INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Semantic Edge Detection for Tracking Vocal Tract Air-tissue Boundaries in Real-time Magnetic Resonance Images

Krishna Somandepalli, Asterios Toutios, Shrikanth S Narayanan

Signal Analysis and Interpretation Laboratory, University of Southern California, Los Angeles, USA

somandep@usc.edu, toutios@usc.edu, shri@sipi.usc.edu

Abstract

Recent developments in real-time magnetic resonance imaging (rtMRI) have enabled the study of vocal tract dynamics during production of running speech at high frame rates (e.g., 83 frames per second). Such large amounts of acquired data require scalable automated methods to identify different articulators (e.g., tongue, velum) for further analysis. In this paper, we propose a convolutional neural network with an encoder-decoder architecture to jointly detect the relevant air-tissue boundaries as well as to label them, which we refer to as 'semantic edge detection'. We pose this as a pixel labeling problem, with the outline contour of each articulator of interest as positive class and the remaining tissue and airway as negative classes. We introduce a loss function modified with additional penalty for misclassification at air-tissue boundaries to account for class imbalance and improve edge localization. We then use a greedy search algorithm to draw contours from the probability maps of the positive classes predicted by the network. The articulator contours obtained by our method are comparable to the true labels generated by iteratively fitting a manually created subject-specific template. Our results generalize well across subjects and different vocal tract postures, demonstrating a significant improvement over the structured regression baseline.

Index Terms: Real-time magnetic resonance imaging (MRI), vocal-tract dynamics, contour drawing, convolutional neural networks (CNN)

1. Introduction

Recent imaging protocols [1] have been able to achieve high frame rates such as 83 frames per second (fps) and higher for capturing the dynamics of vocal tract shaping during speech. In this context, there is a need for scalable methods to automatically identify outlines (contours) of the different articulators involved in speech production for further analysis. Moreover, these methods have to generalize across subjects, accounting for variation in size and shape of the vocal tract and positioning of the person in the MRI scanner.

Over the years, different ways (e.g., [2, 3, 4, 5]) have been explored to semi-automatically identify contours corresponding to the articulators of interest. An edge detection method in the spatial frequency domain using an anatomical template was proposed in [3]. This method performs nearly as well as manual tracing of the contours. However, this system requires careful design of an initialization template for each subject. Additionally, the optimization steps are computationally expensive and require a few minutes per image frame using its MATLAB implementation [3].

The objective of our work is to obtain a non self-intersecting polyline contour description corresponding to the twelve articulators shown in Fig 1B. In this paper, we formulate this as a multi-class pixel labeling problem. Image processing techniques such as morphological operations do not yield reliable results due to low signal-to-noise ratio (SNR) in these images. However, there is a rich structure (spatial relation) among the articulators of interest. Exploiting this structure among the outputs, one can pose this as a structured regression problem. Methods such as dlib [6] have been successfully used for face landmark detection [7]. In contrast to face images, the MR images we use here (Fig 1A) have a low resolution and high noise and lack textural variation. As such, designing features can be challenging for MR images of the vocal tract. Convolutional neural networks are best suited for this problem since they can learn the spatial relationship between different regions in the image, and are being increasingly used with good performance for medical image analysis [8].

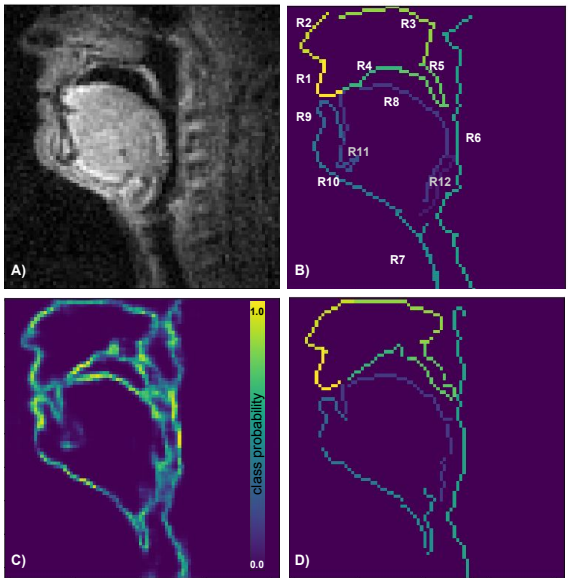

Figure 1: *A) Input midsagittal real-time MR image B) Vocal tract contour labels: R1-upper lip, R2-nose, R3-nasal cavity, R4-palate, R5-velum, R6-pharynx, R7-trachea, R8-tongue, R9-lower lip, R10-jaw, R11-incisor, R12-epiglottis C) Output class probability map D) Proposed system output contours*

