Bioinformatics, 32, 2016, i52–i59
doi: 10.1093/bioinformatics/btw252
ISMB 2016

Classifying and segmenting microscopy images with deep multiple instance learning

Oren Z. Kraus,[1, 2,*] Jimmy Lei Ba[1] and Brendan J. Frey[1,2]

[1]Department of Electrical and Computer Engineering, University of Toronto, Toronto, M5S 2E4, Canada and [2]The Donnelly Centre for Cellular and Biomolecular Research, University of Toronto, Toronto, M5S 3E1, Canada

*To whom correspondence should be addressed.

Abstract

Motivation: High-content screening (HCS) technologies have enabled large scale imaging experiments for studying cell biology and for drug screening. These systems produce hundreds of thousands of microscopy images per day and their utility depends on automated image analysis. Recently, deep learning approaches that learn feature representations directly from pixel intensity values have dominated object recognition challenges. These tasks typically have a single centered object per image and existing models are not directly applicable to microscopy datasets. Here we develop an approach that combines deep convolutional neural networks (CNNs) with multiple instance learning (MIL) in order to classify and segment microscopy images using only whole image level annotations.
Results: We introduce a new neural network architecture that uses MIL to simultaneously classify and segment microscopy images with populations of cells. We base our approach on the similarity between the aggregation function used in MIL and pooling layers used in CNNs. To facilitate aggregating across large numbers of instances in CNN feature maps we present the Noisy-AND pooling function, a new MIL operator that is robust to outliers. Combining CNNs with MIL enables training CNNs using whole microscopy images with image level labels. We show that training end-to-end MIL CNNs outperforms several previous methods on both mammalian and yeast datasets without requiring any segmentation steps.
Availability and implementation: Torch7 implementation available upon request.
Contact: oren.kraus@mail.utoronto.ca

1 Introduction

High-content screening (HCS) technologies that combine automated fluorescence microscopy with high-throughput biotechnology have become powerful systems for studying cell biology and for drug screening (Liberali *et al.*, 2015; Singh *et al.*, 2014). These systems can produce more than 10^5 images per day, making their success dependent on automated image analysis. Previous analysis pipelines heavily rely on hand-tuning the segmentation, feature extraction and classification steps for each assay. Although comprehensive tools have become available (Carpenter *et al.*, 2006; Eliceiri *et al.*, 2012; Held *et al.*, 2010) they are typically optimized for mammalian cells and not directly applicable to model organisms such as yeast and *Caenorhabditis elegans*. Researchers studying these organisms often manually classify cellular patterns by eye (Breker *et al.*, 2013; Tkach *et al.*, 2012).

Recent advances in deep learning have proven that deep neural networks trained end-to-end can learn powerful feature representations and outperform classifiers built on top of extracted features (Krizhevsky *et al.*, 2012; Vincent *et al.*, 2010). Although object recognition models have been successfully trained using images with one or a few objects of interest at the center of the image, microscopy images often contain hundreds of cells with a phenotype of interest, as well as outliers. Training similar recognition models on HCS screens is challenging due to the lack of datasets labeled at the single cell level.

In this work, we describe a convolutional neural network (CNN) that is trained on full resolution microscopy images using multiple instance learning (MIL). The network is designed to produce feature maps for every output category, as proposed for segmentation tasks in Long *et al.* (2014). We pose cellular phenotype classification as a MIL problem in which each element in a class-specific feature map (approximately representing the area of a single cell in the input space) is considered an instance an entire class specific feature map (representing the area of the entire image) is considered a bag of instances annotated with the whole image label. Typically binary MIL problems assume that a bag is positive if at least one instance within the bag is positive. This assumption does not hold for HCS images due to heterogeneities within cellular populations and imaging artifacts (Altschuler and Wu, 2010). We explore the performance of several global pooling operators on this problem and propose a new operator capable of learning the proportion of instances necessary to activate a label.

The main contributions of our work are the following. We present a unified view of the classical MIL approaches as pooling layers in CNNs and compare their performances. To facilitate MIL aggregation in CNN feature maps we propose a novel MIL method, 'Noisy-AND', that is robust to outliers and large numbers of instances. We demonstrate the utility of convolutional MIL models on an interpretable dataset of cluttered hand written digits. We evaluate our proposed model on both mammalian and yeast datasets, and find that our model significantly outperforms previously published results at phenotype classification. We show that our model is capable of learning a good classifier for full resolution microscopy images as well as individual cropped cell instances, even though it is only trained using whole image labels. Finally, we demonstrate that the model can localize regions with cells in the full resolution microscopy images and that the model predictions are based on activations from these regions.

1.1 Related work

1.1.1 Current approaches for microscopy image analysis

Several sophisticated and modular tools (Eliceiri *et al.*, 2012) have been developed for analyzing microscopy images. *CellProfiler* (Carpenter *et al.*, 2006) is a popular tool that was previously used to analyze the datasets described below. All existing tools rely on extracting a large set of predefined features from the original images and subsequently selecting features that are relevant for the learning task Kraus *et al.*, 2016. This approach can be limiting for assays that differ from the datasets used to develop these tools. For example, recent proteome-wide studies of protein localization in yeast resorted to evaluating images manually (Breker *et al.*, 2013; Tkach *et al.*, 2012). Also, a separate toolbox has been published for *CellProfiler* specifically for analyzing *C. Elegan* images (Wählby *et al.*, 2012).

Applying deep neural networks to microscopy screens has been challenging due to the lack of large datasets labeled at the single cell level. Other groups have applied deep neural networks to microscopy for segmentation tasks (Ciresan *et al.*, 2012; Ning *et al.*, 2005) using ground truth pixel-level labels. Pachitariu *et al.* (2013) use convolutional sparse coding blocks to extract regions of interest from spiking neurons and slices of cortical tissue without supervision. These publications differ from our work as they aim to segment or localize regions of interest within the full resolution images. Here we aim to train a CNN for classifying cellular phenotypes for images of arbitrary size based on only training with weak labels.

