Automatic cell counter for cell viability estimation

Jasna Lojk[1], Luka Šajn[2], Uroš Čibej[2], Mojca Pavlin[1]
[1]Faculty of electrical engineering, University of Ljubljana, Ljubljana, Slovenia
[2]Faculty of computer and information science, University of Ljubljana, Ljubljana, Slovenia
email: mojca.pavlin@fe.uni-lj.si

Abstract - Despite several methods that exist in different fields of life sciences, certain biotechnological applications still require microscopic analysis of the samples and in many instances, counting of cells. Some of those are drug delivery, transfection or analysis of mechanism fluorescent probes are used to detect cell viability, efficiency of a specific drug delivery or some other effect. For analysis and quantification of these results it is necessary to either manually or automatically count and analyze microscope images. However, in everyday use many researchers still count cells manually since existing solutions require either some specific knowledge of computer vision and/or manual fine tuning of various parameters.

Here we present a new software solution (named CellCounter) for automatic and semi-automatic cell counting of fluorescent microscopic images. This application is specifically designed for counting fluorescently stained cells. The program enables counting of cell nuclei or cell cytoplasm stained with different fluorescent stained. This simplifies image analysis for several biotechnological applications where fluorescent microscopy is used. We present results and validate the presented automatic cell counting program for cell viability application.

We give empirical results showing the efficiency of the proposed solution by comparing manual counts with the results returned by automated counting. We also show how the results can be further improved by combining manual and automated counts.

I. Introduction

Cell counting is one of the most fundamental tools in biology, medicine, and other life sciences. Several methods exist for quantification of different phenomena, but substantial part uses fluorescent dyes in combination with various protocols in order to analyze different processes. Fluorescently stained cells can be analyzed using flow cytometry[1], spectrofluorometry[2] and/or fluorescent microscopy[3]. There are of course other existing solutions for counting cells without the use of florescence, like counting with counting chambers (hemocytometer)[4], but those can be time consuming and also require detachment of the cells, which can result in their loss and damage. Fluorescent microscopy is also a standard method for visualization and quantification of different phenomena and many researchers rely on microscopic images and image analysis tools to obtain reliable quantitative data. Cell counting using microscopic images can be automated, and several semi-automatic and fully automatic solutions have been developed. In practice, however, such solutions require some additional input regarding various image parameters, such as thresholds, approximate nuclei distances, contrast settings, etc. These parameters require specific knowledge of image processing, which is not common among life-science experts.

Currently, several programs for cell counting are already available. From many commercial solutions, such as MetaMorph [5] (Molecula Devices, Downingtoetn, PA, USA), BioQuant [6] (Image Analysis Corporation, Nashville, TN, USA), Image-Pro [7] (Media Cybernetics, Bethesda, MD, USA) and SynenTec [8], to free software such as ITCN ImageJ Plugin [9] (US National Institute of Health, Bethesda, MD; USA), CellProfiler [10], UTHSCSA ImageTool (University of Texas Health Science Center, San Antorio, TX, USA) [11] and CellC [12]. However, all of the existing solutions require some additional input or even some image pre-processing of the raw images, which is not only time-consuming but can also be also less user-friendly for life scientists [13].

Besides the human factors, full automatisation of cell counting faces several technical obstacles. Some of the most common problems, are: i) a wide variety of assays and cell types resulting in a range of objects which are hard to generalize, ii) this can be even more problematic due to uneven illumination and other equipment related factors, such as electronic and/or optical noise that lead to images having variable contrast and quality [14], iii) cells can have a varying density in the cell culture and can overlap [15], iv) extracellular debris and internal cell structures can interfere with the recognition algorithm, giving false results and v) depending on the microscopy technique used, images can have low contrast therefore detection of individual nuclei in multinucleated cells is difficult.

In this article we present a novel structured method for effective cell counting which does not require any specific knowledge of image processing. The method is specifically designed for counting fluorescently stained cells, that are used in many applications from assays for measuring cell viability [16] to determination of transfection efficiency [17–19]. We also show that the presented solution has been already intensively implemented for cell counting of different sets of experiments and it has drastically improved the typical workflow of researchers, as it is shown in our empirical evaluation.

