Digital Pathology Annotation Data for Improved Deep Neural Network Classification

Edward Kim, SaiLakshmiDeepika Mente, Andrew Keenan, Vijay Gehlot

Department of Computing Sciences, Villanova University, Villanova, PA

ABSTRACT

In the field of digital pathology, there is an explosive amount of imaging data being generated. Thus, there is an ever growing need to create assistive or automatic methods to analyze collections of images for screening and classification. Machine learning, specifically deep learning algorithms, developed for digital pathology have the potential to assist in this way. Deep learning architectures have demonstrated great success over existing classification models but require massive amounts of labeled training data that either doesn't exist or are cost and time prohibitive to obtain. In this project, we present a framework for representing, collecting, validating, and utilizing cytopathology features for improved neural network classification.

Keywords: deep learning, cytology, thyroid, data augmentation, crowdsourcing, segmentation

1. INTRODUCTION

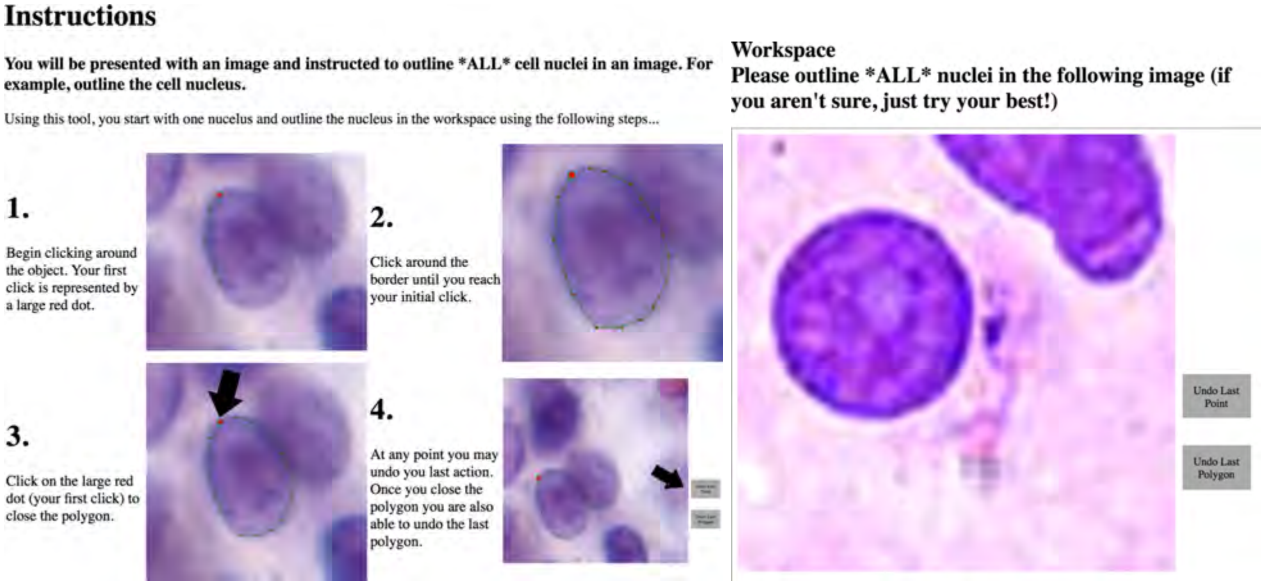
According to the National Cancer Institute's Surveillance, Epidemiology, and End Results Program (SEER),[1] there will be 64,300 new cases of thyroid cancer and 1,980 people will die of this disease in 2016. Additionally, thyroid cancer cases have been rising on average 4.5% each year over the last 10 years. Fortunately, it has been shown that classifying thyroid cancers and their variants through nuclear structure can be done,[2–5] and can be further improved with large amounts of reliable training data. However, obtaining expert annotation data at the cellular level (counting nuclei, measuring shape of cells, etc.) is one the principle challenges in the field of digital pathology where a single biopsy digitized at 40x resolution can contain 2.5-4 billion pixels of data. To put this into context, one digitized biopsy contains more than *800 cellphone pictures* captured at 5 megapixels, or *30 times* more information than an entire 3D body radiological CT scan.[6,7] This problem is multiplied by the tens of millions of biopsies performed every year in the United States. Due to the explosive amount of data generated, there is an ever growing need to create assistive or automatic methods to analyze collections of digital pathology "big data". Thus, the primary goal of our work is to improve computerized pathology diagnosis through the collection of large amounts of labeled data. This data can then be used to improve image classification using modern machine learning architectures.

