Deep Interactive Volume Exploration Through Pre-Trained 3D CNN and Active Learning

Marwa Salhi[1] and Riadh Ksantini[2] and Belhassen Zouari[1]
[1] *Mediatron Lab, Higher School of Communications of Tunis, University of Carthage, Tunisia*
[2] *Department of Computer Science, College of IT, University of Bahrain, Bahrain*

Keywords: Volume Visualization, Image Processing, Active Learning, CNN, Deep Features, Supervised Classification.

Abstract: Direct volume rendering (DVR) is a powerful technique for visualizing 3D images. Though, generating high-quality efficient rendering results is still a challenging task because of the complexity of volumetric datasets. This paper introduces a direct volume rendering framework based on 3D CNN and active learning. First, a pre-trained 3D CNN was developed to extract deep features while minimizing the loss of information. Then, the 3D CNN was incorporated into the proposed image-centric system to generate a transfer function for DVR. The method employs active learning by involving incremental classification along with user interaction. The interactive process is simple, and the rendering result is generated in real-time. We conducted extensive experiments on many volumetric datasets achieving qualitative and quantitative results outperforming state-of-the-art approaches.

1 INTRODUCTION

The development of scientific visualization technology has enabled humans to analyze and observe different shapes in 3D scanned data. These evolutions are useful for data analysis, drug development, and disease detection. Based on many machine learning and deep learning techniques, researchers proposed relevant approaches for data exploration.

Generating visual images from volume data using direct volume rendering (DVR) is a common workflow. The key technique for DVR is the transfer function (TF) by assigning colors and opacity to the voxel data. Many data-centric and image-centric transfer function approaches have been proposed. The main difference between the data-centric and image-centric approaches is that data-centric TFs are designed based on volumetric data information. On the contrary, image-centric methods employ rendered images themselves, not volume properties. The main difference between the data-centric and image-centric approaches is that data-centric TFs are designed based on volumetric data information. Despite the good visual performance achieved by many proposed data-centric methods, the interaction process was difficult for the end user. This is owing to the complex widgets like histograms and clusters that users have to manipulate. On the contrary, image-centric methods employ rendered images themselves, not volume properties. As a result, the interaction process became easy for the user. Many effective image-centric methods were proposed but they are not real-time. Recently, a novel image-centric approach was proposed in (Salhi et al., 2022). It allows simple interaction with real-time rendering. However, the method uses handcrafted features for data description. These handcrafted features work well when the shape structure is relatively simple. Nevertheless, as the complexity of the data increases and the target becomes finer, it becomes increasingly challenging to accurately characterize the structure using manually designed features.

On the other hand, Convolutional Neural Networks (CNNs) have become prominent in the solution of many computer vision problems. In fact, CNNs are suitable for image analysis and deep features extraction. However, despite the promising impact of CNNs, they are still too slow for practice systems since they require a sufficient amount of time and large datasets for training. Pre-trained 2D CNNs show promising results with 2D Data, but they are not able to work with 3D efficiently. In addition, it has been shown that 3D CNN is still underdeveloped and still needs more research (Guo et al., 2020). Moreover, because of the lack of 3D publicly available datasets that is large and diverse enough for universal pretraining, using transfer learning methods based on

Salhi, M., Ksantini, R. and Zouari, B.
Deep Interactive Volume Exploration Through Pre-Trained 3D CNN and Active Learning.
DOI: 10.5220/0011638500003417
In *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2023) - Volume 1: GRAPP*, pages 170-178
ISBN: 978-989-758-634-7; ISSN: 2184-4321

2D CNNs can help to reduce computing time and the minimum dataset size that is needed to obtain significant results. For this reason, many researchers tried to use 2D CNNs with 3D data. Doing this requires the transformation of the 3D image. Nevertheless, transforming a 3D object into 2D grids results in losing information which is not optimal for 3D vision tasks.

In this paper, we propose an image-centric framework that generates a visual presentation of volume data in real-time. Specifically, we aim to automatically extract deep features for 3D data using a novel pre-trained 3D CNN while minimizing the loss of information and computational cost. Then, we integrate them into an incremental based interactive system. The proposed framework is simple and intuitive, allowing both user image-based interactions and deep data processing. The user can guide the generation of final results without manipulating complex widgets. After the generation of deep features, a simple GUI containing the most three informative slices is presented to the user to select some critical voxels and assign them to the classes. Then, we use an incremental classifier that learns incrementally the voxels chosen for training and classification of the volume. The different classes are mapped automatically to colors and opacity, and the visual result is rendered to the user in real-time thanks to the incrementality.
Our main contributions are as follows.

