EXPERIMENTALLY-GENERATED GROUND TRUTH FOR DETECTING CELL TYPES IN AN IMAGE-BASED IMMUNOTHERAPY SCREEN

Joseph Boyd, Zelia Gouveia, Franck Perez, Thomas Walter

EXPERIMENTALLY-GENERATED GROUND TRUTH FOR DETECTING CELL TYPES IN AN IMAGE-BASED IMMUNOTHERAPY SCREEN

Joseph Boyd[1,2,3], Zelia Gouveia[2,4], Franck Perez[2,4], and Thomas Walter[1,2,3]

[1]MINES ParisTech, PSL Research University, Centre for Computational Biology, F-75006 Paris, France
[2]Institut Curie, 26 rue d'Ulm, 75005 Paris
[3]INSERM, U900, F-75005 Paris, France
[4]INSERM, UMR 144, F-75005 Paris, France

October 22, 2020

Abstract

Chimeric antigen receptor is an immunotherapy whereby T lymphocytes are engineered to selectively attack cancer cells. Image-based screens of CAR-T cells, combining phase contrast and fluorescence microscopy, suffer from the gradual quenching of the fluorescent signal, making the reliable monitoring of cell populations across time-lapse imagery difficult. We propose to leverage the available fluorescent markers as an experimentally-generated ground truth, without recourse to manual annotation. With some simple image processing, we are able to segment and assign cell type classes automatically. This ground truth is sufficient to train a neural object detection system from the phase contrast signal alone, potentially eliminating the need for the cumbersome fluorescent markers. This approach will underpin the development of cheap and robust microscope-based protocols to quantify CAR-T activity against tumor cells in vitro.

Keywords— H igh Content Screening, deep learning, object detection, phase contrast microscopy

1 Introduction

Therapies based on chimeric antigen receptor (CAR) show promise for improving the prognosis of cancers such as acute lymphoblastic leukemia, the most

common and fatal form of pediatric cancer in the United States[1]. In a microscope setting where both transmitted light and fluorescence microscopy can be taken for the same cells simultaneously, interesting opportunities arise. Recently, successful attempts have been made ([2], [3]) to predict fluorescent signals from transmitted light images, demonstrating that for certain biological experiments, the relevant information is wholly contained in the phase contrast signal. This is an attractive prospect because fluorescence microscopy, despite its power, has various drawbacks, with several experimental complications (such as fading dyes), in particular when imaging assays are performed over several days. In addition, the fluorescent marking of cells is expensive, time-consuming, and potentially invasive to the experiment.

However, predicting fluorescence is only goes partway towards quantifying the contents of the image. In our context, we would like to derive a quantitative profile from live cell imaging data. Here we present a setup that allows us to leverage fluorescence in order to automatically train a neural object detection system without any manual annotation. Annotation by experiment is a promising strategy: we can easily collect a large ground truth and, in addition, the "experimental ground truth" is much more objective than a manual one. A similar strategy has already been applied to image segmentation [4].

In Section 2 we describe an acquisition and preprocessing pipeline for an experimentally-generated object detection ground truth. In Section 3 we specify our object detection system. In Section 4 we describe our evaluation methodology and report model performance on two manually annotated datasets.

2 Experimentally-Generated Ground Truth

We obtained a set of time-lapse microscopy images from CAR-T experiments performed on an IncuCyte machine. Images of cell populations in microplate wells were taken every two hours over a three day period. In this paper we consider the most basic setting, consisting only of cells from the Raji cell line, a cancer cell line originally derived from human B cells. Note that our methodology should naturally extend to other experimental settings, such as those involving T cells, which we intend to address in future work. Our learning task therefore comprises two cell phenotypes: living and dead. The image frames consist of a

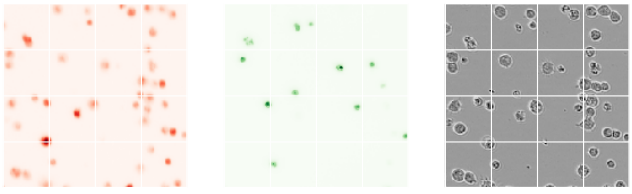


Figure 1: Aligned image channel crops (200×200px) marking living Raji cells in mCherry (left), dead cells in GFP (center), and phase contrast (right).

phase contrast image and corresponding mCherry and green fluorescent protein (GFP) overlays as shown in Figure 1. The mCherry fluorescence is present in all Raji cells while the GFP only appears in dead cells. Our ultimate objective is to track the respective cell phenotype numbers over time.

