Semantic-Based Brain MRI Image Segmentation Using Convolutional Neural Network

Yao Chou[1], Dah Jye Lee[1(✉)], and Dong Zhang[2]

[1] Electrical and Computer Engineering,
Brigham Young University, Provo, UT, USA
djlee@ee.byu.edu
[2] School of Electronics and Information Technology, Sun Yat-sen University,
Guangzhou, Guangdong, China

Abstract. Segmenting Magnetic Resonance images plays a critical role in radiotherapy, surgical planning and image-guided interventions. Traditional differential filter-based segmentation algorithms are predefined independently of image features and require extensive post processing. Convolutional Neural Networks (CNNs) are regarded as a powerful visual model that yields hierarchies of features learned from image data, however, its usage is limited in medical imaging field as it requires large-scale data for training. In this paper, we propose a simple binary detection algorithm to bridge CNNs and medical imaging for accurate medical image segmentation. It applies high-capacity CNNs to extract features from image data. When labeled training medical images are scarce, the proposed algorithm splits data into small regions, and labels them to boost training data size automatically. Rather than replaces classic segmentation methods, this paper presents an alternative that is unique and provides more desirable segmentation results.. . .

1 Introduction

In computer vision, the goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries in image [1]. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain common characteristics [2]. Many applications require image segmentation, such as content-based image retrieval, machine vision, medical imaging [3], object detection [4], recognition tasks [5], control systems, and video surveillance.

Computer-aided image analysis systems can enhance the diagnostic capabilities of physicians and reduce the time required for accurate diagnosis [6]. As one of the major techniques, medical image segmentation plays a significant role in clinical diagnosis. It is considered challenging because medical images often have low contrast, various types of noise, and missing or diffused boundaries [7]. Research efforts have been devoted to processing and analyzing medical images to segment meaningful information such as volume, shape, and motion

G. Bebis et al. (Eds.): ISVC 2016, Part I, LNCS 10072, pp. 628–638, 2016.
DOI: 10.1007/978-3-319-50835-1_56

of organs, to detect abnormalities, and to quantify changes in follow-up studies [8]. Many image segmentation techniques are available in the literature. Some use only the gray level histogram [1] or spatial details and others use fuzzy set theoretic approaches. Most of these techniques are sensitive to noise, and thus not suitable for medical imaging. The Markov Random Field model is robust to noise, but involves a huge amount of computations [9]. Manual segmentation is an expensive, time consuming task. It is subject to manual variation and subjective judgments, which increases the possibility that different observers will reach different conclusions about the presence or absence of tumors. Even the same observer will occasionally reach different conclusions on different occasions [10]. An efficient and consistent medical image segmentation algorithm would help avoid these confusions.

Deep learning algorithms have shown remarkable results in various image processing fields for most benchmark image datasets including MNIST (classify handwritten digits) [11], CIFAR-10 (classify 32×32 color images for 10 categories) [12], CIFAR-100 (classify 32×32 color images for 100 categories) [13], STL-10 (similar to CIFAT-10 but with 96×96 images)[14], and SVHN (the street view house numbers dataset)[15], etc. Convolutional Neural Networks (CNNs), as a milestone model of deep learning, are driving advances in image analysis. CNNs not only improve the performance of whole-image classification, but also make progress on extracting features. CNNs make a prediction for every pixel and are able to take the advantage of the detailed features of an object image. Krizhevsky et al. made a significant improvement in image classification accuracy on the ImageNet large-scale visual recognition challenge (2012) [16]. Different from traditional image processing methods (e.g. SIFT [17], HOG [18], etc.), which involve a hand-crafted feature descriptor, CNNs are deep architectures for learning features. All the features are learned hierarchically from pixels to classifier, and each layer extracts features from the output of previous layers [19]. However, to obtain superior performance, CNNs usually require a large-scale training process. To collect an abundance of medical images is costly and not feasible. The training process also consumes too much time and resources to provide manually annotated training datasets.

In this paper, we propose a brand new concept on how to use CNNs for brain image segmentation with implicit features that link medical imaging to deep learning. We divide training images into regions and label them automatically to boost the size of the training dataset. A CNN learning framework is designed to capture the local structure of the ROIs and automatically learn the most relevant features.

After a brief introduction to the background, the problem formulation along with the data generation is provided in Sect. 2. In Sect. 3, we present the details of a CNN architecture. Section 4 shows the results and includes discussion. Finally, the paper is summarized and concluded with future research directions in Sect. 5.