1.1.2 Fully convolutional neural networks

Fully CNNs (FCNNs) have recently achieved state-of-the-art performance on image segmentation tasks (Chen *et al.*, 2014; Long *et al.*, 2015). These networks build on the success of networks previously trained on image recognition tasks (Krizhevsky *et al.*, 2012; Simonyan and Zisserman, 2014; Szegedy *et al.*, 2014) by converting their fully connected layers to 1×1 convolutions, producing feature maps for each output category instead of a single prediction vector. The pretrained networks are fine-tuned using different techniques to generate output images of the same dimension as input images from the down-sampled feature maps. These networks are trained with pixel level ground truth labels. Pathak *et al.* (2014) use MIL with a FCNN to perform segmentation using weak labels. However, dense pixel level ground truth labels are expensive to generate and arbitrary, especially for niche datasets such as microscopy images. In this work we aim to develop a classification CNN using MIL that does not require labels for specific segmented cells, or even require the cells to be segmented.

1.1.3 Multiple instance learning

MIL deals with problems for which labels only exist for sets of data points. In this setting sets of data points are typically referred to as bags and specific data points are referred to as instances. A commonly used assumption for binary labels is that a bag is considered positive if at least one instance within the bag is positive (Dietterich *et al.*, 1997). Several functions have been used to map the instance space to the bag space. These include Noisy-OR (Zhang *et al.*, 2005), log-sum-exponention (LSE) (Ramon and De Raedt, 2000), generalized mean (GM) and the integrated segmentation and recognition (ISR) model (Keeler *et al.*, 1991). Xu *et al.* (2014) use the GM pooling function for classifying features extracted from histopathology breast cancer images.

2 methods

2.1 Convolutional MIL model for learning cellular patterns

We propose a CNN capable of classifying microscopy images of arbitrary size that is trained with only global image level labels. The weakly supervised CNN is designed to output class-specific feature maps representing the probabilities of the classes for different locations in the input image. The CNN produces an image level classification over images of arbitrary size and varying number of cells through a MIL pooling layer. Individual cells can be classified by passing segmented cells through the trained CNN or by mapping the probabilities in class specific feature maps back to the input space.

2.2 Pooling layers as MIL

Formally, assuming that the total number of classes is N_{class} for a full resolution image I, we can treat each class i as a separate binary classification problem with label $t_i \in \{0, 1\}$. Under the MIL formulation, one is given a bag of N instances that are denoted as $x = \{x_1, \cdots, x_N\}$ and $x_n \in \mathrm{R}^{\mathrm{D}}$ is the feature vector for each instance. The class labels t_i are associated with the entire bag instead of each instance. A binary instance classifier $p(t_i = 1|x_j)$ is used to generate predictions p_{ij} across the instances in a bag. The instance predictions $\{p_{ij}\}$ are combined through an aggregate function $g(\cdot)$, e.g. noisy-OR, to map the set of instance predictions to the probability of the final bag label $p(t_i = 1|x_1, \cdots, x_N)$. In a CNN, each activation in the feature map is computed through the same set of filter weights convolved across the input image. The pooling layers then combine activations of feature maps in convolutional layers. It is easy to see the similarity between the pooling layer and the MIL aggregation function, where features in convolutional layers correspond to instance features $\{x_n\}$ in MIL. In fact, if class specific feature maps are treated as bags of instances, the classical approaches in MIL can be generalized to global pooling layers over these feature maps.

We formulate the MIL layer in CNNs as a global pooling layer over a class specific feature map for class i referred to as the bag p_i. Without loss of generality assume that the i^{th} class specific convolutional layer in a CNN computes a mapping directly from input images to sets of binary instance predictions $I \rightarrow \{p_{i1}, \cdots, p_{iN}\}$. It first outputs the logit values z_{ij} in the feature map corresponding to instance j in the bag i. We define the feature level probability of an instance j belonging to class i as p_{ij} where $p_{ij} = \sigma(z_{ij})$ and σ is the sigmoid function. The image level class prediction is obtained by applying the global pooling function $g(\cdot)$ over all elements p_{ij}. The global pooling

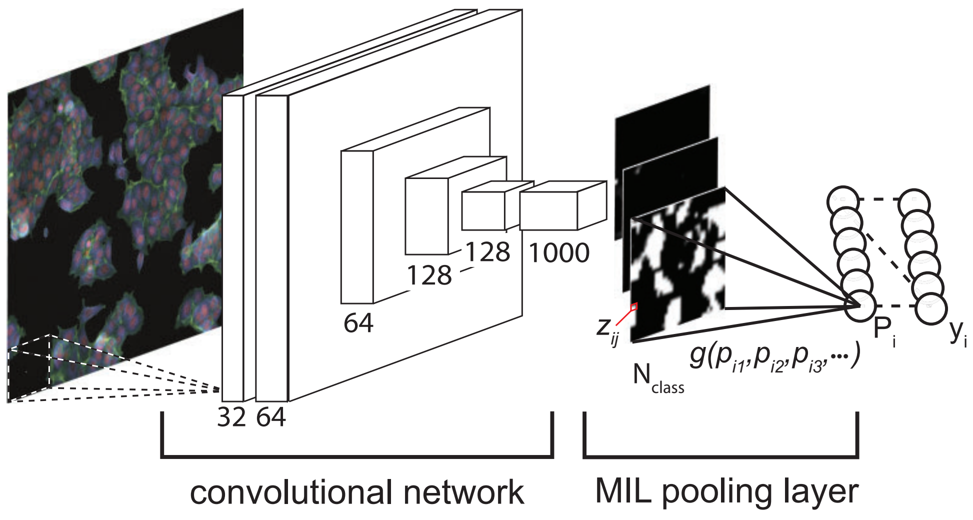

Fig. 1. Convolutional MIL model. $g(\cdot)$ is global pooling function that aggregates instance probabilities p_{ij}

function $g(\cdot)$ maps the instance space probabilities to the bag space such that the bag level probability for class i is defined by

$$P_i = g(p_{i1}, p_{i2}, p_{i3}, \cdots) \tag{1}$$

The global pooling function $g(\cdot)$ essentially combines the instance probabilities from each class specific feature map p_i into a single probability. This reduction allows us to train and evaluate the model on inputs of arbitrary size. In the next section, we describe the global pooling functions we explored in our experiments.

