A Convolutional Neural Network for Spot Detection in Microscopy Images

Matsilele Mabaso[1], Daniel Withey[1] and Bhekisipho Twala[2]
[1]*MDS(MIAS), Council for Scientific and Industrial Research, Pretoria, South Africa*
[2]*Department of Electrical and Mining Engineering, University of South Africa, Pretoria, South Africa*
{MMabaso, DWithey}@csir.co.za, Twalab@unisa.ac.za

Abstract. This paper developed and evaluated a method for the detection of spots in microscopy images. Spots are subcellular particles formed as a result of biomarkers tagged to biomolecules in a specimen and observed via fluorescence microscopy as bright spots. Various approaches that automatically detect spots have been proposed to improve the analysis of biological images. The proposed spot detection method named, detectSpot includes the following steps: 1) A convolutional neural network is trained on image patches containing single spots. This trained network will act as a classifier to the next step. (2) Apply a sliding-window on images containing multiple spots, classify and accept all windows with a score above a given threshold. (3) Perform post-processing on all accepted windows to extract spot locations, then, (4) finally, suppress overlapping detections which are caused by the sliding window-approach. The proposed method was evaluated on realistic synthetic images with known and reliable ground truth. The proposed approach was compared to two other popular CNNs namely, GoogleNet and AlexNet and three traditional methods namely, Isotropic Undecimated Wavelet Transform, Laplacian of Gaussian and Feature Point Detection, using two types of synthetic images. The experimental results indicate that the proposed methodology provides fast spot detection with precision, recall and *F_score* values that are comparable to GoogleNet and higher compared to other methods in comparison. Statistical test between detectSpot and GoogleNet shows that the difference in performance between them is insignificant. This implies that one can use either of these two methods for solving the problem of spot detection.

Keywords: Microscopy Images, Convolutional Neural Network, Spot Detection.

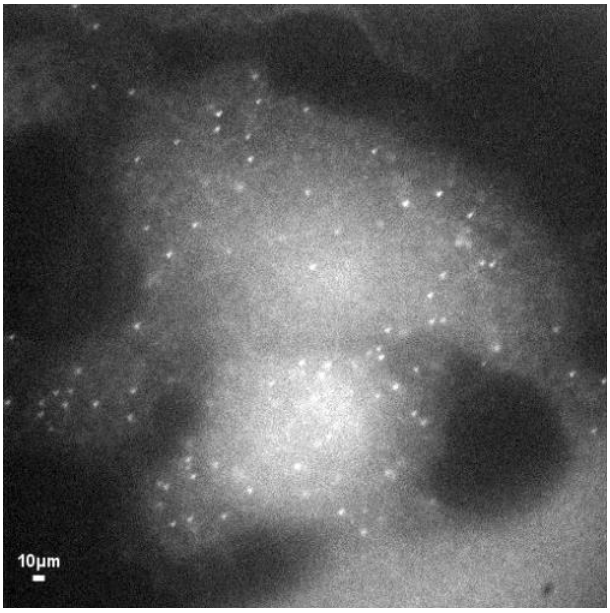

Fig. 1. A sample of real fluorescence image with bright particles obtained using confocal microscopy. [1]

1 INTRODUCTION

The ability to accurately detect and monitor sub-cellular structures in the biological environment has potential in addressing open questions in biology, such as understanding the variations between pathological and normal situations in a cell. Investigation and study of malaria [2], cancer [3], inflammatory processes and wound healing are examples of biological applications which can be tackled in various ways, ranging from biochemistry to microscopy imaging. Detection of objects in images is one of the fundamental computer vision problems that arises in many real-world applications ranging from surveillance [4], robotics [5] to biology [6]. Object detection involves two main steps: (1), classification (determining objects of interest in a given image and, (2), localization (computing the location of these objects in the image).

However, a lot of existing state-of-the-art methods often treat classification and localization separate with localization regarded as a difficult problem compared to classification.

This work focuses on the detection of small bright particles, referred to as spots, in fluorescence microscopy images. These spots may represent, for example, chromosomes, vesicles or genes in a cell depending on the staining method used. A set of identified spots is shown in **Fig. 1**. Accurate detection of spots is of significant interests for biomedical researchers as it plays a significant step for further analysis.

A number of procedures in biology and medicine require the detection and counting of spots, for example, an individual's health can be deduced based on the number of red and white blood cells. The man goal of spot detection is to find all spots in a given image. There exist several challenges which hinders the performance of spot detection methods, such as noise and inhomogeneity which exist in the background. Besides all these challenges, a lot of applications in bioimage analysis such as spot tracking [7], require high performance and reliable detection results which increases the need for efficiency.

