GLCM, kNN and Meanshift for neuron detection on Nissl-stained brain slice images

S. A. Nosova, V. E. Turlapov
svetlana.nosova@itmm.unn.ru; vadim.turlapov@gmail.com
Lobachevsky university of Nizhnij Novgorod

The method for neuron detection on Nissl-stained brain slice images is proposed. The method uses textural features of neurons extracted from 4 GLC-matrices. The method includes the following steps: image preprocessing, kNN classification by the textural features and Meanshift clustering of neuron pixels. Preprocessing includes the following steps: grayscale conversion, histogram equalization, histogram quantization. Gray conversion by blue component gives the best result. It is shown that using 2-,4- bin histogram gives close detection quality with 8-bin histogram ($F1 = 0,83..0,85$). For pixel classification kNN algorithm was used. The results demonstrate that kNN is better choice for current task in comparing with NBC. The reached detection quality for given approach is $precision = 0,82, recall = 0,92, F1 = 0,86$. Is is shown that our results are near the same or some better in *recall* characteristic in comparing with other neuron detection method. In our future work we'll prolong this investigation for great volume of dataset and special dataset for important diseases.

Keywords:*machine learning; optical microscopy image; Nissl staining brain slices; classification; clusterization; textural features; kNN; GLCM*

DOI: 10.21469/22233792.4.3.04

1 Introduction

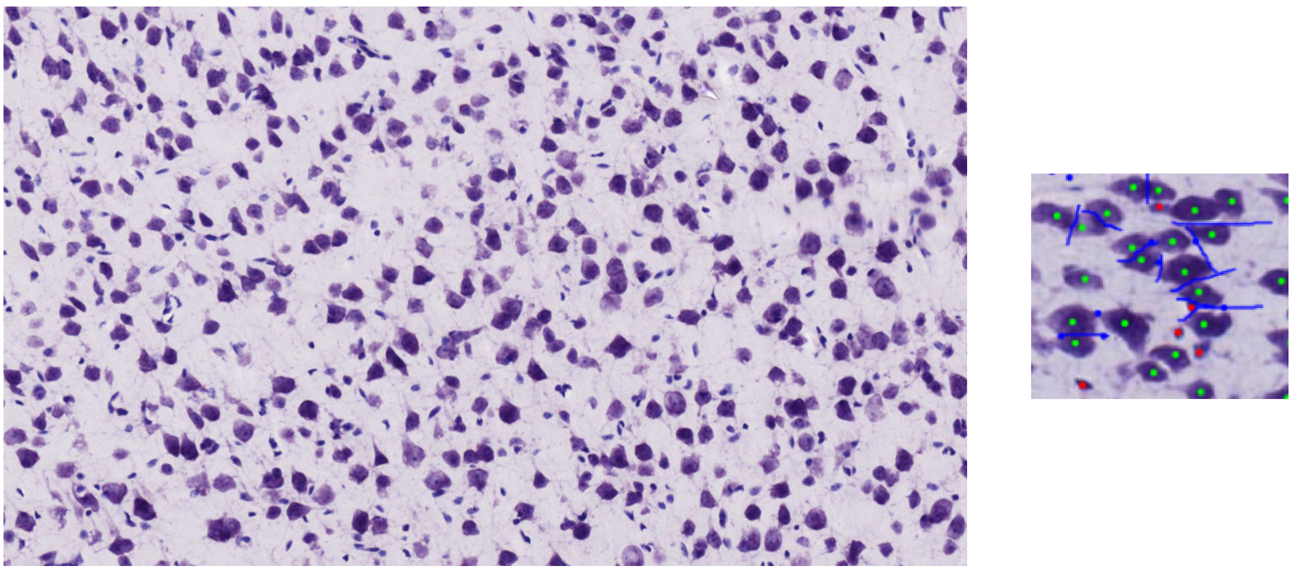
The study of the brain begins with optical microscopy and Nissl staining leads in it. For processing of Nissl-stained cerebral cortex images the following problems are solved: detection of borders for cortex layers; detection and classification of neurons, statistics collection of neuron and astrocytes locations. Usually such tasks are solved by expert in neuromorphology. Currently, there are solutions (including artificial intelligence methods) in automatization of morphological analysis for Nissl-stained brain slice images [1]. In the same time there are many unresolved issues.

In this paper we consider the problem of automating the detection of neurons. The choice of the method of solving the problem depends on two factors: the distinctness of the characteristics of neurons as objects of the image; quality and quantity of training data marked by an expert. From the point of view of the distinctive features, there are the following problems: 1) neurons of the brain are different both in type and form; 2) the sizes of neurons in one image may differ by more than 2 times; 3) images of bodies of neurons are often overlapped; 4) images of bodies of neurons can have a low contrast with the background; 5) images of neurons of the same type can have significant differences in the histogram; 6) images of neurons can have "voids"inside their bodies.

From the point of view of the quality of solving problems of detection and classification by methods of machine learning, the best option is the large size of the training base, when significant resources are spent on qualitative marking of objects (tens to hundreds of complex objects, thousands of simple ones). In this case, various variants of deep learning [2] or complex methods of traditional machine learning [1] are often used today to solve the detection tasks.

In Fig.1 an example of marking neurons is shown on a small fragment of the image of a Nissl-stained brain slice. You can consider this example as an example of markup of simple objects.