VGG [9] is one of the most widely used CNN architectures for various image recognition tasks. In particular, many have implemented encoder-decoder architectures for pixel labeling tasks based on VGG. While the encoder networks in these are similar to VGG in topology, they differ in the decoder network, training and inference. For example, a fully convolutional network for semantic segmentation tasks with upsampling in the decoder network was proposed by [10] to predict pixel-level labels. SegNet [11] has additionally used novel decoder layers that upsample based on the corresponding indices from max-

http://dx.doi.org/10.21437/Interspeech.2017-1580

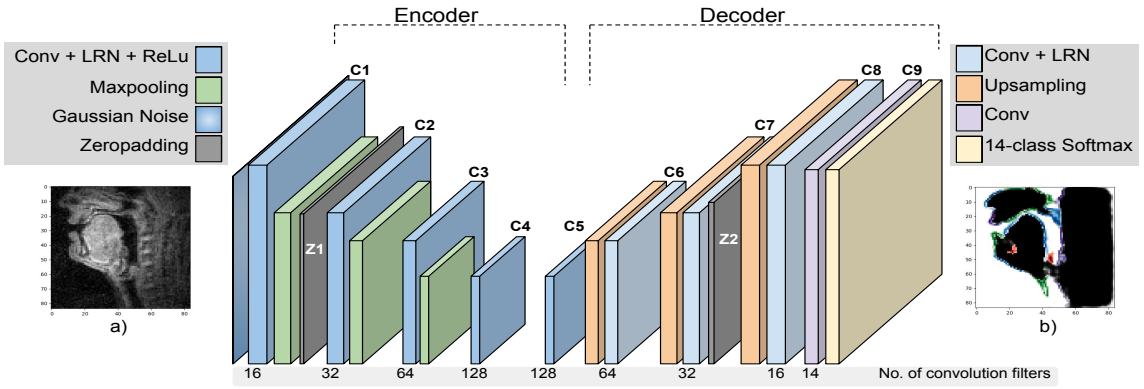

Figure 2: *Schematic diagram of the proposed encoder-decoder network architecture; Conv: convolution, LRN: local response normalization, ReLu: rectified linear unit; a) input b) output class probability map*

pooling layers of the encoder.

Along similar lines, fully convolutional networks have also been successfully used for edge detection (e.g., [12, 13, 14]). However, to the best of our knowledge there has been no work that can jointly detect edges and label them. We refer to this task as *semantic edge detection*. Building upon the ideas from the aforementioned papers, we propose a fully convolutional encoder-decoder network. Here, it is important to make the distinction between edges and contours. Generally, edges are computed as extrema points of the image gradient in the direction of the gradient. Ordered edge points constitute of a contour. In the case of open contours that lack parametrization (especially, deformable, non-convex contours) which is the case for this data, ordering points can be a very difficult problem. For this purpose, we explore a greedy search approach for drawing contours on the class probability maps obtained from the CNN.

2. Methods

2.1. Data and label generation

All the data and labels used in this study are publicly available (sail.usc.edu/span/test-retest). Dynamic rtMRI data were collected during production of running speech from eight healthy volunteers (4F/4M, age: 28.3 ± 4.3). The details of the experimental protocol used can be found in [15]. All imaging was performed on a GE Signa Excite 1.5T scanner with a custom eight-channel upper airway coil [1]. Sequence parameters were as follows: field-of-view (FOV) 200x200 mm, in-plane reconstruction, spatial resolution 2.4x2.4 mm, slice thickness 6 mm, TR 6 ms, TE 3.6 ms, flip angle 15°, and 13 spiral interleaves for full sampling. The scan plane was manually aligned with the mid-sagittal plane of the subject's head. Images were retrospectively reconstructed at 83.33 fps as described in [1] resulting in 2D images of size 84X84 (see Fig 1A).

We used the method proposed in [3] to generate *true labels* for training the CNN. The polyline points obtained from this method were linearly interpolated to generate pixel-level labels for twelve different articulators (*positive classes*, see Fig 1B). The remaining tissue and airway were labeled as *negative classes*. Airway was identified as the region of intersection of the entire image FOV and the convex hulls generated from the twelve articulators. It is important to note the performance of our system is somewhat limited by using the pixel-level labels thus generated as true labels since the shortcomings of the method as described in [3] are carried over. One in particular, is the lack of discriminability when two articulators are in contact.