While the MIL layer learns the relationship between instances of the same class, the co-occurrence statistics of instances from different classes within the bag could also be informative for predicting the bag label. We extend our model to learn relationships between classes by adding an additional fully connected layer following the MIL pooling layer. This layer can either use softmax or sigmoid activations for either multi-class or multi-label problems. We define the softmax output from this layer for each class i as y_i. We formulate a joint cross entropy objective function at both the MIL pooling layer and the additional fully connected layer defined by

$$J = -\sum_{i=1}^{N_{\text{class}}} (\log p(t_i|P_i) + \log p(t_i|y_i)). \tag{2}$$

$p(t_i|P_i)$ is the binary class prediction from the MIL layer, $p(t_i|P_i) = P_i^{t_i}(1-P_i)^{(1-t_i)}$ and $p(t_i|y_i)$ is either the binary or the multi-class prediction from the fully connected layer. Our proposed MIL CNN model is shown in Figure 1 and is trained using standard error backpropgation.

2.3 Global pooling functions

Classifying cellular phenotypes in microscopy images presents a challenging and generalized MIL problem. Due to heterogeneity within cellular populations (Altschuler and Wu, 2010), imaging artifacts, and the large number of potential instances in an image, it cannot be assumed that images with a negative label do not contain any instances of the specific phenotype. A more reasonable assumption is that bag labels are determined by a certain proportion of instances being present. Relevant generalizations for MIL have been proposed that assume that all instances collectively contribute to the bag label. Here we take an approach similar to Xu and Frank (2004) in which bag predictions are expressed as the geometric or arithmetic mean of instances, however we adapt the the bag level formulation to model thresholds on instance proportions for different categories.

We explore the use of several different global pooling functions $g(\cdot)$ in our model. Let j index the instance within a bag. Previously proposed global pooling functions for MIL have been designed as differentiable approximations to the max function in order to satisfy the standard MIL assumption:

$$g(\{p_j\}) = 1 - \prod_j (1-p_j) \quad \text{Noisy-or,}$$

$$g(\{p_j\}) = \sum_j \frac{p_j}{1-p_j} \Big/ \left(1 + \sum_j \frac{p_j}{1-p_j}\right) \quad \text{ISR,}$$

$$g(\{p_j\}) = \left(\frac{1}{|j|}\sum_j p_j^r\right)^{\frac{1}{r}} \quad \text{Generalized mean,}$$

$$g(\{p_j\}) = \frac{1}{r}\log\left(\frac{1}{|j|}\sum_j e^{r\cdot p_j}\right) \quad \text{LSE.}$$

We initially attempted to include Noisy-OR (Zhang *et al.*, 2005) and ISR (Keeler *et al.*, 1991) in our analysis. We found that both are sensitive to outliers and failed to work with microscopy datasets (as shown in Fig. 2). LSE and GM both have a parameter r that controls their sharpness. As r increases the functions get closer to representing the max of the instances. In our analysis we use lower values for r than suggested in previous work (Ramon and De Raedt, 2000) to allow more instances in the feature maps to contribute to the pooled value.

2.3.1 Noisy-and pooling function

Since existing pooling functions are ill-suited for the task at hand, we developed an alternative pooling function. Formally, we assume that a bag is positive if the number of positive instances in the bag surpasses a certain threshold. This assumption is meant to model the case of a human expert annotating images of cells by classifying them according to the evident phenotypes or drugs with known targets affecting the majority of the cells in an image. We define the Noisy-AND pooling function as follows,

$$P_i = g_i(\{p_{ij}\}) = \frac{\sigma(a(p_{i\bar{j}} - b_i)) - \sigma(-ab_i)}{\sigma(a(1-b_i)) - \sigma(-ab_i)}, \tag{3}$$

Where $\quad p_{i\bar{j}} = \frac{1}{|j|}\sum_j p_{ij}$

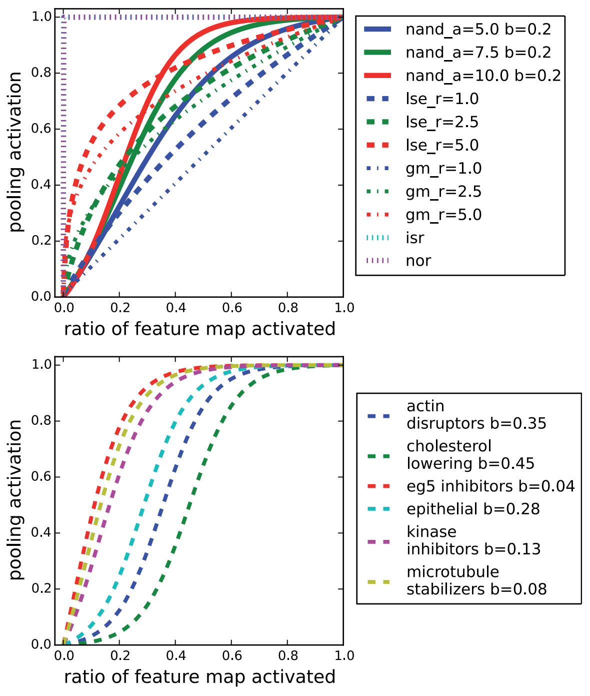

Fig. 2. MIL Pooling functions. Top, pooling function activations by ratio of feature map activated (p_{ij}). Bottom, activation functions learned by Noisy-AND a_{10} (nand_a = 10.0) for different classes of the breast cancer dataset

The function is designed to activate a bag level probability P_i once the mean of the instance level probabilities $p_{\bar{ij}}$ surpasses a certain threshold. This behaviour mimics the logical AND function in the probabilistic domain and therefore we named the pooling function Noisy-AND. The parameters a and b_i control the shape of the activation function. b_i is a set of parameters learned during training and is meant to represent an adaptable soft threshold for each class i. a is fixed parameter that controls the slope of the activation function. The terms $\sigma(-ab_i)$ and $\sigma(a(1-b_i))$ are included to normalized P_i to [0,1] for b_i in [0,1] and $a > 0$. Figure 2 shows plots of relevant pooling functions.

