3D Binary Classification of Ink Detection using Deep Learning

GWID: G41223192 Name: Neng Zhou

GitHub repo: https://github.com/Vizards8/CSCI_6364_ML

1. Abstract:

The 3D binary classification is a challenging task which needs to classify objects in three-dimensional space into two categories. It has a wide range of real-world applications, such as medical imaging, autonomous navigation, and object recognition. In this project, I will apply 3D binary classification to ink detection to determine whether it contains ink or not. I trained two models, one is 3D CNN, the other is 3D ResNet. The models are trained and tested on the open-source dataset with ground-truth labels and no pre-training. Through the comparison, 3D ResNet achieved the best performance. I suppose deeper model with residual mechanism can extract more information in this 3D binary classification task.

2. Introduction:

Early in 2000s, processing of 3D objects for visual intelligence has been based on hand-crafted features [1][2][3]. The handcrafted features require significant domain knowledge and manual feature engineering, which can be time-consuming may limit the scalability of the approach. However, in recent years, scientists and researchers have turned to deep learning-based methods for developing 3D classification models. One of the early works on 3D CNNs for binary classification was the VoxNet architecture proposed by Maturana et al. [4] in 2015. VoxNet used a 3D CNN with max pooling and fully connected layers to classify 3D objects from voxelized data. Another popular 3D binary classification model is the 3D U-Net architecture proposed by Çiçek et al. [5] in 2016. The 3D U-Net used an encoder-decoder architecture with skip connections and 3D convolutions to extract features from 3D medical images for brain tumor segmentation in MRI scans. V-Net is another 3D CNN proposed by Milletari et al. [6] in 2016. The V-Net architecture used a 3D CNN with residual connections to extract features from 3D medical images for segmentation tasks. PointNet, proposed by Qi et al. [7] in 2017, is a popular deep learning architecture for point cloud classification. PointNet uses a shared MLP to extract features from individual points in a point cloud, and a symmetric function to aggregate the features of all the points in the cloud. PointNet showed promising results on a variety of 3D shape classification and segmentation tasks. PointNet++ [8] is an extension of PointNet proposed by Qi et al. in 2017. PointNet++ uses a hierarchical neural network architecture to capture local and global features of point clouds for improved point cloud classification and segmentation. Kaul et al. [9] proposed FatNet in 2021 that introduces a novel attention-infused layer, called the FAT layer, combining both global and local features. It also uses weightings over two different feature aggregation methods, residual connections, and shared-weight MLPs to enhance network performance. Over the past few years, the research in 3D binary classification models in deep learning has seen significant achievements in recent years. The deep learning models are becoming more and more accurate and efficient. As research continues, it is expected that more novel architectures and techniques will be developed to further advance the field and enable new applications.

3. Dataset and Resources:

I will use the Vesuvius Challenge Ink Detection dataset. There are thousands of scrolls that were once part of a library located in a Roman villa in Herculaneum, a town adjacent to Pompeii. It was buried by the Vesuvius eruption almost 2000 years ago. Unfortunately, due to the high temperature of the volcano, the scrolls were carbonized, making them impossible to open without damaging them.

The dataset contains 3D x-ray scans of scrolls with and without ink. The scans are provided in .tif format and each fragment contains 65 slices with corresponding ground truth labels. The dataset also has a binary mask to show which pixels contain data. The size of the dataset is 37.02 GB in total. Therefore, it provides sufficient data for me to do 3D binary classification of Ink Detection. In the experiment, I will split the dataset into training, validation sets. I will use my laptop with a 3060 GPU to train my model on this dataset.