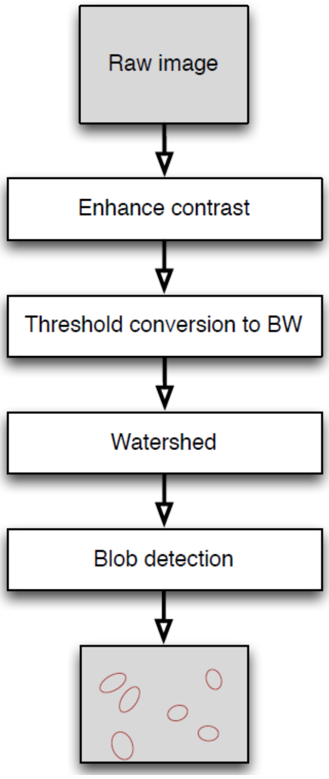


Figure 1. Schematic overview of the cell-counting method

II. Materials and Methods

A. Algorithm used

The presented program was written in C\# and uses an open source C\# framework designed for developers and researchers in the fields of Computer Vision and Artificial Intelligence (AForge.NET [20]). The algorithm used in the program can be viewed as a simple dataflow of images passing through various image processing algorithms. The sequence of these algorithms was determined empirically by trying to overcome as many problems on a sample set of images as possible. The schematic overview of our method is shown in Fig. 1 and will be described in detail below.

The algorithm first addressed the problem of the image contrast, which is can be really low on microscopic images and not necessarily distributed evenly throughout the image due to uneven illumination or other optical problems. The contrast of the image is thus improved using the Contrast Limited Adaptive Histogram Equalization (CLAHE) algorithm [21–23]. It differs from ordinary histogram equalization in the respect that the adaptive method computes several histograms, each corresponding to a distinct section of the image, and uses them to redistribute the brightness values of the image. It is therefore suitable for improving the local contrast of an image and bringing out more detail. In our solution, the image is cut into 64 (8 × 8) disjoint sections and histogram equalization is applied on each part independently. Within each image part, local maximum and minimum pixel intensity are calculated and the intensities are proportionally adjusted to the scale ranging from 0 to 255. The individual parts are then recomposed again using interpolation at image borders. The final image has better contrast and can reveal more local information (Fig. 2).

Next step is converting grey-scale images to black-white images where the object borders are defined using a threshold algorithm. In this step, Otsu Threshold algorithm [24], [25] was used, as it gave slightly better results compared to other tested algorithms (Huang, Renyi entropy method, and other). In this algorithm, the threshold value t is defined as the value that gives the minimal intra-class variance defined by:

$$\omega(t)\, \sigma_1^2(t) + \omega_2(t)\sigma_2^2(t)$$

where ω_1 (ω_2) is the probability that the pixel value in the image is lower (higher) than t and σ_1^2 (σ_2^2) is the variance of the pixel values lower (higher) than t.

In cell cultures, cells tend to grow in close proximity, touch or even overlap, resulting in connected objects on the images. To split the overlapping (touching) objects, the Watershed algorithm [26] was applied. The image is eroded on the watershed lines, which usually represent exactly the borders between different cells. This enables the detection of individual cells, even multinucleated and overlapping cells (Fig. 2). Next, the connected regions are counted using classical blob detection algorithm [27], which counts the objects separated by black background.

Although the counting algorithm is robust, as we will show in the following sections, the miscounts can still occur on more problematic images. To correct erroneously counted objects, a function was added to the user interface that enables the user to add or remove objects from the count with a mouse click. This will be

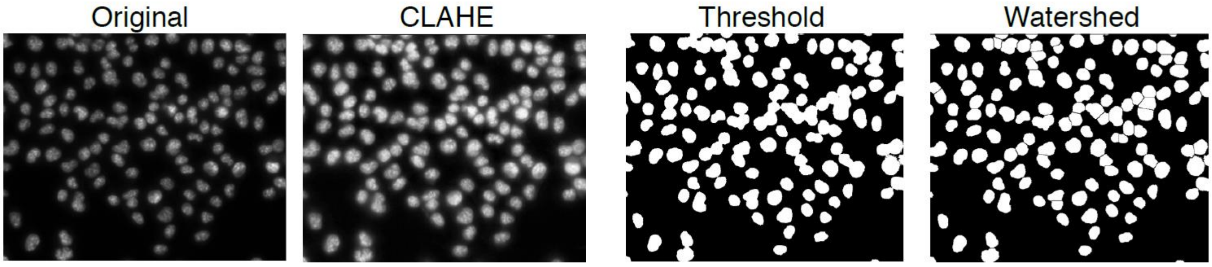


Figure 2. A sequence of images produced by the program on a sample image.

referred as a semi-automatic counting henceforth.