2. BACKGROUND

We developed a prototype of an interactive tool that facilitates the collection and analysis of cytopathological features using online crowdsourcing. The tool's core mechanic involves the identification and delineation of certain cellular morphological features. An important goal of this tool was to be designed so that a user does not need to have any medical background knowledge in order to positively contribute to the analysis of anonymized cytological and histological images. To ensure the quality of the data, we employed several well defined quality assurance measures. Recent research has shown that crowdsourced annotations for small, basic pathology tasks by non-experts has good agreement with expert pathologist annotations.[8]

Given this annotation data, we utilize new big data machine learning methodologies that have emerged i.e. deep learning through convolutional neural networks (CNN). These CNNs have demonstrated great success over existing classification models[9–11] but require massive amounts of labeled training data that do not exist for digital pathology. Our data from the interactive platform is used to train data-intensive classification models of cytopathological characteristics.

For the task of image segmentation, there have been many deep learning approaches proposed in recent years. Many have utilized powerful models for region proposal,[12] or fully convolutional approaches pixel-wise segmentation.[13] Other work has employed sliding window based approaches to the segmentation task.[14] However,



Instructions

You will be presented with an image and instructed to outline *ALL* cell nuclei in an image. For example, outline the cell nucleus.

Using this tool, you start with one nucelus and outline the nucleus in the workspace using the following steps...

1. Begin clicking around the object. Your first click is represented by a large red dot.

2. Click around the border until you reach your initial click.

3. Click on the large red dot (your first click) to close the polygon.

4. At any point you may undo you last action. Once you close the polygon you are also able to undo the last polygon.

Workspace

Please outline *ALL* nuclei in the following image (if you aren't sure, just try your best!)

(a) MTurk instructions

(b) MTurk workspace interface

Figure 1. Examples from Amazon Mechanical Turk illustrating the (a) given instructions for the nucleus cell segmentation task. The image annotation tasks can be completed using a standard web browser. The workspace is shown in (b) where turkers can directly edit the SVG document model through javascript. Annotation results are stored as SVG polygon elements and ultimately stored in our relation database.

there are two immediate downsides to a sliding window approach. First, there is the added computational cost of running multiple image patches through the network to produce the segmented image. And secondly, restricting the network to viewing only patches of the image removes any relevant information which may be outside of the current image patch. For our image segmentation task, we modify and retrain the SegNet method.[15] SegNet is a deep fully convolutional neural network that consists of an encoder network and corresponding decoder network, followed by a pixel-level segmentation.

3. METHODOLOGY

For our methodology, we first define the structure of our image annotations, and then we describe how this structure and other quality assurance measures are created for the online crowdsourcing tool. Finally, we describe the machine learning architecture that can utilize our annotation data for a final pixel-level image classification algorithm.

3.1 Scalable Vector Graphics for Image Annotation

We use scalable vector graphics (SVG) to define the structure of the annotations.[16] SVG is an extensible and versatile language built using XML. Given the extensibility of the framework, we are able to encode low level image features and high level semantics. Using JQuery, we can enable interactions for image annotation. An annotator can begin clicking around an object (in our case a cell nucleus), and SVG circles will be created to represent their click coordinates. SVG line elements will connect the dots and when the user clicks on their first circle to close the loop, the circles and lines are replaced by an SVG polygon element and appended to the SVG document object model.