4. Proposed work:

The main objective of this project is to classify 3D x-ray scans of scrolls into whether they contain ink or not. In the rest of the paper, I will talk about the proposed work one by one. Here is the proposed work:

1. Preprocess the 3D x-ray data (.tif) by down sampling and normalizing.
2. Train several deep learning models (e.g., 3D CNN, 3D ResNet) on the training set and optimize the hyperparameters to achieve high accuracy on the validation set.
3. Evaluate the performance of the trained model on the validation set using evaluating metrics such as accuracy, precision, recall and F1-score.
4. Compare the performance of different deep learning models and try to identify the best model for this problem.

5. Preprocess:

To preprocess the fragments, I loaded all the surface volumes using Pillow library and its corresponding binary mask. Then, I stacked them together in the dimension of Z. To save the memory, I observed that the middle slices contain most of the information. Therefore, I only kept 48 slices in the middle and dropped the remaining slices. As the original fragment has a high resolution of 6330 * 8181, I divided it into 64 * 64. To avoid noise, I also removed edge voxels which are not fully contained within the mask. They may contain incomplete or partial information about the fragments. Finally, as the input data is stored as 16-bit unsigned integers, I divided them by 65535 to normalize the voxel values.

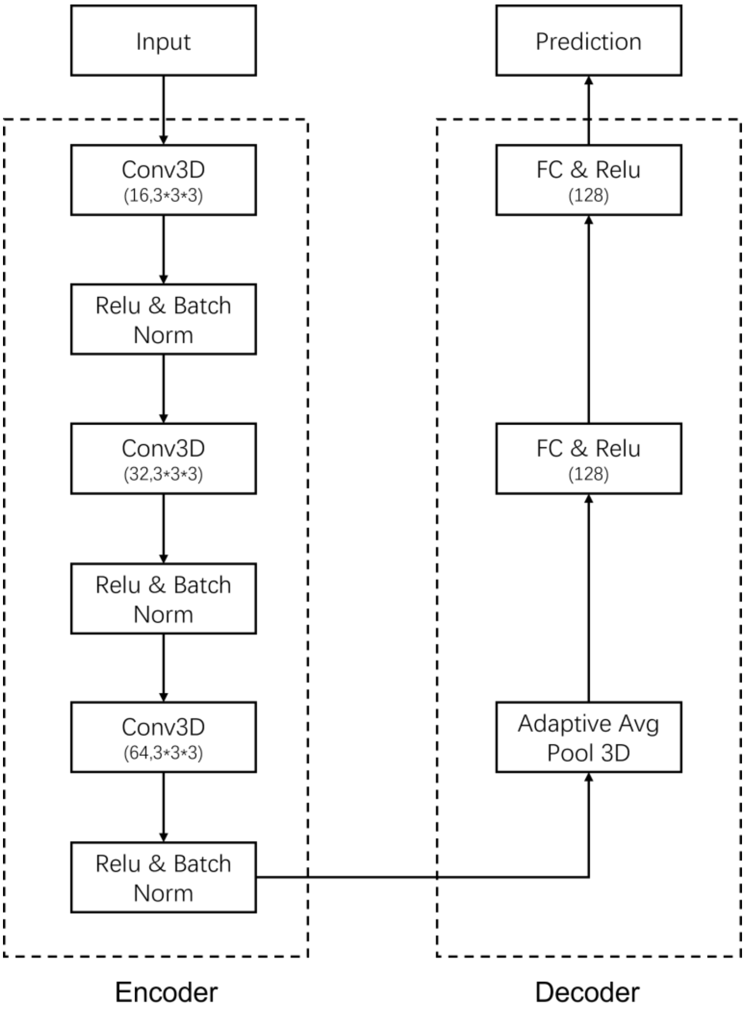

Figure 1 3D CNN

6. Models:

6.1 3D CNN:

The first model I used is a simple CNN model. To adapt the model to 3D data, I replaced several 2D convolutional layers with 3D convolutional layers. This model has three convolutional layers, each followed by a Relu activation function and batch normalization. The output of the last convolutional layer is sent to an adaptive average pooling layer. It then be flattened and fed into two fully connected layers. Each layer has a Relu activation function. Finally, the model outputs the prediction. The architecture of the model is shown in Figure 1.

