The Workshops of the Thirty-Second
AAAI Conference on Artificial Intelligence

Output Encoding by Compressed Sensing for Cell Detection with Deep Convnet

Yao Xue, Nilanjan Ray
yxue2@ualberta.ca, nray1@ualberta.ca
Department of Computing Science, University of Alberta, Canada

Abstract

Output encoding often leads to superior accuracies in various machine learning tasks. In this paper we look at a significant task of cell detection/localization from microscopy images as a test case for output encoding. Since the output space is sparse for the cell detection problem (only a few pixel locations are cell centers), we employ compressed sensing (CS)-based output encoding here. Using random projections, CS converts the sparse, output pixel space into dense and short (i.e., compressed) vectors. As a regressor, we use deep convolutional neural net (CNN) to predict the compressed vectors. Then applying a L_1-norm recovery algorithm to the predicted vectors, we recover sparse cell locations in the output pixel space. We demonstrate CS-based output encoding provides us with the opportunity to do ensemble averaging to boost detection/localization scores. We experimentally demonstrate that the proposed CNN + CS framework (referred to as CNNCS) is competitive or better than the the state-of-the-art methods on benchmark datasets for microscopy cell detection. In the AMIDA13 MICCAI grand competition, we achieve the 3rd highest F1-score in all the 17 participated teams.

Introduction

Output encoding transforms the labels (such as 1-hot vectors in classification) of training examples into a different representation, where an inverse transformation can be applied to recover the original label. Then an ensemble of machine learners is trained to predict the transformed label. Finally, a inverse transformation or decoding is applied to retrieve the original output labels.

One of the earliest works in the output encoding with error correcting ability (Dietterich and Bakiri 1995) had shown superior accuracy. In the recent past, redundancy in the output representation (Tsoumakas and Vlahavas 2007) yielded more accurate predictions. The RAKEL method constructs many random a k-labelsets that are subsets of the multiple labels and trains classifiers for those. Then, it combines the ensemble by voting (Tsoumakas and Vlahavas 2007).

When the output label is sparse, a natural question for output encoding is how to exploit this sparsity. Eventually, output encoding of sparse vectors borrowed an elegant tool called compressed sensing or compressive sensing (CS) (Donoho 2006; Candes and Romberg 2005) from the signal processing community. CS is theoretically and algorithmically rich and has several practical applications, such as reducing MRI scan time for patients (Birns et al. 2016) and building a smaller and cheaper camera (Duarte et al. 2008). Under the premise of CS, an unknown signal of interest is observed (sensed) through a limited number of linear observations. Then, it is possible to obtain a stable reconstruction of the unknown signal from these observations, under a general assumption that the signal is sparse or can be represented sparsely with respect to a linear basis (Candes and Romberg 2005; Donoho 2006). The signal recovery techniques typically rely on convex optimization with a L_1 norm regularization. Examples include orthogonal matching pursuit (OMP) (Cai and Wang 2011) and dual augmented Lagrangian (DAL) method (Tomioka, Suzuki, and Sugiyama 2011).

The principle behind CS-based output encoding is straightforward. First, the (sparse) output label signal is projected to a shorter and dense vector. A machine learner is then trained to regress this short and compressed vector. A recovery algorithm, which is typically a L_1-norm convex optimization, recovers the sparse output vector from the predicted compressed vector.

In the past, CS-based encoding was used in conjunction with linear and non-linear predictions (Hsu et al. 2009; Kapoor, Viswanathan, and Jain 2012; Joly 2016). Hsu et al. (Hsu et al. 2009) proves a generalization error bound for CS-based output encoding. Not surprisingly, the generalization error is bound by the sum of two components- the prediction error from the machine leaner (regressor) and the reconstruction error of the recovery algorithm (Hsu et al. 2009). In particular, Hsu et al. (Hsu et al. 2009) showed that theoretically, using a linear predictor along with CS-based encoding is no more difficult than using a linear predictor alone (Hsu et al. 2009).

Use of non-linear predictors with CS-based output encoding is somewhat recent in machine learning. Viswanathan et al. (Kapoor, Viswanathan, and Jain 2012) used Bayesian inference with CS and showed good accuracy in prediction. Recently, decision trees and gradient boosting have been used in conjunction with CS encoding to yield good prediction accuracies (Joly 2016).

