Cell Detection with Deep Convolutional Neural Network and Compressed Sensing

Yao Xue and Nilanjan Ray

Computing Science, University of Alberta, Canada

September 25, 2017

Abstract

The ability to automatically detect certain types of cells or cellular subunits in microscopy images is of significant interest to a wide range of biomedical research and clinical practices. Cell detection methods have evolved from employing hand-crafted features to deep learning-based techniques to locate target cells. The essential idea of these methods is that their cell classifiers or detectors are trained in the pixel space, where the locations of target cells are labeled. In this paper, we seek a different route and propose a convolutional neural network (CNN)-based cell detection method that uses encoding of the output pixel space. For the cell detection problem, the output space is the sparsely labeled pixel locations indicating cell centers. Consequently, we employ random projections to encode the output space to a compressed vector of fixed dimension. Then, CNN regresses this compressed vector from the input pixels. Using L_1-norm optimization, we recover sparse cell locations on the output pixel space from the predicted compressed vector. In the past, output space encoding using compressed sensing (CS) has been used in conjunction with linear and non-linear predictors. To the best of our knowledge, this is the first successful use of CNN with CS-based output space encoding. We experimentally demonstrate that proposed CNN + CS framework (referred to as CNNCS) exceeds the accuracy of the state-of-the-art methods on many

benchmark datasets for microscopy cell detection. Additionally, we show that CNNCS can exploit ensemble average by using more than one random encodings of the output space. In the AMIDA13 MICCAI grand competition, we achieve the 3rd highest F1-score in all the 17 participated teams. More ranking details are available at http://amida13.isi.uu.nl/?q=node/62. Implementation of CNNCS is available at https://github.com/yaoxuexa/CNNCS.

Keywords: Cell Detection, Convolutional Neural Network, Compressed Sensing.

1 Introduction

Automatic cell detection is to find whether there are certain types of cells present in an input image (e.g. microscopy images) and to localize them in the image. These targets might also refer to cellular subunits (mitotic figures, nucleus, apoptotic debris) but in the rest of the manuscript are referenced as cells. It is of significant interest to a wide range of medical imaging tasks and clinical applications. An example is breast cancer, where the number of proliferating tumor cells (involved in cell cycle) is an important indicator associated with the severity of the disease. Fig. 1 shows histological images of breast cancer with annotated mitotic figures which indicate some proliferative status.

Cell detection and localization constitute several challenges that deserve our attention. First, target cells are surrounded by clutter represented by complex histological structures like capillaries, adipocytes, collagen, etc. In many cases, the target cell is small in size, and consequently, almost indistinguishable from the clutter. Second, the target cells can appear very sparsely (only in tens), moderately densely (in tens of hundreds) or highly densely (in thousands) in a typical 2000-by-2000 pixel high resolution microscopy image. Additionally, significant variations in the appearance among the targets can be seen (mitotic figures as shown in Figure 1 are characteristically quite variable). These challenges render the cell detection/localization/counting problems far from being solved at the moment, in spite of significant progresses in computer vision research.

Extended object detection, such as detection of humans and vehicles, have witnessed much progress in the computer vision community. For example, Region-based Convolutional

Neural Networks (R-CNN) [15] and its variants [14], [33], Fully Convolutional Networks (FCN) [35] with recent optimization [32] have become the state-of-the-art algorithms for the extended object detection problem. Unfortunately, these solutions cannot be easily translated to small object localization, because assumptions and challenges are different for the latter. For example, for an extended object, localization is considered successful if a detection bounding box is 50% overlapping with the actual bounding box. For small object localization, tolerance is typically on a much tighter side in order for the localization to be meaningful.

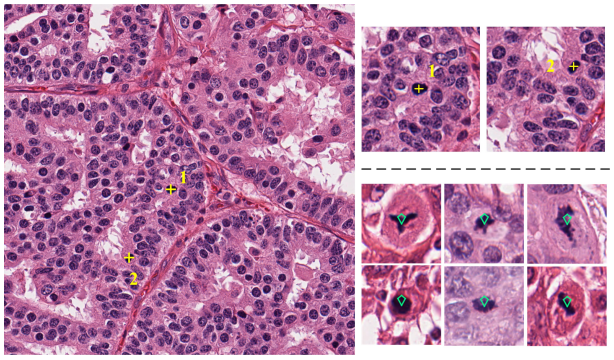

Figure 1: Left picture shows a microscopy image with two target cells annotated by yellow crosses on their centers. Right top pictures give details about the two target cells whose nuclei are in mitotic phase. Right bottom pictures provide more examples of mitotic figures, where a green diamond is attached to indicate the pixel region of each target.

In the last few decades, different cell recognition methods had been proposed, most of them depended on segmentation of cells that is summarized by Meijering [13]. Classically, it includes some basic image processing techniques, such as intensity thresholding, feature detection, morphological filtering, region accumulation, and deformable model fitting. For example, a popular choice is blob detector using Laplacian-of-Gaussian (LoG) filter [24] that offers many interesting texture properties and have been attempted for cell detection task [1].

To summarize, conventional cell detection approaches follow a "hand-crafted feature representation"+"classifier" framework. First, detection system extracts features as the representation of input images. Image processing techniques offer a range of feature extraction algorithms, such as Sobel operator, histogram of oriented gradients (HOG) [8], scale-invariant

feature transforms (SIFT) [28], local binary pattern (LBP) [31], etc. After feature extraction, machine learning based classifiers (e.g., support vector machine) work on the feature vectors to identify or recognize regions containing target cells or clutter. While being a powerful technique, "hand-crafted feature representation"+"classifier" approaches suffer from the following limitations:

(1) It is a non-trivial and difficult task for humans to select suitable features for detection of certain cells. In many cases, it requires significant prior knowledge about the task/data to find suitable features that can discriminate target cells and background.

(2) Additionally, most hand-crafted features contain many parameters that are crucial for the overall performance. But, the users need to perform a lot of trial-and-error experiments to tune the parameters.

(3) Usually, one particular feature is not versatile enough- the feature may often be tightly coupled with a particular type of target cell and may not work well when presented with a different type of target cell.

(4) The performance of a hand-crafted feature-based classifier soon reach an accuracy plateau, even when trained with plenty of training data.

In comparison, feature learning by deep neural networks recently has been applied to a variety of computer vision problems, and has achieved better performance on several benchmark vision datasets [25], [15], [35] over the classical approaches with hand-crafted features.

The most compelling advantage of deep learning is that it has evolved from fixed feature design strategies towards automated learning of problem-specific features directly from target data [27]. By providing massive training images and problem-specific labels, users do not have to go into the elaborate procedure for the extraction of features. Instead, deep neural network (DNN) is subsequently optimized using a mini-batch gradient descent method over the training data, so that the DNN allows autonomic learning of implicit relationships within the data. For example, shallow layers of DNN focus on learning low-level features (such as edges, lines, dots), while deep layers of DNN form more abstract high-level semantic representations (such as probability maps, or object class labels).

With the advent of deep learning in the computer vision community, it is no wonder

that the state-of-the-art methods in cell detection are based on convolutional neural network (CNN) [25]. For example, one such system [41] published in 2015, employs a fully convolutional network (FCN) [35], a close cousin of CNN, to predict the density map of cells. A cell density map is a heat-map that defines the number of cells per square area at any given pixel location. Next, post processing is attributed to extract centroid locations of the cells from the density map.

