To appear in *Computer Methods in Biomechanics and Biomedical Engineering*
Vol. 00, No. 00, 00 00, 1–14

Microscopy Cell Counting and Detection with Fully Convolutional Regression Networks

Weidi Xie[1*],J. Alison Noble[1] and Andrew Zisserman[1]

[1] *Department of Engineering Science, University of Oxford,UK*

(*Received 00 Month 20XX; accepted 00 Month 20XX*)

This paper concerns automated cell counting and detection in microscopy images. The approach we take is to use Convolutional Neural Networks (CNNs) to regress a cell spatial density across the image. This is applicable to situations where traditional single-cell segmentation based methods do not work well due to cell clumping or overlap.

We make the following contributions: **(i)** we develop and compare architectures for two Fully Convolutional Regression Networks (FCRNs) for this task; **(ii)** since the networks are fully convolutional, they can predict a density map for an input image of arbitrary size, and we exploit this to improve efficiency by end-to-end training on image patches; **(iii)** we show that FCRNs trained entirely on synthetic data are able to give excellent predictions on real microscopy images without fine-tuning, and that the performance can be further improved by fine-tuning on the real images. Finally, **(iv)** by inverting feature representations, we show, to what extent the information from an input image has been encoded by feature responses in different layers.

We set a new state-of-the-art performance for cell counting on standard synthetic image benchmarks, show that the FCRNs trained entirely with synthetic data can generalize well to real microscopy images both for cell counting and detections for the case of overlapping cells.

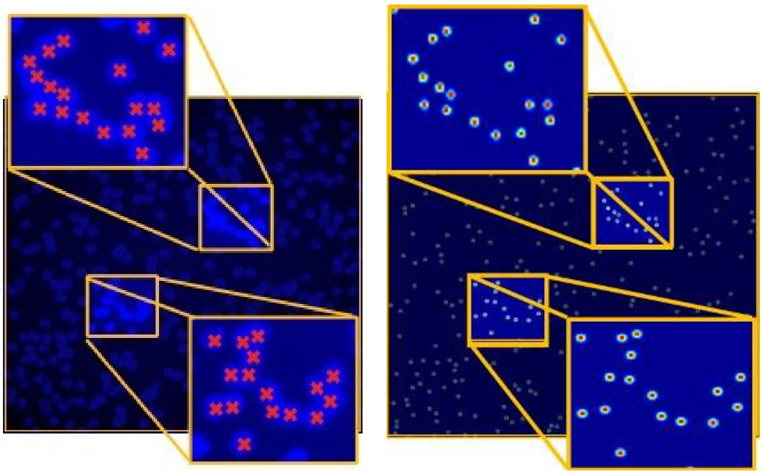
Keywords: Microscopy image analysis, Cell counting, Cell detection, Fully Convolutional Regression Networks, Inverting feature representations.

1. INTRODUCTION

Counting and detecting objects in crowded images or videos is an extremely tedious and time-consuming task encountered in many real-world applications, including biology (Arteta, et al. 2012, 2014, 2015; Fiaschi, et al. 2012), surveillance (Chan, et al. 2008; Lempitsky and Zisserman 2010), and other applications (Barinova, et al. 2012). In this paper, we focus on cell counting and detection in microscopy, but the developed methodology could equally be used in other counting or detection applications. Numerous procedures in biology and medicine require cell counting and detection, for instance: a patient's health can be inferred from the number of red blood cells and white blood cells; in clinical pathology, cell counts from images can be used for investigating hypotheses about developmental or pathological processes; and cell concentration is important in molecular biology, where it can be used to adjust the amount of chemicals to be applied in an experiment. While detection on its own, is able to determine the presence (and quantity) of an object of interest, such as cancer cells in a pathology image, furthermore, detection can be used as seeds for further segmentation or tracking.

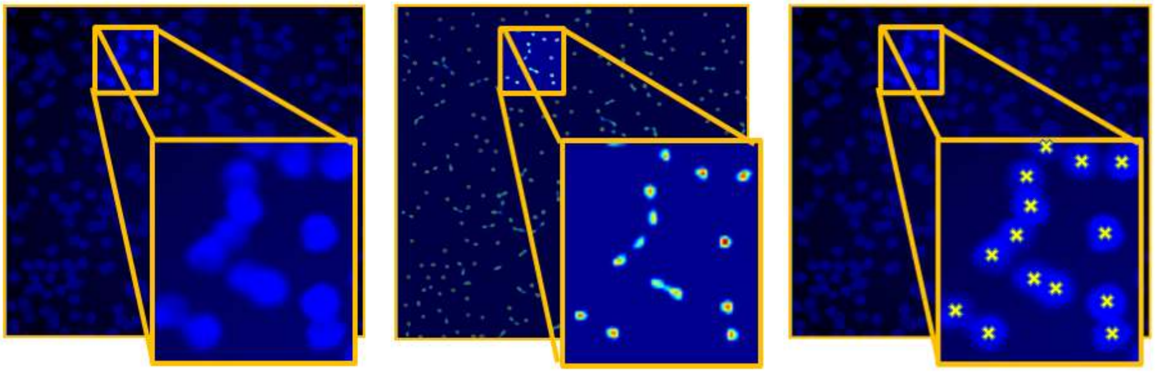
Automatic cell counting can be approached from two directions, one is detection-based counting (Girshick, et al. 2014; Arteta, et al. 2012, 2015), which requires prior detection or segmentation; the other is based on density estimation without the need for prior object detection or segmentation

[*] Corresponding author. Email: weidi.xie@eng.ox.ac.uk (Weidi Xie)



(a) Training Image: $I(x)$ (b) Density Map: $D(x)$

Figure 1.: The training process aims to find a mapping between $I(x)$ and the density map $D(x)$.
(a) Red crosses on $I(x)$ are dot annotations near the cell centres.
(b) The density map $D(x)$ is a superposition of Gaussians at the position of each dot. Integration of the density map $D(x)$ over specific region gives the count of cells.



(a) Test Image: $I'(x)$ (b) Estimated Density Map: $D'(x)$ (c) Cell Detection: $T'(x)$

Figure 2.: During the inference process, given the test image $I'(x)$ in (a)
(b) The trained model aims to predict the density map $D'(x)$.
The integration of the density map $D'(x)$ over a specific region gives the cell counts
(c) Cell detections in $T'(x)$ can be obtained by taking local maxima on the density map $D'(x)$. (**Yellow crosses**)

(Arteta, et al. 2014; Fiaschi, et al. 2012; Lempitsky and Zisserman 2010). In our work, we take the latter approach, and show that cell detection can be a side benefit of the cell counting task.

