River Ice Segmentation with Deep Learning

Abhineet Singh  Hayden Kalke  Mark Loewen  Nilanjan Ray
University of Alberta
asinghl,kalke,mrloewen,nray1@ualberta.ca

Abstract

This paper deals with the problem of computing surface ice concentration for two different types of ice from river ice images. It presents the results of attempting to solve this problem using several state of the art semantic segmentation methods based on deep convolutional neural networks (CNNs). This task presents two main challenges - very limited availability of labeled training data and the great difficulty of visually distinguishing the two types of ice, even for humans, leading to noisy labels. The results are used to analyze the extent to which some of the best deep learning methods currently in existence can handle these challenges.

1. Introduction

The study of surface ice concentration and variation over time and place is crucial for understanding the river ice formation process. The temporal and spatial ice distributions thus computed can help to validate models of this process. The additional ability to distinguish frazil and the sediment-carrying anchor ice can also help to increase the estimation accuracy of the sediment transportation capacity of the river. Towards this end, a large amount of video data has been captured using UAVs and bridge mounted game cameras from two Alberta rivers during the winters of 2016 and 2017. The objective of this work is to analyze this data and perform dense pixel wise segmentation on these images and videos to be able to automatically compute the concentrations of the two types of ice.

The main challenge in this task is the lack of labeled data since it is extremely difficult and time consuming to manually segment images into the three categories due to the arbitrary shapes that the ice pans can assume. As a result, there are currently only 50 labeled images (Fig. 1) to accompany 564 unlabeled test images and over 100 minutes of unlabeled 4K videos. These labeled images along with 205 additional images with only ice-water labeling have already been used to train an SVM [17, 16, 15] to perform the segmentation. It provided water-ice classification accuracies ranging from 80.1% - 93.5% and surface ice concentration errors of 0.7% - 3.3%. Though it was fairly successful at separating ice from water, it had much greater difficulty in distinguishing between frazil and anchor ice pans, especially in cases where they are not physically separated and are hard to differentiate even for human eyes. This project is mainly concerned with handling these more difficult cases.

To address the limitations of SVM, this work uses recent deep CNN based methods of semantic segmentation. Since these methods need large amounts of training data to work well, the training images have been subjected to several data augmentation techniques (Sec. 3.2) to generate enough data.

2. Background

The river ice images look very similar to microscopic images of cells in the bloodstream so my initial idea was to try existing cell classification networks in medical imaging after fine tuning them on the training images. I found several promising works employing a range of architectures including ConvNet [30], LeNet [27, 23], Resnet [19] and Inception [20] that might have provided the base network for this work. However, more detailed examination revealed that medical imaging tasks are mainly concerned with the detection and localization of specific kinds of cells rather than performing pixel wise segmentation that is needed here.
Further, I looked for unsupervised or semi-supervised video segmentation techniques to utilize the large amount of high-quality video data. I found that most of them use optical flow for performing motion segmentation [10], though some appearance based [2] and hybrid [11, 14] methods have also been proposed. A recent work [24] proposes an unsupervised bootstrapping approach, somewhat similar to the one mentioned above. Under the assumption that all the moving pixels belong to the same foreground object, it uses the motion segmented images as training data to learn an implicit representation of this object. The model is used for refining the motion segmentation and the improved results are in turn used to bootstrap further refinements.

However, there are two assumptions underlying this work, and motion segmentation in general, which renders such methods unsuitable for our task. Firstly, they assume that there is a single moving foreground object. Our task, on the other hand, requires distinguishing between two different types of moving ice both of which are foreground objects. Secondly, they assume a static background while the river, which makes up the background in our case, is itself moving. Preliminary attempts to perform optical flow-based motion segmentation on these videos confirmed their unsuitability for this work.

I did find a recent method [29] for performing simultaneous optical flow estimation and segmentation which might be able to address these limitations to some extent. However, I was unable to get its Matlab code working in time and so deferred its further exploration to future work. Finally, I looked for existing applications of deep learning for surface ice analysis and though I did find one [6], it uses microwave sensor data instead of images.

### 3. Methodology

#### 3.1. Image Segmentation

Since neither cell classification nor video segmentation methods seemed promising, it was decided to use supervised image segmentation instead. After extensive research through several excellent resources for these methods [22, 9], four of the most widely cited and best performing methods with publicly available code were selected.