Continuing this trend, in this work, we combine CS-based

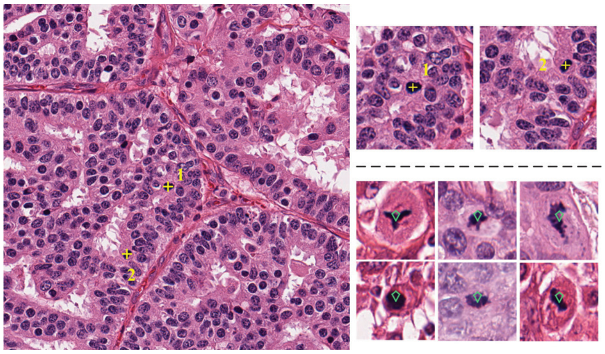

Figure 1: Left picture shows a microscopy image with two target cells annotated by yellow crosses on their centers. Right top pictures give details about the two target cells. Right bottom pictures provide more examples of target cells, where a green diamond is attached to indicate the pixel region of each target cell. This figure is best viewed in color.

output encoding with deep convolutional neural net (CNN). We refer to our proposed framework as CNNCS (convolutional neural network + compressed sensing). There are some advantages of using CS-based output encoding. 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, over-fitting in the CNN would be under control. Thus, compressing the output space can be viewed as a form of regularization on the network. Next, there are plenty of opportunities to apply ensemble average to improve generalization accuracy. In the present work, we exploit this opportunity by creating compressed but redundant representations. Furthermore, CS-theory dictates that pairwise distances in the sparse space are approximately maintained in the compressed space. So, even after the output space encoding, CNN still targets the original output space in an equivalent distance norm.

A significant test application for CNNCS is automatic cell detection from microscopy images. Automatic cell detection is to find whether there are certain types of cells present in an input image and to localize these cells in the image. It is of significant interest to a wide range of medical imaging tasks and clinical applications, such as diagnosis of breast cancer. Fig. 1 shows some examples of microscopy images with annotated cells of interest. To see, why the cell detection application fits the sparsity assumption in CS, consider the following. 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.

Our contributions in this work are as follows. First, this is one of the first attempts to combine deep learning with CS-based output encoding to solve cell detection and localization. Second, we introduce redundancies in the CS-based output encoding that are exploited by CNN to boost accuracy in cell detection and localization. Third, on benchmark datasets CNNCS achieves excellent accuracy compared to the state-of-the-art method. In one such dataset, CNNCS secures the first position and in another it ranks second among its competitors.

Related Work on Cell Detection

Cell detection and localization constitute several challenges that deserve our attention. First, target cells are surrounded by clutters formed by vessels, fat and fibers. And in many cases, the size of the target cell is quite small, consequently, it is hardly distinguishable from neighboring non-target cells. 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 target cells are also present (as shown in Fig. 1). 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 the last few decades, different cell detection methods had been proposed, most of them depended on segmentation of cells. A summary can be found here (Meijering 2012). With the advent of deep learning in computer vision, it is no wonder that the state-of-the-art methods in cell detection are now based on CNN. Recently, a Fully Convolutional Network (FCN) (Shelhamer, Long, and Darrell 2017) was proposed for the image segmentation problem and had shown remarkable performance. Soon after the FCN was proposed, (Xie, Noble, and Zisserman 2015) developed a FCN-based framework for cell detection and counting, where their FCN predicts a spatial cell density map of target 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. This FCN-based method can be thought of a regression predicting in the output (sparse) pixel space.

As the winner of ICPR 2012 mitosis detection competition, (Ciresan et al. 2013) used deep max-pooling convolutional neural networks to detect mitosis in breast histology images. The networks are trained to classify each pixel in the images, using a patch centered on the pixel as context. Then post processing is applied to the network output. Shadi et al. (Albarqouni et al. 2016) 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. More recently, a cascaded network has been proposed for cell detection (Chen et al. 2016) that in principle is similar to the FCN-based method (Xie, Noble, and Zisserman 2015).