In the past years, several methods were developed for detecting spots in fluorescence microscopy images, some of these methods include Wavelets [8], Mathematical morphology [9]. A review of some of these methods can be found in [10, 11]. According to Smal et al. [10], existing methods for spot detection can be categorized into 'supervised' and 'unsupervised' methods. Supervised methods are methods which require labeled data for training while unsupervised methods refer to methods which do not require training. Smal et al. [10] claimed supervised methods gives better detection results when tested to images with low signal-to-noise ratio (SNR).

Convolutional neural network (CNN) is one of the popular in a family of deep learning techniques which based on the ImageNet2012 classification challenge, it managed to reduce the classification error rate by half. According to the study conducted by He et al. [12] a well-trained CNN technique can outperform humans in identifying objects. The CNNs have since been adopted to various applications in computer vision community [13] and medical image analysis [14]. There exist different forms of CNNs architectures in the literature, such as, AlexNet [15], VGGNet [16], ResNet and GoogLeNet [17] among others. Despite the diverse range of their applications in different fields, the application of these methods to biological data is still lacking, especially for the detection of spots. Recent works suggest that CNNs can be used to resolve some of the existing challenges in biology [18, 19].

Currently, there exist no technique based on CNN developed for spot detection in microscopy images. As such, this work proposes a method based on a deep convolutional neural network with sliding-window approach capable of detecting spots in the presence of high levels of noise and high spot density, and with high accuracy in the presence of inhomogeneity in the background.

This paper is organized as follows: Section 2 describes the methodology used in the study, while Section 3 presents the results and finally, Section 4 concludes the paper.

2 MATERIALS AND METHODS

2.1 Methodology

2.1.1 Convolutional Neural Network (CNN)

Convolutional neural network (CNN) h is defined in [1] and described in equation (1) as composition of sequence of L layers $(h_1 \ldots\ldots h_L)$ that maps an input vector x to an output vector y, i.e., [1]

$$\begin{aligned} y &= f(x; w_1, \ldots, w_L) \\ &= h_L(\cdot\,; w_L) \circ h_{L-1}(\cdot\,; w_{L-1}) \circ \cdots \\ &\circ h_2(\cdot\,; w_2) \circ h_1(x\,; w_1) \end{aligned} \tag{1}$$

where w_l is the weight and bias vector for the l^{th} layer h_l and h_l is determined to perform one of the following: a) convolution with a bank of kernels; b) spatial pooling; and c) non-linear activation. For any given N training datasets $\{(x^i, y^i)\}_{i=1}^{N}$, we can estimate the weights, $w_1, \ldots, w_L$ by solving the optimization problem:

$$\underset{w_1,\ldots,w_L}{\text{augmax}} \frac{1}{N} \sum_{i=1}^{N} \ell\left(f(x^i; w_1, \ldots, w_L)\right) \tag{2}$$

Where ℓ is defined as the loss function. The numerical optimization of equation (2) is often performed via backpropagation and stochastic gradient descent methods [20].

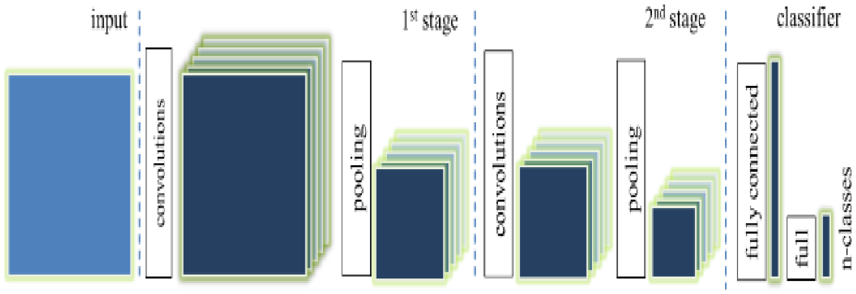

Fig 2. An illustration of the traditional convolutional neural network which consists of two repeatable stages followed by the classifier.