In this work, deviating from past approaches, we introduce output space encoding in the cell detection and localization problem. Our observation is that the output space of cell detection is quite sparse: an automated system only needs to label a small fraction of the total pixels as cell centroid locations. To provide an example, if there are 5000 cells present in an image of size 2000-by-2000 pixels, this fraction is $5000/(2000 * 2000) = 0.00125$, signifying that even a dense cell image is still quite sparse in the pixel space.

Based on the aforementioned observation, we are motivated to apply compressed sensing (CS) techniques in the cell detection task. First, a fixed length, compressed vector is formed by randomly projecting the cell locations from the sparse pixel space. Next, a deep CNN is trained to predict the encoded, compressed vector directly from the input pixels (i.e., microscopy image). Then, L_1 norm optimization is utilized to recover sparse cell locations. We refer to our proposed cell detection framework as CNNCS (convolutional neural network + compressed sensing).

Output space encoding/representation/transformation sometimes yields more accurate predictions in machine learning [9], [38]. In the past, CS-based encoding is used in conjunction with linear and non-linear predictions [17], [23], [20]. We believe, the proposed CNNCS is the first such attempt to solve cell detection and localization that achieved excellent competitive results on benchmark datasets.

There are several advantages of using CS-based output encoding for cell detection and localization. First, the compressed output vector is much shorter in length than the original sparse pixel space. So, the memory requirement would be typically smaller and consequently, there would be less overfitting. Next, there are plenty of opportunities to apply ensemble average to improve generalization. Furthermore, CS-theory dictates that pairwise distances are approximately maintained in the compressed space [10], [11]. Thus, even after output

space encoding, the machine learner still targets the original output space in an equivalent distance norm. From earlier research, we also point out a generalization error bound for such systems.

The rest of the paper is organized as follows. In section 2 the background and related work on cell detection and localization is explored. In section 3, we provide a detailed description of CNNCS. In section 4, experiments are reported. Section 5 concludes and points out future directions.

2 Background and Related Work

In this section, we provide essential contexts for the proposed CNNCS framework.

2.1 General Object Detection

Prior to deep learning, object detection pipeline consisted of feature extraction followed by classifiers or detectors.

Handcrafted feature based approaches

Computer vision problems like image classification and object detection had traditionally been addressed using the handcrafted features like SIFT [28], HOG [8], LBP [31], etc. SIFT (Scale Invariant Feature Transform) [28] was introduced by Lowe in 1999, and later became the basis for the popular Bag-of-Visual-Words model [43] for object recognition. SIFT was a robust solution to the problem of comparing image patches, for example, when we take multiple images of the same physical object while rotating the camera, the SIFT descriptors of corresponding points are very similar in their 128-D space. Around 2005, a new feature descriptor HOG [8] was initially proposed to the problem of pedestrian detection, then successfully applied as a general object descriptor. Bag-of-Visual-Words [43] model based algorithms once enjoyed significant success in image classification tasks. At that time, progress in computer vision greatly depended on the invention of more discriminative hand-crafted features.

Deep learning based approaches

In 2012 during the ImageNet classification competition, which involved the task of classifying an image into one of a thousand categories, for the first time, a Convolutional Neural Network (CNN) based deep learning model [25] brought down the error rate on that task by almost a half, beating traditional, hand-crafted feature approaches. CNN, a particular form of deep learning models, have since been widely adopted by the computer vision community.

In particular, the network trained by Alex Krizhevsky, popularly called AlexNet [25] had been used and modified for various vision problems. After that, several milestone networks (VGGNet [21], ResNet [16], Inception nets [36] and DenseNets [18] among many others) have been proposed since 2012. Deep learning has become one of the most influential methods in computer vision, and achieved successes in most major computer vision tasks like classification [25], [16], [18], detection [15], [14], [33], [32] and semantic image [35] and video segmentation [39].

2.2 Deep Learning for Cell Detection and Localization

Recently, a Fully Convolutional Network (FCN) [35] was proposed for the image segmentation problem and had shown remarkable performance. Soon after the FCN is proposed, [41] presented a FCN-based framework for cell counting, where their FCN is responsible for predicting a spatial density map of target cells, and then the number of cells can be estimated by an integration over the learned density map.

Once again, state-of-the-art methods in detection and localization today include deep learning techniques for cell detection and localization. Mitosis detection has been proposed by CNN-based prediction followed by ad-hoc post processing in [7]. Recently, expectation maximization has been utilized within deep learning framework in an end-to-end fashion for mitosis detection [34]. A cascaded network has been proposed for the same task more recently [6], which is in principle similar to the FCN-based method [41] that tries to predict a probability map.

2.3 Compressed Sensing

During the past decade, compressed sensing or compressive sensing (CS) [10] has emerged as a new framework for signal acquisition and reconstruction, and has received growing attention, mainly motivated by the rich theoretical and experimental results as shown in many reports [11], [12], [10], and so on. As we know, the Nyquist-Shannon sampling theorem states that a certain minimum sampling rate is required in order to reconstruct a band-limited signal. However, CS enables a potentially large reduction in the sampling and computation costs for sensing/reconstructing signals that are sparse or have a sparse representation under some linear transforms (e.g. Fourier transform).

Under the premise of CS, an unknown signal of interest is observed (sensed) through a limited number of linear observations. Many authors [11], [12], [10] have proven that it is possible to obtain a stable reconstruction of the unknown signal from these observations, under the general assumptions that the signal is sparse (or can be represented sparsely with respect to a linear basis) and matrix incoherence. The signal recovery techniques typically rely on convex optimization with a penalty expressed by L_1 norm, for example orthogonal matching pursuit (OMP) [2] and dual augmented Lagrangian (DAL) method [37].

3 Proposed Method

3.1 System Overview

The proposed detection framework consists of three major components: (1) cell location encoding phase using random projection, (2) a CNN based regression model to capture the relationship between a cell microscopy image and the encoded signal y, and (3) decoding phase for detection. The flow chart of the whole framework is shown in Fig. 2.

During training, the ground truth location of cells is indicated by a pixel-wise binary annotation map B. We propose a cell location encoding scheme, which converts cell location from pixel space representation B to compressed signal representation y. This encoding may consist of reshaping a sparse matrix B into a sparse vector f by row or column major fashion. Then, f is multiplied by a sensing matrix (usually, a random Gaussian matrix) to form a

