Neuroinformatics (2019) 17:253–269
https://doi.org/10.1007/s12021-018-9399-4

ORIGINAL ARTICLE

Automated Neuron Detection in High-Content Fluorescence Microscopy Images Using Machine Learning

Gadea Mata[1] · Miroslav Radojević[2] · Carlos Fernandez-Lozano[3,4] · Ihor Smal[2] · Niels Werij[2] · Miguel Morales[5] · Erik Meijering[2] · Julio Rubio[1]

Published online: 13 September 2018

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 high-content screening and analysis. 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 well-established machine learning techniques for this purpose. These include support vector machines, random forests, k-nearest neighbors, and generalized linear model 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 on a dataset of rat hippocampal neurons from our own studies to find 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 random forests classifier using the right feature subset ranks best for the considered task, although its performance is not statistically significantly better than some support vector machine based classification models.

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

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 (2007) 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

Gadea Mata, Miroslav Radojević and Carlos Fernandez-Lozano contributed equally to this work.

✉ Gadea Mata
gadea.mata@unirioja.es

[1] Department of Mathematics and Computer Science, University of La Rioja, Logroño, Spain

[2] Biomedical Imaging Group Rotterdam, Departments of Medical Informatics and Radiology, Erasmus University Medical Center, Rotterdam, Netherlands

[3] Department of Computer Science, University of A Coruña, A Coruña, Spain

[4] Instituto de Investigación Biomédica de A Coruña, Complexo Hospitalario Universitario de A Coruña, A Coruña, Spain

[5] Molecular Cognition Laboratory, Biophysics Institute, CSIC-UPV/EHU, Campus Universidad del País Vasco, Leioa, Spain

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 explore classifiers based on precalculated image features in order to determine which combinations of classifiers and features work best in a practical setting where there is very limited annotated data for training. Specifically, we consider various state-of-the-art classifiers based on support vector machines (SVM), random forests (RF), k-nearest neighbors (KNN), and generalized linear models (in particular GLMNET), 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).

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.

Image Dataset

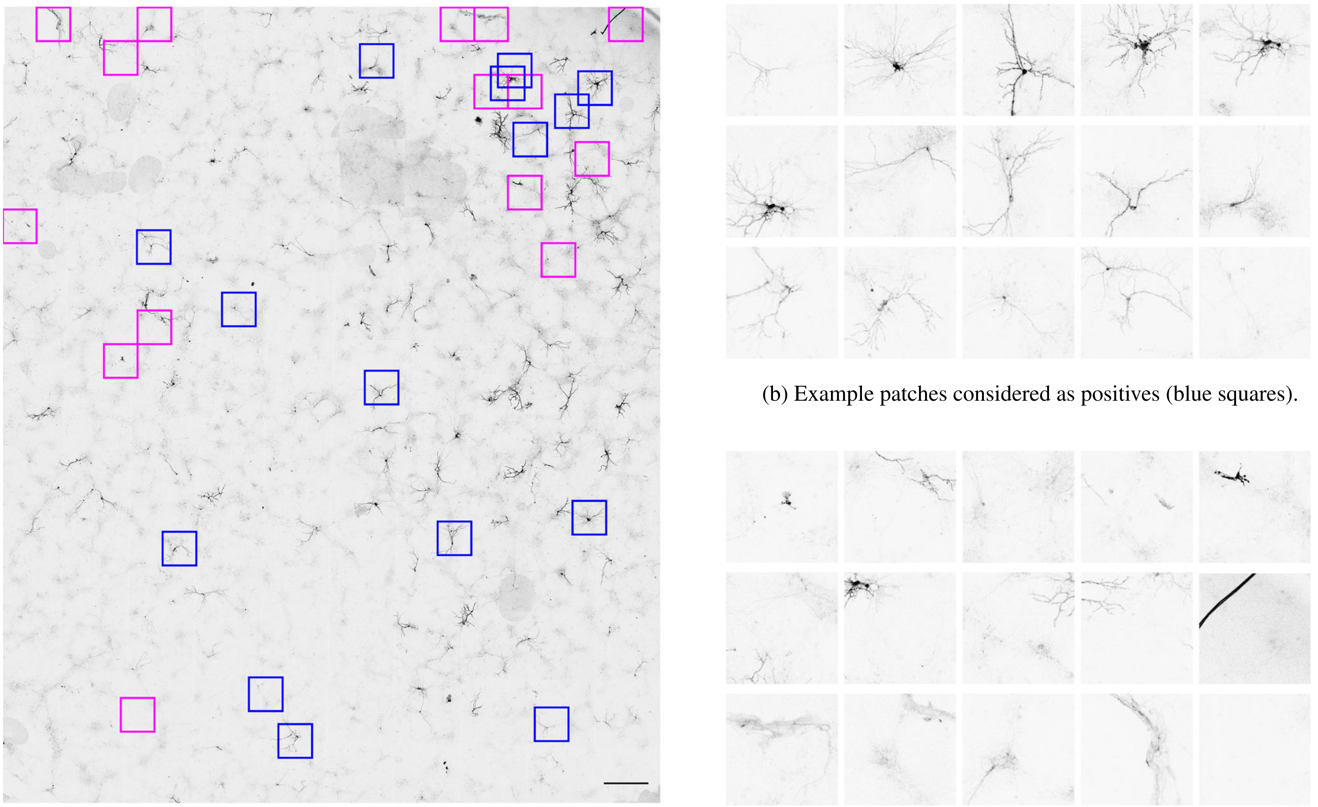
The high-content image data used in this study is from our ongoing research into effective treatments for neurological disorders (Cuesto et al. 2011; Enriquez-Barreto et al. 2014; Enriquez-Barreto and Morales 2016). We describe the acquisition of the images, their annotation, and the strategy we used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

