Stenosis Detection with Deep Convolutional Neural Networks

Karol Antczak^{1,*} and Łukasz Liberadzki¹

¹Military University of Technology, Institute of Computer and Information Systems, Warsaw, Poland

Abstract. Recent popularity of deep learning methods inspires to find new applications for them. One of promising areas is medical diagnosis support, especially analysis of medical images. In this paper we explore the possibility of using Deep Convolutional Neural Networks (DCNN) for detection of stenoses in angiographic images. One of the biggest difficulties is a need for large amounts of labelled data required to properly train deep model. We demonstrate how to overcome this difficulty by using generative model producing artificial data. Test results shows that DCNN trained on artificial data and fine-tuned using real samples can achieve up to 90% accuracy, exceeding results obtained by both traditional, feed-forward networks and networks trained using real data only.

1 Introduction

1.1 Deep Learning concept

Neural networks are used as a method to learn from observational data without the need to formally specify all the knowledge the computer needs. Historically, neural networks were able to learn to recognize concepts or patterns but were limited in terms of their complexity due to restrictions on number of layers. Deep learning is a set of techniques that allows neural networks to learn from experience and understand the world in terms of a hierarchy of concepts, with each concept defined through its relation to simpler concepts [1].

Deep learning approach for problem solving makes it one of state-of-the-art tools in the image processing field. As a matter of fact, in 2017 nearly all teams participating in Image Net Large Scale Visual Recognition Challenge [2] used deep learning-based methods. Description of the presented input through the hierarchy of concepts allows deep learning methods to unveil hidden features in the image that may not be noticeable to human researcher.

1.2 Deep Learning in medical imaging

Proven effectiveness of deep learning in other areas entailed multiple proposals of uses in medical imaging. In radiology these includes [3]: segmentation of lungs, tumours and other structures in the brain, biological cells and membranes, tibial cartilage, bone tissue and cell mitosis. Additional use cases include automatically annotating chest radiographs with diseases and descriptions of the context of a disease, for example location, severity and list of affected organs. A report by Litjens G. et al. [4] provides more details about use of

deep learning models in other fields of medicine. They include, among others, classification of skin lesion with multi-stream convolutional neural network, detection of regions of interests around anatomical regions (heart, aortic arch and descending aorta) on CT scan, estimation of local similarity between CT and MRI images of the head through two types of stacked auto-encoders.

1.3 About this paper

In this paper, we describe a problem of identifying stenosis on coronagraphic images. Afterwards, analysis of currently used commercial grade solutions is presented. Next, a neural network-based approach to stenosis detection, using deep learning method in a form of convolutional neural networks, is proposed. During formulation of architecture of solution common problems, that one can encounter when dealing with medical imaging, are shown, along with proposed alleviations. Lastly, results of testing the proposed neural networks are presented.

2 Stenosis detection

A stenosis is an unnatural narrowing in any passage or orifice of the body [5]. A specific form, known as atherosclerosis, is a narrowing of arteries, caused by accumulation of white blood cells, cholesterol and triglycerides. It can occur, among others, in coronary arteries where it is a common cause of coronary artery disease (CAD).

*Corresponding author: karol.antczak@wat.edu.pl

2.1 Coronary angiography

There are many methods, more or less invasive, to detect coronary stenoses. One of the most frequent ones is coronary angiography. In this method, a thin catheter is inserted into an artery through a puncture made with a needle. The catheter is then used to inject a radiocontrast agent. This reveals a structure of arteries on X-ray images (see Fig. 1) allowing to visually detect stenoses and other visible abnormalities by a diagnostician. Coronary angiography is a fair and reliable diagnostic method, often used as reference standard for identification of obstructive stenoses [6].

Fig. 1. Example of coronary angiographic image with visible stenosis

In order to provide a diagnosis, one must manually review an angiogram. This requires an experienced physician, with a detailed knowledge of normal coronary arterial anatomy and its common variants. Without this, it is easy to misinterpret the angiographic findings, with potentially serious clinical consequences [7].

2.2 Computer-aided stenosis detection

Over the course of years, several feature detection methods were proposed for stenosis detection. However, most of them are model-based methods, meaning that they utilize a mathematical model of coronary vessels and tries to retrieve such model from coronary angiogram. For example, Arnoldi and others [8] investigated an algorithm used by COR Analyzer™ software. To the best of our knowledge, this is the most popular commercial solution for automated stenosis detection. It uses a statistical model (trained upon several hundred examples) for identification of coronary artery and other vessels. The reconstructed vessels are segmented and a set of parameters is extracted for each segment (like vessel and lumen cross-sectional area, presence and size of atherosclerotic plaque, bifurcations, noise level etc.). The parameters are then matched to the characteristics that were used during

training. We could not find information regarding exact algorithm used for matching. This method reached 0.74 sensitivity and 0.83 specificity on per-vessel analysis and 1.00 sensitivity and 0.65 specificity on per-patient analysis.