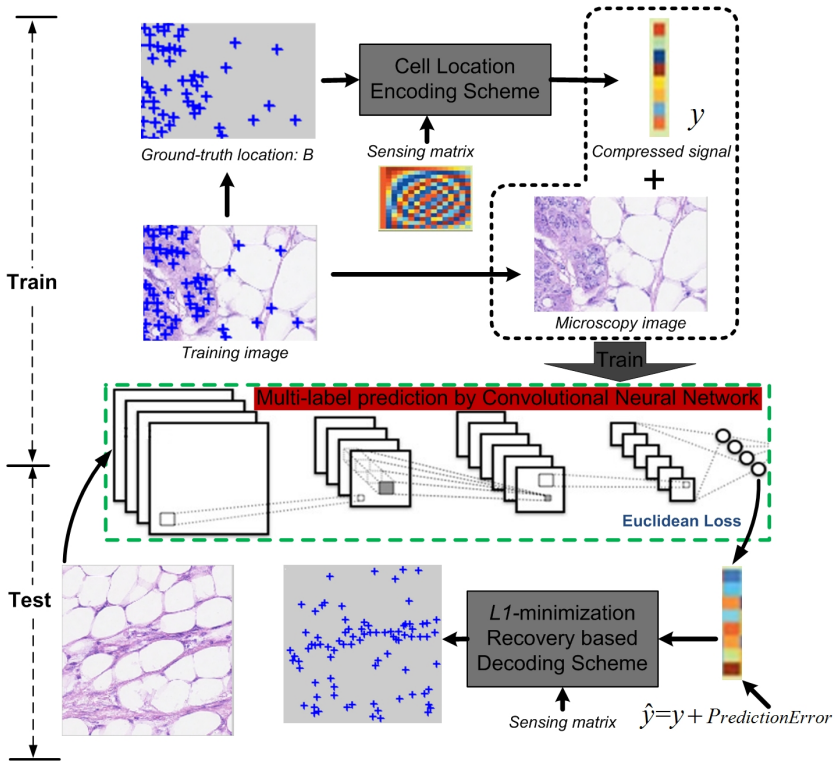

Figure 2: The system overview of the proposed CNNCS framework for cell detection and localization.

compressed and encoded vector y. The encoding scheme can also be more sophisticated as discussed later. Then, a training pairs, each consisting of a cell microscopy image and the signal y, train a CNN to work as a multi-label regression model. We employ the Euclidean loss function during training, because it is often more suitable for regression. Image rotations may be performed on the training sets for the purpose of data augmentation as well as making the system more robust to rotations.

During testing, the trained network is responsible for outputting an estimated signal $\hat{y}$ for each test image. After that, a decoding scheme is designed to predict the cell location by performing L_1 minimization recovery on the estimated signal $\hat{y}$, with the known sensing matrix.

3.2 Cell Location Encoding and Decoding Scheme

3.2.1 Encoding Schemes

In our experiments, we employ two types of random projection-based encodings as described below.

Scheme-1: Encoding by Reshaping

For the cell detection problem, cells are often annotated by pixel-level labels. The most common way is to attach a dot or cross at the center of every cell, instead of a bounding box around the cell. So, let us suppose there is a pixel-wise binary annotation map B of size w-by-h, which indicates the location of cells by labeling 1 at the pixels of cell centroids, otherwise labeling 0 at background pixels. To vectorize the annotation map B, the most intuitive scheme is to concatenate every row of B into a binary vector f of length wh. Thus, a positive element in B with $\{x, y\}$ coordinates will be encoded to the $[x + h(y - 1)]$-th position in f. f is also a k-sparse signal, so, there are at most k non-zero entries in f. Here, we refer this intuitive encoding scheme as "Scheme-1: Encoding by Reshaping".

After the vector f is generated, we apply a random projection. CS theory guarantees that f could be fully represented by linear observations y:

$$y = \Phi f, \tag{1}$$

provided the sensing matrix Φ satisfies the restricted isometry property (RIP) condition [11], [12]. In many cases, Φ is typically a $M \times N$ ($M \ll N = hw$) random Gaussian matrix. Here, the number of observations M is much smaller than N, and obeys: $M \geq C_M klog(N)$, where C_M is a small constant greater than one.

Scheme-2: Encoding by Signed Distances

For the encoding scheme-1, the space complexity of the interim result f is $\mathcal{O}(w * h)$. For example, to encode the location of cells in a 260-by-260 pixel image, scheme-1 will produce f as a 67,600-length vector; so that in the subsequent CS process, a huge sensing matrix in size of M-by-67600 is required in order to match the dimension of f, which will make the system quite slow, even unacceptable once meeting with larger images. To further optimize the encoding scheme, we propose a second scheme, where the coordinates of every

cell centroid are projected onto multiple observation axes. We refer the second encoding scheme as "Scheme-2: Encoding by Projection".

To encode location of cells, we create a set of observation axes $OA = \{oa_l\}, l = 1, 2, \ldots, L$, where L indicates the total number of observation axes used. The observation axes are uniformly-distributed around the image space (See Fig. 3, left-most picture) For the l-th observation axis oa_l, the location of cells is encoded into a R-length ($R = \sqrt{w^2 + h^2}$) sparse signal, referred as f_l (See Fig. 3, third picture). We calculate the perpendicular signed distances (f_l) from cells to oa_l. Thus, f_l contains signed distances, which not only measure the distance, but also describe on which side of oa_l cells are located. After that, the encoding of cell locations under oa_l is y_l, which is obtained by the following random projection:

$$y_l = \Phi f_l, \tag{2}$$

We repeat the above process for all the L observation axes and obtain each y_l. After concatenating all the y_l, $l = 1, 2, \ldots, L$, the final encoding result y is available, which is the joint representation of cells location. The whole encoding process is illustrated by Fig. 3.

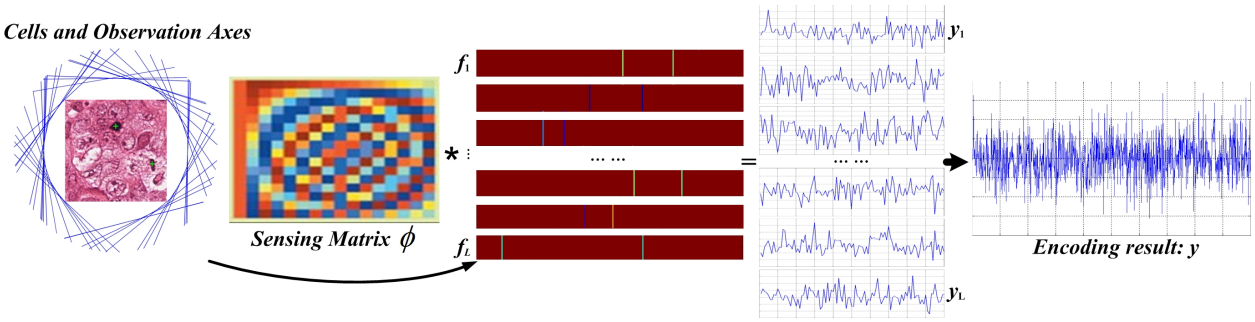

Figure 3: Cell location encoding by signed distances (Scheme-2).

For encoding scheme-2, the size of the sensing matrix Φ is M-by-$\sqrt{w^2 + h^2}$. In comparison, encoding scheme-1 requires a much larger sensing matrix of size M-by-wh. The first advantage of encoding scheme-2 is that it dramatically reduces the size of the sensing matrix, which is quite helpful for the recovery process, especially when the size of images is large. On the other hand, the encoding result y carries much information of cell locations obtained from L observation axes. In the subsequent decoding phase, averaging over the redundant information makes the final detection more reliable. More details can be found in experiments section. A final point is that in case more than one cell locations are projected

to the same bin in a particular observation axis, such a conflict will not occur for the same set of cells at other observation axes.

