmark_loss
```

- **Perceptual loss**

```
    # perceptual loss
    # Id_feature and id_label are [batchsize, c] identity features for reconstruction
                                    images and input images
def Perceptual_loss(id_feature,id_label):
    id_feature = tf.nn.l2_normalize (id_feature, dim = 1)
    id_label = tf.nn.l2_normalize (id_label, dim = 1)
    # cosine similarity
    sim = tf.reduce_sum(id_feature*id_label,1)
    loss = tf.reduce_sum(tf.maximum(0.0,1.0 -
sim))/tf.cast(tf.shape(id_feature)[0],tf.float32)
    return loss
```

- **Regulation loss**

```
# coefficient regularization to ensure plausible 3d faces
def Regulation_loss(id_coeff,ex_coeff,tex_coeff,opt):
    w_ex = opt.w_ex
    w_tex = opt.w_tex
    regulation_loss = tf.nn.l2_loss(id_coeff) + w_ex * tf.nn.l2_loss(ex_coeff) + w_tex
* tf.nn.l2_loss(tex_coeff)
    regulation_loss = 2*regulation_loss/ tf.cast(tf.shape(id_coeff)[0],tf.float32)
    return regulation_loss
```

- **Gamma Loss**

```
# Gamma regularization to ensure a nearly-monochromatic light
def Gamma_loss(gamma):
    gamma = tf.reshape(gamma,[-1,3,9])
    gamma_mean = tf.reduce_mean(gamma,1, keep_dims = True)
    gamma_loss = tf.reduce_mean(tf.square(gamma - gamma_mean))
    return gamma_loss
```

- **Reflectance Loss**

```
# albedo regularization to ensure an uniform skin albedo
def Reflectance_loss(face_texture,facemodel):
    skin_mask = facemodel.skin_mask
    skin_mask = tf.reshape(skin_mask,[1,tf.shape(skin_mask)[0],1])
    texture_mean = tf.reduce_sum(face_texture*skin_mask,1)/tf.reduce_sum(skin_mask)
    texture_mean = tf.expand_dims(texture_mean,1)
    # minimize texture variance for pre-defined skin region
    reflectance_loss = tf.reduce_sum(tf.square((face_texture -
texture_mean)*skin_mask/255.0))/(tf.cast(tf.shape(face_texture)[0],tf.float32)*tf.red
uce_sum(skin_mask))
```

- **Train Method**

```
# main function for training
def train():
    # read BFM face model
    # transfer original BFM model to our model
    if not os.path.isfile('./BFM/BFM_model_front.mat'):
        transferBFM09 ()
    with tf.Graph().as_default() as graph:

        # training options
        args = parse_args()
        opt = Option(model_name=args.model_name)
        opt.data_path = [args.data_path]
        opt.val_data_path = [args.val_data_path]

        # load training data into queue
        train_iterator = load_dataset(opt)
        # create reconstruction model
        model = Reconstruction_model(opt)
        # send training data to the model
        model.set_input(train_iterator)
        # update model variables with training data
        model.step(is_train = True)
        # summarize training statistics
        model.summarize()
        # several training stattistics to be saved
        train_stat = model.summary_stat
        train_img_stat = model.summary_img
        train_op = model.train_op
```

```
        photo_error = model.photo_loss
        lm_error = model.landmark_loss
        id_error = model.perceptual_loss
        # load validation data into queue
        val_iterator = load_dataset(opt,train=False)
        # send validation data to the model
        model.set_input(val_iterator)
        # only do foward pass without updating model variables
        model.step(is_train = False)
        # summarize validation statistics
        model.summarize()
        val_stat = model.summary_stat
        val_img_stat = model.summary_img
        # initialization saver, train_writer,val_writer,
         sess = restore_weights_and_initialize(opt)
        # freeze the graph to ensure no new op will be added during training
        sess.graph.finalize()
        # training loop
        for i in range(opt.train_maxiter):
          _,ph_loss,lm_loss,id_loss = sess.run([train_op,photo_error,lm_error,id_error])
           print('Iter: %d; lm_loss: %f ; photo_loss: %f; id_loss:
%f\n'%(i,np.sqrt(lm_loss),ph_loss,id_loss))
           # summarize training stats every <train_summary_iter> iterations
           if np.mod(i,opt.train_summary_iter) == 0:
              train_summary = sess.run(train_stat)
              train_writer.add_summary(train_summary,i)
```

```
# summarize image stats every <image_summary_iter> iterations
        if np.mod(i,opt.image_summary_iter) == 0:
            train_img_summary = sess.run(train_img_stat)
            train_writer.add_summary(train_img_summary,i)

        # summarize validation stats every <val_summary_iter> iterations
        if np.mod(i,opt.val_summary_iter) == 0:
            val_summary,val_img_summary = sess.run([val_stat,val_img_stat])
            val_writer.add_summary(val_summary,i)
            val_writer.add_summary(val_img_summary,i)

        # save model variables every <save_iter> iterations
        if np.mod(i,opt.save_iter) == 0:
            saver.save(sess,os.path.join(opt.model_save_path,'iter_%d.ckpt'%i))

if __name__ == '__main__':
    train()
```

4. Conclusion

We have introduced DECA, which enables detailed expression capture and animation from single images by learning an animatable detail model from a dataset of in-the-wild images, on the other hand, we introduced a CNN-based single-image face reconstruction method that employs hybrid-level image information for weakly supervised learning without ground-truth 3D shapes. In total, DECA is trained from about 2M in-the-wild to face images without 2D-to-3D supervision. DECA achieves state-of-the-art shape reconstruction performance approved by a shape consistency loss.