2.2. Proposed CNN architecture

The proposed encoder-decoder model was inspired from some of the architectures discussed in Section 1. The important modifications we introduced are: 1) reducing the network size in order to accommodate for the low resolution of MR images and the number of parameters to be trained; 2) using local response normalization (LRN, [16]) for the activations; and 3) using a custom loss function penalizing for tissue-airway boundaries to address class imbalance per image and improve edge localization (see Section 2.3)

Our network was trained for a multiclass pixel labeling task, specifically twelve positive classes (articulator edges) and two negative classes (airway and tissue). Fig 2 depicts the the overall architecture of the proposed network along with the number of filters used in the convoultional layers, C1–C9. We use LRN instead of Batch Normalization (subtracting the mean activity) since LRN operates on local neighborhoods. This allows for detection of high-frequency features (which indicate edges in an image) with a big response, while damping others that are uniformly large in the neighborhood. In comparison to Batch Normalization, this resulted in higher output prediction probabilities for the pixel along the contour of interest. Initial testing showed about 8-12% improvement in the loss function on the training as well as validation data. The hyper-parameters of the LRN were set to the values proposed in [16]. The input to our network is a 84X84 grayscale MR image of the upper airway and the kernel size for convolution filters, C1–C9 was set to 3x3. Mmax-pooling was performed over a spatial pixel neighborhood of 3x3 with a stride of 2 pixels. While max-pooling ensures invariance to translation of the images, LRN effectively retains activation responses to image edges as described earlier.

Upsampling was performed with a factor of two for both image dimensions. Additionally, zero-padding layers (Z1, Z2 in Fig 2) were introduced in both the encoder and decoder to retain the dimensionality of the image. From our experiments, we observed that increasing the number of layers did not improve the overall performance but only increased the number of parameters to be learned. This is perhaps because most of the articulators are spatially localized and can be sufficiently represented by the kernel sizes used in the convolutional layers. Finally, the last convolution layer from the decoder was fed into a 14-class softmax layer. Due to the heavy imbalance between positive and the negative classes per image, we examined the probability maps corresponding to the articulators instead of the pixel-level class labels generated by the network to obtain the final contours.

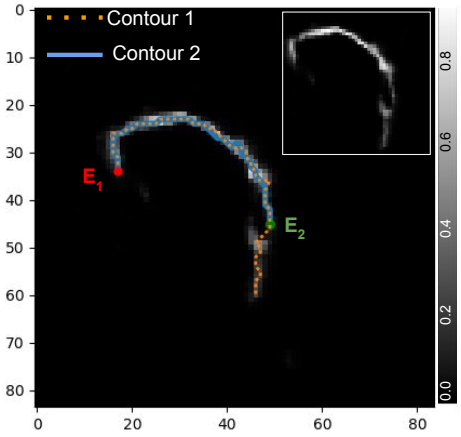

Figure 3: *Contour drawing example; Inset: probability map.*

2.3. Airway-tissue boundary penalized loss function

We used the cross-entropy loss function [10] to train our network which is summed up over all the pixels in the mini-batch. Since cross-entropy penalizes uniformly for classification error between all classes, we first address the class imbalance within each image. Approaches such as median frequency balancing [11] have been used in case of class imbalance across the entire training set. This method uses the ratio of the median of class frequencies across the entire training set to the class frequency to weigh the loss function. However, methods such as these do not address the image-level class imbalance as in our case.

In preliminary experiments, we observed that the confusion of labels between the positive classes and the airway was more pronounced than those between the positive classes or between the positive classes and tissue. This is likely due to low SNR in the MR images. To this end, we introduced an additional regularization term similar to that proposed in [14]. The motivation behind this is to penalize more for a mis-classification of one of the positive classes to airway. To compensate for the class imbalance, we kept this penalty asymmetric, i.e., no additional penalty was added for mis-classification of airway to one of the positive classes.

