

CLASSIFICATION ALGORITHMS FOR THE DETECTION OF THE PRIMARY TUMOR BASED ON MICROSCOPIC IMAGES OF BONE METASTASES

Sladjan Kantar¹, Aleksandar Pluškoski², Igor Ciganovic³ and Jelena Vasiljević⁴

^{1,2,3}Master student at “The Faculty of Computer Science”

⁴Professor at “The Faculty of Computer Science”,
University of Belgrade, Serbia

Science Assistant in “Institute Mihajlo Pupin”

ABSTRACT

This paper presents the analysis of techniques for microscopic images in order to find a primary tumor based on the of bone metastases. Was done algorithmic classification into three groups, kidney, lung and breast. In order to speed up the treatment of the patient and easier for doctors and therefore reduce room for human error. Digital microscope images of bone metastases were analyzed, for which it is known that the primary tumor is in one of the three human organs: kidney, lung or breast. We tested several solutions for classification, were tested two methods of image analysis. Multifractal analysis and convolutional neural network. Both methods were tested with and without preprocessing image. Results of multifractal analysis were then classified using different algorithms. Images were processed using CLAHE and k-means algorithm. At the end, the results obtained using a variety of techniques are presented.

KEYWORDS

Cancer classification, Microscopic images, Image preprocessing, Multifractal analysis, Classification algorithms

1. INTRODUCTION

Cancer is the leading cause of death in the world. Every year more and more people being diagnosed, and advances in medicine, unfortunately, not fast enough, although in recent years much has been done. In computer science, many scientific papers dealing with automation of diagnosis and determining ways of treating cancer. Our work deals with the analysis of various techniques for algorithmic classification (determination) of a primary tumor based on the microscopic images of metastases in bones.

In the last few years, machine learning algorithms experienced the large popularization and expansion. It is increasingly used in various fields of industry and science. They became part of

the products we use every day. Also, used in medicine all over. However, these algorithms are statistically and require very large amounts of data for their application. In recent years, more and more of data become available and is the main cause of the increasing use of this type of algorithms. The basic idea of each of these algorithms is based on large amount of data to notice statistical similarity within class from training data and then attempt to extrapolate "learned" to new data and classified them into one of the given class.

When it comes to image analysis with machine learning algorithms, convolutional neural networks are unavoidable, but in medicine is also applied fractal and multifractal analysis, due to the observed fractal characteristics of medical images. Convolution neural network classification algorithm presents whole classifying algorithm that take images from the input and classifying it into one of the classes. In both cases, the images that are analyzed may be a "raw", obtained directly from the microscope, or preprocessed. We tested both cases, in order to compare results.

2. MODELS AND METHODS

Detection and classification of cancer using microscopic images with biopsy is the arduous and complex task. Many different techniques is entered in the proposed method, which includes, changing the contrast of the image, segmentation of the cell, multifractal analysis, and at the end classification. In the Figure1 show two methods for classification are presented. First proposed, for improving the microscopic images of tissue use CLAHE algorithm, which is then passed through the K-means algorithm for segmentation of the cell. With this method we have highlighted important biological and clinical forms, as well as morphological features, which include shades of gray, color segmentation and segmentation in texture ^{[1][2]}. In the end, it applies multifractal analysis and some clustering algorithm. The second proposal is that on microscopy images of biopsy put direct through multifractal analysis, and then clustering algorithm. These algorithms were tested on randomly selected 1050 (350 in each group - kidney, lung, breast) microscopic image of biopsy. At the end it was measured and analyzed performance of each classification algorithm. Below is described the work of each of the algorithms used in this experiment.

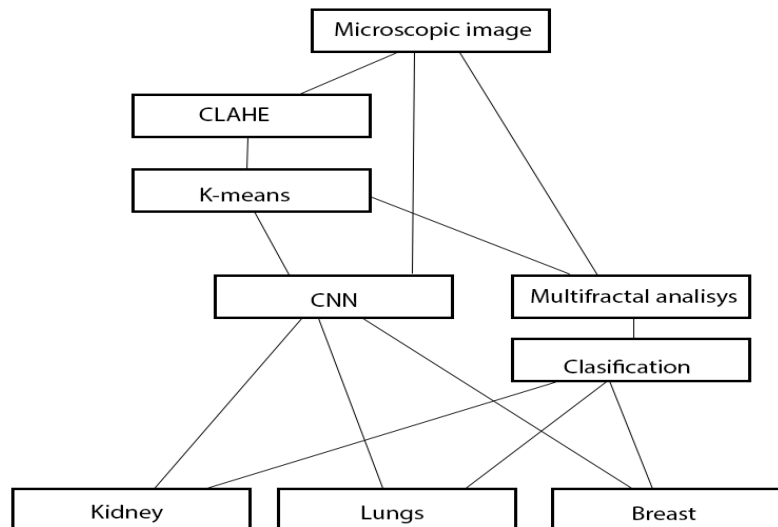


Figure 1

2.1. Image Enhancement

The main purpose was to reduce noise and to make higher contrast, so to express for use important regions. During the shooting of microscopic images it occurs image deformation due to small focal length (fisheye) and poor brightness. Thus, images are over or underexposed and even unclear edges. Due to all of this we have used CLAHE algorithm to enhance microscopic images. Figure 2 shows us image before and after CLAHE algorithm.

