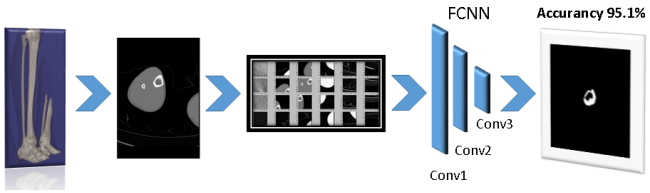
BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2016
http://excel.fit.vutbr.cz

# FCNN training for segmentation of CT scans with limited dataset

David Hlavoň*




**Abstract**
Thanks to the increase in computation performance observed in past decade convolutional neural networks started to gain momentum in image processing. This paper deals with segmentation of 3D images, specifically computed tomography (CT) scans, using Convolutional Neural Network (CNN). The paper shows efficacy of training the neural network on whole images or patches when limited by the size of dataset and computation performance. The solution to this problem lies in using Fully Convolutional Neural Network (FCNN) with no special preprocessing or postprocessing procedures except creating patches and composing them into whole images. FCNN has to work effectively and with satisfactory scoring in segmentation. Paper shows several FCNN configurations and training results with different size of patches and different approaches to training FCNN on whole images. Also an approach to the selection of patches for training, which helps to make training and evaluation of FCNN more effective, is shown. The experiments were run using BVLC Caffe framework. The results show that training on patches is more effective than training on whole images when limited by the size of the dataset. Paper shows that for results with score over 90% it is sufficient to use small-scale FCNN with no preprocessing and postprocessing procedures.

**Keywords:** FCNN — CT scans — segmentation – limited dataset

**Supplementary Material:** No supplementary materials

*hlavon.david@gmail.com, *Faculty of Information Technology, Brno University of Technology*

## 1. Introduction

**[Motivation]** Segmentation of CT scans is an important and challenging task in medical diagnostics. CT images often differ in quality and vary in noise levels, which makes recognition of target object for segmentation a difficult task. FCNN can learn, if provided with sufficient dataset, how to segment objects in low quality and noisy CT scans.

**[Problem definition]** This paper deals with pixel-wise segmentation of tibia (shin bone). The primary problem when training on medical datasets is limited amounts of available CT scans of different patients. Final FCNN will be used as a part of application for end users, where users would segment their own dataset, train FCNN and manage segmentation of their own CT scan. Therefore the challenge is to find solution for training simple FCNN with small dataset and with common PC components. Comparison of the outcomes of different approaches - training on whole images versus patches, is shown. The results of segmentation are rated by F-measure metric using precision and recall [1].

**[Existing solutions]** There are several diffent approaches to pixel-wise segmentation.

Jonathan Long uses fully convolution neural network for segmentation 2D images. Instead of last fully connected layers there are convolution layers with size of kernels producing output size $1x1$ pixel. The last layer there is a deconvolution layer which has inverze behavior to convolution layer and its parameters are set to produce original image size as input has [2]. This approach is faster than fully conected layers. Scheme of this neural network is shown in figure 1. Other solution

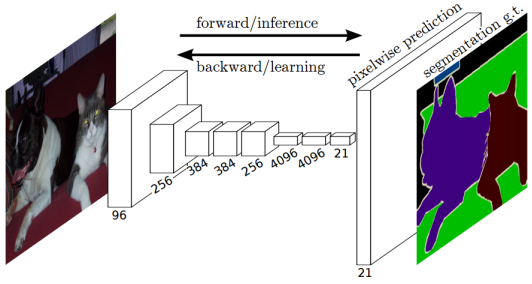



**Figure 1.** FCNN with deconvolution layer[2]

[3] of pixel-wise segmentation of 2D images uses two components. The first component is FCNN and the second component uses Conditional Random Field (CRF). The output of FCNN is upsampled by bi-linear interpolation. A fully connected CRF is applied to refine the segmentation result. The result is that segmented object has much better border segmentation against FCNN. In next solution of 2D image segmentation [4] there is presented two stage training for segmentation of 2D images when first stage is regular FCNN and second stage is deconvolution neural network. Both networks are connected by fully connected layers for image classification. Deconvolution neural network is mirrored version of FCNN. Output size of this neural network is same as input, but time of training is double times more then only FCNN because of devonvolution stage. Scheme of deconvolution network is shown in figure 2. Solution of segmentation 3D images specially