General Conclusion

General Conclusion

In This Thesis, we provided a thorough overview of the technical background of the 3D reconstruction camera model with an optical lens, face-specific geometric models represented, next we deep dive into the fundamental of deep learning and deep neural networks.

We have discussed the CNN architecture by providing a different architecture and clarifying its limits. Next, we made a comparison between two methods that allow us to identify the weakness the performance of each them. The main difference between them is how does the training data is made and loss function that enable generating 3D face within low error reconstruction.

We discovered the relevant important functions constructed to design 3D reconstruction from single image and more precisely on the 3D face reconstruction.

Finally, we can conclude that 3D face reconstruction has been approached with many deep learning techniques successfully, but still can be further improved by incorporating alternatives solutions that would leverage the computation complexity related to real-world applications. Motivated by the promising results of this studies, we suggest to do an investigation on 3D generative model to alleviate the issues related to data generation issues and high computation resources.

Abstract

Abstract:

In This Thesis, we provided a thorough overview of the technical background of the 3D reconstruction camera model with an optical lens, face-specific geometric models represented, next we deep dive into the fundamental of deep learning and deep neural networks. We have discussed the CNN architecture by providing a different architecture and clarifying its limits. Next, we made a comparison between two methods that allow us to identify the weakness the performance of each them. The main difference between them is how does the training data is made and loss function that enable generating 3D face within low error reconstruction. We discovered the relevant important functions constructed to design 3D reconstruction from single image and more precisely on the 3D face reconstruction. Finally, we can conclude that 3D face reconstruction has been approached with many deep learning techniques successfully, but still can be further improved by incorporating alternatives solutions that would leverage the computation complexity related to real-world applications. Motivated by the promising results of this studies, we suggest to do an investigation on 3D generative model to alleviate the issues related to data generation issues and high computation resources.

Resume:

Dans cette thèse, nous avons fourni un aperçu complet de l'arrière-plan technique du modèle de caméra de reconstruction 3D avec une lentille optique, des modèles géométriques spécifiques au visage représentés, puis nous nous sommes penchés sur les principes fondamentaux de l'apprentissage en profondeur et des réseaux de neurones profonds. Nous avons discuté de l'architecture CNN en proposant une architecture différente et en clarifiant ses limites. Ensuite, nous avons fait une comparaison entre deux méthodes qui nous permettent d'identifier la faiblesse de la performance de chacune d'elles. La principale différence entre eux réside dans la manière dont les données d'entraînement sont créées et la fonction de perte qui permet de générer un visage 3D avec une reconstruction à faible erreur. Nous avons découvert les fonctions importantes pertinentes construites pour concevoir une reconstruction 3D à partir d'une seule image et plus précisément sur la reconstruction de visage 3D. Enfin, nous pouvons conclure que la reconstruction de visage 3D a été abordée avec succès avec de nombreuses techniques d'apprentissage en profondeur, mais peut encore être améliorée en incorporant des solutions alternatives qui tireraient parti de la complexité de calcul liée aux applications du monde réel. Motivés par les résultats prometteurs de ces études, nous suggérons de faire une enquête sur le modèle génératif 3D pour atténuer les problèmes liés aux problèmes de génération de données et aux ressources de calcul élevées.