- We propose a pre-trained 3D CNN that learns from a pre-trained 2D CNN to minimize the training data and time.
- We use the proposed CNN to extract deep features from the original data to minimize the loss of information.
- We design an image-driven transfer function approach based on deep features generation and active learning by coupling incremental classification with expert knowledge.
- We use incremental classification to achieve real-time rendering.
- We validate the effectiveness and usefulness of the framework with a set of experiments on many 3D data and compare results with other data-centric and image-centric methods.

2 RELATED WORK

The following section briefly summarizes proposed methods for direct volume rendering. We review both data-centric and image-centric approaches.

2.1 Data-Centric Methods

The 1D histogram Transfer function was the first and the most common technique used in various studies (Drebin et al., 1988), (He et al., 1996), (Li et al., 2007), (Fogal and Krüger, 2010), (Childs et al., 2012). However, since using solely the intensity property was not efficient when separating voxels with different features, many researchers proposed 2D or higher dimensional TFs to overcome the limitations of 1D TFs. They included many properties, including gradient magnitude (Kniss et al., 2002), color distance gradient (Morris and Ebert, 2002), curvature (Kindlmann et al., 2003), occlusion spectrum (Correa and Ma, 2009), and view-dependent occlusion (Correa and Ma, 2010). Athawale et al. (Athawale et al., 2020) recently proposed a 2D transfer function framework to explore uncertain data.

Since using higher dimensions of the feature space results in more complications in designing the transfer function, many data-centric methods used clustering techniques to generate visual results. They coupled clustering with other properties, including LH histogram (Sereda et al., 2006), density distribution (Maciejewski et al., 2009), spherical self-organizing map (Khan et al., 2015), and Lattice Boltzmann (Ge et al., 2017) to generate TFs.

More recently, researchers include deep learning techniques to design their TF frameworks. These techniques provide hierarchical architecture and allow better analysis of data. Yang et al. (Yang et al., 2018) combined a nonlinear neural network with an improved LH to design the transfer function. Also, Hong et al. (Hong et al., 2019) incorporated CNNs and GANs for volume exploration. Torayev et al. (Torayev and Schultz, 2020) applied a proposed dual-branch CNN to extract features used for classification.

Data-centric approaches are based on volume properties. Augmentation of the number of used domain properties allows efficient distinction of the structures and achieves better volume rendering. However, It becomes more complex for the end user to interact with the system, which results in the need for much effort to design applicable TFs.

2.2 Image-Centric Methods

Unlike data-centric, image-centric methods are designed on the volume itself, not on its properties. Thus, image-centric TFs are much easier for the user to interact with.

Ropinski et al. (Ropinski et al., 2008) presented a transfer function design where users could draw directly on a monochromatic view to define strokes.

Then, defined strokes are used for the classification and generation of the transfer functions. Guo et al. (Guo et al., 2011) presented a popular technique. It is based on enabling the end user to use a set of intuitive tools to modify visual appearances. Then, the users' changes are used to update the rendering results. A similar approach was proposed recently in (Khan et al., 2018). They enable users to directly interact with a set of selected slices from the 3D image. Then, the rendering result is generated based on some user-selected voxels. Recently, Salhi et al. (Salhi et al., 2022) proposed an extension of khan et al.'s method. They simplified the interaction process and achieved real-time rendering. However, the method does not use deep features, which affects the quality of rendering results.

In addition, many deep learning-based image-centric methods were proposed. Luo et al. (Drebin et al., 1988) introduced a method to improve the utilization of the GPU and visualize 3D data quickly. Han et al. (He et al., 1996) introduced a generative adversarial network-based model for volume completion. Berger et al. (Berger et al., 2019) employed GANs for synthesizing renderings of 3D data.

Recently, some image-centric methods based on CNN were proposed. A CNN allows the effective description of complex data thanks to its deep features. Shi and Tao (Shi and Tao, 2019) used a CNN to solve the viewpoint selection problem for volume visualization. They trained the CNN using a generated dataset containing a large number of viewpoint-annotated images. Kim et al. (Kim et al., 2021) proposed a TF design based on CNN to automatically generate a DVR result. They used Internet images to construct labeled datasets that are used next to train a CNN model. Then, they extracted label colors using the trained CNN to construct the final labeled TF.