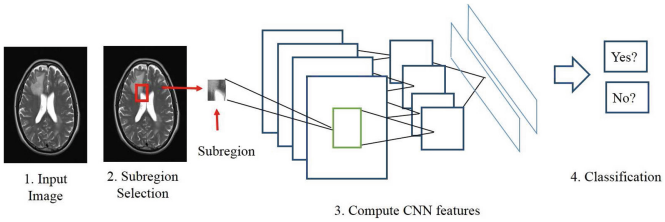

Fig. 1. Segmentation system overview. (1) Brain MRI input image, (2) region extraction, (3) feature computation for each region using a convolutional neural network, and (4) region classification to detect ROI pixels.

2 Region-Based Segmentation

Image segmentation is a process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics. Therefore, assigning pixel labels using CNNs based on the features obtained from the image data is a reasonable strategy for segmentation. The main highlight of a deep learning algorithm is that all features are learned from the image data directly. The neural network architecture has more than 60 million parameters, which makes training on GPUs a necessity. A straightforward way to improve the performance of CNNs is by increasing the size of training data. Acquiring such data is not always feasible for medical imaging. In order to take the advantage of CNNs to obtain accurate segmentation, we propose a method that can solve the limited training data problem from which CNNs generally suffer. Figure 1 presents an overview of our method. After boosting the size of training data, we perform the stochastic gradient descent (SGD) training of CNN parameters using this large dataset. The result is a customizable segmentation operation whose performance and behavior reflect the segmentation criteria learned directly from the training data. The proposed method is composed of three main steps:

1. Generate enough training data from the limited original data
2. Label data efficiently
3. Augment the dataset

2.1 Generate Training Data

In dealing with Magnetic Resonance Imaging (MRI) images, one of the most challenging aspect is the process of partitioning some specific cells and tissues from the rest of the image. An MRI image from our dataset is shown in Fig. 2(a). Experienced doctors segment out the tumor area (white area in the lower half) in the image manually to get the binary ground-truth segmentation as shown in Fig. 2(b) (zoomed in for clarity). In Fig. 2(b), pixels in the tumor area are set to black and pixels in the background are set to white. In order to provide enough annotated training images for CNN, a sliding window of $n \times m$ pixels is applied

to extract small regions' proposals. These patches from one image sample are used for training.

2.2 Label Training Data

To label the patches obtained with a sliding window, the image regions tightly enclosing ROI pixels are regarded as positive examples (e.g. window c in Fig. 2(b), while the non-ROI regions, which have nothing to do with the tumor area, are treated as negative examples (window d). Regions that partially overlap the ROI are treated with a central area overlapping process. The ground-truth segmentation is also regarded as positive example if it goes through the central area of a region's proposals. Otherwise, the region is considered as negative. Figures 2(c–f) show the zoomed in areas in four boxes in Fig. 2(b).

2.3 Augment Positive Data

After data generation and labeling, we convert one image into a large dataset which includes around 300 positive regions and 20,000 negative regions. Because the size of the positive training set is much smaller than the negative training set, data augmentation is necessary to improve classification performance. According to the characteristics of our data, we choose 2 ways flips (horizontal and vertical) and 35 rotations (10°, 20°, 30°, ..., 340°, 350°). Here, we rotate the whole original image and ground truth image by different angles, then use the same algorithm mentioned above to obtain the positive patches. After this step, we increase the positive training examples from 300 to 11,400.

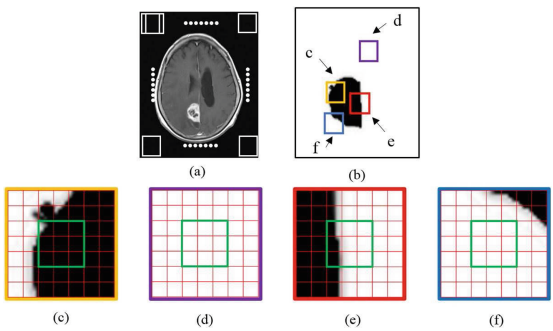

Fig. 2. An overview of data generation. (a) The original brain MRI image, (b) zoomed in segmentation labeled by doctors, (c) positive sample which tightly encloses the ROI pixels, (d) negative sample which is background pixels (e) positive sample which ground-truth segmentation line (the boundary) falls within the central area, (f) negative sample which the boundary does not pass through the central box.