Let the image dimensions, width and height be w and h respectively. The total number of pixels to be labeled be n, where $n = w \times h$ where, $w = h = 84$. For each image in a mini-batch training set: $\{x^{(i)}, y^{(i)}\}_{i=1}^{n}$ where $x^{(i)}$ is the $i-$th image pixel. We denote the class labels as $y^{(i)}$ where $i \in \{-1, 0, 1, ...K\}$, with $K = 12$. If $y^{(i)} = \{-1, 0\}$, then the pixel, $x^{(i)}$ belongs to airway and tissue respectively. If $y^{(i)} > 0$, then $x^{(i)}$ belongs to one of the positive classes. Let $a_k^{(i)} : k \in [-1, K]$ be the output of the $k-$th unit of the softmax layer following layer C9 in the decoder for $x^{(i)}$. The probability that the class label, $y^{(i)} = k$ is given by

$$p_k^{(i)} = \frac{exp(a_k^{(i)})}{\sum_{j=-1}^{K} exp(a_j^{(i)})} \tag{1}$$

The standard cross-entropy loss that for each image, I in the training set is:

$$L_0(I) = -\sum_{i=1}^{n} \sum_{k=-1}^{K} 1(y^{(i)} = k) log p_k^{(i)} \tag{2}$$

where $1(.)$ is the indicator function and $L_0(I)$ is summed up across all images in the mini-batch. As discussed earlier, adding the regularization term to Eq. 2 gives the final loss function:

$$L(I) = L_0(I) - \Bigg[\sum_{i=1}^{n} \alpha \Big(1(y^{(i)} = -1) log p_{-1}^{(i)} + \sum_{k=1}^{K} 1(y^{(i)} = k) log(1 - p_{-1}^{(i)}) - \sum_{k=1}^{K} 1(y^{(i)} = -1) log(1 - p_k^{(i)}) \Big) \Bigg] \tag{3}$$

The last two summation terms in the Eq. 3 are separated to highlight the asymmetric nature of the penalty. Keeping only the first two summation terms in Eq. 3 would penalize uniformly for mis-classifications between positive classes and airway. The last term ensures that no additional penalty is added for incorrectly classifying airway as a positive class. Here, α is the parameter that controls the asymmetric penalty. When it is small, $L(I) \approx L_0(I)$ and increasing α would make Eq. 3 a loss function for a binary classification problem between airway and any of the positive classes, thereby increasing classification errors among the positive classes. In all our experiments, we set α to 1.

We used standard backpropagation on the modified loss function to optimize the parameters of the network. All our models were implemented in Keras (https://keras.io/) with a mini-batch size of 250 and the code is publicly available (https://github.com/krsna6/rtmri-segnet)

2.4. Contour drawing on output probability maps

Algorithm 1: Contour drawing on probability map

Input: Set of probabilities $P = \{p_{ij}\}$ and end points $E = \{e_{ij}\}$ with $|E| = l$
Output: Ordered contour points $O = \{o_{ij}\}$
$S = \{\}$
while $|E| > 0$ **do**
  $O^{(l)} = \{e_{ij}\}$
  Flag set, $F^{(l)} = \{\}$ to prevent loops
  $p_{ij} \leftarrow e_{ij}$
  while $\{p_{ij}\} \neq \emptyset$ **do**
    Select the next point with maximum probability
    $q_{ij} \leftarrow \{\arg\max_p N_8(p_{ij})\} \cap O^{(l)} \cap F^{(l)}$
    Add non-maxima neighbors to a flag set
    $F^{(l)} \leftarrow \{N_8(p_{ij}) \cap q_{ij}\} \cup F^{(l)}$
    Update the next point in the contour
    $O^{(l)} \leftarrow O^{(l)} \cup \{q_{ij}\}$
    $p_{ij} \leftarrow q_{ij}$
  end
  $E \leftarrow E \cap \{e_{ij}\}$
  $S \leftarrow S \cup \{\sum_{k=1}^{|O^{(l)}|} O_k^{(l)}\}$
end
Pick the contour that maximizes the sum of probabilities
$O = \arg\max_S O^{(l)}$

One approach to draw a contour on a probability map is finding the longest path [17] on a weighted directed acyclic graph (DAG) from the pixels. However, two important factors to design a DAG are absent in our output: 1) starting and ending points of the path along the contour; and 2) length of the path or order of points.

