Using Deep Learning for Segmentation and Counting within Microscopy Data

Carlos Xavier Hernndez
Stanford University
Stanford, CA
cxh@stanford.edu

Mohammad M. Sultan
Stanford University
Stanford, CA
msultan@stanford.edu

Abstract

Cell counting and segmentation is a laboriously difficult task that would greatly benefit from automation. In this paper, we describe methods for automatic cell-segmentation and counting by combining the Mask R-CNN method with a VGG style neural network for segmenting and counting the number of cells given on an image.

1. Introduction

Cell segmentation and counting is a currently a laborious task requiring the use of gridded averages[1].The scientist manually estimates the number of cells in a local grid on a rectangular plate. This is repeated a few times at various grid points across the plate to get a mean density which is then used for estimating the total number of cells. These density based techniques suffer from several problems. One, they require the scientist to physically count the number of cells, likely leading to errors. Two, they require a significant amount of time commitment which could be better used for understanding, designing, and performing a new set of experiments. Three, it is not completely obvious how error bars could be obtained from such an analysis.

More succinctly, we aim to automate the tedious process of counting cells. Our methodology follows a two step approach.

- Cell segmentation to find a mask capable of outlining the boundaries of cells.
- Cell counting using the masking results generated in the previous step.

We also chose to model uncertainties in our model using methods outlined in [2].

2. Previous Work

The cell segmentation problem is an instance of mask-prediction problem and has been under investigation for a while. There are several instances of using convolutional neural networks (CNNs) for predicting

To our knowledge, there is currently no CNN architecture for counting objects in an image, much less counting cells.

3. Technical Approach

3.1. Cell Segmentation using Mask R-CNN

we first use Facebook's feature pyramid network(FPN) coupled to a Mask R-CNN to segment cells. The feature pyramid network[3] is a feature extraction network designed to build feature maps at multiple spatial scales. It is a computationally efficient implementation that relies on cross-links1 between successive convolution layers. These cross-links enable the network to infer not only the relevant features but their spacial correlations as well.

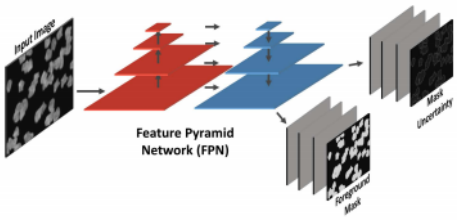

Figure 1: A schematic of our Feature Pyramid Network for generating a foreground mask.

3.2. Cell Counting using the VGG-11 Network

For the counting network, we chose to look at the work of Visual Geometry Group(VGG). The VGG networks(Figure 2) are deep convolutional neural networks that won the ImageNet challenge in 2014. We used the VGG-11 network which consists of 11 layers of two dimensional convolutions with a filter size of 3*3 pixels. The number of filters varied from 64 to 512. The convolutional layers were followed by max-pooling layers and

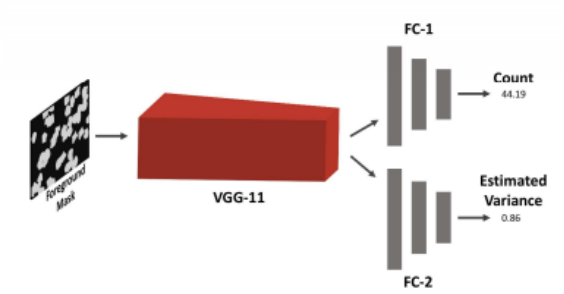

Figure 2: A schematic of our VGG-11-based network for counting cells from a foreground mask generated using the network from Figure 1

Leaky ReLu. Furthermore, after every convolution layer, we also added a batch normalization layer which has been shown to improve training stability by preventing gradients from blowing up or going to zero. The VGG-11 network was used as a feature extractor for the Counting network. For the actual count prediction, we used three fully connected layers which were again separated by a batch normalization layer and leaky ReLu. For the final layer, we used a ReLu layer to prevent the network from outputting values below 0(aka negative counts). A similar fully connected network was used to predict the variance. The full configuration and code is given in the Github repository.

3.3. Variance prediction for Cell counting and segmentation

Most neural networks are not designed to assign associated error bars to their predicted values. This makes it difficult to assess the model's confidence in its prediction. Such uncertainties might be useful in evaluating the model's strengths or/and weaknesses. These uncertainties could also help us to get an idea about what kind of data is underrepresented in our training ensemble.