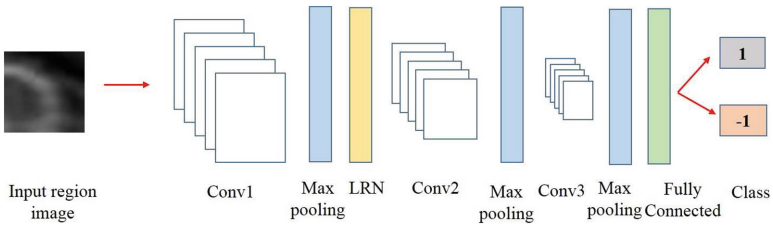

Fig. 3. Illustration of Convolutional Neural Network (CNN) architecture

3 CNN Architecture and Model Learning

The architecture of the CNN used in this paper is illustrated in Fig. 3. This CNN has three convolution-max pooling layers followed by a 2-way softmax output layer. The CNN is configured with Rectified Linear Units (ReLUs), as they train several times faster than their equivalents with tanh connections. This section articulates details of those layers. 21×21 patches were used as data for the CNN in this study. Patches are all gray images, and 1 channel is used for the input data. The first convolutional layer uses 96 kernels of size 5×5 with a stride of 4 pixels and padding of 2 pixels on the edges, followed by a 3×3 max pooling layer with a stride of 2. A Local Response Normalization (LRN) layer is applied after the first pooling layer. The second convolutional layer uses 128 filters of size 3×3 with a stride of 2 pixels and padding of 2 pixels on the edges. A second pooling layer has the same specification as the first one. The third convolutional layer uses 128 filters of size 3×3 with stride and padding of 1. The third pooling layer also has the same configuration as the two before it and leads to a softmax output layer with two labels corresponding to ROI pixel (1) and non-ROI pixel (-1) classes.

4 Experiments and Discussions

The algorithm learned the segmentation model after all regions were trained using our CNN. We tested our segmentation algorithm on different brain MRI slices. Our goal was to output a same size binary segmentation image similar to the ground-truth image the doctors segmented manually. For the test images, we applied a sliding window of the same size 21×21 to obtain the region proposals, then forward propagated the proposal through the CNN model in order to determine the class to be positive or negative. We recorded the location of each region's central pixel in the original image for constructing the binary segmentation. If the region was classified as positive, which means the center of this region is considered as a ROI pixel, the central pixel was set to 0 (black) in the segmentation output image. Otherwise, the central pixel was considered a non-ROI pixel or background and was set to 1 (white).

In clinical MRI applications, transverse plane, coronal plane, and sagittal plane are three main planes of the body used to describe the location of body

parts in relation to one another. The transverse plane is a horizontal plane that divides the body into superior and inferior parts, the coronal plane is any vertical plane that divides the body into ventral and dorsal sections, and the sagittal plane is any vertical plane which divides the body into right and left halves. Scans of different plane vary significantly. We trained different models for different planes in Sect. 4.1. We also experimented with creating one general model to detect tumors in images from all three scans. Since our model is based on deep learning, this challenge is easily addressed by extending the training data set to cover all three cases, The results are shown in Sect. 4.2.

In general, a primary brain tumor has only one large lesion. It is usually associated with extensive local edema and is easy to be detected. Whereas, a secondary brain tumor usually has several very small lesions without local edema and is hard to be detected. We chose images with secondary brain tumors as our test samples to demonstrate the superiority of our method. All images chosen for study had small brain tumors, and they were not all visible in all slices. Because of the limitation of the medical image resource, in our experiments, we used the MRI images from 5 patients (A–E). The patients' information is listed in Table 1. We picked the slices in which the tumor can be seen and labeled by experienced doctors.

Table 1. The information of patients

Number	Gender	Age	Occupy	Diagnosis
A	F	52	Farmer	Brain metastases of breast carcinoma
B	F	74	Worker	Brain metastases of breast carcinoma
C	F	46	Farmer	Brain metastases of lung carcinoma
D	F	28	Farmer	Brain metastases of breast carcinoma
E	F	57	Farmer	Brain metastases of lung carcinoma

In each SGD iteration of our training, we uniformly sample 32 positive regions and 32 negative examples to construct a minibatch of size 64. We biased the sampling towards positive regions, because they are extremely rare compared to the background or negative regions. The CNN portion of our experiments used Caffe framework [20] running on the NVIDIA Kepler series K40 GPUs. We used Matlab to produce the final segmentation results. The CNN model presented in Fig. 3 was trained using region images several times to increase its ability to automatically detect ROI pixels in any test image with a variant resolution.