Contrary to the above-mentioned studies, the proposed method is simple, fast, effective, and doesn't require a lot of data for training. It enables user involvement in designing the system without manipulating complex widgets. Besides, the method combines deep and Active learning for generating efficient and real-time image-centric TF.

3 PROPOSED CNN-based IMAGE-CENTRIC METHOD

Designing an effective TF depends on three essential elements, speed, ease of use, and good quality of the rendering result. For this reason, our proposed approach is an image-centric method where the user can interact directly with 2D images that he is familiar with. It makes use of a 3D CNN designed to minimize the loss of information while extracting deep features and incremental classification to generate real-time rendering.

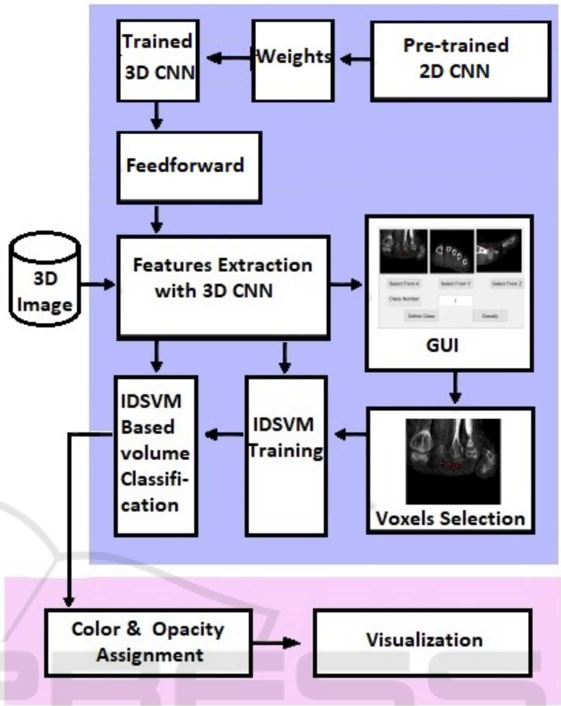

Figure 1: The CNN-based image-centric TF block diagram.

3.1 User Interaction for Voxels Selection

The performance of most of the existing studies is affected by the human-computer interaction side of the system. Researchers enable the domain expert to interact with the system not just for usage but also to control the final result. However, manipulating complex techniques can hinder their workflow. In our framework, we aim to gather knowledge from the domain expert through easy-to-use controls because he is the best one that knows what to look for. For this reason, the proposed system enables users to perform selection operations on volume slices. To ensure that the GUI is simple enough and not crowded with useless images, we pick the most three representative slices along the X, Y, and Z directions. For this reason, we used image entropy to calculate the quantity of information in each slice. The slices with the highest entropy values are the ones containing the most variations. So, we select the best one from each direction as the most informative slice. Through a simple GUI view, as presented in Figure 2, the user can interact with the slices to select regions that he is interested in. He chooses the number of classes he wants to see, selects some voxels with the mouse, and affects them to the corresponding groups.

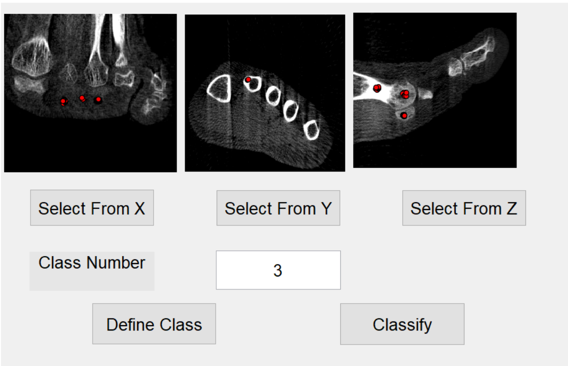

Figure 2: The GUI with extracted slices and selected voxels.

3.2 Classes Generation from Labeled Voxels

After the user decides what he wants to see, we used the labeled voxels as training samples to classify the volume voxels accordingly. The first step is to characterize each voxel with a set of deep features using the 3D CNN detailed above. Then, we used the incremental classifier IDSVM (Incremental Discriminant based Support Vector Machine) to classify the Whole volumetric data. Both steps are detailed next.