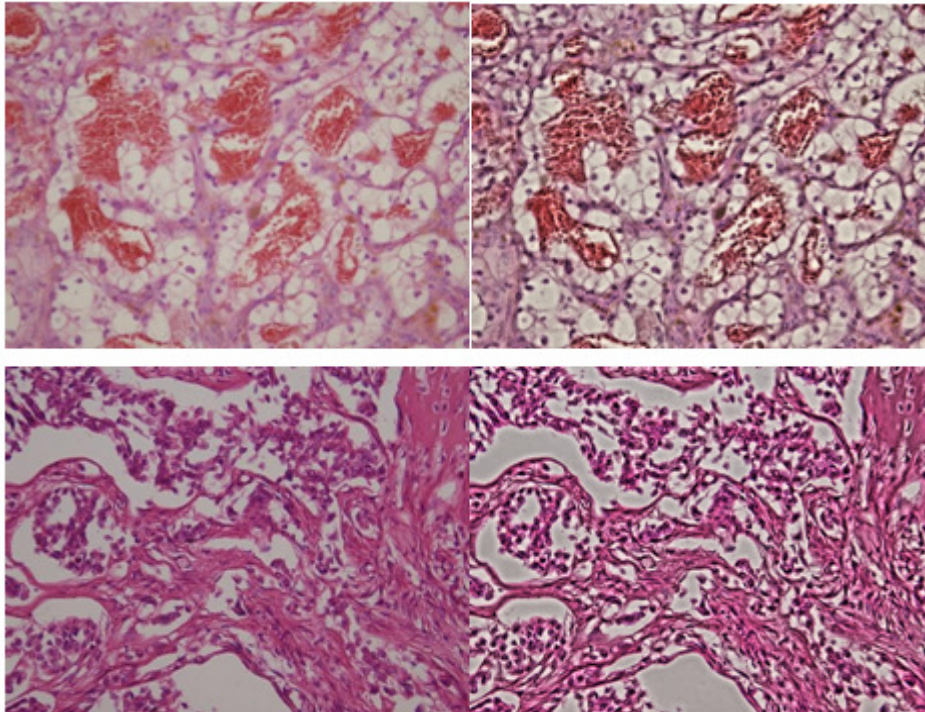


Figure 2. Left images are original, images on the right are after CLAHE algorithm.

2.2. Segmentation

There are two ways of image segmentation, based on cells, cytoplasm and nucleus. First way is to use human but it will take too much time and there is the possibility for human error. Second way, is to automate this process by using some types of algorithms. We have tested a lot of algorithms, but the best, and only useful, was k-means colors based segmentation. K-means is vector segmentation, first developed in signal processing. K-means algorithm first partition n objects into k clusters^[3]. Our microscopic images are 8-bit color. That means the maximum number of colors that can be displayed at any one time is 256. So we have clustered into 3 clusters ($k=3$). Now each picture is represented by three colors. With this method, we lose some data, but at the same time we simplify model that is used for image classification.

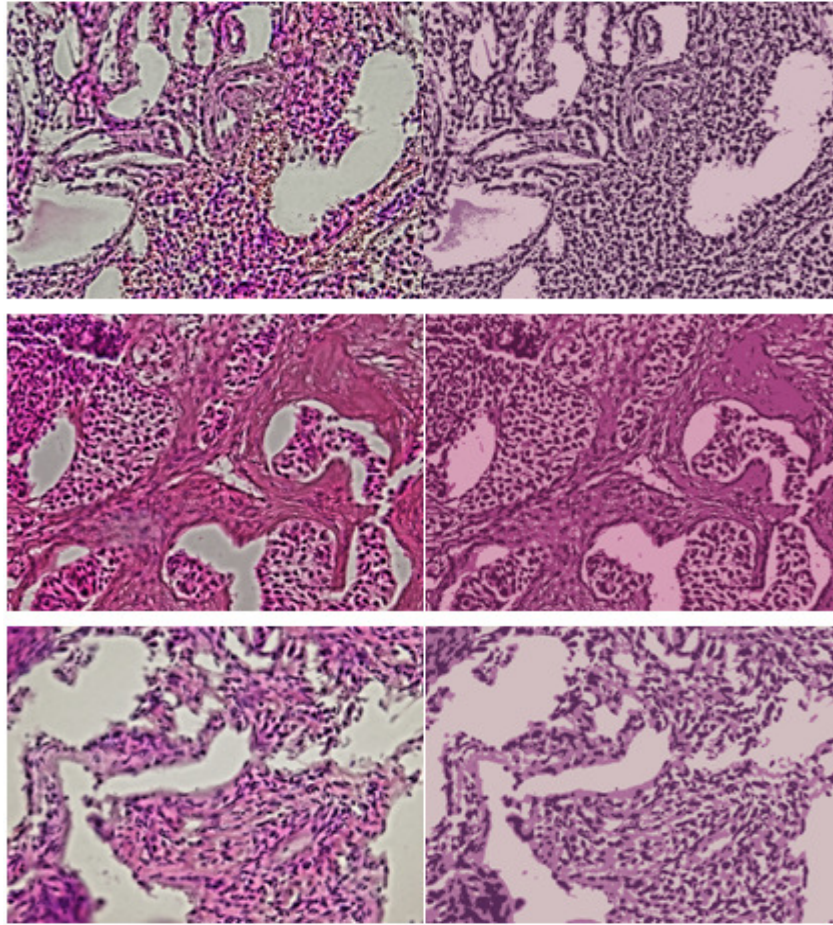


Figure 3. Images on the left before and on the right after k-means algorithm. Going from the top down, there are kidney, breast and lung.