2.4 Localizing cells with Jacobian maps

Researchers conducting HCS experiments are often interested in obtaining statistics from single cell measurements of their screens. We aimed to extend our model by localizing regions of the full resolution input images that are responsible for activating the class specific feature maps. We employ recently developed methods for visualizing network activations (Zeiler and Fergus, 2014; Simonyan *et al.*, 2013) toward this purpose. Our approach is similar to Simonyan *et al.* (2013) in which the pre-softmax activations of specific output nodes are back-propagated through a classification network to generate Jacobian maps w.r.t. specific class predictions. Let $a^{(l)}$ be the hidden activations in layer l and $z^{(l)}$ be pre-nonlinearity activations. We define a general recursive non-linear back-propagation process computing a backward activation $\overleftarrow{a}$ for each layer, analogous to the forward propagation:

$$\overleftarrow{a}^{(l-1)} = f\left(\frac{\partial z^{(l)}}{\partial z^{(l-1)}}\overleftarrow{a}^{(l)}\right) \quad (4)$$

Where $f(x) = \max(0, x), \overleftarrow{a}^{L}_{ij} = P_i \cdot p_{ij}$

In our case, we start the non-linear back-propagation (a^{L}_{ij}) from the MIL layer using its sigmoidal activations for the class i specific feature maps $\{p_{ij}\}$ multiplied by the pooling activation for each class $P_i \cdot p_{ij}$. Similar to Springenberg *et al.* (2014), we find that applying the ReLU activation function to the partial derivatives during back propagation generates Jacobian maps that are sharper and more localized to relevant objects in the input. To generate segmentation masks we threshold the sum of the Jacobian maps along the input channels. To improve the localization of cellular regions we use loopy belief propagation (Frey, 1998) in an MRF to de-noise the thresholded Jacobian maps.

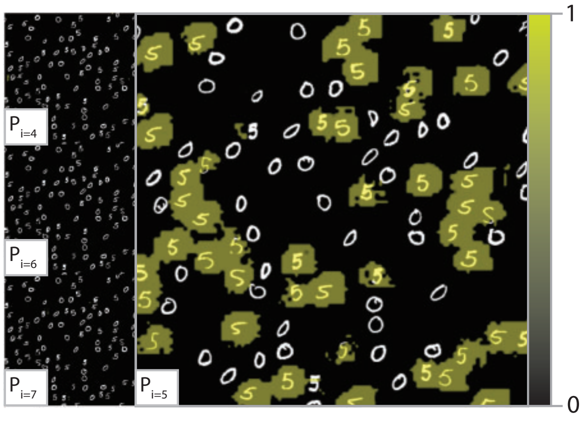

Fig. 3. Class feature map probabilities for hand written digit sample. Class specific feature map probabilities (P_i) overlaid on a sample from the cluttered hand written digit dataset labelled as five. The model successfully classifies regions with fives, and is not sensitive to the background or distractors

3 Results

3.1 Datasets

3.1.1 Cluttered hand written digits

We generated an interpretable dataset of images containing populations of digits from the MNIST hand written digit dataset (LeCun *et al.*, 1998) in order to demonstrate the effectiveness of the convolutional MIL models. Each image in the dataset contains 100 digits cluttered on a black background of 512 × 512 pixels. The dataset contains nine categories (digits $\in \{1, 2, , 9\}$) and zeros are used as distractors. To simulate the conditions in cell culture microscopy, among the 100 digits x samples are chosen from a single category and the remaining 100-x samples are zeros. x is fixed for each category is equal to 10 times the digit value of the chosen category, as shown in Figure 3. For example, an image with label four contains 40 fours and 60 zeros. We used 50 images per category for training and 10 images per category for testing.

3.1.2 Breast cancer screen

We used a benchmarking dataset of MFC-7 breast cancer cells available from the Broad Bioimage Benchmark Collection (image set BBBC021v1) (Ljosa *et al.*, 2012). The images contain three channels with fluorescent markers for DNA, actin filaments, and β-tubulin at a resolution of 1024 × 1280 (Fig. 4). Within this dataset 103 treatments (compounds at active concentrations) have known effects on cells based on visual inspection and prior literature and can be classified into 12 distinct categories referred to as mechanism of action (MOA). We sampled 15% of images from these 103 treatments to train and validate our model. The same proportion of the data was used to train the best model reported in Ljosa *et al.* (2013). In total we used 300 whole microscopy images during training and 40 for

