3D convolutional logistic regression for 3D plant root reconstruction

Lab Vision Systems: Learning Computer Vision on GPU's

Babayev, Rufat

Universität Bonn
s6rubaba@uni-bonn.de, Matrikelnummer: 2901528

Abstract. In this paper, we studied the root and soil classification performance on 3D Magnetic Resonance Imaging (MRI) data of soil grown plants. Our goal in this topic is to use Convolutional logistic regression. Real MRI images both exhibits local water content of the investigated plant and also water samples in the soil. To denoise water samples of the soil in the image, MRI scans are refined using specific reconstruction algorithms. In our case, we performed training on the reconstructed image and then tested classification accuracy on the noisy image. With this in mind, our classification strategy denoises MRI scan and obtains the approximation for the reconstructed image at the end. The empirical evaluation results show the strength and optimality of our classification strategy on different noise added MRI scans.

1 Introduction

We present a method for root and soil classification from noise added virtually generated MRI scans. When invasive root imaging is not used, scanning soil for plant root detection is hindered by different restrictions. For 3D Magnetic Resonance Imaging (MRI), signals are emitted from water in the roots of a plant and in a low degree from water samples in the soil. In turn, along with the scan of real root structure, noisy examples from the soil are obtained which opens a pathway for refinement and reconstruction algorithms for MRI scans.

Scanning equipment ensures MRI measurements with different spatial resolutions. In higher resolutions, while wider view of plant's root structure is obtained, noisy samples from soil water exhibits themselves more evidently. In this case, ratio of the noise samples to real root samples increase drastically which is creating imbalanced learning problem for our classifier. To handle imbalance between root samples and soil samples we used specific sampling technique (random undersampling) to balance the distribution of the data for our classification strategy. Then, we utilized 3D convolutions of the obtained data and logsitic layer to build a 3D convoltional classifier. At the end, we chose noise added MRI scan of anohter plant to test the robustness and reconstruction capabilities of our method.

The primary concern with the classification on imbalanced data is that the distribution of data on different classes varies significantly which jeopardizes the predictive performance of standard classification algorithms. In our dataset, all resolutions of the MRI scans exhibit high imbalance between root samples and soil samples. It is obvious that, soil samples are far more than the root samples. We denote plant root pixels as minority class and soil pixels as majority class in a chosen MRI scan. When logistic regression is applied to the scan, the weights or weight set of the hypothesis obtained at the end of classification that describes the root samples are weaker than those of soil samples, since the minority class concepts/soil samples are underrepresented. We review the other challenges and the solutions to this problem in the next section.

There are some reasons that 3D Convolutional networks are an appropriate option for our task. First, they can clearly take advantage of the spatial structure of our MRI scans. Especially, they can train local spatial filters which is helpful for the classification task. On the other hand, our root and soil classification strategy is pixel-wise binomial classification, therefore the method proposed is able to extract features from spatial dimensions by performing 3D convolutions. Based on the 3D convolution, different types of 3D CNN topologies can be proposed to analyze this kind of MRI scans. In our case, we simply used two 3D convolutional layers for continuous feature extraction and logistic layer for classification.

2 Related Work

In today's world, classification tasks are widely solved using different standard classification algorithms such as decision trees, support vector machines, neural networks, logistic regression and so on. In our classification strategy we do a pixel-level root and soil classification of 3D MRI scan of a soil grown plant. We build a learning method that, using MRI image as input, is able to discriminate between soil samples and root samples on that image. We investigate a class of deep neural networks, specifically the convolutional neural networks to set up this learning method. The main novelty of our approach is to use 3D convolutions on the whole MRI image, which is new in *TensorFlow* engine (Abadi et al., 2016). Another novelty of our study is that state-of-the-art research is using medical MRI images to determine whether or not specific patient has a disease. However, our study is concentrating on root and soil classification of MRI images rather than working on medical ones (Payan and Montana, 2015; Sarraf and Tofighi, 2016).