However, recent work in computer vision[2] attempts to correct this by using a Bayesian deep learning frame work. With in this framework, the final loss is modified to include an additional term.

$$Loss = \frac{\|y - \hat{y}\|}{2\sigma^2} + \log \sigma^2 \tag{1}$$

Here, y is the true label, $\hat{y}$ is the predicted label, and σ is the predicted uncertainty. Under this scheme, the model tries to predict not just the correct output y but also an associated variance. In the original paper[2], it was shown empirically that the modified loss allowed the model to assign uncertainties to masking results. More concretely, it allowed the model to assign higher uncertainties to areas of the mask where multiple objects were overlapping for example. We chose to include the uncertainty estimation in both the segmentation and counting network. We believe this approach allows us to not only understand the weaknesses within our dataset and our model but also provide error estimates for downstream scientific analysis.

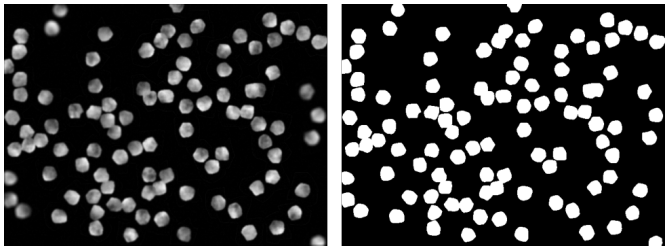

Figure 3: Sample in-focus image from the Broad dataset. The raw image is shown on the left while the masked image is shown on the right.

4. Dataset

We used the BBBC005 dataset from the Broad Institute's Bioimage Benchmark Collection3. This dataset is a collection of 9,600 simulated microscopy images of stained cells. These images were simulated for a given cell count with a clustering probability of 25%. Focus blur was simulated by applying Gaussian filters to the images. Each image is 696 x 520 pixels in 8-bit TIFF format (eventually converted to JPEG and scaled down to 256 x 192 pixels), with cell areas matched to the average cell areas of human U2OS cells.

Of the 9,600 images, 600 images have a corresponding foreground mask. All 9600 images have associated cell counts with an upper limit of 100. The FPN was trained on the 600 images while the Counting network was trained on the full dataset. We use about 100 of those for fast prototyping. We used a standard 80-20 train/test split for the final models.

5. Results

5.1. Segmentation Results

5.2. Counting Results

After 50 epochs of training, our best model is able to achieve an R^2 value of .987. 80% of the time, the ground truth falls within the predicted 95% confidence interval.

6. Model interpretation

A critical flaw of modern ML is that the resulting models are difficult to interpret. Such interpretation can increase our understanding of the underlying problem, providing insight into critical dynamics or highlighting interesting nontrival aspects within our dataset. These insights might be useful for designing future experiments or perhaps even improving the model itself.

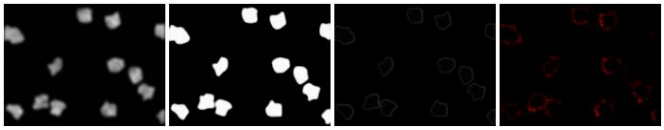

Figure 4: Results of the saliency maps for a random input example. The first image is the raw input image. The second and third columns correspond to the outputs of the FPN network. The saliency map is shown in the fourth column with red values indicating greater importance.

Therefore, we decided to probe our model using saliency maps[4]. These maps are a standard technique in CNN literature to probe the internal states of the neural network. At a simplistic level, they are designed to highlight pixels in the data that maximally influence the predicted score. Saliency maps tend to highlight important features within the data which can then be used to understand what it is that the model is maximally looking at.

We applied the saliency map technique to the counting network. The results are show in Figure 4. Our analysis shows that the network is trying to find outlines of cells within individual images. More importantly, the saliency maps are agnostic to the number, size, and, orientation of cells within the images, indicating the models' generalizability.

7. Future Work

There are several possible extensions to the methodologies presented in this paper. On the methodology side, we have not optimized the depth or architecture of the counting network. It is entirely possibly that other variants of the VGG network or newer networks such as Residual Networks might perform better.

On the engineering side, it would be interesting to implement the current models into a smart phone application to allow for scientific researchers to use our model in a research environment. Ultimately, we believe cell-counting should change from a researcher sitting at a lab using a counter to simply taking a picture and writing down the corresponding number.

References

[1] Counting cells using a hemocytometer. `http://www.abcam.com/protocols/counting-cells-using-a-haemocytometer`. Accessed: 2017-06-09.

[2] A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? *CoRR*, abs/1703.04977, 2017.

[3] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. *CoRR*, abs/1612.03144, 2016.

[4] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034, 2013.