Table 1. Proposed CNN architecture. [1]

Layer	**Kernel size, stride**	**Output $\boldsymbol{w \times h \times c}$**
Input	$-$	$29 \times 29 \times 3$
Convolution	$9 \times 9, 1$	$21 \times 21 \times 32$
ReLu		$21 \times 21 \times 32$
Max-Pool	$2 \times 2, 1$	$20 \times 20 \times 32$
Convolution	$7 \times 7, 1$	$14 \times 14 \times 64$
ReLU		$14 \times 14 \times 64$
Max-Pool	$2 \times 2,1$	$13 \times 13 \times 64$
Convolution	$5 \times 5, 1$	$9 \times 9 \times 80$
ReLu		$9 \times 9 \times 80$
Max-Pool	$2 \times 2, 1$	$7 \times 7 \times 80$
FC	$-$	128
ReLu+Dropout	$-$	128
FC	$-$	128
ReLu+Dropout	$-$	128
FC	$-$	2
Softmax	$-$	2

2.1.2 Problem Formulation

Consider a labeled grayscale training images patches denoted as $I_i \in R^{w \times h \times 3}$, where i ranges from 1 to N with dimensionality $w \times h \times 3$. The task is to develop a classifier based on CNN to predict if patch, I_i contains a spot or not. Image patches with a full spot contained in the image are labelled as positive, otherwise negative.

2.1.3 Proposed CNN

In general, a CNN achitecture can include some of these layers as shown in Fig 2 described below:

(1) **Convolution layers**, A convolutional layer exploits the local information encoded in the image by computing convolutions between the layer's input (e.g., the original image or the output of a previous convolutional layer) and multiple convolution kernels. A convolution refers to the summation of the elementwise dot product of the values between the kernel the input image.

(2) **Pooling or down-sampling layers**. Pooling layers (also known as down-sampling layers) are usually added to the deep network to reduce the dimensionality of the feature maps but retain the most important information and are added just after the convolution layers.
(3) **Fully connected layers (FC)**: After the high-level of features are detected by the preceding convolution and pooling layers, a fully connected layer is attached to at the end of the network with the aim of converting the feature maps to a 1D feature vector. It performs a linear combination of the input vector with a weight matrix.

Given the described building blocks for CNN, we propose CNN architecture for spot detection, named detectSpot as shown in Table 1. detectSpot consists of 5 layers (3 convolution layers and 2 fully connected layers) with learnable weights. Rectified Liner Unit (ReLu) [21] are used as activation function for the first four layers proceeded with softmax for the last layer. To avoid overfitting, dropout with probability of 0.5 for the first two fully connected layers (FC) was introduced. The weights were initialized using truncated random normal. Cross-entropy loss was minimized using Adam optimization with the initial learning rate of 0.001.

2.1.4 Sliding-Window

A sliding window approach is adopted for detecting all spots positions in a given image. A sliding-window is an approach based on moving a rectangular window across an image as illustrated by red and green rectangles in Fig. 3. This is done in order to analyze subpart of the image and extract some information.

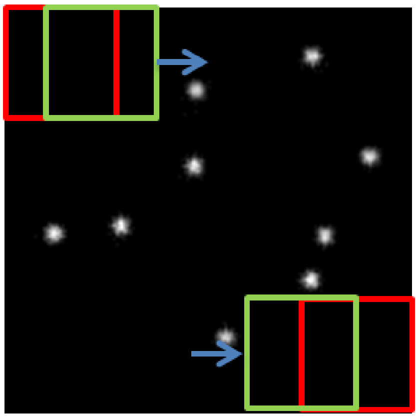

Fig. 3. Illustration of sliding-window approach [1].

2.1.5 Dataset

Image patches of size 29×29 sampled from a synthetic image of size 512×512 were used for training a proposed CNN method. Patches containing spot center were classified as positive while negative patches are those without spot as denoted in equation (3). There was a disproportion between negative patches and positive patches, with the number of negative patches being large. In order to make training and validation set more balanced two measure were considered. Firstly, negative patches were randomly discarded so that there is 50* the number of positive patches, and secondly, each positive patch was rotated resulting in 4 extra patches. In total, 21300 patches formed from images with a signal to noise ratio (SNR) in range (20, 10, 5, 2, 1). Then, these image patches were divided as follows:

- 80% for training
- 20% evaluation

Each of the positive patches has > 0.6 Jaccard-similarity with any ground truth spot while the negative patches has < 0.2 Jaccard-similarity. Jaccard-similarity is denoted as:

$$J(X_{patch}, Y_{ground}) = \frac{|X_{patch} \cap Y_{ground}|}{|X_{patch} \cup Y_{ground}|} \tag{3}$$

2.1.6 Implementation and Training

The proposed CNN was implemented on TFLearn [22], which is a tensorflow [23] wrapper that allows simple implementation and training of deep learning models. Adam [24] was used for the optimization of the algorithm. Linux machine with 16GB RAM and Nvidia GTX680 running TFLearn (v0.3) and tensorflow (v1.3.0) was used for training the network.