To ensure balanced distribution of the data, specific sampling techniques are utilized. (Estabrooks et al., 2004; Laurikkala, 2001; Weiss and Provost, 2001) have shown that for different base-level classification algorithms, a balanced data distribution provides better predictive performance than an imbalanced data distribution. The results obtained there justify the usage of sampling techniques for imbalanced datasets. For imbalanced learning, two easy-to-understand and convenient sampling methods shine: *(i)* random oversampling, *(ii)* random undersampling. The idea of the former is to replicate the minority class samples to balance the distribution with majority class samples. In this case, original dataset size increases with the replicas. However, the latter removes the data from the original dataset. Particularly, samples from the majority class are randomly selected and removed from the dataset in a way that distribution between majority concepts and minority concepts becomes balanced. For the root and soil classification of the refined MRI image we use random undersampling to remove overrepresented soil samples from the image where distribution is balanced with the root samples. Then the standard logistic regression classifier is applied to the balanced data.

3 Methods

Our learning method takes as input refined/reconstructed MRI image and for all pixels in that image, we do soil or root classification using the combination of 3D Convolutional filter and logistic function. In this regard, we use convolutional layers and logistic layer together. A convolutional layer with stride 1 produces same size output image as input. The output value for each pixel denotes a probability of belonging to the root. The probability value of each pixel is then transferred into 0 or 1 through logistic function. Here, probability which is around 0 denotes soil pixels and the bigger probability is related to the root pixels. Our method consists of a training phase and a testing phase. In the training phase, we get rid of imbalance in the MRI scan, filter, resample, and extract important pixels from 3D voxels. After selecting and extracting important pixels using rectangular sliding window/filter, we trained our classifier on the extracted root and soil pixels and their labels in a logistic layer. In the testing stage, we took another refined/reconstructed MRI scan at the same resolution, by adding uniform salt-and-pepper noise we pushed our learning method to denoise the scan and classify network output for root and soil. With that in mind, we detected classification accuracy of our classifier on an unseen root and soil samples and interpret the results. On the other hand, we also determined the training time for the MRI scan of the training plant.

3.1 Usage of sampling to balance the class distribution

Typically, the utilization of sampling methods in imbalanced learning applications consists of the replication or removal of an imbalanced data set by some techniques to get a balanced distribution. Many papers have been published for solving imbalanced learning problem in different domains with the usage of sampling (Chawla et al., 2002; Han et al., 2005; He et al., 2008; Liu et al., 2009). Common examples include: random oversampling, random undersampling, synthetic sampling, etc. In this specific implementation of our classification method, we opt for a simple approach where vast amount of soil pixels are removed from the MRI scan to make it evenly distributed with root pixels. In a broader extent, consider pixels in a given training MRI image as a set S with m pixels (i.e., $|S| = m$). The we define S_{soil} as a set of majority soil pixels and S_{root} as a set of minority root pixels where $S_{root} \cup S_{soil} = \{S\}$ and $S_{root} \cap S_{soil} = \{\Phi\}$.

Moreover, any sets produced from sampling applied on S are labeled as E. Then, in random undersampling procedure, we randomly select a sample set E of majority soil pixel samples in S_{soil} and remove these samples from S where $|S| = |S_{root}| + |S_{soil}| - |E|$. This approach is used in all our experiments. However, for a test image, we did not perform this imbalanced learning technique where we relied on balanced classification performance of our trained classifier. However, in the presence of application specific domains, the intensity and optimality of the sampling procedure can be enhanced.

3.2 Convolutional filter selection and different representations

For a set of labeled root and soil pixels, each pixel's neighborhood is scanned for the attributes of neighboring pixels. Neighborhood attributes are produced by imposing a pre-defined size filter centered at the chosen pixel, in which pixels inside the filter are regarded as neighboring pixels. In our learning method, we choose filter size K for serialized 1D representation, square filter size $K \times K$ for 2D representation with K channels and cube filter size $K \times K \times K$ for 3D representation, centered at the chosen pixel (K is an odd number, typically 3 or 5). Neighborhood pixels' attributes inside the filter are chosen and reordered to enclose a K, K^2, or K^3 attribute/feature vector associated with the chosen pixel. According to the size of the kernel (3×3, 5×5, etc), the filter uses attributes of neighboring pixels of the centered pixel to produce continuous value. Then that value is transferred to a probability value between 0 and 1 by the logistic layer. Basically, convolution is a procedure on two functions f and g, which generates a third function that can be judged as a *filtered* version of f. In this explanation, we consider g as the filter. In general, for functions $f(x)$ and $g(x)$ of a continuous variable x, convolution is specified as:

$$f(x) * g(x) = \int_{-\infty}^{\infty} f(\tau) \cdot g(x - \tau) d\tau$$

where $*$ denotes convolution and $\cdot$ denotes normal multiplication.

3.3 Testing on a noisy data

There are two basic ways to corrupt the images: (i) by adding Gaussian noise or (ii) by applying salt-and-pepper noise. Salt-and-pepper noise is a kind of noise that is used sometimes to corrupt/reconstruct the image in encoder-decoder tasks. It presents itself as sparsely occurring white and black pixels in different parts of the image. One method can be the normalization of all pixels in a certain interval. But, an effective noise reduction method for this type of noise is a median filter or a morphological filter (Jayraman and Esakkirajan, 2009) . When we consider autoencoders, they minimize the lost function $L(x, g(f(\hat{x})))$, where $\hat{x}$ is a copy of the original data point x that is corrupted by salt-and-pepper noise or any other kind of noise. In our learning method, we trained our convolutional layers and logistic layer network without noise, and then testd with noisy MRI scan of the test plant. We did not specifically use autoencoders, because we did pixel-level classification and scanning neighboring pixels of the centered pixel through filter ensures noise detection and removal. As a loss fucntion we used cross entropy with logits:

$$H(p,q) = -\sum_{x} p(x)\, \log q(x).$$

where p and q represent discrete probability distributions (De Boer et al., 2005).

4 Results

In our empirical evaluation all experiments are performed on the same computer with **Intel(R) Core(TM) i7-3610QM** processor. The environment provides 8GB of RAM with 8 CPUs at the frequency of 2.3 GHz. Our NVIDIA graphics card does not support current *TensorFlow* version. Therefore we run the experiments on CPU only.

The MRI scan dataset that we have used for training and test phase consists of MRI images of 5 different plants at 5 different spatial resolutions. The resolutions are $56 \times 206 \times 56$, $75 \times 275 \times 75$, $111 \times 411 \times 111$, $148 \times 548 \times 148$, and $221 \times 821 \times 221$. We chose the resolution of $111 \times 411 \times 111$ to save training time and memory space. In the test phase, we tested our classifier on a different plant at the same resolution. As we mentioned before all MRI scans are refined/reconstructed 3D plant root images. They are virtually generated; synthetic MRI images that are produced using *SimRoot*, a functional-structural model that simulates the architecture of plant roots (Postma and Lynch, 2011). Figure 1, shows how synthetic MRI scan of a soil grown plant looks like.

As you can see it is the resolution of a plant at $221 \times 821 \times 221$. Number of pixels is 40,098,461 which encompasses both root and soil pixels in the image. On the other hand, number of root pixels is 130,707 in MRI scan which is making relative imbalance between root pixels and soil pixels be equal to *306:1*. However, we have chosen MRI scan with the resolution of $111 \times 411 \times 111$ where the between class imbalance is *134:1*. This imbalance is significant which can decrease predictive performance of standard classification algorithms. Therefore, we randomly undersampled soil pixels to make it relatively comparable to the root pixels.

After solving imbalance issue, we builded upon two successive convolutional layers with logistic layer. For both convolutional layers we used *TensorFlow's* **tf.nn.conv3d()** and **tf.nn.max_pool3d()** functions with *convolution patch size = 3* and *strides = 1*. With this we get the same size output image as input. We also apply **tf.nn.bias_add()** and **tf.nn.relu()** to the output of convolutional layers before moving it to the next layer. At the end, we vectorize the output of the second pooling layer and pass it to the logistic layer for classification.