ملخص:

في هذه الرسالة، قدمنا نظرة عامة شاملة على الخلفية التقنية لنموذج كاميرا إعادة الإعمار ثلاثية الأبعاد باستخدام عدسة بصرية ، ونماذج هندسية خاصة بالوجه ممثلة ، وبعد ذلك تعمقنا في أساسيات التعلم العميق والشبكات العصبية العميقة. لقد ناقشنا من خلال توفير بنية مختلفة وتوضيح حدودها. بعد ذلك ، قمنا بإجراء مقارنة بين طريقتين تسمحان لنا بتحديد ضعف CNN بنية أداء كل منهما. يتمثل الاختلاف الرئيسي بينهما في كيفية عمل بيانات التدريب ووظيفة الخسارة التي تمكن من إنشاء وجه ثلاثي الأبعاد ضمن إعادة بناء منخفضة للخطأ. اكتشفنا الوظائف المهمة ذات الصلة التي تم إنشاؤها لتصميم إعادة بناء ثلاثية الأبعاد من صورة واحدة وبشكل أكثر دقة على إعادة بناء الوجه ثلاثي الأبعاد. أخيرًا ، يمكننا أن نستنتج أنه تم التعامل مع إعادة بناء الوجه ثلاثي الأبعاد باستخدام العديد من تقنيات التعلم العميق بنجاح ، ولكن لا يزال من الممكن تحسينها أكثر من خلال دمج الحلول البديلة التي من شأنها الاستفادة من تعقيد الحساب المرتبط بتطبيقات العالم الحقيقي. بدافع من النتائج الواعدة لهذه الدراسات ، نقترح إجراء تحقيق في نموذج التوليد ثلاثي الأبعاد للتخفيف من المشكلات المتعلقة بقضايا إنشاء البيانات وموارد الحساب العالية.

Reference

[1] Y. C. H. Wang, “Pinhole SPECT with Dierent Data Acquisition Geometrics:Usefulness of Unied Projection Operators in Homogeneous Coordinates,”IEEE Transactions on Medical Imaging, vol. 26, no. 3, 2007.

[2] J. Blinn, “A Trip Down the Graphics Pipeline: Line Clipping,” IEEE Computer Graphics and Applications, p. 8, 1991.

[3]. 3D reconstruction from multiple images,Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/3D_reconstruction_from_multiple_images. (Accessed 6 August 2019)

[4]. Tamas Fazakas, Róbert Tamás Fekete, “3D reconstruction system for auton-omous robot navigation”, 2010 11th International Symposium on Computa-tional Intelligence and Informatics, Budapest (November 2010)

[5]. Gupta, Sharad & Shukla, Dericks. “Application of drone for landslide map-ping, dimension estimation and its 3D reconstruction.” Journal of the Indian Society of Remote Sensing. . (January 2018).

[6]. Steffen Herbort and Christian Wöhler “An introduction to image-based 3D surface reconstruction and a survey of photometric stereo methods”. (Sep-tember 2011) 23.

[7]. Joseph S Lappin, “What is Binocular Disparity. (August 2014) 1-4.

[8]. Nayar, S.K.; Nakagawa, Y. “Shape from Focus”, Pattern Analysis and Ma-chine Intelligence, IEEE Transactions on Volume 16, Issue 8. (1994) 824 – 831.

[9]. StereoKinetic phenomenon from Michael Bach’s "Optical Illusions & Visual Phenomena" (January 2013)

[10]. Parallax scrolling, Wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/Parallax scrolling. (Accessed 8 may 2019)

[11]. George Mather, The use of image blur as a depth cue: (February 1997)

[12]. Marshall J. A., Burbeck C. A., Ariely D., Rolland J. P., Martin K. E. “Occlu-sion edge blur: A cue to relative visual depth” , Journal of the Optical Socie-ty of America A, (1996). 13, 681–688.

[13]. Pilar Merchán, Antonio Adan, Santiago Salamanca, “Depth Gradient Image Based On Silhouette: A Solution for Reconstruction Of Scenes in 3D Envi-ronments”. (January 2006).

[14]. Xiaojun Wu, High-quality Software Development Through Collaborations with Major Universities in China, https://www.nttreview.jp/archive/ntttechnical.php?contents=ntr201110fa6.pdf&mode=show_pdf. (Accessed 11 may 2019)

[15]. Qingqing Wei, “Converting 2D to 3D: A Survey”. (Dec. 2005) 12-13.

[16]. Qingqing Wei, “Converting 2D to 3D: A Survey”. (Dec. 2005) 11-12.

[17]. Ahmad, M.B.; Tae-Sun Choi “Fast and accurate 3D shape from focus using dynamic programming optimization technique”, Proc. (ICASSP '05), IEEE International Conference on Acoustics, Speech, and Signal Processing Vol. 2. (2005) 969 – 972.

[18]. Cozman, F.; Krotkov, E. “Depth from scattering”, IEEE Computer society, conference on Computer Vision and Pattern Recognition, Proceedings. (1997) 801–806

[19]. Michael W. Tao, Pratul P. Srinivasan, Jitendra Malik, Szymon Rusinkiewicz and Ravi Ramamoorthi ,“Depth from Shading, Defocus, and Correspondence Using Light-Field Angular Coherence”, (june 2015) 1-2.

[20]. Li, M.; Magnor, M.; Seidel, H. P. “Hardware-Accelerated Visual Hull Re-construction and Rendering”, Proceedings of Graphics Interface 2003, Hali-fax, Canada (2003).
[21]. David G. Lowe, “Object Recognition from Local Scale-Invariant Features”, Proc. of the International Conference on Computer Vision, Corfu (Sept. 1999).