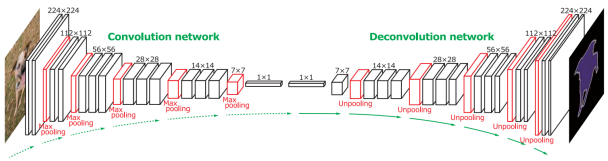



**Figure 2.** Deconvolution neural network[4]

CT scans presents [5]. This approach combines several methods for creating two segmentation frameworks for segmentation pankreas in CT image. Input image is preprocessed by algorythm SLIC which create image of superpixels [6]. This superpixel image is segmented by CNN and dense output of CNN is process by superpixel random forest classifier. Schema of CNN used for segmenting pancreas is shown in figure 3.

**[Our solution]** The main approach is to use small,

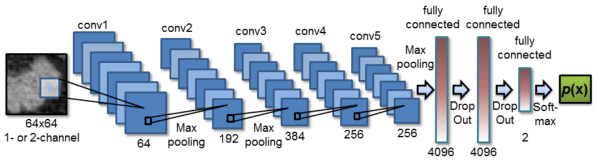



**Figure 3.** CNN for segmentatation of pankreas[5]

effective and fast FCNN with no additional processing, example of simple FCNN is shown in figure 4. FCNN works with CT scans in DICOM standart, which also defines that CT image consists of slices and every slice is saved in separate file. This paper talks about one slice as whole image, because input of CNN is image with 1 channel. So thanks to this standart problem of segmentation of 3D images is converted to segmentation of separated 2D images forming the 3D image.

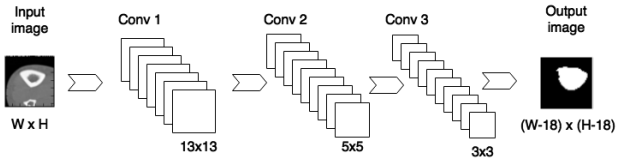



**Figure 4.** Simple FCNN

Because of small dataset the idea is to train FCNN on image patches, because every patch becomes an input image and dataset becomes several times larger. Using of trained FCNN expects that input volume is cut into patches and every patch is segmented separately. We also apply approach of selecting patches which reduce dataset of useless patches which contain only background. For comparison we also train network on whole images.

**[Contributions]** Our solution reaches score over 90% with metric F-measure only after 20 000 iterations of mini batches. Training of FCNN with best configuration takes about 1 hour. In our approach is problem with borders of segmented bone. Using addtional postprocessing like CRF may increase quality of borders of segmented bone.

## 2. Metodology

### 2.1 Dataset

We have avaible dataset consists of 10 CT scans in DICOM format and it consiscts of 9725 images. 9 of CT scans (8817 images) was used for training and 1 CT scan (908 images) was used for testing. CT volumes show scaned lower leg and there is segmented a shin bone (tibia) in these volumes. For creating dataset we used several python scripts. One of scripts creates patches from whole DICOM volume and cut labeled volumes same way as data volumes. Labeled patches had to be normalize to labels $0-background, 1-tibia$.