B. Cell culturing

The experiments used for program validation were done on Chinese hamster ovary cells (CHO) grown in Ham's tissue culture medium for mammalian cells (HAM) supplemented with 10% fetal bovine serum (FBS) at 37°C in 5% CO_2-enriched air at saturation humidity. All experiments were performed on 24 h old cell cultures in exponential growth phase.

C. Nanoparticles and cell viability

The cell viability experiment was performed as described previously [12]. Briefly, cells were incubated with increasing concentration of polycationic polymer coated magnetic nanoparticles for 24 h and stained with two fluorescent dyes; Hoechst 33342, which stained all cell nuclei, and propidium iodide (PI), which differentially stained only dead cells. At least 15 visual fields at 200 × magnification were taken of each sample for each used fluorescent dye using a fluorescent microscope (Zeiss 200, Axiovert, Germany). The images were recorded by MetaMorph imaging system software (Visitron, Germany) and saved in TIF format.

D. Cell counting

The algorithm was developed and optimized based on microscopic images obtained from several different types of experiments and experimental repeats, but for evaluation of the program efficiency, one random experiment was selected.

Cells on the images of the selected experiment were counted using only CellCounter (automatic), by manually correcting CellCounter's results (semi-automatic) and by manually counting using ImageJ software (manual) (v1.45s, National Institute of Health, USA). Three people counted the experiment independently to determine inter-personal error. Counting time was noted for each counting step. Results are expressed as mean ± standard error (SE).

III. Results and discussion

We have developed an algorithm for automatic cell counting of fluorescently stained cells. As an example to evaluate the program, cells were stained with blue fluorescent dye Hoechst and with red fluorescent dye PI. The program can also be used to count cells or objects labeled with other fluorescent markers, like green fluorescent protein (GFP) [13], [14].

The algorithm of our program works by improving the contrast of bright objects on the images (in our case fluorescently stained nuclei and/or cytoplasm) and converts the obtained grey-scale images to black-white images on which the automatic counting is applied. As such, the program does not require images with objects that have only clear borders and was found quite flexible/robust in determining accurate numbers of cells even if the quality of images is not ideal. As seen from Fig. 3, the program is able to count cells on out of focus images with relatively small errors. This allows less accurate and thus faster acquisition of images for counting. Also, because the algorithm does not rely on size or shape of the objects, the program can be applied to fluorescent images of any appropriately stained cell line, as already confirmed for several different cell types (results not shown).

Cells of the same experiment were stained with Hoechst, which stains all cell nuclei, thus giving us the total number of the cells (Fig. 4), and with PI, which only stains dead or damaged cells (Fig. 5, Fig. 6). This combination of dyes can be used to determine cell viability [12]. To obtain the number of viable cells, which was the purpose of the experiment, the number of dead cells was subtracted from the number of all cells (Fig. 7).