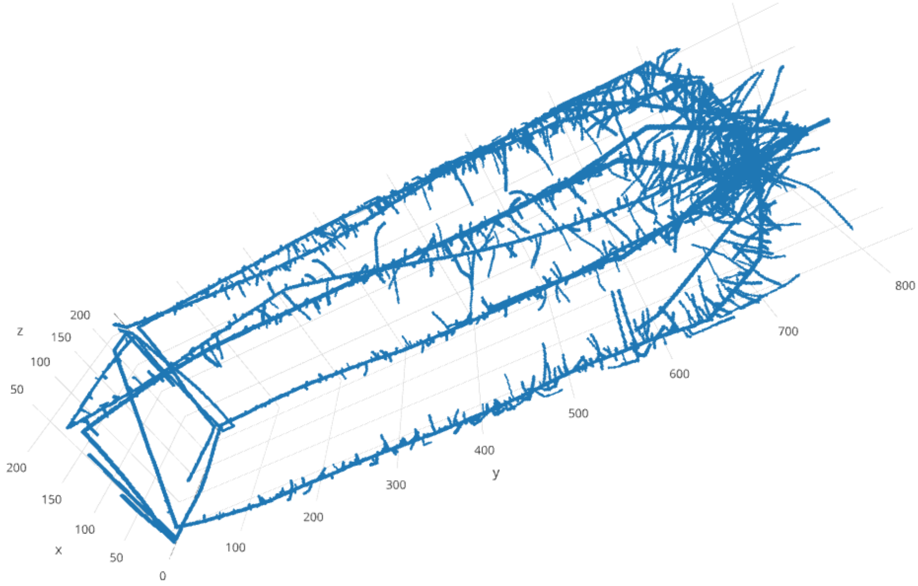

Fig. 1. Refined and reconstructed synthetic MRI image of a soil grown plant at the resolution of $221 \times 821 \times 221$.

In binary logistic regression, output is one continuous variable and there are two states for that variable either 0 or 1. And we use Logistic (softmax) function

$$\sigma(z) = \frac{\exp(z)}{\exp(z) + 1} = \frac{1}{1 + \exp(-z)}$$

where it is used to model the probability of input being a part of either binary class 0 (*soil*) or 1 (*root*). We minimized the error using cross entropy. For that purpose, we utilize **tf.reduce_sum()** for the output of the network topology and original class label.

After training we tested our learning method in an MRI scan of a different plant. By adding salt-and-pepper noise uniformly at the rate of 0.15 we obtained a noisy MRI scan to test the method. This rate corresponds to SNR (signal to noise ratio) of 150 which is an achievable SNR in real MRI data. It is the only SNR level that we tested in our experiments. SNR of MRI scan is calculated with following way:

$$snr = \frac{\#pixels_{root}}{\#pixels_{noise}}$$

Our method is robust enough so that by scanning the neighborhood of the centered pixel it successfully denoises the pixels in a given noisy MRI scan and outputs an approximation for the original refined/reconstructed version of it.

We utilized 1 convolutional filter in both convolutional layers. Training and test data corresponds to MRI scans of two different plants at the same resolution.

For detecting classification accuracy we used normal formula:

$$acc = \frac{(tp + tn)}{(tp + tn + fp + fn)}$$

where tp refers to true positives, tn to true negatives, fp to false postive and fn to false negatives.

Because, of time constraints our implementation is not fully finished. Therefore, we do not report classification accuracy, output image generated by the network and training time of our learning method. However, as we do pixel-level classification, depending on the noise level the classification accuracy can reach up to 99%.

5 Conclusion

We showed strength of our method by training and testing it on virtually generated MRI scans. First, we took one plant for training and another plant for testing. Then we generated an approximation for an original test plant. However, single training and test method does not provide statistical stability as *n-fold* cross validation. As we have 5 MRI images, it is absolutely possible to do 5-fold cross validation which is more robust than normal training and test method.