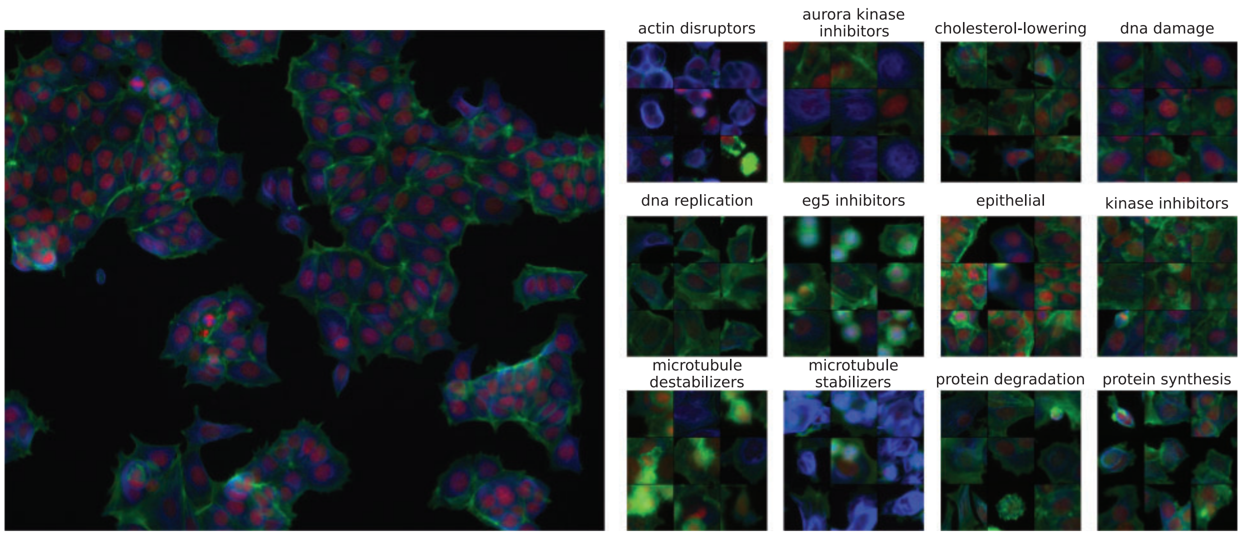

Fig. 4. Breast cancer screen. Left, sample full resolution image with epithelial MOA. Right, samples of segmented cells sampled from 12 MOA categories

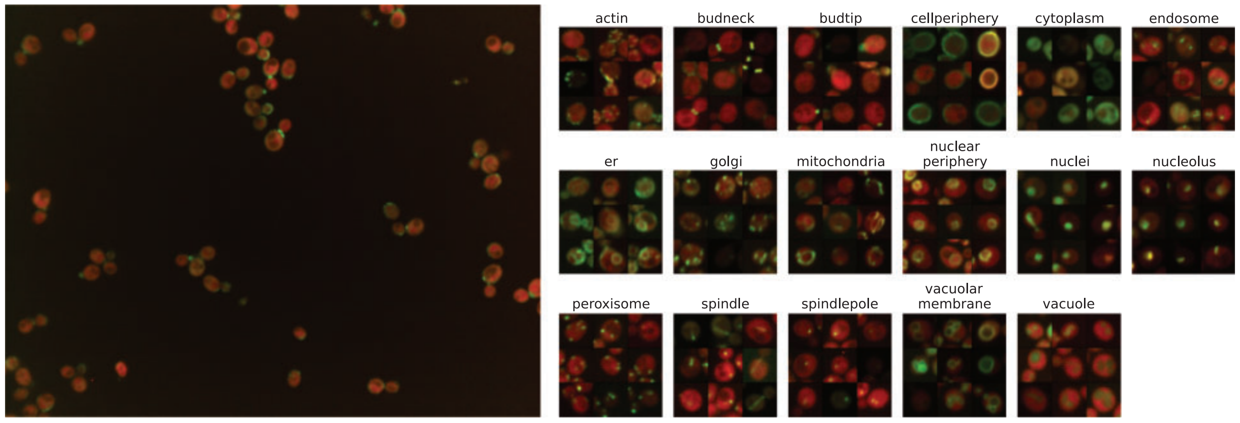

Fig. 5. Yeast protein localization screen. Left, sample full resolution image with budneck, budtip, cell periphery, and cytoplasm protein localizations. Right, segmented cells with protein localizations manually labelled in Chong *et al.* (2015)

testing. We evaluated all the images in the screen and report the predicted treatment accuracy across the treatments.

3.1.3 Yeast protein localization screen

We used a genome wide screen of protein localization in yeast (Chong *et al.*, 2015) containing images of 4144 yeast strains from the yeast GFP collection (Huh *et al.*, 2003) representing 71% of the yeast proteome. The images contain two channels, with fluorescent markers for the cytoplasm and a protein from the GFP collection at a resolution of 1010×1335 (Fig. 5). We sampled 6% of the screen and used 2200 whole microscopy images for training and 280 for testing. We categorized whole images of strains into 17 localization classes based on visually assigned localization annotations from a previous screen (Huh *et al.*, 2003). These labels include proteins that were annotated to localize to more than one sub-cellular compartment. We evaluated all the proteins in the screen and report the test error for the 998 proteins that are localized to a single compartment and mean average precision for the 2592 proteins analyzed in Chong *et al.* (2015).

3.2 Model architecture

We designed the CNN such that an input the size of a typical cropped single cell produces output feature maps of size 1×1. The same network can be convolved across larger images of arbitrary size to produce output feature maps representing probabilities of target labels for different locations in the input image. We also aimed to show that training such a CNN end-to-end allows the model to work on vastly different datasets. We trained the model separately on both datasets while keeping the architecture and number of parameters constant.

The basic CNN architecture includes the following layers: ave_pool0_3×3, conv1_3×3×32, conv2_3×3_64, pool1_3×3, conv3_5×5_64, pool2_3×3, conv4_3×3_128, pool3_3×3, conv5_3×3_128, pool4_3×3, conv6_1×1_1000, conv7_1×1_N_{class}, MIL_pool, FC_N_{class} (Fig. 1). To use this architecture for MIL we use a global pooling function $g(\cdot)$ as the activation function in the MIL_pool layer. $g(\cdot)$ transforms the output feature maps z_i into a vector with a single prediction P_i for each class i. We explore the pooling functions described above (Section 2.3). All of these pooling functions are defined for binary categories and we use them in a multi-label setting (where each output category has a separate binary target). To extend this framework we add an additional fully connected output layer to the MIL_pool layer in order to learn relations between different categories. For the breast cancer screen this layer uses softmax activation while for the yeast dataset this layer uses a sigmoidal activation (since proteins can be annotated to multiple localization categories).

3.3 Model training

We trained models with a learning rate of 10^{-3} using the Adam optimization algorithm (Kingma and Ba, 2014). We extracted slightly smaller crops of the original images to account for variability in image sizes within the screens (we used 1000×1200 for the breast cancer dataset and 1000×1300 for the yeast dataset). We normalized the images by subtracting the mean and dividing by the standard deviation of each channel in our training sets. During training we cropped random 900×900 patches from the full resolution images and applied random rotations and reflections to the patches. We use the ReLU activation for the convolutional layers and apply 20% dropout to the pooling layers and 50% dropout to layer conv6. We trained the models within 1–2 days on a Tesla K80 GPU using ~9 Gb of memory with a batch size of 16.