6.2 3D ResNet:

3D ResNet (R3D) proposed by Tran et al. [10] is a popular 3D CNN architecture that extends the concept of 2D convolutional layers to 3D. It was originally proposed for video classification and action recognition which extracted from video clips. The original model only has R3D-18 and R3D-34 corresponding to ResNet18 and ResNet34. In my experiment, I extended R3D to R3D-50 which is similar to ResNet50 proposed by K. He et al. [11]. Every layer consists of bottleneck blocks shown in Figure 2. Each block is a stack of three layers, including 1×1×1, 3×3×3, 1×1×1 3D convolution layers, and has a shortcut directly links the input and output. The shortcut permits linear transformation for the information of every trained neural network, which means the performance will not degrade as the number of layers increases. The architecture of the model is shown in Table 1.

layer name	R3D-50
conv1	3×7×7, 64, stride 1×2×2
conv2_x	$\begin{bmatrix}1\times1\times1,64\\3\times3\times3,64\\1\times1\times1,256\end{bmatrix}\times 3$
conv3_x	$\begin{bmatrix}1\times1\times1,128\\3\times3\times3,128\\1\times1\times1,512\end{bmatrix}\times 4$
conv4_x	$\begin{bmatrix}1\times1\times1,256\\3\times3\times3,256\\1\times1\times1,1024\end{bmatrix}\times 6$
conv5_x	$\begin{bmatrix}1\times1\times1,512\\3\times3\times3,512\\1\times1\times1,2048\end{bmatrix}\times 3$
	average pool, fc

Table 1 R3D-50

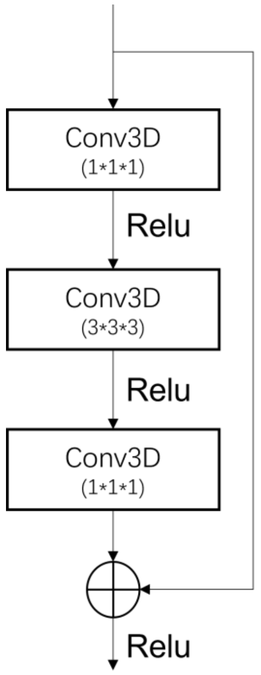

Figure 2 3D bottleneck

7. Implementation Details:

Both input channel and output channel are set to 1. To better compare the performance of the two models, I didn’t use any pretrained models. During the training, all networks used SGD optimizer and initial learning rate was set to 1e-3. Due to the large size of the dataset, I only used 3200 voxels in the validation set while the remaining voxels were used in the training set. I used an Nvidia RTX3060 Laptop GPU to train the models.

8. Results:

I trained the model for about 60k steps. I recorded the loss, accuracy and other evaluation metrics every 500 steps. The training process is really time consuming, so I ended the training when it is basically converged. Figure 3 shows the loss curves for training and validation set of the two models. Both of them are basically converged at 60kstep.

