Convolutional Neural Networks for Spot Detection

Applied to Fluorescent Microscopy Images

Jared Galloway[1], Nick Wagner[1], Annie Wang[1], Jake Searcy[2], Sarah Stednitz[3,4], Philip Washbourne[4]

1.Bioinformatics and Genomics Master's Program, University of Oregon 2.Research Advanced Computing Services, University of Oregon 3.Max Planck Institute for Biological Cybernetics 4.Institute of Neuroscience, University of Oregon

BGMP@UO Bioinformatics and Genomics Master's Program

Phil and Penny Knight Campus for Accelerating Scientific Impact

Abstract

Spot detection is an important task in many fields such as biology, astronomy, and physics. Unfortunately, the task of counting spots in images can take experts many laborious hours to do by hand. Recent studies have shown that machine learning has the potential to automate this task, however, it remains a problem to acquire pixel-level annotations needed as targets for any type of supervised learning. Here, we have written a dot simulator to emulate fluorescent spots on background noise, dubbed **SimuSpot**, and thus allowing users to completely side-step the need for human annotated data sets. Our results exemplify that this method offers a competitive F_1 score on empirical, fluorescent microscopy images when compared to similar studies utilizing supervised machine learning methods.

Fluorescent Confocal Immunolabeling

Imaging method that utilizes fluorescently tagged antibodies to target specific areas within cells, tissue, and organs. This is a very common method in the field of Biology because of how versatile it is. The versatility comes from being able to design both primary and secondary antibodies.

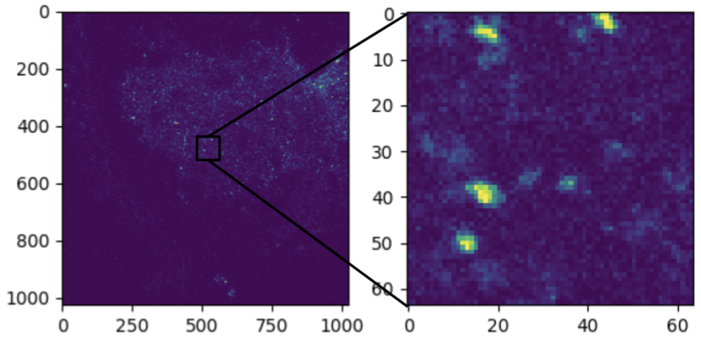

Figure 1:*Left* shows a 1024 × 1024 pixel empirical image of only the z channel. For this channel anitibodies were designed to target the synaptotagmin protein that is present during synapse communication. *Right* shows a 64 × 64 pixel zoomed section that feed into our model for prediction.

Convolutional Neural Networks

Convolutional neural networks (CNN) is commonly used for image classification and recognition. It takes in an input image and divides it into small sections. With the small sections, it is able to detect patterns and output to a class.

Spot Detection for Synapse Identification

Within the context of neuroscience, a tool like this would be very beneficial for understanding and measuring synapse formation and communication as an observable phenotypic trait. We could then compare this trait between individuals to learn more about synapse formation and the genetic basis for common neurological diseases using genome wide association studies (GWAS) or other phenotypic-based experiments.

Methods

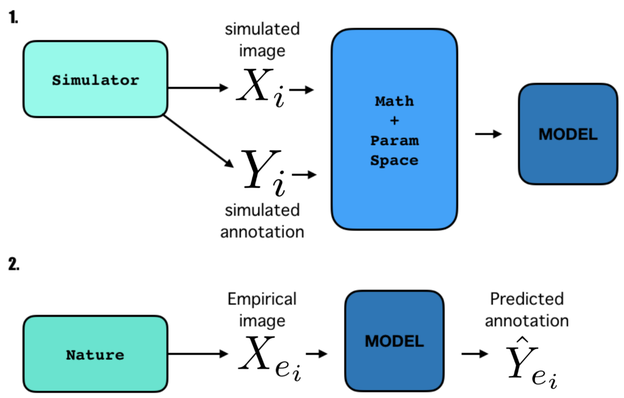

Figure 2:Part 1 (*top*) describes the simulation and training procedure - Each simulation is a pair of both the noisy image, X_i and its respective ground truth binary annotation Y_i. These data set are then used to fit our convnet to produce a predictive model. Part 2 (*bottom*) shows an empirical image (in our case a confocal fluoresce image, X_{e_i}) being fed into the same model to produce a prediction for synapse location, $\hat{Y}_{e_i}$.

