Neuroinformatics manuscript No.
(will be inserted by the editor)

Automated neuron detection in high-content fluorescence microscopy images using machine learning

Gadea Mata* · **Miroslav Radojević*** · **Carlos Fernandez-Lozano*** · **Ihor Smal** · **Miguel Morales** · **Erik Meijering** · **Julio Rubio**

Abstract The study of neuronal morphology in relation to function, and the development of effective medicines to positively impact this relationship in patients suffering from neurodegenerative diseases, increasingly involves image-based screening of large numbers of cultured neurons. The first critical step toward fully automated high-content image analyses in such studies is to detect all neuronal cells and distinguish them from possible non-neuronal cells or artifacts in the images. Here we investigate the performance of state-of-the-art machine learning techniques for this purpose. These include support vector machines, random forests, elastic nets, and k-nearest neighbor classifiers, operating on an extensive set of image features extracted using the compound hierarchy of algorithms representing morphology and the scale-invariant feature transform. We present experiments carried out on a dataset of rat hippocampal neurons from our own studies in order to find out the most suitable classifier(s) and subset(s) of features in the common practical setting where there is very limited annotated data for training. The results indicate that a support vector machine using the right feature subset ranks best for the considered task, although its performance is not statistically significantly better than some random-forest and elastic-net based classifiers.

Keywords Neuron detection · High-content analysis · Fluorescence microscopy · Machine learning

G. Mata · J. Rubio
Department of Mathematics and Computer Science
University of La Rioja, Logroño, Spain
E-mail: gadea.mata@unirioja.es

M. Radojević · I. Smal · E. Meijering
Biomedical Imaging Group Rotterdam
Departments of Medical Informatics and Radiology
Erasmus University Medical Center, Rotterdam, Netherlands

C. Fernandez-Lozano
Department of Information and Communications Technologies
Faculty of Computer Science
University of A Coruña, A Coruña, Spain

M. Morales
Institute of Neurosciences
Department of Biochemistry and Molecular Biology
Autonomous University of Barcelona, Barcelona, Spain

*These authors contributed equally to this work.

1 Introduction

Neurons are special cells in the sense that they codify and transmit information in the form of action potentials. Networks consisting of many billions of neurons, such as in the brains of higher organisms, are extraordinarily complex and perform many different functions. Since the pioneering work of Ramón y Cajal (1899) it is well known that the morphology of neurons vary widely in different parts of the brain and that neuronal morphology and function are intricately linked. Moreover, in healthy conditions, neuronal (sub)networks within the brain are dynamic and continuously readjust their connections during the lifetime of an organism in response to external stimuli, in order to refine existing functions or learn new ones (Ascoli 2015). Conversely, in pathological conditions, disease processes destructively alter neuronal morphology and cause progressive loss of function, such as in Alzheimer's and Parkinson's disease, but also in aging (van Pelt et al. 2001). Thus the study of neuronal cell morphology in relation to function, in health and disease, is of high importance for developing suitable drugs and therapies (Meijering 2010).

A convenient tool to visualize large numbers of cultured cells for phenotypic profiling and analysis in drug discovery is high-content fluorescence microscopy imaging (Xia and Wong 2012; Antony et al. 2013; Singh et al. 2014; Bougen-Zhukov et al. 2017). By automated acquisition it produces very large amounts of image data, which cannot be analyzed manually but require automated high-content analysis

(HCA) in order to take full advantage of all captured information. HCA is also used increasingly in neuroscience research (Dragunow 2008; Anderl et al. 2009; Jain et al. 2012) and various image processing pipelines have been developed for quantitative analysis of neuronal cells in high-content images (Vallotton et al. 2007; Zhang et al. 2007; Wu et al. 2010; Dehmelt et al. 2011; Radio 2012; Charoenkwan et al. 2013; Smafield et al. 2015). However, especially in screening applications, where the image quality is often relatively low and may vary widely between experiments, the challenge remains to develop more accurate and more robust image analysis methods (Sommer and Gerlich 2013; Kraus and Frey 2016; Meijering et al. 2016).

The first critical step in any HCA pipeline is the detection of the objects of interest in the images. It is well recognized now in many areas of microscopic image analysis that machine learning based classification methods are an excellent choice for this task and typically outperform non-learning methods based on manually defined rules (Horvath et al. 2011; Sommer and Gerlich 2013; Kraus and Frey 2016; Arganda-Carreras et al. 2017). However, which classifiers work best, and on which sets of image features, may depend on the specific image data and detection task, and needs to be determined experimentally before using HCA on a routine basis in a given application.

In this paper we investigate the performance of machine learning methods for the specific task of detecting neuronal cells in high-content fluorescence microscopy images as a first step toward fully automated HCA in our neuroscientific studies. We recently presented an early version of our work at a conference (Mata et al. 2016) and report here on a significant extension of that work including more classifiers, more extensive experiments and results, and a much deeper and more solid (statistical) analysis and discussion of the findings. We consider various state-of-the-art classifiers based on support vector machines (SVM), elastic nets (in particular GLMNET), random forests (RF), and k-nearest neighbor (KNN) classifiers, operating on more than a thousand image features extracted using the compound hierarchy of algorithms representing morphology (CHARM) and the scale-invariant feature transform (SIFT). Deep-learning based methods are not included in this study because of their excessive need for annotated data. Instead we explore classifiers based on precalculated image features in order to determine which combinations of classifiers and features work best in the common practical setting where there is very limited annotated data for training.

2 Materials and Methods

To facilitate reproducibility of our study we made use of published image data and employed publicly available software tools. Here we successively describe the image dataset, the used methods for extracting image features, and the considered machine learning methods.

2.1 Image Dataset

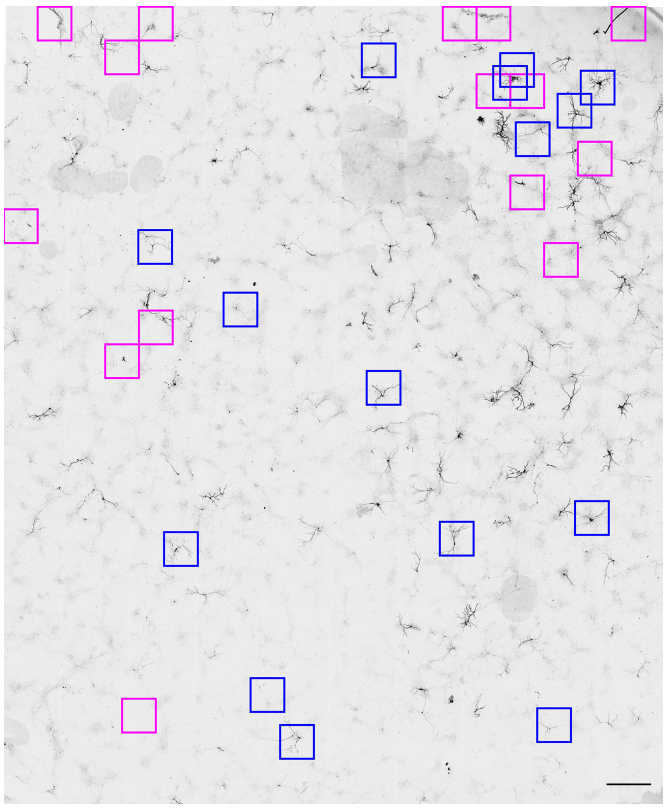
The high-content image data used in this study is from our ongoing research into effective treatments for neurological disorders (Cuesto et al. 2011; Enriquez-Barreto et al. 2014; Enriquez-Barreto and Morales 2016). We describe the acquisition of the images, their annotation, and the strategy we used to obtain a well-balanced dataset for training of the machine learning algorithms.