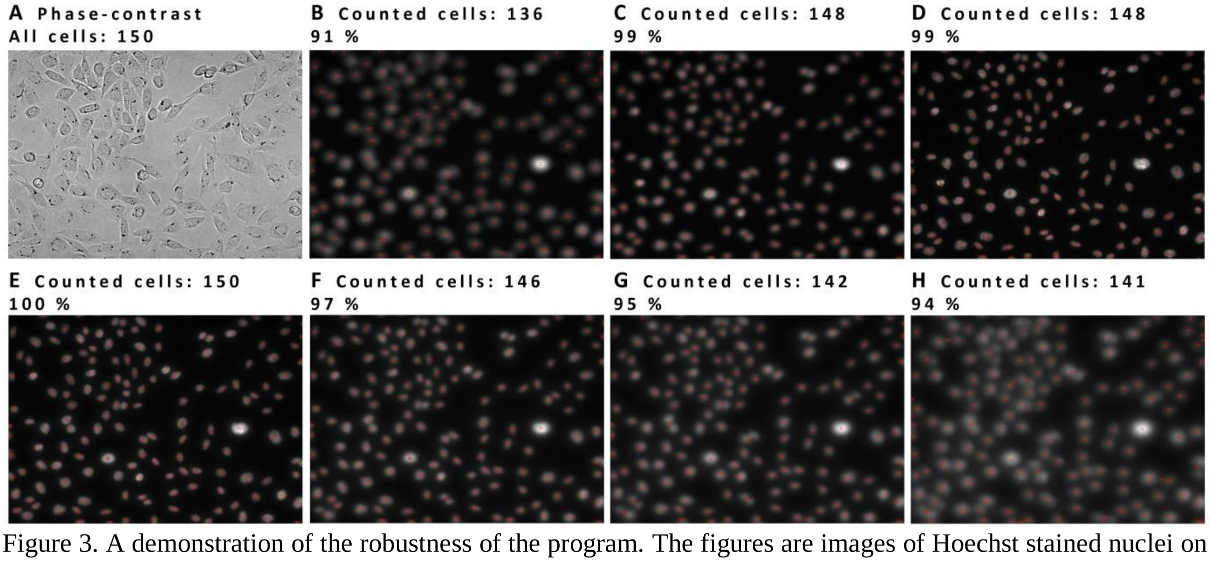


Figure 3. A demonstration of the robustness of the program. The figures are images of Hoechst stained nuclei on the same visual field obtained on different out of focus planes, under (B, C, D) or above (F, G, H) the plane in focus (E). The total number of cells was obtained by manually counting the cells on phase contrast image. Above each image are given the total number of automatically counted cells and the percentage of the cells compared to manual count. The fluorescent image in focus (E) gave a 100% counting accuracy.

All images were counted by three users to obtain the inter-personal error and the differences in obtained counting results for all three applied counting methods; manual, semi-automatic and automatic counting.

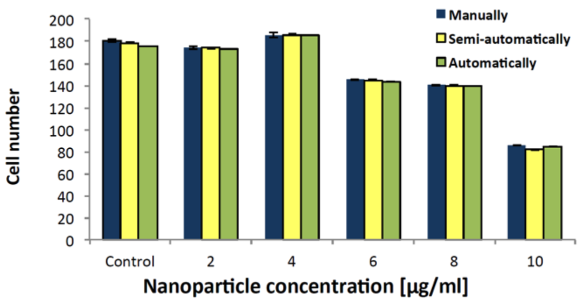


Figure 4. Comparison of counting results obtained by counting fluorescently stained cell nuclei to determine the total number of CHO cells after exposure to increasing concentration of nanoparticles for 24 h. Three counting methods were used: manually by counting Hoechst positive cells using ImageJ software, semi-automatically by correcting CellCounter counts and automatically by using CellCounter only. The results presented the average percentage of viability for separate samples compared to non-exposed control obtained by independent counting of the same experiment by three users with corresponding standard error (N=3).

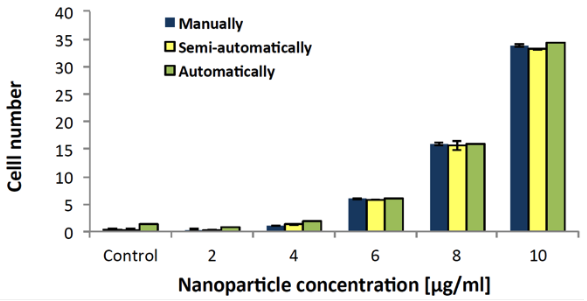


Figure 5: Comparison of counting results obtained by counting fluorescently PI stained cell to determine the number of dead CHO cells as a measure of cytotoxicity of increasing concentration of nanoparticles for 24 h. Three counting methods were used: manually by counting PI positive cells using ImageJ software, semi-automatically by correcting CellCounter counts and automatically by using CellCounter only. The results presented are average cell numbers for separate samples obtained by independent counting of the same experiment by three users with corresponding standard error (N=3).

