Categorization of ultrasound images of carotid arteries

MARTIN KOSTELANSKY

Faculty of Electrical Engeneering
Czech Technical University
kostema3@fel.cvut.cz

Abstract

The computer vision has entered many domains in the recent years and together with deep learning are taking place in many different domains. In our work we aim to sort thousands of ultrasound carotid images of different categories, namely traversal, longitudinal and conical. We compare performance of one small CNN and two deep CNN - VGG16 and ResNet50 with different types of preprocessing. We also try to take an advantage by enriching our training dataset with different, already sorted, dataset. The deep architectures has show their ability to be fit on new data and the test accuracy has reached 100%.

I. INTRODUCTION

In the last decade, convolutional neural nets have been setting new benchmarks in many domains, namely image recognition [1, 2]. Many neural network architectures exist, such as recurrent CNN [3], very deep CNN [1], NN with depth-wise separable convolutions [4] and deep residual CNN [5], etc. This research ha been widely applied in many fields, including medicine and processing of biological images. Convolutional neural nets have shown their potential in segmentation[6], localization[7] and classification[8].

As a first step, when using convolutional net is data analysis. In our work we deal with thousands images of carotid arteries of different types, which need to be sorted in order to use them further. For supervised learning are crucial labeled data and thus, we have decided to combine by-hand sorted subset of original images combined with Ultrasound image database dataset from Signal processing laboratory at VUT Brno which is already labeled.

	Prague	Brno - Train	Test
Traversal	383	84	102
Longitudinal	175	84	45
Conical	123	-	66

Table 1: *Sizes of train and test sets for all of the categories.*

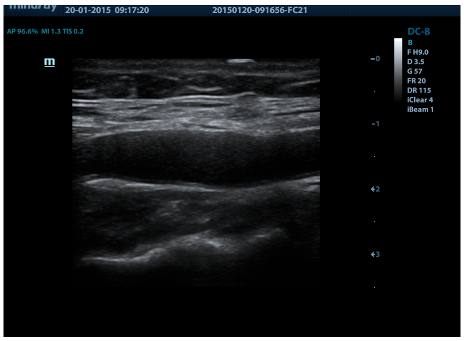
II. DATASET

Our dataset consists from thousands carotid arteries images of three types, namely traversal, longitudinal and conical, see 1. We categorized subset of our image database (this subset will be called for simplicity Prague dataset for simplicity). This set was divided into train and test groups. In some of our experiments we used Brno dataset to enlarge longitudinal and traversal categories, sample images can be seen in 2. The sizes of these are defined in 1.

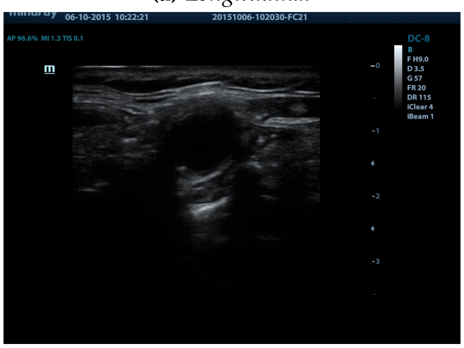
III. CLASSIFICATION

i. Preprocessing

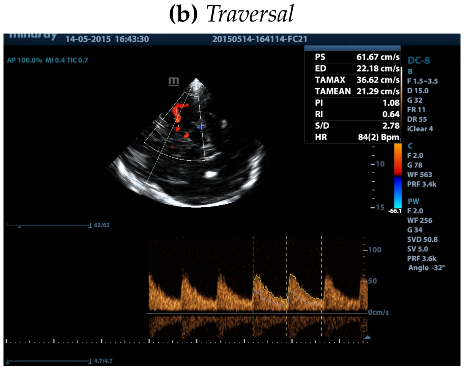
We are comparing performance of three different architectures of CNNs, well described in



(a) *Longitudinal*

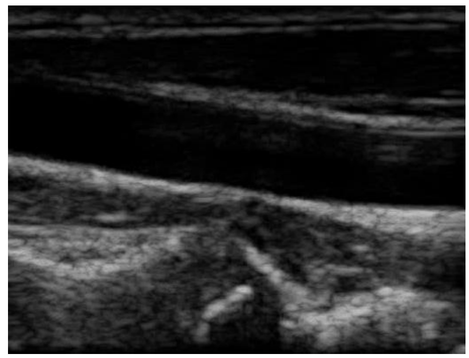


(b) *Traversal*



(c) *Conical*

Figure 1: *The examples of categories in the dataset.*



(a) *Longitudinal*

(b) *Traversal*

Figure 2: *The examples of categories in the Brno dataset.*

Simple	Complex
Resize Normalize	RandomHorizontalFlip RandomVerticalFlip Resize Normalize GaussianNoiseTransform