3.2.1 Pretarained 3D CNN for Deep Features Extraction

Designing the 3D CNN is a foundation step for the proposed image-centric framework. In this section, we detail the architecture of the novel pre-trained 3D CNN. The proposed CNN is motivated by the benefit of the amazing success of the 2D CNN to build 3D pre-trained CNN without sacrificing data information. As the 3D image is composed of a set of 2D slices, weights obtained from training on 2D content can be meaningful to describe volume voxels. So, the idea is to transfer learning from a pre-trained 2D CNN to a novel 3D CNN. Figure 3 shows the 3D CNN architecture. We designed a new architecture of the 3D CNN inspired by the architecture of Resnet50 trained with ImageNet. First, we create the input layer without a restriction on the input size, as in the case of Resnet50. As we said before, we want to minimize the loss of information. Therefore, the input layer receives the 3D image without doing a resize. After that, we limit the number of 3D convolutions to eight layers.

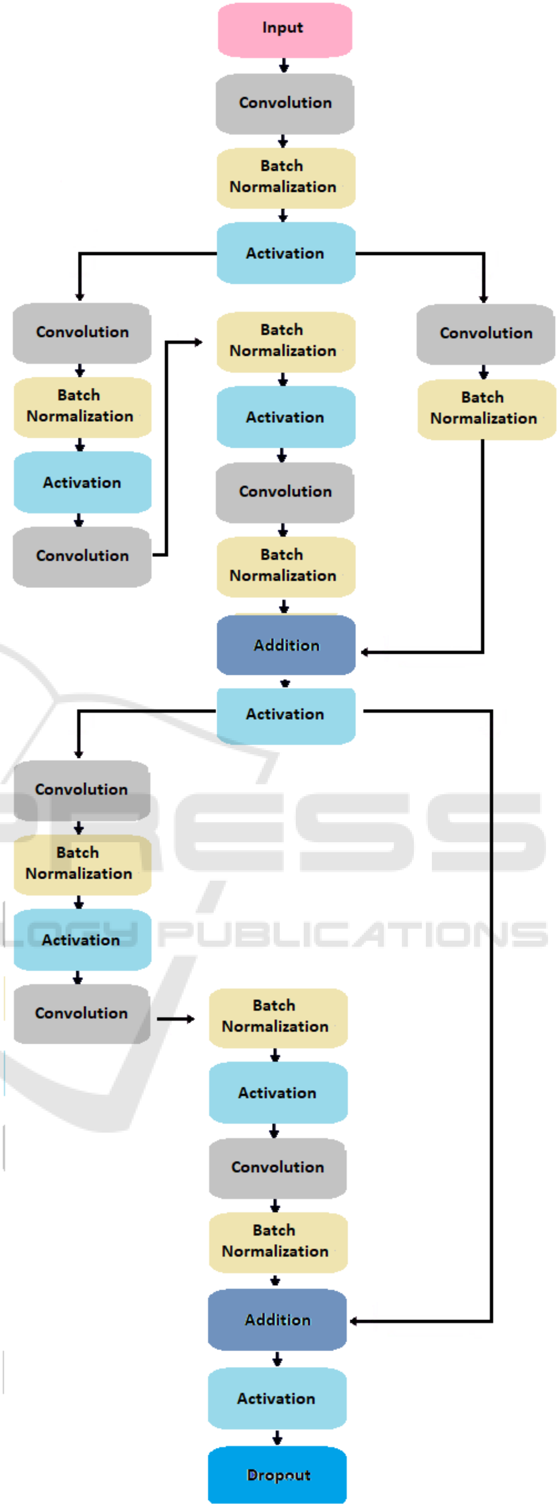

Figure 3: The 3D CNN architecture.

Those layers are initialized with the weights of the Resnet50. 2D weights can be defined as 2D matrices. Therefore, we need to convert a 2D matrix into a 3D tensor by filling the tensor and copying the values along one axis. The convolution receptive field is $7*7*7$. Each 3D convolutional layer is proceeded by a batch normalization layer and then the activation function. Since we want to use the 3D CNN as a feature extractor for volumetric data, the model has to be capable of extracting a set of features for each voxel. So, the data have to be passed through all the convolutions layers without resizing. For this reason, we didn't use pooling layers in the proposed architecture. For the same reason, we fixed the stride to 1, and the padding to SAME. As discussed before, methods mostly use hand-crafted features. However, the problem with those features is that they cannot extract voxel information very properly. Since present-day volumetric data have some inherent noise associated with them, losing information might prove to be costly and may not properly classify the volume data for volumetric rendering. Nevertheless, the proposed model can generate a set of deep features for each voxel while minimizing the loss of information due to the use of the proposed 3D CNN. As a result, in the proposed model, one significant advantage of using a 3D CNN is that generated features are not statics but adapted to different used datasets.