4.1 Three Plane Models

We performed two experiments for each plane. In the first experiment (Test 1, 3 and 5), we used different slices from the same patient for training and testing. For the second experiment (Test 2, 4 and 6), we picked slices from multiple

patients (excluded testing patient) for training. Table 2 shows the detail of the experiments' setup. Experiment results are shown in Figs. 4 and 5. All training images are listed in the Appendix A. To evaluate the performance, we compared our method with Otsu' method [21]. Test 1 shown in Fig. 4(a–d) used one slice for training. The algorithm was able to detect the tumor areas accurately although with some noise. Segmentation result for Test 2 shown in Fig. 4(g) is almost identical to the doctors' labeling. Result of Test 2 was better than Test 1 mostly because more slices were used for training. Results of Test 1 to 2 show the algorithm was able to effectively and accurately locate the tumor.

Table 2. The details of experiments' setup

	Transverse		Coronal		Sagittal	
	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6
Training	A	A C D	B	C E	A	B C D
Testing	A	B	B	B	A	A

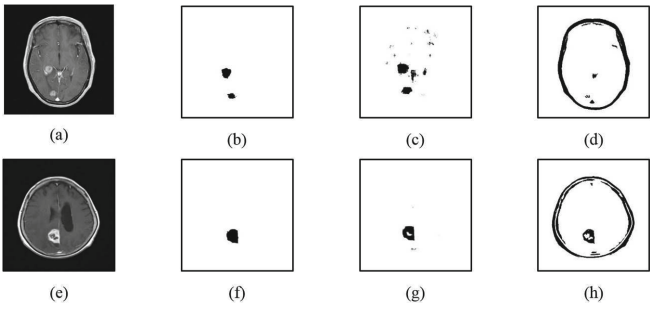

Fig. 4. Transverse plane segmentation. Test 1: (a) One transverse slice from patient A as the testing image, (b) ground-truth of (a), (c) our segmentation, (d) Otsu's segmentation. Test 2: (e) One transverse slice from patient B as the testing image, (g) ground-truth of (e), (g) our segmentation, (h) Otsu's segmentation.

Compared with Otsu's method, our method was able to distinguish boundary pixels of the skull from tumor pixels. However, Otsu's method failed to differential tumor area from the skull boundary. So we had to apply morphological post-processing to remove the boundary of the results for comparison ('Otsu-p') as shown in Table 3. As mentioned before, since we chose 3×3 central window, this would dilate the final result. For a fair comparison, we also applied a simple erosion method to our raw result, where 'Ours-1' means the erosion operation was applied once, and 'Ours-2' means the erosion operation was applied twice. The comparison performance is presented in Table 3.

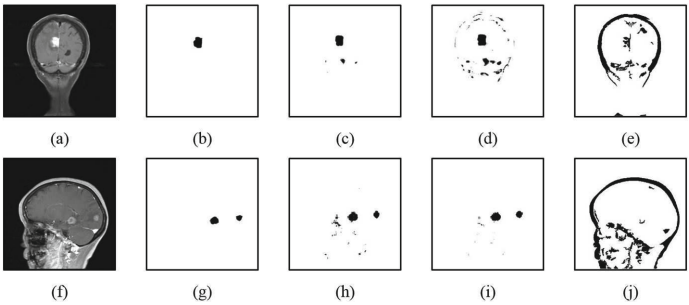

Fig. 5. Coronal and sagittal plane segmentations. (a) Testing image in Test 3 and 4, (b) segmentation of (a) labeled by doctors, (c) our segmentation in Test 3, (d) our segmentation in test 4, (e) Otsu's segmentation of (a), (f) testing image in Test 5 and 6, (g) segmentation of (f) labeled by doctors, (h) our segmentation in Test 5, (i) our segmentation in Test 6, (j) Otsu's segmentation of (f).

Test 3 and 4 used the same image Fig. 5(a) to test, and both can locate the tumor with high recall scores listed in Table 3. However, for Test 4, there are only two patients whose tumors could be seen in coronal plane scan. Since the training data were scarce and much different from the test image, the result of Test 4 presented in Fig. 5(d) showed some contour noise which can be removed by simple post processing techniques, e.g. 'Ours-2' boosted the precision score to 0.64 from 0.29. Tests 5 and 6 show our method has a strong response for the two tumor areas, which outperformed Otsu's method. Test 6 has better recall and precision scores than Test 5, since Test 6 took use of more training images. Better accuracy could be obtained if more training data were available.