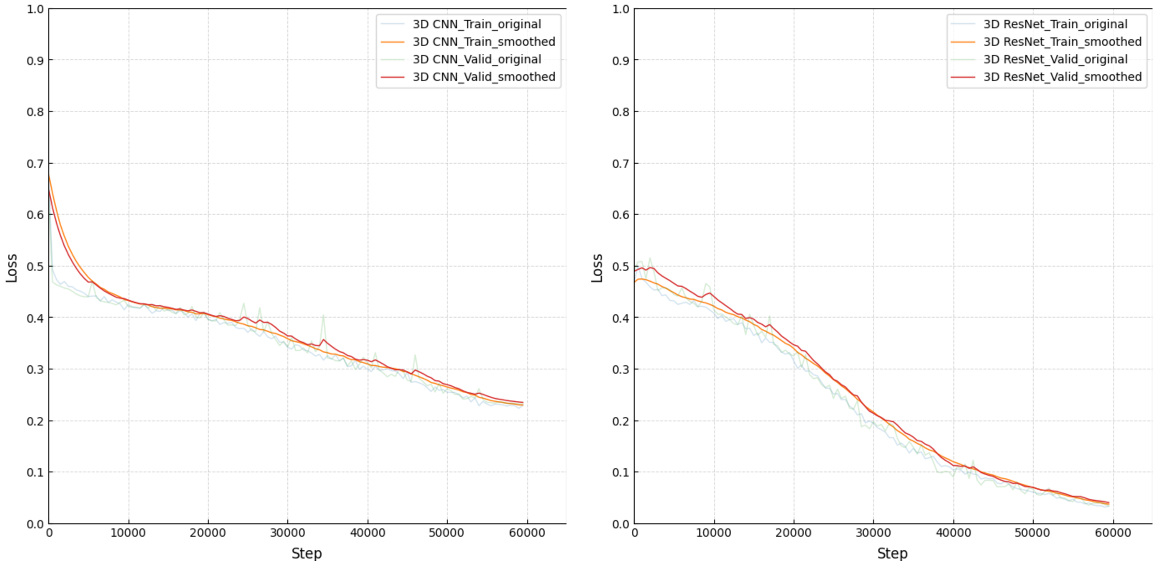

Figure 3 Loss Curves

Figure 4 indicates the overall performance on validation dataset between two models. The results clearly show that the 3D ResNet model outperforms the 3D CNN in all metrics. Specifically, the 3D ResNet model achieves an accuracy of 98.37%, an F1 score of 0.9555, a precision of 0.952, and a recall of 0.9672.

9. Discussion:

ResNet is one of the most well-known models that has been pretrained on the ImageNet dataset. This experiment shows that it can also preforms well on 3D binary classification. This is likely because 3D ResNet is a deeper model which can learn more information about the features from the input data. In addition, I also implemented two more models VNet and VoxelCNN (in "models" directory). However, these models need more GPU memory which can’t run on my laptop. In conclusion, 3D ResNet model performs better than the 3D CNN model on this ink detection dataset.