2.2 Detection of Spots in Test Images

The proposed CNN architecture, detectSpot is trained to classify an image patch as containing a spot or not. Fig. 4 illustrates the pipeline for the detection of spots using detectSpot method including some post-processing steps. To detect all spots in a given image, a window of size $(w \times h)$ is run through the image. At each iteration, the extracted window is passed onto a detectSpot to compute a probability *S*, which defines whether a spot is contained in the sub-window. Then, if *S* is bigger than a given threshold *T*, the corresponding window is considered to contain spot. All the windows which were classified as containing spots, were subject for further processing to get spot centers (x, y) including the bounding circles marking the spots location in an image. The proposed detectSpot contains two main important parameters, window-size $(w \times h)$ and stride. These parameters influence both speed and detection rate. This approach can only detect spots with fixed size but it can be extended to spots with different sizes by introducing image pyramids. If one select a small stride value, e.g. stride = 1, this will give many overlapping detections of the same spot but at slightly different positions. So, to overcome this challenge, we developed a method that is capable of removing overlapping detections. The proposed approach group all nearby detections so that every spot is detected once. More details about this technique can be found in [1].

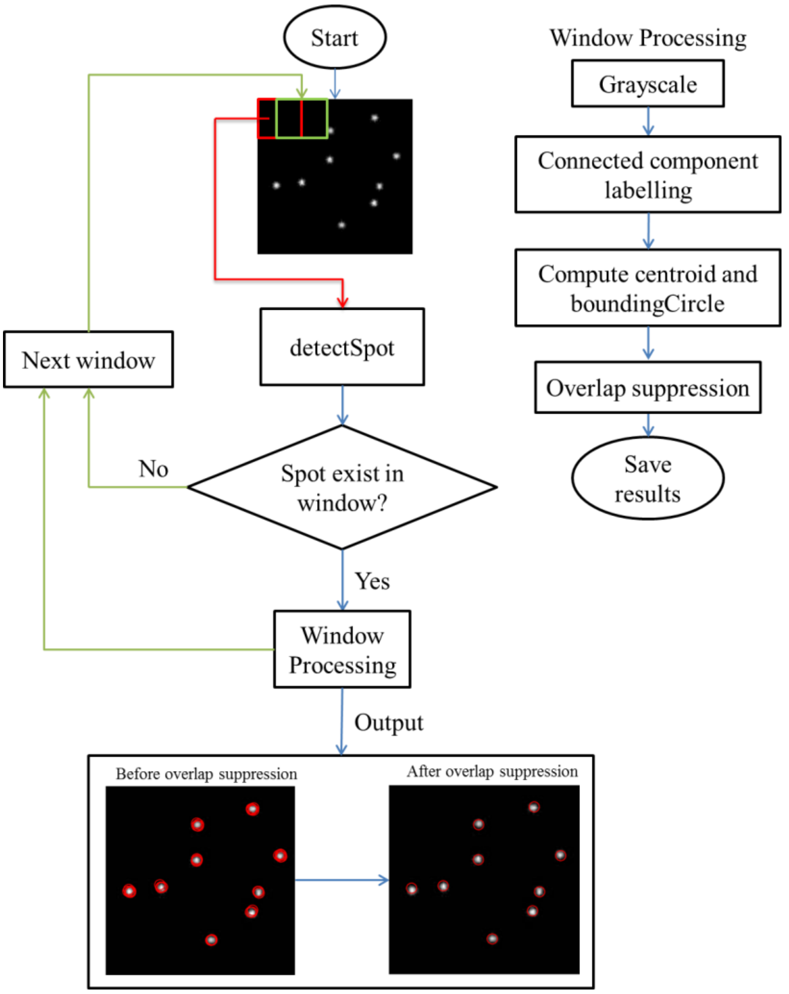

Fig. 4. The proposed architecture for spot detection in microscopy images [1].

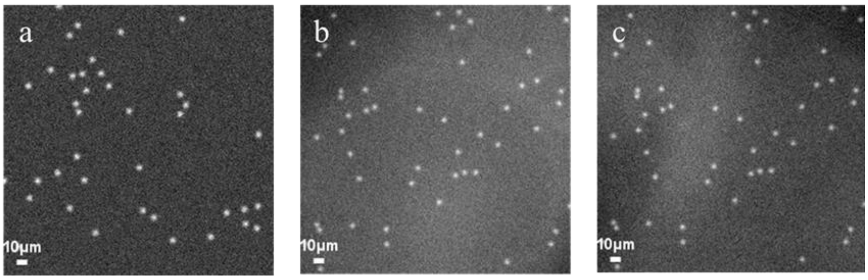

Fig. 5. Examples of synthetic images used for testing with approximately 50 spots per image. (a) Type A, (b) Type B1, and (c) Type B2 [1].