3.2.2 Incremental Classifcation

After extracting deep features from the 3D image, the IDSVM classifier is trained using the extracted deep features of the user-selected voxels. The IDSVM was proposed and described in detail in (Salhi et al., 2022). It is motivated by building a separating hyperplane based on the merits of both discriminant-based classifiers and the classical SVM. The IDSVM calculates first the within-class scatter matrix and the between-class scatter matrix in the input space to extract the global properties of the training distribution. Then, it integrates the resulting matrix into the optimization problem of the SVM to consider the local characteristics. The fundamental advantage of our image-centric method is the combination between deep learning and active learning to achieve efficiency and speed. Active learning is defined by a combination of user interaction with incremental classification. The utility of incremental classification in the proposed method derives from the improvement in computational complexity, both in terms of needed memory and time. This is because the IDSVM doesn't require retraining each time a new voxel is selected. Also, calculating the scatter matrices in the input space results in minimizing required memory and time. The classifier is capable of separating the user-labeled training samples with a hyperplane placed in an optimum way. Therefore, the whole 3D data is classified efficiently, and the rendering result is shown to the user in real time.

3.3 TF Generation Using Harmonic Colors

Assigning optical properties to the voxel data in each structure impacts strongly the final result. For this reason, in the proposed method, the user does not need to handle color and opacity, but values are automatically generated using color harmonization, which is a concept borrowed from the arts (Itten, 1961). Consequently, the visual result can be rendered more quickly and efficiently. In this study, the opacity is calculated based on the user's perception. To define the appropriate color for voxel data, we need to define the hue (H), saturation (S), and lightness (V) values for the HSV color space.

- The hue value (H) for each voxel of each class is calculated as follows.

$$H_g^v = \xi_l^g + (\xi_h^g - \xi_l^g) \times F_v,\ \ g = 1...N. \quad (1)$$

  Where, ξ_l^g and ξ_h^g are the limits of the hue range of each classe g, N is the number of groups, and F_v is the Gradient Factor of the voxel.

- The Saturation (S) is calculated for a group g according to the spatial variance σ_g. Since classes having larger spatial variances cover a larger viewing area compared to classes with smaller spatial variances, we can assume that the classes with smaller spatial variances need to have a more saturated color to highlight them properly. Hence, the saturation value for each class is calculated by:

$$S_g = \frac{1}{1+\sigma_g^2}, \quad g = 1,\ldots,N. \quad (2)$$

- Similarly, we can intuitively find that voxel classes closer to the center of the volume requires to be brighter than voxel classes that are farther from the center so that they are not overshadowed by them. Hence, The lightness (V) of each class is defined by:

$$V_g = \frac{1}{1+D_g}, \quad g = 1,\ldots,N. \quad (3)$$

  Here, D_g denotes the distance between the centroid of the group g and the center of the 3D image. Using this method to assign the saturation and brightness values, we can increase visualization of voxel classes that are relatively more difficult to highlight.

- The opacity of a voxel v belongs to a class g is calculated as:

$$O = (1 - \frac{\sigma_g^2}{\max \sigma_g^2}) \times (1 + F_v), \quad g = 1, \ldots, N. \quad (4)$$

As we can see, the opacity value is calculated based on the spatial variance. To avoid blocking the viewing of classes with smaller spatial variances by those with larger spatial variances, groups with smaller spatial variances will be assigned higher opacity values. In Addition, using the multiplication $(1 + F_v)$, the method allows boundaries enhancement to reveal the embedded structures.

4 RESULTS AND DISCUSSION

In this section, we first provide the details of the datasets used to validate our approach and then present a quantitative and qualitative comparison between the results of the proposed method and other state-of-the-art techniques, including image-centric (Khan et al., 2018), (Salhi et al., 2022) and data-centric methods (Khan et al., 2015). To show the usefulness of our method, we used five datasets namely, CT scans of an Engine, VisMale, Cross, Hydrogen, and Foot to present the rendering results. Details of the datasets are listed in Table 1.

Table 1: Details of used datasets and training times for each system in seconds.

Dataset	Dimensions	Data-centric	Image-centric	Ours
Hydrogene	128*128*128	119	4.667	0.035
VisMale	128*256*256	284	8.861	0.673
Cross	64*64*64	78	1.082	0.021
Foot	256*256*256	388	24.720	0.790
Engine	256*256*256	371	11.887	0.190

After training the CNN and extracting deep features, we present a simple GUI to the user. The user can select some voxels as regions of interest. The voxels of the selected regions are used to train the IDSVM and classify the volume. The colored groups according to the TF are then rendered in real time.