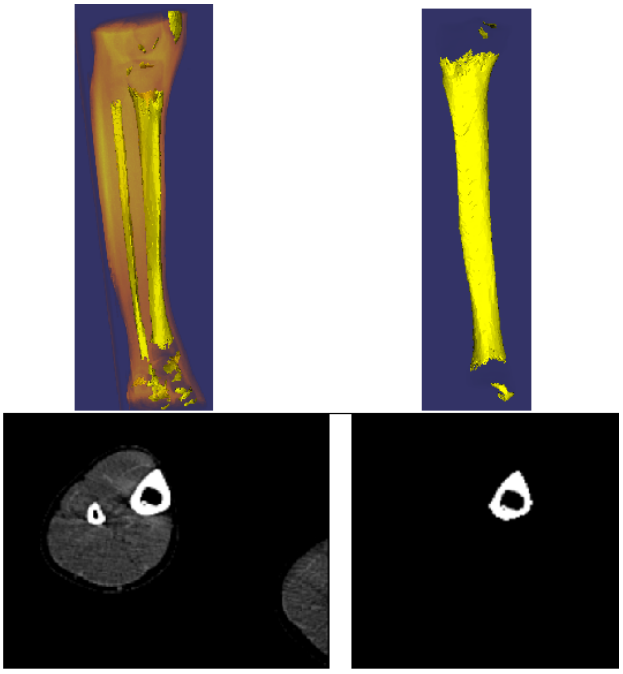


**Figure 5.** Data DICOM volume and segmented labels in 3D and sample of 2D slice from dataset

Then we create LMDB database for Caffe data layer as input to FCNN. In terminology of Caffe pathes have size of $CxWxH$ where $C = 10, W - H$ respectively. $C$ represents channels. Every created dataset is designed for pixel-wise segmentation and labels create separate LMDB database off data LMDB database. Important for training is uniform distribution of positive labels and negative labels in dataset which improves process of traing. For uniform distribution we created python script which loads all data patches or whole images into array and shuffles their indices during creating LMDB. LMDB databases for data and labels are created according this indices.

## 2.2 Train on patches

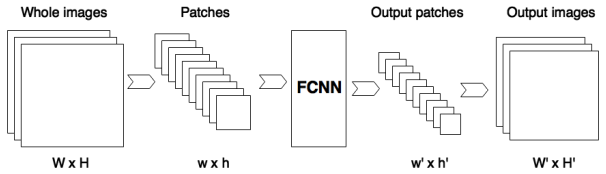



**Figure 6.** Training on patches. Whole images are cut into patches. These patches go through FCNN, output patches are downsampled. Output patches are put together to create whole donwsampled image.

Patches was created as 2D image with size of $WxHxD$ where $W = H, D = 1$ respectively. Script for creating patches loads whole DICOM volume and iterate through slices one by one. Script cuts slice into patches of size $WxH$ by vertical and horizontal step with output size of FCNN $W_{FCNN}xH_{FCNN}$. Important is overlapping of created patches from DICOM volume. Example of cutting image to patches with overlap size same as output of FCNN is shown in figure 7. Depending on size of patch this approach increase size of dataset to 88535 and 125 817 images.

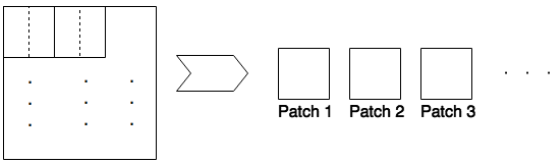



**Figure 7.** Creating of 2D patches with overlapping. Step of trimming window is smaller than size of patches.

## 2.3 Train on whole images

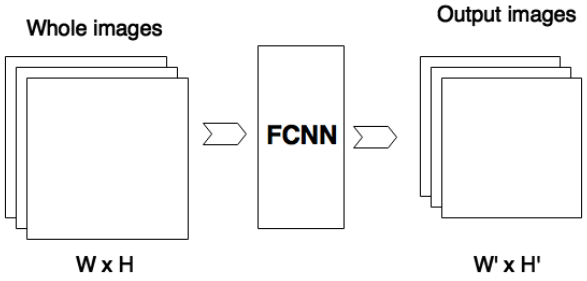



**Figure 8.** Simple FCNN trained on whole images. These images go throught FCNN and whole donwsampled images are output.