An alternative, proposed by us, is a classifier-based approach. Stenosis detection on angiographic image can be viewed as a typical object detection task. As such, it consists of three major steps: pre-processing, windowing and classification. The whole process is demonstrated on Figure 2. First, the input stenographic image is scaled down. In our case input images had resolution 256x256 and were scaled down to 128x128. This reduces the amount of data for later processing and serves as a very simple denoising technique. Next, image is windowed using a window of fixed size (32x32 in our case), producing a set of patches. Every patch is then classified with a binary classifier, i.e. having only two possible outputs, e.g. “P” if stenosis was not detected on a patch and “N” otherwise. Note that they may be multiple patches classified as “P”, even if there is only one stenosis on image, as seen in Figure 2. This is due to the fact that patches may overlap.

Fig. 2. General algorithm for classifier-based stenosis detection

2.3 Convolutional Neural Networks

There are many classifiers suitable for aforementioned stenosis detection algorithm. The one investigated in this paper is a *convolutional neural network (CNN)*. It belongs to deep learning methods and gained considerable popularity in recent years due to its relative simplicity (both conceptional and computational) combined with state-of-the-art performance for computer vision tasks like image classification [9] or face recognition [10].

CNNs consist of several layers of various types (see Fig. 3). The most important are *convolution* layers. They can be viewed as filters represented by matrices. Each filter is convolved across input, computing the dot product between the filter values and a fragment (called *receptive field*) of input. The second type is a *pooling* layer. They are used for reducing the spatial size of data by dividing it into small regions and taking maximum (*max pooling*) or average (*average pooling*) of values of each region. In practice, pooling layers are inserted between successive convolutional layers. Last layers of convolutional neural network are typically *fully connected* ones, with each neuron of previous layer connected to each neuron of next layer.

Fig. 3. Architecture of convolutional neural network

3 Artificial data generation

In order to train a good knowledge-based classifier, a lot of data is needed. In case of deep learning methods, the problem is even more crucial. Datasets used for training CNNs can vary from 10 samples per class (for face recognition [10]) to 29 160 samples per class (for character recognition [9]).

In our case, the initial dataset consisted of 250 patches 32x32 pixels each with values normalized to [0;1] range. Exactly half of them represented stenosis. Patches were produced from real steganographic images acquired from internal as well as internet sources. This is far too few for training deep classifier.

3.1 Data augmentation

A common strategy for dealing with small data sets is data augmentation [11]. In case of image classification popular approach is using combination of affine transformations, like rotations, translations and distortions, in order to produce modified images from original dataset. Another approach is to use another neural network, which, trained on initial data set, generates images for target network. This is known as generative adversarial networks (GANs) [12].

Both of these approaches have common weakness – they are based on initial dataset. As a result, all generated elements will be related (in terms of search space) to real elements. As such, classifier trained using this data will not acquire completely new knowledge, but will rather generalize what can be obtained from real data. It can be significant obstacle, if data set is very small.

3.2 Generating artificial data

To overcome problem of insufficient training data, one can generate completely artificial data, that according to some predefined model.

We introduced a simple algorithm¹ for generating artificial patches, which can be used for training. General assumption was that patches from angiographic images can be modelled as grayscale images representing set of curves of various lengths, drawn on a gradient background, distorted by some noise. The algorithm itself has following steps:

1. Draw random, background gradients
2. Draw several Bézier curves representing veins.

3. (Optional) Draw Bézier curves with narrowing, representing stenosis
4. Add white noise
5. Apply Gaussian blur to the whole image
6. Add white noise again

Figure 4 shows some real patches along with the generated ones. As one can see, proposed algorithm is far from generating faithfully looking images, and it is fairly easy to distinguish between real images and generated ones. Additionally, real negative set consisted mainly of images without veins at all, while generated patches had always at least one “vein” visible. Because of this, classifier should be trained two-way: first using artificial patches, then tuned up using real data set.

Fig. 4. Natural and artificial patches of angiographic images

4 Test Results

Our goal was to create the best CNN (in terms of sensitivity and specificity) for stenosis detection. This was a two-stage process. First, several networks with different configurations were trained using only artificial data. Then, the best-performing network was tuned-up using real patches.

4.1 Network configurations

Choosing optimal network configuration was performed using grid-search algorithm with some additional heuristics. As starting points, several well-known CNN architectures for image classification were used, developed by Simard & others [13], Simonyan & Zisserman [14] and Krizhevsky & others [15].

