Neural Networks in 3D Image Processing

Student A
Faculty of Electrical Engineering and Informatics
Technical University of Kosice
Letná 9, 042 00 Kosice

Student C
Faculty of Electrical Engineering and Informatics
Technical University of Kosice
Letná 9, 042 00 Kosice

Student B
Faculty of Electrical Engineering and Informatics
Technical University of Kosice
Letná 9, 042 00 Kosice

Student D
Faculty of Electrical Engineering and Informatics
Technical University of Kosice
Letná 9, 042 00 Kosice

***Abstract*—In this review paper we are going to discusses the advancements in machine learning and graphics processing technologies that have led to an increase in the use of deep learning models. We provide a brief description of different 3D graphics processing methods and afterwards the most commonly used method theses days. In this paper, we will also review significant research in the field of 3D medical imaging analysis using 3D CNNs in different medical areas such as classification, segmentation, detection, and localization. The paper concludes by discussing the challenges associated with the use of 3D CNNs in the medical imaging domain, how far we came in graphics processing and possible future trends in the field.**

***Index Terms*—Deferred Neural Rendering, Neural Point-Based Graphic, Implicit Differentiable Rendering, 3D CNNs**

I. Introduction

It's interesting to know that the rapid advancements in machine learning, graphics processing technologies and the availability of medical imaging data have led to a rapid increase in the use of deep learning models in the medical domain. In recent years, 3D CNNs have been employed for the analysis of medical images [1], but combining it with neural rendering could be a turning point, since we would be able to provide more angles for the learning and detection. In this paper, we will trace the previous methods on the 3D graphics processing, how it was developed from its machine learning roots and provide a brief description of 3D CNN.
Some of the previous works in neural rendering are:
- The deferred neural rendering [2] Section 2.A
- The neural point-based graphics [3] Section 2.B
- The implicit differentiable rendering [4] Section 2.C

These implementations use a neural network to compute pixel values in the screen space and not the texture space, thus computing only visible pixels. In all of these implementations, we assume that the scene is static or that our viewing distance and perspective are not totally free in the 3D space.

Medical imaging technologies such as computed tomography, magnetic resonance imaging, and X-rays have revolutionized the field of medicine by enabling non-invasive and highly accurate diagnosis of various diseases. However, interpreting and analyzing these complex medical images can be a challenging and also is a time-consuming task, even for experienced radiologists. That is why 3D deep learning comes in as a powerful tool for automated image analysis and interpretation as it could save us a lot of time. 3D deep learning is a sub-field of artificial intelligence that involves the use of deep neural networks to extract features and patterns from 3D medical images. In this review, we will explore the current state-of-the-art techniques in 3D deep learning for medical image analysis and discuss their potential applications in clinical practice. We will also examine the challenges and limitations of these techniques, since one of the biggest drawbacks in using CNNs is being hard to interpret the decision process.

II. Related Work

A. The deferred neural rendering

This approach learns the deferred rendering process directly from real images without needing any expert knowledge and allows us to resynthesize novel views of an object, edit scenes and synthesize animations. It proposes a unified neural rendering pipeline based on video footage of a target object we estimate its 3D geometry. This geometry is coarse and not suited to be processed by the standard graphics pipeline to achieve photo-realistic images. It's also able to learn specific components of the rendering pipeline such that we can generate new images based on course geometry. Given a 3D model, we parametrize its surface using a UV atlas, then using the standard rasterizer, we draw corresponding UV maps of the object to the image space.

This UV map is input to the method, and the pipeline consists of learnable neural textures and a learnable renderer. Instead of classical textures, this neural texture can contain learned features which are interpreted by the rendering network. Afterward, the UV maps from the rasterizer are used to sample from the neural textures and a small unit is used to translate these features to color values. This pipeline can