Dataset for whole image training consider every slice of volume as input patch so size if dataset is only 8817 images for training. This approach is shown in figure 8 Trainnig on this dataset cosumes more computer performance than training on patches.

## 2.4 Patch selection for training

The approach of creating dataset with patches brings problem of data patches with no useful information. These patches contains only background. We use approach which check every data patch if it contains at least one pixel different from background during creating LMBD database. If patche doesn't contain any pixel different from background so data patch and label patch are not included to training dataset. This approach is omited during creating dataset of whole images because every slice contains at least one pixel different from background. This approach is also applied on trained FCNN. Every evaluated patch is checked and if contains only background then this patch doent't go as input to FCNN but new patch with zero activations and size of output of FCNN is created and it is inserted on particular position into output array which contains segmented patches of volume.

### 2.5 Neural net configurations

Every neural net configuration read data from separate LMDB database and labels from next sepatare LMDB datase and it caused by pixel-wise segmentation where labels are images too and every pixel of this image represents one of two segmentation classes, background and shin bone. We trained small FCNNs with many kernel outputs and FCNNs with few kernel outputs. We combined approaches of trainong on patches and whole images. We trained on two sizes of patche $64x64$ pixels and $46x46$ pixels. Whole image was normalized to size $192x192$ pixels. We trained using Stochastic gradient Descendant with 32 size of mini-batch. We trained with fixed learning rate during training, but varying by FCNN configurations. We trained FCNN with many configuration deffering in number of layers from 1 to 2, number of kernel outputs 32,512 and 1024, activation functions *TanH* or *ReLU* and learning rate $5x10^{-7}, 10^{-7}, 5x10^{-6}, 10^{-6}$. For comparison this paper show only interesting results and best configuration in compare to other traing approaches, whole images and different size of patch.

## 3. Experiments

Every experiment was done for patch approach with 2 sizes of patches, $64x64$, $46x46$ and whole image approach. We train with Stochastic Gradient Descendant. At first we tried to train FCNN with basic configuration consists of 3 convolution layers, $13x13$, $5x5$ and $3x3$ with kernel outputs 512, 1024 and 1 for the first, the second and the third layer respectively for patch approach. For training on whle images we used count of kernel outputs 96,128 and 1 for the first, the second and the third layer respectively. Learning coeficient was $5x10^{-7}$ and $10^{-7}$ for patch and whole images training. As activation function was choosen Hyberbolic Tangens.

We tried to change count of kernel outputs and other parameters of FCNN stood same in next experiment. So we get FCNN with 3 convolution layers witch count of kernel outputs 32 , 32 and 1 respectively for both approaches.

Next experiment was characteristed by changing the activation function to Rectified Linear Unit (ReLU). Other parameters was same as previous configuration. Next configuration of FCNN had set the learning rate to 10 times bigger value than value before. New learning rate was established to $10^{-6}$ and $5x10^{-6}$ for patch approach and whole image approach respectively. Configurations are summarized in table 1.

**Table 1.** Configurations of FCNN for experiments. The first column specify number of experiment. The second column specify activation function. The third column specify learning rate. The last column specify configuration of FCNN. For each layer there is size of kernel and number of outputs. Values for training on patches are separated off values for training on whole images by comma.

| | Act.func. | LR | FCNN conf. |
|---|---|---|---|
| 1 | TanH | $5x10^{-7}, 10^{-7}$ | $13x13-512,96$<br>$5x5-1024,128$<br>$3x3-1,1$ |
| 2 | TanH | $5x10^{-7}, 10^{-7}$ | $13x13-32,32$<br>$5x5-32,32$<br>$3x3-1,1$ |
| 3 | ReLU | $5x10^{-7}, 10^{-7}$ | $13x13-32,32$<br>$5x5-32,32$<br>$3x3-1,1$ |
| 4 | ReLU | $5x10^{-6}, 10^{-6}$ | $13x13-32,32$<br>$5x5-32,32$<br>$3x3-1,1$ |