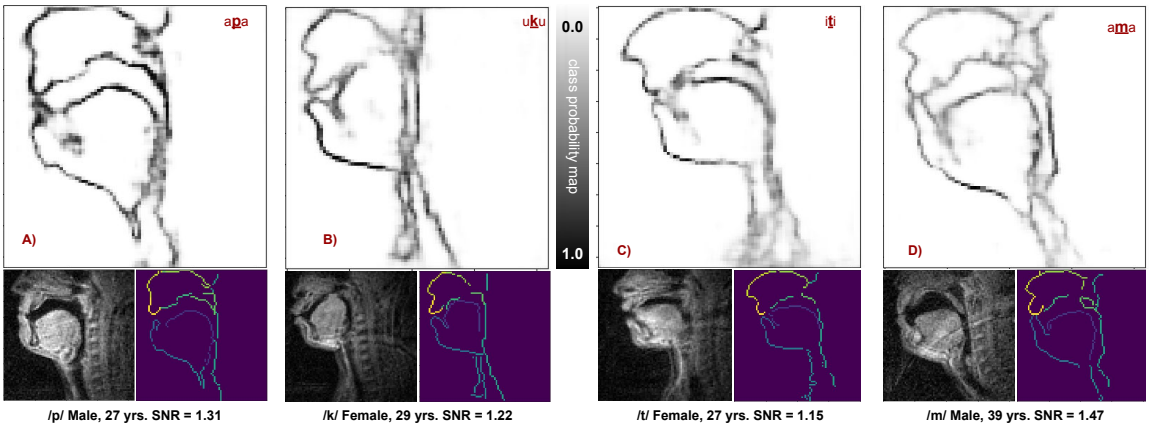

Figure 4: *Generalization of our proposed approach across subjects, articulatory gestures and image SNR*

We propose instead a greedy search algorithm for contour drawing inspired from connected component labeling [18]. Let $P^{'} = \{p_{ij}\}$ be the set of all pixels in the probability map where the coordinates $i, j \in [1, 84]$ and the set of non-zero probabilities, $P = \{p_{ij} \mid p_{ij} > \theta\}$. The threshold parameter, θ is set to the average of all probability values below 0.01. Denote the function to generate non-zero probabilities in the 8-connected neighborhood of a pixel p_{ij} as $N_8(p_{ij})$. We approximate the set of end points, E as the pixels which have exactly two neighbors, at least one of which is well connected, i.e., $N_8(p_{ij}) > 2$. With these end points as initial seeds, the points along the contour were determined as described in Algorithm 1. Fig 3 illustrates an example of contour drawing for the tongue region. *Contour 1* obtained from the end point, E_1 was chosen over *Contour 2* from E_2 to maximize the sum of probabilities along the path. Since we use pixel connectivity to determine the next point along the contour, the order is implicitly determined. It is important to note that the proposed algorithm is a heuristic and its theoretical guarantees have not been studied. Modifying the network to be able to incorporate a DAG framework for the probability maps would be part of our future work.

3. Experiments and Results

All experiments were performed in a leave-one-subject-out (LOSO) fashion for the eight subjects in our database, the reported measures are averaged across all LOSO models. The incisor and epiglottis (R11, R12 in Fig 1B) were excluded from further analysis since the *true labels* for these regions are not precise. A few examples illustrating the generalizability of our approach across subjects, articulatory gestures and image SNR are shown in Fig. 4. One of the drawbacks of our method is the lack of discriminability when articulators are in contact as in Fig. 4B. This is in part due to the ambiguity of true labels used for training generated from [3].

For each LOSO model, a total of 77000 images (11000 per subject), were randomly sampled to form the training data. The average training loss as per Eq. 2 (after 20 epochs, mini batch size=250) across the eight LOSO models for the proposed network was $8.9 \pm 0.2\%$ (c.f. $12.4 \pm 0.4\%$ from SegNet [11]). Additionally, the number of parameters to be learned in our network was about four times smaller than that of SegNet. The training loss for our network dropped to $7.1 \pm 0.1\%$ with the modified loss function as per Eq. 3. Although the gains in the accuracy with the modified loss function were minimal, we obtained sharper probability maps for the articulators.

As a baseline, we used a structured regression approach, dlib[6], to identify the points along the contours as landmarks. To compare the output contours from our approach and that of dlib with the true labels, we computed the average of the least Cityblock distance from every point of the output contour to the true labels and vice versa. The rectilinear nature of the Cityblock measure allows us to capture the relative smoothness between two contours. The smaller this measure, more similar is the output contour to the true labels. As shown in Table 1, our method outperforms dlib significantly (paired t-test $p << 0.01$ to reject $H_0 : \mu_0 \leq \mu_1$) for each articulator separately. Furthermore, dlib is sensitive to head size, position of the image and as such it did not generalize well across subjects. Our method performs best for articulators which have a distinct boundary with the airway (R1, R2, R9, R10), The performance is least for R5 (velum) due to the low contrast with adjacent tissue and higher noise in this region.