2.3. Multifractal Analysis

For each of these three groups, 350 were selected Image (Total 1050). Multifractal analysis of digital medical images was done using ImageJ free program, and plug-in FracLac^[4]. The parameters that were passed to the program are as follows: Q in the range from -5 to +5 with step Qs of 0.25, number of scans per image was is April 1st.

2.4. Classification Algorithms

Classification is the most common task in machine learning. It's a problem of identifying the items belonging to a predefined set. The classification of an object is based on finding similarities with predefined objects that belongs to different classes. When classifying each object is assigned to a class, with a certain probability of accuracy. Classification algorithm has the task of creating model based on training data which to exert new classification of unknown object. In classification the number of classes is know in advance and is not changeable.

In this paper it was used python open source library scikit-learn 0.18. It was written in python, C and C++. Featuring a variety of algorithms for classification such as SVM, Nearest Neighbors,

Naive Bayes, Decision Trees, Ensemble methods; for regression and clusterization such as: K-Means and Spectral clustering ^{[5][6]}.

2.4.1. Linear Discriminant Analysis

LDA is a method often used in statistics and machine learning to find a vector (span or linear combination) that separates two or more classes. Result may be used as linear classifier or dimension reduction before applying other well-known classifiers.

LDA multi class classifier is generalization of LDA two class classifier that can work with arbitrarily number of classes. Multi class LDA is based on two scatter matrices: within class scatter matrix and between classes scatter matrix. Let's assume x_1, \dots, x_n is training data and their class labels are y_1, \dots, y_n , then scatter matrix within class is:

$$S_w = \sum_{i=1}^n (x_i - \mu_{y_i})(x_i - \mu_{y_i})^T$$

Where μ_k is mean values for class k.

Scatter matrix between classes can be written as:

$$S_b = \sum_{k=1}^m n_k (\mu_k - \mu)(\mu_k - \mu)^T$$

Here m is number of classes, μ is mean value, and n_k is number of samples in class K.

Then, multi class LDA can be written as a problem that find vector w which maximize difference between classes scatter and within class scatter.

$$\hat{w} = \operatorname{argmax}_w \frac{w^T S_b w}{w^T S_w w}$$

Generally, we need maximum m-1 vector to get discrete values for m classes. We can write it like:

$$S_b w = \lambda S_w w$$

This algorithm isn't recommended when we have a lot of classes, due to high complexity of calculating ^[7].

2.4.2. Gaussian naive Bayes

Naïve Bayes classifiers are a set of algorithms that uses Bayes theorem, assuming that training data is independently of each other. It's often used as basic classifier, because it's easy and fast to implement. This algorithm is degraded due to poor performance yet with good data preprocessing and well-tuned parameters it can be as good as SVM. ^{[8][9]}

There are a few types of naïve Bayes classifiers and difference between them is how they make distribution $P(x_i | y)$. However in nature random things aren't so random. If you look closely you will see some regularity and in the most of time it will be Gaussian probability, or often called normal distribution or bell curve due to its shape. Thus, Gaussian naïve Bayes classifier is one of the most widely used. ^{[5][6][10]} Let's assume that X is training data and Y is array of all classes which X_i can belongs to. First of all, we split training data by classes and for each class we find variance $2y$ and mean value y . Then probability distribution of class X_i is:

$$P(x_i | y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

2.4.3. SVM (SVC)

Support Vector Machine (SVM) is popular method for supervised machine learning for classification, regression and other machine tasks. SVM usually gives very accurate predictions, and they do not need too much parameters adjustment. They also are not sensitive on data size.

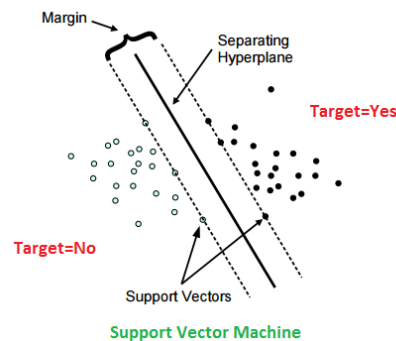


Figure 4. Schema of SVM algorithm

SVM first finds decision boundary. For this task it does not need all the training data. It's take data that are very close to other but are in class. Dots that are exactly on the boundary we call them support vectors, and that lines we call support planes. As we have sad earlier, we need boundary data, and now we calling them support vectors. By use of them we calculate separating (optimal) hyperplane.

How algorithm calculate boundary? First two hyperplanes that best represent data. For example, white dots (Target=No) are represent like $w^T x + b_0 > 1$, while black dots (Target=Yes) are represent like $w^T x + b_0 < -1$, where is w weight vector, x is set of training data i.e. vectors, and b_0 is bias. Weight vector decides orientation of decision boundary while bias decides its location. Now decision boundary is in the midway of these two value (hyperplanes). ^[11]

SVC algorithm is a wider set of code which is capable now just calculating locations of hyperplanes it's also classifying data inside. SVC as a input take array that classifying into n classes. ^{[5][6]}

2.4.4. K neighbors

K-NN is non parametric method for classification and regression, which is successful in cases where decision edge is irregular.^[12] Input in this algorithm is training data, which we get using multi fractal analysis. Output is the class to which it belongs. An object belong to that class, which have the most of his k neighbors ($k > 0$). Parameter k is defined before starting an algorithm. If $k=1$, then output is a class, which that one nearest object belongs. Distance between input data (points) can is calculated in different ways, but usually it is Euclid distance. To find right parameter k is complex process. It depends on type of training data. If it's k large, then we suppers noise, but on the same time we loses that precise border between classes.^{[5][6]} In this paper, we have three classes, based on a type of tissue (kidney, breast, and lung).