[22] Anil, J., and L. Padma Suresh. “Literature survey on face and face expression recognition.” Circuit, Power and Computing Technologies (ICCPCT), 2016 International Conference on. IEEE, 2016.

[23] Aloysius, N., & Geetha, M. (2017, April). A review on deep convolutional neural networks. In 2017 International Conference on Communication and Signal Processing (ICCSP) (pp. 0588-0592). IEEE.

[24] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, arXiv preprint arXiv:1207.0580, 2012.

[25] Hsieh CC, Hsih MH, Jiang MK, Cheng YM, Liang EH. Effective semantic features for facial expressions recognition using svm. Multimedia Tools and Applications. 2016 Jun 1;75(11):6663-82.

[26] Ramachandran R, Rajeev DC, Krishnan SG, P Subathra, Deep learning an overview, IJAER, Volume 10, Issue 10, 2015, Pages 25433-25448.

[27] M. Jaderberg, A. Vedaldi, and A. Zisserman, Deep features for text spotting, in ECCV, 2014.

[28] R. Zhao, W. Ouyang, H. Li, and X. Wang, Saliency detection by multicontext deep learning, in CVPR, 2015.

[29] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. “Deep learning.” Nature 521.7553 (2015): 436-444.

[30] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, Decaf: A deep convolutional activation feature for generic, 2014.

[31] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, Learning hierarchical features for scene labeling, PAMI, 2013.

[32] Nithin, D Kanishka and Sivakumar, P Bagavathi, Generic Feature Learning in Computer Vision, Elsevier, Vol.58, Pages202-209, 2015.

[33] Y. Guo, j. zhang, J. Cai, B. Jiang, and J. Zheng, "CNN-Based RealTime Dense Face Reconstruction with Inverse-Rendered PhotoRealistic Face Images," IEEE Trans. Pattern Anal. Mach. Intell., vol. 41,no.6,pp.1294-1307,Jun.2019.

[34] K. Fukushima, Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, Biological cybernetics, 1980. [21] A. X. Chang et al., "ShapeNet: An Information-Rich 3D Model Repository," CoRR, vol. abs/1512.0, 2015.

[35] B. B. Le Cun, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard,and L. D. Jackel, Handwritten digit recognition with a backpropagation network, in NIPS. Citeseer, 1990.

[36] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learningapplied to document recognition, Proceedings of the IEEE, 1998.

[37] Alex Krizhevsky, Sutskever I, and Hinton G.E, Imagenet classification with deep convolutional neural networks. In NIPS, 2012.

[38] A. Berg, J. Deng, and L. Fei-Fei, Large scale visual recognition challenge 2010, www.image-net.org/challenges. 2010.

[39] S. Vicente, J. Carreira, L. Agapito, and J. Batista. Reconstructing pascal voc. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 41–48, 2014.

[43] Y. Xiang et al., "ObjectNet3D: A Large Scale Database for 3D Object Recognition," in European Conference Computer Vision (ECCV), 2016.

[44] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond PASCAL: A Benchmark for 3D Object Detection in the Wild," in IEEE Winter Conference on Applications ofComputer Vision (WACV), 2014.

[45] A. Geiger, "Are We Ready for Autonomous Driving? The KlTTl Vision Benchmark Suite," in Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012,pp.3354-3361 .

[46] D. Vlasic, M. Brand, H. Pfister, and J. Popovi´c. Face transfer with multilinear models. ACM transactions on graphics (TOG), 24(3):426–433, 2005.

[47] C. Cao, Y. Weng, S. Lin, and K. Zhou. 3d shape regression for real-time facial animation. ACM Transactions on Graphics (TOG), 32(4):41, 2013.

[48] C. Cao, Q. Hou, and K. Zhou. Displaced dynamic expression regression for real-time facial tracking and animation. ACM Transactions on graphics (TOG), 33(4):43, 2014.

[49] S. Saito, T. Li, and H. Li. Real-time facial segmentation and performance capture from rgb input. In European Conference on Computer Vision (ECCV), pages 244–261, 2016.

[50] T. Hassner and R. Basri. Example based 3d reconstruction from single 2d images. In IEEE Conference on Computer Vision and Pattern Recognition Workshop, 2006.

[51] I. Kemelmacher-Shlizerman and S. M. Seitz. Face reconstruction in the wild. In International Conference on Computer Vision (ICCV), pages 1746–1753, 2011.

[51] T. Hassner. Viewing real-world faces in 3d. In International Conference on Computer Vision (ICCV), pages 3607–3614, 2013.

[52] E. Richardson, M. Sela, and R. Kimmel. 3d face reconstruction by learning from synthetic data. In International Conference on 3D Vision (3DV), pages 460–469, 2016.

[53] P. Dou, S. K. Shah, and I. A. Kakadiaris. End-to-end 3d face reconstruction with deep neural networks. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 21–26, 2017.