3.2 Online Crowdsourcing Annotations

The interactive data collection platform built around SVG and JQuery is accessible on standard web browsers. This enabled us to utilize online methods for soliciting annotations. We used Amazon Mechanical Turk, a crowdsourcing platform where small tasks can be distributed to a scalable online workforce e.g. turkers. The turkers are given thorough instructions on how to annotate the images. They are also shown diverse examples of

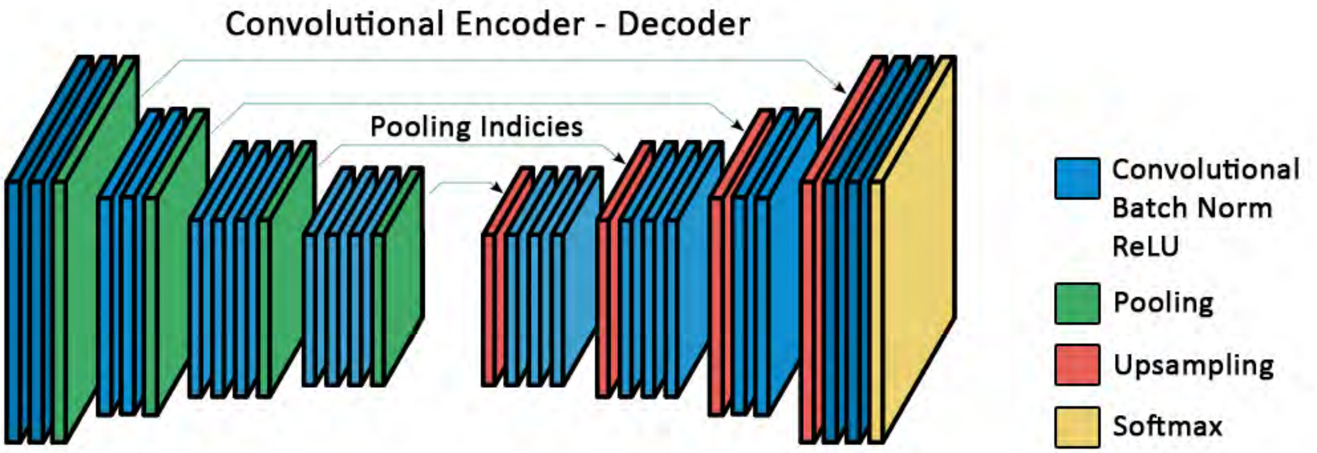

Convolutional Encoder - Decoder
Pooling Indicies
Convolutional Batch Norm ReLU
Pooling
Upsampling
Softmax

Figure 2. An illustration of the modified convolutional encoder-decoder architecture used for our segmentation results. The input to the network is a 128x128 image, and the output is a binary classification result of the same width and height. The convolutional layers are a subset of the VGG16[20] network.

what a cell nucleus looks like and then asked to the outline the cell nucleus in an SVG workspace. Functionality to undo and redo an user added SVG markup is also built into our platform. An illustration of this can be found in Figure 1.

A critical component of any crowdsourcing system is to perform quality assurance, which ensures that the workers understand the task, and perform the task as it was intended. Our system employs the following quality and evaluation metrics which can provide immediate feedback (online) or could be done in an offline fashion.