Following (Lempitsky and Zisserman 2010), we first cast the cell counting problem as a supervised learning problem that tries to learn a mapping between an image $I(x)$ and a density map $D(x)$, denoted as $F : I(x) \rightarrow D(x)$ $(I \in R^{m \times n}, D \in R^{m \times n})$ for a $m \times n$ pixel image, see *Figure 1*. During the inference, given the input test image, the density map and cell detections can be obtained, as shown in *Figure 2*.

We solve this mapping problem by adapting the Convolutional Neural Networks (CNNs) (LeCun, et al. 1998; Krizhevsky, et al. 2012), which has re-emerged as a mainstream tool in the computer vision community. CNNs are also starting to become popular in biomedical image analysis and have achieved state-of-the-art performance in several areas, such as mitosis detection (Cireşan, et al. 2013), neuronal membranes segmentation (Cireşan, et al. 2012), analysis of developing C. elegans embryos (Ning, et al. 2005), and cell segmentation (Ronneberger, et al. 2015). However, they have not yet been applied to solve the target problem here of regression in microscopy cell for counting and detection simultaneously.

In this paper, we develop a Fully Convolutional Regression Networks (FCRNs) approach for regression of a density map. In **section 2**, we describe several related works. In **section 3**, we design and compare two alternative architectures for the FCRNs, and discuss how the networks can be trained efficiently with images of arbitrary sizes in an end-to-end way. In **section 4**, we present results on a standard synthetic dataset for counting, and show that the networks trained only on synthetic data can generalize for different kinds of real microscopy images, and the performance can be further improved by fine-tuning parameters with annotated real data. **Overall**, experimental results show that FCRNs can provide state-of-the-art cell counting for a standard synthetic dataset, as well as the capability for cell detection. And as **an extension** to our previous paper, which was published in the MICCAI 1st Deep Learning Workshop (Weidi, et al. 2015), we also propose to visualize, to what extent the information from input image has been encoded by feature responses of different layers in the trained networks.

2. RELATED WORK

During the past several years, lots of work have been done on counting (Chan, et al. 2008) and detection tasks based on natural images, for instance, detections based on region proposal and classification networks (Girshick, et al. 2014; He, et al. 2014; Ren, et al. 2015), sliding window and classification networks (Sermanet, et al. 2014), regression of heat maps (Tompson, et al. 2014; Pfister, et al. 2015), bounding box regression with CNNs features (Lenc and Vedaldi 2015). In this paper, we are only interested in cell counting and detection, therefore, we briefly review how regression-based methods have been used for these tasks separately, and the fully convolutional networks for semantic segmentation.

2.1 *Counting by density estimation*

Cell counting in crowded microscopy images with density estimation avoids the difficult detection and segmentation of individual cells. It is a good alternative for tasks where only the number of cells is required. Over the recent years, several works have investigated this approach. In (Lempitsky and Zisserman 2010), the problem was cast as density estimation with a supervised learning algorithm, $D(x) = c^T \phi(x)$, where $D(x)$ represents the ground-truth density map, and $\phi(x)$ represents the local features. The parameters c are learned by minimizing the error between the true and predicted density map with quadratic programming over all possible sub-windows. In (Fiaschi, et al. 2012), a regression forest is used to exploit the patch-based idea to learn structured labels, then for a new input image, the density map is estimated by averaging over structured, patch-based predictions. In (Arteta, et al. 2014), an algorithm was proposed that allows fast interactive counting by simply solving ridge regression with various local features.

2.2 *Fully Convolutional Networks*

(Long, et al. 2015) developed the fully convolutional network for semantic segmentation for natural images. By reinterpreting the fully connected layers of a classification net as convolutional, and fine-tuning upsampling filters combined with several skip layers, e.g. feature responses from low level edge-like filters, the networks can actually take an input of arbitrary size and produce a correspondingly-sized output during training and inference process.

2.3 *Detection by regression*

One work that has been developed independently and shares similar ideas to our own on detection is (Yuanpu, et al. 2015). In their work, they cast the detection task as a structured regression problem with the dot annotation near the cell centre. They train CNNs model that takes an image patch of fixed size as input, then predict a so-called proximity patch of half resolution of the original input patch. During training, the defined proximity mask $\mathcal{M}$ corresponding to image I is calculated as,

$$\mathcal{M}_{ij} = \begin{cases} \frac{1}{1+\alpha D(i,j)} & \text{if } D(i,j) \leq \gamma, \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where $D(i,j)$ represents the Euclidean distance from pixel (i,j) to the nearest manually annotated cell centre ($\alpha = 0.8$ and $\gamma = 5$ in their paper). Therefore, $\mathcal{M}_{ij}$ gives value 1 for cell centre, and decreases with the distance from the cell centre. During inference, in order to calculate the proximity map for an entire testing image, they propose to fuse all the generated proximity patches together in a speed-up sliding window way. After this, the cell detection is obtained by finding the local maximum positions in this average proximity map.

In contrast to these previous works, our paper focuses on models that enable end-to-end training and prediction of density map for images of arbitrary size with fully convolutional regression networks. Cell counting and detection in the specific region of microscopy images can then be obtained simultaneously from the predicted density map.