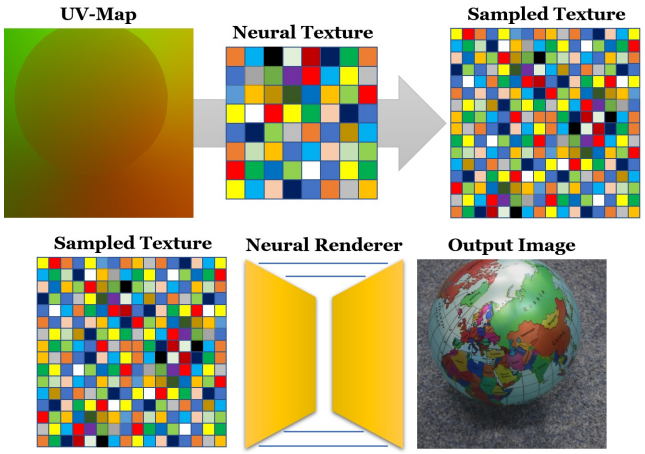

Fig. 1. Overview of the neural rendering pipeline: Given an object with a valid UV-map parameters and an associated Neural Texture map as input, the standard graphics pipeline is used to render a view-dependent screen-space feature map. The screen space feature map is then converted to photo-realistic imagery based on a Deferred Neural Renderer. [2]

be trained end to end given ground truth pairs of UV and color maps from the original object. Since the neural textures are attached to the object surface we are able to edit a scene. Given the input sequence and the 3D reconstruction, we can easily duplicate objects in a scene, and using this modified geometry we rasterize the corresponding UV map, which is an input to our rendering pipeline.

Neural textures are also a powerful tool that can enable animation synthesis. To this end, we can demonstrate it on facial reenactment given a source and a target actor, where we reconstruct the facial geometry and transfer the expressions from the source to the target mesh. The modified UV map, as well as a background image from the target video, will be used as input for the rendering approach.

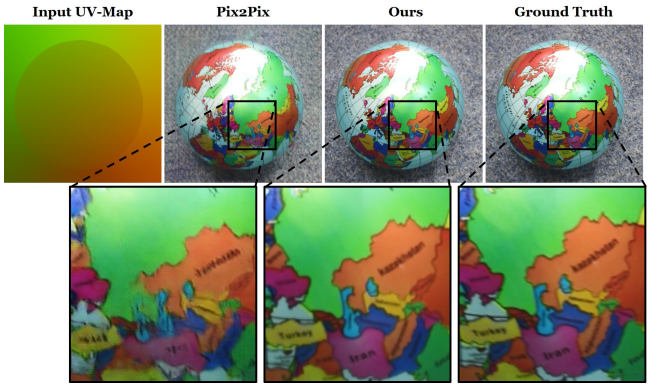

Fig. 2. Comparison to the image-to-image translation approach Pix2Pix. As can be seen, the novel views synthesized by the deferred neural rendering is higher quality, e.g. less blurry and results are close to the ground truth.

The rendering quality is dependent on the resolution of the texture. A texture at a single resolution might be too coarse or too fine, which leads to over-fitting and under-sampling during training. Because of this, we are better off using a hierarchy of textures resulting in better sampling behavior, particularly for higher texture resolutions. The number of training images also influences the rerendering quality. In particular, a view-dependent effects are not well captured with a reduced set of images, such as specular highlights slowly degrade.

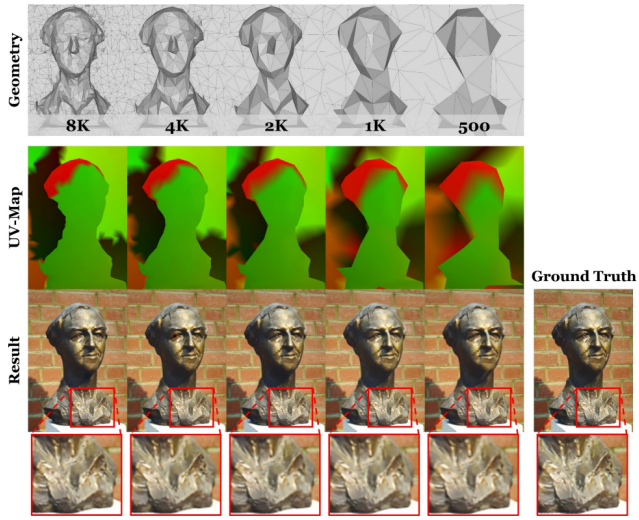

Fig. 3. The resolution of the underlying geometry proxy. Using quadric edge collapse we gradually reduce the number of triangles of the geometry from 8000 to 500. Even with the lowest resolution, a photo-realistic image can be generated. The MSE measured on the test sequence increases from 11.068 (8K), 11.742 (4K), 12.515 (2K), 18.297 (1K) to 18.395 for a proxy mesh with 500 triangles (MSE w.r.t. color channels in [0, 255]) [2]