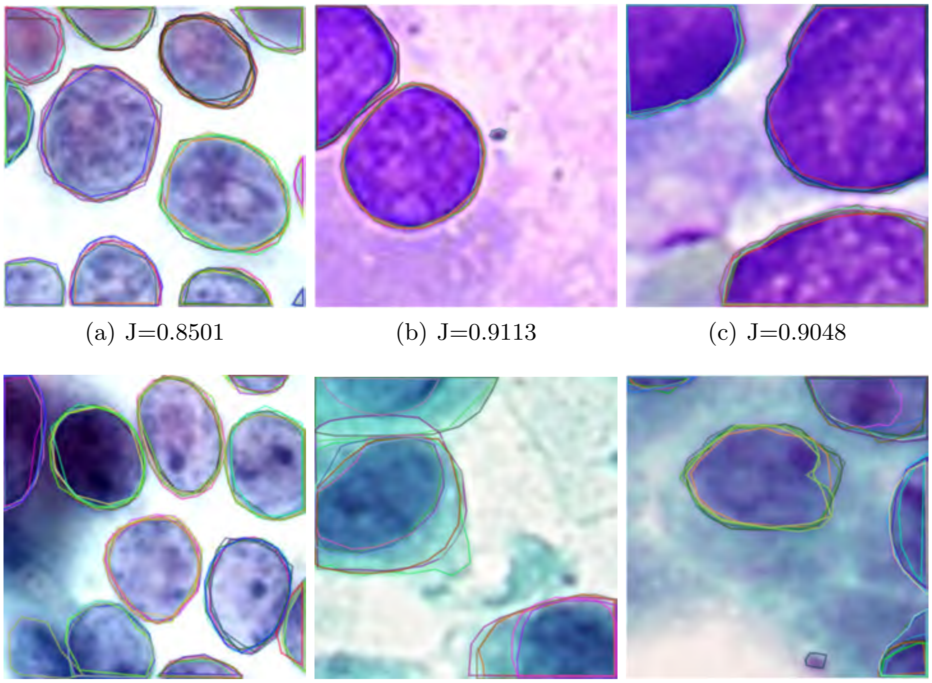
Quality Assurance 1 - Multiple Annotations - The primary offline strategy to ensure quality is task redundancy, where a researcher collects multiple annotations for each image. Agreement across multiple users can be used as a measure to define correctness or find outliers; however, one disadvantage to the researcher is the additional cost of task redundancy. For this project, we requested 7 redundant annotations from unique workers.

Quality Assurance 2 - Known Ground Truth - This strategy is referred to as the gold standard assurance measure,[17] where known images and their results are injected into the evaluation process. If the worker's results agree with the gold standard, we have greater confidence that the additional work performed by the worker is correct. Expert annotations are obtained from previous work[18,19] and through manual annotation by trained researchers.

3.3 Improved Deep Neural Network Training

Deep learning is a recent machine learning technique that has started to gain traction in digital pathology image analysis.[21] Deep learning models high level abstractions in data by using a deep graph with multiple processing layers, composed of multiple linear and non-linear transformations. Performance in virtually all computer vision tasks have improved dramatically by using a type of deep learning network, feed-forward Convolutional Neural Networks (CNNs). Unfortunately, many deep learning frameworks require tens or hundreds of thousands of training images to reliably estimate the millions of parameters of a deep neural network.[9] Deep learning uses an artificial neural network with multiple hidden layers of units between the input and output layers. Convolutional networks are distinct in that they use a convolutional filter layer that can process the 2D structure of images. These deep models, trained on large scale image data sources have outperformed all other known methods in large scale computer vision challenges.

As mentioned above, our architecture for the task of cell nuclei image segmentation is a modified SegNet fully convolutional encoder-decoder as seen in Figure 2. The first 10 encoder layers match the configuration of the first 10 convolutional layers in the VGG16[20] network. Each convolutional filter kernel is 3x3 and the max pooling layers are 2x2 with a stride of 2. The input to the neural network is an RGB 128x128 image and the output of the softmax layer is a binary output with the max probability class label set for every pixel. The final size of the



(a) J=0.8501 (b) J=0.9113 (c) J=0.9048

(d) J=0.772 (e) J=0.6202 (f) J=0.805

Figure 3. Examples of Amazon Mechanical Turk image annotations from seven unique workers per image. Each SVG polygon is displayed using a random color and visualized on top of the RGB image. The final user segmentation is computed by a majority vote rule i.e. a region is labeled as foreground if the area is marked by >4 turkers. Turkers are asked to annotate all nuclei, even those that touch the image border. The Jaccard index between the user segmentation and ground truth is shown under each image.

last layer is of the same width and height as the input. Since the beginning layers are identical to the VGG16 network, we are also able to transfer the trained weights from a pre-trained network. This would typically lead to a better initialization and faster convergence. In our application, we did not see much of a difference in the final segmentation accuracy between transfer learning and training from scratch, thus our results are based upon a network that is trained only with the data collected from our online workforce.

Another feature of our neural network is batch normalization.[22] Batch normalization was implemented to accelerate network training. Batch normalization is done between the output of the convolutional layers, and before the ReLU activation function. Batch normalization also acts as a regularizer. Because the normalization is computed on a per batch basis, and the batches are randomly selected from the training set, batch normalization introduces non-deterministic behavior into the training process which has regularizing effects, as the network must accommodate for noise.