2.1.1 Image Acquisition

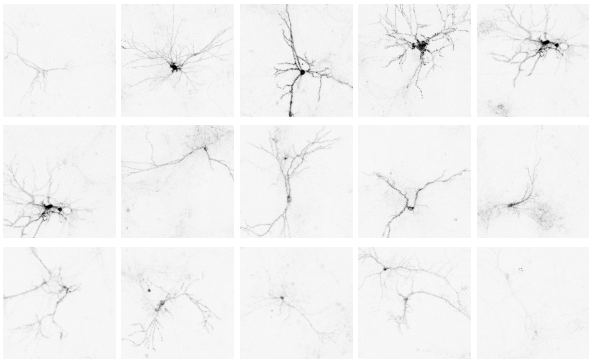
Rat hippocampal neurons were cultured and transfected with green fluorescent protein (GFP) and imaged with a Leica SP5 automated confocal fluorescence microscope using its Matrix modules and a 20× lens. The imaged neurons, coming from a part of the brain (the hippocampus) that is well known to be involved in higher functions such as learning and memory (Squire 1992), typically have a pyramidal soma with a complex dendritic tree (Goslin et al. 1998), and their in-vivo morphological features are well conserved in culture conditions. We acquired eight two-dimensional (2D) high-content images (total size >1 GB), each with a size of about $10{,}000 \times 12{,}000$ pixels, covering approximately $70\,\mathrm{mm}^2$ of culture dish, and containing on the order of 40 transfected neurons (Fig. 1). Our specimens usually have about 100 neurons, but more than half of them are not or only partly imaged, as they are in different optical planes or close to the borders of the dish, making the automated detection of relevant image structures (complete neurons) as opposed to irrelevant image structures (incomplete neurons, astrocytes, and artifacts) quite challenging.

2.1.2 Image Annotation

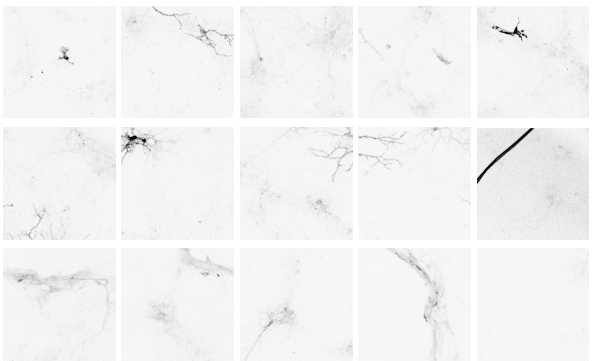
To obtain a reference dataset for training and testing of the machine learning methods, an expert neurobiologist manually marked all the regions of interest (ROIs) containing neurons in these images, about 400 in total. We established that relevant neurons typically cover an area of around 500×500 pixels in our images and therefore we fixed the ROI size to these dimensions. Using the same window size, we automatically sampled additional patches from the remaining parts of the images, containing all different types of irrelevant image structures. More specifically, to ensure evenly distributed sampling of background patches across the images, we defined a regular grid and included every patch from the grid having less than 50% overlap with any of the neuron ROIs marked by the expert, resulting in approximately 4,500 non-neuron patches. In the sequel we refer to



(a) Example high-content image. Scale bar: 500μm.



(b) Example patches considered as positives (blue squares).



(c) Example patches considered as negatives (magenta squares).

Fig. 1 Part of a high-content fluorescence microscopy image (a) and example patches containing neuronal structure (b) and background (c). The intensities of the shown images are inverted compared to their originals for better visualization.

the neuron ROIs as 'positives' and the non-neuron image patches as 'negatives' (Fig. 1).

2.1.3 Dataset Balancing

Due to the sparseness of our image data, the patches of the negative class far outnumbered those of the positive class, with a ratio of approximately 10:1, resulting in an imbalanced dataset. It is well known that the performance of classification algorithms may be negatively impacted by the data being imbalanced (Chawla et al. 2004; Daskalaki et al. 2006; Forman and Scholz 2010; Branco et al. 2016), as the algorithms may overfit the majority class and underfit the minority class, and favor the former, yielding biased results (García et al. 2014; Li et al. 2018). Approaches to deal with class imbalance can roughly be divided into two categories (He and Garcia 2009; Krawczyk 2016; Haixiang et al. 2017): data-level approaches, which modify the collection of data samples to balance the class distributions, and algorithm-level approaches, which modify the learning algorithms to alleviate their bias, for example by introducing costs to balance the importance of the different classes. Since in our case the class imbalance was substantial, and we used mostly existing algorithms and aimed to evaluate their performance without tweaking them for our application, we opted to oversample the minority class (Chawla et al. 2002) in order to obtain approximately the same number of samples in each class. Specifically, for each neuron ROI marked by the expert, we also considered as potential positive samples all patches having at least 50% overlap with that ROI (Fig. 2). However, the higher the overlap percentage of a patch, the higher the relevance of that patch, as it contains more neuron structure. Therefore, we assigned a weight to each potential patch corresponding to the overlap percentage, and taking this into account we randomly sampled from the pool of all potential patches in order to avoid bias (Fig. 3). This resulted in a positive class and a negative class each consisting of approximately 4,500 samples.

2.2 Images Features

To train the machine learning algorithms we used a large number of predefined features extracted from the positive and negative image patches. In this study two very comprehensive feature extraction approaches were employed: the compound hierarchy of algorithms representing morphology

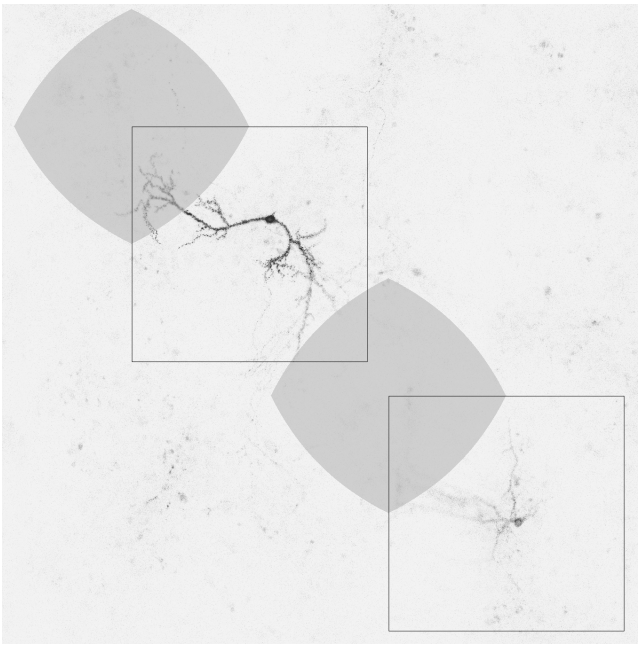


Fig. 2 Two example neurons with their expert-marked ROIs (black squares) and their potential alternative positive patch locations (gray regions). The latter comprise all possible top-left corner positions of patches with the same size as the given ROI and having 50% or more area overlap with that ROI.

(CHARM) and the scale-invariant feature transform (SIFT). Here we briefly describe each of them.

2.2.1 CHARM Features

For the extraction of the CHARM features we used the open-source software library WND-CHARM (Shamir et al. 2008; Orlov et al. 2008), which has been successful for many pattern recognition applications in biology (Shamir et al. 2010; Uhlmann et al. 2016) as well as in astronomy (Shamir 2012; Kuminski et al. 2014) and in art (Shamir and Tarakhovsky 2012). It can extract a large number of generic image descriptors and also includes a classifier based on the weighted neighbor distance (WND) between feature vectors. However, since the performance of this classifier was rather limited in our initial results (Mata et al. 2016), we decided to explore alternative machine learning algorithms for our classification task, but using the image features calculated by this sofware library. In total we calculated 1,059 CHARM features for each positive and negative patch (recent versions of WND-CHARM can extract even more features but at an increased computational cost).

The calculated image features can be divided into four categories: polynomial decompositions, high-contrast features, pixel statistics, and texture descriptors. The first category includes features based on the Zernike polynomials and Chebyshev polynomials (Gradshteyn and Ryzhik 1994) as well as Chebyshev-Fourier statistics. Features from the second category include various statistics calculated from the Prewitt edges (Prewitt 1970), Gabor wavelets (Gabor 1946), and object masks obtained by Otsu thresholding (Otsu 1979). The third category consists of image features calculated from the multiscale intensity histogram (Hadjidementriou et al. 2001) and various statistics based on the image moments. The last category includes the Haralick (Haralick et al. 1973) and Tamura (Tamura et al. 1978) texture features. In addition, the software calculates various image transforms, including the Radon, Fourier, wavelet, Chebyshev, and edge transforms, as well as transforms of image transforms. For more technical descriptions of all features and transforms we refer to Orlov et al. (2008).