3. FULLY CONVOLUTIONAL REGRESSION NETWORKS (FCRNs)

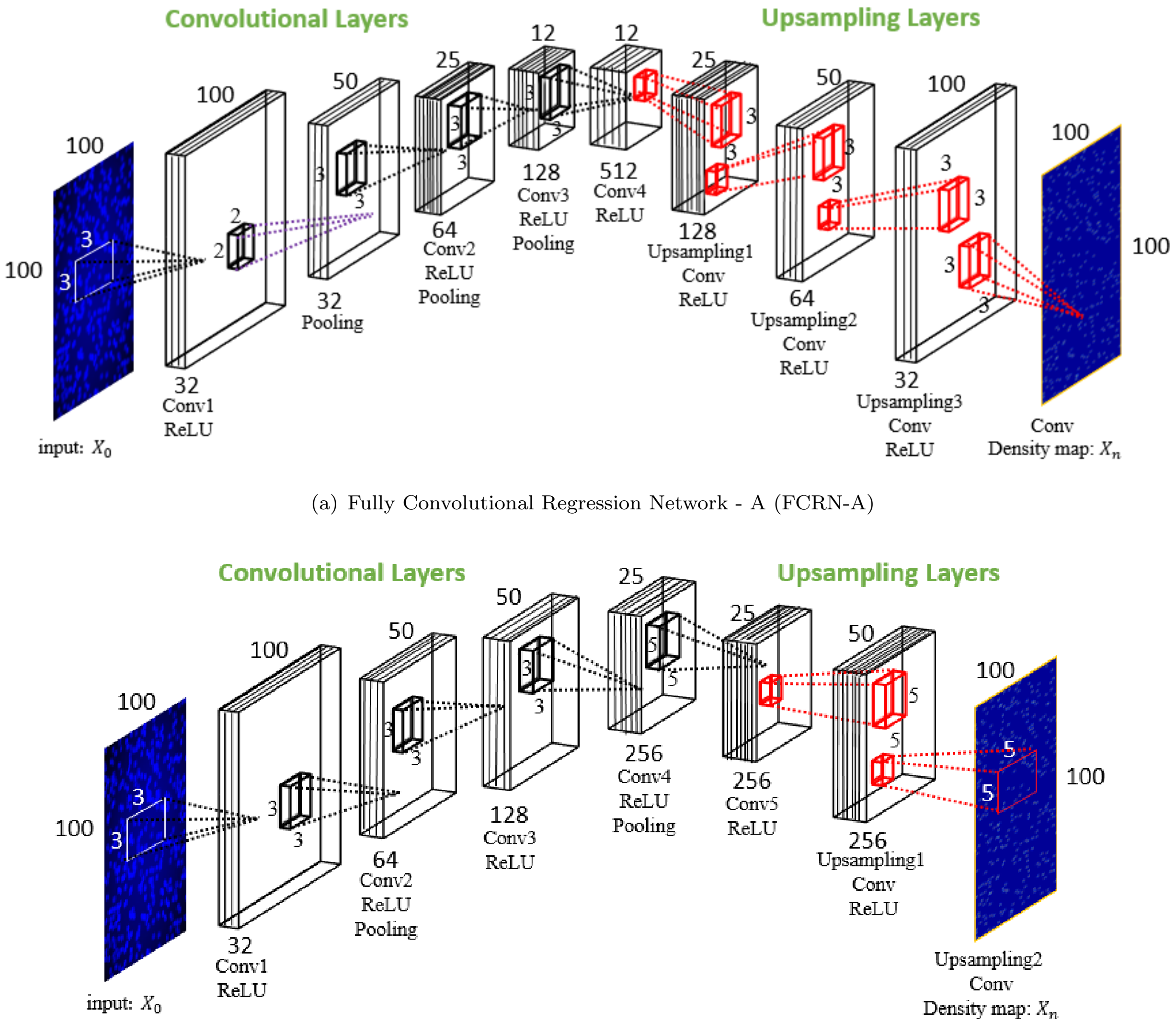
3.1 *Architecture Design*

The problem scenario of cell counting and detection is illustrated in *Figure 1,2*. For training, the ground-truth is provided by dot annotations, where each is represented by a Gaussian, and a density map $D(x)$ is formed by the superposition of these Gaussians. The central task is to regress this density map from the corresponding cell image $I(x)$, then the cell count in a specific region can be obtained by integrating over $D(x)$, and cell detection by local maxima detection on $D(x)$.

In this paper, we propose to solve this problem by training Fully Convolutional Regression Networks (FCRNs). We present two network architectures, namely FCRN-A, FCRN-B, as shown in *Figure 3*. In order to design the network architectures, we factored in the following considerations:

- Unlike the existence of ImageNet (Russakovsky, et al. 2014) in the computer vision field, biomedical image data is limited, expensive, and time-consuming to annotate. Therefore, in this paper, we aimed to train small networks with few parameters. In other words, our goal is to only design networks with simple architectures (no skip layers are used in our network designs).
- For cell counting and detection problems, cells are usually small compared to the whole image. Therefore, deep networks and highly semantic information are not necessary.
- We aim to explore, if networks trained only on synthetic data can generalize to real microscopy images. Related ideas were investigated in (Jaderberg, et al. 2014) for text detection and recognition based on synthetic data, then give good results on real images.
- Although cell shapes in the standard synthetic dataset are roughly the same, cell clumps can have very complicated shapes. We were interested in finding out if the networks are able to deal with cell clumps.

The popular CNN architecture for classification contains convolution-ReLU-pooling (Krizhevsky, et al. 2012). ReLU refers to rectified linear units. Pooling usually refers to max pooling and results



(a) Fully Convolutional Regression Network - A (FCRN-A)

(b) Fully Convolutional Regression Network - B (FCRN-B)

Figure 3.: Fully Convolutional Regression Networks in this paper. (FCRN-A & FCRN-B)
(a): FCRN-A is designed to use small 3 × 3 kernels for every layer. Each convolutional layer is followed by pooling.
(b): FCRN-B is designed to use fewer pooling layers than FCRN-A, 5 × 5 kernels are used.
In each FCRN design:
- The *size* of the input image or feature maps is shown on top of each block, indicating whether pooling has been used.
- The *number of feature maps* in each layer is shown at the bottom of each block.
- The *size* of kernels is shown beside the small black or red blocks.

Conv – Convolution; **Pooling** – 2 × 2 Max pooling;
ReLU – Rectified Linear Units; **Upsampling** – Bilinear Upsampling;

in a shrinkage of the feature maps. However, in order to produce density maps that have same resolution as the input, we reinterpret the fully connected layers as convolutional layers and undo the spatial reduction by performing upsampling-convolution-ReLU, mapping the feature maps of dense representation back to the original resolution (*Figure 3*). During upsampling, we use bilinear interpolation, followed by trainable convolution kernels that can be learnt during end-to-end training.

Inspired by the very deep VGG-net (Simonyan and Zisserman 2015), in both regression networks, we only use small kernels of size 3×3 or 5×5 pixels. The number of feature maps in the higher layers is increased to compensate for the loss of spatial information caused by max pooling. In FCRN-A, all of the kernels are of size 3×3 pixels, and three max-poolings are used to aggregate spatial information leading to an effective receptive field of size 38 × 38 pixels (i.e. the input footprint corresponding to each pixel in the output). FCRN-A provides an efficient way to increase the receptive field, while contains about 1.3 million trainable parameters. In contrast, max pooling is used after every two convolutional layers to avoid too much spatial information loss in FCRN-B. In this case, the number of feature maps is increased up to 256, with this number of feature maps then retained for the remaining layers. Comparing with FCRN-A, in FCRN-B we train 5×5 upsampling kernels leading to the effective receptive field of size 32 × 32 pixels. In total, FCRN-B contains about 3.6 million trainable parameters, which is about three times as many as those in FCRN-A.