3.2.2 Decoding Scheme

Accurate recovery of f can be obtained from the encoded signal y by solving the following L_1 norm convex optimization problem:

$$\hat{f} = \arg\min_{f} \|f\|_1 \qquad \text{subject to} \qquad y = \Phi f \tag{3}$$

After $\hat{f}$ is recovered, every true cell is localized L times, i.e. with L candidate positions predicted. The redundancy information allows us to estimate more accurate detection of a true cell.

The first two images of Fig. 4 from left present examples of the true location signal f and decoded location signal $\hat{f}$, respectively. The noisy signed distances of $\hat{f}$ are typically very close to each observation axis. That is why we create observation axes outside of the image space, so that these noisy distances can be easily distinguished from true candidate distances. This separation is done by mean shift clustering, which also groups true detections into localized groups of detections. Two such groups (clusters) are shown in Fig. 4, where the signed distances formed circular patterns of points (in green) around ground truth detections (in yellow). Averaging over these green points belonging to a cluster provides us a predicted location (in red) as shown in Fig. 4.

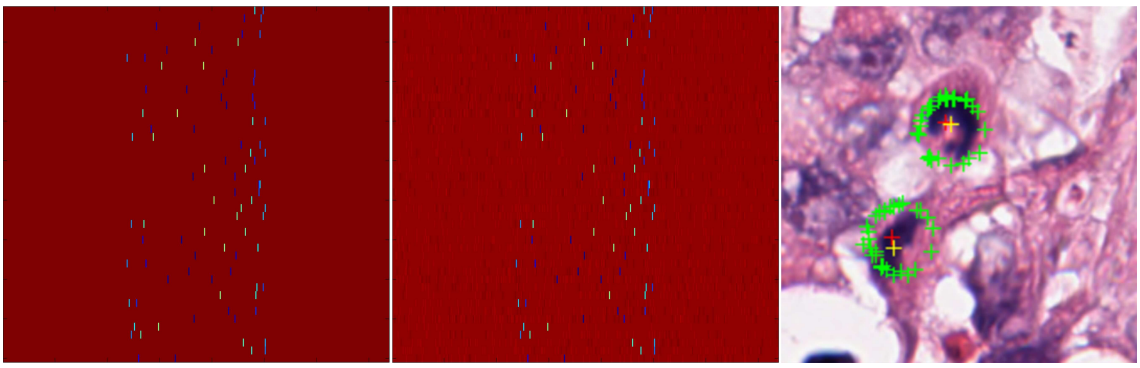

Figure 4: Cell Location Decoding Scheme. From left to right: true location signal f, decoded location signal $\hat{f}$ and detection results. Yellow crosses indicate the ground-truth location of cells, green crosses are the candidates points, red crosses represent the final detected points.

3.3 Signal Prediction by Convolutional Neural Network

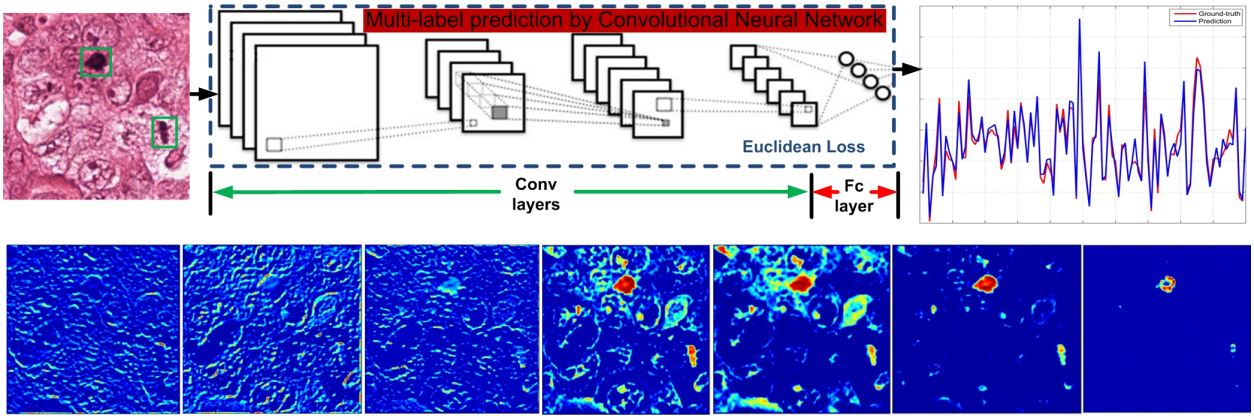

Figure 5: An illustration of the process of signal prediction by convolutional neural network. The bottom row presents the feature maps learned from Convolutional (Conv) layers of the CNN with training process going on. The current CNN follows the AlexNet architecture. These feature maps come from the Conv1, Conv1, Conv2, Conv3, Conv3, Conv4 and Conv5 respectively. The top-right picture shows the ground-truth compressed signal (red) and compressed signal (blue) predicted from the Fullly-connected (Fc) layer of the CNN. From the picture, we can observe that the predicted signal approximate the pattern of ground truth signal well.

We utilize CNN to build a regression model between a cell microscopy image and its cell location representation: compressed signal y. We employ two kinds of CNN architectures. One of them is AlexNet [25], which consists of 5 convolution layers + 3 fully connected layers; the other is the deep residual network (ResNet) [16] where we use its 152-layer model. In both the architectures, the loss function is defined as the Euclidean loss. The dimension of output layer of AlexNet and ResNet has been modified to the length of compressed signal y. We train the AlexNet model from scratch, in comparison, we perform fine-tuning on the weights in fully-connected layer of the ResNet.

To prepare the training data, we generate a large number of square patches from training images. Along with each training patch, there is a signal (i.e. the encoding result: y), which indicates the location of target cells present in the patch. After that, patch rotation is performed on the collected training patches for data augmentation and making the system

rotation invariant.

The trained CNN not only predicts the signal from its output layer, the feature maps learned from its Conv layers also provide rich information for recognition. Fig. 5 visualizes the learned feature maps, which represents the probabilistic score or activation maps of target cell regions (indicated by green boxes in the left image) during training process. It can be observed that higher scores are fired on the target regions of score masks, while most of the non-target regions have been suppressed more and more with training process going on.

3.3.1 Multi-Task Learning

To further optimize our CNN model, we apply Multi-Task Learning (MTL) [5]. During training a CNN, two kinds of labels are provided. The first kind is the encoded vector: y, which carries the pixel-level location information of cells. The other kind is a scalar: cell count (c), which indicates the total number of cells in a training image patch. We concatenate the two kinds of labels into the final training label by $label = \{y, \lambda c\}$, where λ is a hyper parameter. Then, Euclidean loss is applied on the fusion label. Thus, supervision information for both cell detection and cell counting can be jointly used to optimize the parameters of our CNN model.

3.4 Theoretical Justification

Equivalent Targets for Optimization

We first show that from the optimization standpoint, compressed vector is a good proxy for the original, sparse output space. This result directly follows from the CS theory. As mentioned before, f indicates the cell location represented in pixel space, and y is the cell location represented in compressed signal space. They follow the relationship: $y = \Phi f$, where Φ is the sensing matrix. Let us assume that f_p and f_g are respectively the prediction and ground-truth vectors in the pixel space. Similarly, we have y_p and y_g as their compressed counterparts, respectively.

Claim: $\|y_g - y_p\|$ and $\|f_g - f_p\|$ are approximately equivalent targets for optimization.