In total, 6 networks were created, described in Table 1. Network A was a typical Feed-Forward Neural Network with sigmoid activation function. Remaining networks were Convolutional Neural Networks with receptive field 7x7 in each convolutional layer and Rectified Linear Unit (ReLU) as an activation function. Each network had dropout layers after 1st, 3rd and 4th layer with dropout rate 0.5, applied during training. Network E had max pooling

¹ Source code can be found online at: github.com/KarolAntczak/DeepStenosisDetection

applied after last convolutional layer. Network F had additional dense layer after last convolutional layer.

patches, F_n is a number of wrongly classified negative patches. By “positive” patches, we mean those containing stenosis, whereas “negative” ones are patches without stenosis.

Table 1. Network configurations

net layer	A	B	C	D	E	F
1	128 Dense	8 Conv	8 Conv	8 Conv	8 Conv	8 Conv
2	1 Dense	16 Conv	16 Conv	16 Conv	16 Conv	8 Conv
3			16 Conv	8 Conv	8 Conv	8 Conv
4					Max Pool	8 Conv
5						16 Dense

Table 2. Network training results

net	loss	accuracy	val. loss	val. accuracy
A	0.67	0.59	0.69	0.54
B	0.44	0.79	0.57	0.71
C	0.41	0.81	0.49	0.76
D	0.46	0.78	0.46	0.78
E	0.43	0.79	0.44	0.79
F	0.43	0.79	0.45	0.80

4.2 Training

An artificial dataset was created for training, using algorithm introduced in previous chapter. It consisted of 10 000 grayscale images of fixed size 32 x 32, divided into training and validation sets with 8 000 and 2 000 elements, respectively. The only pre-processing was normalization of pixel values from [0;255] to [0;1] range.

Networks were trained with 1 000 epochs of stochastic gradient descent method with momentum 0.8. Loss function used was binary cross entropy, defined as:

$$\mathcal{L}(x, y) = -\frac{1}{n} \sum_{i=1}^n (x_i \log(y_i) + (1 - x_i) \log(1 - y_i)) \quad (1)$$

where x_i is label of i-th patch (1 for patch containing stenosis, 0 otherwise) and y_i is network response for that patch. For each training iteration, a batch of $n = 100$ elements was used.

Training results are presented in Table 2. For each trained network, loss and accuracy were calculated against training set (columns 2 and 3) and validation set (columns 4 and 5). Accuracies were calculated using following formula:

$$Acc = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \quad (2)$$

where T_p is a number of properly classified positive patches, T_n is a number of properly classified negative patches, F_p is a number of wrongly classified positive

4.3 Tuning

After training network on artificial data set, we have performed additional tuning using patches from real images. This time, only networks F and A (as a reference) were used. Real data set consisted of 250 patches, divided into tuning and test data sets, 125 elements each.

Algorithm used for tuning was, again, stochastic gradient descent with momentum 0.8. Because tuning data set was significantly less than training data set, only 500 epochs were performed. We have measured accuracy on training set in three cases: when only training is performed, only tuning is performed and both training and tuning are performed on network.

Table 3. Network tuning results

net	test accuracy		
	tuning only	training only	training + tuning
A	0.55	0.69	0.87
F	0.50	0.74	0.90

5 Discussion

5.1 Choosing optimal network configuration

As indicated by training results, in general, networks with more convolutional layer performed better than shallow ones. However, it seems that adding more than four convolutional layers doesn't provide significant performance boost. Increasing number of units in convolutional layer can result in network overfitting, as seen in network C, which had the best loss and accuracy on training set but performed poorly on validation set. Adding non-convolutional layer - max pooling (E) or dense (F) – at the top of convolutional layers can further improve the accuracy. The reference feed-forward network A obtained results worse than any of CNNs, indicating that CNN-based architectures are good choice for this task.

5.2 Effectiveness of tuning

As one can see in Table 3, attempting to train using only real dataset resulted in poor quality of networks. On the other side, using only artificial data set, both networks were able to properly detect some stenoses in test cases. This is significant result, due to the fact, that artificially generated training set was completely unrelated to test set. The best accuracy (0.90) was achieved with combining training and tuning for deep convolutional neural network F.

5.3 Learning hierarchical features

A distinguishing property of deep learning models is ability to learn concepts in a hierarchical way, from simpler to more complex ones. As one can expect, properly trained network for stenosis detection should also decompose the concept of stenosis into a hierarchy of simpler concepts, represented by filters. We have performed analysis of convolutional layers in network F to determine that this phenomenon occurred in our case.