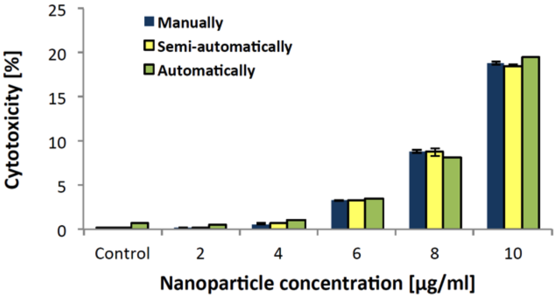


Figure 6. Comparison of counting results obtained by counting fluorescently stained cell nuclei to determine the number of dead CHO cells as a measure of cytotoxicity of increasing concentration of nanoparticles for 24 h. Three counting methods were used: manually by counting PI positive cells using ImageJ software, semi-automatically by correcting CellCounter counts and automatically by using CellCounter only. The results presented the average percentage of toxicity for separate samples compared to the number of all cells in non-exposed control obtained by independent counting of the same experiment by three users with corresponding standard error (N=3).

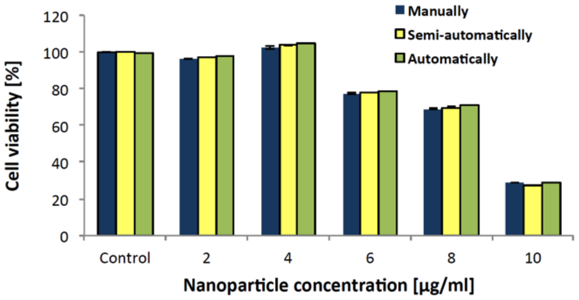


Figure 7. Comparison of counting results obtained by counting fluorescently stained cell nuclei (stained with Hoechst) reduced by the number of dead cells (stained with PI) to determine the viability of CHO cells exposed to increasing concentration of nanoparticles for 24 h. Three counting methods were used: manually by counting Hoechst positive cells using ImageJ software, semi-automatically by correcting CellCounter counts and automatically by using CellCounter only. The results presented are average cell numbers for separate samples obtained by independent counting of the same experiment by three users with corresponding standard error (N=3).

The counting results for all three counting methods showed reasonably good correlation and small standard errors, indicating also a relatively low inter-personal error as shown in Figs. 4-7. The differences between manual or semi-automatic counting and automatic counting are mostly due to two or more objects in substantial contact being counted as one or due to objects positioned predominantly outside of the margin of the image. Generally, we observed better performance in images with fewer and randomly scattered objects. On the other hand, the program had problems recognizing the right objects on images with low contrast, either images with no cells or with dimly stained cells. Still, the obtained differences are small compared to the expected deviations between repeated experiments, indicating the algorithm is robust enough and can be used as a helpful aid for image analysis.

Simultaneously, time required for counting with each counting method (automatic, semi-automatic or manual) was recorded for all three users. As seen from Fig. 8, the program significantly reduces the time necessary to count the acquired images from hours (manual counting) to minutes (semi-automatic counting) or even seconds (automatic counting), depending on the required accuracy of the results.

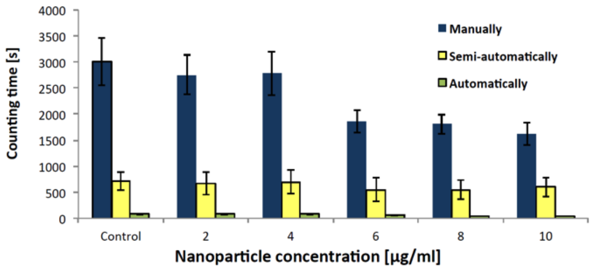
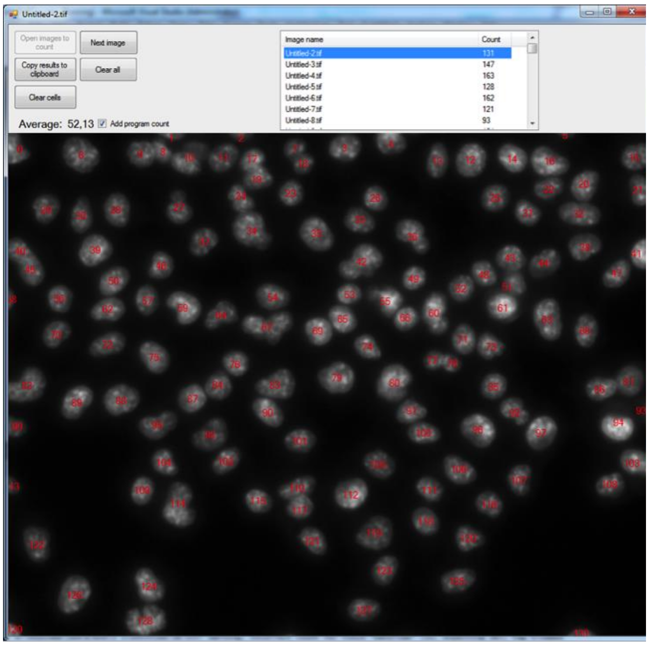