2.2.2 SIFT Features

The SIFT algorithm (Lowe 2004) is another popular tool to extract meaningful features from images for pattern recognition tasks. It has been used for a very wide range of applications in thousands of studies, including in biomedical image analysis (Ni et al. 2009; Jiang et al. 2010; Mualla et al. 2013; Zhang et al. 2013; Lee et al. 2016; Yu et al. 2016). The extraction of SIFT features from a patch consists of four main steps. First, a Gaussian scale space is calculated, and potentially interesting points are identified by searching over all scales and locations for extrema in the difference-of-Gaussian function. Next, key points are selected from this list of candidates based on their measures of stability, and their precise location and scale are determined by model fitting. Then, based on the local gradient directions, each key point is assigned to one or more orientations (binned angles). And lastly, orientation histograms are constructed from the local gradients in a region around each key point, relative to the key point's assigned orientation, and the histogram entries constitute the elements of a (typically 128-dimensional) feature vector. By normalizing the feature vector we obtain a key point descriptor that is relatively invariant to spatial distortions and changes in illumination. All key point descriptors of a patch taken together form the SIFT features of that patch.

A problem in comparing image patches based on their SIFT features is that the number of key points, and thus the number of descriptors, may be different for each patch. The comparison is facilitated by applying a transform that represents each patch by a feature vector of fixed length. A very effective and popular approach to achieve this is to use the bag-of-words (BoW) model (Sivic 2009). Here, all descriptors of all available patches are divided into a fixed number of clusters by k-means clustering (MacQueen 1967), and the mean of each cluster represents a visual 'word', a vector of the same dimensionality as the descriptors. Subsequently, for any given patch, each of its descriptors is assigned to the cluster to which it is closest according to the Mahalanobis

distance. This yields a vector of fixed length k, with each vector element being the number of patch descriptors assigned to the corresponding cluster.

To obtain the SIFT-BoW feature vector for each positive and negative patch, we used the VLFeat software library (Vedaldi and Fulkerson 2008) in conjunction with MATLAB (MathWorks 2016). The vector length is a user parameter, and we evaluated the classification performance of the different machine learning algorithms for lengths of 20, 40, 60, 80, 100, 150, 200, and 230.

2.3 Machine Learning

Four different machine learning algorithms were considered for the classification task in this study. We summarize the algorithms and their hyperparameters, and explain the resampling strategies we used in the training and testing of the algorithms, and the feature selection approach.

2.3.1 Classification Algorithms

Support Vector Machines (SVM) are one of the best known and most successful machine learning algorithms for both classification and regression problems (Boser et al. 1992; Vapnik 1998, 1999; Bishop 2006). In classification problems, the principal aim of SVM is to find the hyperplane in the feature space that best separates the given samples (in our case neuron and non-neuron patches), by maximizing the distance between the samples and the hyperplane (Burges 1998). If the problem requires more complex (non-linear) separation functions, SVM can still be used, by employing so-called kernel functions that transform the high-dimensional feature space such that a hyperplane (linear) can still be used as the separation function. Generally speaking one could interpret a kernel as a similarity measure (Vert et al. 2004). Different types of kernels have been proposed, the Gaussian radial basis function (RBF) being one of the most popular (Cristianini and Shawe-Taylor 2000). Two hyperparameters need to be optimized for best performance, one related to the SVM algorithm itself, the other related to the Gaussian RBF kernel. The first ('cost') is the trade-off between the misclassification of the samples and the simplicity of the decision surface. The second ('gamma') is the free parameter of the Gaussian function. In the grid search in our experiments we set the range of possible values for both parameters to $[2^{12}, 2^{-12}]$.

Random Forest (RF) is another well-known and successful machine learning algorithm (Breiman 2001) for classification and regression. As a classifier it operates by randomly taking multiple bootstrapped subsets of the data, fitting a decision tree to each one of them, and outputting the mode of the class outputs of the individual trees. This approach reduces the possibility of overfitting the training dataset and generally produces more accurate results than a single decision tree. The RF has two main hyperparameters. The first ('node size') is the minimum size of the terminal nodes of the decision trees. In our experiments we considered integer values of 1…5 for this parameter. The second ('mtry') is the number of features randomly sampled as possible candidates at each split. For this parameter we considered integer values of 5…36.

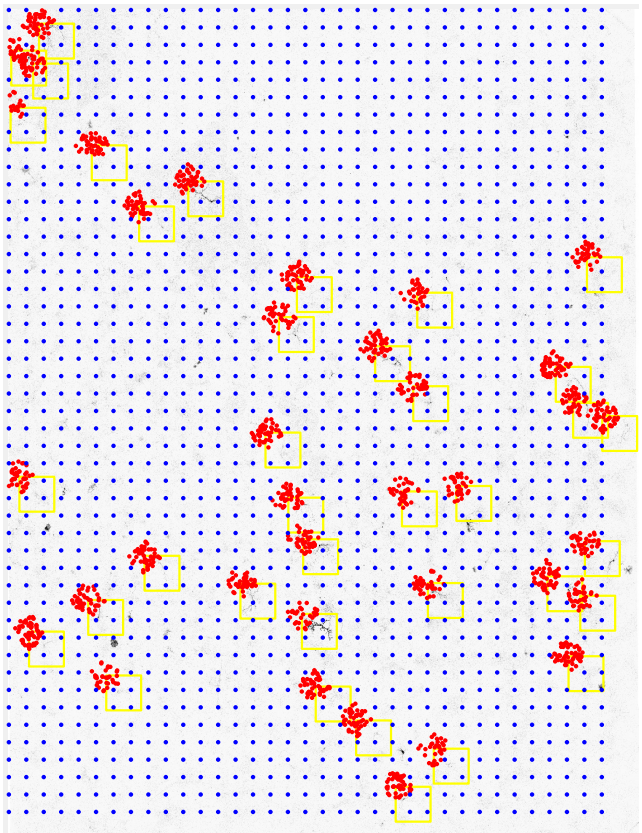


Fig. 3 Example of positive patch oversampling. The background shows a high-content fluorescence microscopy image (with intensities inverted), and the graphical overlay shows the neuron ROIs marked by the expert (yellow squares), the top-left corners of the patches randomly sampled from all possible patches considered as alternative positives (red dots), and the intersection points (blue dots) of the regular grid used for negative patch sampling.

k Nearest Neighbor (KNN) classification operates by comparing an unclassified patch to patches with known class labels (the reference set), then selecting the k most similar of these patches (the nearest neighbors) according to some distance metric in the feature space, and outputting the most frequently occurring class label of these patches (Cover and Hart 1967). In this study we used a weighted KNN algorithm (Hechenbichler and Schliep 2004; Samworth 2012) which employs the Minkowski distance and classifies patches using the maximum of summed kernel densities. This algorithm uses kernel functions to weigh the neighbors accord-

ing to their distances. The KNN algorithm requires optimization of only one hyperparameter ('k'), for which we considered integer values of 3...9.

Elastic Net (GLMNET) is a regularization method (Zou and Hastie 2005) based on the least absolute shrinkage and selection operator (LASSO) (Tibshirani 1996). Similar to the LASSO, this method simultaneously performs automatic feature selection and continuous shrinkage (regularization), and is able to select groups of correlated features. Specifically, we used GLMNET (Friedman et al. 2010), which fits a generalized linear model using a penalized maximum likelihood approach, and combines l_1 and l_2 penalties for regularization. GLMNET has two hyperparameters. The first ('alpha') is in the range $[0, 1]$ and linearly weighs the contributions of the different types of penalities, with value 0 corresponding to l_2 regularization, and 1 to l_1 regularization. In our experiments we used values 0, 0.15, 0.25, 0.35, 0.5, 0.65, 0.75, 0.85, and 1. The second parameter ('lambda') determines the degree of regularization, for which we considered values of 0.0001, 0.001, 0.01, 0.1, and 1.

For our experiments we used the statistical computing software tool R (R Core Team 2016) and the R packages mlr (Bischl et al. 2016), e1071 (Meyer et al. 2017), randomForest (Liaw and Wiener 2002), kknn (Schliep and Hechenbichler 2016), and GLMnet (Friedman et al. 2010), to evaluate all the machine learning algorithms.

2.3.2 Resampling Strategies

The mentioned hyperparameters of the machine learning algorithms need to be optimized for best performance. To accomplish this, and at the same time make an honest comparison of the algorithms under equal conditions, we used a nested resampling approach (Simon 2007; Bischl et al. 2012) involving an inner loop and an outer loop. In this approach, the actual performance assessment of the algorithms takes place in the outer loop, which we implemented as three independent runs of a 10-fold cross-validation experiment, with stratification (to ensure having the same proportion of positive and negative samples in all partitions of the cross-validation), where the final performance scores are obtained by averaging. In each iteration of the outer loop, the corresponding training set is used in an inner loop, to find the optimal values of the hyperparameters of the algorithms. The inner loop was implemented using a holdout approach, where the given training set from the outer loop is redivided into a training subset (2/3rd of the set) and a validation subset (1/3rd of the set), and a grid search is run on the hyperparameters. The hyperparameter values that give the best performance are subsequently used to retrain the algorithms on the given training set from the outer loop. This nested resampling strategy is statistically sound but computationally expensive. To make the experiments computationally feasible, we discretized the search space using the hyperparameter values listed in the previous section.