## 4. Results

One CT scan, that was not used for training, was used for FCNN success rate evaluation. When training on patches, the score was calculated for all the patches, which were selected according to training patch selection methodology. Precision and recall were used for calculating the score of segmentation. Our best configuration of FCNN gives 95.1% accuracy in segmentation. Figures 9 , 10, 11 show input patches, label patches and output of FCNN respectively. This score was reached with configuration number 4 with training on $46x46$ patches. The training on patches was much faster than training on whole images. For instance training of FCNN with the best scoring configuration on patches took 1 hour and segmentation of test CT scan using this FCNN took about 15 minutes. FCNN with equal configuration trained on whole images took about 3 hours with lower score outcome. Evaluation of FCNN with high number of kernel outputs takes about 12 hours when using FCNN trained on patches. Results are summarized in table 2.

## 5. Conclusions

This paper deals with problem of different approaches to training Fully Convolutional Neural Network – patches versus whole images – when only limited training dataset is available. It shows that training on patches

**Table 2.** Result of experimetns. Score is measured with F-measure metric.

| Experiment | Patch 64$x$64 | Patch 46$x$46 | Whole images |
|---|---|---|---|
| 1 | 0.92469 | 0.93955 | 0.50371 |
| 2 | 0.92161 | 0.94 | 0.46274 |
| 3 | 0.92394 | 0.94376 | 0.5088 |
| 4 | 0.93 | 0.951 | 0.61293 |