B. *The neural point-based graphic*

This approach allows you to capture diverse static 3D scenes from image data and render them from new viewpoints. It also uses point clouds to represent scene geometry and a raw point cloud as the geometric representation of a scene. Such point clouds can be obtained by stereo-matching registered RGB image sets. Alternatively, we can obtain them by fusing depth maps from registered RGBD videos. Even with a noisy and incomplete underlying point cloud, this approach can generate complete and realistic views for novel viewpoints. To facilitate rendering from new viewpoints it assigns each point a neural descriptor of low dimensionality, where the descriptor contains information about local photo-metric and geometric properties. (For the experimentation were used eight-dimensional descriptors.)

To render a new view, we can project points onto the novel view at several resolutions using descriptor values as pseudo color. Afterward, we use a rendering convolutional network to transform the multi-resolution rendering into a photo-realistic image, and to capture the appearance of a scene we will fit the descriptors and the rendering network parameters to a given set of views. The fitting uses back-propagation and minimizes the loss between the rendered views and the ground truth frames.

When compared to the neural rendering system, which uses a mesh rather than a point cloud, is a geometric proxy. In general, both methods achieve similar results, but the neural

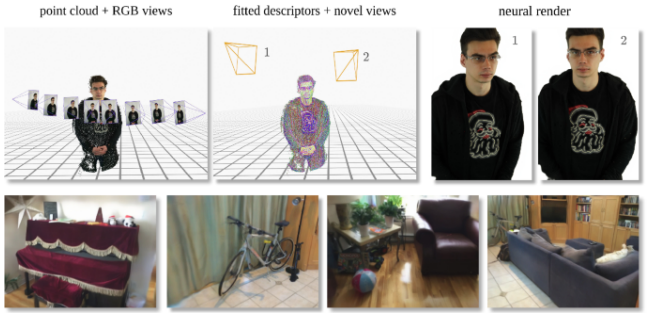

Fig. 4. Given a set of RGB views and a point cloud (top-left), this approach fits a neural descriptor to each point (top-middle), after which new views of a scene can be rendered (top-right). The method works for a variety of scenes including 3D portraits (top) and interiors (bottom). [3]

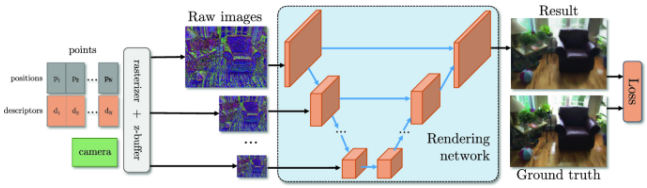

Fig. 5. An overview of our system. Given the point cloud P with neural descriptors D and camera parameters C, we rasterize the points with z-buffer at several resolutions, using descriptors as pseudo-colors. We then pass the rasterizations through the U-netlike rendering network to obtain the resulting image. Our model is fit to new scene(s) by optimizing the parameters of the rendering network and the neural descriptors by back-propagating the perceptual loss function. [3]

point-based approach performs better whenever meshing fails. For example, thin object parts are a problem for the mesh (Figure 6), whereas a point-based solution does not have this problem. Although when the images in the training set suffer from inconsistent exposure or have inaccurate registration point-based method can suffer from temporal flickering. This effect can be alleviated by applying temporal smoothing and spatial anti-aliasing to the rendering of the point cloud before they are processed by the rendering network.

Among other objects, we assess the performance of this method on human portraits. We can combine two scenes into a single one and for that, we fit both of them while keeping the coefficients of the rendering network the same, we then merge the point clouds after rough geometric alignment. The rendering network can then be applied to the combined point cloud.