As noted before, due to the use of deep features that well describe voxels, the system doesn't need a high number of voxels for training. Some selected voxels from the user are good enough for an efficient classification and visualization of the different groups. Such a low number of training voxels is reasonable here since, as we can see from the slices shown in figure 4, the selection process is not easy with small datasets like the Cross dataset, especially when small classes are not visually distinguishable in slices. Besides, using a low number of training voxels results in reducing the training time. Table 1 illustrates the results of the comparison of the training time required by the proposed system with a batch-based image-centric (Khan et al., 2018) and a data-centric method (Khan et al., 2015). As we said, the proposed method doesn't need a lot of training samples to achieve good visualization thanks to the use of deep features. Besides, it makes use of incremental classification. As a result, it can be seen from the Table that it doesn't need a lot of time to complete the training. On the other hand, the data-centric method takes a lot of time since the training process requires a lot of training voxels (in the range of millions) (Khan et al., 2015). In addition, even if the other method is image-centric, using batch classification results in the augmentation of required training times.

Figure 4 illustrates obtained rendering results for different datasets from the comparative experiments. As can be seen, the volumetric data are visualized using the proposed deep-based method, and useful structural information can be gathered from these results. As we can see, even for small datasets, the critical structures can be highlighted easily. Also, we can see that the inner bones are visible for the foot dataset. Besides, for the engine, the tubes are clearly detected. This definitely depicts the effectiveness of our method. Compared to the other results, the power of deep features is apparent. Since the classification is based on user interaction and the user can freely select whatever voxels he is interested in, the technique is very flexible. Also, As a result of the use of incrementality, it is easy to yield real-time rendering each time the user selects new voxels or adds a new class.

To thoroughly evaluate the performance of the proposed framework, besides visual results, we conducted a quantitative evaluation using the Dice over-

Table 2: The obtained dice similarity results for each approach (best results are in bold, and second-best are emphasized).

Dataset	Batch Image-centric	Incremental Image-centric	Data-centric	Ours
Hydrogene	0.5691	0.7491	*0.7569*	**0.8452**
VisMale	0.7088	*0.7477*	0.7101	**0.8087**
Cross	0.6733	0.8041	*0.8924*	**0.9320**
Foot	0.8045	*0.84877*	0.8153	**0.8603**
Engine	0.6584	0.8256	*0.9195*	**0.9295**