3.2 *Implementation details*

The implementation is based on MatConvNet (Vedaldi and Lenc 2015). During training, we cut large images into patches, for instance, we randomly sample patches of size 100×100 pixels from 256×256 images. Simple data augmentation techniques are also used, e.g. small rotations, horizontal flipping. Before training, each patch is normalized by subtracting its own mean value and then dividing by the standard deviation.

The cost function is defined as :

$$l(W;X_0) = \frac{1}{M}\sum_{i=1}^{M}(Y_n - X_n^{(i)})^T(Y_n - X_n^{(i)}) \qquad \text{(Mean Square Error)} \qquad (2)$$

where W are all the trainable parameters, X_0 is the input patch, Y is the ground-truth annotation with Gaussians of $\sigma = 2$, X_n is the predicted density map for the input patch.

The parameters of the convolution kernels are initialized with an orthogonal basis (Saxe, et al. 2014). Stochastic gradient descent (SGD) with momentum are used for optimization. Then the parameters w are updated by:

$$\triangle w_{t+1} = \beta \triangle w_t + (1-\beta)(\alpha \frac{\partial l}{\partial w}) \qquad \text{(Include Momentum)} \qquad (3)$$

where β is the momentum parameter. The learning rate α is initialized as 0.01 and gradually decreased by a factor of 10. The momentum is set to 0.9, weight decay is 0.0005, and no dropout is used in either network. Since the non-zero region in the ground-truth density map is really small, most of the pixels in ground-truth density map remains to be zero. Moreover, even for non-zero regions, the peak value of a Gaussian with $\sigma = 2$ is only about 0.07, the networks tend to be very difficult to train. To alleviate this problem, we simply scale the Gaussian-annotated ground truth (*Figure 1b*) by a factor of 100, forcing the network to fit the Gaussian shapes rather than background zeros.

After pretraining with patches, we fine-tune the parameters with whole images to smooth the estimated density map, since the 100×100 image patches sometimes may only contain part of a cell on the boundary.

4. EXPERIMENTAL VALIDATION

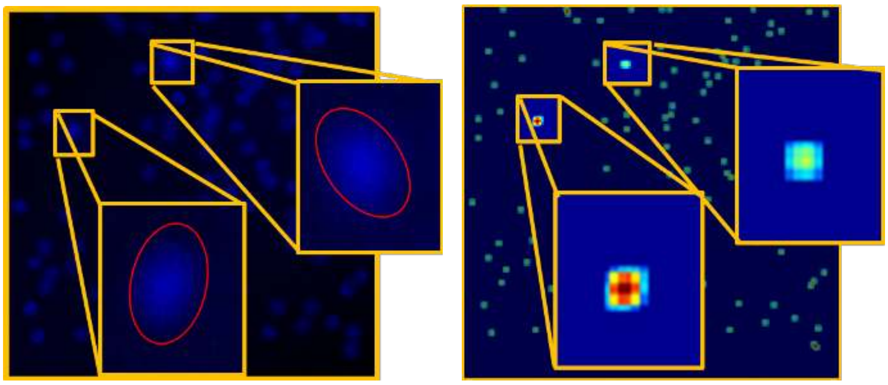
In this section, we first determine how FCRN-A and FCRN-B compare with previous work on cell counting using synthetic data. Then we apply the network trained only on synthetic data to a variety of real microscopy images without fine-tuning. Finally, we compare the performance before and after fine-tuning on real microscopy images.

In terms of cell detection, we consider it as a side benefit of our main counting task. The detection results are obtained simply by taking local maxima based on our predicted density map. We show detection results both on synthetic data and real microscopy images.

4.1 *Dataset and evaluation protocol*

4.1.1 Synthetic data

The synthetic dataset (Lempitsky and Zisserman 2010) consists of 200 images of cell nuclei on fluorescence microscopy generated with (Lehmussola et al., 2007). Each synthetic image has an average of 174±64 cells. Severe overlap between instances are often observed in this dataset, which makes it challenging for counting or detection. As shown in *Figure 4*, under this situation, it even becomes impossible for human expert to tell the difference between overlap cells and single cell. The synthetic dataset is divided into 100 images for training and 100 for testing, and several random splits of the training set are used. Such splits consist of five sets of N training images and N validation images, for N = 8, 16, 32, 64. We report the mean absolute errors and standard deviations for FCRN-A and FCRN-B.



(a) Synthetic Image: $I(x)$ (b) Groundtruth Density Map: $D(x)$

Figure 4.: Annotation noise in the standard dataset.
(a) The image from a standard synthetic dataset. For reader convenience, the rough boundaries of the cells have been manually drawn with a red ellipse.
(b) Put a Gaussian at the centre of each generated cell.
The highlighted *upper-right* region contains one single cell, while the *lower-left* region actually contains two overlapping cells.

4.1.2 Real data

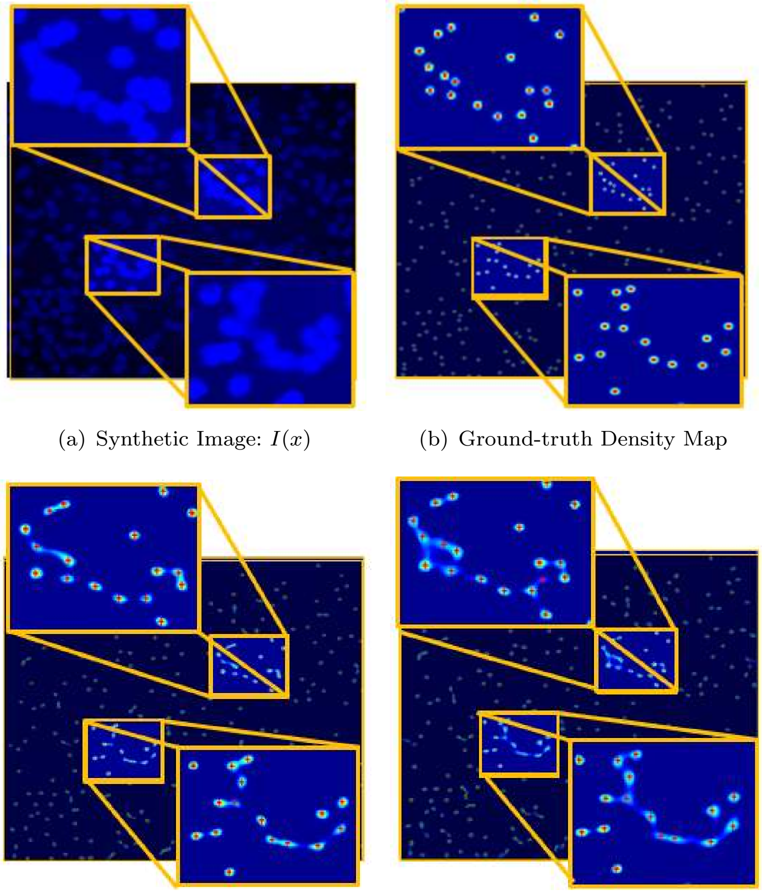
We evaluated FCRN-A and FCRN-B on four different kinds of data; (1) retinal pigment epithelial (RPE) cell images. The quantitative anatomy of RPE can be important for physiology and pathophysiology of the visual process, especially in evaluating the effects of aging (Panda-Jonas, et al. 1996); (2) embryonic stem cells. Cell counting is essential to monitor the differentiation process (Faustino, et al. 2009); (3) plasma cell. The relative number of plasma cells in a bone marrow specimen is a clinical parameter important in the diagnosis and management of plasma cell dyscrasia (Went, et al. 2006); (4) images of precursor T-Cell lymphoblastic lymphoma. Lymphoma is the most common blood cancer, usually occurs when cells of the immune system grow and multiply uncontrollably.