1) The accelerated neural point-based graphic: (NPBG++) [5] It is built upon the Neural Point-Based Graphics method and improves it in several significant ways. Fundamentally, it lifts the limitation of pre-scene optimization by directly predicting the neural features from the input images and processing the source views one at a time by an alignment to a canonical orientation, followed by feature extraction, obtaining features that will be used to update the state of the point cloud. The input image alignment ensures, that the features are consistent since the feature extractor is not rotation-equivariant by default.

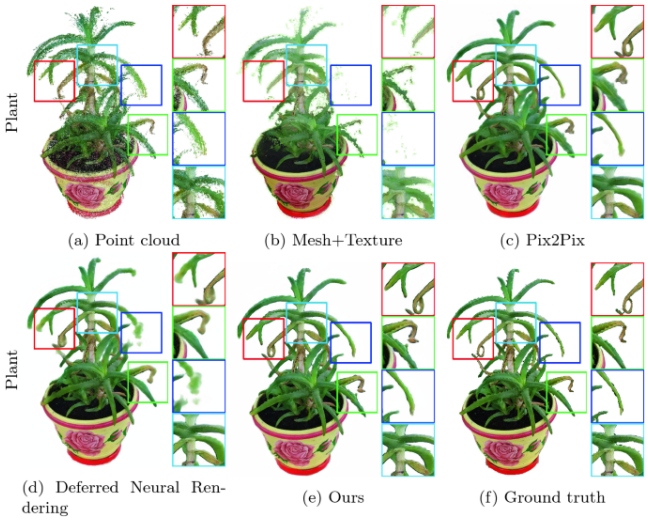

Fig. 6. Comparative results on the holdout frame from the 'Plant' scene. The neural point-based method better preserves thin parts of the scene, while the deferred neural rendering method blurs or leaves out this detail.

Once all of the views have been processed, we finalize the aggregation procedure which will be described next. The procedure results in view-dependent descriptors which can be used to synthesize new views following the rendering scheme from Neural Point-Based Graphics. The resulting image is rotated from the canonical orientation to the target view alignment, yielding the final output. The system is end-to-end trainable, resulting in a model that can easily adapt to new scenes, without performing additional optimization.

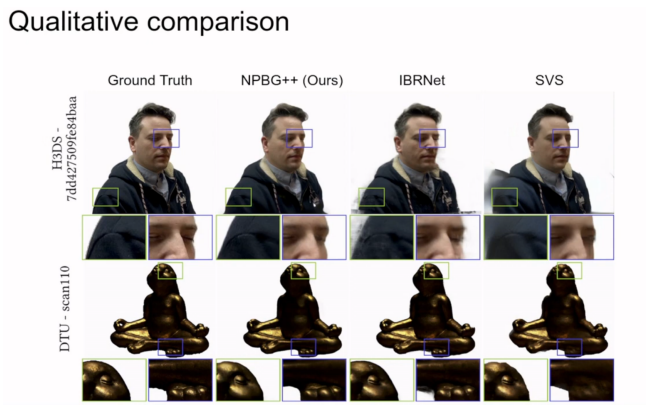

Fig. 7. Comparative results on multiple methods of rendering, among which we can see that NPBG++ is the superior one in terms of details and sharpness.

C. The implicit differentiable rendering (IDR)

This is a neural network architecture, that can learn three unknowns (Geometry, Appearance, Camera parameters) simultaneously and can produce high-fidelity 3D surface reconstruction, by disentangling geometry and appearance, learned solely from masked 2D images and rough camera estimations. We achieve this by processing these three unknowns with a differentiable rendering system. That means, simulating the

rendering of the scene from a given view, and comparing the rendered image to the original image.

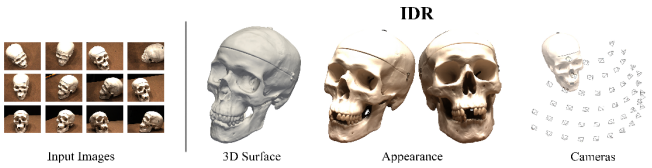

Figure 1: We introduce IDR: end-to-end learning of geometry, appearance and cameras from images.