Feature Set	CHARM Features	SIFT Features							
		20	40	60	80	100	150	200	230
01	✓								
02		✓							
03			✓						
04				✓					
05					✓				
06						✓			
07							✓		
08								✓	
09									✓
10	✓	✓							
11	✓		✓						
12	✓			✓					
13	✓				✓				
14	✓					✓			
15	✓						✓		
16	✓							✓	
17	✓								✓

Table 1 Feature sets used in the experiments. Each of the 17 sets consists of either CHARM features, or SIFT features with a given BoW vector length, or a combination of both, as indicated.

2.3.3 Feature Selection

Although a priori it is appropriate to consider as many features as possible, and increasing computational power allows us to construct larger and larger feature sets, in the end many features may be irrelevant or may even negatively impact the performance of the machine learning algorithms. Thus we also aimed to investigate which of all considered features positively contribute most to the performance of the algorithms in our application. Knowledge of the best features allows one to build potentially better and computationally more efficient classifiers. Moreover, it may shed light on which image information is most relevant to the classification task, which in turn may provide useful hints to improve the imaging process. There exist various approaches for feature selection using machine learning algorithms in supervised classification problems, including filter, wrapper, and embedded approaches (Saeys et al. 2007). In this study we used the filter approach, as it is independent of the classifier, fast, scalable, and needs to be applied only once, after which the different algorithms can be evaluated.

3 Experimental Results

All experiments in this study were carried out using the BioCAI HPC cluster facility at the University of A Coruña. To quantitatively assess and compare the performances of the machine learning algorithms we used the area under the receiver operating curve (AUROC) measure as it captures both Type I and Type II errors (Fawcett 2006). We first performed an initial exploratory experiment on various combinations of CHARM and SIFT feature sets to find out which of these deserved closer investigation. Using the most promising feature sets we conducted an in-depth performance evaluation of all the algorithms. Subsequently we investigated which specific features of the complete set contributed most to the performance. And finally we performed an analysis to see whether the differences in performance of the algorithms were statistically significant or not.

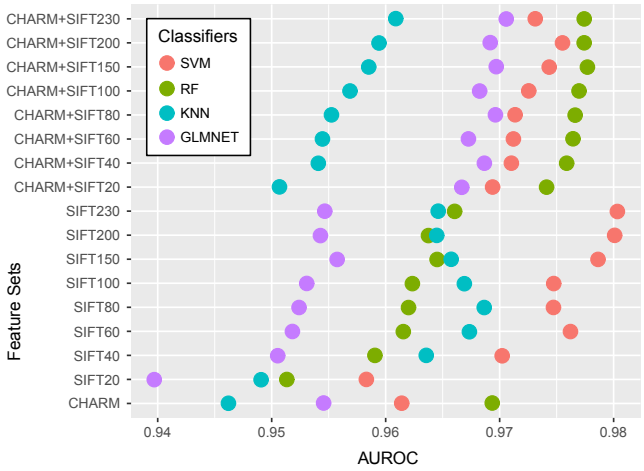


Fig. 4 Results of the initial exploratory experiment. Each of the considered classifiers (SVM, RF, KNN, GLMNET) was evaluated for each of the 17 feature sets (Table 1) according to the performance measure (AUROC) using a double holdout approach.

3.1 Initial Exploratory Results

For the initial experiment we constructed 17 different feature sets (Table 1) from (combinations of) the CHARM features and the SIFT features for different BoW vector lengths. To avoid prohibitive computation times in the cross-validation experiment (described next), we first explored which of these feature sets would likely yield the best classification results with the considered machine learning algorithms. The feature sets were preprocessed by normalizing each feature to zero mean and unit standard deviation over all patches, and removing constant features (if present), to reduce the effect of possible outliers. To make this exploratory experiment computationally feasible, we did not use the full cross-validation approach as described, but a simpler holdout approach. Specifically, 1/10th of the data was used for testing, and the remaining 9/10th for training. The optimal hyperparameters of the classification algorithms were obtained using a grid search on 2/3rd of the training set and validated on the remaining 1/3rd. Using the found hyperparameter values the algorithms were trained on the full training set.

From the results (Fig. 4) we observe that both the absolute and the relative performance of the classifiers was quite different for the different feature sets. Specifically, for SVM and KNN, the best results were obtained with the SIFT features alone (for sufficiently large BoW vector lengths), while the CHARM features alone produced inferior results, and with the combination of CHARM and SIFT features these classifiers performed somewhere in between. For RF and GLMNET, on the other hand, the SIFT features alone yielded inferior results, and with the CHARM features alone these classifiers did not fare much better, but the combination of CHARM and SIFT features (for all BoW vector lengths) produced the best results.

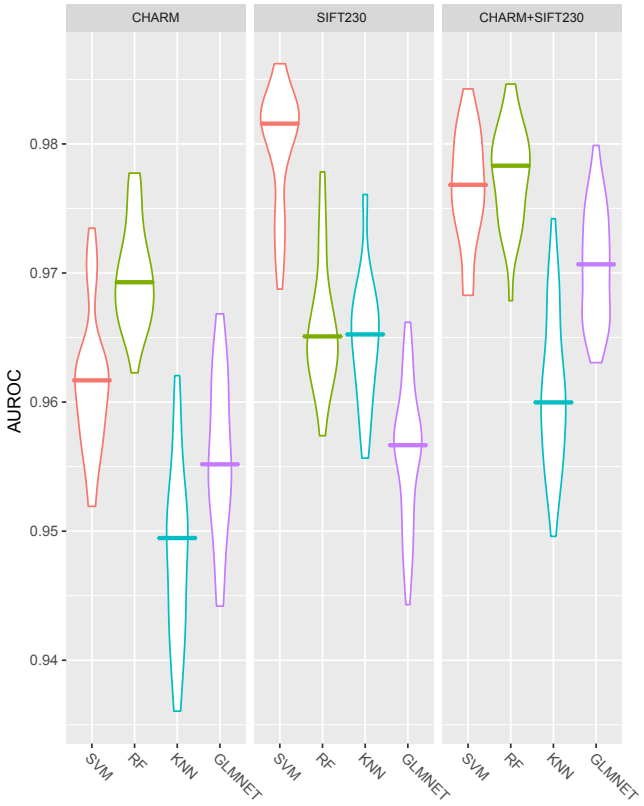


Fig. 5 Results of the cross-validation experiment. Each of the considered classifiers (SVM, RF, KNN, GLMNET) was evaluated for each of the selected feature sets (CHARM, SIFT230, CHARM+SIFT230) using the performance measure (AUROC). The results are shown as violin plots, where the horizontal bar indicates the median value, the vertical extent is the interquartile range, and the width indicates the estimated probability density.

Thus we concluded that the cross-validation experiment should include both the CHARM and SIFT feature sets alone, as well as their combination, and the only way to reduce the

