

# Cell Detection in Microscopy Images with Deep Convolutional Neural Network and Compressed Sensing

Yao Xue and Nilanjan Ray

Computing Science, University of Alberta, Canada

## 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. 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. 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. Furthermore, it is possible to stably recover sparse cell locations on the output pixel space from the predicted compressed vector using  $L_1$ -norm optimization. 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 made substantial experiments on several benchmark datasets, where the proposed CNN + CS framework (referred to as CNNCS) achieved the highest or at least top-3 performance in terms of F1-score, compared with other state-of-the-art methods.

**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. It is of significant interest to a wide range of medical imaging tasks and clinical applications. An example is breast cancer, where the tumor proliferation speed (tumor growth) is an important biomarker indicative of breast cancer patients' prognosis. In practical scenario, the most common method is routinely performed by pathologists, who examine histological slides under a microscope based on their empirical assessments, which could be really accurate in several cases, but generally is slow and prone to fatigue induced errors.

Cell detection and localization constitute several challenges that deserves our attention. First, target cells are surrounded by clutters represented by complex histological structures like capillaries, adipocytes, collagen etc. In many cases, the size of the target cell is small, and consequently, it can be difficult to distinguish from the aforementioned 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 as shown in Fig. 1. Additionally, significant variations in the appearance among the targets can also be seen. These challenges render the cell detection/localization/counting problems far from being solved at the moment, in spite of significant recent progresses in computer vision research.

In recent years, object detection has been significantly advanced following the big success by deep learning. However, cell detection or localization task is not simply a sub-task of a general object detection, which typically deals with extended objects, such as humans and vehicles that occupy a significant portion of the field of view in the image. Extended object detection and localization have witnessed much progress in the computer vision community. For example, Region-based Convolutional Neural Networks (R-CNN) [17] and its variants [16], [31], Fully Convolutional Networks (FCN) [34] with recent optimization [30] have become the state-of-the-art algorithms for the extended object detection problem. These solutions cannot be easily translated to cell detection, since assumptions and challenges are different for the latter. For example, for an extended object, localization is considered suc-

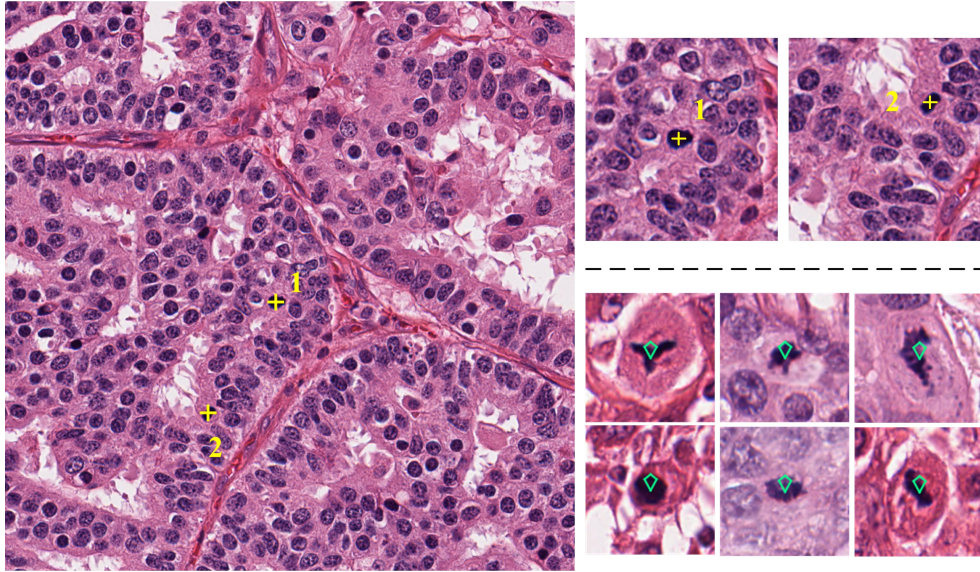


Figure 1: Left picture shows a microscopy image with two target cells annotated by yellow crosses on their centers. Right top pictures give details about the two target cells whose nuclei are in mitotic phase. Right bottom pictures provide more examples of mitotic figures.

successful if a detection bounding box is 50% overlapping with the actual bounding box. For cell detection, tolerance is typically on a much tighter side in order for the localization to be meaningful.

### Conventional cell detection approaches

In the last few decades, different cell recognition methods had been proposed [15]. Traditional computer vision based cell detection systems adopt classical image processing techniques, such as intensity thresholding, feature detection, morphological filtering, region accumulation, and deformable model fitting. For example, Laplacian-of-Gaussian (LoG) [24] operator was a popular choice for blob detection; Gabor filter or LBP feature [29] offers many interesting texture properties and had been attempted for a cell detection task [3].

Conventional cell detection approaches follow a “hand-crafted feature representation” + “classifier” framework. First, detection system extracts one (or multiple) kind of features as the representation of input images. Image processing techniques offer a range of feature extraction algorithms for selection. After that, machine learning based classifiers work on the feature vectors to identify or recognize regions containing target cells. “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. In many cases, it requires significant prior knowledge about the target cells and background.

(2) Most hand-crafted features contain many parameters that are crucial for the overall performance. Consequently, users need to perform a lot of trial-and-error experiments to tune these 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 reaches an accuracy plateau, even when trained with plenty of training data.