Thus, in summary, state-of-the-art methods in cell detection apply deep learning-based regression or classification directly in the output pixel space. In this work, deviating from these approaches, we introduce CS-based output space encoding in the cell detection and localization problem. 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,

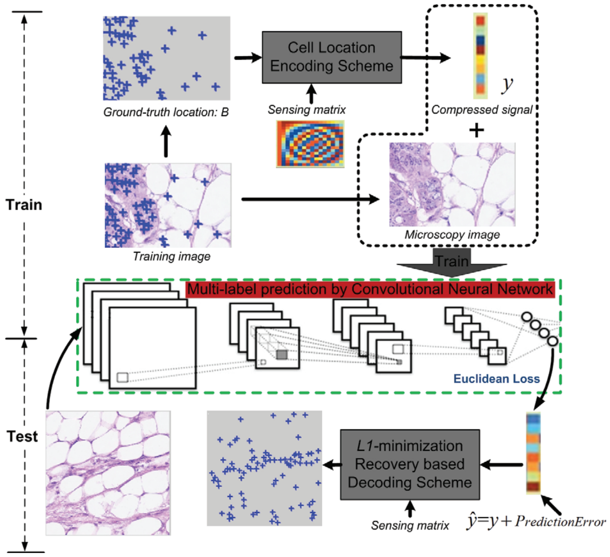

Figure 2: The system overview of the proposed CNNCS framework for cell detection and localization.

compressed vector directly from the input pixels (i.e., microscopy image). Then, L_1 norm optimization is utilized to recover sparse cell locations.

Proposed Method

System Overview

The proposed detection framework consists of three components: (1) cell location encoding phase using random projections, (2) a CNN-based regression model to capture the relationship between a cell microscopy image and the encoded signal, 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. Probably in the simplest form, this encoding may consist of reshaping the 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 compressed and encoded vector y. The encoding scheme can also be more sophisticated as discussed later. Then, each training pair, consisting of a cell microscopy image and the signal y, trains 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.

Cell Location Encoding and Decoding Scheme

Encoding Scheme 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 locations of cells are 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 distances, 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:

$$y_l = \Phi f_l, \tag{1}$$

where, Φ is typically a $M \times N$ random Gaussian matrix. Here, the number of observations M is much smaller than N. 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 cell locations. The whole encoding process is illustrated by Fig. 3.

For this encoding, the size of the sensing matrix Φ is M-by-$\sqrt{w^2 + h^2}$. In comparison, if a reshaping-based encoding was applied, a much larger sensing matrix of size M-by-wh would be required. Also note that the encoding result y carries redundant 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. 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.

Decoding Scheme Recovery of f_l can be estimated from the predicted compressed signal $\hat{y}_l$ by solving the following L_1 norm convex optimization problem:

$$\hat{f}_l = \operatorname*{argmin}_a \|a\|_1 \qquad \text{subject to} \qquad \hat{y}_l = \Phi a \tag{2}$$

In all subsequent experiments, we choose to use the DAL (Tomioka, Suzuki, and Sugiyama 2011) method, because it is very efficient and quite accurate among other algorithms.

After $\hat{f}_l$ 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 present examples of the true location signal f and decoded location signal $\hat{f}_l$. The noisy signed distances of $\hat{f}_l$ 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.

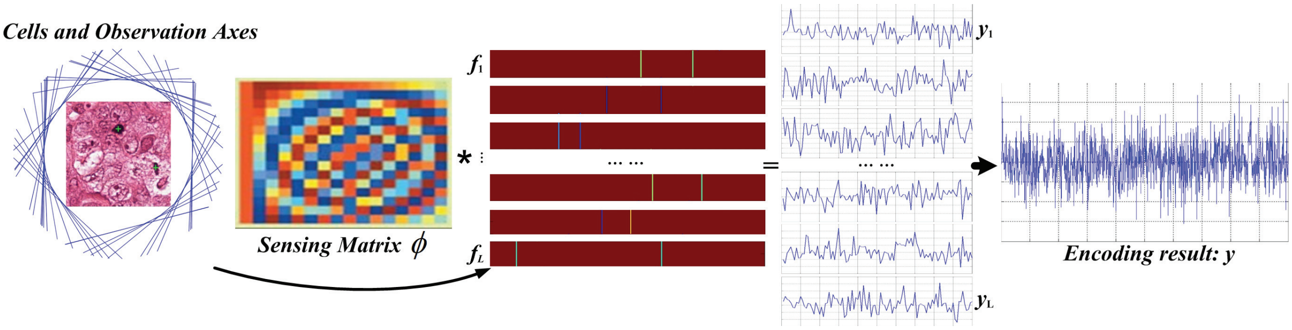

Figure 3: Cell location encoding by signed distances. This figure is best viewed in color.