4.2 *Synthetic Data*

4.2.1 Network Comparison

During testing, each image is mapped to a density map first, then integrating over the map for a specific region gives the count, or taking local maxima gives the cell detection of that region (*Figure 5*). The performances of the two networks for cell counting are compared in *Table 1* as a function of the number of training images.

As shown in *Table 1*, FCRN-A performs slightly better than FCRN-B. The *size* of the receptive field turns out to be more important than being able to provide more detailed information *over* the receptive field, we hypothesis that this is because the real difficulty in cell counting lies in regression for large cell clumps, and a larger receptive field is required to span these. For both networks, the performance is observed to improve by using more training images from $N = 8$ to $N = 32$, and only a small additional increase for N to 64.



(a) Synthetic Image: $I(x)$

(b) Ground-truth Density Map

(c) Density Map by FCRN-A

(d) Density Map by FCRN-B

Figure 5.: Counting inference process for pre-trained FCRNs.
(a) Input image from test set.
(b) Ground-truth density map. **Count**: 18 (Upper-left), 16 (Lower-right).
(c) Estimated density map from FCRN-A. **Count**: 17(Upper-left) 16(Lower-right).
(d) Estimated density map from FCRN-B. **Count**: 19(Upper-left) 16(Lower-right).
Red crosses on (c) and (d) indicate cell detection results.

The key three sources of error mainly come from the following:

- The first source of error comes from the dataset itself. As shown in *Figure 4*, the annotation for the dataset itself is noisy. In this case the L2 regression loss tends to over-penalize. In

Method	174±64 cells			
	N=8	N=16	N=32	N=64
Lempitsky and Zisserman (2010)	8.8±1.5	6.4±0.7	5.9±0.5	N/A
Lempitsky and Zisserman (2010)	4.9±0.7	3.8±0.2	3.5±0.2	N/A
Fiaschi, et al. (2012)	**3.4±0.1**	N/A	3.2±0.1	N/A
Arteta, et al. (2014)	4.5±0.6	3.8±0.3	3.5±0.1	N/A
Proposed FCRN-A	3.9±0.5	**3.4±0.2**	**2.9±0.2**	**2.9±0.2**
Proposed FCRN-B	4.1±0.5	3.7±0.3	3.3±0.2	3.2±0.2

Table 1.: *Mean absolute error* and *standard deviations* for cell counting on the standard synthetic cell dataset (Lempitsky and Zisserman 2010; Lehmussola, et al. 2007). The columns correspond to the number of training images. Standard deviation corresponds to five different draws of training and validation sets.

future research, we will investigate other regress loss functions to address this.

- The second source of error is from the boundary effect due to bilinear up-sampling. Cells on the boundary of images tend to produce wrong predictions in this case.
- Thirdly, from very large cell clumps, where four or more cells overlap. In this case, larger clumps can be more variable in shape than individual cells and so are harder to regress; further, regression for large cell clumps requires the network to have an even larger receptive field that can cover important parts of the entire clumps, like concavity information, or curved edges in specific directions. Since our networks are relatively shallow and only have a receptive field of size 38×38 pixels and 32×32 pixels, for elongated cell clumps, their curved edges can usually be covered, and correct predictions can be expected. However, for a roughly round cell clump with four or more cells, it can be bigger than our largest receptive field, and this usually leads to an incorrect prediction.

4.2.2 Comparison with state-of-the-art

Table 1 shows a comparison with previous methods on the synthetic cell dataset. FCRN-A shows about 9.4% improvement over the previous best method of (Fiaschi, et al. 2012) when $N = 32$.

4.3 *Real Data*

We test both regression networks on real datasets for counting and detection. Here, we only show figures for results from FCRN-A in *Figure 6,7,8* (without fine-tuning) and *Figure 9* (before and after fine-tuning). During fine-tuning, two images of size 2500 × 2500 pixels, distinct from the test image, are used for fine-tuning pre-trained FCRNs in a patch-based manner, the same annotations following *Figure 1b* were performed manually by one individual, each image contains over 7000 cells. It can be seen that the performance of FCRN-A on real images can be improved by fine-tuning, reducing the error of 33 out of 1502 (before fine-tuning) to 17 out of 1502 (after fine-tuning).

When testing FCRN-B on two datasets of real microscopy data, for RPE cells: Ground-truth / Estimated count = 705 / 699, and for Precursor T-Cell LBL cells: Ground-truth / Estimated count = 1502 / 1473 (Without fine-tuning). Surprisingly, FCRN-B achieves slightly better performance on real data than FCRN-A. Our conjecture is that the real data contains smaller cell clumps than synthetic data, therefore, the shape of cell clumps will not vary a lot. The network is then able to give a good prediction even with a small receptive field.