Fig. 8. Learning of geometry, appearance and cameras from images.

The key challenge is how to make the renderer a function of the geometry, that is independent of the object material or the scene lighting. A recent work by Niemeyer et al. [7] introduced such a differential renderer model that can represent arbitrary texture, but cannot handle reflectance and lighting effects, nor can it handle noisy cameras. In this work, we reconstruct an implicit surface together with a surface light field that is separated from the geometry and we can handle both exact and noisy camera information. The geometry is represented as the zero level set of a neural network f, which model each 3D point its sign distance function to the shape. Given a learnable camera position and some fixed image pixels, we would like to produce differentiable RGB values. The camera orientation and pixel define a viewing direction, and we can trace the first intersection of the viewing ray with the implicit surface.

The first step is to represent the intersection point and its normal as differential functions of the implicit geometry and the camera parameters. We implement it by simply composing the neural network with a fixed linear computation at its entrance and another at its output.

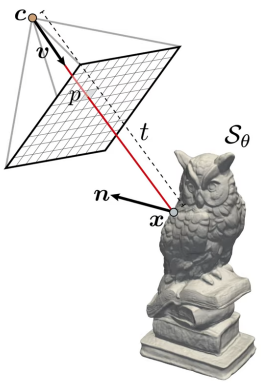

Fig. 9. Notation.

Secondly, we want to approximate the color of the pixel, determined by the radiance reflected from the surface to the camera. Our interest is that the renderer will not memorize any part of the geometry or the viewing cameras. Therefore, we suggest representing the light field as a function of the surface position, surface normal, and viewing direction. This is implemented using a second neural network to output RGB values. Incorporating the normal and the viewing direction is necessary for 3D reconstruction that is decoupled from the appearance and the cameras.

This claim can be explained using this simple example (Figure 10). A rendered without normal will produce the same light estimation in those two cases (Figure 10 *a, b*), therefore can result in a renderer that compensates over geometry properties. A renderer without viewing direction will produce the same light estimation in those two cases (Figure 10 *a, c*), wherein real-life the scene appearance is likely to be view-dependant.To complete the IDR model, we can use the global feature vector of the geometry, to get our final radiance approximation. Which is then compared to the ground truth pixel color, to simultaneously train the model's parameters.

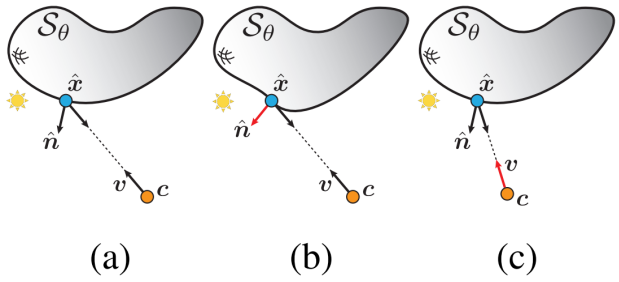

Fig. 10. Neural renderers without n and/or v are not universal.

III. Three-Dimensional Deep Learning

A typical CNN architecture can be made up out of four main components: local receptive fields, shared weights, pooling, and fully connected layers. By stacking several convolutional and pooling layers, followed by one or more fully connected layers at the end, a deep CNN architecture is constructed.

While one-dimensional CNNs can extract spectral features from data and two-dimensional CNNs can extract spatial features from input data, 3D CNNs can simultaneously extract both spectral and spatial features from input volumes. This makes 3D CNNs particularly useful for analyzing volumetric data in medical imaging. The mathematical formulation of 3D CNNs is similar to that of a two-dimensional CNNs, with the addition of an extra dimension. Figure 11 shows the basic architecture of a 3D CNN.

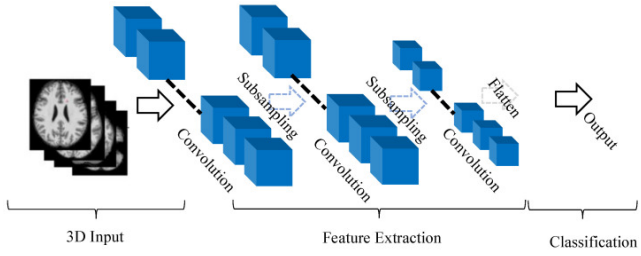

Fig. 11. Typical architecture of 3D CNN.