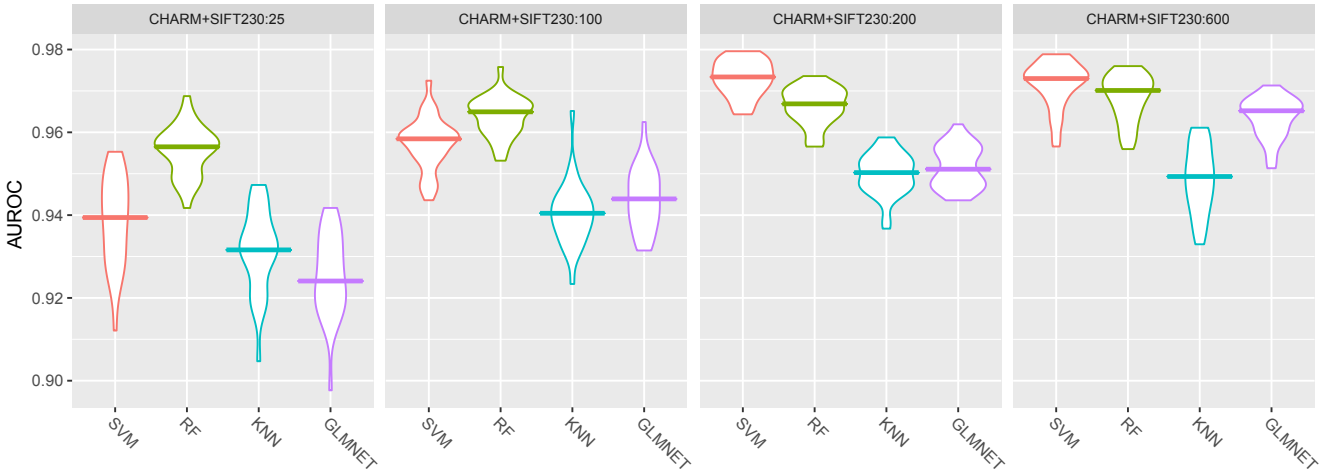


Fig. 6 Performance (AUROC) of the considered classifiers (SVM, RF, KNN, GLMNET) for different feature subsets (the top 25, 100, 200, and 600 features from the CHARM+SIFT230 set). The results are shown as violin plots, where the horizontal bar indicates the median value, the vertical extent is the interquartile range, and the width indicates the estimated probability density.

computational cost of that experiment was to select a specific SIFT-BoW vector length. Overall, the results seemed to indicate that in most cases it is better to use larger vector lengths, and simply taking the maximum considered length (230) is a good choice.

3.2 Cross-Validation Results

Based on the results of the initial exploratory experiment we selected feature sets #01 (CHARM features only), #09 (SIFT230 features only), and #17 (CHARM+SIFT230 features) to evaluate the four machine learning classifiers using a cross-validation experiment, involving an outer loop (3×10-fold) for performance assessment and an inner loop (holdout) for hyperparameter optimization as described. The results (Fig. 5) show that virtually all classifiers achieved AUROC values of >95% and, generally, SVM and RF outperformed KNN and GLMNET. Considering the different feature sets, we observe that all classifiers except RF achieved better performance with the SIFT230 feature set than with the CHARM feature set. This is interesting since the latter is much more extensive (1,059 features of many different types) than the former (230 BoW clusters). Apparently the SIFT230 features are more descriptive of the image content in our application. This is confirmed by the results with the CHARM+SIFT230 feature set, which are consistently better than with the CHARM feature set alone. However, whereas RF and GLMNET performed best using the more extensive CHARM+SIFT230 set, SVM and KNN performed best using the SIFT230 set alone. Overall, the best results were obtained with the SVM classifier using the SIFT230 feature set, although SVM and RF using the combined CHARM+SIFT230 features performed comparably (we discuss statistical significance in Section 3.4).

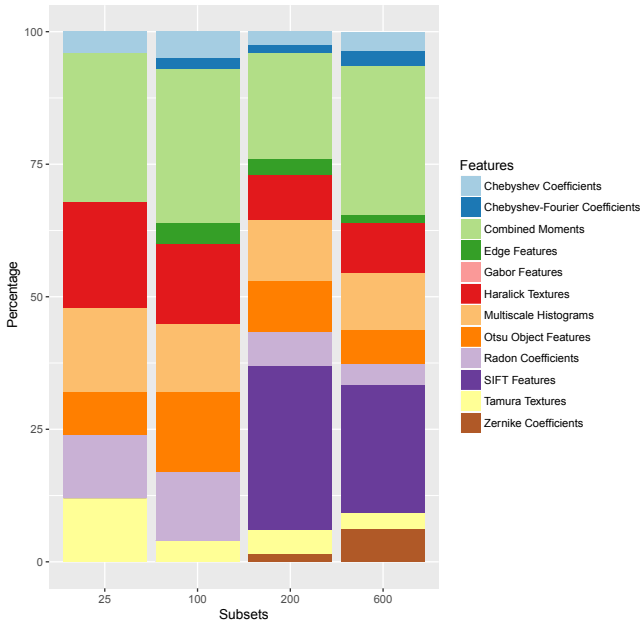


Fig. 7 Cumulative percentages of the different types of features contained in the four subsets (the top 25, 100, 200, and 600 features selected from the CHARM+SIFT230 set).

3.3 Feature Selection Results

Next we subjected the complete CHARM+SIFT230 feature set to a feature selection experiment. Specifically, we wanted to find out which features contributed most to the performance of the different classifiers, and whether these features alone could yield similar or even better classification performance than using the complete set, as that would make the classification task computationally cheaper.

To this end we ranked all 1,289 features using a Wilcoxon test (Wilcoxon 1945) and considered four subsets, consisting of the top 25, 100, 200, and 600 features. The results

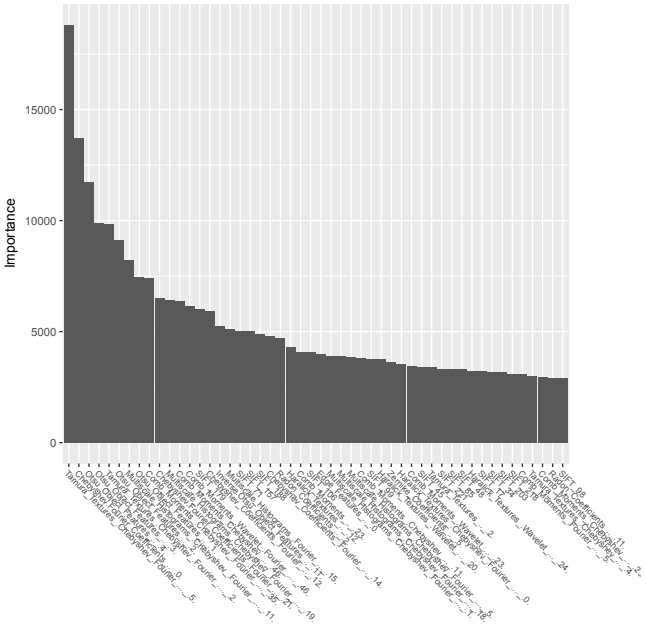


Fig. 8 The 50 most important features from the CHARM+SIFT 230:200 feature subset used by the SVM classifier. Importance was calculated according to the w^2 value of the SVM hyperplane optimally separating positive and negative neuron patches. The importance value for each feature was averaged over all runs and folds of the cross-validation experiment.

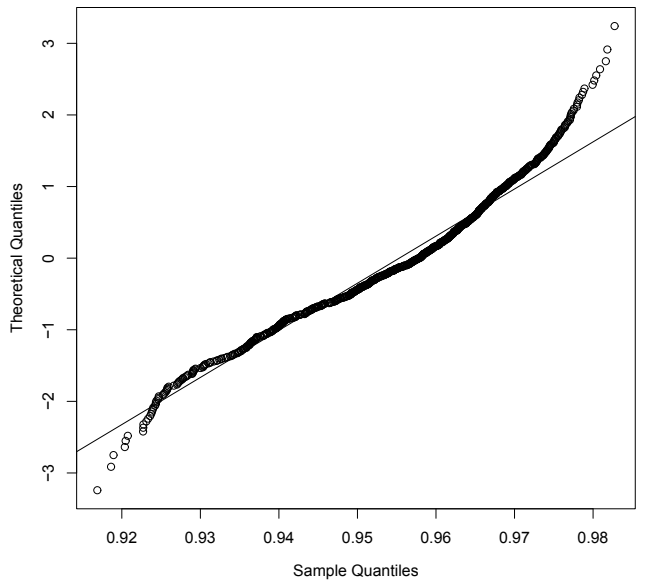


Fig. 9 Quantile-quantile (Q-Q) plot of the theoretical normal distribution and our data samples. Clearly, the computed values (small circles) deviate substantially from a straight line (the solid line is the least squares fit) and reveal a nonlinear relationship, leading to the conclusion that our data is not normally distributed.

(Fig. 6) agree with those of the previous experiment in that SVM and RF consistently outperformed KNN and GLMNET for all feature subsets. We also observe that the larger the number of top features, the better the performance of all four classifiers, but for most of them there was little improvement beyond the top 200 features. In fact, the scores of the best performing classifiers, SVM and RF, were very similar for the CHARM+SIFT230:200 subset and the full CHARM+SIFT230 set, and with smaller standard deviations (we discuss statistical significance in Section 3.4). This indicates that the non-selected features provided noise rather than useful information to the classifiers.