2.2.1 Using Pre-Trained Models

2.2.1.1 Pre-Trained Models

The proposed detectSpot method was quantitatvely compared to two other CNNs methods, namely, AlexNet and GoogleNet.

AlexNet: This method was proposed by Krizhevsky et al. [15] and won the ImageNet ILSVRC-2012 challenge. The model is made-up of 8 layers (5 convolutional layers and three fully connected layers).

GoogleNet: This method won the ImageNet ILSVRC-2014 challenge and it was proposed by Szegedy et al. [17] from Google. This network has 12X fewer parameters compared to AlexNet yet deeper (22 layers). The main contribution of GoogleNet is the introduction of inception module.

2.3 Synthetic Datasets and Evaluation Criteria

2.3.1 Synthetic Test Datasets

To evaluate the performance of the methods, three kinds of synthetic datasets (Type A and Type B1 and B2) containing spots were used. These images were created using a framework proposed in [25]. Synthetic images are important because they contain ground truth information of each spot in an image, as a result this will demonstrate the effectiveness of the proposed detectSpot model as shown in Fig. 5. Each image contained 50 spots cluttered on the background of size 256×256 pixels. The,, a Gaussian noise was then added to the image dataset was corrupted by white noise. The following signal to noise ratios (SNR) levels was explored {10, 8, 6, 4, 2, 1} where the spot intensity was 20 gray levels. The signal to noise ratio is defined as of spot intensity, SP_{max}, divided by the noise standard deviation, σ_{noise},

$$SNR = \frac{SP_{max}}{\sigma_{noise}} \tag{4}$$

The Icy-plugin [26] was used to randomize the spots position in order to mimic properties available in real microscopy images. MATLAB was used to add spots and the OMERO.matlab-5.2.6 toolbox [27] was used to read and save images.

2.3.2 Evaluation Criteria

The well-known measures for evaluating various spot detection methods in microscopy images are F-measure, precision, and recall. Parameters involved in the computations include*;* *TP*, *FP* and *FN* which denote the number of true positives (number of detected spots that corresponds to the ground-truth), number of false positives (number of detected spots which do not correspond to the ground-truth) and number of false negatives (number of missed ground-truth spots), respectively. A detection result is labelled as *TP* if the overlap region, O_r between the detection and ground-truth exceeds a predefined threshold *T*. Explanation of these measures ca be found in [1].

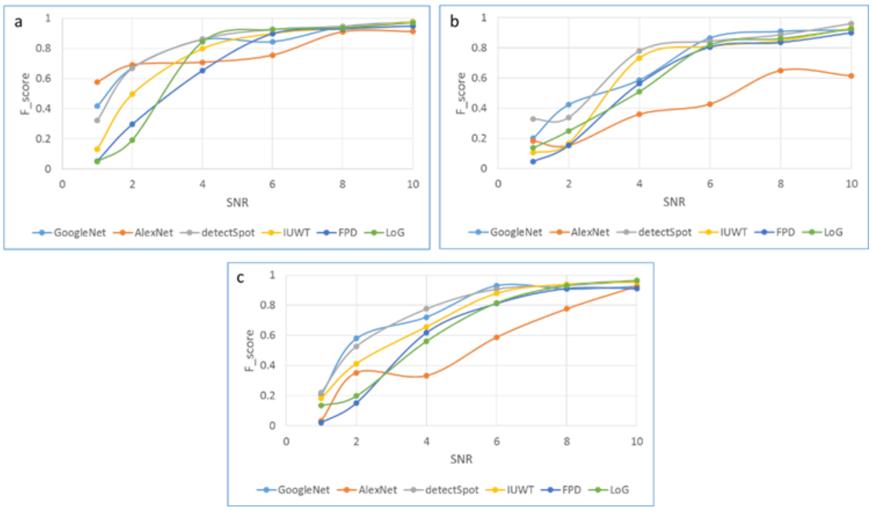

Fig. 6. *F_score* vs SNR curves for all six methods applied to two kinds of synthetic images (a) Synthetic type A, and (b-c) Synthetic type B.