3.4 Model evaluation

The models we trained on the cluttered hand written digits achieve 0% test error across all classes. We achieve these error rates despite the fact images labelled as digit one actually contain 90 zeros and only 10 ones. The reason the model doesn't confuse zeros for ones in these samples is because zeros also appear in images labelled with other categories. Another important note is that since there are only 50 training samples per digit, the model only sees 500 distinct ones during training. The classic MNIST training dataset contains 6000 cropped and centered samples per category. We achieve the superior test performance with fewer training samples using the MIL formulation because the model's predictions are based on aggregating over multiple instances. The model can ignore samples that are difficult to classify but still rely on easier instances to predict the overall image correctly. Because we use different sampling rates for each digit category, this experiment also shows that the convolutional MIL models are robust to different frequencies of the label class being present in the input image. Finally, in Figure 3 we show class specific feature map activations (P_i) for a test sample labelled as five overlaid onto the input image. Here we see that the model successfully classifies almost all the fives in the image and is not sensitive to the background or distractors (i.e. zeros).

We evaluated the performance of models trained on each screening dataset at several tasks. For the yeast dataset (Table 1), proteins are annotated to localize to one or more sub-cellular compartments. We report the accuracy and mean classifier accuracy (across 17 classes) for a subset of 998 proteins annotated to localize to a single sub-cellular compartment in both Huh *et al.* (2003) and Chong *et al.* (2015). We also report the mean average precision for the all the proteins we analyzed from the screen (2592) and a test set of individual images. For the breast cancer dataset (Table 3) we report accuracy on a test set of full resolution images and at predicting the MOA of all the different treatments by taking the median prediction across the 3 experimental replicates of the screen. For these predictions we use the output from the last layer of the network.

Table 1. Yeast dataset results on whole images.

Model	Mean average prec.		Classification	
	full image	Huh	single loc acc.	single loc mean acc.
Chong *et al.* (2015)	—	0.703	0.935	0.808
Noisy-AND a_5	0.921	0.815	0.942	0.821
Noisy-AND $a_{7.5}$	0.920	0.846	0.963	0.834
Noisy-AND a_{10}	0.950	0.883	0.953	0.876
LSE r_1	0.925	0.817	0.945	0.828
LSE $r_{2.5}$	0.925	0.829	0.953	0.859
LSE r_5	0.933	0.861	0.960	0.832
GM r_1 (avg. pooling)	0.915	0.822	0.938	0.862
GM $r_{2.5}$	0.888	0.837	0.922	0.778
GM r_5	0.405	0.390	0.506	0.323
max pooling	0.125	0.133	0.346	0.083

Huh indicates agreement with manually assigned protein localizations (Huh *et al.*, 2003). Single loc acc. and single loc mean acc. indicate the accuracy and mean accuracy across all classes for a subset of proteins that localize to a single compartment. Full image indicates mean average precision on full resolution image.

Table 2. Yeast dataset results on segmented cells

Model	Mean average precision	
	Segmented cells with noisy labels	Segmented cells with manual labels
CNN trained on segmented cells with noisy labels	0.855	0.742
Noisy-AND a_5	0.701	0.750
Noisy-AND $a_{7.5}$	0.725	0.757
Noisy-AND a_{10}	0.701	0.738
LSE r_1	0.717	0.763
LSE $r_{2.5}$	0.715	0.762
LSE r_5	0.674	0.728
GM r_1 (avg. pooling)	0.705	0.741
GM $r_{2.5}$	0.629	0.691
GM r_5	0.255	0.258
max pooling	0.111	0.070

For the yeast dataset, we had access to cell coordinates from a previous segmentation pipeline. A subset of these cropped cells was labelled manually for Chong *et al.* (2015). We trained a traditional CNN on the segmented cells with noisy, whole image level labels and compared the performance on the manually labelled cropped cells. As an additional baseline, a traditional CNN trained on the manually labeled cells achieves a test accuracy of 89.8%. We evaluate our CNN-MIL models trained on whole images with whole image level labels on classifying segmented cells [with both protein level annotations based on Huh *et al.* (2003) and manual labels based on Chong *et al.* (2015)]. We report mean average precision across the localization classes.

Table 3. Breast cancer dataset results

Model	full image	treatment
Ljosa *et al.* (2013)	—	0.94
Noisy-AND a_5	0.915	0.957
Noisy-AND $a_{7.5}$	0.915	0.957
Noisy-AND a_{10}	0.958	0.971
LSE r_1	0.915	0.943
LSE $r_{2.5}$	0.888	0.871
LSE r_5	0.940	0.957
GM r_1 (average pooling)	0.924	0.943
GM $r_{2.5}$	0.924	0.957
GM r_5	0.651	0.686
max pooling	0.452	0.429

Full image indicates accuracy on full resolution images. Treatment indicates accuracy predicting treatment MOA by taking the median over three experimental replicates.