Analyzing the types of features contained in the four subsets (Fig. 7), we note that the top 25 and top 100 subsets do not contain any of the 230 SIFT features, whereas in the top 200 and top 600 subsets they play a prominent role. This suggests that individual or very small groups of SIFT features by themselves are not as effective as many of the CHARM features, but as soon as there is room for a sufficient number of them to work together, their importance in the classification process rises rapidly. According to the feature selection results (Fig. 6), the best performing classification model is the SVM using the CHARM+SIFT230:200 feature subset. Studying the importance of each of the features in this subset according to the w^2 value of the SVM hyperplane that best separates positive and negative neuron patches, we observe (Fig. 8) that the most important features are actually coming from the CHARM set (notably Tamura texture features, Chebyshev-Fourier coefficients, Otsu object features, moment based features, and multiscale histogram based features), but indeed a substantial portion of the features are from the SIFT set.

3.4 Statistical Analysis Results

Finally we analyzed the statistical significance of the performance results of the considered classification algorithms on the selected feature (sub)sets, to see if any particular model (combination of features and classifier with corresponding optimal hyperparameters) is to be preferred for our application. There exist mainly two types of statistical test to do this: parametric and non-parametric. Although parametric tests can be more powerful, they require normality, independence, and heteroscedasticity of the data (Fernandez-Lozano et al. 2016). To check the first condition, we used the Shapiro-Wilk test (Shapiro and Wilk 1965) with the null hypothesis that our data follows a normal distribution, and we rejected the null hypothesis with very significant values of $W = 0.96857$ and $p < 1.78 \cdot 10^{-12}$ (see also the Q-Q plot in Fig. 9). Since this already disqualifies parametric testing, there was no need to check the other conditions.

Thus we used a non-parametric test, the Friedman test (Friedman 1940), with the null hypothesis that all models yield the same performance on our data, and we rejected the null hypothesis with very significant values of $\chi^2 = 757.56$ and $p < 2.2 \cdot 10^{-16}$. Since this means that at least some models are statistically significantly better or worse than others,

we subsequently tested for significant differences between all pairs of models using the post-hoc Nemenyi test (Nemenyi 1963), with the control model being the SVM classifier using the SIFT230 feature set, as it performed best in the cross-validation experiment (Fig. 5).

The results (Fig. 10) show that several other models performed statistically similar to the control model. These include the SVM classifier using the CHARM+SIFT230 feature set or even just its top 200 or 600 features. Other statistically similar models include the RF classifier using the CHARM, SIFT230, or CHARM+SIFT230 feature set, or just the top 600 features of the latter. Of the models based on the KNN and GLMNET classifiers, only the GLMNET classifier using the full CHARM+SIFT230 feature set performed statistically similar.

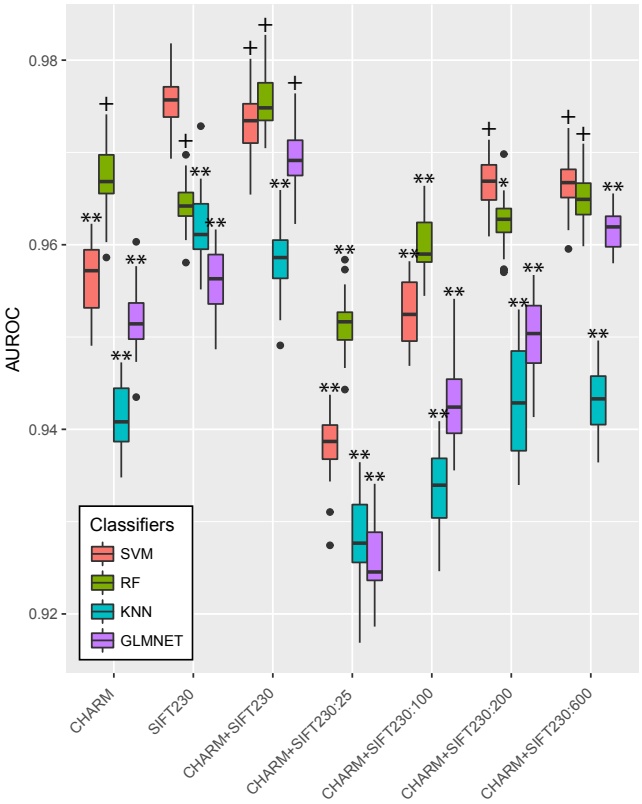


Fig. 10 Results of the Nemenyi test showing the statistical significance of the differences in performance of the considered models (classifiers SVM, RF, KNN, and GLMNET, using any of the selected feature (sub)sets CHARM, SIFT230, CHARM+SIFT230, and the top 25, 100, 200, and 600 features of the latter) with respect to the control model (SVM using SIFT230). Performance values (AUROC) of each model from all runs and folds of the cross-validation experiment are summarized using a standard R box plot. Significance with respect to the control model is indicated with various symbols, depending on whether $p > 0.05$ (+), or $0.01 < p < 0.05$ (*), or $p < 0.01$ (**).

4 Discussion and Conclusions

Our goal with the presented study was to find out which machine learning based classification algorithms and which commonly used feature extraction algorithms would be most suited for the task of detecting neurons in high-content fluorescence microscopy image data typically acquired in screening experiments. To this end, we considered four popular classifiers (SVM, RF, KNN, GLMNET) and two popular feature extraction tools (CHARM and SIFT), and performed various experiments and statistical analyses to narrow down and compare the many possible models (combinations of classifiers and (sub)sets of features).

From the results we conclude that of all considered classifiers, SVM and RF generally work best, provided they are fed with the right sets of features. We observed statistically similar performance with the following models: SVM using SIFT (230 features), CHARM+SIFT (the full 1,289 features or only the top 200 or 600), RF using CHARM (1,059 features), SIFT (230 features), or CHARM+ SIFT (the full 1,289 features or only the top 600), and GLMNET using the full CHARM+SIFT feature set. Since, from the perspective of computational efficiency, smaller sets of features are to be preferred, we conclude that the SVM model using the top 200 features of the CHARM+SIFT set is most suitable for the task, closely followed by the SVM or RF models using the 230 SIFT features.

In the spirit of Occam's razor principle (Iacca et al. 2012; Hong et al. 2013; Ebrahimpour et al. 2017), which considers the simplest explanation of natural phenomena to be the closest to the truth, we have sought the smallest possible classification model capable of determining with high accuracy whether or not a new unseen image patch contains neuron structures. Generally speaking, in order to achieve good generalization in a classification task, it is required to have a sufficient number of samples and to minimize model complexity (Gupta et al. 2017). Since currently our data is rather limited, we started out by considering state-of-the-art classification algorithms requiring explicit calculation of features, and using state-of-the-art algorithms for extracting a very wide variety and large number of features. In the future, when more annotated data becomes available in our studies, we aim to compare the presented results with those of artificial neural networks (ANNs), in particular convolutional neural networks (CNNs), which are nowadays increasingly used in many applications (LeCun et al. 2015) but require large amounts of annotated data, as well as computational power for training, and careful engineering to avoid overfitting (Bianchini and Scarselli 2014; Greenspan et al. 2016; Tajbakhsh et al. 2016; Shaikhina and Khovanova 2017; Litjens et al. 2017; Shen et al. 2017).

Achieving AUROC values between 0.96 and 0.98, the best models considered in the present study are already very suitable for detecting neurons in high-content fluorescence microscopy images. As an example we applied the model using the SVM classifier and the SIFT230 feature set to one

Fig. 11 Example of neuron detection in a high-content fluorescence microscopy image. The image is shown with inverted grayscale intensities compared to the original for better visualization. Here we used the SVM classifier with the SIFT230 feature set to classify patches from a superimposed grid as neuron (the overlaid blue squares) versus non-neuron (no color). Scale bar: 500 μm.

of our images (Fig. 11). Here, a very simple and low-cost detection approach was used, where square patches from a superimposed grid were classified individually as neuron versus non-neuron. If needed, more sophisticated (but more computationally costly) detection schemes with higher localization precision could be easily made, by using finer grids with overlapping patches (keeping the same patch size) and segmenting the positive responses. In our case, detection is the first step in a much more comprehensive pipeline we are developing for fully automated neuron screening, where the actual analysis will take place in much higher-resolution images taken at the detected locations in the low-resolution high-content images. From the results presented in this study we conclude that machine learning approaches are very suitable for the initial detection task.