In order to upsample and reverse the pooling layers, the decoder network memorizes the max-pool indices from the encoder portion of the network. The reverse step produces a sparse feature map that is convolved with subsequent layers to produce a dense feature map. The results also pass through a batch normalization layer and ReLU activation. The final architecture is constructed using the Caffe deep learning software package.[10]

4. EXPERIMENTS AND RESULTS

For our experiments and results, we present our data from the online annotation crowdsourcing tasks followed by the utilization of this data for the nuclei segmentation task.

4.1 Crowdsourcing Annotation Results

For the image annotations, each Amazon Mechanical Turk task e.g. HIT, is described by a title and description, and priced at $0.07 per task. The HIT has a specific set of instructions along with visual examples of the expected

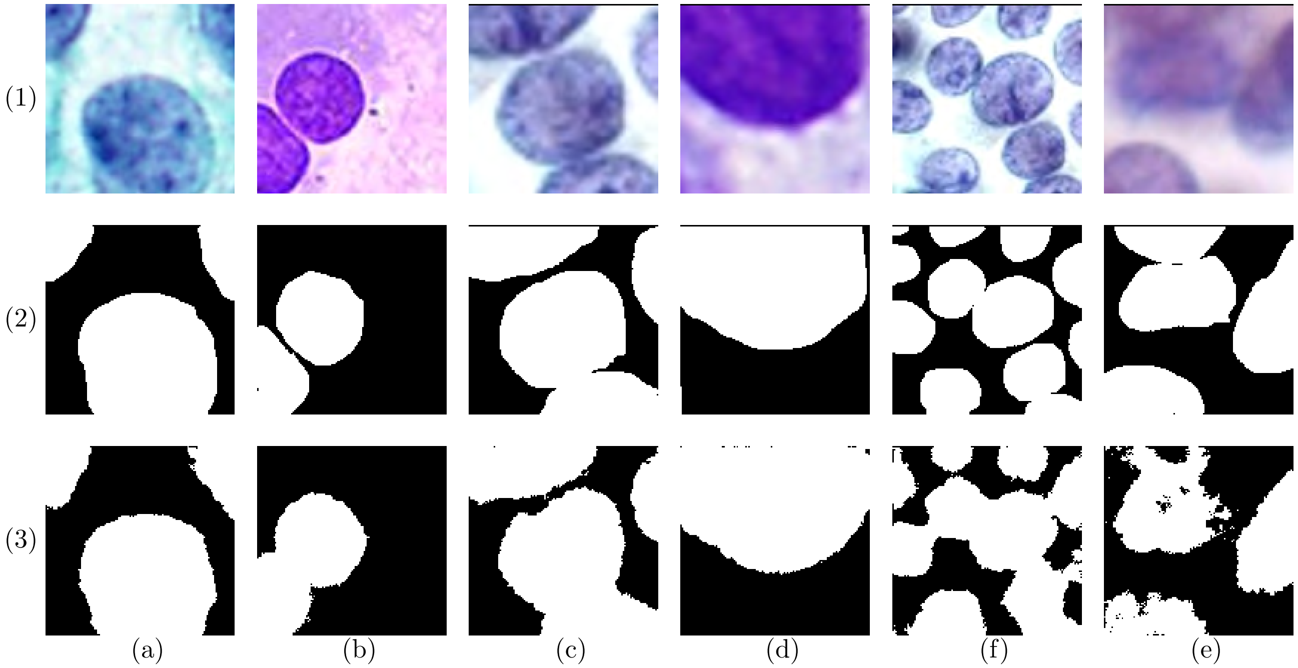


Figure 4. Qualitative segmentation results comparing the computed result against the ground truth. (1) The RGB input image to the classification neural network. (2) Ground truth nuclei segmentation created by a trained human. (3) The computed segmentation result from the encoder-decoder neural network. (a)-(e) Results presented for different types of stain and cytopathological characteristics.

quality of result on a sample image. Our dataset consists of 79 high magnification images that contain various stains, diseases, and cellular structures. We asked for 7 unique workers to outline all nuclei from these 79 images for a total of 553 requested tasks. Within two hours, all 553 tasks were completed by 56 unique workers. Because, some images contained multiple cells, the total number of nuclei outlined was 2,374. The financial burden was quite inexpensive to collect this annotation data; all tasks resulted in a total cost of $44.24 (including mturk fees).