### **Deep learning based cell detection approaches**

In comparison to the conventional cell detection methods, 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], [17], [34]. 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 training data [26]. By providing massive amount of 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 deep neural networks. The essential idea behind all these methods is that detectors are trained as a classifier in the image pixel space, either as a pixel labeling [4] or as a region proposal network [8]. Thus, these methods predict the  $\{x, y\}$ -coordinates of cells directly on a 2-D image. Because, target cell locations are sparse in an image, the classifiers in these methods face the class imbalance issue. Moreover, target cells are often only subtly different from other cells. Thus, these

methods tend to produce a significant amount of false positives.

### **Compressed sensing-based output encoding**

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 observation about sparse cell locations within a microscopy image, we are motivated to apply compressed sensing (CS) [12] 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 [11], [36]. In the past, CS-based encoding was used in conjunction with linear and non-linear predictions [19], [23], [21]. We believe, the proposed CNNCS is the first such attempt to solve cell detection and localization that achieved 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 risk of overfitting. Next, there are plenty of opportunities to apply ensemble averages to improve generalization. Furthermore, CS-theory dictates that pairwise distances are approximately maintained in the compressed space [12], [13]. 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. Our contribution is summarized below:

First, this is the first attempt to combine deep learning with CS-based output encoding

to solve cell detection and localization problem.

Second, we try to overcome the aforementioned class imbalance issue by converting a classification problem into a regression problem, where sparse cell locations are distributed by a random projection into a fixed length vector as a target for the regression.

Third, we introduce redundancies in the CS-based output encoding that are exploited by CNN to boost generalization accuracy in cell detection and localization. This redundancies also help to reduce false detections.

Fourth, we demonstrate that the proposed CNNCS framework achieves competitive results compared to the state-of-the-art methods on several benchmark datasets and challenging cell detection contests.

## 2 Background and Related Work

### 2.1 General Object Detection

Prior to deep learning, general object detection pipeline consisted of feature extraction followed by classifiers or detectors. Detection had traditionally been addressed using the hand-crafted features such as SIFT [27], HOG [10], LBP [29], etc. At that time, progress in object detection greatly depended on the invention of more discriminative hand-crafted features. Following the big success of Convolutional Neural Network (CNN) in image classification task [25], deep learning model have been widely adopted in the computer vision community. For example, Region-based Convolutional Neural Networks (R-CNN) [17] and its variants [16], [31], Fully Convolutional Networks (FCN) [34] with recent optimization [30] have become the state-of-the-art algorithms for the extended object detection problems.

Once again, cell detection or localization task is not simply a sub-task of a general object detection, those state-of-the-art solutions for general object detection are able to provide useful clues to right direction, but cannot be easily translated to cell detection.

## 2.2 Cell Detection and Localization

The state-of-the-art methods in detection and localization today include deep learning techniques for cell detection and localization. Recently a deep Fully Convolutional Network (FCN) [34] was proposed for the image segmentation problem and had shown remarkable performance. Soon after the FCN is proposed, [38] presented a FCN-based framework for cell counting, where their FCN is responsible for predicting a spatial density map of target cells, and the number of cells can be estimated by an integration over the learned density map. Slightly similar to [38], a cascaded network [8] has been proposed for cell detection. [8] uses a FCN for candidate region selection, and then a CNN for further discrimination between target cells and background.

In [9], a mitosis detection method has been proposed by CNN-based prediction followed by ad-hoc post processing. As the winner of ICPR 2012 mitosis detection competition, [9] used deep max-pooling convolutional neural networks to detect mitosis in breast histology images. The networks were trained to classify each pixel in the images, using a patch centered on the pixel as context. Then post processing was applied to the network output. In [33], expectation maximization has been utilized within deep learning framework in an end-to-end fashion for mitosis detection. This work presents a new concept for learning from crowds that handle data aggregation directly as part of the learning process of the convolutional neural network (CNN) via additional crowd-sourcing layer. It is the first piece of work where deep learning has been applied to generate a ground-truth labeling from non-expert crowd annotation in a biomedical context.

## 2.3 Compressed Sensing

During the past decade, compressed sensing or compressive sensing (CS) [12] 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 [13], [14], [12], 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 transformation (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 works [13], [14], [12] 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 a convex optimization with a penalty expressed by  $L_1$  norm, for example orthogonal matching pursuit (OMP) [5] and dual augmented Lagrangian (DAL) method [35].

## 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 recovery and detection. The flow chart of the whole framework is shown in Fig. 2.

During training, the ground truth location of cells are indicated by a pixel-wise binary annotation map  $B$ . We propose two cell location encoding schemes, which convert cell location from the pixel space representation  $B$  to a compressed signal representation  $y$ . Then, training pairs, each consisting of a cell microscopy image and the compressed 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 a regression task. 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 estimate the ground truth cell location by performing  $L_1$  minimization recovery on the estimated signal  $\hat{y}$ , with the known sensing matrix.