The fluorescent markers of our image set provide a quasi-ground truth annotation for the corresponding phase contrast image. Such a provision in principle obviates the need of laborious manual annotation, except for evaluation purposes.

Our pipeline begins by applying a Gaussian filter (tuned to $\sigma = 2$) to the phase contrast image. We then segment cells by subtracting a background image formed with a mean filter of diameter 19px, before clipping to zero as in [5]. We fill object holes with a morphological reconstruction by erosion and use a morphological opening to remove small details. We further filter objects outside a reasonable size range ($< 6 \times 6\text{px} \approx 50\mu m^2$, determined by ranking cell areas), as these tend to be dust and other non-cellular particles on the well surface. An Otsu threshold on the distribution of averaged GFP signal per cell is then used to allocate a class (living/dead) for each connected component. To train our object detector (Section 3), we also randomly sample background crops from the images, allowing for partial overlaps with cells.

3 Object detection system

In order to track cell phenotype populations over time, we require a robust object detection system to identify individual cells. The core of our system is a convolutional neural network and is detailed below.

3.1 Training as a classifier

Our preprocessing pipeline is imperfect and does not give a complete annotation of the cell populations as would be required by state-of-the-art detection systems such as [6]. We therefore opt for crop-wise training, where the bounding boxes of successfully segmented cells are padded, to create 24×24px crops, centered on the cells. Due to the low image resolution, we found this sizing provided sufficient contextual information to the network. Combined with background crops, this amounts to approximately 100,000 training examples in three classes. Samples are given in Figure 2.

Our network architecture is detailed in Table 1. All weighted layers have a ReLU activation, except Output_o and Output_c, which have softmax activations, and Output_b, which remains linear. The convolutions are all valid, and a 24×24px input image is reduced to 1×1px by the final layer. We implement this network in the `Keras` deep learning framework[7] and all code for our system is publicly available[1]. We train the network with stochastic gradient descent with learning rate 5×10^{-3} and Nesterov momentum ($\mu = 0.9$). Mini-batches of size 128 are sampled stochastically and simple data augmentation (horizontal and

[1] `https://github.com/jcboyd/detecting-lymphocytes`

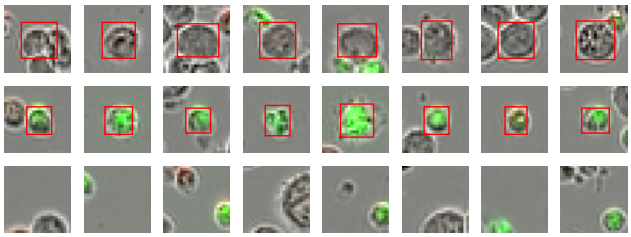


Figure 2: Samples of living Raji cells (top), dead cells (middle), background (bottom) annotated with bounding boxes. Fluorescence is included for clarity only and is not used in training.

vertical flipping) is performed on the fly. We regularise the network with batch normalisation [8] and weight decay ($\lambda = 3 \times 10^{-5}$).

Layer	Connection	Size	Output ($w \times h \times d$)
Input	-	-	$24 \times 24 \times 1$
Conv_1	Input	3×3	$22 \times 22 \times 16$
Conv_2	Conv_1	3×3	$20 \times 20 \times 16$
MaxPool_1	Conv_2	2×2	$10 \times 10 \times 16$
Conv_3	MaxPool_1	3×3	$8 \times 8 \times 64$
Conv_4	Conv_3	3×3	$6 \times 6 \times 64$
MaxPool_2	Conv_4	2×2	$3 \times 3 \times 64$
Conv_5	MaxPool_2	1×1	$1 \times 1 \times 128$
Conv_6	Conv_5	1×1	$1 \times 1 \times 128$
Output_o	Conv_6	1×1	$1 \times 1 \times 1$
Output_c	Conv_6	1×1	$1 \times 1 \times 1$
Output_b	Conv_6	1×1	$1 \times 1 \times 2$

Table 1: Specification of the network architecture. We distinguish three multi-task outputs.