2.4.5. Decision Tree

DT is non parametric supervised learning used for classification and regression. The goal is to create a model that predicts the value by learning simple decision rules concluded from the training data.^{[5][6]} DT easy manipulate with big data and this makes it good for hyperspectral data.^[13] It is consists of nodes and edges. Each node is connected with a set of possible answer and in this way from input set of data we make smaller subset. Each represent possible test cases. DT like other machine learning algorithms uses recursion to make decision rules for classification from the training data.^[14]

2.4.6. Random Forests

RF ensemble learning method for classification and regression. In RF each decision tree is built during the training, and output is class that is most times appear as output on every single tree.^{[6][15]} This means that makes the diverse set of classifiers, by introducing randomness, at the very design RF classifier. RF pick random vector (X, Y) from training set. Vector $X=\{X_1, \dots, X_n\}$ and $\square X \in R^+$ contains predictors and $Y \in \bar{Y}$ where \bar{Y} is class label or a numerical response. Classifier t is a mapping:

$$t: R^+ \in \bar{Y}$$

In RF, every tree in forest depends on the one random vector from training data and is equal probability of distribution for all trees.

2.4.7. Convolutional Neural Network

A convolutional neural network (CNN) is a type of neural network that is used primarily for pattern recognition. It is mostly used in image recognition problems. We will talk about two dimensional convolutional neural network, but in practice number of dimensions that we can use are not limited. The idea behind CNN is to make neural network invariant to certain input transformations. This is the problem with fully connected neural networks, because layers are fully connected with each other which removes spatial information of the input.

CNN involves special convolutional and pooling (sub-sampling) layers. Output of which is usually flattened (transformed form matrix to vector) and used as input for fully connected network for classification. The architecture of CNN is shown in Figure 5. Convolution and sub-

sampling operations can be repeated as many times as needed. There are three basic principles behind CNN, local receptive fields, weight sharing and sub-sampling.

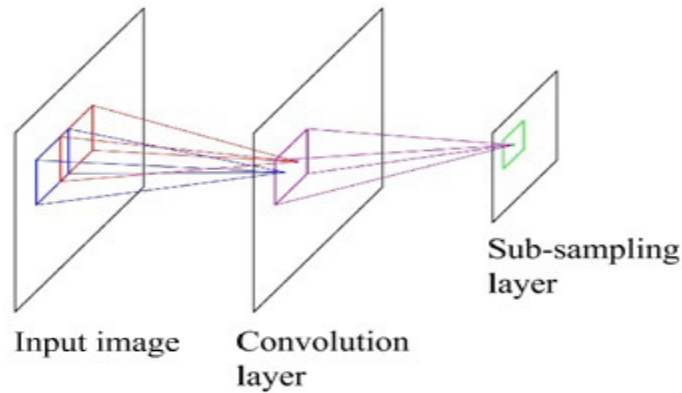


Figure 5. The architecture of a standard CNN. It shows a convolutional layer and is followed by a pooling layer.

2.4.7.1. Local receptive fields

In the fully connected feed forward neural network every neuron from one layer is connected to every neuron from the next. In CNN this is not the case. Every input in a convolutional layer is only connected to a small field of the previous layer. This is called local receptive field. In this way we preserve spatial information from the input. Figure 6. illustrates this idea. Here we have the first neuron of the convolutional layer connected only to nine pixels in the small 3 x 3 area of the input layer.^[11]

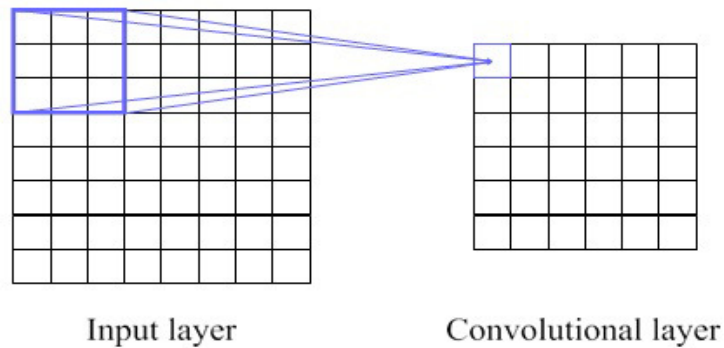


Figure 6. An example of local receptive fields on a convolutional layer.

2.4.7.2. Weight sharing

Each convolutional layer is organized into multiple parallel hidden layers called feature maps. Neurons from different feature maps with the same index are always connected to the same area of the input. The idea of weight sharing is that all the neurons from the same feature map share the same weights, also called filter or kernel. Figure 7. Illustrates this.^[16]

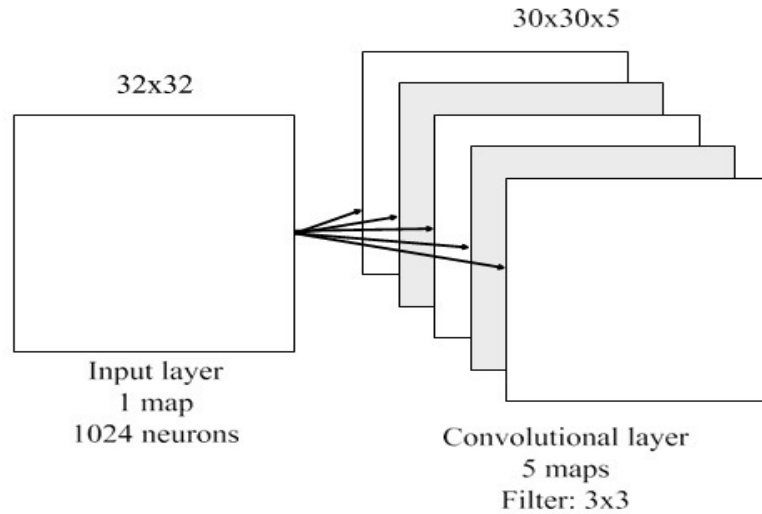


Figure 7. An example of a convolutional layer. The filter size is 3×3 and 5 feature maps.

