Brain tumor classification using logistic regression and linear support vector classifier

8 authors, including:

Lukasz Fulawka
Dolnośląskie Centrum Onkologii
9 PUBLICATIONS   17 CITATIONS

Piotr Krajewski
Dolnoslaskie Centrum Analizy Danych
2 PUBLICATIONS   0 CITATIONS

Witold Dyrka
Wroclaw University of Science and Technol…
16 PUBLICATIONS   64 CITATIONS

Brain tumor classification using logistic regression and linear support vector classifier

Grzegorz Żurek[*1,3], Michał Błach[2], Piotr Giedziun[1], Jakub Czakon[1],
Łukasz Fuławka[4], Łukasz Hałoń[4], Piotr Krajewski[†1], and
Witold Dyrka[‡1,2]

[1]Stermedia Sp. z o. o. Wroclaw, Poland
[2]Department of Biomedical Engineering, Wroclaw University of
Technology, Poland
[3]Faculty of Mathematics, Wroclaw University of Technology, Poland
[4]Department of Pathology, Lower Silesian Oncology Centre, Wroclaw,
Poland

September 28, 2015

1 Introduction

Image analysis for computer-aided medical diagnosis is a rapidly growing discipline
of scientific research. Many medical tasks, like segmenting MRI images of the brain
or counting blood cells, can be automated, which not only is economically beneficial
but above all permits doctors to devote more time to patients. In the era of cloud
computing and deep learning more and more problems that were previously regarded
as too complex for computers to handle are being solved, and tools that implement
those solutions are being delivered to the society. However, there are a variety of
challenges in contemporary medicine that have not been properly worked out yet. In
this paper we focus on one of these challenges. Namely, the objective of our work is to
present an algorithm that would perform binary classification to one of the two brain
cancer types: Oligodendroglioma and Astrocytoma, based on histological images of
the tumor tissue.

2 Methodology

2.1 Preprocessing and fragmentation

Available input data consisted of histological and MRI images of 32 brain cancer
patients. In this piece of work, we chose to focus on the histological data only.
Each histological image was a set of slices collected at different magnification, which

---

[*]grzegorz.zurek@stermedia.pl
[†]p.krajewski@stermedia.pl
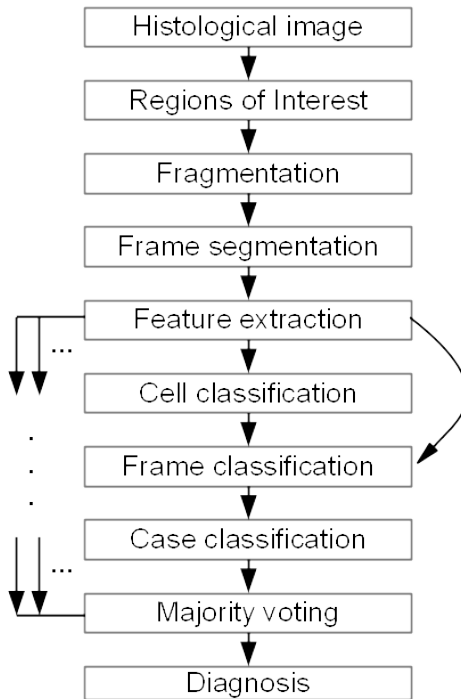[‡]witold.dyrka@pwr.edu.pl

Figure 1: Overall scheme of the classification algorithm

provided considerably more data to handle than in a typical image processing task. However, in order to assign a histological image to a specific cancer type, it is sufficient to find only a few relatively small frames evidencing pathological cancerous behavior. Consequently, the first step of the algorithm consisted on selecting Regions of Interest (RoI) for further analysis. More specifically, it is known that cancer types considered in this challenge result in overpopulating affected regions with astrocytes or oligodendrocytes [4]. Therefore, RoIs were extracted by segmenting low resolution histological images according to the density of cells. Only regions where the density was higher than a certain threshold were classified as RoIs and processed in next stages of the algorithm.

Having extracted the RoIs, we divided them into smaller equal-sized frames of about 1000×1000 pixels so that each frame could be processed separately. The frames were transformed using the Reinhard's normalization [1] and the Hematoxylin channel was extracted [2]. Next, the k-means clustering based on image intensity was performed in order to segment cells from background. Since it is assumed that each patient is labeled with one and only one cancer type, it is enough to classify relatively small number of frames in order to diagnose the patient. Therefore, for further analysis, we chose sets of 50 frames with the highest number of cells from each patient. Sample results of the RoI extraction and fragmentation are shown in Figure 2.
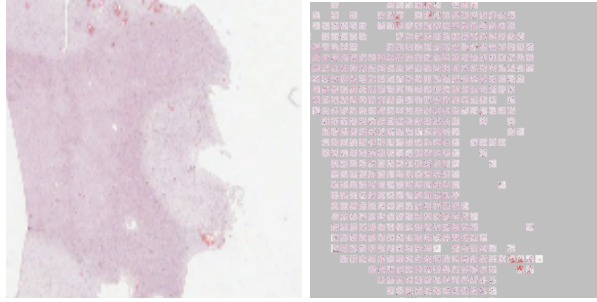
Figure 2: Example of extraction of Regions of Interest