[54] A. Bas, P. Huber, W. A. Smith, M. Awais, and J. Kittler. 3d morphable models as spatial transformer networks. In International Conference on Computer Vision Workshop on Geometry Meets Deep Learning, pages 904–912, 2017.

[55] A. Tewari, M. Zollh¨ofer, H. Kim, P. Garrido, F. Bernard, P. P´erez, and C. Theobalt. MoFa: model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In International Conference on Computer Vision (ICCV), pages 1274–1283, 2017.

[56] A. T. Tran, T. Hassner, I. Masi, and G. Medioni. Regressing robust and discriminative 3d morphable models with a very deep neural network. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1493–1502, 2017.

[57] K. Genova, F. Cole, A. Maschinot, A. Sarna, D. Vlasic, and W. T. Freeman. Unsupervised training for 3d morphable model regression. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.

[58] Feng, Y., Feng, H., Black, M. J., & Bolkart, T. (2020). Learning an Animatable Detailed 3D Face Model from In-The-Wild Images. arXiv preprint arXiv:2012.04012.

[59] Chen, A., Chen, Z., Zhang, G., Mitchell, K., Yu, J.: Photo-realistic facial details synthesis from single image. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 9429{9439 (2019).

[60] Dou, P., Shah, S.K., Kakadiaris, I.A.: End-to-end 3d face reconstruction with deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5908{5917 (2017).

[61] Feng, Y., Wu, F., Shao, X., Wang, Y., Zhou, X.: Joint 3d face reconstruction and dense alignment with position map regression network. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 534{551 (2018).

[62] Galteri, L., Ferrari, C., Lisanti, G., Berretti, S., Del Bimbo, A.: Deep 3d morphable model refinement via progressive growing of conditional generative adversarial networks. Computer Vision and Image Understanding 185, 31{42 (2019).

[63] Guo, Y., Cai, J., Jiang, B., Zheng, J., et al.: Cnn-based real-time dense face reconstruction with inverse-rendered photo-realistic face images. IEEE transactions on pattern analysis and machine intelligence 41(6), 1294{1307 (2018).

[64] Jackson, A.S., Bulat, A., Argyriou, V., Tzimiropoulos, G.: Large pose 3d face reconstruction from a single image via direct volumetric cnn regression. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1031-1039 (2017).

[65] Liu, F., Zhu, R., Zeng, D., Zhao, Q., Liu, X.: Disentangling features in 3d face shapes for joint face reconstruction and recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5216{5225 (2018).

[66] Richardson, E., Sela, M., Kimmel, R.: 3d face reconstruction by learning from synthetic data. In: 2016 Fourth International Conference on 3D Vision (3DV). pp.460{469. IEEE (2016).

[67] Tran, A.T., Hassner, T., Masi, I., Paz, E., Nirkin, Y., Medioni, G.G.: Extreme 3d face reconstruction: Seeing through occlusions. In: CVPR. pp. 3935{3944 (2018).

[68] Tran, L., Liu, F., Liu, X.: Towards high-delity nonlinear 3d face morphable model. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1126{1135 (2019).

[69] Tuan Tran, A., Hassner, T., Masi, I., Medioni, G.: Regressing robust and discriminative 3d morphable models with a very deep neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5163{5172 (2017).

[70] Yi, H., Li, C., Cao, Q., Shen, X., Li, S., Wang, G., Tai, Y.W.: Mmface: A multimetric regression network for unconstrained face reconstruction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7663{7672 (2019).

[71] Zeng, X., Peng, X., Qiao, Y.: Df2net: A dense-ne- dense-fine-finer network for detailed 3d face reconstruction. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2315{2324 (2019).

[72] Zhu, X., Lei, Z., Liu, X., Shi, H., Li, S.Z.: Face alignment across large poses: A 3d solution. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 146{155 (2016).

[73] A. D. Bagdanov, A. Del Bimbo, and I. Masi. The Florence 2d/3d hybrid face dataset. In The Joint ACM Workshop on Human Gesture and Behavior Understanding, pages 79–80, 2011.

[74] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou. Facewarehouse: A 3d facial expression database for visual computing. IEEE Transactions on Visualization and Computer Graphics (TVCG), 20(3):413–425, 2014.

[75] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato. A 3d facial expression database for facial behavior research. In International Conference on Automatic Face and Gesture Recognition, pages 211–216, 2006.

[76] M. Sela, E. Richardson, and R. Kimmel. Unrestricted facial geometry reconstruction using image-to-image translation. In IEEE International Conference on Computer Vision (ICCV), pages 1585–1594, 2017.

[77] L. Tran and X. Liu. Nonlinear 3d face morphable model. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 7346–7355, 2018.

[78] A. Tewari, M. Zollh¨ofer, P. Garrido, F. Bernard, H. Kim, P. P´erez, and C. Theobalt. Self-supervised multi-level face model learning for monocular reconstruction at over 250 hz. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2549–2559, 2018.