2.4.7.3. Sub-sampling

Sub-sampling layer, also called pooling layer, is usually used immediately after a convolutional layer. Purpose of pooling layer is to create translation invariant features. This is done by computing statistical information of the activations from receptive field in the feature map of the convolutional layer. Figure 8. Illustrates process of sub-sampling.^[16]

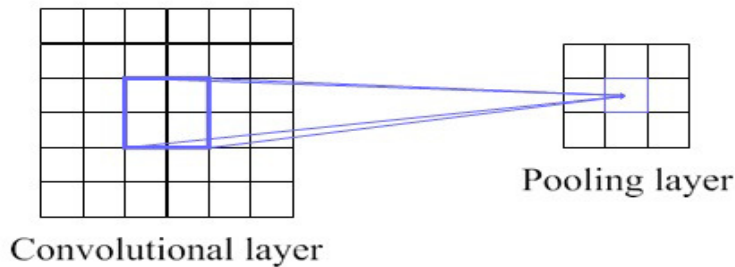


Figure 8. An example of a pooling layer in a feature map. This example uses 2×2 pooling size.

3. RESULTS AND ANALYSIS

3.1 CNN

We used convolutional neural network with relatively simple architecture built using TensorFlow framework. Consisting of up to six convolutional layers, all paired with pooling layer. 'ReLU' was used as the activation function. Parameters used for the convolutional layers were strides = [1, 1, 1, 1] and padding = 'SAME' and for the pooling layer strides = [1, 1, 1, 1], ksize = [1, 2, 2, 1] and padding = 'SAME'. Number of feature maps was calculated using formula:

$$n = 2^{4+i},$$

where 'i' is index of the current convolutional layer. The output of the last pooling layer was used as an input into fully connected network consisting of two hidden layers and an output layer. First fully connected layer had 30% dropout rate, to reduce overfitting. Activation function used for hidden layers was 'sigmoid' and 'softmax' was used for output. 'AdamOptimizer' was used to train the network. Network was trained using different hyperparameters and using two data sets, processed and raw images. Images in both data sets were resized to 300 x 226 resolution, to reduce network complexity. Results are shown in Figure 9 and 10, showing accuracy progression over time for training using raw and preprocessed data sets respectively. Legends are shown separately for better visibility. Different colored lines represent different learning rates used and different number of convolutional layers.

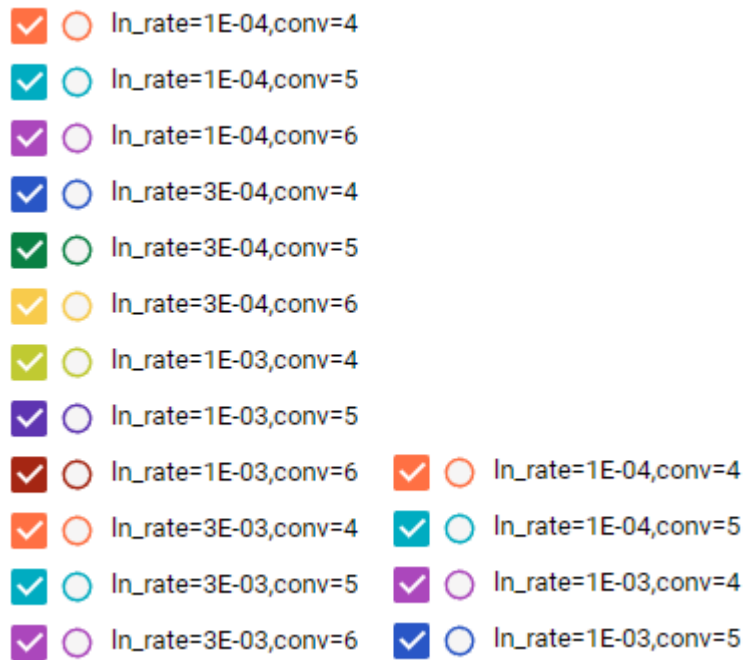


Figure 8. Legend the left for the plot representing accuracy for raw data set. Legend on the right for the plot representing accuracy for preprocessed data set.

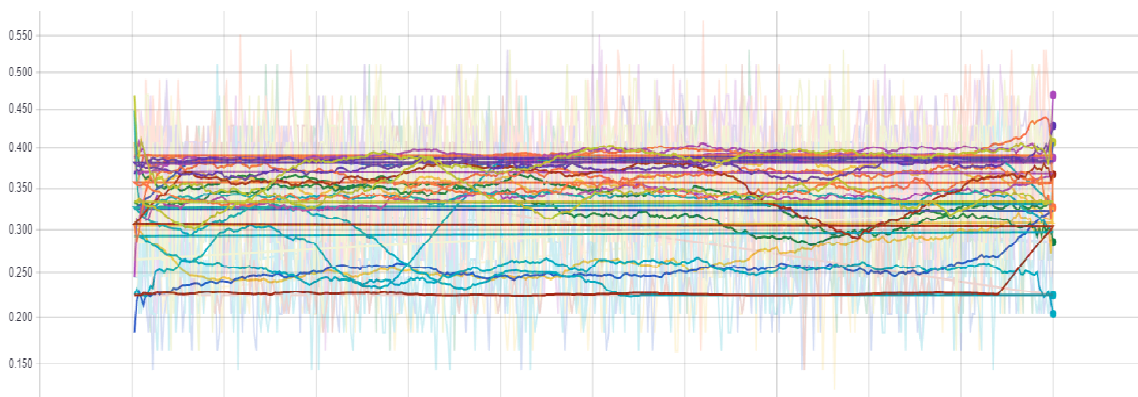


Figure 9. Accuracy over time for raw data set.