Inspired by [6], we formulate a multi-task prediction in which we predict $Pr(o)$, where o indicates the presence of an object in the center of the receptive field and, separately, $Pr(c|o)$, that is, the probability of cell phenotype class c given the presence of an object. These probabilities are combined at inference time (Section 3.2). In addition, our network performs regression on the height h and width w of the bounding box of the cell, measured as a fraction of the crop size from the crop centre. This information is readily available when generating the training set. Note that we make the assumption that our chosen crop size represents a hard maximum on the size of a cell's bounding box, a reasonable simplification for our dataset. Our network is therefore trained to minimise the loss function,

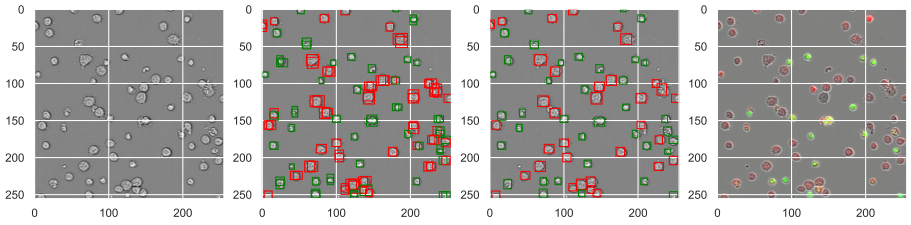


Figure 3: The processing of a test image. From left to right: phase contrast input image; raw model bounding box predictions; non-maximum suppression post-processing; finally, for comparison, the corresponding full fluorescence image.

$$\mathcal{L}(\mathbf{x}_i, o_i, c_i, w_i, h_i; \theta) = l_o(\hat{o}_i, o_i) + \mathbb{1}_o^i\big[l_c(\hat{c}_i, c_i)\big] + \mathbb{1}_o^i\big[l_b(\hat{w}_i, w_i, \hat{h}_i, h_i)\big] \tag{1}$$

with respect to model parameters θ, where l_o and l_c are each a standard cross entropy and $l_b(\hat{w}_i, w_i, \hat{h}_i, h_i) = (\hat{w}_i - w_i)^2 + (\hat{h}_i - h_i)^2$. The estimates $\hat{o}_i, \hat{c}_i$ $\hat{w}_i$, and $\hat{h}_i$ are the network outputs for object presence, object class, and bounding box width and height. The indicator function $\mathbb{1}_o^i = 1$ when training example $\mathbf{x}_i$ contains an object and $\mathbb{1}_o^i = 0$ otherwise.

We benchmarked our network as a classifier of cropped cells against a logistic regression trained on features extracted from a pre-trained 50-layer ResNet[9]. In order to do this, we resized our cropped cells to 32×32px and recorded the final convolutional layer of the ResNet, a vector of dimension 2048. This baseline achieved an accuracy of 0.83 on balanced test data, whereas our own network achieved 0.96. Though deep pre-trained networks are known to be powerful general-purpose feature extractors[10], they may also be over-parameterised for many problems[11].

3.2 Inference as a detector

Following [12] we designed our network to be *fully convolutional* (FCN). A FCN is capable of performing inference on any size of input, and is extended naturally to object detection. Thus, once trained on cell crops as a classifier, inference may be performed on an entire image in a single forward pass, producing a map of softmax probabilities at every location in the image. Note that even on CPU, a full 1408×1040px image is processed by the network in about 1s. Fully-convolutional whole-image inference emulates sliding-window detection, albeit without the tremendous inefficiency of executing the model separately at every spatial position. Note that the resolution of the output will depend on the number of pooling layers in the network. For example, our network includes two max pooling layers, hence we make detections at a stride of 4 across the input image domain.

At inference time, the object and conditional class probabilities are combined to give the marginal class probabilities $Pr(c) = Pr(c|o) \cdot Pr(o)$. Note that $Pr(c|\neg o) = 0$. These probabilities are thresholded and pruned with non-maximum suppression (NMS), providing a final detection mask for each class. For the NMS algorithm, we use an intersection over union threshold of 0.35. An example of this procedure is shown in Figure 3.