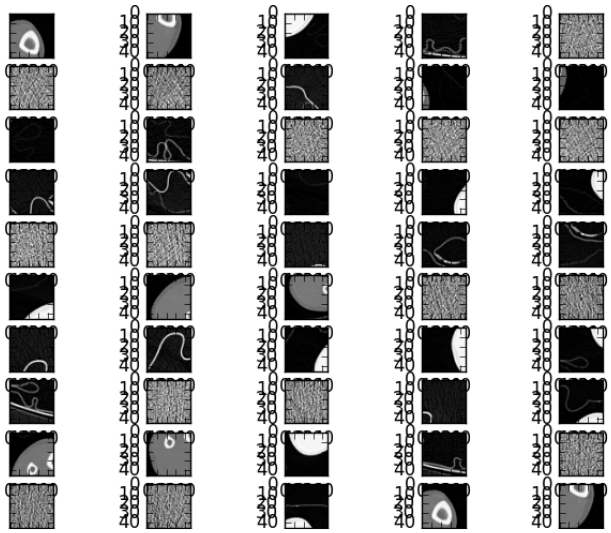


**Figure 9.** data patches for the FCNN.

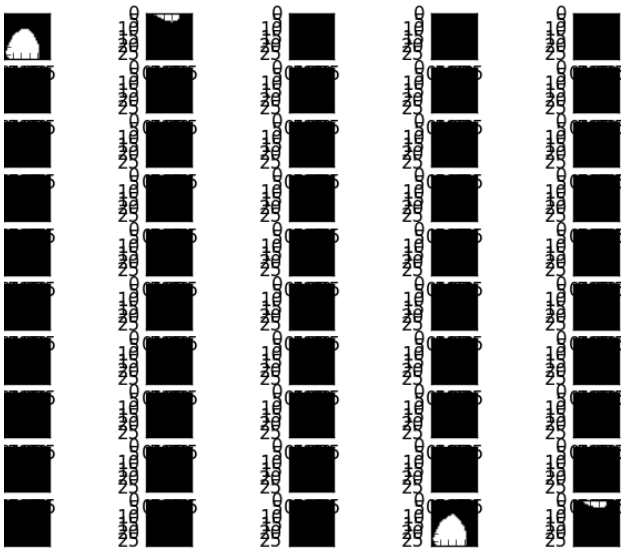


**Figure 10.** Ground truth patches.

is superior to whole image training mainly in effectivity. Uniform distribution of segmented classes in dataset is important, namely when FCNN is trained on patches. Segmentation technique score is measured with precision and recall F-measure. The best configuration of FCNN gives score with 95.1% accuracy in segmentation. This score was reached with patch size of 46$x$46, ReLU activation function, learning rate $5 * 10^{-6}$ and with 3 convolution layers $13x13-32$, $5x5-32$, $3x3-1$. Future work will be focused on improving border segmentation.

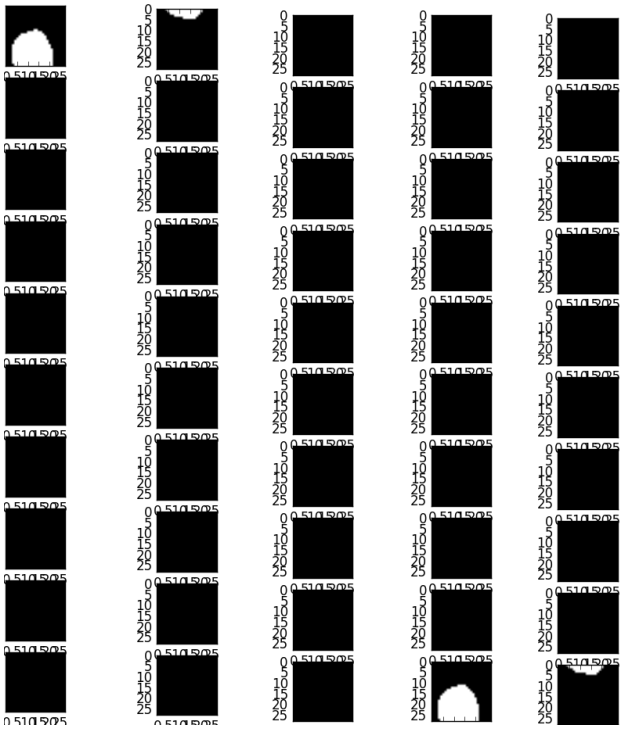


**Figure 11.** Dense output.


## Acknowledgements

I would like to thank my supervisor Ing. Michal Španěl Ph.D. and Ing. Michal Hradiš Ph.D. for helping me during solving this issue. I also want to thank to 3Dim Laboratory s.r.o. for providing me the dataset.


## References


[1] Krzysztof Dembczynski. Optimizing the f-measure in multi-label classification: Plug-in rule approach versus structured loss minimization. http://jmlr.org/proceedings/papers/v28/dembczynski13.pdf.

[2] Trevor Darrell Jonathan Long, Evan Shelhamer. Fully convolutional networks for semantic segmentation. http://www.cs.berkeley.edu/~jonlong/long_shelhamer_fcn.pdf.

[3] George Papandreou Iasonas Kokkinos etc., Liang-Chieh Chen. Semantic image segmentation with deep convolutional nets and fully connected crfs. http://arxiv.org/pdf/1412.7062v3.pdf.

[4] Bohyung Han Hyeonwoo Noh, Seunghoon Hong. Learning deconvolution network for semantic segmentation. http://www.cv-foundation.org/openaccess/content_iccv_2015/papers/Noh_Learning_Deconvolution_Network_ICCV_2015_paper.pdf.

[5] Le Lu Senior Member IEEE etc., Amal Farag. A bottom-up approach for pancreas segmentation using cascaded superpixels and (deep) image patch labeling. http://arxiv.org/pdf/1505.06236.pdf.

[6] Kevin Smith Aurelien Lucchi Pascal Fua Radhakrishna Achanta, Appu Shaji and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. EEE Transactions on Pattern Analysis and Machine Intelligence.