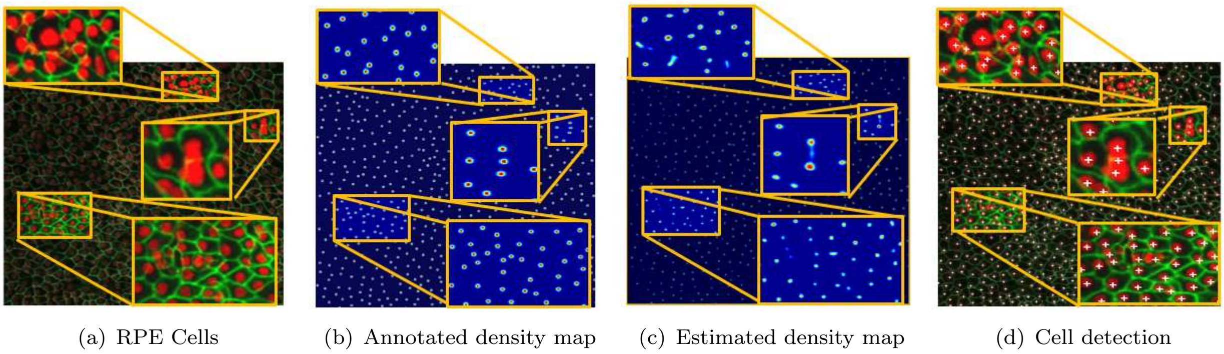

(a) RPE Cells (b) Annotated density map (c) Estimated density map (d) Cell detection

Figure 6.: FCRN-A applied on retinal pigment epithelial (RPE) cells made from stem cells. Only nucleus channel is used.
Cell Count: Ground-truth Vs Estimated : 705 / 697
The data is from: http://sitn.hms.harvard.edu/waves/2014/a-stem-cell-milestone-2/

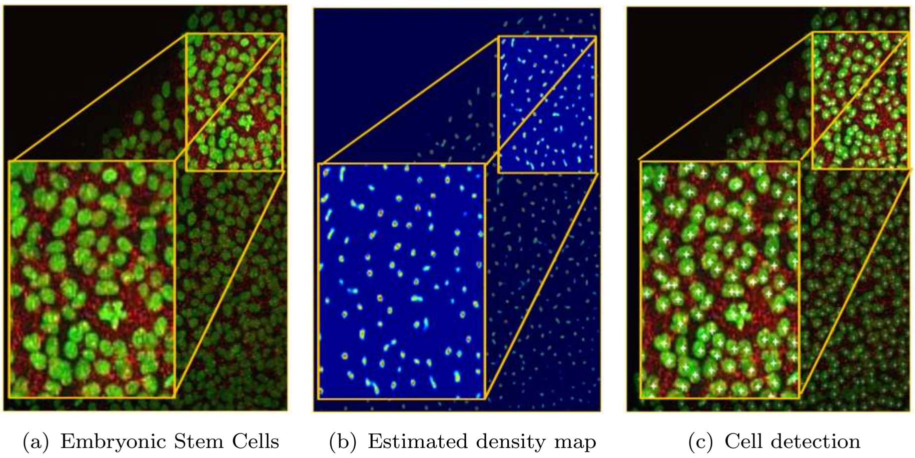

(a) Embryonic Stem Cells (b) Estimated density map (c) Cell detection

Figure 7.: FCRN-A applied on Embryonic Stem Cells. Only nucleus channel is used.
Cell Count: Ground-truth Vs Estimated : 535 / 530

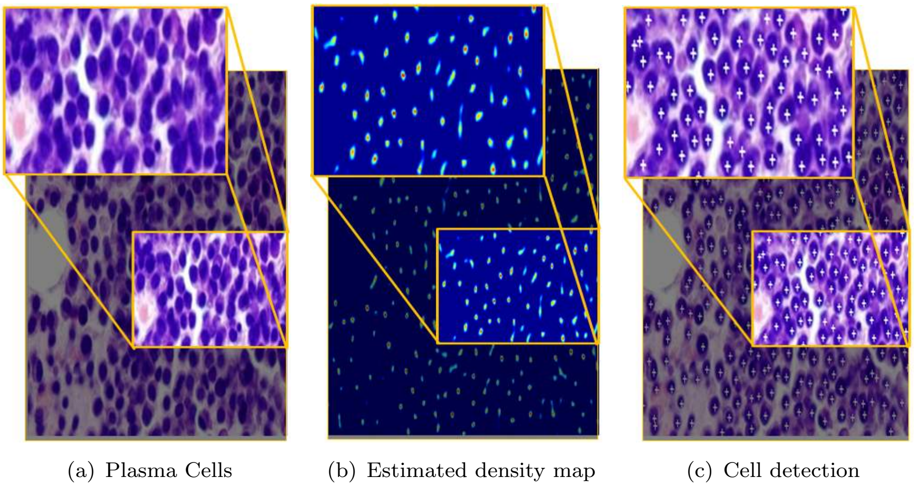
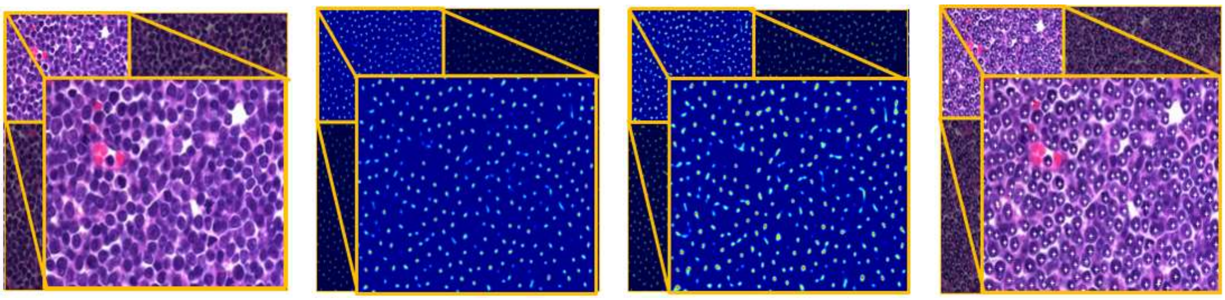

(a) Plasma Cells (b) Estimated density map (c) Cell detection

Figure 8.: FCRN-A applied on Plasma Cells: Only gray-scale image is used.
Cell Count: Ground-truth Vs Estimated : 297 / 294



(a) Precursor T-Cell LBL (b) Estimated density map (Without fine-tuning) (c) Estimated density map (After fine-tuning) (d) Cell detection

Figure 9.: FCRN-A applied on Precursor T-Cell. Only gray-scale image is used.
Cell Count: Ground-truth Vs No fine-tuning 1502 Vs 1469
Cell Count: Ground-truth Vs Fine-tuning 1502 Vs 1485

5. INVERTING FEATURE REPRESENTATIONS

5.1 *Problem Description*

In order to understand the features that have been captured by the deep networks, we considered the following question: "given an encoding of an image, to what extent is it possible to reconstruct that image?" In other words, we sought to visualize how much information of the input image has been captured by the features representations of different layers in the deep networks (Mahendran and Vedaldi 2015).