[79] Y. Feng, F. Wu, X. Shao, Y. Wang, and X. Zhou. Joint 3d face reconstruction and dense alignment with position map regression network. In European Conference on Computer Vision (ECCV), 2018.

[80] M. Piotraschke and V. Blanz. Automated 3d face reconstruction from multiple images using quality measures. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3418–3427, 2016.

[81] Michael Nielsen. Neural Networks and Deep Learning. Determination Press, 2015. [Online; accessed 30-July-2016].

[82] Dmytro Mishkin, Nikolay Sergievskiy, and Jiri Matas. Systematic evaluation of cnn advances on the imagenet. arXiv preprint arXiv:1606.02228, 2016.

[83] Ichim, A. E., Bouaziz, S., & Pauly, M. (2015). Dynamic 3d avatar creation from hand-held video input. ACM Transactions on Graphics (ToG), 34(4), 1-14.

[84] SUWAJANAKORN, S., KEMELMACHER-SHLIZERMAN, I. AND SEITZ, S. M. (2014): Total Moving Face Reconstruction. In FLEET, D., PAJDLA, T., SCHIELE, B. AND TUYTELAARS, T., EDITORS: ECCV Volume 8692,, ISBN 978–3–319–10592–5, 796–812

[85] PIOTRASCHKE, M. AND BLANZ, V. (2016): Automated 3D Face Reconstruction from Multiple Images Using Quality Measures. In CVPR, 3418–3427

[86] KLAUDINY, M., MCDONAGH, S., BRADLEY, D., BEELER, T. AND MITCHELL, K. (2017): Real-Time Multi-View Facial Capture with Synthetic Training. Computer Graphics Forum (Proceedings of Eurographics), 36 (2), 325–336, ISSN 1467–8659.

[87] ROTH, J., TONG, Y. T. AND LIU, X. (2017): Adaptive 3D Face Reconstruction from Unconstrained Photo Collections. IEEE TPAMI, 39 (11), 2127–2141, ISSN 0162–8828.

[88] DUONG, C. N., LUU, K., QUACH, K. G. AND BUI, T. D. (2016): Deep Appearance Models: A Deep Boltzmann Machine Approach for Face Modeling., arXiv: 1607.06871 hURL: https: //arxiv.org/abs/1607.06871

[89] MEDINA, J. ALABORT-I AND ZAFEIRIOU, S. (2017): A Unified Framework for Compositional Fitting of Active Appearance Models. IJCV, 121 (1), 26–64, ISSN 1573–1405.

[90] CAO, C., WENG, Y., LIN, S. AND ZHOU, K. (2013): 3D Shape Regression for Real-time Facial
Animation. ACM ToG, 32 (4), 41:1–10, ISSN 0730–0301.

[91] GARRIDO, P., VALGAERTS, L., WU, C. AND THEOBALT, C. (2013): Reconstructing detailed dynamic face geometry from monocular video. ACM ToG, 32 (6), 158:1–10 hURL: http://gvv.mpi-inf.mpg.de/projects/MonFaceCap/i, ISSN 0730–0301.

[92] GARRIDO, P., ZOLLHÖFER, M., CASAS, D., VALGAERTS, L., VARANASI, K., PÉREZ, P. AND THEOBALT, C. (2016): Reconstruction of Personalized 3D Face Rigs from Monocular Video. ACMToG, 35 (3), 28:1–15 hURL: http://gvv.mpi-inf.mpg.de/projects/PersonalizedFaceRig.

[93] THOMAS, D. AND TANIGUCHI, R. I. (2016): Augmented Blendshapes for Real-Time Simultaneous 3D Head Modeling and Facial Motion Capture. In CVPR, 3299–3308.

[94] SHI, F., WU, H.-T., TONG, X. AND CHAI, J. (2014): Automatic Acquisition of High-fidelity Facial Performances Using Monocular Videos. ACM ToG, 33 (6), 222:1–13, ISSN 0730–0301.

[95] TRAN, A. T., HASSNER, T., MASI, I. AND MEDIONI, G. (2017): Regressing Robust and Discriminative 3D Morphable Models with a very Deep Neural Network. In CVPR hURL: https://www.openu.ac.il/home/hassner/projects/CNN3DMM/i, 1493–1502.

[96] RICHARDSON, E., SELA, M. AND KIMMEL, R. (2016): 3D Face Reconstruction by Learning from Synthetic Data. In 3DV, 460–469.