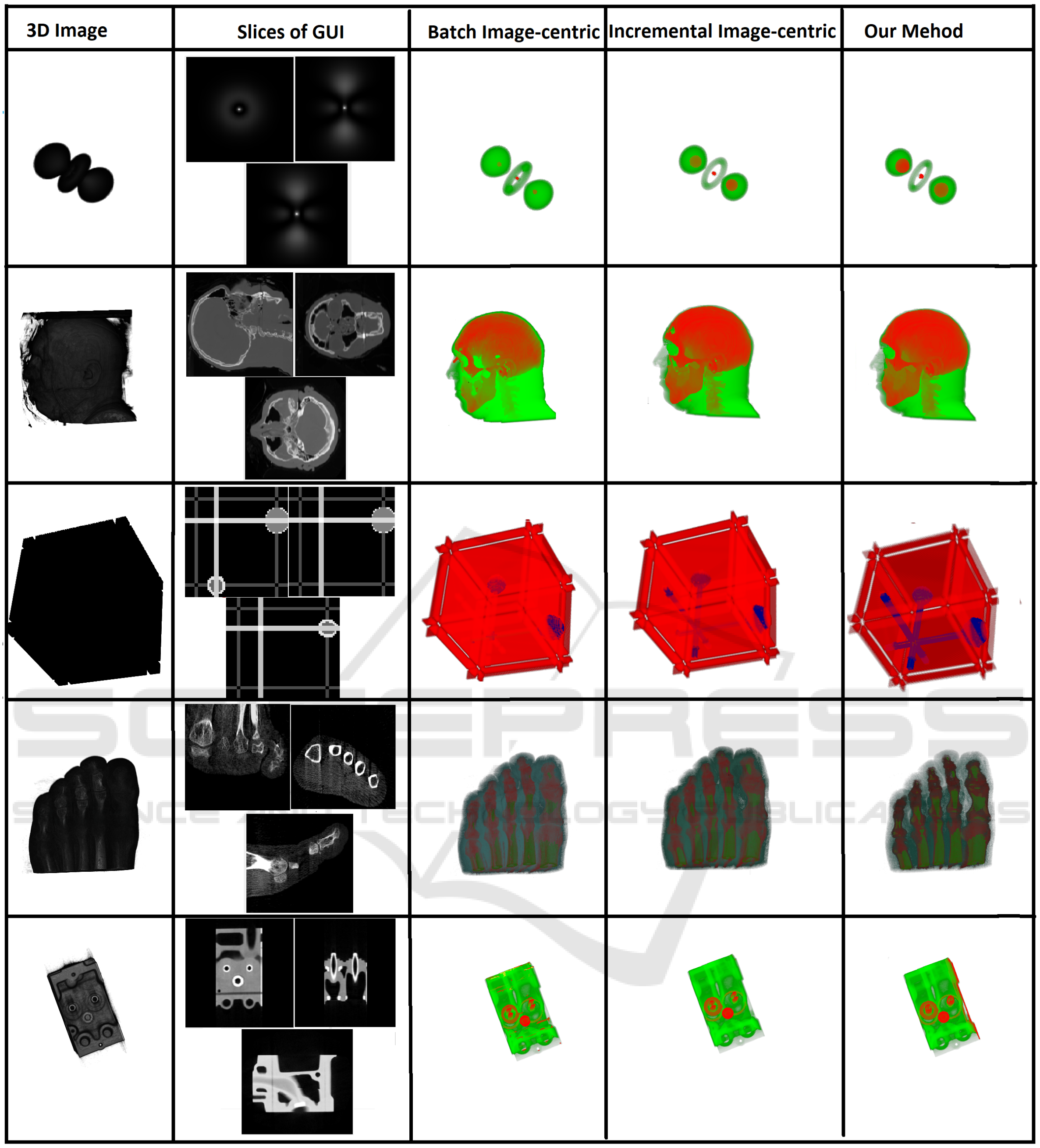

Figure 4: Comparison of rendering images with batch-based (Khan et al., 2018) and incremental-based (Salhi et al., 2022) image-centric approaches using the same user-selected voxels from Slices of GUI.

lap coefficient. The dice coefficient metric compares the obtained rendering results with the ground truth to compute the accuracy using the following equation. As we can see from Table 2, results indicate that the proposed technique achieves the best dice similarity results compared to the other methods for all used datasets. In addition, the proposed method achieves an average dice similarity coefficient equal to 0.8751. This value proves the efficiency of the proposed approach and makes it suitable for use in applications that requires high precision, like the medical domain.

5 CONCLUSIONS

In this paper, we tackle the problem of hand-crafted feature learning limitations in the context of volume rendering. We introduced a pre-trained 3D CNN deep learning framework, consisting of a new architecture inspired by Resnet50, that learns weights from the 2D Resnet50 for initialization. The novel CNN allows for gathering deeper information from the data voxels. We incorporate the 3D CNN into an incremental-based image-centric method to improve the feature learning process and classification efficiency. The performance of the image-centric was evaluated on many popular 3D datasets. Results were compared against other methods. We demonstrated that the new framework performs with the highest accuracy on all three datasets. The empirical results confirmed that the 3D CNN improves the performance of visualization-based classification. This framework allows users to interact with an intuitive user interface and control the final rendering results. Finally, we compared the required training time for the proposed system with other methods and showed that the proposed CNN-based framework outperforms the other methods in all experimented datasets. As a result, the proposed method achieves real-time rendering of visual results while the users select regions of interest, thanks to the use of incremental classification. In the future, we aim to improve the network performance. The proposed method has a few limitations that will be worked on. In this approach, we worked with the best slices along X,Y, and Z directions to create a simple GUI without being crowded with unnecessary images. The choice of the slices is based only on entropy. To ensure that we pick the best representative ones, other criteria could also be used like information gain or mutual information. Besides, the final result depends strongly on the user's choice. Even if the system needs to be used by a domain expert who is familiar with grayscale images and knows exactly what he looks for, a bad choice does not provide satisfactory results. So rather than restarting the system, the elimination of some selected voxels needs to be taken into consideration. As a result, using a decremental classification could be useful in some cases to perform satisfactory results. Also, we can investigate the effectiveness of the 3D CNN-based method to explore volumetric data in real-time clinical applications.

REFERENCES

Athawale, T. M., Ma, B., Sakhaee, E., Johnson, C. R., and Entezari, A. (2020). Direct volume rendering with nonparametric models of uncertainty. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1797–1807.

Berger, M., Li, J., and Levine, J. A. (2019). A generative model for volume rendering. *IEEE transactions on visualization and computer graphics*, 25(4):1636–1650.