A. Segmentation

Segmentation is a process that helps us to focus on specific regions in medical images and assists radiologists in quantitative assessment and treatment planning. The authors mention several research works that have contributed to the use of 3D CNNs in medical image segmentation, including DeepMedic, which won the ISLES 2015 competition for brain lesion segmentation (Figure 12), and U-Net, which was proposed for segmentation of 2D biomedical images.

The authors also discuss the challenges associated with lesion segmentation, such as class imbalance and variations

in lesion sizes across different scans. Several approaches to address these challenges are mentioned, including the use of residual connections, data augmentation techniques, and high-end GPUs.

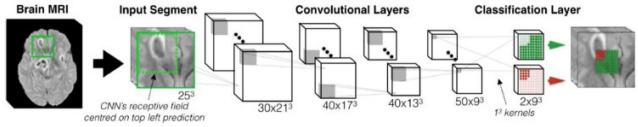

Fig. 12. The baseline architecture of 3D convolution neural network (CNN) for lesion segmentation.

B. Topology and Structure

Several different versions of CNN have been proposed in the literature to improve model performance. In 2011, Krizhevsky et al. presented a deep CNN architecture. AlexNet (shown in Figure 13) has five convolutional layers and three fully connected layers (the last FC layer was the SoftMax layer). The network was trained on 1.2 million images with 60 million parameters.

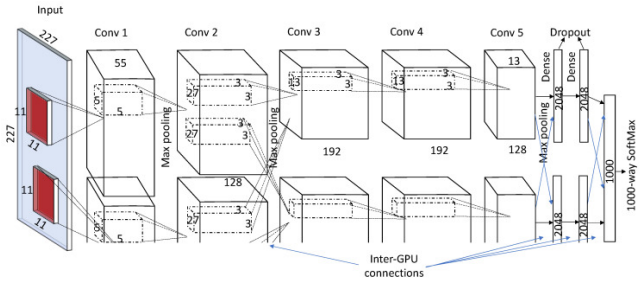

Fig. 13. The baseline architecture of 3D convolution neural network (CNN) for lesion segmentation.

The authors do mention the use of supervised learning, more specifically, they state that deep learning techniques have become an alternative to many machine learning algorithms that were traditionally used in medical imaging. Deep learning is a type of supervised learning where neural networks containing multiple interconnected layers of artificial neurons are used to learn patterns in data samples.

The authors discuss several deep learning architectures, such as 3D CNNs, AlexNet, GoogLeNet, VGGNet, ResNet, U-Net, and Inception-ResNet, which are used for various tasks such as classification, segmentation, detection, and localization in 3D medical imaging. The proposed architecture also intrinsically handles the class imbalance problem that arises due to the use of the Jaccard loss function and the data used in real world being unevenly distributed.

C. Classification

The network has been tested on the 3D IRCADs database and achieved state-of-the-art outcomes, outperforming the other very well-established liver segmentation approaches. They have achieved dice scores of 0.982 and 0.937 for liver and tumor segmentation, respectively. (Figure 14)

Ref.	Methods	Data	Task	Performance Evaluation
Zhou et al. [56]	A 3D variant of FusionNet (One-pass Multi-task Network (OM-Net))	BRATS 2018	brain tumor segmentation	0.916 (WT), 0.827 (TC), 0.807(EC)
Chen et al. [57]	Separable 3D U-Net	BRATS 2018	--do--	0.893(WT), 0.830(TC), 0.742(EC)
Peng et al. [60]	Multi-Scale 3D U-Nets	BRATS 2015	--do--	0.850(WT), 0.720(TC), 0.610(EC)
Kayalıbay et al. [58]	3D U-Nets	BRATS 2015	--do--	0.850 (WT), 0.872(TC), 0.610(EC)
Kamnitsas et al. [54]	11 layers deep 3D CNN	BRATS 2015 and ISLES 2015	--do--	0.898 (WT), 0.750 (TC), 0.720(EC)
Kamnitsas et al. 2016 [53]	3D CNN in which features extracted by 2D CNNs	BRATS 2017	--do--	0.918 (WT), 0.883(TC), 0.854 (EC)
Casamitjana et al. [55]	3D U-Net followed by fully connected 3D CRF	BRATS 2015	--do--	0.917(WT), 0,836(TC), 0.768(EC)
Isensee et al. [59]	3D U-Nets	BRATS 2017	--do--	0.850(WT), 0.740(TC), 0.640(EC)