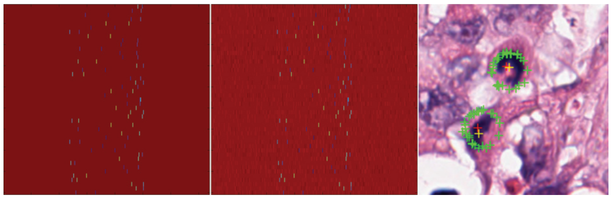

Figure 4: Cell location decoding. From left to right: true location signal f, decoded location signal $\hat{f}$ and detection results. Yellow crosses indicate the ground-truth location of cells, green crosses are the candidates points, red crosses represent the final detected points. This figure is best viewed in color.

Compressed Output Prediction by CNN

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 (Krizhevsky, Sutskever, and Hinton 2012), which consists of 5 convolution layers + 3 fully connected layers; the other is the deep residual network (ResNet) (He et al. 2016) 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, but also provides feature maps at different layers that contain rich information for recognition. Fig. 5 visualizes the learned feature maps, which represent the probabilistic score 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.

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
Mitosis	2084×2084	35/15	5.31	1-19
AMIDA-13	2000×2000	447/229	3.54	0-9

Multi-Task Learning To further optimize our CNN model, we apply Multi-Task Learning (MTL) (Caruana 1997). 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.

Experiments

Datasets and Evaluation Criteria

We experiment with two publicly available benchmark datasets. The first dataset is the 2012 ICPR Mitosis contest dataset (Roux et al. 2013) that has 50 high-resolution (2084-by-2084) RGB microscope slides of Mitosis. The second dataset is the AMIDA-13 challenge dataset (Veta et al. 2015), 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. These datasets with the cell annotation are shown in Fig.6. Some details of datasets are provided in Table.1.

For evaluation, we adopt the criteria of the 2012 ICPR Mitosis contest (Roux et al. 2013), a detection would be counted as true positive (TP) if the distance between the predicted centroid and ground truth cell centroid is less

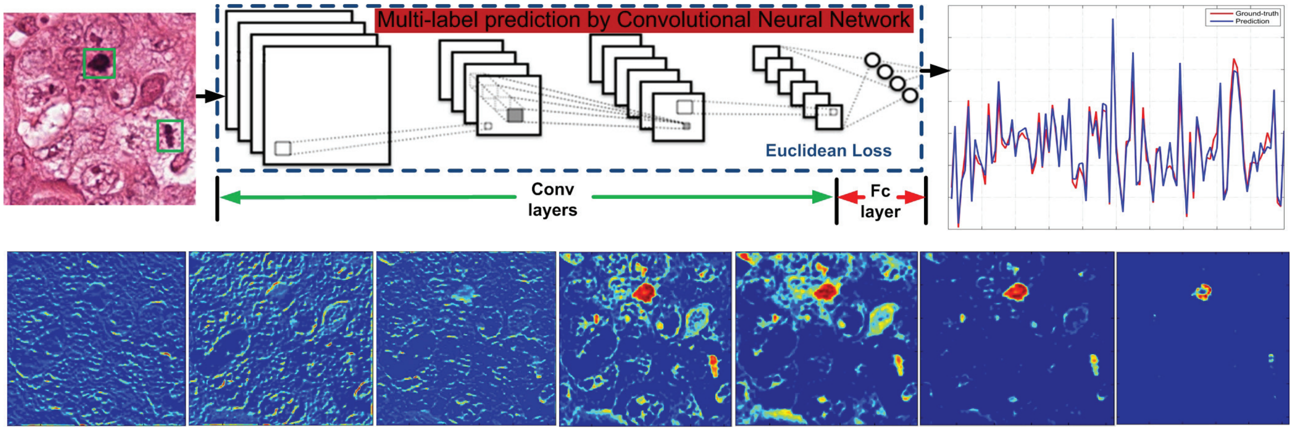

Figure 5: An illustration of signal prediction by convolutional neural network. The bottom row presents the feature maps learned from Convolutional (Conv) layers of the CNN with training process going on. The current CNN follows the AlexNet architecture. These feature maps come from the Conv1, Conv1, 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. It can be observed that the predicted signal approximate the pattern of ground truth signal well.