2.2 Feature extraction and cell classification

In the next step each disjoint cluster of pixels representing a cell was classified either as oligodendrocyte (O) or not oligodendrocyte (A) based on extracted features including morphology and shape parameters, a measure of density of cells (see [3]), and a set of features describing the halo effect. The cell classifiers were trained in a supervised learning manner. We first built the training set by choosing a subset of frames representative to a given cancer type because of the highest number of cells or because they were classified as such by domain experts. Then, each cell in the representative frames was labeled according to the frame label. The training input consisted of feature vectors describing all cells in the representative frames and their labels. We used a random sample from the training input to fit meta-parameters for machine learning of two classifiers: Logistic Regression (LR) and Linear Support Vector Classifier (SVC), both implemented in Python's scikit-learn package [5]. The trained classifier returned probability that a cell was an oligodendrocyte (see Figure 3).
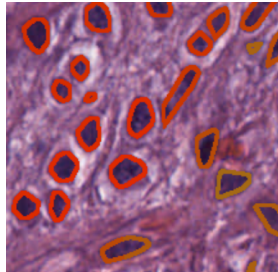

Figure 3: Visualization of cells classification where degree of affiliation is given by color.

2.3 Frame and case classification

Subsequently, we assigned each frame several characteristics such as quantiles of the oligodendrocyte probability distribution (including the maximum), density, and propensity to large structure formation. Based on these features, the classification of each frame was performed using Gaussian Naive Bayes (GNB) or LR classifiers from scikit-learn package [5]. After that, we combined frame scores using either the median or the arcsin formula: $Y = \frac{1}{\pi n} \sum_{i}^{n} \arcsin(2y_i - 1)$, where y is a prediction returned for each frame. Oligodendroglioma was diagnosed if the combined score exceeded a certain threshold.

3

3 Results

3.1 Learning the best parametrization

In order to select the best parametrization of the model, we designed a cross-validation procedure in which we randomly divided the training set in three parts made of 12, 10 and 10 cases. The first part was used to train the cell type classifiers, while the second part was utilized to train the case classifiers. The last part was used for estimating accuracy of fitted models. Each setup was run 20 times. We found that the best solutions were obtained by combining LR for the cell classification with GNB for the frame classification, or by combining SVC for the cell classification and LR for the frame classification; in 6 LR+GNB and in 4 SVC+LR cases the average accuracy reached 73%.

3.2 Obtaining final diagnoses

In order to obtain our ultimate classifiers, we fitted the 10 best models selected in the cross-validation. Cell classifiers were trained on cells from 10 most populous frames from all 32 training cases (ALL), or from a subset of these frames verified by domain experts (EXP). Then frame classifiers were learned on cell classifier outputs and other features calculated for 50 most populous frames from all training cases.

Eventually, fitted models were used to diagnose 20 test cases. Diagnoses obtained within each set of models (LR+GNB and SVC+LR) and training inputs (ALL and EXP) were combined by means of majority voting. Interestingly, while combined diagnoses given by models trained on ALL and EXP differed, combined diagnoses given by LR+GNB and SVC+LR models were identical.

References

[1] Y.Y. Wang, S.C. Chang, L.W. Wu, S.T. Tsai and Y.N. Sun. A color-based approach for automated segmentation in tumor tissue classification. In: Proc. Conf. of the IEEE Engineering in Medicine and Biology Society. pp.6577–6580, 2007.

[2] D. Magee, D. Treanor, D. Crellin, M. Shires, K. Smith, K. Mohee and P. Quirke. Colour Normalisation in Digital Histopathology Images. In: Proc. Optical Tissue Image analysis in Microscopy, Histopathology and Endoscopy (MICCAI Workshop), pp.100-111, 2009.

[3] K.K. Koeller and E.J. Rushing. Oligendroglioma and Its Variants: Radiologic-Pathologic Correlation. RadioGraphics 25:1669-88, 2005.

[4] R. Kordek (red.), *Onkologia. Podrecznik dla studentów i lekarzy.*, VIA MEDICA, Gdańsk, 2007.

[5] F. Pedregosa, G. Varoquaux, A. Gramfort et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research 12:2825-2830, 2011.