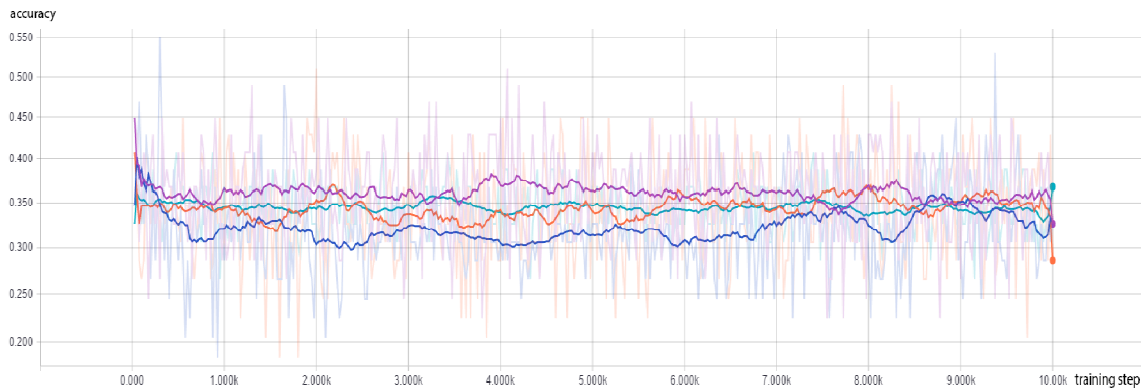


Figure 10. Accuracy over time for preprocessed data set.

As we can see from the graphs, simple convolutional neural networks are not a good solution to this kind of problems. We can see that some configurations of hyperparameters perform better than other. But with all configurations network converges very quickly and does not continue to learn new features. From the results we can see that there is no big difference between raw and preprocessed images.

Results show better than random guessing accuracy. Our opinion is that more uniform data set and more complex neural network architecture would provide better results. But experimenting with different network architectures is beyond the scope of this project.

3.2 Classification Algorithms

Since this is a classic classification problem, we test some well known machine learning algorithms. Results are show in the figure

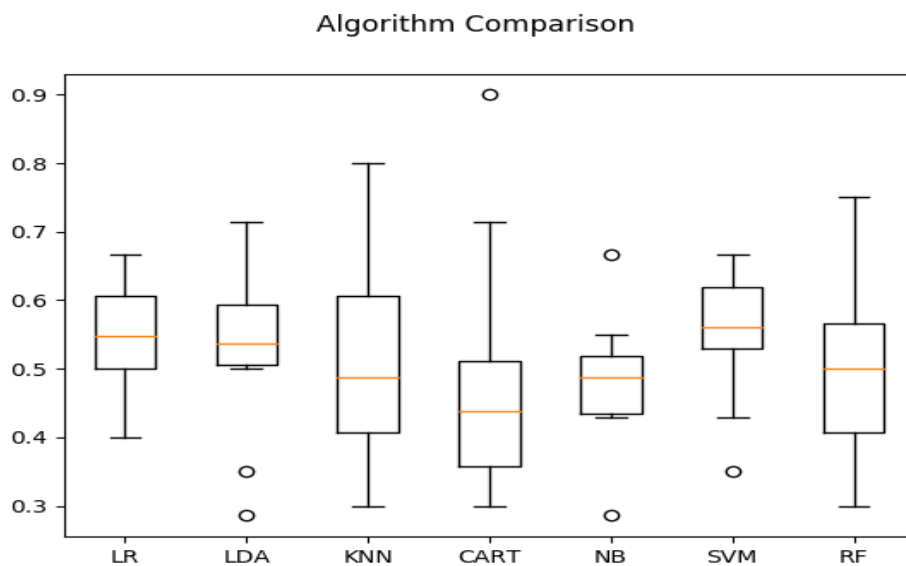


Figure 11: Done multiple of cross validation. Orange lines show average accuracy and boxes' shows average variance.

3.2.1 KNN analysis

We saw that the KNN has the greatest accuracy. We approached the examination of the result for this algorithm. First step was feature selection.

Feature selection

Feature selection is a process where you automatically select those features in your data that contribute most to the prediction variable or output in which you are interested.

Having too many irrelevant features in your data can decrease the accuracy of the models. Three benefits of performing feature selection before modeling your data are:

- Reduces Overfitting: Less redundant data means less opportunity to make decisions based on noise.
- Improves Accuracy: Less misleading data means modeling accuracy improves.
- Reduces Training Time: Less data means that algorithms train faster.