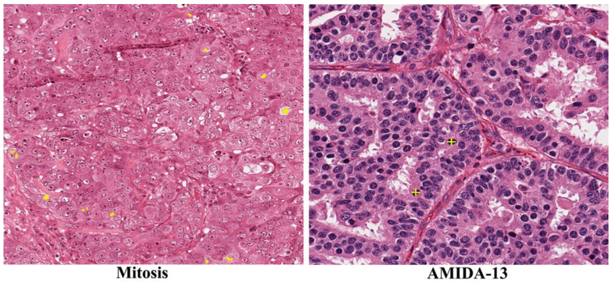

Figure 6: Dataset examples and their annotations.

than ρ. Otherwise, a detection is considered as false positives (FP). The missed ground truth cells are counted as false negatives (FN). In 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.

Since the 2012 ImageNet competition, convolutional neural networks have become popular in large scale image recognition tasks, several milestone networks (including AlexNet (Krizhevsky, Sutskever, and Hinton 2012), VGGNet (Simonyan and Zisserman 2014), 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 (He et al. 2016). 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).

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 90° image rotations were performed on each sub-sampled image for data augmentation. Data augmentation resulted in a total of 8,960 training images. We perform random search with a validation set to tune three hyper parameters: (1) the number of rows in the sensing matrix: M, (2) the number of observation lines: L and (3) the weight (λ) of cell count during MTL. Best performance on the validation set 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 (Chen et al. 2016), 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

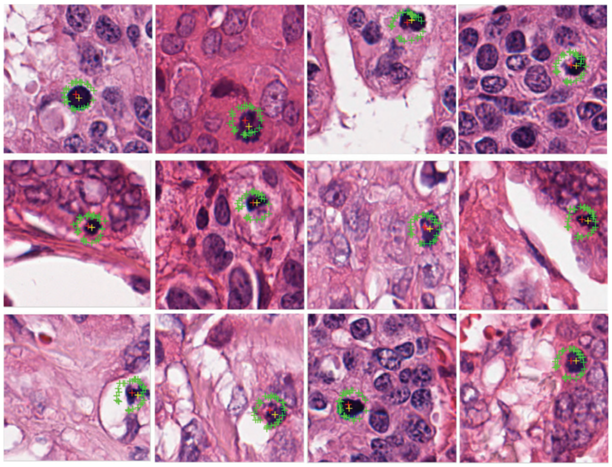

Figure 7: Results on AMIDA-13 dataset. Yellow cross indicates the ground-truth position of target cells. Green cross indicates cell position predicted by an observation axis. Red cross indicates the final detected cell position, which is the average of all green crosses.

cell detection and cell counting provides further benefits at negligible additional computations.

AMIDA-13 dataset (Veta et al. 2015) 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.7 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 (Veta et al. 2015), 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.

Table 2: Results on 2012 ICPR MITOSIS Dataset.

Method	Precision	Recall	F_1-score
UTRECHT (Veta et al. 2015)	0.511	0.680	0.584
NEC (Malon and Cosatto 2013)	0.747	0.590	0.659
IPAL (Irshad 2013)	0.698	0.740	0.718
DNN (Ciresan et al. 2013)	0.886	0.700	0.782
RCasNN (Chen et al. 2016)	0.720	0.713	0.716
CasNN-S (Chen et al. 2016)	0.738	0.753	0.745
CasNN-A (Chen et al. 2016)	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**

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

Method	Precision	Recall	F_1-score
IDSIA (Ciresan et al. 2013)	0.610	0.612	0.611
DTU	0.427	0.555	0.483
CNNCS (our method)	0.3588	0.5529	0.4352
AggNet (Albarqouni et al. 2016)	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

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 challenging 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 in the training. The end-to-end framework has the potential optimize output encoding by modifying random projection matrices adapting to the training data.

References

Albarqouni, S.; Baur, C.; Achilles, F.; Belagiannis, V.; Demirci, S.; and Navab, N. 2016. Aggnet: Deep learning from crowds for mitosis detection in breast cancer histology images. *IEEE Trans. Med. Imaging* 35(5):1313–1321.

Birns, S.; Kim, B.; Ku, S.; Stangl, K.; and Needell, D. 2016. A practical study of longitudinal reference based compressed sensing for MRI. *CoRR* abs/1608.04728.

Cai, T. T., and Wang, L. 2011. Orthogonal matching pursuit for sparse signal recovery with noise. *IEEE Transactions on Information Theory* 57(7):4680–4688.