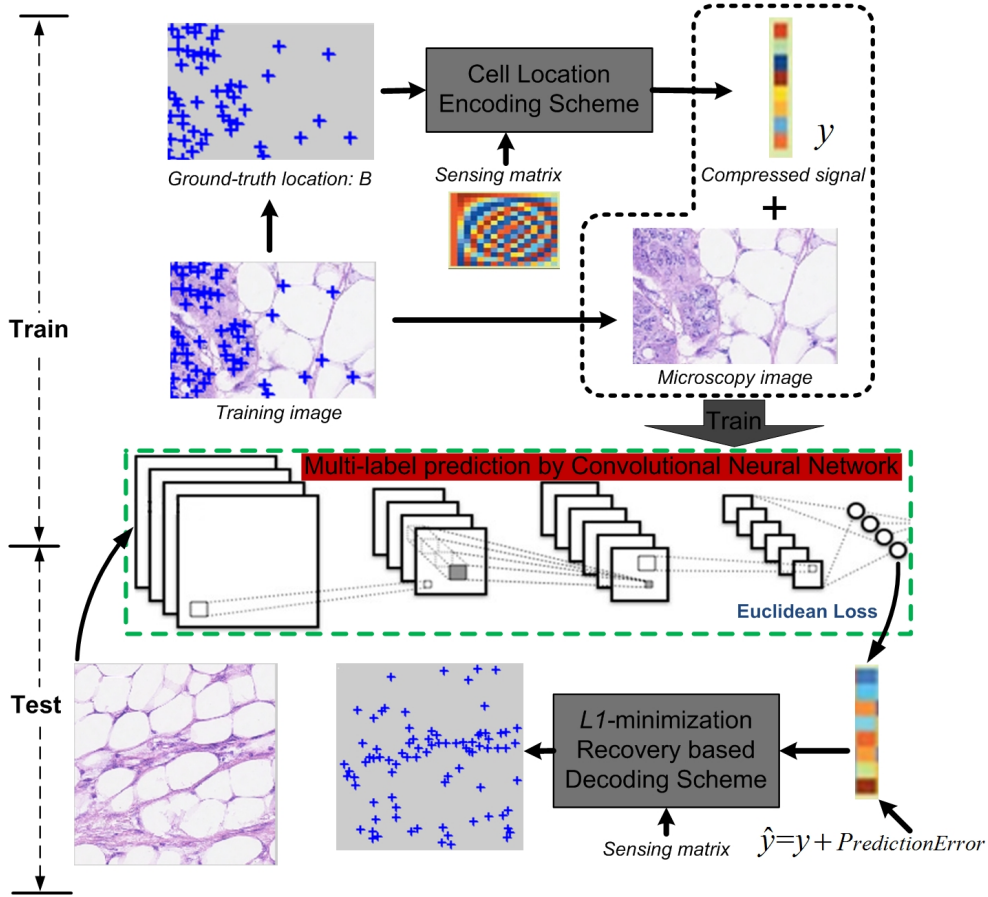


Figure 2: The system overview of the proposed CNNCS framework for cell detection and localization.

## 3.2 Cell Location Encoding and Decoding Scheme

### 3.2.1 Encoding Schemes

In the CNNCS framework, 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  $\Phi$  satisfies a restricted isometry property (RIP) condition [13], [14]. In many cases,  $\Phi$  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:  $M \geq C_M k \log(N)$ , 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  $\mathcal{O}(wh)$ . 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 for 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  $OA = \{oa_l\}, l = 1, 2, \dots, L$ , where  $L$  indicates the total number of observation axes used. The observation axes are uniformly-distributed around an image (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  $y_l$ ,  $l = 1, 2, \dots, L$ , the final encoding result  $y$  is available, which is the joint representation of cells location. The whole encoding process is illustrated by Fig. 3.

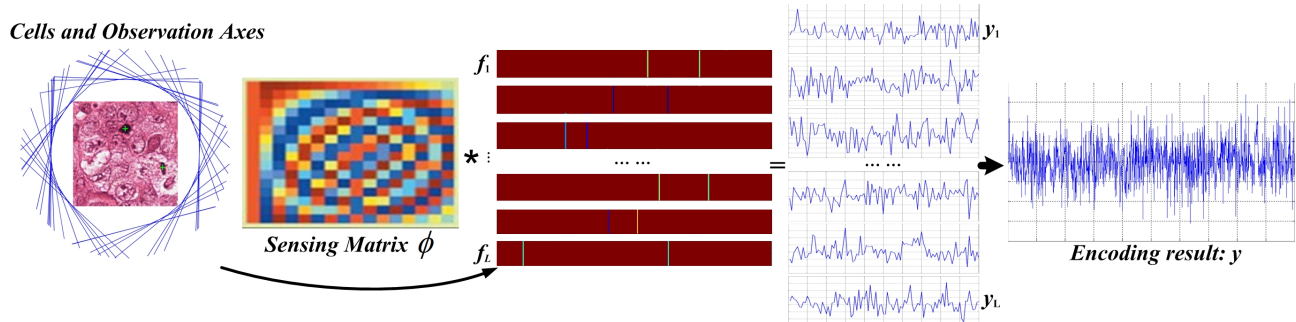


Figure 3: Cell location encoding by signed distances (Scheme-2).

For encoding scheme-2, the size of the sensing matrix  $\Phi$  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. Secondly, the encoding result  $y$  carries **redundant** information about cell locations. 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:

$$\hat{f} = \arg \min_f \|f\|_1 \quad \text{subject to} \quad y = \Phi f \quad (3)$$

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.