Proof: According to the CS theory, a sensing matrix $\Phi \in \mathbb{R}^{m \times d}$ should satisfy the $(k, \delta) - restricted\ isometry\ property\ ((k, \delta) - RIP)$, which states that for all $k-$sparse $f \in \mathbb{R}^d$, $\delta \in (0, 1)$, the following holds [11], [12], [10]:

$$(1 - \delta) \|f\| \leq \|\Phi f\| \leq (1 + \delta) \|f\|. \tag{4}$$

Note that if the sensing matrix Φ satisfies $(2k, \delta)$-RIP, then (4) also holds good. Now replace f with $(f_g - f_p)$ and note that $(f_g - f_p)$ is $2k$-sparse. Thus,

$$(1 - \delta) \|f_g - f_p\| \leq \|y_g - y_p\| \leq (1 + \delta) \|f_g - f_p\|. \tag{5}$$

From the right hand side inequality, we note that if $\|f_g - f_p\|$ is small, then $\|y_g - y_p\|$ would be small too. In the same way, if $\|y_g - y_p\|$ is large, then the inequality implies that $\|f_g - f_p\|$ would be large too. Similarly, from the left hand side inequality, we note that if $\|f_g - f_p\|$ is large then $\|y_g - y_p\|$ will be large, and if $\|y_g - y_p\|$ is small then $\|f_g - f_p\|$ will small too. These relationships prove the claim that from the optimization perspective $\|y_g - y_p\|$ and $\|f_g - f_p\|$ are approximately equivalent.

A Bound on Generalization Prediction Error

In this section we mention a powerful result from [17]. Let h be the predicted compressed vector by the CNN, f be the ground truth sparse vector, $\hat{f}$ be the reconstructed sparse vector from prediction, and Φ be the sensing matrix. Then the generalization error bound provided in [17] is as follows:

$$\|\hat{f} - f\|_2^2 \leq C_1 \cdot \|h - \Phi f\|_2^2 + C_2 \cdot sperr(\hat{f}, f), \tag{6}$$

where C_1 and C_2 are two small constants and *sperr* measures how well the reconstruction algorithm has worked [17]. This result demonstrates that expected error in the original space is bound by the expected errors of the predictor and that of the reconstruction algorithm. Thus, it makes sense to apply a very good machine learner such as deep CNN that can minimize the first term in the right hand side of (6). On the other hand, DAL provides one of the best L_1 recovery algorithms to minimize the second term in the right side of (6).

4 Experiments

4.1 Datasets and Evaluation Criteria

First, we describe five cell datasets, on which the proposed method and other comparison methods are evaluated. The first dataset [22] involves 100 H&E stained histology images of colorectal adenocarcinomas. The second dataset [3] consists of 200 highly realistic synthetic emulations of fluorescence microscopic images of bacterial cells. The third dataset [42] comprises of 55 high resolution microscopic images of breast cancers double stained in red (cytokeratin epithelial marker) and brown (nuclear proliferative marker). The fourth dataset is the 2012 ICPR Mitosis contest dataset [26] including 50 high-resolution (2084-by-2084) RGB microscope slides of Mitosis. The last dataset is the AMIDA-13 challenge dataset [40], which contains a total of 606 breast cancer histology images, belonging to 23 patients. Suspicious breast tissue are collected and stained using hematoxylin and eosin (H&E), then the dotted annotation was done by at least two expert pathologists. For each dataset, the annotation that represents the location of cells is shown in Fig.6, details of datasets are summarized in Table. 1.

Table 1: *Size* is the image size; *Ntr/Nte* is the number of images selected for training and testing; AC indicates the average number of cells; *MinC-MaxC* is the minimum and maximum numbers of cells.

Cell Dataset	Size	Ntr/Nte	AC	MinC-MaxC
Nuclei [22]	500×500	50/50	310.22	1-1189
Bacterial [3]	256×256	100/100	171.47	74-317
Ki67 Cell [42]	1920×2560	45/10	2045.85	70-4808
Mitosis [26]	2084×2084	35/15	5.31	1-19
AMIDA-13 [40]	2000×2000	447/229	3.54	0-9

For evaluation, we adopt the criteria of the 2012 ICPR Mitosis contest [26], a detection would be counted as true positive (TP) if the distance between the predicted centroid and ground truth cell centroid is less than ρ. Otherwise, a detection is considered as false positives (FP). The missed ground truth cells are counted as false negatives (FN). In

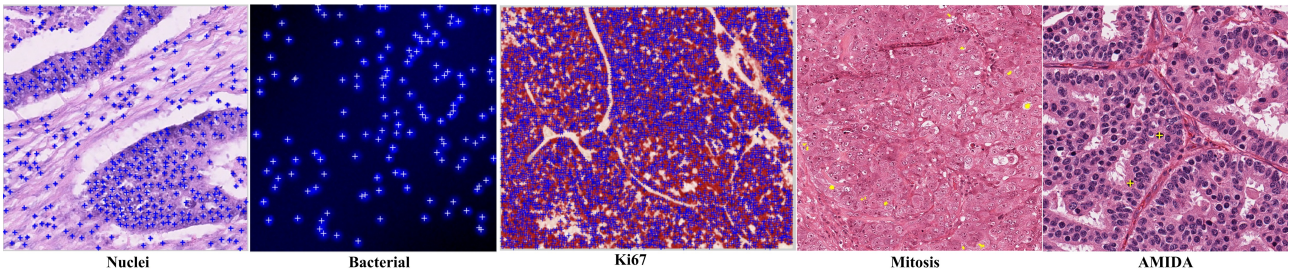

Figure 6: Dataset examples and their annotation.

our experiments, ρ is set to be the radius of the smallest cell in the dataset. Thus, only centroids that are detected to lie inside cells are considered correct. The results are reported in terms of Precision: $P = TP/(TP + FP)$ and Recall: $R = TP/(TP + FN)$ and F_1-score: $F_1 = 2PR/(P + R)$ in the following sections.

4.2 Experiments with Encoding Scheme-1

To evaluate, we carry out experimental performance comparison with CNNCS and three state-of-the-art cell detection methods ("FCN-based" [41], "Le.detect" [4], "CasNN" [6]). In this experiment, the scheme-1: encoding by reshaping is applied in CNNCS.

For the four methods to provide different values of Precision-Recall as shown in Fig. 7, we tune hyper parameters of every method. With scheme-1, CNNCS has a threshold T to apply on the recovered sparse signal $\hat{f}$ before re-shaping it to a binary image B. T is used to perform cell vs. non-cell binary classification and can be treated as a hyper parameter during training. In "FCN-based" [41], there is also a threshold applied to the local probability-maximum candidate points to make final decision about cell or non-cell. Similarly, in the first step of "Le.detect" [4], researchers use a MSER-detector (a stability threshold involved here) to produce a number of candidate regions, on which their learning procedure determines which of these candidates regions correspond to cells. In the first experiment, we analyze the three methods using Precision-Recall curves by varying their own thresholds.

Fig. 7 presents Precision-Recall curves. All the four methods give reliable detection performances in the range of recall=[0.1-0.4]. After about recall=0.6, the precision of "FCN-based" [41] drops much faster. This can be attributed to the fact that "FCN-based" [41]

works by finding local maximum points on a cell density map. However, the local maximum operation fails in several scenarios, for example when two cell density peaks are close to each other, or large peak may covers neighboring small peaks. Consequently, to obtain the same level of recall, "FCN-based" [41] provides many false detections.