Acknowledgments Partially supported by the Spanish Ministry of Economy, Industry and Competitiveness (project numbers MTM2014-54151-P, UNLC08-1E-002, UNLC13-13-3503), its Juan de la Cierva Fellowship Program (project number FJCI-2015-26071), the University of La Rioja (project number FPI-UR-13), the European Regional Development Fund (FEDER) of the European Union, the Netherlands Organization for Scientific Reseach (project number 612.001.018), and the Erasmus University Medical Center Fellowship Program.

References

Anderl JL, Redpath S, Ball AJ (2009). A neuronal and astrocyte co-culture assay for high content analysis of neurotoxicity. *Journal of Visualized Experiments* 5(27):1173.

Antony PMA, Trefois C, Stojanovic A, Baumuratov AS, Kozak K (2013). Light microscopy applications in systems biology: opportunities and challenges. *Cell Communication and Signaling* 11(24):1–19.

Arganda-Carreras I, Kaynig V, Rueden C, Eliceiri KW, Schindelin J, Cardona A, Seung HS (2017). Trainable Weka Segmentation: a machine learning tool for microscopy pixel classification. *Bioinformatics* 33(15):2424–2426.

Ascoli GA (2015). *Trees of the Brain, Roots of the Mind*. MIT Press, Cambridge, MA.

Bianchini M, Scarselli F (2014). On the complexity of neural network classifiers: a comparison between shallow and deep architectures. *IEEE Transactions on Neural Networks and Learning Systems* 25(8):1553–1565.

Bischl B, Mersmann O, Trautmann H, Weihs C (2012). Resampling methods for meta-model validation with recommendations for evolutionary computation. *Evolutionary Computation* 20(2):249–275.

Bischl B, Lang M, Kotthoff L, Schiffner J, Richter J, Jones Z, Casalicchio G (2016). *mlr: Machine Learning in R*. URL `https://CRAN.R-project.org/package=mlr`.

Bishop CM (2006). *Pattern Recognition and Machine Learning*. Springer, New York, NY.

Boser BE, Guyon IM, Vapnik VN (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pp. 144–152.

Bougen-Zhukov N, Loh SY, Lee HK, Loo LH (2017). Large-scale image-based screening and profiling of cellular phenotypes. *Cytometry Part A* 91(2):115–125.

Branco P, Torgo L, Ribeiro RP (2016). A survey of predictive modeling on imbalanced domains. *ACM Computing Surveys* 49(2):31:1–31:50.

Breiman L (2001). Random forests. *Machine Learning* 45(1):5–32.

Burges CJC (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery* 2(2):121–167.

Charoenkwan P, Hwang E, Cutler RW, Lee HC, Ko LW, Huang HL, Ho SY (2013). HCS-Neurons: identifying phenotypic changes in multi-neuron images upon drug treatments of high-content screening. *BMC Bioinformatics* 14(S16):S12.

Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research* 16(1):321–357.

Chawla NV, Japkowicz N, Kotcz A (2004). Editorial: Special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter* 6(1):1–6.

Cover T, Hart P (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory* 13(1):21–27.

Cristianini N, Shawe-Taylor J (2000). *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, New York, NY.

Cuesto G, Enriquez-Barreto L, Caramés C, Cantarero M, Gasull X, Sandi C, Ferrús A, Acebes Á, Morales M (2011). Phosphoinositide-3-kinase activation controls synaptogenesis and spinogenesis in hippocampal neurons. *Journal of Neuroscience* 31(8):2721–2733.

Daskalaki S, Kopanas I, Avouris N (2006). Evaluation of classifiers for an uneven class distribution problem. *Applied Artificial Intelligence* 20(5):381–417.

Dehmelt L, Poplawski G, Hwang E, Halpain S (2011). NeuriteQuant: an open source toolkit for high content screens of neuronal mor-

phogenesis. *BMC Neuroscience* 12(100):1–13.

Dragunow M (2008). High-content analysis in neuroscience. *Nature Reviews Neuroscience* 9(10):779–788.

Ebrahimpour MK, Zare M, Eftekhari M, Aghamolaei G (2017). Occam's razor in dimension reduction: using reduced row Echelon form for finding linear independent features in high dimensional microarray datasets. *Engineering Applications of Artificial Intelligence* 62:214–221.

Enriquez-Barreto L, Morales M (2016). The PI3K signaling pathway as a pharmacological target in autism related disorders and schizophrenia. *Molecular and Cellular Therapies* 4:2.

Enriquez-Barreto L, Cuesto G, Dominguez-Iturza N, Gavilán E, Ruano D, Sandi C, Fernández-Ruiz A, Martín-Vázquez G, Herreras O, Morales M (2014). Learning improvement after PI3K activation correlates with de novo formation of functional small spines. *Frontiers in Molecular Neuroscience* 6:54.

Fawcett T (2006). An introduction to ROC analysis. *Pattern Recognition Letters* 27(8):861–874.

Fernandez-Lozano C, Gestal M, Munteanu CR, Dorado J, Pazos A (2016). A methodology for the design of experiments in computational intelligence with multiple regression models. *PeerJ* 4:e2721.

Forman G, Scholz M (2010). Apples-to-apples in cross-validation studies: pitfalls in classifier performance measurement. *ACM SIGKDD Explorations Newsletter* 12(1):49–57.

Friedman J, Hastie T, Tibshirani R (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software* 33(1):1–22.

Friedman M (1940). A comparison of alternative tests of significance for the problem of *m* rankings. *Annals of Mathematical Statistics* 11(1):86–92.

Gabor D (1946). Theory of communication. *Journal of the Institution of Electrical Engineers — Part III: Radio and Communication Engineering* 93(26):429–457.

García V, Mollineda Ra, Sánchez JS (2014). A bias correction function for classification performance assessment in two-class imbalanced problems. *Knowledge-Based Systems* 59:66–74.

Goslin K, Asumussen H, Banker G (1998). Rat hippocampal neurons in low-density culture. In *Culturing Nerve Cells*, The MIT Press, Cambridge, MA, pp. 339–370.

Gradshteyn IS, Ryzhik IM (1994). *Table of Integrals, Series and Products*. Academic Press, New York, NY.

Greenspan H, van Ginneken B, Summers RM (2016). Deep learning in medical imaging: overview and future promise of an exciting new technique. *IEEE Transactions on Medical Imaging* 35(5):1153–1159.

Gupta P, Batra SS, Jayadeva (2017). Sparse short-term time series forecasting models via minimum model complexity. *Neurocomputing* 243:1–11.

Hadjidementriou E, Grossberg M, Nayar S (2001). Spatial information in multiresolution histograms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. I.702–I.709.

Haixiang G, Yijing L, Shang J, Mingyun G, Yuanyue H, Bing G (2017). Learning from class-imbalanced data: review of methods and applications. *Expert Systems with Applications* 73:220–239.

Haralick RM, Shanmugam K, Dinstein I (1973). Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics* 3(6):610–621.

He H, Garcia EA (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering* 21(9):1263–1284.

Hechenbichler K, Schliep K (2004). Weighted k-nearest-neighbor techniques and ordinal classification. *Sonderforschungsbereich* 386(399):1–16.

Hong X, Gao J, Chen S, Harris CJ (2013). Particle swarm optimisation assisted classification using elastic net prefiltering. *Neurocomputing* 122:210–220.

Horvath P, Wild T, Kutay U, Csucs G (2011). Machine learning improves the precision and robustness of high-content screens: using nonlinear multiparametric methods to analyze screening results. *Journal of Biomolecular Screening* 16(9):1059–1067.

Iacca G, Neri F, Mininno E, Ong YS, Lim MH (2012). Ockham's razor in memetic computing: three stage optimal memetic exploration. *Information Sciences* 188:17–43.

Jain S, van Kesteren RE, Heutink P (2012). High content screening in neurodegenerative diseases. *Journal of Visualized Experiments* 59:e3452.

Jiang RM, Crookes D, Luo N, Davidson MW (2010). Live-cell tracking using SIFT features in DIC microscopic videos. *IEEE Transactions on Biomedical Engineering* 57(9):2219–2228.

Kraus OZ, Frey BJ (2016). Computer vision for high content screening. *Critical Reviews in Biochemistry and Molecular Biology* 51(2):102–109.

Krawczyk B (2016). Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence* 5(4):221–232.

Kuminski E, George J, Wallin J, Shamir L (2014). Combining human and machine learning for morphological analysis of galaxy images. *Publications of the Astronomical Society of the Pacific* 126(944):959–967.