In our novel approach we used 3D convolutions on the whole MRI image using the popular *TensorFlow* engine. 3D convolutions corresponds to $K \times K \times K$ representation of 3D volume. It would also be interesting to test our learning method in a serialized 1D vector and $K \times K$ 2D images + K-channel representations in terms of training time and classification accuracy. However, recent studies in medical MRI data suggest that 3D convolutions of the MRI image yields better performance than 2D convolutions (Payan and Montana, 2015).

Last but not least, we use random undersampling to overcome class imbalance issue. However, both random undersampling and random oversampling has its own disadvantages in an imbalanced learning. In the case of undersampling, the problem is pretty obvious: removing samples from the majority class (soil pixels) may cause the standard classifier to unlearn vital concepts related to the majority class. When we look at oversampling, the problem is a little cloudier than undersampling: since oversampling simply adds replicas to the majority class while increasing original dataset size, multiple data points become overrepresented which leads to overfitting (Mease et al., 2007). In this case, the training accuracy can be high, but the predictive performance on the unseen test data is commonly worse (Holte et al., 1989). As an alternative to these methods, sophisticated sampling techniques such as *EasyEnsemble*, *BalanceCascade* (Liu et al., 2009), *SMOTE* (Chawla et al., 2002), *Borderline-SMOTE* (Han et al., 2005), *Adaptive Synthetic Sampling (ADASYN)* (He et al., 2008), etc have been proposed where they do not possess the mentioned disadvantages. However, they generally increase learning time by bringing some overhead or computation costs to the learning method (Mease et al., 2007).

6 References

Abadi, Martın et al. (2016). "Tensorflow: Large-scale machine learning on heterogeneous distributed systems". In: *arXiv preprint arXiv:1603.04467*.

Chawla, Nitesh V. et al. (2002). "SMOTE: synthetic minority over-sampling technique". In: *Journal of artificial intelligence research* 16, pp. 321–357.

De Boer, Pieter-Tjerk et al. (2005). "A tutorial on the cross-entropy method". In: *Annals of operations research* 134.1, pp. 19–67.

Estabrooks, Andrew et al. (2004). "A multiple resampling method for learning from imbalanced data sets". In: *Computational intelligence* 20.1, pp. 18–36.

Han, Hui et al. (2005). "Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning". In: *International Conference on Intelligent Computing*. Springer, pp. 878–887.

He, Haibo et al. (2008). "ADASYN: Adaptive synthetic sampling approach for imbalanced learning". In: *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. IEEE, pp. 1322–1328.

Holte, Robert C et al. (1989). "Concept Learning and the Problem of Small Disjuncts." In: *IJCAI*. Vol. 89. Citeseer, pp. 813–818.

Jayraman, S and S Esakkirajan (2009). *veerakumar T,"Digital Image Processing"*.

Laurikkala, Jorma (2001). "Improving identification of difficult small classes by balancing class distribution". In: *Conference on Artificial Intelligence in Medicine in Europe*. Springer, pp. 63–66.

Liu, Xu-Ying et al. (2009). "Exploratory undersampling for class-imbalance learning". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39.2, pp. 539–550.

Mease, David et al. (2007). "Boosted classification trees and class probability/quantile estimation". In: *Journal of Machine Learning Research* 8.Mar, pp. 409–439.

Payan, Adrien and Giovanni Montana (2015). "Predicting Alzheimer's disease: a neuroimaging study with 3D convolutional neural networks". In: *arXiv preprint arXiv:1502.02506*.

Postma, JA and JP Lynch (2011). "Root cortical aerenchyma enhances the acquisition and utilization of nitrogen, phosphorus, and potassium in Zea mays L". In: *Plant Physiology* 156, pp. 1190–1201.

Sarraf, Saman and Ghassem Tofighi (2016). "Classification of Alzheimer's Disease Structural MRI Data by Deep Learning Convolutional Neural Networks". In: *arXiv preprint arXiv:1607.06583*.

Weiss, Gary M and Foster Provost (2001). "The effect of class distribution on classifier learning: an empirical study". In: *Rutgers Univ*.