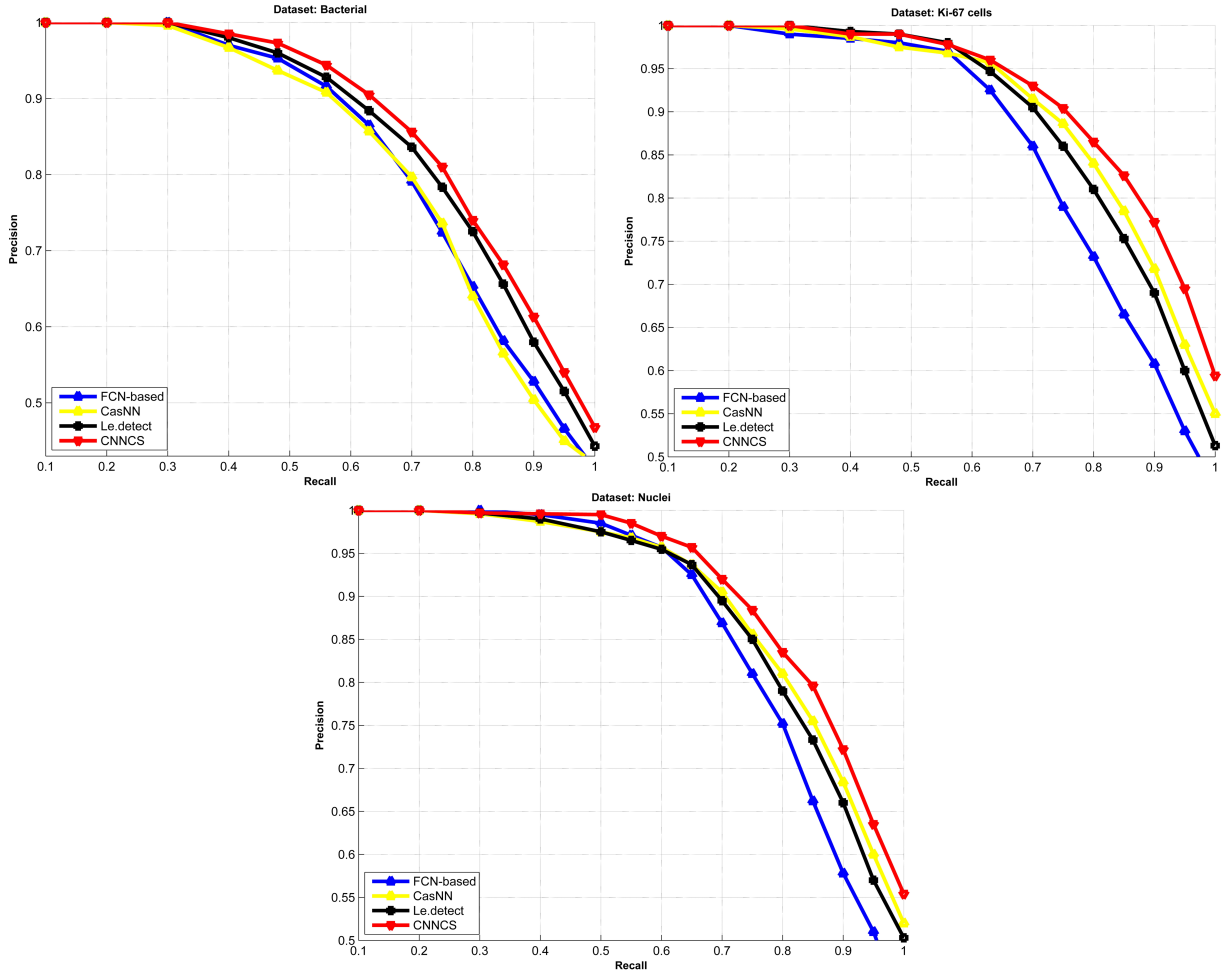

Figure 7: Precision and recall curves of four methods on three datasets.

Furthermore, it also can be observed that CNNCS has an improvement over "Le.detect" [4] (red line clearly outperforms black line under varying recall values). This can be largely explained by the fact that traditional methods (no matter if [4] or [41] is used) always try to predict the coordinates of cells directly on a 2-D image. The coordinates are sensitive to system prediction bias or error, considering the nature of cell detection that cells are small and quite dense in most cases. It is not surprising that "Le.detect" [4] will miss some cells and/or detect other cells in wrong locations. In comparison, CNNCS transfers

the cell detection task from pixel space to compressed signal space, where the location information of cells is no longer represented by $\{x, y\}$-coordinates. Instead, CNNCS performs cell detection by regression and recovery on a fixed length compressed signal. Compared to $\{x, y\}$-coordinates representation, the compressed signal is more robust to system prediction errors. For example, as shown in Fig. 5, even though there are differences between the ground-truth compressed signal and predicted compressed signal, the whole system can still give reliable detection results as shown in Fig. 7.

To get a better idea of the CNNCS method, we visualize a set of cell images with their detected cells and ground-truth cells in Fig. 8. It can be observed that CNNCS is able to accurately detect most cells under a variety of conditions.

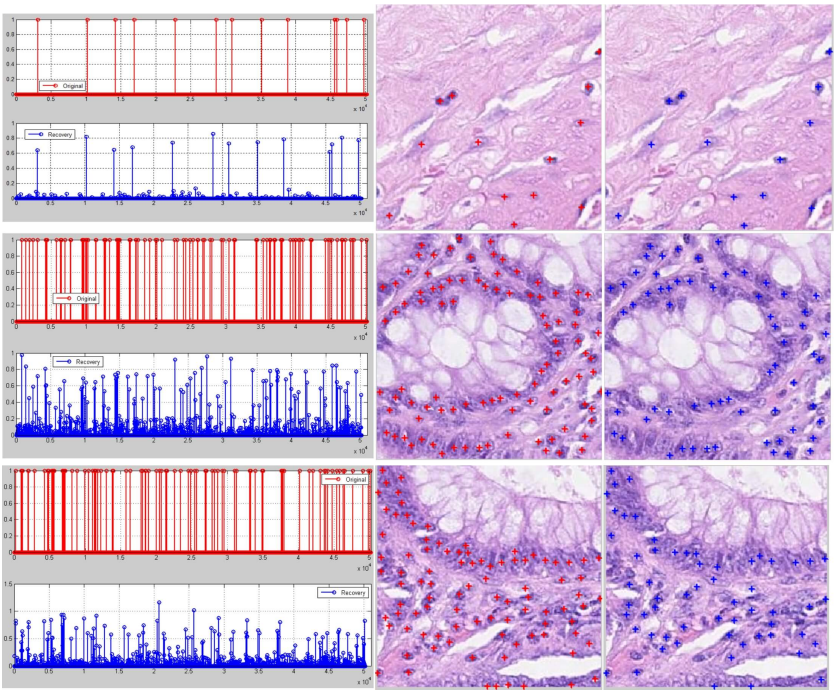

Figure 8: Detection results. Ground-truth: red, Prediction: blue. Left part shows the sparse signal that carries the location information of cells; right part shows the ground-truth and detected cells.