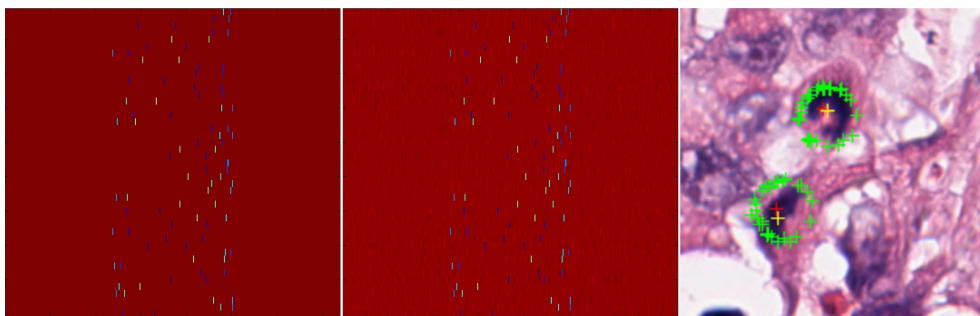


Figure 4: Cell Location Decoding Scheme. From left to right: true location signal  $f$ , decoded location signal  $\hat{f}$  and detection results. Yellow crosses indicate the ground-truth location of cells, green crosses are the candidates points, red crosses represent the final detected points.

### 3.3 Signal Prediction by Convolutional Neural Network

We utilize a 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) [18] 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

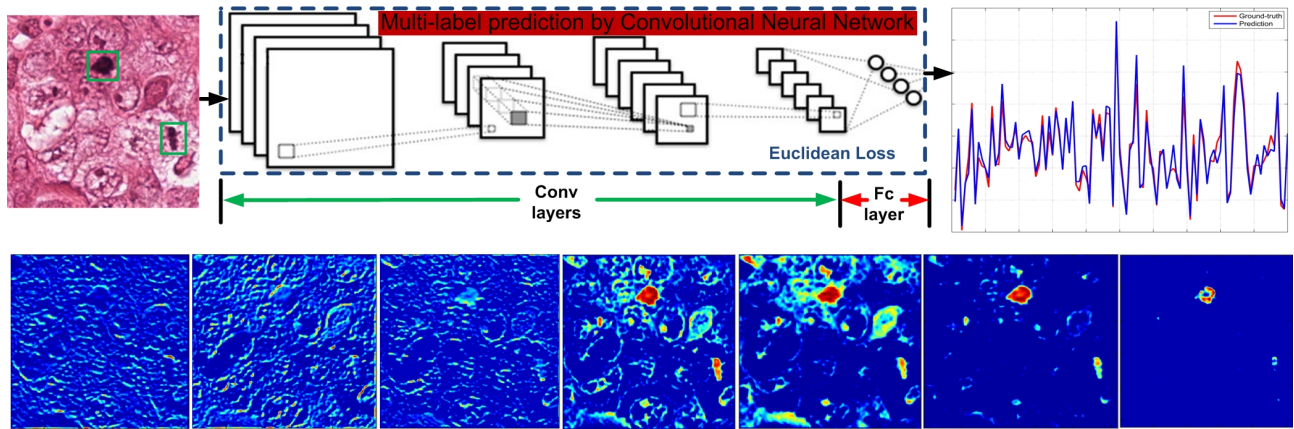


Figure 5: An illustration of the process of signal prediction by convolutional neural network. The bottom row presents the feature maps learned from Convolutional (Conv) layers of the CNN with training process going on. The current CNN follows the AlexNet architecture. These feature maps come from the 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 Fully-connected (Fc) layer of the CNN. From the picture, we can observe that the predicted signal approximate the pattern of ground truth signal well.

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.

To further optimize our CNN model, we apply Multi-Task Learning (MTL) [7]. 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  $\lambda$  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  $\Phi$  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 [13], [14], [12]:

$$(1 - \delta) \|f\| \leq \|\Phi f\| \leq (1 + \delta) \|f\|. \quad (4)$$

Note that if the sensing matrix  $\Phi$  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,

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

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 [19]. 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  $\Phi$  be the sensing matrix. Then the generalization error bound provided in [19] is as follows:

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

where  $C_1$  and  $C_2$  are two small constants and  $\text{sperr}$  measures how well the reconstruction algorithm has worked [19]. 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

We utilize seven cell datasets, on which CNNCS and other comparison methods are evaluated. The 1st dataset [22] involves 100 H&E stained histology images of colorectal adenocarcinomas. The 2nd dataset [6] consists of 200 highly realistic synthetic emulations of fluorescence microscopic images of bacterial cells. The 3rd dataset [39] comprises of 55 high resolution microscopic images of breast cancers double stained in red (cytokeratin epithelial marker) and brown (nuclear proliferative marker). The 4th dataset is the ICPR 2012 mitosis detection contest dataset [32] including 50 high-resolution (2084-by-2084) RGB microscope slides of Mitosis. The 5th dataset [1] is the ICPR 2014 grand contest of mitosis detection, which is a follow-up and an extension of the ICPR 2012 contest on detection of mitosis. Compared with the contest in 2012, the ICPR 2014 contest is much more challenging, which contains way more images for training and testing. The 6th dataset is the AMIDA-2013 mitosis detection dataset [37], which contains 676 breast cancer histology images belonging to 23 patients. The 7th dataset is the AMIDA-2016 mitosis detection dataset [2], which is an extension of the AMIDA 2013 contest on detection of mitosis. It contains 587 breast cancer histology images belonging to 73 patients for training, and 34 breast cancer histology images for testing with no ground truth available. For each dataset, the annotation that represents the location of cell centroids is shown in Fig.6, details of datasets are summarized in Table. 1.