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 × 12,000 pixels, covering approximately 70 mm^2 of culture dish. Each image is a mosaic made up of tiles of size 1024 × 1024 pixels, automatically acquired and stitched using the Leica Matrix module. Prior to imaging, the user has to select the desired culture area within the field of view, and the module calculates the tiles to be imaged in order to cover the chosen area, considering 10% overlap between neighboring tiles. Each mosaic contains 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.

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



(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**) where the blue squares highlight some example patches containing neuronal structure and the magenta squares depict some example patches containing background. These squares are enlarged in (**b**) and (**c**) for a better visualization. The intensities of the shown images are inverted compared to their originals for displaying purposes

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

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 in order to obtain approximately the same number of samples in each class. To this end we employed the popular synthetic minority oversampling technique (SMOTE) (Chawla et al. 2002) of which several variants exist (Sáez et al. 2015; Krawczyk 2016; Gosain and Sardana 2017). 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

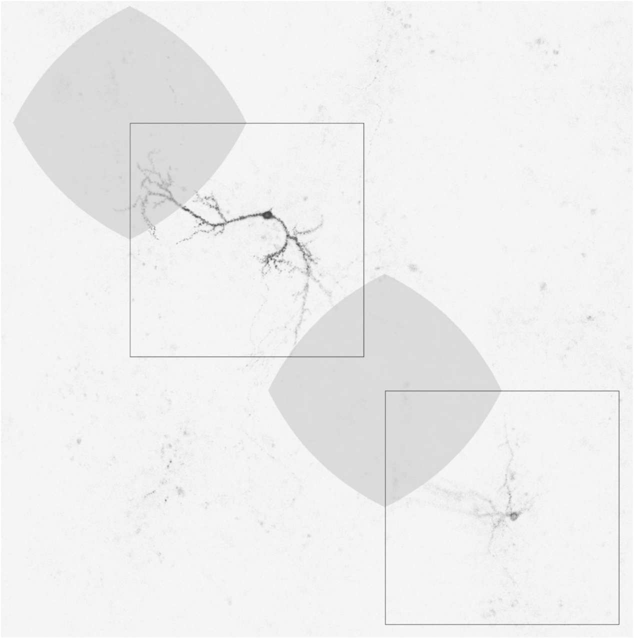


Fig. 2 Two example neurons with their expert-marked ROIs (black squares) and their potential alternative positive patch locations (gray regions). The latter comprise all possible top-left corner positions of patches with the same size as the given ROI and having 50% or more area overlap with that ROI

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 in total.