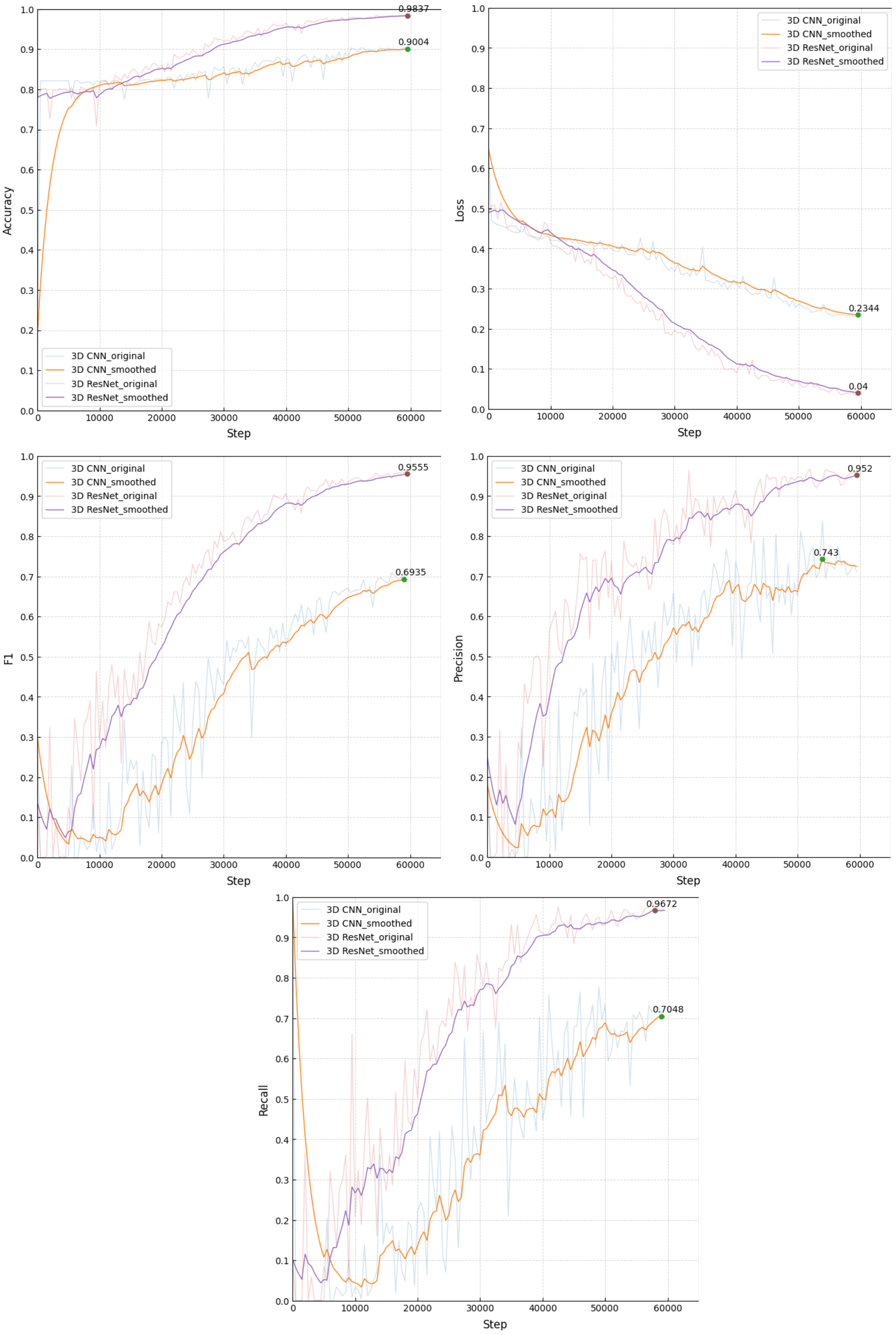

Figure 4 Visual comparison of the evaluation metrics

10. Reference:

[1]. Chen, H., & Bhanu, B. (2007). 3D free-form object recognition in range images

using local surface patches. Pattern Recognition Letters, 28(10), 1252-1262.

[2]. Zhong, Y. (2009, September). Intrinsic shape signatures: A shape descriptor for 3D object recognition. In 2009 IEEE 12th international conference on computer vision workshops, ICCV workshops (pp. 689-696). IEEE.

[3]. Rusu, R. B., Blodow, N., Marton, Z. C., & Beetz, M. (2008, September). Aligning point cloud views using persistent feature histograms. In 2008 IEEE/RSJ international conference on intelligent robots and systems (pp. 3384-3391). IEEE.

[4]. Maturana, Daniel, and Sebastian Scherer. "Voxnet: A 3d convolutional neural network for real-time object recognition." 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, 2015.

[5]. Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., & Ronneberger, O. (2016). 3D U-Net: learning dense volumetric segmentation from sparse annotation. In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19 (pp. 424-432). Springer International Publishing.

[6]. Milletari, Fausto, Nassir Navab, and Seyed-Ahmad Ahmadi. "V-net: Fully convolutional neural networks for volumetric medical image segmentation." 2016 fourth international conference on 3D vision (3DV). Ieee, 2016.

[7]. Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 652-660).

[8]. Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in neural information processing systems, 30.

[9]. Kaul, Chaitanya, Nick Pears, and Suresh Manandhar. "FatNet: A feature-attentive network for 3D point cloud processing." 2020 25th International conference on pattern recognition (ICPR). IEEE, 2021.

[10]. Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., & Paluri, M. (2018). A closer look at spatiotemporal convolutions for action recognition. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 6450-6459).

[11]. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).