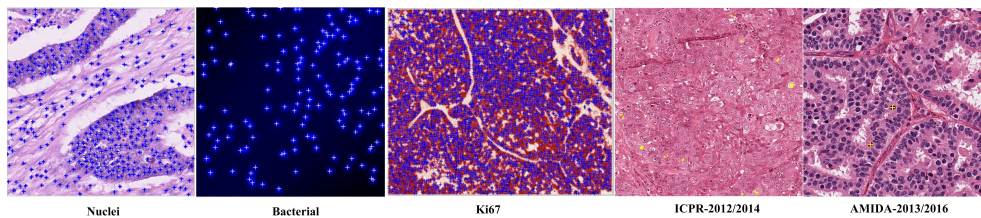


Figure 6: Dataset examples and their annotation.

For evaluation, we adopt the criteria of the ICPR 2012 mitosis detection contest [32], which is also adopted in several other cell detection contests. A detection would be counted as true positive ( $TP$ ) if the distance between the predicted centroid and ground truth cell



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 per image.

Cell Dataset	Size	Ntr/Nte	AC
Nuclei [22]	500×500	50/50	310.22
Bacterial [6]	256×256	100/100	171.47
Ki67 Cell [39]	1920×2560	45/10	2045.85
ICPR 2012 [32]	2084×2084	35/15	5.31
ICPR 2014 [1]	1539×1376	1136/496	4.41
AMIDA 2013 [37]	2000×2000	447/229	3.54
AMIDA 2016 [2]	2000×2000	587/34	2.13

centroid is less than  $\rho$ . Otherwise, a detection is considered as false positives (*FP*). The missed ground truth cells are counted as false negatives (*FN*). In our experiments,  $\rho$  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 performance comparison experiment between CNNCS and three state-of-the-art cell detection methods (“FCN-based” [38], “Le.detect” [4], “CasNN” [8]). 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” [38], 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 on three cell datasets. 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” [38] drops much faster. This can be attributed to the fact that “FCN-based” [38] 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” [38] provides many false detections.

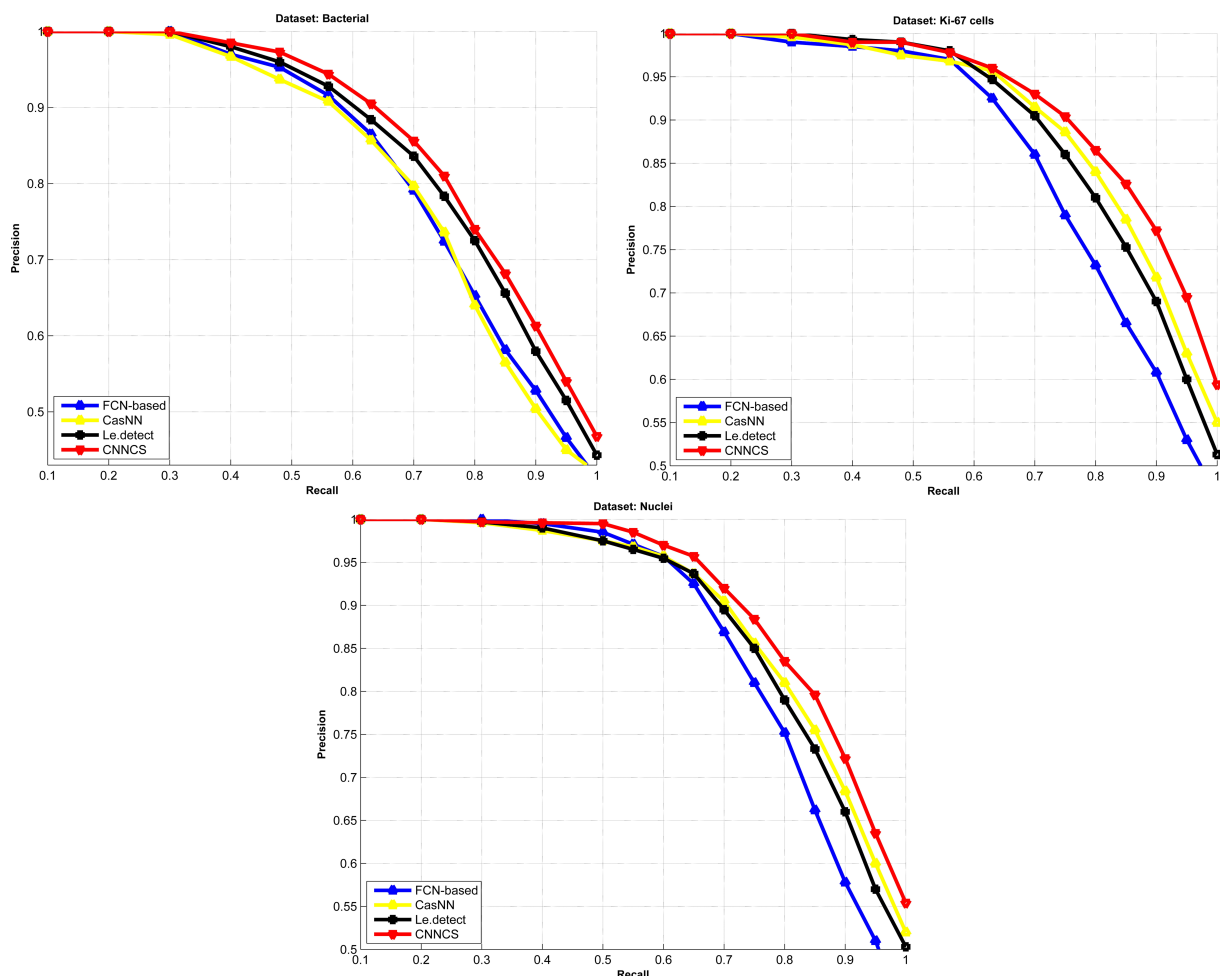


Figure 7: Precision and recall curves of four methods on three datasets.