Table 3. Comparison results in Test 1–6

Methods			Ours	Otsu'	Otsu'-p	Ours-1	Ours-2
Transvers	Test 1	Recall	**0.9640**	0.0661	0.1178	0.8921	0.7587
		Precision	0.3929	0.0070	0.0661	0.6060	**0.7140**
	Test 2	Recall	0.8900	0.7613	**0.9824**	0.7945	0.6621
		Precision	0.8429	0.1053	0.2945	**0.9131**	0.9114
Coronal	Test 3	Recall	**0.8845**	0.5961	0.5961	0.8362	0.7509
		Precision	0.6878	0.0530	0.1650	0.7957	**0.8364**
	Test 4	Recall	**0.9184**	0.5961	0.5961	0.8785	0.8011
		Precision	0.2940	0.0530	0.1650	0.5005	**0.6423**
Sagittal	Test 5	Recall	**0.9856**	0.1150	0.1150	0.8802	0.6166
		Precision	0.4901	0.0060	0.0152	0.7905	**0.8355**
	Test 6	Recall	**0.9984**	0.1150	0.1150	0.9217	0.7252
		Precision	0.6250	0.0060	0.0152	0.8266	**0.8566**

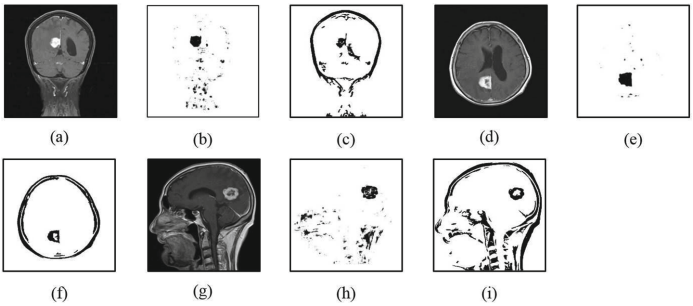

Fig. 6. Test 7 results, (a–c) transvers plane, (d–f) coronal plane, (g–i) sagittal plane.

Table 4. Comparison results in Test 7.

Methods		Ours	Otsu'	Otsu'-p	Ours-1	Ours-2
Transvers	Recall	**0.9910**	0.7993	0.7993	0.9595	0.8815
	Precision	0.2851	0.0643	0.1764	0.5664	**0.7618**
Coronal	Recall	**0.9955**	0.7365	0.7365	0.9707	0.8734
	Precision	0.7056	0.0978	0.2288	0.8613	**0.8918**
Sagittal	Recall	**0.7922**	0.6442	0.6442	0.5136	0.3073
	Precision	0.7056	0.0978	0.2288	0.8613	**0.8918**

As shown in the Table 3, the proposed segmentation algorithm has the best recall score in every Test except Test 2. Otsu's method with post processing performed better in this case. However, its precision is pretty low. For Precision, our method with simple post processing performed the best.

From the perspective of running speed, one pass for our CNN model takes close to 1 ms. Each pass can be done individually to take the advantage of parallel processing. Whereas, the Otsu' method takes one whole second and its computation cannot be parallelized. Our method has great potential to be implemented in hardware for real time segmentation.

4.2 General Model

We selected one slice from each of three scans from patient B as the test image for Test 7. The general model was trained using one slice of each scan that tumor areas were visible from all other patients. The results for this experiment are listed in Fig. 6. We also compared our results with Otsu' method shown in Table 4. We observed noisier segmentation using a general model than using a specialized model, but the general model was able to segment tumor on all three planes. Table 4 shows our methods have the best recall and precision scores.

5 Conclusion

In this paper, we propose to use a Convolutional Neural Network for brain MRI image segmentation. We train a CNN with ROIs and non-ROIs patterns iteratively so that it is able to automatically segment tumor areas effectively. Our result is very promising. Since all features are learned from labeled data, our model is able to accurately locate the tumor areas.