For our features we have following data shown in a figure XXX

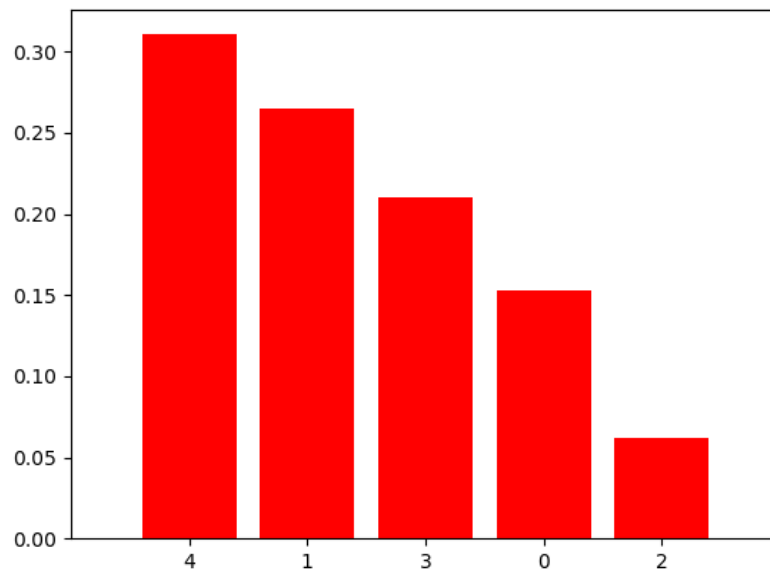


Figure 12 Shows feature importance: 1. feature 4 (0.310558) 2. feature 1 (0.264456) 3. feature 3 (0.210032) 4. feature 0 (0.152823) 5. feature 2 (0.062130)

We used 3 most important features to see disposition of organs:

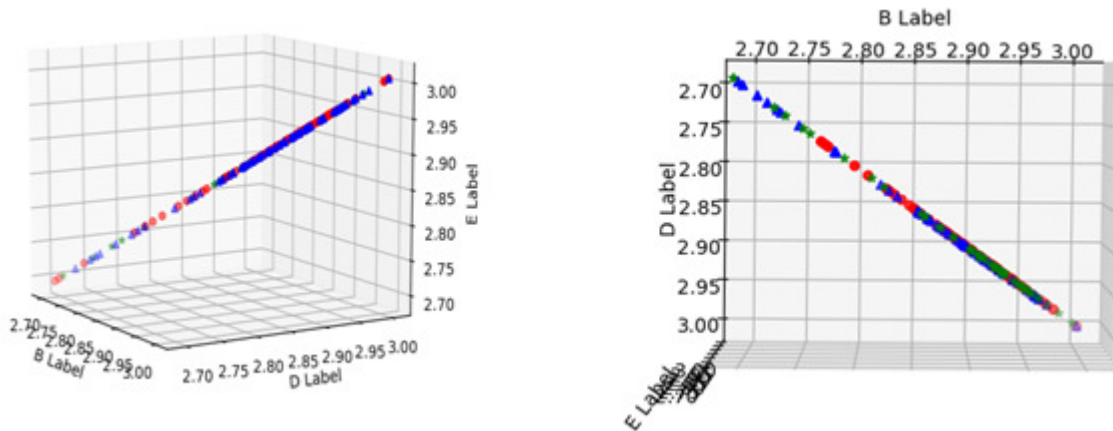


Figure 13: red dots - Breast, blue dots - Kidney, green dots- Lungs. Picture on the right is from the top of diagram.

Algorithm tuning is a final step in the process of applied machine learning before presenting results.

It is sometimes called Hyperparameter optimization where the algorithm parameters are referred to as hyperparameters whereas the coefficients found by the machine learning algorithm itself are referred to as parameters. Optimization suggests the search-nature of the problem.

Phrased as a search problem, you can use different search strategies to find a good and robust parameter or set of parameters for an algorithm on a given problem.

We used the random search as an approach to parameter tuning that will sample algorithm parameters from a random distribution (i.e. uniform) for a fixed number of iterations. A model is constructed and evaluated for each combination of parameters chosen.

As the best parameters we got: **n_neighbors=11 metric=euclidean** and results shown in table XXX

Table 1: Shows accuracy of KNN algorithm for each tissue and total.

	Total	Correct	Accuracy
Lungs	14	7	50.00%
Kidney	14	8	57.14%
Breast	14	13	92.85%
Total	42	28	64.29%

3.2.2 SVC analysis

This algorithm has the best results in algorithm comparison show in figure XXX We also used random search for algorithm tuning and the best parameters found on development set are: **C=1000, kernel=linear**.

Table 2. Shows accuracy of SVC algorithm for each tissue end total

	Total	Correct	Accuracy
Lungs	11	8	72%
Kidney	9	7	78%
Breast	8	6	75%
Total	28	21	75%

ACKNOWLEDGEMENTS

This work was supported by the Ministry of Education and Science, Republic of Serbia, Science and Technological Development grants III 45005 and TR32037.