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 [38] 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 the right top corner of 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 performance 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.

## 4.3 Experiments with Encoding Scheme-2

### 4.3.1 Experiment on ICPR 2012 mitosis detection dataset

To evaluate the performance of encoding scheme 2, we carry out the second group of performance comparison experiments. In the first experiment, we apply the proposed method on the ICPR 2012 mitosis detection 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  $90^\circ$  image rotation is performed on each sub-sample for data augmentation, resulting in a total of 8,960 training dataset. In addition, we perform 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 ( $\lambda$ ) of cell count during MTL. After that, the best performance is achieved when  $M = 112, L = 27, \lambda = 0.20$ . Furthermore, we trained five CNN

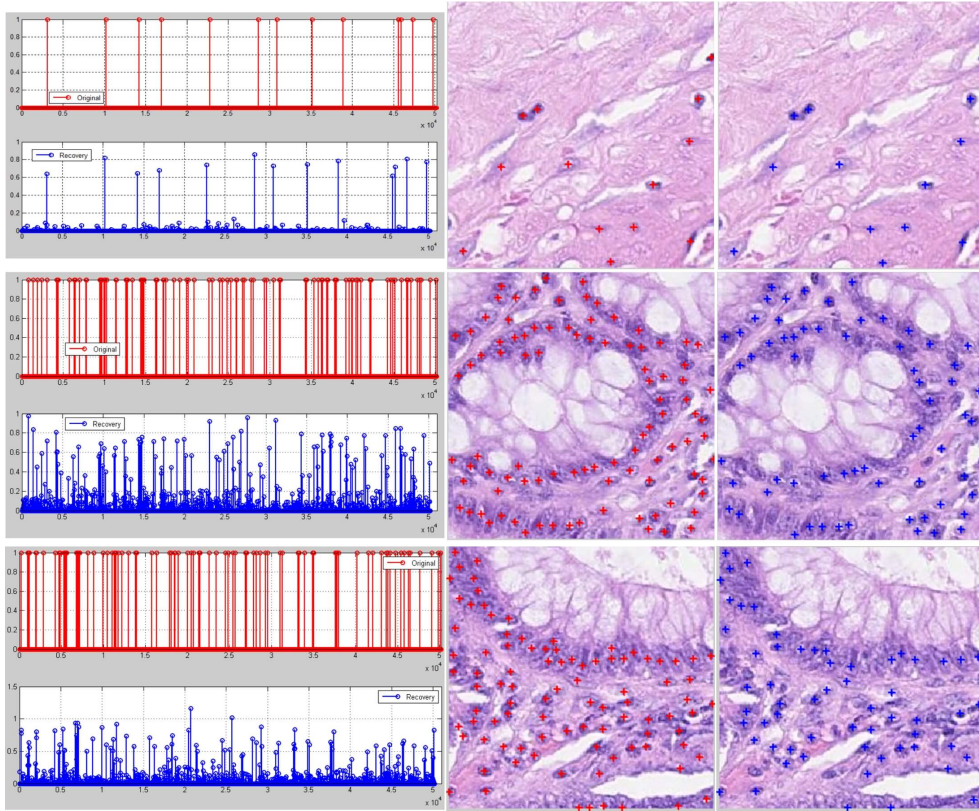


Figure 8: Detection results. Ground-truth: red, Prediction: blue. Left part shows the ground truth signal and the predicted sparse signal that carries the location information of cells; right part shows the ground-truth and detected cells.

models to reduce the performance variance introduced by a single model and to improve the robustness of the whole system. 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 [18]. 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 the experiment, we have explored the performance of CNNCS with different neural network architectures (AlexNet and ResNet). Finally, CNNCS gets the **highest** F1-score among all the comparison methods, details are summarized in Table 2.

Compared to the state-of-the-art method: CasNN-average [8], CNNCS with ResNet and MTL achieved a better performance with  $F_1$ -score 0.837. It can be observed from Table 2 that the precision of our method outperforms the previous best precision by 0.06-0.07, and recall also has recorded about 0.02 improvement. This phenomenon can be attributed to the

Table 2: Results of ICPR 2012 grand challenge of mitosis detection.

Method	Precision	Recall	F <sub>1</sub> -score
UTRECHT	0.511	0.680	0.584
NEC [28]	0.747	0.590	0.659
IPAL [20]	0.698	0.740	0.718
DNN [9]	0.886	0.700	0.782
RCasNN [8]	0.720	0.713	0.716
CasNN-single [8]	0.738	0.753	0.745
CasNN-average [8]	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	<b>0.837</b>

detection principle of our method, where every ground-truth cell is localized with multiple candidate points guaranteed to be 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 other 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 from cell detection and cell counting provides further benefits at negligible additional computations.

#### 4.3.2 Experiment on ICPR 2014 mitosis detection dataset

In the second experiment, we evaluated CNNCS on the ICPR 2014 contest of mitosis detection dataset (also called MITOS-ATYPIA-14), which is a follow-up and an extension of the ICPR 2012 contest on detection of mitosis. Compared with the contest in 2012, the ICPR 2014 contest is much more challenging, which contains more images for training and testing. It provides 1632 breast cancer histology images, 1136 images for training, 496 images for testing. Each image is in the size of  $1539 \times 1376$ . We divide the training images into training set (910 images) and validation set (226 images). We perform random search on the validation set to optimize the hyper parameters. The best performance on MITOS-ATYPIA-14 dataset is achieved when  $M = 103, L = 30, \lambda = 0.24$ . On the test dataset, we have achieved

the **highest** F1-score among all the participated teams. The F1-score of all the participated teams are shown in Table 3. As we see, the CNNCS method shows significant improvement compared to the results of other teams in all the histology slice groups. On an average, CNNCS has almost doubled the F1-score of teams at the second place.