Figure 8. Comparison of counting time using three counting methods: manually by counting Hoechst and PI positive cells using ImageJ software, semi-automatically by correcting CellCounter counts and automatically by using CellCounter only. The results presented are average counting times for separate samples of three independent users with corresponding standard error (N = 3).

Moreover, the requirements of installation of CellCounter program are minimal, the interface is user friendly (Fig. 9), requiring no additional image manipulation or tuning of parameters to get reliable results (RAW images as obtained from the microscope can be used). Images can be loaded through the interface or by simple drag and drop action. The program counts the cells automatically as soon as the user selects the image, and allows simple corrections to the program count by adding or removing hits on the selected image with simple mouse clicks. The program also enables to skip a non-representative image. The results are automatically ordered and can be easily copied to clipboard to transfer them to the editing program of choice.



Average: 52,13 ☑ Add program count

Image name	Count
Untitled-2.tif	131
Untitled-3.tif	147
Untitled-4.tif	163
Untitled-5.tif	128
Untitled-6.tif	162
Untitled-7.tif	121
Untitled-8.tif	93

Figure 9. A screenshot of the CellCounter program interface showing the result of one automatic count. Each object (cell) is labeled so the user can quickly assert the accuracy of the counting event.

IV. Conclusions

In this article, we present an alternative solution for automatic and semi-automatic counting of cells on fluorescent microscopic images. As we showed in this paper, the program enables consistent, robust, fast and adequately accurate determination of fluorescent cells and can therefore be applied to a range of different applications in different fields of life sciences where fluorescent labeling is used for quantification of different phenomena.

Acknowledgment

The authors would like to thank David Karlaš for the help with the program and Klemen Strojan and Maja Jankovec for their help with the manual cell counting. This work was supported by Slovenian Research Agency within project J4-4324, young researchers program and MRIC UL IP-0510 Infrastructure program.

References

[1] M. Rieseberg, C. Kasper, K. F. Reardon, and T. Scheper, “Flow cytometry in biotechnology,” *Appl. Microbiol. Biotechnol.*, vol. 56, no. 3–4, pp. 350–360, Aug. 2001.

[2] J. R. Lakowicz, *Principles of Fluorescence Spectroscopy*, 3rd ed. Springer, 2006.

[3] D. J. Stephens and V. J. Allan, “Light Microscopy Techniques for Live Cell Imaging,” *Science*, vol. 300, no. 5616, pp. 82–86, Apr. 2003.

[4] F. J. Dein, A. Wilson, D. Fischer, and P. Langenberg, “Avian Leucocyte Counting Using the Hemocytometer,” *J. Zoo Wildl. Med.*, vol. 25, no. 3, pp. 432–437, Sep. 1994.

[5] Metamorph - http://www.moleculardevices.com/products/software/meta-imaging-series/metamorph.html

[6] BioQuant - http://lifescience.bioquant.com/

[7] Image-Pro - http://www.mediacy.com/index.aspx?page=IPP

[8] SynenTec - http://www.synentec.com/

[9] S. R. Gallagher, “Digital Image Processing and Analysis with ImageJ,” in *Current Protocols Essential Laboratory Techniques*, John Wiley & Sons, Inc., 2008.