Candes, E., and Romberg, J. 2005. Practical signal recovery from random projections. *IEEE Transactions on Signal Process.*

Caruana, R. 1997. Multitask learning. *Machine Learning* 28(1):41–75.

Chen, H.; Dou, Q.; Wang, X.; Qin, J.; and Heng, P.-A. 2016. Mitosis detection in breast cancer histology images via deep cascaded networks. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, 1160–1166. AAAI Press.

Ciresan, D. C.; Giusti, A.; Gambardella, L. M.; and Schmidhuber, J. 2013. Mitosis detection in breast cancer histology images with deep neural networks. In Mori, K.; Sakuma, I.; Sato, Y.; Barillot, C.; and Navab, N., eds., *MICCAI (2)*, volume 8150 of *Lecture Notes in Computer Science*, 411–418. Springer.

Dietterich, T. G., and Bakiri, G. 1995. Solving multiclass learning problems via error-correcting output codes. *J. Artif. Int. Res.* 2(1):263–286.

Donoho, D. L. 2006. Compressed sensing. *IEEE Trans. Inf. Theor.* 52(4):1289–1306.

Duarte, M. F.; Davenport, M. A.; Takhar, D.; Laska, J. N.; Sun, T.; Kelly, K. F.; and Baraniuk, R. G. 2008. Single-pixel imaging via compressive sampling. *IEEE Signal Processing Magazine* 25(2):83–91.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778. IEEE Computer Society.

Hsu, D. J.; Kakade, S. M.; Langford, J.; and Zhang, T. 2009. Multi-label prediction via compressed sensing. *CoRR* abs/0902.1284.

Irshad, H. 2013. Automated mitosis detection in histopathology using morphological and multi-channels statistics features. *Journal of Pathology Informatics* 4.

Joly, A. 2016. Exploiting random projections and sparsity with random forests and gradient boosting methods. *arXiv:1704.08067*.

Kapoor, A.; Viswanathan, R.; and Jain, P. 2012. Multilabel classification using bayesian compressed sensing. In Pereira, F.; Burges, C. J. C.; Bottou, L.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc. 2645–2653.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In Pereira, F.; Burges, C. J. C.; Bottou, L.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc. 1097–1105.

Malon, C. D., and Cosatto, E. 2013. Classification of mitotic figures with convolutional neural networks and seeded blob features. *Journal of Pathology Informatics*.

Meijering, E. 2012. Cell segmentation: 50 years down the road [life sciences]. *IEEE Signal Process. Mag.* 29(5):140–145.

Roux, L.; Racoceanu, D.; Loménie, N.; Kulikova, M. S.; Irshad, H.; Klossa, J.; Capron, F.; Genestie, C.; Naour, G. L.; and Gurcan, M. N. 2013. Mitosis detection in breast cancer histological images, an icpr 2012 contest. *Journal of Pathology Informatics* 4.

Shelhamer, E.; Long, J.; and Darrell, T. 2017. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 39(4):640–651.

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

Tomioka, R.; Suzuki, T.; and Sugiyama, M. 2011. Superlinear convergence of dual augmented lagrangian algorithm for sparsity regularized estimation. *The Journal of Machine Learning Research*.

Tsoumakas, G., and Vlahavas, I. 2007. Random k-labelsets: An ensemble method for multilabel classification. In *Proceedings of the 18th European Conference on Machine Learning*, ECML '07, 406–417. Berlin, Heidelberg: Springer-Verlag.

Veta, M.; Diest, P. J. V.; Willems, S. M.; Wang, H.; Madabhushi, A.; Cruz-Roa, A.; Gonzalez, F.; Larsen, A. B. L.; Vestergaard, J. S.; Dahl, A. B.; Cirean, D. C.; Schmidhuber, J.; Giusti, A.; Gambardella, L. M.; Tek, F. B.; andChing Wei Wang, T. W.; Kondo, S.; Matuszewski, B. J.; Precioso, F.; Snell, V.; Kittler, J.; de Campos, T. E.; Khan, A. M.; Rajpoot, N. M.; Arkoumani, E.; Lacle, M. M.; Viergever, M. A.; and Pluim, J. P. 2015. Assessment of algorithms for mitosis detection in breast cancer histopathology images. *Medical Image Analysis* 237–248.

Xie, W.; Noble, J. A.; and Zisserman, A. 2015. Microscopy cell counting with fully convolutional regression networks. In *MICCAI 1st Workshop on Deep Learning in Medical Image Analysis*.