[97] CRISPELL, D. AND BAZIK, M. (2017): Pix2face: Direct 3D Face Model Estimation. In ICCV Workshops

[98] DOU, P., SHAH, S. K. AND KAKADIARIS, I. A. (2017): End-to-end 3D face reconstruction with deep neural networks. In CVPR

[99] SCHÖNBORN, S., EGGER, B.,MOREL-FORSTER, A. AND VETTER, T. (2017): Markov Chain Monte Carlo for Automated Face Image Analysis. IJCV, 123 (2), 160–183, ISSN 1573–1405

[100] GUO, Y., ZHANG, J., CAI, J., JIANG, B. AND ZHENG, J. (2017): 3DFaceNet: Real-time Dense
Face Reconstruction via Synthesizing Photo-realistic Face Images., arXiv:1708.00980 hURL: https://arxiv.org/abs/1708.00980

[101] TEWARI, A., ZOLLHÖFER, M., KIM, H., GARRIDO, P., BERNARD, F., PÉREZ, P. AND THEOBALT,
C. (2017): MoFA: Model-based Deep Convolutional Face Autoencoder for Unsupervised Monocular Reconstruction. In ICCV hURL: http://gvv.mpi-inf.mpg.de/projects/MZ/Papers/ArXiv2017_FA/page.htmli, 3735–3744

[102] RICHARDSON, E., SELA, M., OR-EL, R. AND KIMMEL, R. (2017): Learning Detailed Face Reconstruction from a Single Image. In CVPR , ISSN 1063–6919, 5553–5562

[103] RICHARDSON, E., SELA, M., OR-EL, R. AND KIMMEL, R. (2017): Learning Detailed Face Reconstruction from a Single Image. In CVPR , ISSN 1063–6919, 5553–5562

[104] LAINE, S., KARRAS, T., AILA, T., HERVA, A., SAITO, S., YU, R., LI, H. AND LEHTINEN, J. (2017): Production-level Facial Performance Capture Using Deep Convolutional Neural Networks. In Proceedings of the Symposium on Computer Animation (SCA) , ISBN 978–1–4503–5091–4, 10:1–10

[105] JIANG, L., ZHANG, J., DENG, B., LI, H. AND LIU, L. (2017): 3DFace Reconstruction with Geometry Details from a Single Image., arXiv:1702.05619 hURL: https://arxiv.org/abs/1702.05619i

[106] ROTH, J., TONG, Y. T. AND LIU, X. (2017): Adaptive 3D Face Reconstruction from Unconstrained Photo Collections. IEEE TPAMI, 39 (11), 2127–2141, ISSN 0162–8828

[107] SELA,M., RICHARDSON, E. AND KIMMEL, R. (2017): Unrestricted Facial Geometry Reconstruction Using Image-to-Image Translation. In ICCV, 1585–1594.

[108] CAO, C., BRADLEY, D., ZHOU, K. AND BEELER, T. (2015): Real-time High-fidelity Facial Performance Capture. ACM ToG, 34 (4), 46:1–9, ISSN 0730–0301

[109] GARRIDO, P., ZOLLHÖFER, M., CASAS, D., VALGAERTS, L., VARANASI, K., PÉREZ, P. AND THEOBALT, C. (2016): Reconstruction of Personalized 3D Face Rigs from Monocular Video. ACMToG, 35 (3), 28:1–15 hURL: http://gvv.mpi-inf.mpg.de/projects/PersonalizedFaceRig/

[110] RICHARDSON, E., SELA, M., OR-EL, R. AND KIMMEL, R. (2017): Learning Detailed Face Reconstruction from a Single Image. In CVPR , ISSN 1063–6919, 5553–5562

[111] JIANG, L., ZHANG, J., DENG, B., LI, H. AND LIU, L. (2017): 3DFace Reconstruction with Geometry Details from a Single Image., arXiv:1702.05619 hURL: https://arxiv.org/abs/1702.05619i

[112] ROTH, J., TONG, Y. T. AND LIU, X. (2017): Adaptive 3D Face Reconstruction from Unconstrained Photo Collections. IEEE TPAMI, 39 (11), 2127–2141, ISSN 0162–8828
[113] SELA,M., RICHARDSON, E. AND KIMMEL, R. (2017): Unrestricted Facial Geometry Reconstruction Using Image-to-Image Translation. In ICCV, 1585–1594

[114] BLANZ, V. AND VETTER, T. (1999): A Morphable Model for the Synthesis of 3D Faces. In SIGGRAPH, ISBN 0–201–48560–5, 187–194

[115] Deng, Y., Yang, J., Xu, S., Chen, D., Jia, Y., & Tong, X. (2019). Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (pp. 0-0).

[116] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3d face model for pose and illumination invariant face recognition. In IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS), pages 296–301, 2009.

[117] Y. Guo, J. Z. Zhang, J. Cai, B. Jiang, and J. Zheng. Cnn-based real-time dense face reconstruction with inverserendered photo-realistic face images. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2018.

[118] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou. Facewarehouse: A 3d facial expression database for visual computing. IEEE Transactions on Visualization and Computer Graphics (TVCG), 20(3):413–425, 2014.

[119] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 770–778, 2016.