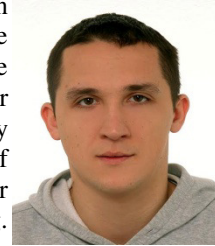
REFERENCES

- [1] E. D. P. Z. M. H. D. E. J. M. P. B. M. Pizer, "Contrast Limited Adaptive Histogram Equalization image processing to improve the detection of simulated spiculations in dense mammograms," *Journal of Digital Imaging*, vol. 193, no. 11, 1998.
- [2] R. J. R. P. G. a. H. P. Srivastava, "Enhancement and restoration of microscopic images corrupted with poisson's noise using a nonlinear partial differential equation-based filter," *Defence Science Journal*, vol. 61, no. 5, p. 452, 2011.
- [3] S. Z. B. M. H. E. D. Pisano, "Contrast Limited Adaptive Histogram Equalization image processing to improve the detection of simulated spiculations in dense mammograms," *Journal of Digital Imaging*, vol. 11, no. 4, pp. 192-200, 1998.
- [4] G. Landini, "Fractals in microscopy," *Jurnal of Microscopy*, vol. 241, no. 1, pp. 1-8, 2011.
- [5] F. V. G. G. A. M. V. T. B. G. O. B. M. P. P. W. R. V. V. J. P. A. C. D. B. M. P. M. D. E. Pedregosa, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 12 10 2011.
- [6] L. B. a. G. L. a. M. B. a. F. P. a. A. M. a. O. G. a. V. N. a. P. P. a. A. G. a. J. G. a. R. L. a. J. V. a. J. a. Bria, "API design for machine learning software: experiences from the scikit-learn project," *European Conference on Machine Learning and Principles and Practices of Knowledge Discovery in Databases*, pp. 108-122, 2013.
- [7] Y. Anzai, *Pattern recognition and machine learning*, Harcourt Brace Jovanovich, 2012.
- [8] L. S. J. T. D. R. K. Jason D. M. Rennie, "Tackling the Poor Assumptions of Naive Bayes Text Classifiers," in *ICML*, 2013.
- [9] L. D.D., "Naive (Bayes) at forty: The independence assumption in information retrieval.," *Machine Learning: ECML-98.*, vol. 1398, pp. 4-15, 1998.
- [10] B. G. K. M. S. Mohammad Ahsanullah, *Normal and Student's t Distributions and Their Applications*, vol. 4, *Atlantis Studies in Probability and Statistics*, 2014, pp. 7-50.

- [11] A. M. & A. K, "OpenCV-Python Documentation", 2017. [Online]. Available: <https://media.readthedocs.org/pdf/opencv-python-tutroals/latest/opencv-python-tutroals.pdf>. [Accessed 23 05 2017].
- [12] N. S. Altman, "An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression," *The American Statistician*, vol. 46, no. 3, pp. 175-185, 2012.
- [13] K. E. F. Glenn De'ath, "CLASSIFICATION AND REGRESSION TREES: A POWERFUL YET SIMPLE TECHNIQUE FOR ECOLOGICAL DATA ANALYSIS," *Ecology*, vol. 81, no. 11, pp. 3178-3192, 2000.
- [14] S. Mannel, "Decision Tree Classification of a Forest Using AVIRIS and Multi-Seasonal TM Data," *South Dakota School of Mines and Technology, Rapid City, South Dakota*, 2012.
- [15] R. T. J. F. Trevor Hastie, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, 2009.
- [16] DewiSuryani, S.KOM., M.ENG., "CONVOLUTIONAL NEURAL NETWORK", 2017. [Online]. Available: <http://socs.binus.ac.id/2017/02/27/convolutional-neural-network> [Accessed 27 05 2017].

AUTHORS

Sladan Kantar was born on 08/12/1993. in Knin, Republic of Croatia. As a young man he came in Arandelovac, Serbia, where he finished elementary and high school. He enrolled Faculty of Computer Science, University UNION in Belgrade where he graduated at the Department of Computer Science in 2016. with the topic "Module for automatically testing programming homework solutions (Grader)". He has successfully ends postgraduate studies at the Faculty of Computer Science. From the beginning of the school year 2014/15 he worked as teaching assistant at the Faculty of Computer Science in subject: compiler construction and introduction to computer programming. He is temporary senior software developer in company Moodolar, located in Belgrade.



Scientific and professional production S. Kantar, M. Jovanović, "Implementation of the control algorithm for the 'buggy' mobile robot via application of infrared distance measuring sensors", *International Conference on Electrical, Electronic and Computing Engineering, Kladovo, Serbia, June (2017)*

Aleksandar Pluškoski was born on 10/12/1993. in Belgrade, Republic of Serbia. Where he was finished elementary and high school. In 2012 he enrolled in The Faculty of Computer Science, University UNION in Belgrade where he graduated at the Department of Computer Science in 2016 with the topic "Alternatives of MS Project, like Project Libre". From the beginning of the school year 2016/2017 he worked as teaching assistance at The Faculty of Computer Science in subject: advance Java programming and web programming. He is successfully finishing postgraduate studies at the Faculty of Computer Science.



Igor Ciganović was born on 08/06/1985. in Dubrovnik, Republic of Croatia. As a young man he came to Pančevo, Serbia, where he finished elementary and high school. He enrolled in Faculty of Computer Science, University UNION in Belgrade where he graduated at the Department of Computer Science in 2016. with the topic "Implementation of the mechanism for communication between extern service for automatic code grading and Open edX platform and grade calculation". He is successfully finishing postgraduate studies at the Faculty of Computer Science.



Jelena Vasiljevic works as Science Assistant in Institute Mihajlo Pupin in Belgrade, Serbia and also as Docent for Intelligent Systems teaching on Faculty for Computer Science. She is familiar in teaching Intelligent Systems (Artificial Intelligence), Genetic Algorithms, Neural networks, Fuzzy Logic, Bioinformatics, Knowledge based Systems on PhD, Master and Primary studies. She is a graduate of Faculty for Electrical Engineering on University in Belgrade, Serbia. PhD research was application of multifractal analysis of digital medical images in cancer diagnostic and for it she has been several times awarded: The First Award on International Medical Fair 2014, The Special Award on International Technical Fair 2014 and many others (also many interviews for TV and newspaper). She is also an author and co-author of many journal and conference peer-reviewed.