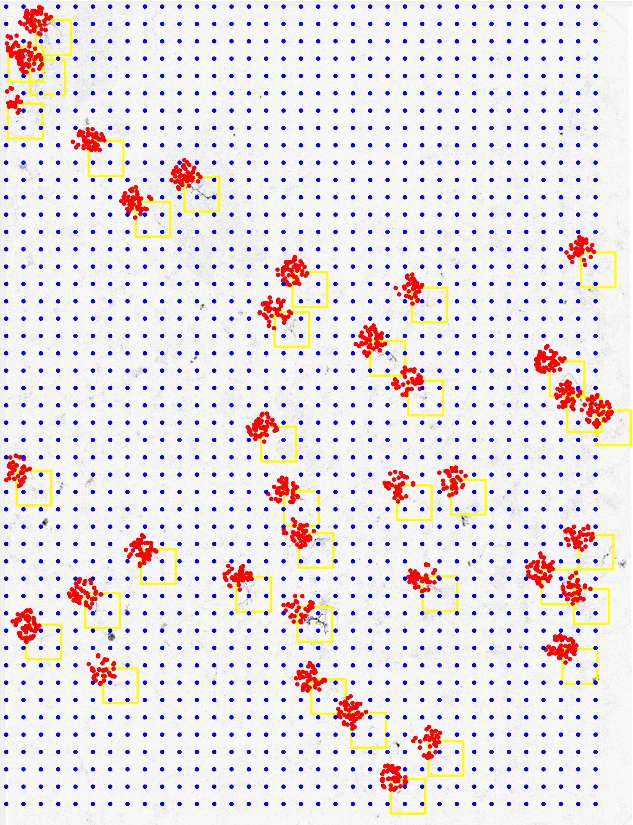


Fig. 3 Example of positive patch oversampling. The background shows a high-content fluorescence microscopy image (with intensities inverted), and the graphical overlay shows the neuron ROIs marked by the expert (yellow squares), the top-left corners of the patches randomly sampled from all possible patches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

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 (CHARM) and the scale-invariant feature transform (SIFT). Here we briefly describe each of them. In the training stage of the machine learning algorithms, feature values were normalized to zero mean and unit variance per feature over the whole data set, and constant features were pruned.

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 2012a; Kuminski et al. 2014) and in art (Shamir and Tarakhovsky 2012b). 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 et al. (1973) and 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).

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 (Yang et al. 2009). A very effective and popular approach to achieve this is to use the bag-of-words (BoW) model (Fei-Fei and Perona 2005). 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 single cluster to which it is closest according to the Mahalanobis distance. Such mapping yields a histogram 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.

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.

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 (nonlinear) 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 considered values 2^k for integer $k = -12, \ldots, 12$ for both parameters.

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.

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

Generalized Linear Model (GLMNET) via penalized maximum likelihood (Friedman et al. 2010) is a regularized statistical model whose response variable is a Bernoulli indicator used for classification. It is based on the least absolute shrinkage and selection operator (LASSO) (Tibshirani 1996). Similar to LASSO, this method simultaneously performs automatic feature selection and continuous shrinkage (regularization), and is able to select groups of correlated features. Specifically, GLMNET combines l_1 and l_2 penalties for regularization, and 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), random-Forest (Liaw and Wiener 2002), kknn (Schliep and Hechenbichler 2016), and GLMnet (Friedman et al. 2010), to evaluate all the machine learning algorithms. Most of the result plots presented in this paper were generated using the R package ggplot2 (Wickham 2009).

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 aggregation. 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 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.

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.

Initial Exploratory Results

For the initial experiment we constructed 17 different feature sets from (combinations of) the CHARM features and the SIFT features: CHARM features only (one set), SIFT features only (eight sets, one for each of the eight BoW vector lengths), and the union of CHARM and SIFT features (eight sets). 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 more computationally feasible, we used a simpler resampling strategy than described, namely a single 10-fold cross-validation in the outer loop, and a holdout approach in the inner loop. In the latter, the optimal hyperparameters of the classification algorithms were obtained using a grid search on 2/3rd of the training set of the outer loop, and validated on the remaining 1/3rd.

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.

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 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.

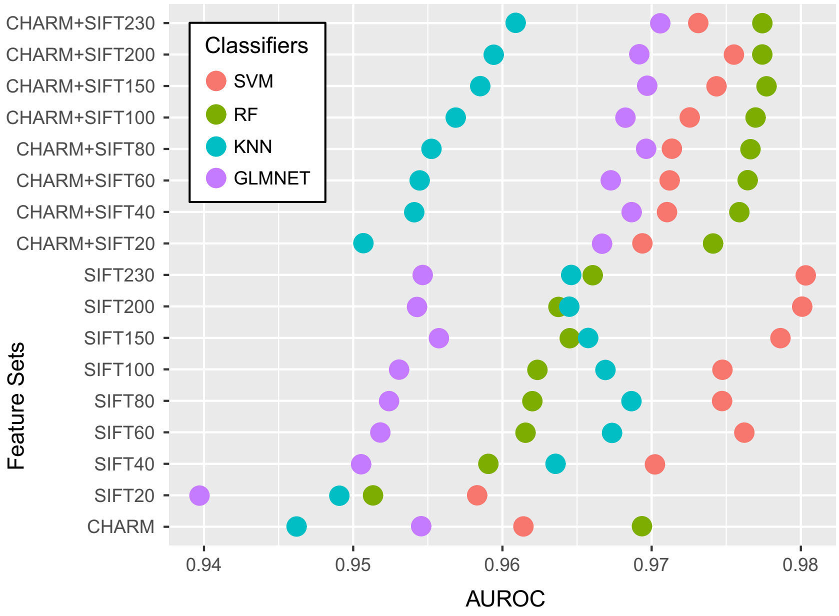


Fig. 4 Results of the initial exploratory experiment. Each of the considered classifiers (SVM, RF, KNN, GLMNET) was evaluated for each of the described 17 feature sets according to the performance measure (AUROC) using the described simplified resampling strategy

Cross-Validation Results

Based on the results of the initial exploratory experiment we selected three feature sets, corresponding to CHARM features only, SIFT230 features only, and 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 "Statistical Analysis Results").