The first of these is UNet [25] from the medical imaging community. It was introduced for neuronal structure segmentation in electron microscopic images and won the ISBI challenge 2015. As the name suggests, UNet combines a contracting part with a symmetric expanding part to yield a U-shaped architecture that can both utilize context information and achieve good localization owing to the two parts respectively. It was shown to be trainable with relatively few training samples while relying heavily on patch based data augmentation which seemed to make it an ideal fit for this study.

The second network is called SegNet [1] and was introduced for segmenting natural images of both outdoor and indoor scenes for scene understanding application. It uses a 13-layer VGG16 net [28] as its backbone and features a somewhat similar architecture as UNet. The contracting and expanding parts are here termed encoder and decoder respectively and the upsampling units in the latter are not trainable, instead utilizing the weights learned by the corresponding max-pooling layers in the former. I have used Keras [8] implementations for both UNet and SegNet, available as part of the same repository [12] along with a couple of variants of the FCN architecture [21, 26]. These latter, however, did not perform as well as the other two and their results are thus excluded from this paper.

The third method is called Deeplab [3] and is one of the best performing methods in the Tensorflow research models repository [4]. It uses convolutions with upsampled filters - the so called atrous convolutions - to both achieve better control over the feature response resolution and to incorporate larger context without increasing computational cost. It also achieves scale-invariance by using a pyramidal max pooling and improves localization accuracy while maintaining spatial invariance by combining the last layer output with a fully connected conditional random field layer. I used a more recent version called Deeplabv3+ [5] which adds a decoder module to produce sharper object boundaries while also incorporating the Xception model [7] for further performance improvements.

The fourth method is based on the DenseNet architecture [13]. To the best of our knowledge, this architecture has not yet been applied for segmentation but is included here due to its desirable property of providing state of the art performance with a much smaller network size. The basic idea of DenseNet is to connect each hidden layer of the network to all subsequent layers so that the feature maps output by each layer are used as input in all subsequent layers. This provides for better feature propagation and reuse while drastically reducing the total number of parameters and mitigating the vanishing gradient problem. The architecture used in this work had 9 such layers though experiments were done with the more layers up to 21 (Sec. 3.2).

As shown in Table 1, DenseNet has by far the fewest parameters of all the models tested here, being over 2 orders of magnitude smaller than the next smallest model.
Figure 2: Performance of SVM (dotted line) and all deep models (solid lines) on the 18-image test set using training with 32 images. Note that the range of Y-axis on the two plots are different to maximize line separability.

### 3.2. Data augmentation and Training

A simple sliding window approach was used to extract a large set of sub-images or patches from each training image. The window was moved by a random stride between 10% to 40% of the patch size $K$. This process was repeated after applying random rotations to the entire image between 15 to 345 degrees divided into four bands of equal width to allow for multiple rotations for each image. Finally, each patch was also subjected to horizontal and vertical flipping to generate two additional patches. All resultant patches were combined together to create the dataset for each $K$.

For testing a model, patches of size $K$ were extracted from the test image using a stride of $K$, segmentation was performed on each patch and the results were stitched back to get the final result.

All models were trained and tested using patch sizes $K \in \{256, 384, 512, 640, 800, 1000\}$. The 50 labeled images were divided into two sets of 32 and 18 for generating the training and testing/validation images respectively. Note that the training sets were used for generating the quantitative performance results on the validation sets while the combined sets generated using all 50 images were used for producing qualitative results on the unlabeled validation set.

UNet and SegNet were both trained for 1000 epochs and the training and validation accuracies were evaluated after each. The trained model used for testing was the one with either the maximum validation accuracy or the maximum mean accuracy depending on how well the training and validation accuracies were matched in the two cases. Deeplab was trained for between 100,000 and 200,000 steps. Batch size of 10 was used for $K = 256$ and 2 for $K \in \{640, 800, 1000\}$ with the latter chosen due to memory limitations. $K = 384$ was tested with batch sizes 6 and 8 while $K = 512$ was tested with 6 and 2. Most tests were conducted using the default stride of 16 with corresponding atrous rates of [6, 12, 18] though one model with $K = 256$ was also trained using Stride 8 with atrous rates of [12, 24, 36].