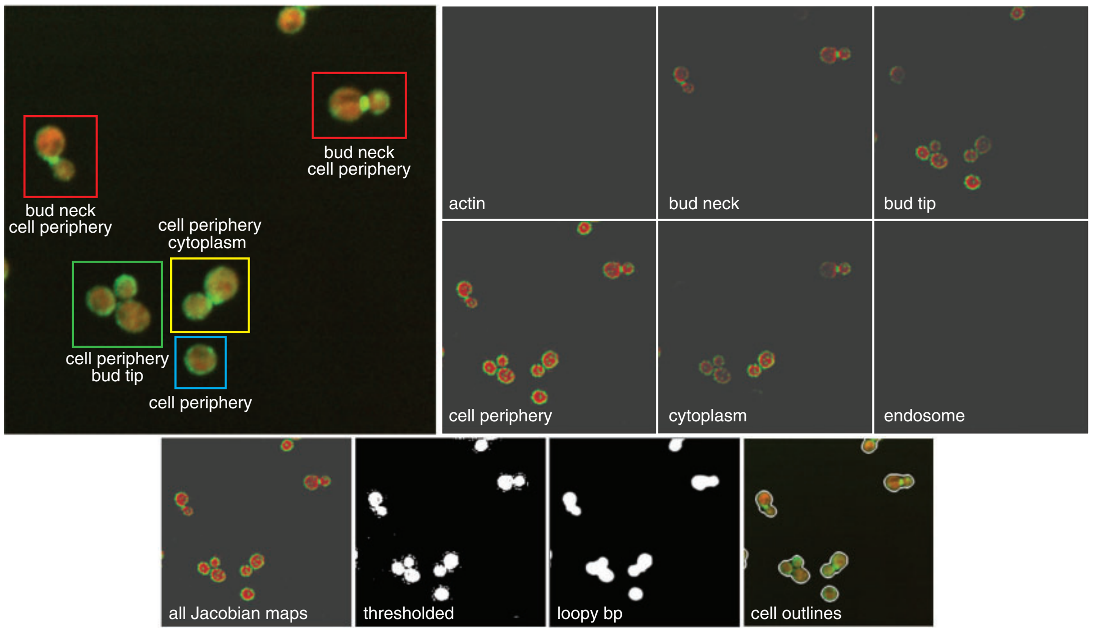

Fig. 6. Localizing cells with Jacobian maps. Top, yeast cells tagged with a protein that has cell cycle dependent localizations and corresponding Jacobian maps generated from class specific feature maps. Bottom, segmentation by thresholding Jacobian maps and de-noising with loopy bp

In addition to the performance on full resolution images, we evaluate the performance of the models on single cropped cells (Table 2). From a previous analysis pipeline using *CellProfiler* we extracted center of mass coordinates of segmented cells and used these coordinates to crop single cells (crop size of 64 × 64) from the full resolution images. This dataset was annotated according to the labels from the full resolution images and *likely includes mislabelled samples*. We also report performance on 6,300 manually labelled segmented cells (*segmented cells with manual labels* in table 2) used to train the SVM classifiers described in Chong *et al.* (2015). For these predictions we use the output from the MIL_pool layer.

Finally, we demonstrate that our model learns to locate regions with cells. We generated segmentation maps identifying cellular regions in the input by back-propagating activations as described in Section 2.4 (Fig. 6). To evaluate our segmentation method we calculated the mean intersection over union (IU) between our maps and segmentation maps generated using the global otsu thresholding module in *CellProfiler* which was used in Chong *et al.* (2015). We achieve a **mean IU of 81.2%** using this method. We found that mask pairs with low IU were mostly incorrect using Otsu thresholding. We also demonstrate that our model can generate class specific segmentation maps by back-propagating individual class specific feature maps while setting the rest of the feature maps to zero. Figure 6 shows the Jacobian maps generated for an image with transient, cell cycle dependent protein localizations.

4 Conclusions

Our proposed model links the benefits of MIL with the classification power of CNNs. We based our model on similarities between the aggregation function $g(\cdot)$ used in MIL models and pooling layers used in CNNs. To facilitate MIL aggregation in CNN feature maps we introduced the Noisy AND layer, a pooling function designed to be robust to outliers and learn the area of cells required to activate a label. This approach allows our model to learn instance and bag level classifiers for full resolution microscopy images without ever having to segment or label single cells. Our results indicate that convolutional MIL models achieve better performance across all evaluations against several benchmarks.

The benchmarks we compare against include a classification CNN with a similar architecture trained on cropped cells given noisy whole image (bag level) labels, and a naive implementation of convolutional MIL which uses global max pooling for $g(\cdot)$. It's clear that the naive max pooling implementation, which perfectly satisfies the standard MIL assumption, isn't suited for convolutional MIL applied to microscopy datasets. For the yeast dataset we compare with results published in Chong *et al.* (2015) using an ensemble of 60 binary SVM classifiers. For the breast cancer dataset we compare with results published in Ljosa *et al.* (2013) using factor analysis. Our models outperform previously published results for both datasets without any pre and post processing steps.

We found that for all the convolutional MIL models, mean average precision is higher when evaluated on manually labelled segmented cells than when evaluated on segmented cells labelled with bag level labels (Table 2). For the model trained on segmented cells with bag level labels, we see the opposite trend (a drop in performance when evaluating on manually labelled cells). This clearly indicates the utility of the MIL pooling layer. In the MIL models, the noisy labels are assigned to whole images and through the MIL pooling layer the model can learn to associate the labels with different instances in the image. The alternative baseline approach of training on segmented cells assigns the noisy labels to each segmented cell, forcing the model to learn incorrect patterns. The MIL models instead learn to identify cells in the full resolution images that correspond true phenotype categories given only the bag level annotations. This result is also shown by the class specific Jacobian maps visualized in Figure 6. We see that different patterns in the full resolution image activate class specific output feature maps.

For all of the bag level evaluations, we see that the Noisy-AND models perform best. We believe this can be explained by the pooling functions plotted in Figure 2. Setting the scaling factors (a, r) to lower values make the pooling functions approach mean of the feature maps, while for higher values the functions approach the max function. Since different phenotype categories may have vastly different densities of cells neither extreme suites all classes. The Noisy-AND pooling function accommodates this variability by learning an adaptive threshold for every class, as shown in Figure 2.

The breast cancer screen differs from the yeast data in several ways. Human cells interact and form denser cultures while yeast is unicellular and are typically imaged at lower densities. Also, drug treatments in the breast cancer screen affect most of the cells (Ljosa *et al.*, 2013) while the protein localization screen contains some localization categories that only occur transiently or in a fraction of the imaged cells. For the breast cancer dataset (Table 3) we also see that that Noisy-AND outperforms the previously reported treatment accuracy (Ljosa *et al.*, 2013).