Fig. 14. 3D CNNs for brain tumor/lesion segmentation on brain tumor segmentation (BRAST) challenges.

IV. Discussion

As we can see in this work, the use of 3D deep learning in medical image analysis has shown great promise in improving the accuracy and efficiency of diagnosis and treatment. However, there are still several challenges and limitations that need to be addressed, so we can fully harness its potential in clinical practice.

One of these challenge (and this one being major) is the lack of large annotated datasets for training and testing 3D deep learning models. Medical image datasets are often small and highly imbalanced, which can lead to over-fitting and poor generalization of deep learning models. Addressing this challenge requires the development of high-quality annotated datasets that are representative of diverse patient populations and diseases.

Another challenge, which makes us untrusty, is the interpretability of deep learning models, as it can be difficult to understand how the models make their predictions. This is particularly important in medical applications where the consequences of incorrect predictions can be severe. Research on explainable AI is needed to develop methods for interpreting and explaining the decisions made by deep learning models, which can improve their trustworthiness and adoption in clinical practice.

Other than challenges, there are also ethical and legal concerns surrounding the use of 3D deep learning when it comes to it being applied in healthcare. Data privacy is a major concern, as medical images contain sensitive information that needs to be protected from unauthorized access or misuse. There are also concerns about algorithm bias, which can lead to disparities in healthcare outcomes for different patient populations. Ensuring regulatory compliance and ethical standards is essential to ensure the safe and responsible use of 3D deep learning in healthcare and some experts discuss that this type of decision making may lack emotions and empathy.

V. Conclusion

Despite all of the challenges, 3D deep learning has already shown significant improvements in various medical applications, including cancer detection, neuro-imaging, and cardiovascular disease diagnosis. As more research is conducted and more data becomes available, the potential for 3D deep learning to revolutionize clinical practice will only increase. It is important that researchers, healthcare providers, and policymakers work together to address the challenges and limitations of 3D deep learning and to ensure its safe and effective integration into healthcare systems.

We still got a way to go, but if we keep up this pace, it shouldn't take long to achieve even greater results in this field. With the grow of faster computation units, we are able to get the results quickly, but we have to work on some kind of optimization as well, since we are running close to the end-point of how much we can push it.

References

[1] Satya P. Singh, Lipo Wang, Sukrit Gupta, Haveesh Goli, Parasuraman Padmanabhan and Balázs Gulyás. 3D Deep Learning on Medical Images: A Review. *arXiv:2004.00218*, 2020. 1, 4

[2] Justus Thies, Michael Zollh¨ofer and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Trans. Graph.*, 38(4), 2019. 1

[3] Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov and Victor Lempitsky. Neural point-based graphics. *arXiv:1906.08240*, 2019. 1, 2

[4] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Ronen Basri and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *arXiv:2003.09852*, 2020. 1, 3

[5] Ruslan Rakhimov, Andrei-Timotei Ardelean, Victor Lempitsky and Evgeny Burnaev. NPBG++: Accelerating Neural Point-Based Graphics. *(CVPR)*, 2022. 2

[6] Hongya Lu, Haifeng Wang, Qianqian Zhang, Sang Won Yoon and Daehan Won. A 3D Convolutional Neural Network for Volumetric Image Semantic Segmentation. *Procedia Manufacturing 39:422-428*, 2019. 4

[7] Michael Niemeyer, Lars Mescheder, Michael Oechsle and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. *arXiv:1912.07372*, 2019.