DenseNet training was a bit trickier. Simply using all the pixels for training caused the network to rapidly converge to a model that labeled all pixels with the class with the maximum number of training pixels - water in most cases. To get meaningful results, the number of pixels belonging to each of the classes had to be balanced. Therefore 10,000 random pixels belonging to each class were selected in each epoch, with different sets of pixels selected each time, and only these were used for computing the loss. Training images with less than 10,000 pixels in any class were discarded. Number of epochs were between 1000 – 1600 for all $K$. In all cases, the performance metrics in Sec. 4.1 were computed on the validation set every 10 epochs and training was stopped when these became high enough (e.g. pix_acc > 0.9) or remained unchanged for over 100 epochs.

### 3.3. Ablation Experiments

One of the principle difficulties in training deep models for performing segmentation is the lack of sufficient labeled data since it is very time-consuming to manually generate dense pixel wise segmentation masks of images. This problem is exacerbated in the current task because of the dif-
ficulty in distinguishing between the two types of ice that exhibit both very high intraclass variation and significant appearance overlap in addition to arbitrary and difficult to delineate shapes. As a result, a highly desirable attribute of a practically applicable model would be its ability to learn from very few images including partially labeled ones.

Two different types of ablation experiments were performed in order to explore the suitability of the tested models in this regard. The first one was to train the models using different subsets of the training set. The second one was to consider the labels from only a small subset of pixels in each image to simulate the scenario of partially labeled training data. Note that the input image itself was left unchanged so that the models did have access to all the pixels but the loss function minimized during training was computed using only the labels from the selected pixels.

4. Results

4.1. Evaluation metrics

Following evaluation metrics [26, 18] have been used:

1. Pixel accuracy:

\[
pix_{\text{acc}} = \frac{\sum_i n_{ij}}{\sum_i t_i} \tag{1}
\]

2. Mean accuracy:

\[
\text{mean}_{\text{acc}} = \frac{1}{n} \sum_i \frac{n_{ii}}{t_i} \tag{2}
\]

3. Mean IOU:

\[
\text{mean}_{\text{iou}} = \frac{1}{n} \sum_i \frac{n_{ii}}{t_i + \sum_j n_{ji} - n_{ii}} \tag{3}
\]

4. Frequency Weighted IOU:

\[
fw_{\text{iou}} = \sum_k (t_k)^{-1} \sum_i \frac{t_i n_{ii}}{t_i + \sum_j n_{ji} - n_{ii}} \tag{4}
\]

where \( n = 3 \) is the number of classes, \( n_{ij} \) is the number of pixels of class \( i \) predicted to belong to class \( j \) and \( t_i \) is the total number of pixels of class \( i \) in the ground truth. Note that accuracy and IOU are respectively equivalent to the recall and precision metrics that are typically used in classification and detection. The former measures the rate of true positives while the latter also accounts for false positives.

All of these metrics measure the combined segmentation performance over all three classes. This can lead to biased results when the number of image pixels is not evenly distributed between the classes. This is particularly so in the current work whose main objective is to distinguish between the two types of ice as it is almost trivial to separate water from ice. However, as shown in Table 2, more than half the pixels in both sets of test images are of water while anchor ice, which is the most difficult to segment, covers only about 1 in 6 pixels. Therefore, the results presented in the next section include class specific versions of these metrics for each type of ice, sometimes in addition to the combined metrics over all three classes.

| test images | water     | anchor ice | frazil ice |
|-------------|-----------|------------|------------|
| 18          | 0.557     | 0.175      | 0.268      |
| 46          | 0.610     | 0.166      | 0.224      |

4.2. Quantitative

DenseNet turned out to perform best with \( K = 800 \) while all other models did so with \( K = 640 \). All subsequent results were therefore obtained using \( K = 640 \). Two of them, Deeplab and UNet, maintain the superiority with frazil ice as well but the other two, which were also the best performers with anchor ice, fall slightly behind. This trend of an inverse relationship between the accuracy of anchor and frazil ice was consistently observed in all the tests. It seems that learning to better distinguish anchor ice from frazil ice comes at the cost of either a decrease in the capability to recognize frazil ice itself or an overcorrection which causes some of the more ambiguous cases of frazil ice to be mistaken for anchor ice. It is likely that the loss function can be equally well minimized by overfitting either to frazil ice or to anchor ice thus leading to two stable training states.

It can also be seen that the performance difference between deep models and SVM decreases when all three classes are averaged, as in mean\(_{\text{acc}}\) and mean\(_{\text{iou}}\), and even more so when the averaging is frequency dependent, as in \(pix_{\text{acc}}\) and \(fw_{\text{iou}}\). As mentioned before, these are the cases when the high segmentation accuracy of water starts to dominate.