LeCun Y, Bengio Y, Hinton G (2015). Deep learning. *Nature* 521(7553):436–444.

Lee DH, Lee DW, Han BS (2016). Possibility study of scale invariant feature transform (SIFT) algorithm application to spine magnetic resonance imaging. *PLOS ONE* 11(4):1–9.

Li J, Fong S, Wong RK, Chu VW (2018). Adaptive multi-objective swarm fusion for imbalanced data classification. *Information Fusion* 39:1–24.

Liaw A, Wiener M (2002). Classification and regression by randomForest. *R News* 2(3):18–22.

Litjens G, Kooi T, Bejnordi BE, Setio AAA, Ciompi F, Ghafoorian M, van der Laak JAWM, van Ginneken B, Sánchez CI (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis* 42:60–88.

Lowe DG (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60(2):91–110.

MacQueen J (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability — Volume 1: Statistics*, University of California Press, Berkeley, CA, pp. 281–297.

Mata G, Radojević M, Smal I, Morales M, Meijering E, Rubio J (2016). Automatic detection of neurons in high-content microscope images using machine learning approaches. In *Proceedings of the IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 330–333.

MathWorks (2016). *Version 9.0.0.341360 (R2016a)*. The MathWorks Inc., Natick, MA.

Meijering E (2010). Neuron tracing in perspective. *Cytometry Part A* 77(7):693–704.

Meijering E, Carpenter AE, Peng H, Hamprecht FA, Olivo-Marin JC (2016). Imagining the future of bioimage analysis. *Nature Biotechnology* 34(12):1250–1255.

Meyer D, Dimitriadou E, Hornik K, Weingessel A, Leisch F (2017). *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*. URL `https://CRAN.R-project.org/package=e1071`.

Mualla F, Scholl S, Sommerfeldt B, Maier A, Hornegger J (2013). Automatic cell detection in bright-field microscope images using SIFT, random forests, and hierarchical clustering. *IEEE Transactions on Medical Imaging* 32(12):2274–2286.

Nemenyi PB (1963). *Distribution-Free Multiple Comparisons*. Princeton University, Princeton, NJ.

Ni D, Chui YP, Qu Y, Yang XS, Qin J, Wong TT, Ho SSH, Heng PA (2009). Reconstruction of volumetric ultrasound panorama based on improved 3D SIFT. *Computerized Medical Imaging and Graphics* 33(7):559–566.

Orlov N, Shamir L, Macura T, Johnston J, Eckley DM, Goldberg IG (2008). WND-CHARM: Multi-purpose image classification using compound image transforms. *Pattern Recognition Letters* 29(11):1684–1693.

Otsu N (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics* 9(1):62–66.

van Pelt J, van Ooyen A, Uylings H (2001). The need for integrating neuronal morphology databases and computational environments in exploring neuronal structure and function. *Anatomy and Embryology* 204(4):255–265.

Prewitt JMS (1970). Object enhancement and extraction. In *Picture Processing and Psychopictorics*, Academic Press, New York, NY, pp. 75–149.

R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, URL `https://www.R-project.org/`.

Radio N (2012). Neurite outgrowth assessment using high content analysis methodology. *Methods in Molecular Biology* 846:247–260.

Ramón y Cajal S (1899). *Histología del sistema nervioso del hombre y de los vertebrados*. CSIC, Madrid, reprinted in 2007.

Saeys Y, Inza I, Larrañaga P (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics* 23(19):2507–2517.

Samworth RJ (2012). Optimal weighted nearest neighbour classifiers. *The Annals of Statistics* 40(5):2733–2763.

Schliep K, Hechenbichler K (2016). *kknn: Weighted k-Nearest Neighbors*. URL `https://CRAN.R-project.org/package=kknn`.

Shaikhina T, Khovanova NA (2017). Handling limited datasets with neural networks in medical applications: a small-data approach. *Artificial Intelligence in Medicine* 75:51–63.

Shamir L (2012). Automatic detection of peculiar galaxies in large datasets of galaxy images. *Journal of Computational Science* 3(3):181–189.

Shamir L, Tarakhovsky JA (2012). Computer analysis of art. *Journal on Computing and Cultural Heritage* 5(2):7.

Shamir L, Orlov N, Eckley DM, Macura T, Johnston J, Goldberg IG (2008). Wndchrm – an open source utility for biological image analysis. *Source Code for Biology and Medicine* 3(1):1–13.

Shamir L, Delaney JD, Orlov N, Eckley DM, Goldberg IG (2010). Pattern recognition software and techniques for biological image analysis. *PLOS Computational Biology* 6(11):e1000974.

Shapiro SS, Wilk MB (1965). An analysis of variance test for normality (complete samples). *Biometrika* 52(3-4):591–611.

Shen D, Wu G, Suk HI (2017). Deep learning in medical image analysis. *Annual Review of Biomedical Engineering* 19:221–248.

Simon R (2007). Resampling strategies for model assessment and selection. In *Fundamentals of Data Mining in Genomics and Proteomics*, Springer, Boston, MA, pp. 173–186.

Singh S, Carpenter AE, Genovesio A (2014). Increasing the content of high-content screening: an overview. *Journal of Biomolecular Screening* 19(5):640–650.

Sivic J (2009). Efficient visual search of videos cast as text retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31(4):591–605.

Smafield T, Pasupuleti V, Sharma K, Huganir RL, Ye B, Zhou J (2015). Automatic dendritic length quantification for high throughput screening of mature neurons. *Neuroinformatics* 13(4):443–458.

Sommer C, Gerlich DW (2013). Machine learning in cell biology – teaching computers to recognize phenotypes. *Journal of Cell Science* 126(24):5529–5539.

Squire LR (1992). Memory and the hippocampus: a synthesis from findings with rats, monkeys, and humans. *Psychological Review* 99(2):195–231.

Tajbakhsh N, Shin JY, Gurudu SR, Hurst RT, Kendall CB, Gotway MB, Liang J (2016). Convolutional neural networks for medical image analysis: full training or fine tuning? *IEEE Transactions on Medical Imaging* 35(5):1299–1312.

Tamura H, Mori S, Yamawaki T (1978). Textural features corresponding to visual perception. *IEEE Transactions on Systems, Man, and Cybernetics* 8(6):460–473.

Tibshirani R (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 58(1):267–288.

Uhlmann V, Singh S, Carpenter AE (2016). CP-CHARM: segmentation-free image classification made accessible. *BMC Bioinformatics* 17(1):51.

Vallotton P, Lagerstrom R, Sun C, Buckley M, Wang D, Silva MD, Tan SS, Gunnersen JM (2007). Automated analysis of neurite branching in cultured cortical neurons using HCA-Vision. *Cytometry Part A* 71(10):889–895.

Vapnik VN (1998). *Statistical Learning Theory*. John Wiley & Sons, New York, NY.

Vapnik VN (1999). *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, NY.

Vedaldi A, Fulkerson B (2008). *VLFeat: An Open and Portable Library of Computer Vision Algorithms*. URL `http://www.vlfeat.org/`.

Vert JP, Tsuda K, Schölkopf B (2004). A primer on kernel methods. In *Kernel Methods in Computational Biology*, MIT Press, Cambridge, MA, USA, pp. 35–70.

Wilcoxon F (1945). Individual comparisons by ranking methods. *Biometrics Bulletin* 1(6):80–83.

Wu C, Schulte J, Sepp KJ, Littleton JT, Hong P (2010). Automatic robust neurite detection and morphological analysis of neuronal cell cultures in high-content screening. *Neuroinformatics* 8(2):83–100.

Xia X, Wong STC (2012). Concise review: a high-content screening approach to stem cell research and drug discovery. *Stem Cells* 30(9):1800–1807.

Yu D, Yang F, Yang C, Leng C, Cao J, Wang Y, Tian J (2016). Fast rotation-free feature-based image registration using improved N-SIFT and GMM-based parallel optimization. *IEEE Transactions on Biomedical Engineering* 63(8):1653–1664.

Zhang R, Zhou W, Li Y, Yu S, Xie Y (2013). Nonrigid registration of lung CT images based on tissue features. *Computational and Mathematical Methods in Medicine* 2013:834192.

Zhang Y, Zhou X, Degterev A, Lipinski M, Adjeroh D, Yuan J, Wong STC (2007). A novel tracing algorithm for high throughput imaging: screening of neuron-based assays. *Journal of Neuroscience Methods* 160(1):149–162.

Zou H, Hastie T (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67(2):301–320.