The problem can be formalized as a reconstruction problem (*Figure 10*). Given a representation function $F : \mathbb{R}^{H\times W\times C} \rightarrow \mathbb{R}^d$ and a representation $\phi_0 = \phi(x_0)$ to be inverted, the reconstruction process aims to find another image $x \in \mathbb{R}^{H\times W\times C}$ that minimizes the objective:

$$x^* = \operatorname*{argmin}_{x\in\mathbb{R}^{H\times W\times C}} l(\phi(x), \phi_0) + \lambda L_2(x) \tag{4}$$

$$l(\phi(x), \phi_0) = \|\phi(x) - \phi_0\|^2 \quad \text{(Euclidean Distance)} \tag{5}$$

where the loss l compares the image representation $\phi(x)$ to the target ϕ_0, and in our case, we choose the L_2 penalty to avoid the large pixel values.

5.2 *Optimization*

Similar to training deep networks, the optimization of equation (4) is also a non-convex problem. However, simple gradient descent(GD) algorithms have been shown to be very effective. In our implementation, *momentum* is also used to speed up the convergence:

$$\Delta x_{t+1} := \beta\Delta x_t - \eta_t \nabla E(x) \tag{6}$$

$$x_{t+1} := x_t + \Delta x_t \tag{7}$$

where $E(x) = l(\phi(x), \phi_0) + \lambda L_2(x)$ is the objective function, weight decaying factor $\beta = 0.9$, learning rate η_t is gradually reduced until convergence.

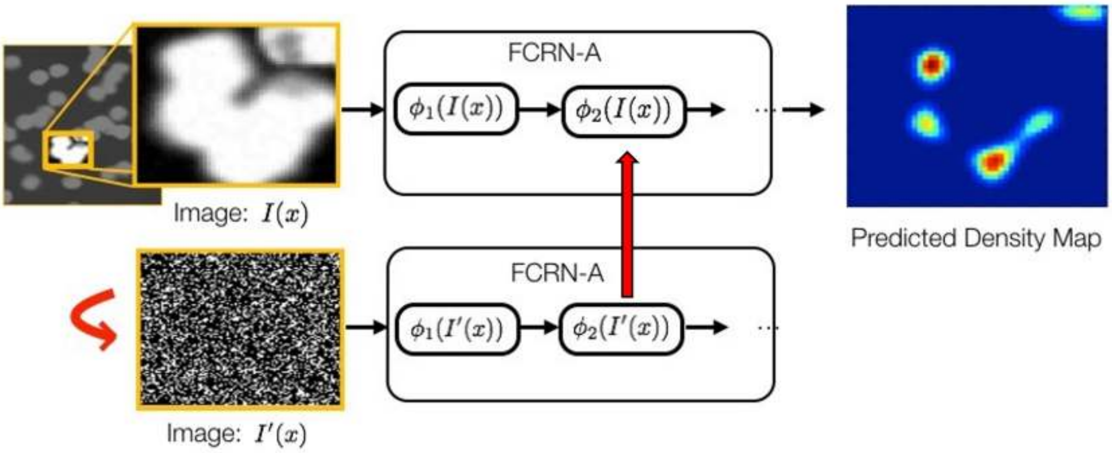


Figure 10.: Example of inverting feature representation for the cell clump based on layer ϕ_2.
Step 1: Feed an input image $I(x)$ to the trained FCRN-A, and make a record of the feature representations $\phi_2(I(x))$.
Step 2: Feed a random input image $I'(x)$ to FCRN-A, similarly, calculate feature representations $\phi_2(I'(x))$.
Step 3: Optimize the random input image $I'(x)$ with gradient descent (GD), such that $\phi_2(I(x)) = \phi_2(I'(x))$. (Shown as the red arrows)

5.3 *Reconstruction Results*

For simplicity, we only show the visualization results from FCRN-A in this paper, but the same procedure can be performed for FCRN-B as well. In essence, CNNs were initially designed as an hierarchical model, which aimed to extract more semantic information as the networks get deeper. For our density map prediction tasks, the biggest challenge is caused by the highly overlapping cell clumps with various shapes. In *Figure 11*, we show, to what extent the information from original cell clump can be encoded by the feature responses of different layers, and try to present an intuition about how the predictions are done by these FCRNs.

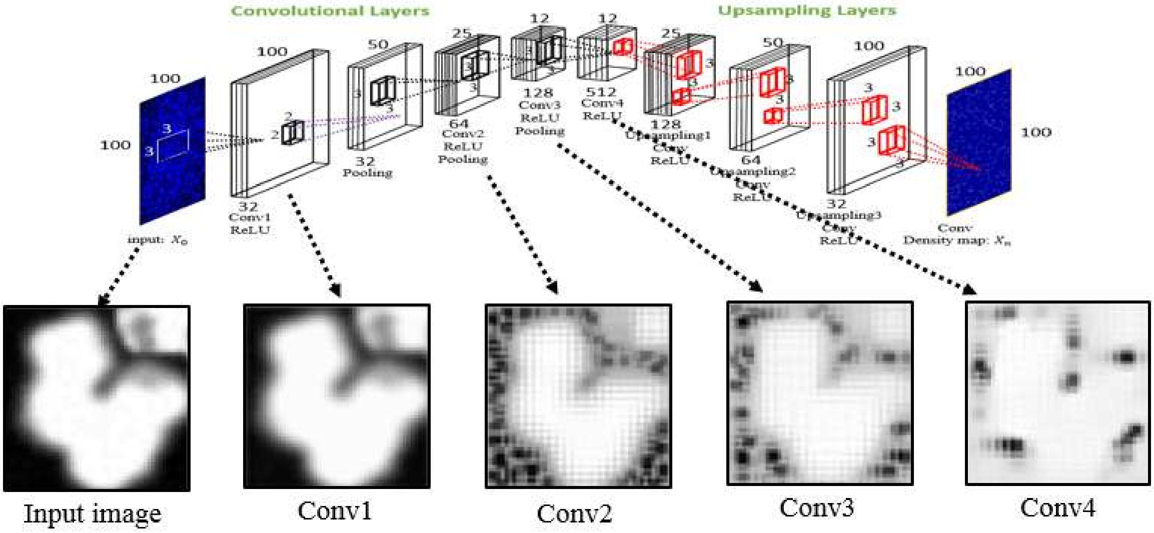


Figure 11.: Reconstruction results from feature representation in different layers of FCRN-A.
As the networks get deeper, feature representations for this cell clump become increasingly abstract. For **Conv4**, which contains most abstract information in this network, only concavity information has been kept for prediction.