Comparing between the deep models themselves, Deeplab turns out to be the best overall, followed closely by...
Figure 3: Results of ablation tests on (clockwise from top left) SVM, Deeplab, UNet and DenseNet using 4, 8, 16, 24 and 32 training images. Accuracy and IOU results are respectively shown with solid and dotted lines.

UNet, though both these models are slightly outperformed by the other two with respect to anchor ice accuracy. The corresponding dip in the frazil ice accuracy of these latter models, to a point that it is even slightly worse than SVM, is consistent with the inverse relationship between anchor and frazil ice performance mentioned above. It is also interesting to note that both of them are worse than that of both Deeplab and UNet in terms of anchor ice IOU. Since accuracy does not penalize false positives while IOU does, this is most likely an indication that these methods misclassify frazil ice as anchor ice more often than the others which is consistent with the inverse relationship hypothesis.

4.2.2 Ablation study with training images

For this study, models were trained using 4, 8, 16, 24 and 32 images and each one was tested using the 18 test images. Results for individual models are given in Fig. 3. Somewhat contrary to expectation, a sufficiently distinct pattern of improvement with more images did not become apparent for most of them. There does seem to be a slight improvement in anchor ice accuracy and IOU for all models except Deeplab but it is too weak to draw any conclusions. Also, experiments have shown that these trends are highly susceptible to the individual checkpoint which happens to be used for inference since re-training some of these models for almost the same number of epochs or simply using a
Figure 4: Comparing ablation performance using (left) anchor ice and (right) frazil ice. Accuracy and IOU results are respectively shown with solid and dotted lines. Note the different y-axis limits on the two plots.

Figure 5: Comparing ablation performance using selective pixels for (left) anchor ice and (right) frazil ice. Accuracy and IOU results are respectively shown with solid and dotted lines.

sufficiently different checkpoint from the same training session was found to change the performance significantly. A more likely conclusion is that the test set is just too similar to the training set and does not contain enough challenging variation to allow the extra information from more training images to be reflected in the performance numbers. This supposition is lent some credence by the fact that the 50 labeled images were specifically chosen for their ease of labeling since it is highly tedious and time-consuming to manually perform pixel-wise segmentations. Combined with the fact that they were labeled by the same person, it would not be unusual for them to be similar, both in terms of content and level of challenge. Both the significant trends observed in the previous section of the inverse relation between anchor and frazil ice performance and the dominating effect of water are apparent in all of these plots as well with the exception of Deeplab. Note, for instance, how the lines for anchor and frazil ice (green and red) are virtually reflec-
tions of each other in all three cases while those for mean over all the classes (blue) are nearly horizontal. Deeplab itself exhibits mostly constant performance with the exception of a significant dip with 16 images which was probably due to the aforementioned susceptibility of these results to the specific checkpoint used for testing.

Results comparing the ablation performance of different models for anchor and frazil ice are given in Fig. 4. Not much of interest is apparent here except that the dominance of Deeplab over the other deep models is much less prominent compared to Fig. 2 and it is DenseNet which seems to be the overall best performer for anchor ice while UNet takes this position for frazil ice. The competitive performance of DenseNet is noteworthy considering that it has over 2 orders of magnitude fewer parameters than the other models, though this might be at least partially attributable to the limited challenges available in the test set. Unfortunately, this smaller size does not translate into higher speed or lesser memory requirement as both of these lie somewhere between UNet and Deeplab. It does result in a much smaller model which might be more suitable for a mobile device with limited storage or when a large number of these models specialized for different tasks are to be available simultaneously and then chosen dynamically at runtime.

### 4.2.3 Ablation study with selective pixels

This study was performed by training models using 2, 10, 100 and 1000 pixels per class from each sub-image. The sub-image set was generated from only 4 training images and not subjected to augmentation. Also, \( K = 640 \) was used for all models including DenseNet to ensure that the number of training pixels remained identical for all of them. Further, unlike the previous sections, these models were tested on all of the remaining 46 labeled images rather than only 18 in an attempt to counteract the limited challenges available there. Finally, SVM was not included here because its super-pixel based method [17] does not lend itself well to training using randomly selected pixels.