However, in all cases of working with images, texture attributes (features) are very useful. The reasons for the utility are several, the most important are the following: 1) the possibility of formalizing the description of the structure of the image; 2) the possibility of a hierarchical (multiscale) application of the description; 3) the possibility of independent parallel processing; 4) the ability to work with three-dimensional images. For acquaintance with surveys on texture analysis methods before 2004 may be used [3].



(a)Nissl-image example

(b) Manual marking

Figure 1 Dataset examples

A special place in a number of methods of texture analysis occupies the GLCM (Gray Level Co-Occurrence Matrice) method, which began in 1973 [4]. The effectiveness of the method is due primarily to the fact that the GLC-matrix is able to explicitly fix a certain specificity of the image. And then to construct from N*N matrix elements a small number of clear, informative and structurally-oriented scalar characteristics: 1) maximum probability entry; 2) element difference moment of order k; 3) contrast; 4) entropy; 5) uniformity; 6) homogeneity. The most popular values of N in the practice of applying the method are apparently N equal to 8 and 16, and the sizes of the image fragments are from 30x30 to 50x50. Due to its qualities, the method is implemented in the Matlab mathematical environment, it has become one of the basic function for image processing in remote sensing of the Earth [5].

In the last decade publication 2012 [6] it is shown that the use of GLCM trace in texture analysis can be used as a feature, and in combination with Haralick features provides better results. It is shown that trace extracted from three dimensional images is also can be used.

The method proposed in [7] uses already 16 different features extracted from GLCM to capture the textures information. It can classify textures at varying orientations, scales and regions with similar textures, and successfully discriminates textures which have similar intensity values. False positive regions are removed using morphology. The evaluation includes more than 80 Brodatz textures.

In publication [8] it is convincingly shown that using parameter distance $d = 1$ for construction of co-occurrence matrix (GLCM) can get better results than using $d = 2$. Using $\theta = 0°$ show comparable results for GLCM processing with $\theta = 90°$. It is concluded that the use of Sobel edge-detector operator together with GLCM prove to be an effective method to quantify the surface texture of an image.

Today, the GLCM method is widely used in the diagnosis of the brain according to MRI. A typical example of this is the publication [9], in which the important features were extracted with the GLCM algorithm. Then the feature set is classified into three types of stroke using support vector machine (SVM) classifier. The lesion area was segmented with accuracy of 90.23%, which is higher than previes method having accuracy of 87.34

However, the promotion of GLCM did not reach the task of neuron detection from Nissl-stained brain slice images. This is what we consider as the task of this work. As a guideline for comparing the our results obtained for optical microscopy with Nissl-stained brain slices we will consider the 2008 year publication [1]. The algorithm steps of this article involve active contour segmentation to find outlines of potential neuron cell bodies followed by artificial neural network training using such segmentation properties as size, optical density, gyration, etc. The algorithm positively identifies $86 \pm 5\%$ neurons on a given set of Nissl-stained images, whereas semi-automatic methods obtain $80 \pm 7\%$.

2 Train Dataset

Manual marking is used for prepating of train dataset. Example of train image one can see of 1(b). Green pixels can be neuron center pixels (class 1). Red and blue pixels are not pixels of neuron centers (class 0). Train dataset consists from texture future of colored pixels by source image using proposed method.

3 Proposed Solution

The proposed solution consists of the following steps: image preprocessing, texture feature extraction, pixel classification and pixel clusterization. Preprocessing step is convertation from color image to gray image.

In proposed method image preprocessing consists of three steps: gray transformation, histogram equalization and histogram quantization. Four methods of conversion (transformation) are used: luminance, blue component, red component, maximum from component. Histogram equalization is used to improve the image tone distribution. Histogram quantization is used to reduce image size for our CUDA-realization [10]. To reduce execution time calculations of texture features are performed on GPU. There is limit of private memory for every CUDA-process. This is why it's needed to quantize histogram to reduce required local memory size.

The Gray Level Coocurrence Matrix (GLCM) method is used for feature extraction [4]. GLCM is used for extraction of second order statistical texture. This method is invariant to rotation and resizing of objects. Texture features are calculated for every pixel of test image. Every pixel feature is 4-component vector based on statistical characteristics of pixel GLCMs.

For pixel classification k nearest neighbors (kNN) algorithm is used. In the

On the step of pixel clusterization only pixels of neuron class are considered. For clusterization Mean shift algorithm with simple filtering is used [11]. Cluster centers are neuron centers.

3.1 Preprocessing

Image is matrix $M * N$. Open-source Nissl-stained images can be ones of two groups: color image and gray image. It's often case if source image is color image. For such image every

pixel is vector of three components: blue, red and green $I(x, y) = (B, G, R)$. For general case we process gray-scale images only. For this it's necessary to convert color image to gray image $G(x, y)$. There are different ways to do it [12]. We use the following formulas:
$Y_{luminance} = 0.3 * B + 0.59 * G + 0.11 * R$,
$Y_{blue} = B$, $Y_{red} = R$, $Y_{max} = max(B, G, R)$.
$Y_{luminance}$ is used as the most universal formula. On the 1(a) one can see that the basic color of this image is violet. It means that there is big blue and red components. This is why Y_{blue},Y_{red} and Y_{max} were used. The results of gray transformations you can see on the fig:figureGray. Histogram $H_{blue}(i)$ of gray image G_{blue} (result of conversion using Y_{blue}) one can see on the 3(a). To align distribution of $G(x, y)$ histogram equalization is used [**?**]. The result $G_{eq}(x, y)$ of the equalization can be calculated using the following formulas:

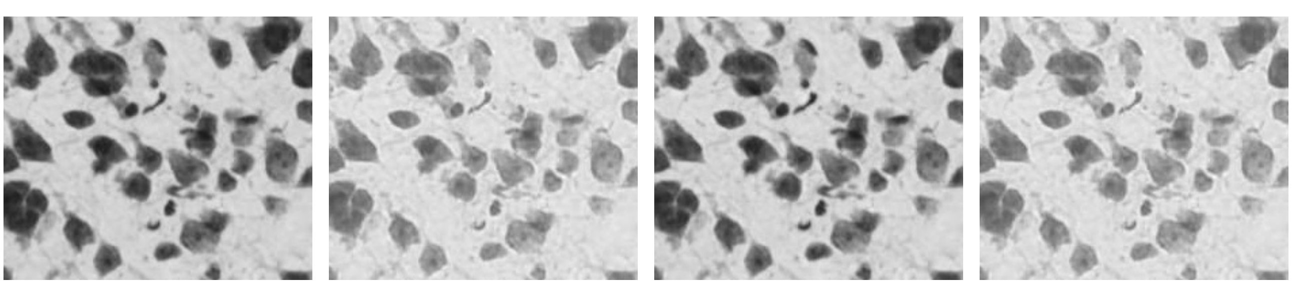
$$H(i) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \begin{cases} 1, \text{if } G(x, y) = i; \\ 0, \text{if } G(x, y) \neq i; \end{cases} \quad (1)$$

$$H_{eq}(i) = \sum_{j=0}^{i-1} H(j)$$

$$G_{eq}(x, y) = H_{eq}(I(x, y)) * 255$$

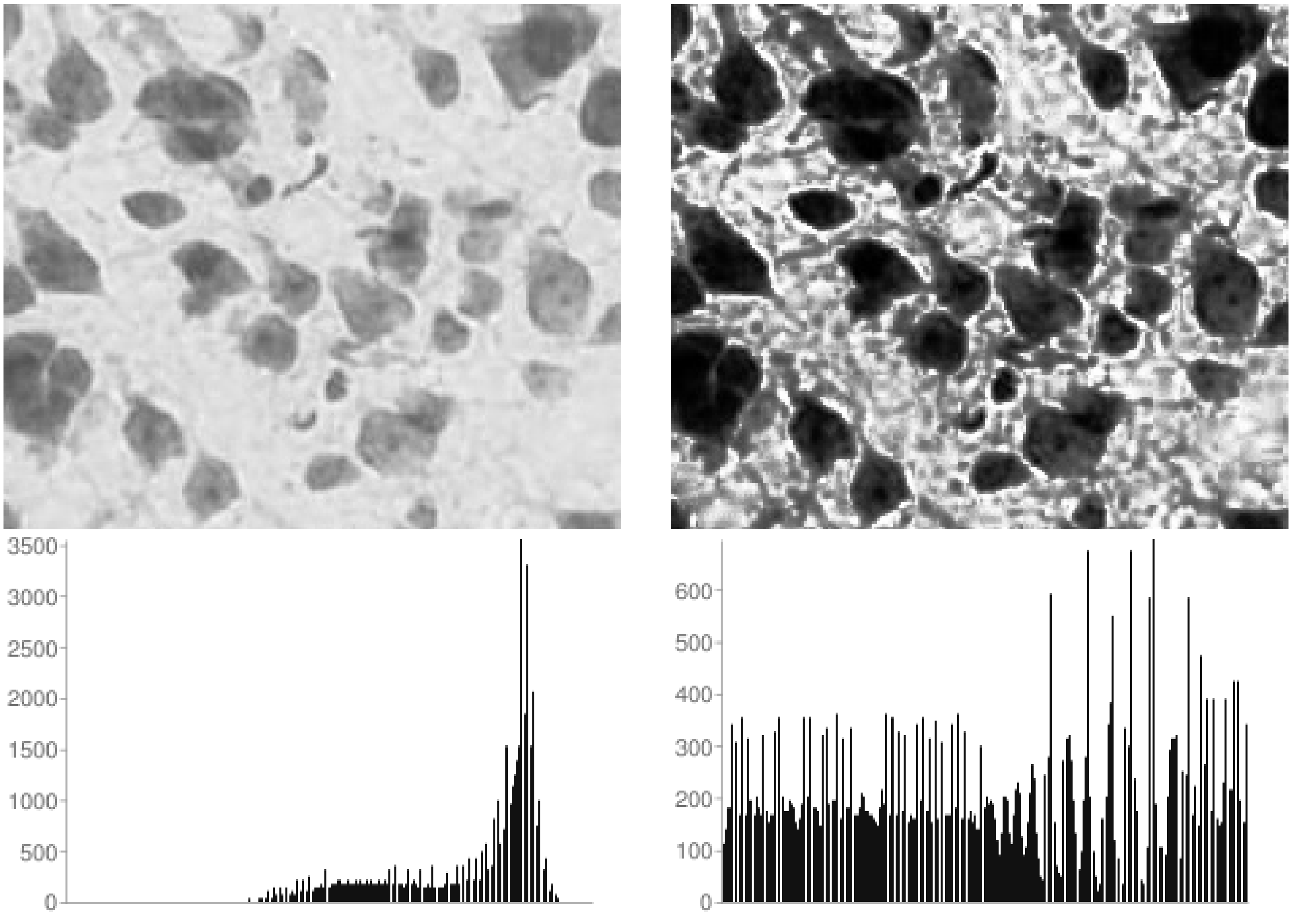
Result of histogram equalization for $G_{blue}(x, y)$ one can see on the fig:figureHistogram(b). Let's suppose that it's needed reduce size of histogram unique numbers from [0..255] to $[0..G_{MAX}]$. The following quantization formula is used:

$$G_{qua}(x, y) = \frac{G_{eq}(x, y)}{G_{MAX}}$$



(a) $Y_{luminance}$ (b) Y_{blue} (c) Y_{red} (d) Y_{max}

Figure 2 Gray transformations



(a) Original (b) Equalized

Figure 3 Histogram equalization results

3.2 Feature extraction

In this paper GLCM texture features were used for feature extraction. Every pixel of $G(x, y)$ is described by 4-component vector of features. This components is statistical features of GLCM's. An occurrence of some gray-level configuration can be described by a matrix of relative frequencies $P_{\theta,d}(G_1, G_2)$. It describes how frequently two pixels with gray-levels G_1, G_2 appear in the window separated by a distance d in direction θ. The information is extracted from the co-occurrence matrix that measures second-order image statistics. Every direction is describe using the following formulas:

$$P_{\theta,d}(G_1, G_2) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \begin{cases} 1, \text{if } G_{qua}(x, y) = G_1 and\ G_{qua}(x + \theta * d, y + \theta * d) = G_2; \\ 0, otherwise. \end{cases}$$

$$N_P = \sum_{G_1=0}^{G_{MAX}-1} \sum_{G_2=0}^{G_{MAX}-1} P_{\theta,d}(G_1, G_2)$$

$$P_{\theta,d}(G_1, G_2) = \frac{P_{\theta,d}(G_1, G_2)}{N_P}$$

It's considered $N * N$-window of (x, y)-pixel neighbors. Here $N = 2 * R + 1$, R - radius of pixel neighborhood. It's considered 0°-,45°-, 90°-, 135°-directions. In pixel coordinate space it looks

as $(0,1)$, $(1,1)$, $(1,0)$, $(0,-1)$ vectors. It means that 4 GLC matrixes are used (for 4 direcions and $d = 1$) for feature calculation. The following statistical characteristics are calculated by $P_{\theta,d}(I_1, I_2)$ [4]:
angular second moment (homogeneity),

$$ASM_{\theta,d} = \sum_{G_1=0}^{G_{MAX}-1} \sum_{G_2=0}^{G_{MAX}-1} P_{\theta,d}(G_1, G_2)^2$$

contrast,

$$CONT_{\theta,d} = \sum_{G_1=0}^{G_{MAX}-1} \sum_{G_2=0}^{G_{MAX}-1} P_{\theta,d}(G_1, G_2) * (G_1 - G_2)^2$$

correlation,

$$COR_{\theta,d} = \sum_{G_1=0}^{G_{MAX}-1} \sum_{G_2=0}^{G_{MAX}-1} \frac{G_1 * G_2 * P_{\theta,d}(G_1, G_2) - \mu_x * \mu_y}{\sigma_x * \sigma_y}$$

$$\mu_x = \sum_{G_1=0}^{G_{MAX}-1} \sum_{G_2=0}^{G_{MAX}-1} G_1 * P_{\theta,d}$$

$$\mu_y = \sum_{G_1=0}^{G_{MAX}-1} \sum_{G_2=0}^{G_{MAX}-1} G_2 * P_{\theta,d}$$

$$\sigma_x = \sum_{G_1=0}^{G_{MAX}-1} (G1 - \mu_x)^2 \sum_{G_2=0}^{G_{MAX}-1} G_1 * P_{\theta,d}$$

$$\sigma_y = \sum_{G_2=0}^{G_{MAX}-1} (G2 - \mu_x)^2 \sum_{G_1=0}^{G_{MAX}-1} G_1 * P_{\theta,d}$$

inverse difference moment (local homogeneity)

$$IDM_{\theta,d} = \sum_{G_1=0}^{G_{MAX}-1} \sum_{G_2=0}^{G_{MAX}-1} \frac{1}{1 + (G_1 - G_2)^2} * P_{\theta,d}(G_1, G_2).$$

$ST_{\theta,d} = [ASM_{\theta,d}, CONT_{\theta,d}, COR_{\theta,d}, IDM_{\theta,d}]$ is vector of statistical characteristics for every direction θ. The feature vector for current pixel is calculated as:

$$ST_d = \frac{1}{4} \sum_{\theta} ST_{\theta,d}$$

3.3 Pixel classification

For every pixel of image feature vector is calculated by GLCM. Using k nearest neighbors (kNN) [13]. The pixel is classified as neuron pixel or non-neuron pixel. Method kNN is one of the most intuitive classification algorithms. Since We use the feature vectors. In the kNN procedure Euclidean distance is calculated from feature vector of current pixel to every feature vector of train base. After that only K nearest test vectors are considered. The class of the current pixel is class which owns the most cout of nearest train feature vectors.