Our motivation for this work is not to replace any existing well-known segmentation methods. This work proposes an interesting concept that could be improved further. Since all important information for segmentation is learned from the data labeled by the experts, our model has demonstrated its capability of mimicking the expert's segmentation style represented in the ground truth. Other segmentation methods require fine tuning the parameters manually for different applications. An advantage of the proposed method is its flexibility and potential to adapt for different applications or imaging modalities without any modifications of the algorithm. Unlike the traditional deep learning methods that require a large scale of training data, which is often not feasible for medical image applications, this algorithm requires only a small set of training images and the ground truth.

The proposed method trains the CNN model with only a couple of images. Training with more images will further improve its performance. Because our dataset size is small, starting with a pre-trained model would also improve its performance. Meanwhile, different experimental settings might change the performance, which needs to be investigated in the future.

Appendix: A

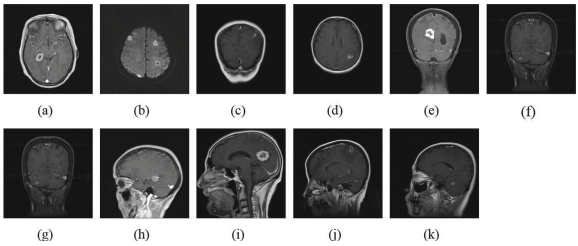

Fig. 7. Training images. (a) Training image in Test 1 and 2, (b–d) in Test 2, (e) in Test 3, (f–g) in Test 4, (h) training image in Test 5 and 6, (i–k) in Test 6.

References

1. Shapiro, L.G., Stockman, G.C.: Computer Vision. Prentice Hall, Upper Saddle River (2001)

2. Barghout, L., Lee, L.: Perceptual information processing system. US Patent App. 10/618,543 (2003)
3. Pham, D.L., Xu, C., Prince, J.L.: Current methods in medical image segmentation 1. Ann. Rev. Biomed. Eng. **2**, 315–337 (2000)
4. Gould, S., Gao, T., Koller, D.: Region-based segmentation and object detection. In: Advances in Neural Information Processing Systems, pp. 655–663 (2009)
5. Zhou, Y., Wang, W., Huang, X.: FPGA design for PCANet deep learning network. In: 2015 IEEE 23rd Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), p. 232. IEEE (2015)
6. El-Dahshan, E.S.A., Mohsen, H.M., Revett, K., Salem, A.B.M.: Computer-aided diagnosis of human brain tumor through MRI: a survey and a new algorithm. Expert Syst. Appl. **41**, 5526–5545 (2014)
7. Sharma, N., Aggarwal, L.M., et al.: Automated medical image segmentation techniques. J. Med. Phys. **35**, 3 (2010)
8. Huang, X., Tsechpenakis, G.: Medical image segmentation
9. Despotović, I., Goossens, B., Philips, W.: MRI segmentation of the human brain: challenges, methods, and applications. In: Computational and Mathematical Methods in Medicine 2015 (2015)
10. Nabizadeh, N., Kubat, M.: Brain tumors detection and segmentation in MR images: Gabor wavelet vs. statistical features. Comput. Electr. Eng. **45**, 286–301 (2015)
11. Wan, L., Zeiler, M., Zhang, S., Cun, Y.L., Fergus, R.: Regularization of neural networks using dropconnect. In: Proceedings of the 30th International Conference on Machine Learning (ICML-13), pp. 1058–1066 (2013)
12. Graham, B.: Fractional max-pooling (2014). arXiv preprint arXiv:1412.6071
13. Clevert, D.A., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (ELUs) (2015). arXiv preprint arXiv:1511.07289
14. Zhao, J., Mathieu, M., Goroshin, R., Lecun, Y.: Stacked what-where auto-encoders (2015). arXiv preprint arXiv:1506.02351
15. Lee, C.Y., Gallagher, P.W., Tu, Z.: Generalizing pooling functions in convolutional neural networks: mixed, gated, and tree (2015). arXiv preprint arXiv:1509.08985
16. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
17. Lowe, D.G.: Object recognition from local scale-invariant features. In: The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999, vol. 2, pp. 1150–1157. IEEE (1999)
18. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, CVPR 2005, vol. 1, pp. 886–893. IEEE (2005)
19. Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S.: CNN features off-the-shelf: an astounding baseline for recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 806–813 (2014)
20. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the ACM International Conference on Multimedia, pp. 675–678. ACM (2014)
21. Zhu, N., Wang, G., Yang, G., Dai, W.: A fast 2d otsu thresholding algorithm based on improved histogram. In: Chinese Conference on Pattern Recognition, 2009, CCPR 2009, pp. 1–5. IEEE (2009)