Table 3: Results of ICPR 2014 contest of mitosis detection in breast cancer histological images. F1-scores of participated teams are shown.

Slice group	CUHK	MINES	YILDIZ	STRAS	CNNCS
A06	0.119	0.317	0.370	0.160	<b>0.783</b>
A08	0.333	0.171	0.172	0.024	<b>0.463</b>
A09	0.593	0.473	0.280	0.072	<b>0.660</b>
A19	0.368	0.137	0.107	0.011	<b>0.615</b>
Average	0.356	0.235	0.167	0.024	<b>0.633</b>

### 4.3.3 Experiment on AMIDA 2013 mitosis detection dataset

The third experiment was performed on the AMIDA-2013 mitosis detection dataset, which contains 676 breast cancer histology images, belonging to 23 patients. Suspicious breast tissue is annotated by at least two expert pathologists, to label the center of each cancer cell. We train the proposed CNNCS method using 377 images, validate on 70 training images and test it on the testing set of AMIDA-2013 challenge that has 229 images from the last 8 patients. We employ ResNet as the network architecture with data balancing and MTL in the training set. Similar to previous experiments, we perform random search on the validation set to optimize the hyper parameters. The best performance on AMIDA-2013 dataset is achieved when  $M = 118$ ,  $L = 25$ ,  $\lambda = 0.32$ . Finally among all the 17 participated teams, we achieve the **third highest** F1-score=0.471, which is quite close to the second place, and has a significant improvement over the fourth place method [33]. For details, Table 4 summarizes the comparison between CNNCS and other methods.

Table 4: Results of AMIDA-2013 MICCAI grand challenge of mitosis detection.

Method	Precision	Recall	F <sub>1</sub> -score
IDSIA [9]	0.610	0.612	0.611
DTU	0.427	0.555	0.483
AggNet [33]	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
UNI-JENA	0.007	0.077	0.013
NIH	0.002	0.049	0.003
CNNCS	0.3588	0.5529	0.4352

#### 4.3.4 Experiment on AMIDA 2016 mitosis detection dataset

In the fourth experiment, we participated in the AMIDA-2016 mitosis detection challenge (also called TUPAC16), which is a follow-up and an extension of the AMIDA-2013 contest on detection of mitosis. Its training dataset has 587 breast cancer histology images in size of  $2000 \times 2000$ , belonging to 73 patients. Its test dataset contains 34 breast cancer histology images in the same size without publicly available ground truth labels.

We train the proposed CNNCS method using randomly chosen 470 training images and validate on the remaining 117 training images. Additionally, we apply the following ensemble averaging technique to further increase precision and recall values. Originally, we have partitioned every test image into about 100 non-overlapping patches. Instead of starting the partitioning from the top-left corner, now we set the starting point of the first patch from  $\{\text{offset}, \text{offset}\}$ . 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

Table 5: Results of AMIDA-2016 MICCAI grand challenge of mitosis detection.

Team	F <sub>1</sub> -score
Lunit Inc.	0.652
IBM Research Zurich and Brazil	0.648
Contextvision (SLDESUTO-BOX)	0.616
The Chinese University of Hong Kong	0.601
Microsoft Research Asia	0.596
Radboud UMC	0.541
University of Heidelberg	0.481
University of South Florida	0.440
Pakistan Institute of Engineering and Applied Sciences	0.424
University of Warwick	0.396
Shiraz University of Technology	0.330
Inha University	0.251
CNNCS (on validation set)	0.634

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. Other implementation settings are similar to the settings in the experiment of AMIDA-2013. Finally, we achieved F1-score=0.634 on the validation set (because of the lack of publicly available test set), which is the **third highest** in all the 15 participated teams. Table 5 provides more details of the contest results. Furthermore, Fig.9 provides twelve examples of our detection results in the AMIDA-2016 grand challenge of mitosis detection.

## 5 Conclusion

This is the first attempt demonstrating 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. In this work, we made substantial experiments on several mainstream datasets and