Table 1: *Average Cityblock distance to the true labels*

Label	dlib	Proposed	Label	dlib	Proposed
R1	3.01	1.52	**R6**	2.25	1.41
R2	2.42	1.91	**R7**	2.71	1.12
R3	2.78	1.42	**R8**	6.32	0.93
R4	2.56	1.70	**R9**	4.81	1.72
R5	4.25	1.92	**R10**	2.49	0.96

4. Conclusions and Future Work

In this paper, we propose a template-free approach for vocal tract articulator segmentation in real-time MR images using a CNN with an encoder-decoder architecture. We then use a greedy search algorithm to draw contours from the probability maps obtained from the network for each of the articulator. Our results demonstrate that the proposed method generalizes well across subjects as well as different articulatory gestures.

One of the drawbacks of our system is the lack of discriminability when two articulators are in contact (e.g., velum). This is in part due to lack of tissue-contrast in the MR images and imprecise training labels. Another potential shortcoming is that our method operates on a frame-by-frame basis, without taking into account the temporal structure of the rtMRI sequences. Our future work will focus on addressing these issues.

5. Acknowledgements

Work supported by NIH grant R01DC007124 and NSF grant 1514544

6. References

[1] S. G. Lingala, Y. Zhu, Y.-C. Kim, A. Toutios, S. S. Narayanan, and K. S. Nayak, "A fast and flexible MRI system for the study of dynamic vocal tract shaping," *Magnetic Resonance in Medicine*, Jan. 2016. [Online]. Available: onlinelibrary.wiley.com/doi/10.1002/mrm.26090/abstract

[2] E. Bresch, J. Adams, A. Pouzet, S. Lee, D. Byrd, and S. S. Narayanan, "Semi-automatic processing of real-time MR image sequences for speech production studies," in *Proceedings of the International Seminar on Speech Production (ISSP)*, Ubatuba, Brazil, Dec. 2006, pp. 427–434.

[3] E. Bresch and S. S. Narayanan, "Region segmentation in the frequency domain applied to upper airway real-time magnetic resonance images," *IEEE Transactions on Medical Imaging*, vol. 28, no. 3, pp. 323–338, Mar. 2009. [Online]. Available: sail.usc.edu/publications/Bresch-Narayanan-TMI2009.pdf

[4] M. I. Proctor, D. Bone, A. Katsamanis, and S. S. Narayanan, "Rapid semi-automatic segmentation of real-time magnetic resonance images for parametric vocal tract analysis," in *In Proceedings of InterSpeech*, Makuhari, Japan, Sep. 2010.

[5] J. Kim, N. Kumar, S. Lee, and S. Narayanan, "Enhanced airway-tissue boundary segmentation for real-time magnetic resonance imaging data," in *10-th International Seminar on Speech Production*, 2014, pp. 222–225.

[6] D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[7] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 1867–1874. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2014.241

[8] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *arXiv preprint arXiv:1702.05747*, 2017.

[9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[10] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 3431–3440.

[11] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for scene segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2017.

[12] J. Yang, B. Price, S. Cohen, H. Lee, and M.-H. Yang, "Object contour detection with a fully convolutional encoder-decoder network," 2016.

[13] G. Bertasius, J. Shi, and L. Torresani, "Deepedge: A multi-scale bifurcated deep network for top-down contour detection," *CoRR*, vol. abs/1412.1123, 2014. [Online]. Available: http://arxiv.org/abs/1412.1123

[14] W. Shen, X. Wang, Y. Wang, X. Bai, and Z. Zhang, "Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[15] J. Töger, T. Sorensen, K. Somandepalli, A. Toutios, S. Goud Lingala, S. Narayanan, and K. Nayak, *Journal of Acoustical Society of America*, in press 2017.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, p. 2012.

[17] A. Bjrklund, T. Husfeldt, and S. Khanna, "Approximating longest directed paths and cycles," in *In Proceedings of the 31st International Colloquium on Automata, Languages and Programming*. Springer, 2004, pp. 222–233.

[18] L. di Stefano and A. Bulgarelli, "A simple and efficient connected components labeling algorithm," in *Proceedings of the 10th International Conference on Image Analysis and Processing*, ser. ICIAP '99. Washington, DC, USA: IEEE Computer Society, 1999, pp. 322–. [Online]. Available: http://dl.acm.org/citation.cfm?id=839281.840794