Neural networks are often regarded to as “black boxes”, due to difficulties in explaining reasoning performed by the network. However, in case of CNN, we are able to visualize network features (both low- and high-level), using *activation maximization* method introduced by Erhan & others [16]. As the name suggests, this method finds input pattern that maximizes activation of hidden unit. Since this is an optimization problem, we can find maximum using gradient ascent:

$$(2) \quad x^* = \arg \max_{x \text{ s.t. } \|x\|=\rho} h_{ij}(\theta, x)$$

where x is input pattern, θ are network weights and h_{ij} is activation between layers i and j .

Figure 5 presents patterns generated after 20 000 iterations of gradient ascent with step 0.01 for each filter in every convolutional layer of trained and fine-tuned network F.

Fig. 5. Visualisation of convolutional filters in network F

Although generated visualizations are not as easily interpretable as those for networks for character recognition [16], we are still able to have some insight in network inner workings. As one can see, activation patterns were found only for some filters, producing random noise in other cases. This can serve as indicator that network structure could be simplified without performance loss. Nonetheless, some filters, even in deeper layers, have distinct activation patterns. Note that many filters resemble Gabor filters used for edge detection. Moreover, some filters in layers 1 and 2 work in a way similar to Sobel operator, also used for edge detection. An interesting property is that filters in lower layers are more global, while in higher layers filter are more locally oriented. It suggests that network was indeed able to learn what “stenosis” is, using lower-level features and combining it in to higher level ones.

6 Conclusions

6.1 General summary

We have demonstrated that it is possible to create effective deep convolutional neural network for specific application in medical diagnosis, namely, stenosis detection. Such network has better performance than traditional models, like FFN or shallow CNNs. Analysis of filters suggests that this is achieved thanks to ability of deep network to decompose the problem of stenosis classification into hierarchical features.

Pre-training with artificial data improves performance and accuracy of resulting model. Moreover, generating artificial data partially alleviates problems with limited amount of training data and disparity between numbers of samples in classification classes. These are common problems with medical image processing where obtaining training data involves sensitive patient data and real-world examinations. Positive class samples are even more scarce as patients with, often rarely occurring diseases, are required. Noted facts imply that data-generating approach may prove useful in the field of medical image processing.

6.2 Application in other fields

Data set prepared for training a model should be suitably numerous for all kinds of neural networks, be it convolutional or recurrent. Considering this fact, sample

generation method can prove useful for different types of models in all fields where obtaining data set is hard, time consuming or expensive.

References

1. I. Goodfellow, Y. Bengio and A. Courville, Deep Learning, MIT Press, (2016)
2. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg and L. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge, IJCV, (2015)
3. L. June-Goo, J. Sanghoon, C. Young-Won, L. Hyunna, K. Guk Bae, S. Joon Beom and K. Namkug , Deep Learning in Medical Imaging: General Overview, Korean Journal of Radiology, (2017)
4. L. Geert, K. Thijs, B. Babak Ehteshami, A. S. Arnaud Arindra, F. Ciompi, M. Ghafoorian, J. van der Laak, B. Van Ginneken and C. Sanchez, A Survey on Deep Learning in Medical Image Analysis, Nijmegen: Diagnostic Image Analysis Group, (2017)
5. H. Marcovitch, Ed., "Stenosis," in *Black's Medical Dictionary: 41st Edition*, Londow, A & C Black Publishers Limited, (2005)
6. V. Gorenoi, M. P. Schönermark and A. Hagen, "CT coronary angiography vs. invasive coronary angiography in CHD," *GMS health technology assessment*, no. 8, (2012)
7. D. S. Baim, "Coronary angiography," in *Grossman's Cardiac Catheterization, Angiography, and Intervention, 7th edn*, D. S. Baim, Ed., Philadelphia, Lippincott Williams & Wilkins, (2005)
8. E. Arnoldi, M. Gebregziabher, U. J. Schoepf, R. Goldenberg, L. Ramos-Duran, P. L. Zwerner, K. Nikolaou, M. F. Reiser, P. Costello and C. Thilo, "Automated computer-aided stenosis detection," *European radiology*, no. 20.5, (2010)
9. D. Ciregan, U. Meier and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, Providence, (2012)
10. S. Lawrence, C. L. Giles, A. C. Tsoi and A. D. Back, "Face Recognition: A Convolutional Neural Network Approach," *Neural Networks, IEEE Transactions on*, **8**, 1 (1997)
11. J. Wang and L. Pere, "The effectiveness of data augmentation in image classification using deep learning," arXiv, (2017)
12. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. W.-F. D. Xu, S. Ozair and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, (2014)
13. P. Y. Simard, D. Steinkraus and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," *ICDAR*, **3** (2003)
14. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv*, **1409**, 1556 (2014)
15. A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, (2012)
16. D. Erhan, Y. Bengio, A. Courville and P. Vincent, "Visualizing Higher-Layer Features of a Deep Network," Dept. IRO, Université de Montréal, Montreal, (2009)