Childs, H., Brugger, E., Whitlock, B., Meredith, J., Ahern, S., Pugmire, D., Biagas, K., Miller, M., Harrison, C., Weber, G. H., et al. (2012). Visit: An end-user tool for visualizing and analyzing very large data.

Correa, C. and Ma, K.-L. (2009). The occlusion spectrum for volume classification and visualization. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1465–1472.

Correa, C. D. and Ma, K.-L. (2010). Visibility histograms and visibility-driven transfer functions. *IEEE Transactions on Visualization and Computer Graphics*, 17(2):192–204.

Drebin, R. A., Carpenter, L., and Hanrahan, P. (1988). Volume rendering. *ACM Siggraph Computer Graphics*, 22(4):65–74.

Fogal, T. and Krüger, J. H. (2010). Tuvok, an architecture for large scale volume rendering. In *VMV*, volume 10, pages 139–146.

Ge, F., Noël, R., Navarro, L., and Courbebaisse, G. (2017). Volume rendering and lattice-boltzmann method. In *GRETSI*.

Guo, H., Mao, N., and Yuan, X. (2011). Wysiwyg (what you see is what you get) volume visualization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2106–2114.

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., and Bennamoun, M. (2020). Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(12):4338–4364.

He, T., Hong, L., Kaufman, A., and Pfister, H. (1996). Generation of transfer functions with stochastic search techniques. In *Proceedings of Seventh Annual IEEE Visualization'96*, pages 227–234. IEEE.

Hong, F., Liu, C., and Yuan, X. (2019). Dnn-volvis: Interactive volume visualization supported by deep neural network. In *2019 IEEE Pacific Visualization Symposium (PacificVis)*, pages 282–291. IEEE.

Itten, J. (1961). The art of color the subjective experience and objective rationale of color.

Khan, N. M., Ksantini, R., and Guan, L. (2018). A novel image-centric approach toward direct volume rendering. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 9(4):1–18.

Khan, N. M., Kyan, M., and Guan, L. (2015). Intuitive volume exploration through spherical self-organizing map and color harmonization. *Neurocomputing*, 147:160–173.

Kim, S., Jang, Y., and Kim, S.-E. (2021). Image-based tf colorization with cnn for direct volume rendering. *IEEE Access*, 9:124281–124294.

Kindlmann, G., Whitaker, R., Tasdizen, T., and Moller, T. (2003). Curvature-based transfer functions for direct volume rendering: Methods and applications. In *IEEE Visualization, 2003. VIS 2003.*, pages 513–520. IEEE.

Kniss, J., Kindlmann, G., and Hansen, C. (2002). Multi-dimensional transfer functions for interactive volume rendering. *IEEE Transactions on visualization and computer graphics*, 8(3):270–285.

Li, J., Zhou, L., Yu, H., Liang, H., and Wang, L. (2007). Classification for volume rendering of industrial ct based on moment of histogram. In *2007 2nd IEEE Conference on Industrial Electronics and Applications*, pages 913–918. IEEE.

Maciejewski, R., Woo, I., Chen, W., and Ebert, D. (2009). Structuring feature space: A non-parametric method for volumetric transfer function generation. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1473–1480.

Morris, C. J. and Ebert, D. (2002). Direct volume rendering of photographic volumes using multi-dimensional color-based transfer functions. In *Proceedings of the symposium on Data Visualisation 2002*, pages 115–ff.

Ropinski, T., Praßni, J.-S., Steinicke, F., and Hinrichs, K. H. (2008). Stroke-based transfer function design. In *Volume Graphics*, pages 41–48. Citeseer.

Salhi, M., Ksantini, R., and Zouari, B. (2022). A real-time image-centric transfer function design based on incremental classification. *Journal of Real-Time Image Processing*, 19(1):185–203.

Sereda, P., Bartroli, A. V., Serlie, I. W., and Gerritsen, F. A. (2006). Visualization of boundaries in volumetric data sets using lh histograms. *IEEE Transactions on Visualization and Computer Graphics*, 12(2):208–218.

Shi, N. and Tao, Y. (2019). Cnns based viewpoint estimation for volume visualization. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(3):1–22.

Torayev, A. and Schultz, T. (2020). Interactive classification of multi-shell diffusion mri with features from a dual-branch cnn autoencoder.

Yang, F., Meng, X., Lang, J., Lu, W., and Liu, L. (2018). Region space guided transfer function design for nonlinear neural network augmented image visualization. *Advances in Multimedia*, 2018.