[10] A. E. Carpenter, T. R. Jones, M. R. Lamprecht, C. Clarke, I. H. Kang, O. Friman, D. A. Guertin, J. H. Chang, R. A. Lindquist, J. Moffat, P. Golland, and D. M. Sabatini, “CellProfiler: image analysis software for identifying and quantifying cell phenotypes,” *Genome Biol.*, vol. 7, no. 10, p. R100, 2006.

[11] “UTHSCSA ImageTool.” [Online]. Available: http://compdent.uthscsa.edu/dig/itdesc.html. [Accessed: 02-Apr-2014].

[12] J. Selinummi, J. Seppälä, O. Yli-Harja, and J. A. Puhakka, “Software for quantification of labeled

bacteria from digital microscope images by automated image analysis," *BioTechniques*, vol. 39, no. 6, pp. 859–863, Dec. 2005.

[13] J. Byun, M. R. Verardo, B. Sumengen, G. P. Lewis, B. S. Manjunath, and S. K. Fisher, "Automated tool for the detection of cell nuclei in digital microscopic images: application to retinal images," *Mol. Vis.*, vol. 12, pp. 949–960, 2006.

[14] R. M. Haralick and L. G. Shapiro, "Image segmentation techniques," *Comput. Vis. Graph. Image Process.*, vol. 29, no. 1, pp. 100–132, Jan. 1985.

[15] K. Li, E. D. Miller, L. E. Weiss, P. G. Campbell, and T. Kanade, "Online tracking of migrating and proliferating cells imaged with phase-contrast microscopy," in *IN: PROC. IEEE CONF. COMP. VISION AND PATT. RECOG. WORKSHOP, P. 65. IEEE COMPUTER SOCIETY PRESS, LOS ALAMITOS*, 2006, p. 65.

[16] V. B. Bregar, J. Lojk, V. Šuštar, P. Veranič, and M. Pavlin, "Visualization of internalization of functionalized cobalt ferrite nanoparticles and their intracellular fate," *Int. J. Nanomedicine*, vol. 8, pp. 919–931, 2013.

[17] M. Kanduser, D. Miklavcic, and M. Pavlin, "Mechanisms involved in gene electrotransfer using high- and low-voltage pulses--an in vitro study," pp. 265–271, Feb. 2009.

[18] T. Marš, M. Stražišar, N. Kotnik, K. Pegan, J. Lojk, Z. Grubič, and M. Pavlin, "Electrotransfection and lipofection show comparable efficiency for in-vitro gene therapy of primary human myoblasts," *unpublished*, 2014.

[19] I. Marjanovic, M. Kandušer, D. Miklavčič, M. Manček Keber, and M. Pavlin, "Comparison of flow cytometry, fluorescence microscopy and spectrofluorometry for analysis of gene electrotransfer efficiency," *unpublished*, 2014.

[20] "AForge.NET Framework." [Online]. Available: http://www.aforgenet.com/aforge/framework/. [Accessed: 02-Apr-2014].

[21] S. A. Ahmad, "Analysis of Compound Enhancement Algorithms (CEA) based on Adaptive Histogram Equalization (AHE) on Intra-oral Dental Radiographs Images," *Int. J. New Comput. Archit. Their Appl. IJNCAA*, vol. 4, pp. 902–916, 2011.

[22] H. D. Cheng and X. J. Shi, "A simple and effective histogram equalization approach to image enhancement," *Digit. Signal Process.*, vol. 14, no. 2, pp. 158–170, Mar. 2004.

[23] K. Zuiderveld, "Contrast limited adaptive histogram equalization," pp. 474–485, 1994.

[24] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *Syst. Man Cybern. IEEE Trans. On*, vol. 9, pp. 62–66, 1979.

[25] X. Xu, S. Xu, L. Jin, and E. Song, "Characteristic analysis of Otsu threshold and its applications," *Pattern Recognit. Lett.*, vol. 32, no. 7, pp. 956–961, May 2011.

[26] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 6, pp. 583–598, Jun. 1991.

[27] F. Chang, C.-J. Chen, and C.-J. Lu, "A linear-time component-labeling algorithm using contour tracing technique," *Comput. Vis. Image Underst.*, vol. 93, no. 2, pp. 206–220, Feb. 2004.