To compute the accuracy of the crowdsourced annotations, we randomly selected 20 images from dataset and compared the user provided annotation to the ground truth annotations using the Jaccard Index (overlap score). The Jaccard index metric is defined as $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$, where A is the user segmentation and B is the ground truth mask. Some illustrations of the user provided annotations can be seen in Figure 3. We visualize every polygon using a random RGB color for visual differentiation. In order to compute a final crowdsourced segmentaiton map, we use a majority vote (>4 turkers) rule to determine the background and foreground. The final results and agreement between the randomly selected user annotation and ground truth masks are quite high, with a Jaccard similarity coefficient at, $J(A,B) = 0.837$ +/- 0.098.

4.2 Deep Learning Segmentation Results

Utilizing the same subset of images of nuclei and their corresponding annotation label, we train a deep encoder-decoder. In order to train the network without substantial over fitting, additional data was generated and the training set was augmented. Each image was randomly cropped into several subsections. Cropped subsections, and the original images were then all resized to 128x128, and then rotation and mirroring was applied to produce a data set with a total size of 1200 images. 1100 images were used for training, and the remaining 100 were reserved as a test set.

Our hyperparameters are as follows, batch size = 20 and learning rate = 0.001. We trained the network over 5000 epochs on a Tesla K40. The final Jaccard index of the computed and the ground truth is 0.892 +/- 0.054. Qualitative results of the original images, ground truth mask, and final segmentation output can be seen in Figure 4.

5. FUTURE WORK

The segmentation results are generally accurate; however, we are noticing some discontinuity in the segmentation results. This can be seen especially around the edges of the segmentation boundary where the softmax layer is generating an output that is roughly the same probability for both classes. Visually, one can see the smoothing issue in Figure 4(3e). Future work will look into alternative neural network architectures that are deeper with larger receptive fields to improve the segmentation smoothness. Also alternative deep network configurations, such as U-Net,[23] that utilize skip connections could also provide more continuity in the final result. We would like to also address the issue of maintaining the thin separation between nuclei. The current architecture has difficulty creating thin boundaries between very close foreground objects. Finally, we would like to collect more data for the current segmentation task, as well as other cytopathological annotation data that can be easily transformed into an online crowdsourcing task.

6. CONCLUSIONS

We presented our work on collecting, evaluation, and utilizing digital pathology annotation data for deep neural network classification. We utilized Amazon Mechanical Turk and SVG to collect boundary information for digital pathology and showed the feasibility of using non-medical professionals to assist with visual tasks. This annotation data was then utilized in a deep neural network for training an encoder-decoder framework. Again, both qualitative and quantitative results demonstrate the effectiveness of the neural network architecture given the layperson's annotation data.

REFERENCES

[1] Howlader, N., Noone, A., Krapcho, M., et al., "Seer cancer statistics review, 1975-2011.[based on the november 2013 seer data submission, posted to the seer web site, april 2014.]," *Bethesda, MD: National Cancer Institute* (2013). `http://seer.cancer.gov/statfacts/html/thyro.html`.

[2] Wang, W., Ozolek, J. A., and Rohde, G. K., "Detection and classification of thyroid follicular lesions based on nuclear structure from histopathology images," *Cytometry Part A* **77**(5), 485–494 (2010).

[3] Karslıoğlu, Y., Celasun, B., and Günhan, Ö., "Contribution of morphometry in the differential diagnosis of fine-needle thyroid aspirates," *Cytometry Part B: Clinical Cytometry* **65**(1), 22–28 (2005).