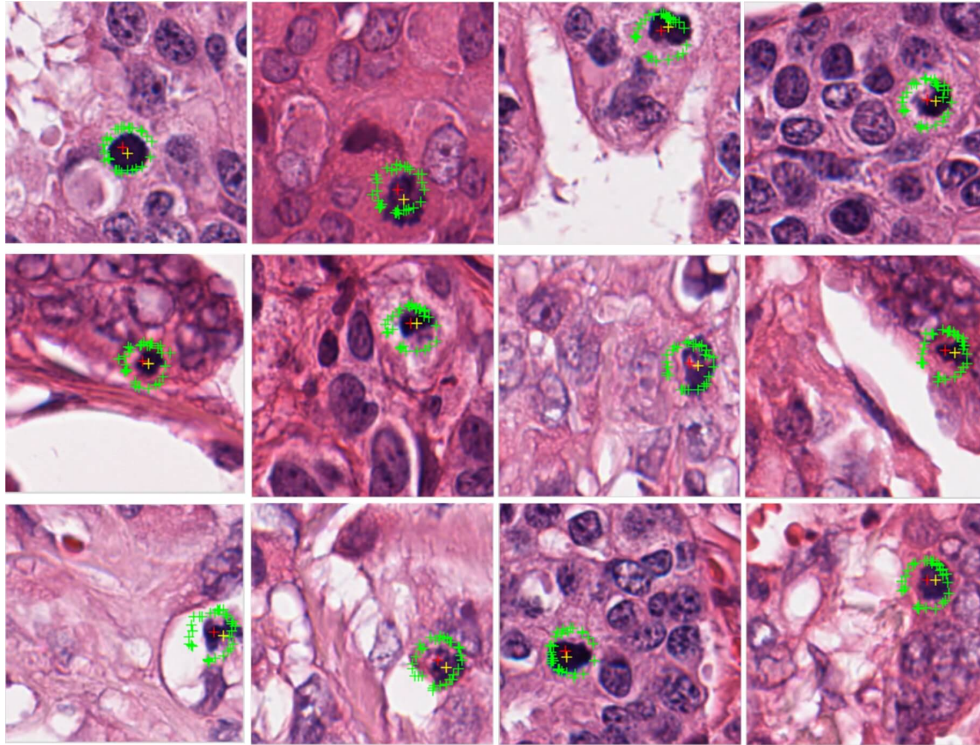


Figure 9: Results on AMIDA-2016 dataset. Yellow cross indicates the ground-truth position of target cells. Green cross indicates cell position predicted by an observation axis. Red cross indicates the final detected cell position, which is the average of all green crosses.

challenging cell detection contests, where the proposed CNN + CS framework (referred to as CNNCS) achieved very competitive (the highest or at least top-3 in terms of F1-score) results compared to the state-of-the-art methods in cell detection task. In addition, the CNNCS framework has the potential to be trained in End-to-End manner, which is our near future plan and could further boost performance.

## References

- [1] <https://mitos-atypia-14.grand-challenge.org/home/>.
- [2] <http://tupac.tue-image.nl/>.
- [3] 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.
- [4] Carlos Arteta, Victor Lempitsky, J. Alison Noble, and Andrew Zisserman. 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] T. Tony Cai and Lie Wang. Orthogonal matching pursuit for sparse signal recovery with noise. *IEEE Transactions on Information Theory.*, 2011.
- [6] Alison Noble Carlos Arteta, Victor Lempitsky and Andrew Zisserman. Learning to count objects in images. *Neural Information Processing Systems.*, 2010.
- [7] Rich Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, Jul 1997.
- [8] 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.
- [9] Dan C. Cirean, Alessandro Giusti, Luca M. Gambardella, and Jrgen Schmidhuber. *Mitosis Detection in Breast Cancer Histology Images with Deep Neural Networks*. 2013.
- [10] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. *Computer Vision and Pattern Recognition*, 2005.
- [11] T. G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, pages 263–286, 1995.

- [12] David L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 2006.
- [13] Justin Romberg Emmanuel Candes. Practical signal recovery from random projections. *IEEE Transactions on Signal Process*, 2005.
- [14] Terence Tao Emmanuel Candesy, Justin Romberg. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory.*, 2006.
- [15] Meijering Erik. Cell segmentation: 50 years down the road [life sciences]. *IEEE Signal Process. Mag.*, 2012.
- [16] Ross Girshick. Fast r-cnn. *International Conference on Computer Vision*, 2015.
- [17] 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.
- [18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Computer Vision and Pattern Recognition.*, 2015.
- [19] Daniel Hsu, Sham M. Kakade, John Langford, and Tong Zhang. Multi-label prediction via compressed sensing. *arXiv:0902.1284v2 [cs.LG]*, 2009.
- [20] Humayun Irshad. Automated mitosis detection in histopathology using morphological and multi-channel statistics features. *Journal of Pathology Informatics*, 2013.
- [21] Arnaud Joly. Exploiting random projections and sparsity with random forests and gradient boosting methods. *arXiv:1704.08067*, 2016.
- [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 classification with deep convolutional neural networks. *Neural Information Processing Systems*, 2012.
- [26] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–444, 2015.
- [27] David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2, ICCV '99*, pages 1150–, Washington, DC, USA, 1999. IEEE Computer Society.
- [28] Christopher D. Malon and Eric Cosatto. Classification of mitotic figures with convolutional neural networks and seeded blob features. *Journal of Pathology Informatics*, 2013.
- [29] 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.
- [30] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016.
- [31] 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.
- [32] Ludovic Roux, Daniel Racoceanu, N. Loménie, Maria S. Kulikova, Humayun Irshad, J. Klossa, Frédérique Capron, Catherine Genestie, Gilles Le Naour, and Metin N. Gurcan. Mitosis detection in breast cancer histological images, an icpr 2012 contest. *Journal of Pathology Informatics*, 4, 05/2013 2013.
- [33] 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.

- [34] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional models for semantic segmentation. *The IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.
- [35] 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.
- [36] 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.
- [37] 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 Cireşan, Jürgen 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*, 20:237–48, 02/2015 2015.
- [38] Weidi Xie, J. Alison Noble, and Andrew Zisserman. Microscopy cell counting with fully convolutional regression networks. *Deep Learning Workshop in MICCAI*, 2015.
- [39] Yao Xue, Nilanjan Ray, Judith Hugh, and Gilbert Bigras. Cell counting by regression using convolutional neural network. *ECCV 2016 workshop on BioImage Computing*, 2016.