3 Results

The fully trained CNNs models along with the FPD [28], IUWT [29] and LoG [30], were each applied on two types of synthetic images described in Section 2.3.1 as shown in Fig. 5 with a signal-to-noise ratio (SNR) in range {10, 8, 6, 4, 2, 1}. Table 2- Table 4 indicates the results for all six methods in terms of average precision, recall and F_score. These values were averaged for all SNR's. Table 2 indicates the results for type A synthetic images. Higher precision value is reported by AlexNet method on type A synthetic images indicating higher accuracy for relevant spots retrieval and less false spots detected. Furthermore, the GoogleNet method recorded larger values for recall rate and F_score followed by detectSpot method. Fig. 6(a) summarises the performance of all six methods on type A synthetic images on all SNR values. The results indicate that at high levels of SNR (≈ 8 and10) the difference in performance of the methods is negligible. However, as SNR drops (< 4) the performance of the methods is reduced and at SNR = 1, deep learning methods reports higher F_score values compared to traditional methods.

Table 3 and Table 4 presents the results for type B synthetic images. The results indicate that when the background is introduced into the synthetic images, detectSpot method reports higher precision, recall and F_score values compared to all other methods in comparison. The performance of the AlexNet method is reduced significantly compared to its performance on Type A synthetic images. Fig. 6 (b-c) gives a clear indication of how each method performs on Type B synthetic images.

Table 2**.** Evaluation metrics calculated on sythetic images for six detection methods.

Model	Precision	Recall	F_score
GoogleNet	0.833	**0.751**	**0.784**
AlexNet	**0.842**	0.703	0.758
detectSpot	0.836	0.740	0.782
IUWT	0.728	0.689	0.705
FPD	0.717	0.584	0.628
LoG	0.656	0.650	0.652

Table 3. Evaluation metrics calculated on realistic synthetic data. B1.

Method	Precision	Recall	F_score
GoogleNet	0.717	**0.585**	0.633
AlexNet	0.443	0.365	0.397
detectSpot	**0.803**	0.614	**0.675**
IUWT	0.618	0.587	0.598
FPD	0.590	0.534	0.550
LoG	0.647	0.552	0.583

Table 4. Evaluation metrics calculated on realistic synthetic data. B2.

Model	Precision	Recall	F_score
GoogleNet	0.733	**0.699**	0.708
AlexNet	0.567	0.476	0.502
detectSpot	**0.780**	0.675	**0.721**

IUWT	0.740	0.636	0.672
FPD	0.620	0.548	0.571
LoG	0.612	0.533	0.600

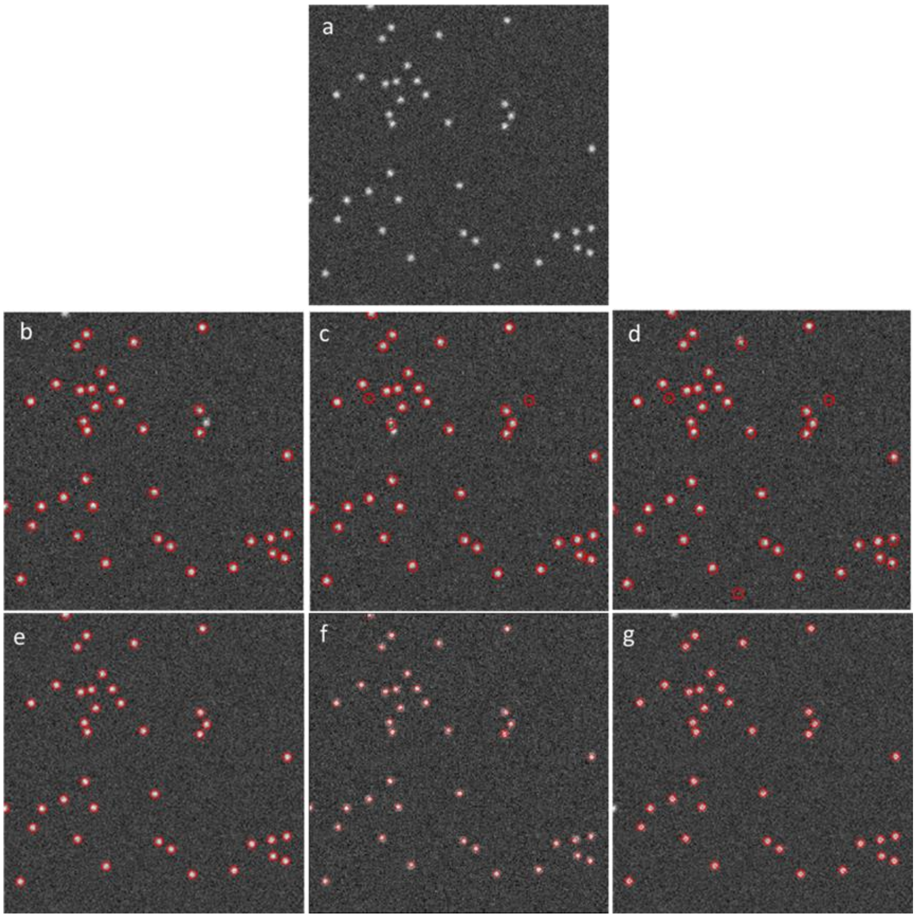

Fig. 7. Illustrates the performance of each method on Type A synthetic images with SNR=10. Detected spots by each method are showed in red circles.(a) Original synthetic image. (b) Spots detected by our approach, detectSpot. (c) Detected spots using GoogleNet. (d) Detected spots with AlexNet, and (e-g) represents the detected spots using, IUWT, LoG, and FPD respectively.