Table 2: *Simple and complex set of transformations used in the preprocessing phase.*

Type	Shape
Conv2d	4, 5
MaxPool2d	2
Conv2d	8, 5
MaxPool2d	2
Linear	128

Table 3: *The architecture of small CNN (Shape for Conv2d represents number of filters and kernel size, for MaxPool2d kernel size and for Linear number of neurons).*

subsections ii and iii. In all of the experiments we are using same sets of transformation functions during the preprocessing. In first set of such functions consists from resizing to the input size and normalization of all channels (naturally, in the case of Prague data, the artery was cropped). Since we have discovered different noise in the Brno and Prague dataset, we add following transformations, in order to increase robustness of the classifier and enlarge the training data set, which are described in the Table2.

ii. Small CNN

As our base line model we use a small CNN with input shape 28x28x3 and it is composed from two convolutional layers followed by maxpooling layer ended by a hidden fully connected layer. The detailed architecture can be seen bellow, in the Table 3.

iii. Deep CNN

The next step in the search for an optimal model we tired to fine tune VGG16 [1], as well as ResNet50 [5] pretrained on the ImageNet dataset [9]. In both of these nets we have created new last linear layer and finetuned the rest of them.

Small CNN		
	Simp. pr.	Com. pre.
Prague	**0.28166**	0.46043
Prg & Brn	0.32735	2.75207
VGG16		
	Simp. pr.	Com. pre.
Prague	**0.00794**	0.02971
Prg & Brn	0.00936	0.02072
ResNet50		
	Simp. pr.	Com. pre.
Prague	0.01334	0.01325
Prg & Brn	**0.01086**	0.01362

Table 4: *The vest validation negative log-likelihood for every model, preprocessing, data combination.*

IV. Results and discussion

All of three models described above, we trained on both of the datasets (Prague and Prague + Brno) each preprocessed with two possible sets of transformations - 4 experiments for every architecture. In every training, we separated 20% and used it as a validation, on which was particular model evaluated during the training and we selected the model with the lowest negative log-likelihood. This validation results are described in the table 4. The best results for each of the model on the test data are in the table 5.

The ResNet50 has achieved the lowest neg. log-likelihood **0.02176** when trained on combination of Prague and Brno dataset preprocessed with simple preprocessing and its accuracy reached **1.0** and it did not misclassify any of the 213 test images.

To use this model in the production, one might to investigate the probability distributions of the probabilities for the classes and select a rejection threshold ensure that model do not make mistakes.s

Model	DS	Preproc.	Neg. LL	Accuracy
Small CNN	P&B	Smpl.	1.20502	0.845
VGG16	P&B	Smpl.	0.02858	0.995
ResNet50	P	Cmplx.	**0.02176**	1.0

Table 5: *Best test negative log-likelihood and accuracy for every model.*

V. Implementation notes

This work has been implemented in Python 3.7 and the models were created using PyTorch 1.6. This project can be found on GitHub.

References

[1] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, 09 2014.

[2] S. Ghosh, I. Das, N. Das, and U. Maulik, "Understanding deep learning techniques for image segmentation," *ACM Computing Surveys*, vol. 52, p. 40, 05 2019.

[3] M. Liang and X. Hu, "Recurrent convolutional neural network for object recognition," pp. 3367–3375, 06 2015.

[4] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, vol. abs/1704.04861, 2017.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[6] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

[7] B. D. de Vos, J. M. Wolterink, P. A. de Jong, T. Leiner, M. A. Viergever, and I. IÅągum, "Convnet-based localization of anatomical structures in 3-d medical images," *IEEE Transactions on Medical Imaging*, vol. 36, no. 7, pp. 1470–1481, 2017.

[8] J. Chang, J. Yu, T. Han, H. Chang, and E. Park, "A method for classifying medical images using transfer learning: A pilot study on histopathology of breast cancer," in *2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pp. 1–4, 2017.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.