Results are given in Fig. 5. Note that the all pixel results are for the augmented sub image test set generated from 4 images while all others are for the unaugmented test set. It turns out that selective pixel training has surprisingly little impact on quantitative performance except perhaps in the case of UNet with anchor ice and DeepLab with frazil ice. Though there is indeed a more strongly marked upward trend in performance compared to training images (Fig. 4), it is not as significant as would be expected. Particularly remarkable is the case of using only 2 pixels per class. When combined with the fact that the unaugmented sub-image test set contained only 46 sub-images, this training was done using only 92 pixels per class or 276 pixels in all. This might be another indicator of the limitations of the test set which is

### 4.3. Qualitative

#### 4.3.1 Images

Fig. 6, 7 and 8 show the results of applying the best configurations of the four models to segment several images from the unlabeled test set. Several interesting observations can be made. Firstly, both UNet and SegNet misclassify water as frazil ice in several cases where most of the image contains water, e.g. in images 2 and 3 respectively and of Fig. 6 and 7. DenseNet too seems to be somewhat susceptible to this issue though to a much lesser extent. Secondly, Deeplab results show the largest degree of discontinuity between adjacent patches due to its tendency to occasionally produce completely meaningless segmentations on some individual patches. Examples include image 6 in Fig. 6, image 5 in Fig. 7 and images 1 and 4 in Fig. 8. Thirdly and consistently with the quantitative results of the previous section, DenseNet is overall the best performing model even though its results are slightly more fragmented than the others. This is particularly noticeable in the more difficult cases of distinguishing between frazil and anchor ice when they both form part of the same ice pan. Examples include images 1 and 7 in Fig. 7 and image 1 in Fig. 8.

Qualitative results on labeled test images are available in Google Photos albums whose links are in Tables 3 and 4. Each row in these albums shows (from left to right) the original image, its ground truth label and its segmentation result.
Figure 6: Results of applying the best configurations of the four models on test images: left to right: UNet, SegNet, Deeplab, DenseNet
Figure 7: Results of applying the best configurations of the four models on test images: left to right: UNet, SegNet, Deeplab, DenseNet
Figure 8: Results of applying the best configurations of the four models on test images: left to right: UNet, SegNet, Deeplab, DenseNet
4.3.2 Videos

All the deep models were evaluated on 1 to 2 minutes sequences from the following 5 videos captured on 3 different days and containing wide variations in the scale and form of ice pans:

1. 20160121\_YUN00001
2. 20160122\_YUN00002
3. 20160122\_YUN00020
4. 20161203\_Deployment\_1\_YUN00001
5. 20161203\_Deployment\_1\_YUN00002

SVM took 5 minutes to process each frame so could only be evaluated on 30 seconds of video 1 and 10 seconds each of videos 3 and 4. Also, selective pixel models were only evaluated on videos 1 and 3. All videos are available in this Google Drive folder.

The most noticeable point in these results is that DeepLab is susceptible to completely misclassifying individual randomly distributed patches which can lead to strong discontinuities when these patches are stitched together to create complete frames and the corresponding video. Another important result is the one mentioned in Sec. 4.2.3 - selective pixel training has significantly greater impact in practice than indicated by Fig. 5. The segmentation masks seem to become more grainy and sparse as the number of pixels is decreased and there is a very noticeable difference between using 2 and 1000 pixels.

5. Conclusions and Future Work

This paper presented the results of using four state of the art deep CNNs for segmenting river ice images into water and two types of ice. Three of these - UNet, SegNet and Deeplab - are previously published and well studied methods while the fourth one - DenseNet - is a new method, though based on an existing architecture. All of the models provided fairly good results, both quantitatively on the labeled validation images as well as qualitatively on the unlabeled test images. These represented a significant increase in accuracy over previous attempts using SVM, especially in distinguishing between the two types of ice nearly 20% for anchor ice and 15% for frazil ice. Among the four models, DenseNet performed the best even though it uses the fewest parameters by far. This provides a promising avenue for future exploration that might be able to yield much better performance with more layers and training images.

This paper also demonstrated reasonable success in handling the lack of labeled images using data augmentation. Further improvements in this direction might be obtained by using the augmented dataset as the starting point for a semi-automated boot-strapping process. This involves training successively better models by manually correcting the segmentation results produced on the test images by each stage of the process and adding these corrected images to the labeled set for training the next stage model.

References

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell., 39(12):2481–2495, 2017.