The curves for each method in Fig. 6 indicate higher F_score values for all methods for SNR > 6 except for AlexNet method. However, as SNR drops from 4 to 1 most methods start to deteriorate at this point while detectSpot indicates higher F_score

value at SNR = 1. According to the curves in Fig. 6(c) for all SNR is the performance of GoogleNet and detectSpot methods show small variability regarding f-score values with both methods having higher F_score values at low SNRs. Traditional spot detection methods also give competitive results at higher SNRs especially the IUWT and LoG methods, and their performance is reduced as SNR decreases. Fig. 7 shows the detected spots for each method on Type A synthetic images with SNR=10.

4 Statistical Significant Test

This section implements a comparison of the detection results from detectSpot and GoogleNet methods applied to the same sample, in order to check whether the methods provide similar results or not. The outcome of this test is the acceptance or rejection of the null hypothesis (H_0). To do this significant test, a student t-test will be used which compares the two means between two groups. A paired two tailed t-test was considered to check the statistical significance of the results, with the null hypothesis (H_0) defined as; $\mu_1 - \mu_2 = 0$, with μ being the mean and the subscript $\{1,2\}$ represents method one and two. The alternate hypothesis (H_a) states the opposite, $\mu_1 - \mu_2 \neq 0$. A Shapiro-Wilk test was used to confirm the normality of the data. The chosen p-value was 0.05. Table 5 indicate the statistical results for two methods, detectSpot and GoogleNet, these two methods provide higher and comparable F_score for all experiments compared to other methods. The p-values for detectSpot and GoogleNet are greater than 0.05 for all experiments, these indicate that the detection results between detectSpot and GoogleNet are statistical insignificant as a result these methods provide similar results.

Table 5. P-values for detectSpot and GoogleNet

Methods	DATA SET		
	NOBGND	BGND1	BGND2
detectSpot & GoogleNet	> 0.05	> 0.05	> 0.05

5 Conclusions

The detection of spots is an important step towards the analysis of microscopy images. A number of different automated approaches have been developed to perform the task of spot detection. In this study, we have presented an automated approach for the detection and counting of spots in microscopy images, termed detectSpot. The proposed approach is based on a convolutional neural network with a sliding-window based approach to detect multiple spots in images. The comparative experiments demonstrated that the GoogleNet and detectSpot methods achieved comparable performance compared to the AlexNet method and other traditional spot detection methods. We also have shown that rather training a CNN from scratch, knowledge transfer from natural images to microscopy images is possible. A fine-tuned pre-trained CNN can give results which are comparable to fully trained CNN.

Acknowledgements

This work was carried out in financial support from the Council for Scientific and Industrial Research (CSIR) and the Electrical and Electronic Engineering Department at the University of Johannesburg.

Bibliography

[1] M. Mabaso, D. Withey and B. Twala, "Spot detection in microscopy images using convolutional neural network with sliding-window approach," in *Proceedings of the 5th International Conference on Bioimaging*, Funchal-Madeira, 2018.

[2] A. Bashir, Z. A. Mustafa, I. Abdelhameid and R. Ibrahem, "Detection of malaria parasites using digital image processing," in *Proceedings of the International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE)*, Khartoum, 2017.

[3] A. Verma and G. Khanna, "Survey on digital image processing techniques for tumor detection," *Indian Journal of Science and Technology,* vol. 9, no. 14, pp. 1-15, 2016.

[4] D. Varga and T. Szirányi, "Detecting pedestrians in surveillance videos based on convolutional neural network and motion," in *24th European Signal Processing Conference (EUSIPCO)*, Budapest, Hungary, 2016.

[5] Z. Wang, Z. Li, B. Wang and H. Liu, "Robot grasp detection using multimodal deep convolutional neural networks," *Advances in Mechanical Engineering*, vol. 8, no. 9, pp. 1-12, August 2016.

[6] R. Li, W. Zhang, H.-I. Suk, L. Wang, J. Li, D. Shen and S. Ji, "Deep learning based imaging data completion for improved brain disease diagnosis," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Quebec City, 2014.