3.4 Pixel clusterization

This is a calculation of the cluster centers as the mass center for the pixels which were classified as "neuron"(x, y) of the radius R_{ms}. In this paper $R_{ms} = 5$. Meanshift clusterization algorithm with simple filtration is used for neuron centers detection. It consists from the following steps: For every pixel:

1. Calculate density

$$K(x,y) = \sum_{i=x-R_{ms}}^{x+R_{ms}} \sum_{j=y-R_{ms}}^{y+R_{ms}} \begin{cases} 1, \text{if } I(i,j) \text{ is neuron;} \\ 0, otherwise \end{cases}$$

$$K(x,y) = \begin{cases} \frac{1}{(2R_{ms}+1)(2R_{ms}+1)} K(x,y), \text{if } K(x,y) > T_{ms}; \\ 0, otherwise \end{cases}$$

2. Calculate mass center (vector of two components):

$$m(x,y) = \frac{\sum_{i=x-R_{ms}}^{x+R_{ms}} \sum_{j=y-R_{ms}}^{y+R_{ms}} K(i,j) * [i,j]}{\sum_{i=x-R_{ms}}^{x+R_{ms}} \sum_{j=y-R_{ms}}^{y+R_{ms}} K(i,j)}$$

3. Calculate mean shift as distance $dist$ between $m(x, y)$ and (x, y). If $dist > \varepsilon$ then $(x, y) = mc(x, y)$ and repeat all steps. Otherwise, go to next pixel.

4 Quality detection metrics

The following numbers are calculated: the number of correct detections of neuron centers (TP, true positive), the number of false detections of neuron centers (FP, false positive) and the number of undetected centers of neurons (FN, false negative)e test base. The following estimates of the detection quality are used:

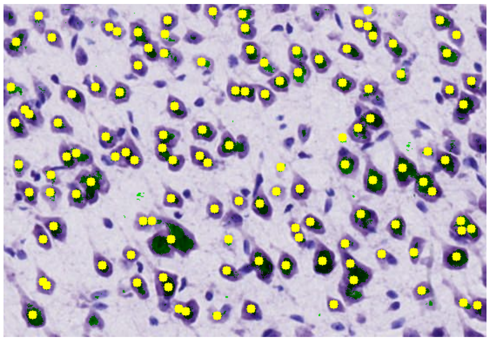
$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

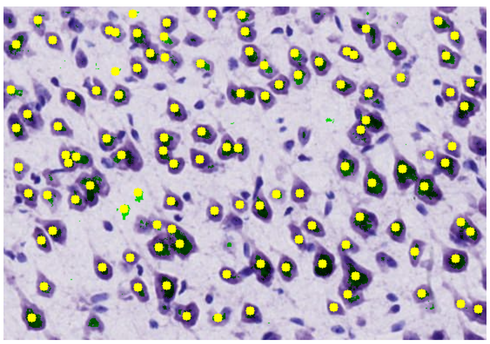
$$F1 = \frac{2 * Recall * Precision}{Recall + Precision}$$

To determine TP, FP and FN the following procedure is used for detected centers c_i and neuron centers m_j.

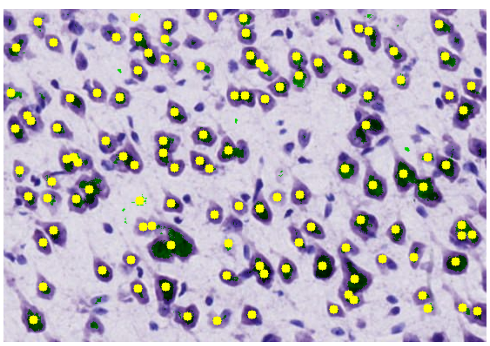
1. $TP = 0$, $FP = 0$, $FN = 0$.
2. For every detected center c_i:
   () Find the nearest m_{near} for every c_i.
   () If $\|m_{near} - c_i\|^2 < R_{acc}$, then $TP = TP + 1$ and remove c_i and m_{near} from searching. Otherwise $FP = FP + 1$
3. For every c_i for which m_{near} wasn't found $FN = FN + 1$