[119] LEWIS, J. P., ANJYO, K., RHEE, T., ZHANG, M., PIGHIN, F. AND DENG, Z. (2014): Practice and Theory of Blendshape Facial Models. In LEFEBVRE, S. AND SPAGNUOLO, M., EDITORS: Eurographics 2014 - State of the Art Reports, 199–218

[120] Oswald Aldrian and William AP Smith. 2013. Inverse Rendering of Faces with a 3D Morphable Model. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 35, 5 (2013), 1080–1093.

[121] Anil Bas, William A. P. Smith, Timo Bolkart, and Stefanie Wuhrer. 2017. Fitting a 3D Morphable Model to Edges: A Comparison Between Hard and Soft Correspondences. In Asian Conference on Computer Vision Workshops. 377–391.

[122] Volker Blanz, Sami Romdhani, and Thomas Vetter. 2002. Face identification across different poses and illuminations with a 3D morphable model. In International

[123] Conference on Automatic Face & Gesture Recognition (FG). 202–207. Volker Blanz and Thomas Vetter. 1999. A morphable model for the synthesis of 3D faces. In SIGGRAPH. 187–194.

[124] Thomas Gerig, Andreas Morel-Forster, Clemens Blumer, Bernhard Egger, Marcel Luthi, Sandro Schönborn, and Thomas Vetter. 2018. Morphable face models-an open framework. In International Conference on Automatic Face & Gesture Recognition (FG). 75–82.

[125] Sami Romdhani and Thomas Vetter. 2005. Estimating 3D shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 2. 986–993.

[126] Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. 2016. Face2Face: Real-time face capture and reenactment of RGB videos. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2387–2395.

[127] Feng-Ju Chang, Anh Tuan Tran, Tal Hassner, Iacopo Masi, Ram Nevatia, and Gerard Medioni. 2018. ExpNet: Landmark-free, deep, 3D facial expressions. In International Conference on Automatic Face & Gesture Recognition (FG). 122–129.

[128] Yu Deng, Jiaolong Yang, Sicheng Xu, Dong Chen, Yunde Jia, and Xin Tong. 2019. Accurate 3D Face Reconstruction with Weakly-Supervised Learning: From Single Image to Image Set. In Computer Vision and Pattern RecognitionWorkshops. 285–295.

[129] Kyle Genova, Forrester Cole, Aaron Maschinot, Aaron Sarna, Daniel Vlasic, and William T. Freeman. 2018. Unsupervised Training for 3D Morphable Model Regression. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 8377–8386.

[130] Hyeongwoo Kim, Michael Zollhöfer, Ayush Tewari, Justus Thies, Christian Richardt, and Christian Theobalt. 2018b. InverseFaceNet: Deep Monocular Inverse Face Rendering. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 4625–4634.

[131] Stylianos Ploumpis, Evangelos Ververas, Eimear O'Sullivan, Stylianos Moschoglou, Haoyang Wang, Nick Pears, William Smith, Baris Gecer, and Stefanos P Zafeiriou. 2020. Towards a complete 3D morphable model of the human head. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) (2020).

[132] E. Richardson, M. Sela, and R. Kimmel. 2016. 3D Face Reconstruction by Learning from Synthetic Data. In International Conference on 3D Vision (3DV). 460–469.

[133] Soubhik Sanyal, Timo Bolkart, Haiwen Feng, and Michael Black. 2019. Learning to Regress 3D Face Shape and Expression from an Image without 3D Supervision. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 7763–7772.

[134] Ayush Tewari, Michael Zollhöfer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Perez, and Christian Theobalt. 2017. MoFA: Model-Based Deep Convolutional Face Autoencoder for Unsupervised Monocular Reconstruction. In IEEE International Conference on Computer Vision (ICCV). 1274–1283.

[135] Anh Tuan Tran, Tal Hassner, Iacopo Masi, and Gerard Medioni. 2017. Regressing Robust and Discriminative 3D Morphable Models With a Very Deep Neural Network. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 1599–1608.

[136] Xiaoguang Tu, Jian Zhao, Zihang Jiang, Yao Luo, Mei Xie, Yang Zhao, Linxiao He, Zheng Ma, and Jiashi Feng. 2019. Joint 3D Face Reconstruction and Dense Face Alignment from A Single Image with 2D-Assisted Self-Supervised Learning. IEEE International Conference on Computer Vision (ICCV) (2019).

[137] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. 2009. A 3D face model for pose and illumination invariant face recognition. In International Conference on Advanced Video and Signal Based Surveillance. 296–301.

[138] Ayush Tewari, Michael Zollhoefer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Perez, and Theobalt Christian. 2017. MoFA: Model-based Deep Convolutional Face Autoencoder for Unsupervised Monocular Reconstruction. In Proc. International Conference on Computer Vision (ICCV).

[139] Results submitted to participate in the NoW challenge are kept confidential https://ringnet.is.tue.mpg.de/challenge