3.3 Smoothing probabilities in time

Because the cells are relatively stationary, we can improve the prediction of our system by leveraging information across time. We find a simple weighted average of prediction probabilities from consecutive frames, computed prior to NMS, improves overall performance. We thus define the *smoothed* probability $p_{ij}^{(t)} \leftarrow 1/4 \cdot p_{ij}^{(t-1)} + 1/2 \cdot p_{ij}^{(t)} + 1/4 \cdot p_{ij}^{(t+1)}$ for the probability at image position (i, j) at time t. The weights were tuned manually for both performance and parsimony.

4 Results

4.1 Evaluation strategy

To evaluate our system, we manually annotated three days worth of frames of size 256×256px from each of two independent experimental replicates, totaling 72 images and approximately 7,000 test object detections. The replicates were chosen to represent different population dynamics: the first exhibits higher levels of cell mitosis; the second exhibits higher levels of cell apoptosis. We henceforth refer to these two datasets as Mitosis and Apoptosis respectively. The annotations consist of manually annotated bounding boxes around the cells. We make this dataset publicly available along with the images used to train the network[2]. Note that despite this manually annotated evaluation dataset, our model is still trained on a ground truth that is automatically generated from the experiment.

We score our detections in terms of the distance of the bounding box centers to the ground truth bounding box centers. We define the following metrics:

- True positive (TP) - a cell is detected in the vicinity of a ground truth cell.
- False positive (FP) - a cell is detected outside the vicinity of any ground truth cell.
- False negative (FN) - no cell is detected within the vicinity of a ground truth cell.

Here we define vicinity to be ≤ 10px, the maximum distance a predicted cell center may fall from a ground true center while still falling within the typical

[2] `https://zenodo.org/record/3515446`

cell bounding box (14×14px). These metrics are computed per cell class, from which we calculate precision, recall, and F_1 scores. Note the F_1 score prevails over the commonly used Matthews correlation coefficient as it does not require us to define true negatives (a meaningless quantity in our framework). These are displayed in Tables 2 and 3 for the Mitosis and Apoptosis test sets. We see the effect of smoothing is globally positive, significantly improving the precision of the dead cell class, and giving the highest average F_1 scores of 83.86 for Mitosis and 81.19 for Apoptosis. Note that the results on the dead cell class are markedly worse. We postulate this is due to the class imbalance at test time, as well as the difficulty of discerning individual cells from cell clusters.

Method	Class	Precision	Recall	F_1
Without smoothing	Living	0.8534	0.8636	0.8585
	Dead	0.7179	0.8693	0.7864
With smoothing	Living	0.8466	0.8883	**0.8669**
	Dead	0.7702	0.8549	**0.8103**

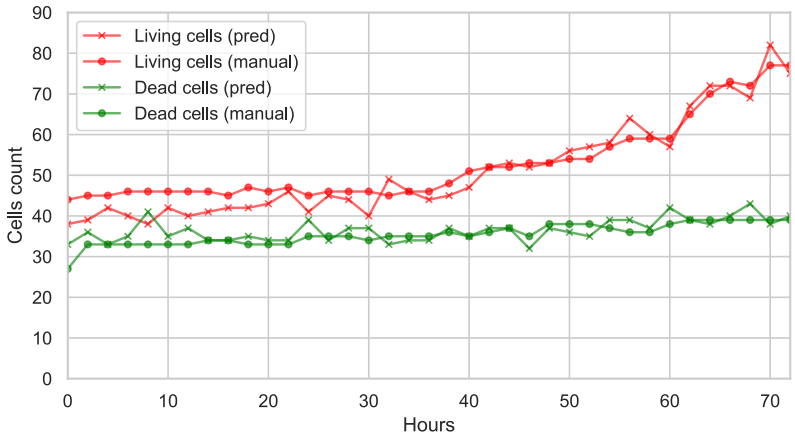
Table 2: Detection performance on the Mitosis test set, stratified by object class. Best results in bold.