6. CONCLUSIONS

In this paper, we have proposed Fully Convolutional Regression Networks (FCRNs) for regressing density maps, which will later be used for both cell counting and detection tasks. The approach allows end-to-end training with images of arbitrary sizes, and is able to perform fast inference for microscopy images. Moreover, we provide intuitive understanding of feature representations from FCRNs by visualizing, to what extent the information has been encoded different layers.

7. ACKNOWLEDGMENT

This work is supported by China Oxford Scholarship Fund, Google Studentship and EPSRC Program Grant: SeeBiByte. Grant Reference: EP/M013774/1.

References

Arteta C, Lempitsky V, Noble JA, Zisserman A. 2012. Learning to detect cells using non-overlapping extremal regions. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2012, Springer. p. 348–356.

Arteta C, Lempitsky V, Noble JA, Zisserman A. 2014. Interactive object counting. In: Proceedings of the European Conference on Computer Vision (ECCV), 2014. p. 504–518.

Arteta C, Lempitsky V, Noble JA, Zisserman A. 2015. Detecting overlapping instances in microscopy images using extremal region trees. Medical Image Analysis (2015), http://dxdoiorg/101016/jmedia201503002.

Barinova O, Lempitsky V, Kholi P. 2012. On detection of multiple object instances using hough transforms. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 2012. 34(9):1773–1784.

Chan AB, Liang ZSJ, Vasconcelos N. 2008. Privacy preserving crowd monitoring: Counting people without people models or tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008. p. 1–7.

Cireşan D, Giusti A, Gambardella LM, Schmidhuber J. 2012. Deep neural networks segment neuronal membranes in electron microscopy images. In: Advances in Neural Information Processing Systems (NIPS), 2012. p. 2843–2851.

Cireşan DC, Giusti A, Gambardella LM, Schmidhuber J. 2013. Mitosis detection in breast cancer histology images with deep neural networks. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2013, Springer. p. 411–418.

Faustino GM, Gattass M, Rehen S, De Lucena CJ. 2009. Automatic embryonic stem cells detection and counting method in fluorescence microscopy images. In: Biomedical Imaging: From Nano to Macro, 2009. ISBI'09. IEEE International Symposium on. p. 799–802.

Fiaschi L, Nair R, Koethe U, Hamprecht FA. 2012. Learning to count with regression forest and structured labels. In: 21st International Conference on Pattern Recognition (ICPR), 2012. p. 2685–2688.

Girshick R, Donahue J, Darrell T, Malik J. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015. p. 580–587.

He K, Zhang X, Ren S, Sun J. 2014. Spatial pyramid pooling in deep convolutional networks for visual recognition. In: Proceedings of the European Conference on Computer Vision (ECCV), 2014. Springer; p. 346–361.

Jaderberg M, Simonyan K, Vedaldi A, Zisserman A. 2014. Synthetic data and artificial neural networks for natural scene text recognition. In: Workshop on Deep Learning, Advances in Neural Information Processing Systems (NIPS), 2014.

Krizhevsky A, Sutskever I, Hinton GE. 2012. ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems (NIPS), 2012. p. 1097–1105.

LeCun Y, Bottou L, Bengio Y, Haffner P. 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE. 86(11):2278–2324.

Lehmussola A, Ruusuvuori P, Selinummi J, Huttunen H, Yli-Harja O. 2007. Computational framework for simulating fluorescence microscope images with cell populations. In: IEEE Transactions on Medical Imaging, 2007; vol. 26. p. 1010–1016.

Lempitsky V, Zisserman A. 2010. Learning to count objects in images. In: Advances in Neural Information Processing Systems (NIPS), 2010. p. 1324–1332.

Lenc K, Vedaldi A. 2015. R-cnn minus r. In: 25th British Machine Vision Conference, 2015.

Long J, Shelhamer E, Darrell T. 2015. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015. p. 3431–3440.

Mahendran A, Vedaldi A. 2015. Understanding deep image representations by inverting them. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015. p. 5188–5196.

Ning F, Delhomme D, LeCun Y, Piano F, Bottou L, Barbano PE. 2005. Toward automatic phenotyping of developing embryos from videos. IEEE Transactions on Image Processing, 2005. 14(9):1360–1371.

Panda-Jonas S, Jonas JB, Jakobczyk-Zmija M. 1996. Retinal pigment epithelial cell count, distribution, and correlations in normal human eyes. American Journal of Ophthalmology, 1996. 121(2):181–189.

Pfister T, Charles J, Zisserman A. 2015. Flowing convnets for human pose estimation in videos. In: IEEE International Conference on Computer Vision (ICCV), 2015.

Ren S, He K, Girshick R, Sun J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems (NIPS), 2015.

Ronneberger O, Fischer P, Brox T. 2015. U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2015, Springer. p. 234–241.

Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, et al. 2014. Imagenet large scale visual recognition challenge. In: International Journal of Computer Vision (IJCV), 2014:1–42.

Saxe AM, McClelland JL, Ganguli S. 2014. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In: International Conference on Learning Representations (ICLR), 2014.

Sermanet P, Eigen D, Zhang X, Mathieu M, Fergus R, LeCun Y. 2014. Overfeat: Integrated recognition, localization and detection using convolutional networks. In: International Conference on Learning Representations (ICLR), 2014.

Simonyan K, Zisserman A. 2015. Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (ICLR), 2015.

Tompson JJ, Jain A, LeCun Y, Bregler C. 2014. Joint training of a convolutional network and a graphical model for human pose estimation. In: Advances in Neural Information Processing Systems (NIPS), 2014. p. 1799–1807.

Vedaldi A, Lenc K. 2015. Matconvnet-convolutional neural networks for matlab. Proceeding of the ACM International Conference on Multimedia, 2015.

Went P, Mayer S, Oberholzer M, Dirnhofer S. 2006. Plasma cell quantification in bone marrow by computer-assisted image analysis. In: Histology and Histopathology, 2006; vol. 21. p. 951–956.

Weidi X, Noble JA, Zisserman A. 2015. Microscopy Cell Counting with Fully Convolutional Regression Networks. In: 1st Deep Learning Workshop, Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2015.

Yuanpu X, Fuyong X, Xiangfei K, Hai S, Lin Y. 2015. Beyond classification: Structured regression for robust cell detection using convolutional neural network. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2015, Springer. p. 358–365.