[2] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. One-shot video object segmentation. In Computer Vision and Pattern Recognition (CVPR), 2017.

[3] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Trans. Pattern Anal. Mach. Intell., 40(4):834–848, 2018.

[4] L.-C. Chen, Y. Zhu, and G. Papandreou. DeepLab: Deep Labelling for Semantic Image Segmentation. Github, 2017. https://github.com/tensorflow/models/tree/master/research/deeplab.

[5] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. arXiv:1802.02611, 2018.

[6] J. Chi and H.-c. Kim. Prediction of arctic sea ice concentration using a fully data driven deep neural network. Remote Sensing, 9(12), 2017.

[7] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In CVPR, pages 1800–1807. IEEE Computer Society, 2017.

[8] F. Chollet et al. Keras. https://keras.io, 2015.

[9] Dr. Tang. Semantic-Segmentation. Github, 2017. https://github.com/tangzhenyu/SemanticSegmentation\_DL.

[10] A. Faktor and M. Irani. Video segmentation by non-local consensus voting. In BMVC, 2014.

[11] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In Computer Vision and Pattern Recognition (CVPR 2010), 2010.

[12] D. Gupta. Image Segmentation Keras: Implementation of Segnet, FCN, UNet and other models in Keras. Github, 2017. https://github.com/divamgupta/image-segmentation-keras.

[13] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In CVPR, pages 2261–2269. IEEE Computer Society, 2017.

[14] S. Jain, B. Xiong, and K. Grauman. FusionSeg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. CVPR, 2017.

[15] H. Kalke. Sediment Transport by Released Anchor Ice: Field Measurements and Digital Image Processing. Master’s thesis, University of Alberta, 2017.

[16] H. Kalke and M. Loewen. Support Vector Machine Learning Applied to Digital Images of River Ice Conditions. Sub-
mitted for review to Cold Regions Science and Technology, September 2017. 1

[17] H. Kalke and M. Loewen. Predicting Surface Ice Concentration using Machine Learning. In 19th Workshop on the Hydraulics of Ice Covered Rivers, Whitehorse, Yukon, Canada, July 9-12 2017. CGU HS Committee on River Ice Processes and the Environment. 1, 7

[18] M. Ker?ner. Image Segmentation Evaluation. Github, 2017. https://github.com/martinkersner/py_img_seg_eval. 4

[19] H. Lei, T. Han, W. Huang, J. Y. Kuo, Z. Yu, X. He, and B. Lei. Cross-modal transfer learning for hep-2 cell classification based on deep residual network. In 2017 IEEE International Symposium on Multimedia (ISM), pages 465–468, Dec 2017. 1

[20] Y. Li and L. Shen. A deep residual inception network for hep-2 cell classification. In Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, pages 12–20. Springer International Publishing, 2017. 1

[21] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In CVPR, pages 3431–3440. IEEE Computer Society, 2015. 2

[22] mrgloom. Awesome Semantic Segmentation. Github, 2017. https://github.com/mrgloom/awesome-semantic-segmentation. 2

[23] D. Parthasarathy. Classifying white blood cells with deep learning. online: https://blog.athelas.com/classifying-white-blood-cells-with-convolutional-neural-networks-2ca6da239331, March 2017. 1

[24] D. Pathak, R. Girshick, P. Dollár, T. Darrell, and B. Hariharan. Learning features by watching objects move. In CVPR, 2017. 2

[25] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In MICCAI (3), volume 9351 of Lecture Notes in Computer Science, pages 234–241. Springer, 2015. 2

[26] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. IEEE Trans. Pattern Anal. Mach. Intell., 39(4):640–651, 2017. 2, 4

[27] A. Shpilman, D. Boikiy, M. Polyakova, D. Kudenko, A. Burakov, and E. Nadezhdina. Deep learning of cell classification using microscope images of intracellular microtubule networks. In 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), pages 1–6, Dec 2017. 1

[28] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. CoRR, abs/1409.1556, 2014. 2

[29] Y.-H. Tsai, M.-H. Yang, and M. J. Black. Video segmentation via object flow. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3899–3908, 2016. 2

[30] L. Zhang, L. Lu, I. Nogues, R. M. Summers, S. Liu, and J. Yao. Deeppap: Deep convolutional networks for cervical cell classification. IEEE Journal of Biomedical and Health Informatics, 21(6):1633–1643, Nov 2017. 1