4.3 Experiments with Encoding Scheme-2

Since the 2012 ImageNet competition, convolutional neural networks have become popular in large scale image recognition tasks, several milestone networks (including AlexNet [25],

VGGNet [21], etc) have been proposed. Recently, deep residual network (ResNet) introduces residual connections into deep convolutional networks and has yielded state-of-the-art performance in the 2015 ILSVRC challenge [16]. This raises the question of whether there is any benefit in introducing and exploiting more recent CNN architectures into the cell detection task. Thus, in this section, we are exploring the performance of CNNCS with different neural network architectures (AlexNet and ResNet).

Table 2: Results on 2012 ICPR MITOSIS Dataset.

Method	Precision	Recall	F_1-score
UTRECHT [29]	0.511	0.680	0.584
NEC [30]	0.747	0.590	0.659
IPAL [19]	0.698	0.740	0.718
DNN [7]	0.886	0.700	0.782
RCasNN [6]	0.720	0.713	0.716
CasNN-single [6]	0.738	0.753	0.745
CasNN-average [6]	0.804	0.772	0.788
CNNCS-AlexNet	0.860	0.788	0.823
CNNCS-ResNet	0.867	0.801	0.833
CNNCS-ResNet-MTL	0.872	0.805	**0.837**

To carry out experimental performance comparisons, we apply the proposed method on the 2012 ICPR Mitosis contest dataset, which consists of 35 training images and 15 testing images. For the training process, we extracted image sub-samples (260-by-260) with no overlap between each other from the 35 training images. After that every 90° image rotation is performed on each sub-sample for data augmentation, finally resulting in a total of 8,960 training dataset. In addition, we perform cross-validation and random-search to tune the three hyper parameters in scheme-2: (1) the number of rows in sensing matrix: M, (2) the number of observation lines: L and (3) the importance (λ) of cell count during MTL. After that, the best performance is achieved when $M = 112, L = 27, \lambda = 0.2$. Furthermore, we trained five CNN models to reduce the performance variance introduced by a single model and to improve the robustness of the whole system. Finally, the performance of CNNCS with model averaging is compared with other methods in Table 2.

Compared to the state-of-the-art method, CasNN-average [6], CNNCS with ResNet and MTL achieved a better performance with F_1-score 0.837 also outperforming all other methods by a significant margin. It can be observed that both precision and recall have increased compared to all other methods, but the overall increase of F_1-score can be contributed to the improvement of precision. As seen in Table 2, precision of our method outperforms the best comparison Precision by 0.06-0.07, while of course, recall also has recorded about 0.02 improvement. This phenomenon can be attributed to the detection principle of our method, where every ground-truth cell is localized with multiple candidate points guaranteed around the true location, then the average coordinates of these candidates is computed as the final detection. As a result, localization closer to the true cell becomes more reliable compared to existing methods, thus leading to a higher precision. In addition, an improvement of F_1-score from 0.833 to 0.837 achieved by MTL demonstrates that the knowledge jointly learned for cell detection and cell counting provides further benefits at negligible additional computations.

AMIDA-13 dataset [40] contains 606 breast cancer histology images, belonging to 23 patients. Suspicious breast tissue is collected and stained using hematoxylin and eosin (H&E), then the dotted annotation is done by at least two expert pathologists, to label the center of each cancer cell. We train the proposed CNNCS method from 1-15 patients (377 HPF images), validate on 20% of the training set (70 HPF images) and test it on the test data of AMIDA-13 Challenge that has 229 HPF images from 8 patients. Each histology images is 2000-by-2000 pixels. We partition it into patches of size of 200-by-200 pixels without any overlap. Fig.9 provides nine examples of our detection results. Similar to the experiment on MITOSIS dataset, we also perform random search on a validation set to optimize the hyper parameters. The best performance on AMIDA dataset is achieved when $M = 103, L = 30, \lambda = 0.2$. Table 3 summarizes comparisons of CNNCS with other methods. For the AMIDA13 MICCAI grand competition [40], we employed ResNet as the network architecture with data balancing in the training set. Furthermore, we apply the following ensemble averaging technique during testing to further increase precision and recall values. Originally, we have partitioned every 5657-by-5657 AMIDA-13 test image into about 100 non-overlapping patches. Instead of starting the partitioning from the top-left corner of an

AMIDA image, now we set the starting point of the first patch from {offset, offset}. The offset values are set as 0, 20, 40,..., 160, and 180 (i.e. every 20 pixel) resulting in a total of 10 different settings. Under every offset setting, CNNCS method is run on all the generated patches and provides detection results. Then, we merge detection results from all the offset settings. The merging decision rule is that if there are 6 or more detections within a radius of 9 pixels, then we accept average of these locations as our final detected cell center. Finally, we achieve the **third** highest F1-score in all the 17 participated teams. More ranking details are available at http://amida13.isi.uu.nl/?q=node/62.

Table 3: Results of AMIDA13 MICCAI Grand Challenge. Ranking according to the overall F1-score.

Method	Precision	Recall	F_1-score
IDSIA [7]	0.610	0.612	0.611
DTU	0.427	0.555	0.483
CNNCS (our method)	0.3588	0.5529	0.4352
AggNet [34]	0.441	0.424	0.433
CUHK	0.690	0.310	0.427
SURREY	0.357	0.332	0.344
ISIK	0.306	0.351	0.327
PANASONIC	0.336	0.310	0.322
CCIPD/MINDLAB	0.353	0.291	0.319
WARWICK	0.171	0.552	0.261
POLYTECH/UCLAN	0.186	0.263	0.218
MINES	0.139	0.490	0.217
SHEFFIELD/SURREY	0.119	0.107	0.113
SEOUL	0.032	0.630	0.061
NTUST	0.011	0.685	0.022
UNI-JENA	0.007	0.077	0.013
NIH	0.002	0.049	0.003

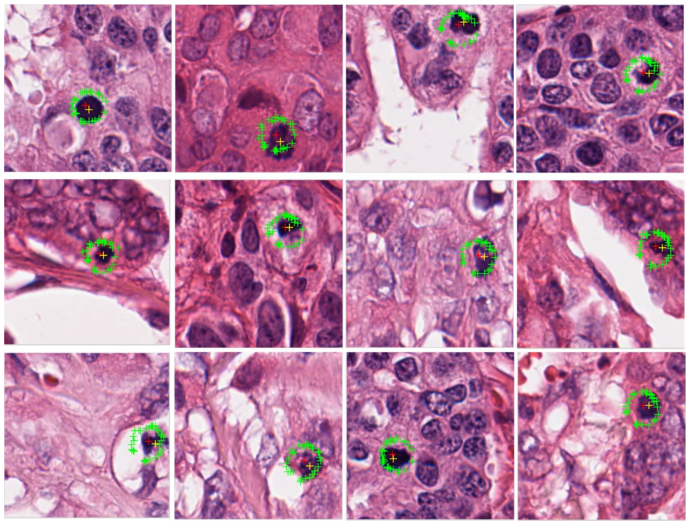

Figure 9: Results on AMIDA-13 dataset [40]. Yellow cross indicates the ground-truth position of target cells. Green cross indicates cell position predicted by an observation axis. Red cross indicates the final detected cell position, which is the average of all green crosses.