[7] A. Genovesio, T. Liendl, V. Emiliana, W. J. Parak, M. Coppey-Moisan and J.-C. Olivo-Marin, "Multiple particle tracking in 3d+t microscopy: Method and application to the tracking of endocytosed quantum dots," *IEEE Trans. Image Process.,* vol. 15, no. 5, pp. 1062-1070, 2006.

[8] J.-C. Olivo-Marin, "Extraction of spots in biological images using multiscale

products," *Pattern Recognition,* vol. 35, no. 9, pp. 1989-1996, 2002.

[9] Y. Kimori, N. Baba and N. Morone, "Extended morphological processing: a practical method for automatic spot detection of biological markers from microscopic images," *BMC Bioinformatics*, vol. 11, no. 373, pp. 1-13, 2010.

[10] I. Smal, M. Loog, W. Niessen and E. Meijering, "Quantitative comparison of spot detection methods in fluorescence microscopy," *IEEE Trans on Medical Imaging,* vol. 29, no. 2, pp. 282-301, 2010.

[11] M. Mabaso, D. Withey and B. Twala, "Spot detection methods in fluorescence microscopy imaging: A review," *Image Analysis & Stereology,* vol. 37, no. 3, pp. 173-190, 2018.

[12] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *arXiv:1512.03385*, 2015.

[13] H. Noh, S. Hong and B. Han, "Learning deconvolution network for semantic segmentation," in *IEEE International Conference on Computer Vision*, 2015.

[14] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. Todd Hurst, C. B. Kendall, M. B. Gotway and J. Liang, "Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning?," *IEEE Transactions on Medical Imaging,* vol. 35, no. 5, pp. 1299-1312, May 2016.

[15] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classication with deep convolutional neural networks," in *Neural Information Processing Systems,* 2012.

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2014.

[17] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going Deeper with Convolutions," in *Computer Vision and Pattern Recognition*, Boston, 2015.

[18] D. A. Van Valen, T. Kudo, K. M. Lane, D. N. Macklin, N. T. Quach, M. M. DeFelice, I. Maayan, Y. Tanouchi, E. A. Ashley and M. W. Covert, "Deep learning automates the quantitative analysis of individual cells in live-cell imaging experiments," *PLoS Comput Biol,* vol. 12, no. 11, pp. 1-24, November 2016.

[19] M. Mabaso, D. Withey and B. Twala, "Spot detection in microscopy images using convolutional neural network with sliding-window approach," in *Proceedings of the 5th International Conference on Bioimaging*, Funchal, 2018.

[20] S. Ruder, "An overview of gradient descent optimization algorithms," 2017. [Online]. Available: http://ruder.io/optimizing-gradient-descent/. [Accessed 10 October 2017].

[21] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," in *Proceedings of the 27th International Conference on*

Machine Learning, 2010.

[22] A. Damien, *TFLearn*, GitHub, 2016.

[23] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, R. Jozefowicz, Y. Jia, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, M. Schuster, R. Monga, S. Moore, D. Murray, C. Olah, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu and X. Zheng, *TensorFlow: Large-scale machine learning on heterogeneous systems*, 12th USENIX Symposium on Operating Systems Design and Implementation, 2015.

[24] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference for Learning Representations*, San Diego, 2015.

[25] M. Mabaso, D. Withey and B. Twala, "A framework for creating realistic synthetic fluorescence microscopy image sequences," in *Bioimaging 2016*, Rome, 2016.

[26] N. Chenouard, "Particle tracking benchmark generator," Institut Pasteur, 2015. [Online]. Available: http://icy.bioimageanalysis.org/plugin/Particle_tracking_benchmark_generator. [Accessed 1 November 2016].

[27] "The open microscopy environment," 2016. [Online]. Available: http://www.openmicroscopy.org/site/support/omero5.2/developers/Matlab.html. [Accessed 15 November 2016].

[28] I. F. Sbalzarini and P. Koumoutsakos, "Feature point tracking and trajectory analysis for video imaging in cell biology," *Journal of Structural Biology*, vol. 151, no. 2, pp. 182-195, 2005.

[29] J.-C. Olivo-Marin, "Extraction of spots in biological images using multiscale products," *Pattern Recognition*, vol. 35, no. 9, pp. 1989-1996, 2002.

[30] A. Raj, P. van den Bogaard, S. A. Rifkin, A. van Oudenaarfen and S. Tyagi, "Imaging individual mRNA molecules using multiple singly labeled probes," *Nature Methods*, vol. 5, no. 10, pp. 877-879, 2008.