Method	Class	Precision	Recall	F_1
Without smoothing	Living	0.9451	0.7778	0.8533
	Dead	0.6253	0.8935	0.7357
With smoothing	Living	0.9447	0.7957	**0.8638**
	Dead	0.6628	0.8904	**0.7600**

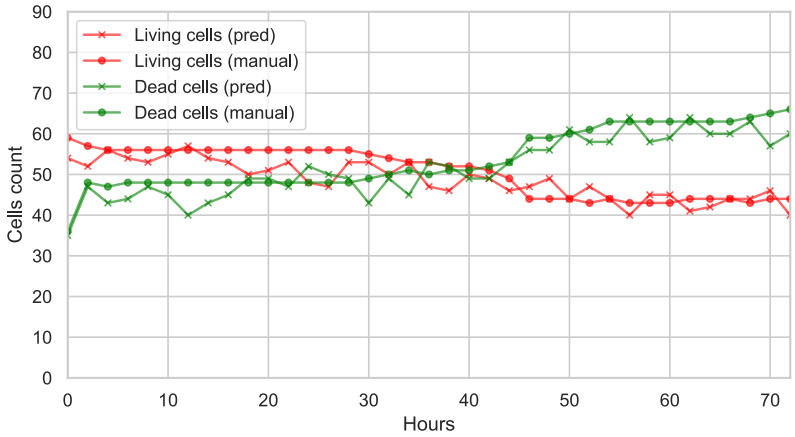
Table 3: Detection performance on the Apoptosis test set, stratified by object class. Best results in bold.

4.2 Tracking population numbers over time

Our detection system is ultimately used to enumerate cell phenotypes over the course of CAR-T experiments. In Figure 4 we plot ground truth population numbers against the numbers inferred by our system. One can see the increasing number of living cells in Figure 4a, corresponding to increasing amounts of cell division, whereas in Figure 4b, one can see increasing amounts of apoptosis. In the former, our system achieves a mean relative error percentage of 5.95% and 5.56% (resp. living and dead cells) and 5.81% and 5.37% in the latter.



(a)



(b)

Figure 4: Population curves for manually-annotated test sets Mitosis (a) and Apoptosis (b), compared with system outputs.

5 Perspectives

In this paper we have shown the viability of predicting phenotypes in the absence of fluorescence, as well as how fluorescence may be used to generate a robust ground truth for machine learning. We have trained a neural object detection system and tested it on two manually annotated datasets. We have also given an example of how time information can be incorporated into the prediction task. We feel the system can be further improved with a more precise and expanded dataset, something we intend to address in future work.

Acknowledgement

This work was funded in part by the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute). J. Boyd has a PhD fellowship granted by PSL Research University.

References

[1] Stephen P Hunger and Charles G Mullighan, "Acute lymphoblastic leukemia in children," *New England Journal of Medicine*, vol. 373, no. 16, pp. 1541–1552, 2015.

[2] Eric M Christiansen, Samuel J Yang, D Michael Ando, Ashkan Javaherian, Gaia Skibinski, Scott Lipnick, Elliot Mount, Alison O'Neil, Kevan Shah, Alicia K Lee, et al., "In silico labeling: Predicting fluorescent labels in unlabeled images," *Cell*, vol. 173, no. 3, pp. 792–803, 2018.

[3] Chawin Ounkomol, Sharmishtaa Seshamani, Mary M Maleckar, Forrest Collman, and Gregory R Johnson, "Label-free prediction of three-dimensional fluorescence images from transmitted-light microscopy," *Nature methods*, vol. 15, no. 11, pp. 917, 2018.

[4] Sajith Kecheril Sadanandan, Petter Ranefall, Sylvie Le Guyader, and Carolina Wählby, "Automated training of deep convolutional neural networks for cell segmentation," *Scientific reports*, vol. 7, no. 1, pp. 7860, 2017.

[5] Thomas Walter, Michael Held, Beate Neumann, Jean-Karim Hériché, Christian Conrad, Rainer Pepperkok, and Jan Ellenberg, "Automatic identification and clustering of chromosome phenotypes in a genome wide rnai screen by time-lapse imaging," *Journal of structural biology*, vol. 170, no. 1, pp. 1–9, 2010.

[6] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[7] François Chollet et al., "Keras," https://keras.io, 2015.

[8] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[10] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2014, pp. 806–813.

[11] Maithra Raghu, Chiyuan Zhang, Jon Kleinberg, and Samy Bengio, "Transfusion: Understanding transfer learning with applications to medical imaging," *arXiv preprint arXiv:1902.07208*, 2019.

[12] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.