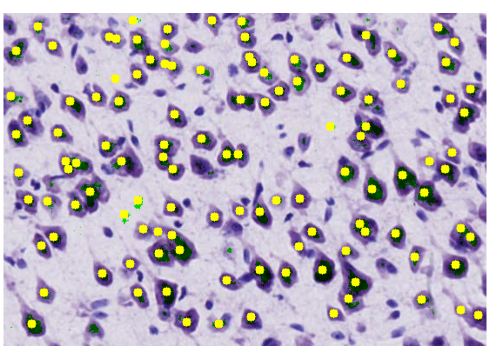
(a) Gray



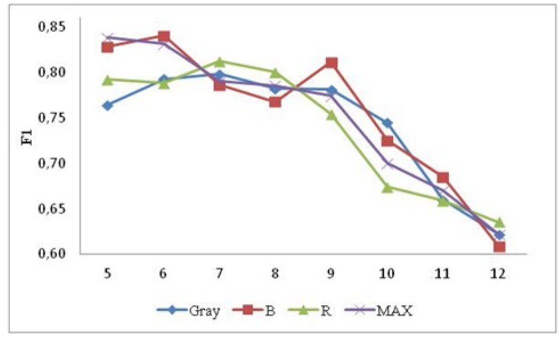
(b) B



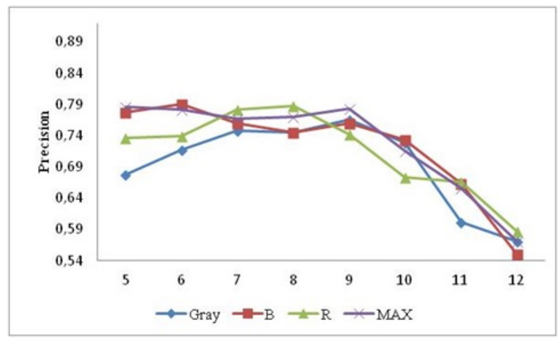
(c) R



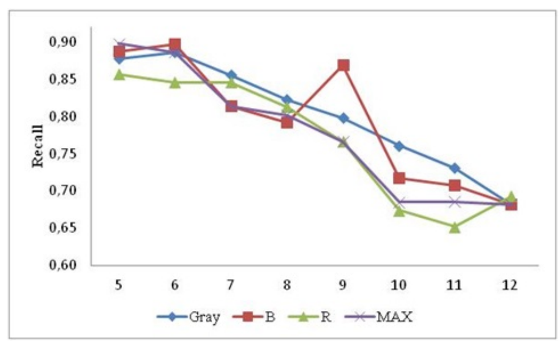
(d) Max



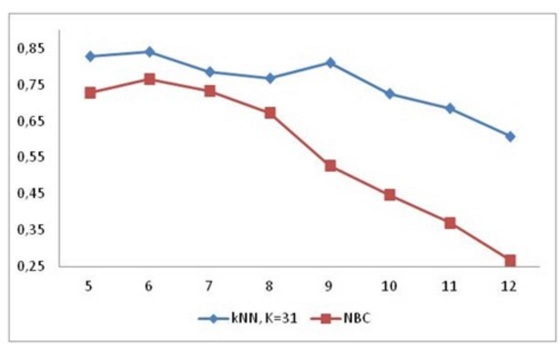
(e) F1



(f) Precision

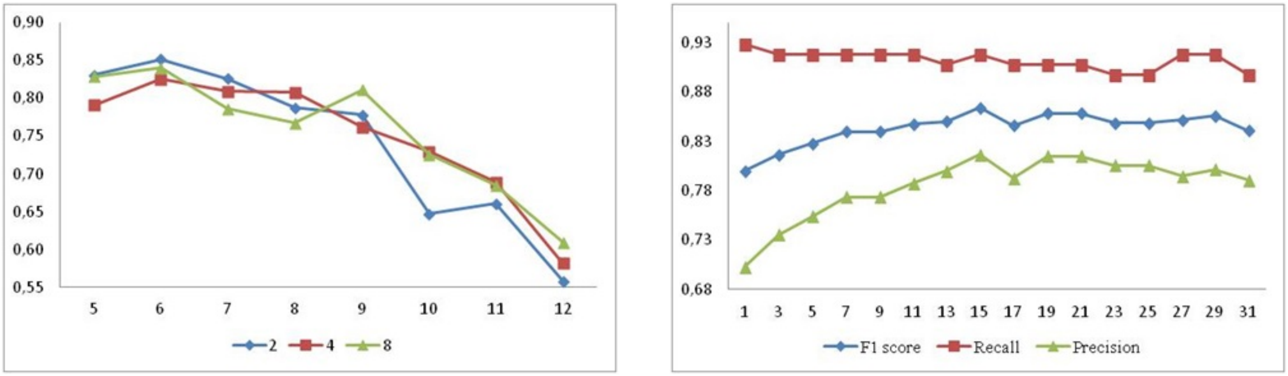


(g) Recall



(h) kNN,NBC

Figure 4 Detection results



(a) Bins (b) K

Figure 5 Detection quality dependence from histogram bin number (a) and k number in kNN method

5 Results

Set of experiments was developed to test detection quality. In our experiments $R_{acc} = 10$, $R_{ms} = 5$, $T_{ms} = \frac{N*N}{4}$, $N = 2 * R + 1$, $R = [5..12]$. kNN realizatin from OpenCV library was used [14]. Detection result dependence from choice of gray transformation algorithm one can see on 4(a-d). Dark green pixels are neuron pixels after classification step. In this experiment kNN algorithm was used and $k = 31$. Quality characteristics are demonstrated on 4(e-g). One can see that gray filtering using blue color channel gives the best result. In the next tests this transformation was used. On the 4(h) comparison detection qualities with kNN and normal bayes classificator (NBC) in OpenCV realization. $K = 31$ kNN fives better result.

Dependence detection quality from K is demonstrated on 5(b). In this test $R = 6$. In our tests $K > 15$ doesn't give increasing quality of detection. Dependence detection quality from K is demonstrated on 5(b). In this test $R = 6$. The best quality is Recall=0,82, Precision=0,92, F1=0,86. In our tests $K > 15$ doesn't give increasing of detection quality.On the 5(a) one can see dependence detection quality from used number of histogram bins. It should be noticed that in $R = 6$ the qualities of detection are very close to each other.

6 Concluding Remarks

The method for neuron detection based on texture features constructed via GLCM was developed. The method includes the following steps: image preprocessing, pixel classification, pixel clusterization.