In summary, we found that the MIL approach we developed offers several advantages for applications requiring classification of microscopy images. Our approach only requires a handful of labelled full resolution microscopy images. For example, we trained the breast cancer model with only 25 images per class. We envision that such training sets can reasonably be included as experimental controls in future screens. We also demonstrate that the convolutional MIL models can successfully be applied to a variety of datasets including hand written digits, yeast cells, and mammalian cells. The MIL models do not require any segmentation steps or per cell labels and they can be trained and tested directly on raw microscopy images in real-time. Finally, using the Jacobian maps we can segment cellular regions in the input image and assign predictions to each identified region.

Funding

B.J.F. received funding from the Natural Sciences and Engineering Research Council of Canada and a John C Polanyi Award. Some of Tesla K80s used for this research were donated by the NVIDIA Corporation.

Conflict of Interest: none declared.

References

Altschuler,S.J. and Wu,L.F. (2010) Cellular heterogeneity: do differences make a difference? *Cell*, **141**, 559–563.

Breker,M. *et al.* (2013) A novel single-cell screening platform reveals proteome plasticity during yeast stress responses. *J. Cell Biol.*, **200**, 839–850.

Carpenter, A.E. *et al.* (2006) Cellprofiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biol.*, **7**, R100.

Chen, L.C. *et al.* (2014) Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv Preprint arXiv*, **1412**, 7062.

Chong,Y.T. *et al.* (2015) Yeast proteome dynamics from single cell imaging and automated analysis. *Cell*, **161**, 1413–1424.

Ciresan,D. *et al.* (2012) Deep neural networks segment neuronal membranes in electron microscopy images. In *Advances in neural information processing systems*. Lake Tahoe, Nevada, pp. 2843–2851 .

Dietterich,T.G. *et al.* (1997) Solving the multiple instance problem with axis-parallel rectangles. *Art. Intel.*, **89**, 31–71.

Eliceiri,K.W. *et al.* (2012) Biological imaging software tools. *Nat. Methods*, **9**, 697–710.

Frey,B.J. (1998) *Graphical Models for Machine Learning and Digital Communication*. MIT Press, Cambridge.

Held,M. *et al.* (2010) Cellcognition: time-resolved phenotype annotation in high-throughput live cell imaging. *Nat. Methods*, **7**, 747–754.

Huh,W.K. *et al.* (2003) Global analysis of protein localization in budding yeast. *Nature*, **425**, 686–691.

Keeler,J.D. *et al.* (1991). Integrated segmentation and recognition of hand-printed numerals. In *NIPS*. Advances in neural information processing systems Denver, Colorado.

Kingma,D. and Ba,J. (2014) Adam: A method for stochastic optimization. *arXiv Preprint arXiv*, **1412**, 6980.

Kraus,O.Z. and Brendan,J.F. (2016) Computer vision for high content screening. *Critical reviews in biochemistry and molecular biology*, **51**, 102–109.

Krizhevsky,A. *et al.* (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. Lake Tahoe, Nevada, pp. 1097sc105.

LeCun,Y. *et al.* (1998) Gradient-based learning applied to document recognition. *Proc. IEEE*, **86**, 2278–2324.

Liberali,P. *et al.* (2015) Single-cell and multivariate approaches in genetic perturbation screens. *Nat. Rev. Genet.*, **16**, 18–32.

Ljosa,V. *et al.* (2012) Annotated high-throughput microscopy image sets for validation. *Nat. Methods*, **9**, 637hods.

Ljosa,V. *et al.* (2013) Comparison of methods for image-based profiling of cellular morphological responses to small-molecule treatment. *J. Biomol. Screen.*, **18**, 1321–1329.

Long,J. *et al.* (2015). Fully convolutional networks for semantic segmentation. In *CVPR*. Boston, Massachusetts.

Ning,F. *et al.* (2005) Toward automatic phenotyping of developing embryos from videos. *Image Process. IEEE Trans.*, **14**, 1360–1371.

Pachitariu,M. *et al.*, (2013). Extracting regions of interest from biological images with convolutional sparse block coding. In *Advances in Neural Information Processing Systems*, Lake Tahoe, Nevada, pp. 1745sces .

Pathak,D. *et al.* (2014) Fully convolutional multi-class multiple instance learning. *arXiv Preprint arXiv*, **1412**, 7144.

Ramon,J.D. and Raedt,L. (2000). Multi instance neural networks. In *ICML workshop on attribute-value and relational learning*. Stanford, California.

Simonyan,K. and Zisserman,A. (2014) Very deep convolutional networks for large-scale image recognition. *arXiv Preprint arXiv*, **1409**, 1556.

Simonyan,K. *et al.* (2013) Deep inside convolutional networks: visualising image classification models and saliency maps. *arXiv Preprint arXiv*, **1312**, 6034.

Singh, S. *et al.* (2014) Increasing the content of high-content screening an overview. *J. Biomol. Screen.*, **19**, 640–650.

Springenberg,J.T. *et al.* (2014) Striving for simplicity: the all convolutional net. *arXiv Preprint arXiv*, **1412**, 6806.

Szegedy,C. *et al.* (2014) Going deeper with convolutions. *arXiv Preprint arXiv*, **1409**, 4842.

Tkach,J.M. *et al.* (2012) Dissecting dna damage response pathways by analysing protein localization and abundance changes during dna replication stress. *Nat. Cell Biol.*, **14**, 966–976.

Vincent,P. *et al.* (2010) Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, **11**, 3371–3408.

Wählby,C. *et al.* (2012) An image analysis toolbox for high-throughput C. elegans assays. *Nat. Methods*, **9**, 714–716.

Xu,X. and Frank,. (2004). Logistic regression and boosting for labeled bags of instances. *In Advances in Knowledge Discovery and Data Mining*. Springer, pp. 2722g.

Xu,Y. *et al.* (2014). Deep learning of feature representation with multiple instance learning for medical image analysis. In *ICASSP*. IEEE.

Zeiler,M.D. and Fergus,R. (2014). Visualizing and understanding convolutional networks. In *Computer Vision://www.ncbi*. Zurich, pp. 818este.

Zhang,C. *et al.* (2005). Multiple instance boosting for object detection. In *NIPS*, Advances in neural information processing systems, Vancouver, British Columbia.