5 Conclusions and Future Directions

This paper demonstrates that deep convolutional neural network can work in conjunction with compressed sensing-based output encoding schemes toward solving a significant medical image processing task: cell detection and localization from microscopy images. We showed that CNN combined with the ensemble averaging provided by CS can beat or be competitive with state-of-the art methods on benchmark datasets. In the future, we plan to apply an end-to-end training to our CNNCS framework. Within this end-to-end framework, the decoding by L_1 optimization will also be included. The end-to-end framework has the potential optimize output encoding by modifying random projection matrices according to training data.

References

[1] Yousef Al-Kofahi, Wiem Lassoued, William Lee, and Badrinath Roysam. Improved automatic detection and segmentation of cell nuclei in histopathology images. *IEEE Transactions on Biomedical Engineering*, 57:841–852, 2010.

[2] T. Tony Cai and Lie Wang. Orthogonal matching pursuit for sparse signal recovery with noise. *IEEE Transactions on Information Theory.*, 2011.

[3] Alison Noble Carlos Arteta, Victor Lempitsky and Andrew Zisserman. Learning to count objects in images. *Neural Information Processing Systems.*, 2010.

[4] J. Alison Noble Andrew Zisserman Carlos Arteta, Victor Lempitsky. Learning to detect cells using non-overlapping extremal regions. *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 348–356, 2012.

[5] Rich Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, Jul 1997.

[6] Hao Chen, Qi Dou, Xi Wang, Jing Qin, and Pheng-Ann Heng. Mitosis detection in breast cancer histology images via deep cascaded networks. *Proceedings of the Thirtieth Conference on Artificial Intelligence (AAAI)*, 2016.

[7] Dan C. Cirean, Alessandro Giusti, Luca M. Gambardella, and Jrgen Schmidhuber. *Mitosis Detection in Breast Cancer Histology Images with Deep Neural Networks*. 2013.

[8] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. *Computer Vision and Pattern Recognition*, 2005.

[9] T. G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, pages 263–286, 1995.

[10] David L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 2006.

[11] Justin Romberg Emmanuel Candes. Practical signal recovery from random projections. *IEEE Transactions on Signal Process*, 2005.

[12] Terence Tao Emmanuel Candesy, Justin Rombergy. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory.*, 2006.

[13] Meijering Erik. Cell segmentation: 50 years down the road [life sciences]. *IEEE Signal Process. Mag.*, 2012.

[14] Ross Girshick. Fast r-cnn. *International Conference on Computer Vision*, 2015.

[15] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *Computer Vision and Pattern Recognition*, 2014.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Computer Vision and Pattern Recognition.*, 2015.

[17] Daniel Hsu, Sham M. Kakade, John Langford, and Tong Zhang. Multi-label prediction via compressed sensing. *arXiv:0902.1284v2 [cs.LG]*, 2009.

[18] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[19] Humayun Irshad. Automated mitosis detection in histopathology using morphological and multi-channel statistics features. *Journal of Pathology Informatics*, 2013.

[20] Arnaud Joly. Exploiting random projections and sparsity with random forests and gradient boosting methods. *arXiv:1704.08067*, 2016.

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

[22] Y.W Tsang I.A. Cree D.R.J. Snead K. Sirinukunwattana, S.E.A. Raza and N.M. Rajpoot. Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images. *IEEE Transactions on Medical Imaging.*, 2016.

[23] Ashish Kapoor, Raajay Viswanathan, and Prateek Jain. Multilabel classification using bayesian compressed sensing. *Advances in Neural Information Processing Systems*, 2012.

[24] Hui Kong, Hatice Cinar Akakin, and Sanjay E. Sarma. A generalized laplacian of gaussian filter for blob detection and its applications. *IEEE Transactions on Cybernetics*, 43:1719–1733, 2013.

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classication with deep convolutional neural networks. *Neural Information Processing Systems*, 2012.

[26] Roux L, Racoceanu D, Lomnie N, Kulikova M, Irshad H, Klossa J, Capron F, Genestie C, Le Naour G, and Gurcan MN. Mitosis detection in breast cancer histological images an icpr 2012 contest. *Journal of Pathology Informatics*, 2013.

[27] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–444, 2015.

[28] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004.

[29] Veta M., Van Diest P. J., Willems S. M., Wang H., Madabhushi A., Cruz-Roa A., Gonzalez F., Larsen A. B., Vestergaard J. S., and Dahl A. B. Assessment of algorithms for mitosis detection in breast cancer histopathology images. *Medical Image Analysis*, 2014.

[30] Christopher D. Malon and Eric Cosatto. Classification of mitotic figures with convolutional neural networks and seeded blob features. *Journal of Pathology Informatics*, 2013.

[31] T. Ojala, M. Pietikinen, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29:51–59, 1996.

[32] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016.

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 2015.

[34] Albarqouni Shadi, Baur Christoph, Achilles Felix, Belagiannis Vasileios, Demirci Stefanie, and Navab Nassir. Aggnet: Deep learning from crowds for mitosis detection in breast cancer histology images. *IEEE Transactions on Medical Imaging.*, 2016.

[35] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional models for semantic segmentation. *The IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[36] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261, 2016.

[37] Ryota Tomioka, Taiji Suzuki, and Masashi Sugiyama. Super-linear convergence of dual augmented lagrangian algorithm for sparsity regularized estimation. *The Journal of Machine Learning Research*, 2011.

[38] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Random k-labelsets for multi-label classification. *IEEE Transactions on Knowledge and Data Engineering*, pages 1079–1089, 2011.

[39] S. Valipour, M. Siam, M. Jagersand, and N. Ray. Recurrent fully convolutional networks for video segmentation. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 29–36, March 2017.

[40] Mitko Veta, Paul J. van Diest, Stefan M. Willems, Haibo Wang, Anant Madabhushi, Angel Cruz-Roa, Fabio Gonzalez, Anders B.L. Larsen, Jacob S. Vestergaard, Anders B. Dahl, Dan C. Cirean, Jrgen Schmidhuber, Alessandro Giusti, Luca M. Gambardella, F. Boray Tek, Thomas Walter, Ching-Wei Wang, Satoshi Kondo, Bogdan J. Matuszewski, Frederic Precioso, Violet Snell, Josef Kittler, Teofilo E. de Campos, Adnan M. Khan, Nasir M. Rajpoot, Evdokia Arkoumani, Miangela M. Lacle, Max A. Viergever,

and Josien P.W. Pluim. Assessment of algorithms for mitosis detection in breast cancer histopathology images. *Medical Image Analysis*, pages 237–248, 2015.

[41] Weidi Xie, J. Alison Noble, and Andrew Zisserman. Microscopy cell counting with fully convolutional regression networks. *Deep Learning Workshop in MICCAI*, 2015.

[42] Yao Xue, Nilanjan Ray, Judith Hugh, and Gilbert Bigras. Cell counting by regression using convolutional neural network. *ECCV 2016 workshop on BioImage Computing*, 2016.

[43] Jun Yang, Yu-Gang Jiang, Alexander Hauptmann, and Chong-Wah Ngo. Evaluating bag-of-visual-words representations in scene classification. *the International Workshop on multimedia Information Retrieval*, 2007.