Different transformations to grayscale were applied and studied. It is noticed that gray convertation by blue component gives the best result. It is shown that using 2-,4- bin histogram gives close detection quality with 8-bin histogram ($F1 = 0,83..0,85$). 2-bin histogram is histogram of binary image. Binarization algorithms should be studied for preprocessing step. Also GLCM texture features calculations for 16-bin histogram should be developed. Perhaps, it'll give better result.

For pixel classification kNN algorithm was used. It's simple and powerfull algorithm for classification in medical problems. The results demonstrate that kNN is better choice for current task in comparing with NBC. For pixel clusterization we used Meanshift algorithm.

The best detection quality for given approach is $precision = 0,82, recall = 0,92, F1 = 0,86$. From [1] it is known recall characteristic (A): $A = 86 \pm 5\%$ neuron with $15 \pm 8\%$ error (mean$\pm$SD) on their datasets. Our result is some better in recall.

In our future work we'll prolong this investigation for great volume of dataset and special dataset for important diseases.

References

[1] Inglis A., Cruz L., Roe D. L., Stanley H. E., Rosene D. L., and Urbanc B. Automated identification of neurons and their locations. // *Journal of Microscopy.*, 2008.Vol. Jun;230(Pt 3):339-52. doi: http://dx.doi.org/10.1111/j.1365-2818.2008.01992.x

[2] Akram S.U., Kannala J., Eklund L., Heikkilä J. Cell Segmentation Proposal Network for Microscopy Image Analysis. In: Carneiro G. et al. (eds) Deep Learning and Data Labeling for Medical Applications. // *Lecture Notes in Computer Science*, 2016. vol 10008. doi: http://dx.doi.org/10.1007/978-3-319-46976-8_3

[3] Ojala T., and Pietikäinen M. exture Classification, Machine Vision and Media Processing Unit. // *University of Oulu, Finland*, 2008.Vol. Jun;230(Pt 3):339–52. http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OJALA1/texclas.htm

[4] Haralick R. M., Shanmugam L., Dinstein I. Textural Features for Image Classification // *IEEE Trans. on Systems, Man and Cybernetics*, 1973.Vol. SMC-3, PP. 610–621.

[5] Gray Level Co-occurrence Matrix Filters (2014). TNTgis - Advanced Software for Geospatial Analysis. *MicroImages*, 2018. URL: http://www.microimages.com/documentation/TechGuides/81GLCM_Filters.pdf

[6] Bino Sebastian. V., Unnikrishnan A., and Balakrishnan K., Shanmugam L., Dinstein I. Grey Level Co-occurrence Matrices: Generalisation and some new features. // *Int. J. of Comp. Science, Engineering and Information Technology (IJCSEIT)*, 2012. Vol. 2,No. 2 PP. 151–157. doi: http://dx.doi.org/10.5121/ijcseit.2012.2213

[7] Rampun A., Strange H., and Zwiggelaar R. Texture segmentation using different orientations of GLCM features // *roceedings of the 6th International Conference on Computer Vision / Computer Graphics Collaboration Techniques and Applications - MIRAGE'13*, 2013.Article No. 17 doi: http://dx.doi.org/10.1145/2466715.2466720

[8] Pathak B., Barooah D. Texture Analysis Based On The Gray-Level Co-Occurrence Matrix Considering Possible Orientations. // *Int. J. of Advanced Research in Electrical, Electronics and Instrumentation Engineering (IJAREEIE)*, Sept. 2013. Vol. 2, Is. 9, PP.4206–4212

[9] Subudhi A., Sahoo S., Biswal P., Sabut S. Segmentation and classification of ischemic stroke using optimized features in brain MRI. // *Biomedical Engineering: Applications, Basis and Communications*, 2018. Vol. 30, No. 3, 1850011 (13 pages) BibDoi10.4015/S1016237218500114

[10] CUDA. URL: https://developer.nvidia.com/cuda-zonef (2018).

[11] Cheng Y. Mean Shift, Mode Seeking, and Clustering // *IEEE transactions on pattern analysis and machine intelligence*, 1995.Vol. 17, N. 8, PP. 790–799. URL: https://members.loria.fr/MOBerger/Enseignement/Master2/Exposes/meanShiftCluster.pdf

[12] Kanan C., Cottrell G. W. Color-to-Grayscale: Does the Method Matter in Image Recognition? // *PLoS ONE 7(1)*, 2012. e29740. doi: http://dx.doi.org/doi:10.1371/journal.pone.0029740

[13] Bishop C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*/ New York, NY, USA: Springer; 2007.

[14] OpenCV. URL: https://docs.opencv.org/. (2018).