[4] Gupta, N., Sarkar, C., Singh, R., and Karak, A. K., "Evaluation of diagnostic efficiency of computerized image analysis based quantitative nuclear parameters in papillary and follicular thyroid tumors using paraffin-embedded tissue sections," *Pathology Oncology Research* **7**(1), 46–55 (2001).

[5] Frasoldati, A., Flora, M., Pesenti, M., Caroggio, A., and Valcavi, R., "Computer-assisted cell morphometry and ploidy analysis in the assessment of thyroid follicular neoplasms," *Thyroid* **11**(10), 941–946 (2001).

[6] Gurcan, M. N., Boucheron, L. E., Can, A., Madabhushi, A., Rajpoot, N. M., and Yener, B., "Histopathological image analysis: A review," *IEEE Reviews in Biomedical Engineering* **2**, 147–171 (2009).

[7] Madabhushi, A., "Digital pathology image analysis: opportunities and challenges," (2009).

[8] Irshad, H., Montaser-Kouhsari, L., Waltz, G., Bucur, O., Nowak, J., Dong, F., Knoblauch, N., and Beck, A., "Crowdsourcing image annotation for nucleus detection and segmentation in computational pathology: Evaluating experts, automated methods, and the crowd," in [*Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*], **20**, 294–305, World Scientific (2014).

[9] Taigman, Y., Yang, M., Ranzato, M., and Wolf, L., "Deepface: Closing the gap to human-level performance in face verification," in [*IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*], 1701–1708, IEEE (2014).

[10] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T., "Caffe: Convolutional architecture for fast feature embedding," in [*Proceedings of the ACM International Conference on Multimedia*], 675–678, ACM (2014).

[11] Yosinski, J., Clune, J., Bengio, Y., and Lipson, H., "How transferable are features in deep neural networks?," in [*Advances in Neural Information Processing Systems*], 3320–3328 (2014).

[12] Ren, S., He, K., Girshick, R., and Sun, J., "Faster r-cnn: Towards real-time object detection with region proposal networks," in [*Advances in neural information processing systems*], 91–99 (2015).

[13] Long, J., Shelhamer, E., and Darrell, T., “Fully convolutional networks for semantic segmentation,” in [*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*], 3431–3440 (2015).

[14] Ciresan, D., Giusti, A., Gambardella, L. M., and Schmidhuber, J., “Deep neural networks segment neuronal membranes in electron microscopy images,” in [*Advances in neural information processing systems*], 2843–2851 (2012).

[15] Badrinarayanan, V., Handa, A., and Cipolla, R., “Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling,” *arXiv preprint arXiv:1505.07293* (2015).

[16] Kim, E., Huang, X., and Tan, G., “Markup svg - an online content-aware image abstraction and annotation tool,” *IEEE Transactions on Multimedia* **13**(5), 993–1006 (2011).

[17] Sorokin, A. and Forsyth, D., “Utility data annotation with amazon mechanical turk,” *Urbana* **51**(61), 820.

[18] Kim, E., Baloch, Z., and Kim, C. S., “Computer assisted detection and analysis of tall cell variant papillary thyroid carcinoma in histological images,” in [*SPIE Medical Imaging*], 94200A–94200A, International Society for Optics and Photonics (2015).

[19] Kim, E., Corte-Real, M., and Baloch, Z., “A Deep Semantic Mobile Application for Thyroid Cytopathology,” in [*SPIE Medical Imaging: Advanced PACS-based Imaging Informatics and Therapeutic Applications.)*], (2016).

[20] Simonyan, K. and Zisserman, A., “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556* (2014).

[21] Malon, C., Miller, M., Burger, H. C., Cosatto, E., and Graf, H. P., “Identifying histological elements with convolutional neural networks,” in [*Proceedings of the 5th international conference on Soft computing as transdisciplinary science and technology*], 450–456, ACM (2008).

[22] Ioffe, S. and Szegedy, C., “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv preprint arXiv:1502.03167* (2015).

[23] Ronneberger, O., Fischer, P., and Brox, T., “U-net: Convolutional networks for biomedical image segmentation,” in [*International Conference on Medical Image Computing and Computer-Assisted Intervention*], 234–241, Springer (2015).