Simulation

In our simulations, we modeled each fluorescent cluster in confocal images with a 2-dimensional Gaussian "bump" scaled by the radius for any given spot. Concretely, suppose (x_f, y_f) is the focal point for a bump. Given a pixel at (x_p, y_p) with distance $d = \sqrt{(x_f - x_p)^2 + (y_f - y_p)^2}$ to the focal point, if d is less than the radius r of the bump, the activation A of the pixel is defined as:

$$A(r, d) = e^{-\frac{\ln 10}{r^2} d^2} + \mathcal{N}(\mu = 0,\, \sigma^2). \tag{1}$$

Here σ is provided by the user as a parameter to our simulator. Both the number of dots in any given image, and the radius of any *one* dot are pulled from a binomial distribution, where n, and p, are parameters provided by the user. Once the number of dots are chosen, the simulator uniformly places focal points in the given dimensions for an $M \times N$ pixel image. Finally, Background noise for any pixel not in the radius of a dot is pulled from a Gaussian distribution.

Our Model

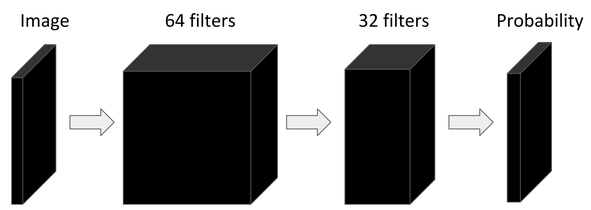

Figure 3:With no down-sampling or pooling, we have three convolutional layers with and 64, 32, and 1 filters, respectively. Each layer has a 3X3 kernel, the first two with leakyReLU activation and the final probability map with a sigmoid activation. Finally, we use binary crossentropy loss function and adam optimizer

Results

Our results highlight the ability for CNN's trained on simulated images to predict on a confocal microscopy image tagged with fluorescent immuno-labeled post-cleft synapse stains. We use F_1 score to quantify our prediction when comparing to an annotation quality checked by a nueroscience expert.

F_1 Score

F_1 score is one of the most common metrics when assessing a machine learning algorithm. It is defined as the harmonic mean of precision and recall, as shown below. In other words, it measures the exactness and completeness of the algorithm. The range of F_1 score is from 0 to 1, where 1 is the best score and 0 is the worst.

$$F_1 = \frac{2}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}} \tag{2}$$

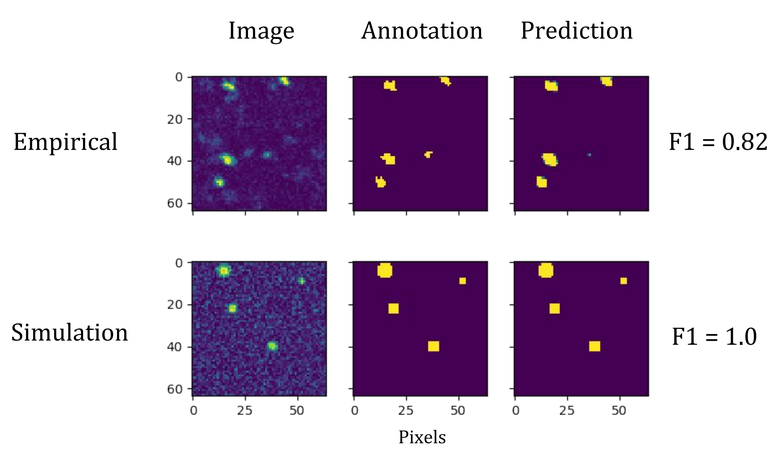

Figure 4:A panel of results showing both a simulated images (*bottom row*) and empirical confocal brain scan (*top row*). The three columns represent the image input to our model, the annotation used for training and accuracy metrics, and the raw probability map output from out model, respectively.

Software	Precision	Recall	F_1 Score
GoogleNet	0.833	0.751	0.784
AlexNet	0.842	0.703	0.758
detectSpot	0.836	0.740	0.782
SimuSpot	**0.762**	**0.895**	**0.824**

Table 1:A table comparing our method, **SimuSpot**, to other spot detection results. The results from all other methods were trained/tested on **different** simulated data - we are simply reporting and comparing results to a study that can be found in Mabaso et. al. 2018.

Acknowledgments

This work benefited from access to the University of Oregon high performance computer, Talapas.

For our Github page and full references please scan:
`https://github.com/jgallowa07/Spot-Detection`

SCAN ME