[15] Yang Y. B, Elbuken C., Ren C. L., Huissoon J. P. Image processing and classification algorithm for yeast cell morphology in a microfluidic chip // *Journal of Biomedical Optics*, 2011, 16(6). doi: http://dx.doi.org/10.1117/1.3589100

Received December 21, 2018

GLCM, kNN and Meanshift в задаче детектирования нейронов по изображениям срезов мозга, окрашенных по Нисслю

С. А. Носова, В. Е. Турлапов
svetlana.nosova@itmm.unn.ru; vadim.turlapov@gmail.com
Нижегородский государственный университет им. Н.И. Лобачевского

Разработан метод обнаружения нейронов на изображениях срезов мозга, окрашенных по Нисслю. Метод использует текстурные признаки нейронов, построенные на основе 4х матриц взаимой встречаемости (GLCM). Метод включает в себя следующие этапы: предобработка изображений, классификация пикселей по текстурным признакам алгоритмом kNN и кластеризация пикселей нейронов алгоритмом Meanshift. Предобработка включает в себя следующие шаги: конвертация в оттенки серого, выравнивание гистограммы, квантование гистограммы. Применены и изучены различные способы преобразования цветного изображения в оттенки серого. Наилучший результат дает преобразование по синей компоненте цвета. Показано, что использование квантования гистограммы на 2 и 4 бина дает близкое качество детектирования с квантованием на 8 бин ($F1 = 0,83..0,85$). Результаты показывают, что kNN является лучшим выбором для текущей задачи классификации по сравнению с NBC. Наш алгоритм обеспечивает следующее качество детектирования:$precision = 0,82; recall = 0,92; F1 = 0,86$. Предложенный метод показа лучший результат по сравнению с аналогами. Планируется продолжить исследования на расширенном наборе данных и данных с социальнозначимыми заболеваниями мозга.

Ключевые слова: *машинное обучение; изображения оптической микроскопии; Ниссл-окрашивание; классификация; кластеризация; текстурные характеристики; kNN; GLCM*

DOI: 10.21469/22233792.4.3.04

Литература

[1] *Inglis A., Cruz L., Roe D. L., Stanley H. E., Rosene D. L., and Urbanc B.* Automated identification of neurons and their locations. // Journal of Microscopy., 2008.Vol. Jun;230(Pt 3):339-52. doi: http://dx.doi.org/10.1111/j.1365-2818.2008.01992.x

[2] *Akram S.U., Kannala J., Eklund L., Heikkilä J.* Cell Segmentation Proposal Network for Microscopy Image Analysis. In: Carneiro G. et al. (eds) Deep Learning and Data Labeling for Medical Applications. // Lecture Notes in Computer Science, 2016. vol 10008. doi: http://dx.doi.org/10.1007/978-3-319-46976-8_3

[3] *Ojala T., and Pietikäinen M.* exture Classification, Machine Vision and Media Processing Unit. // University of Oulu, Finland, 2008.Vol. Jun;230(Pt 3):339–52. http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OJALA1/texclas.htm

[4] *Haralick R. M., Shanmugam L., Dinstein I.* Textural Features for Image Classification // IEEE Trans. on Systems, Man and Cybernetics, 1973.Vol. SMC-3, PP. 610–621.

[5] Gray Level Co-occurrence Matrix Filters (2014). TNTgis - Advanced Software for Geospatial Analysis. MicroImages, 2018. URL: http://www.microimages.com/documentation/TechGuides/81GLCM_Filters.pdf

[6] *Bino Sebastian. V., Unnikrishnan A., and Balakrishnan K., Shanmugam L., Dinstein I.* Grey Level Co-occurrence Matrices: Generalisation and some new features. // Int. J. of Comp. Science,

Engineering and Information Technology (IJCSEIT), 2012. Vol. 2,No. 2 PP. 151–157. doi: http://dx.doi.org/10.5121/ijcseit.2012.2213

[7] *Rampun A., Strange H., and Zwiggelaar R.* Texture segmentation using different orientations of GLCM features // roceedings of the 6th International Conference on Computer Vision / Computer Graphics Collaboration Techniques and Applications - MIRAGE'13, 2013.Article No. 17 doi: http://dx.doi.org/10.1145/2466715.2466720

[8] *Pathak B., Barooah D.* Texture Analysis Based On The Gray-Level Co-Occurrence Matrix Considering Possible Orientations. // Int. J. of Advanced Research in Electrical, Electronics and Instrumentation Engineering (IJAREEIE), Sept. 2013. Vol. 2, Is. 9, PP.4206–4212

[9] *Subudhi A., Sahoo S., Biswal P., Sabut S.* Segmentation and classification of ischemic stroke using optimized features in brain MRI. // Biomedical Engineering: Applications, Basis and Communications, 2018. Vol. 30, No. 3, 1850011 (13 pages) BibDoi10.4015/S1016237218500114

[10] CUDA. URL: https://developer.nvidia.com/cuda-zonef (2018).

[11] *Cheng Y.* Mean Shift, Mode Seeking, and Clustering // IEEE transactions on pattern analysis and machine intelligence, 1995.Vol. 17, N. 8, PP. 790–799. URL: https://members.loria.fr/MOBerger/Enseignement/Master2/Exposes/meanShiftCluster.pdf

[12] *Kanan C., Cottrell G. W.* Color-to-Grayscale: Does the Method Matter in Image Recognition? // PLoS ONE 7(1), 2012. e29740. doi: http://dx.doi.org/doi:10.1371/journal.pone.0029740

[13] *Bishop C. M.* Pattern Recognition and Machine Learning (Information Science and Statistics)/ New York, NY, USA: Springer; 2007.

[14] OpenCV. URL: https://docs.opencv.org/. (2018).

[15] *Yang Y. B, Elbuken C., Ren C. L., Huissoon J. P.* Image processing and classification algorithm for yeast cell morphology in a microfluidic chip // Journal of Biomedical Optics, 2011, 16(6). doi: http://dx.doi.org/10.1117/1.3589100

Поступила в редакцию 21.12.2018