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

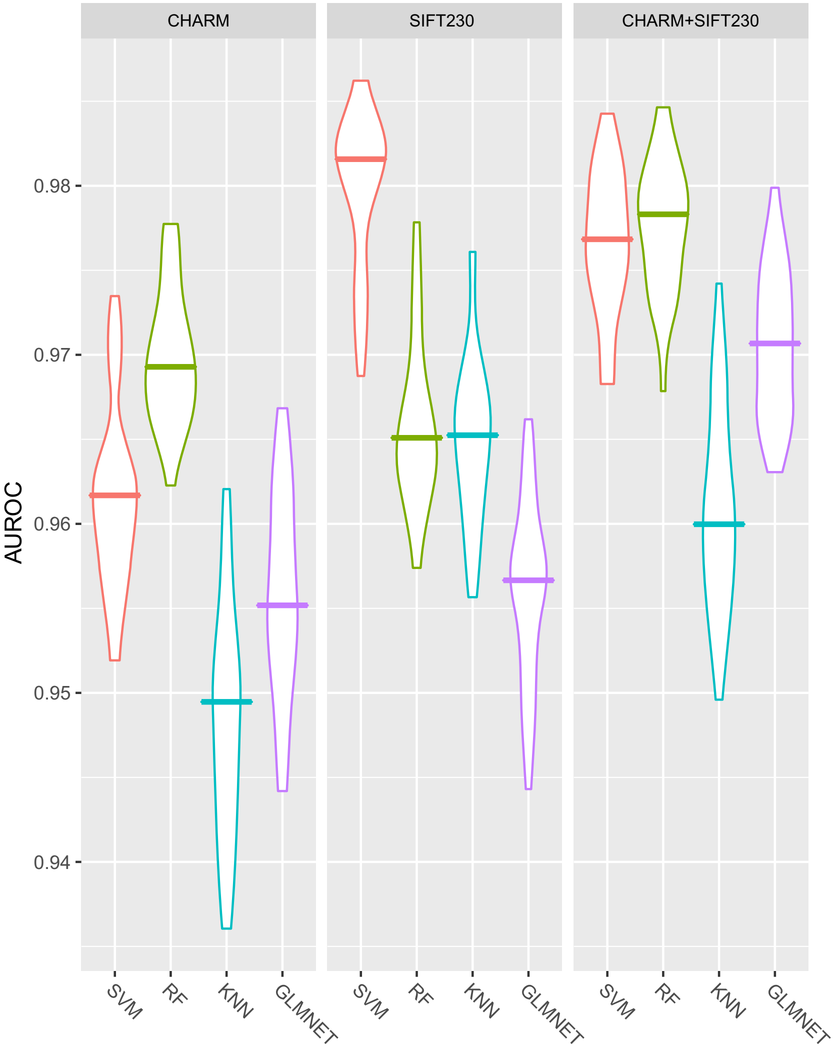


Fig. 5 Results of the cross-validation experiment. Each of the considered classifiers (SVM, RF, KNN, GLMNET) was evaluated for each of the selected feature sets (CHARM, SIFT230, CHARM+SIFT230) using the performance measure (AUROC). The results are shown as violin plots, where the horizontal bar indicates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

that would make the classification task computationally cheaper.

To this end we ranked all 1,289 features using a CForest test (Strobl et al. 2009) and considered four subsets, consisting of the top 25, 100, 200, and 600 features. The results (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 "Statistical Analysis Results"). 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 subset is dominated by the SIFT features and the Zernike coefficients from CHARM, whereas the top 100, 200, and 600 subsets include many other types of features (about twice as many), in roughly similar proportions. These additional features contribute important information to the classification process, as follows from the fact that the performance of the larger subsets is considerably better than that of the top 25 subset. However, the reasons why these specific types of features are dominant, elude us. According to the feature selection results (Fig. 6), the best performing classification model is the RF using the CHARM+SIFT230:600 feature subset (AUROC = 0.9784), followed very closely by the SVM using the CHARM+SIFT230:200 feature subset (AUROC = 0.9783). Studying the importance of the features in the former model according to the Gini index (Breiman 2001), we observe (Fig. 8) that the most important features are indeed from the SIFT set together with the Zernike coefficients from the CHARM set. Other important top features from the CHARM set in decreasing order include the Tamura and Haralick textures, multiscale histograms, combined moments, and others (Fig. 7).

Statistical Analysis Results

Finally we analyzed the statistical significance of the results (AUROC values) 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 $W = 0.97324$ and $p < 2.723 \cdot 10^{-11}$ (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), which is known to yield conservative results in the case of relatively small numbers of algorithms

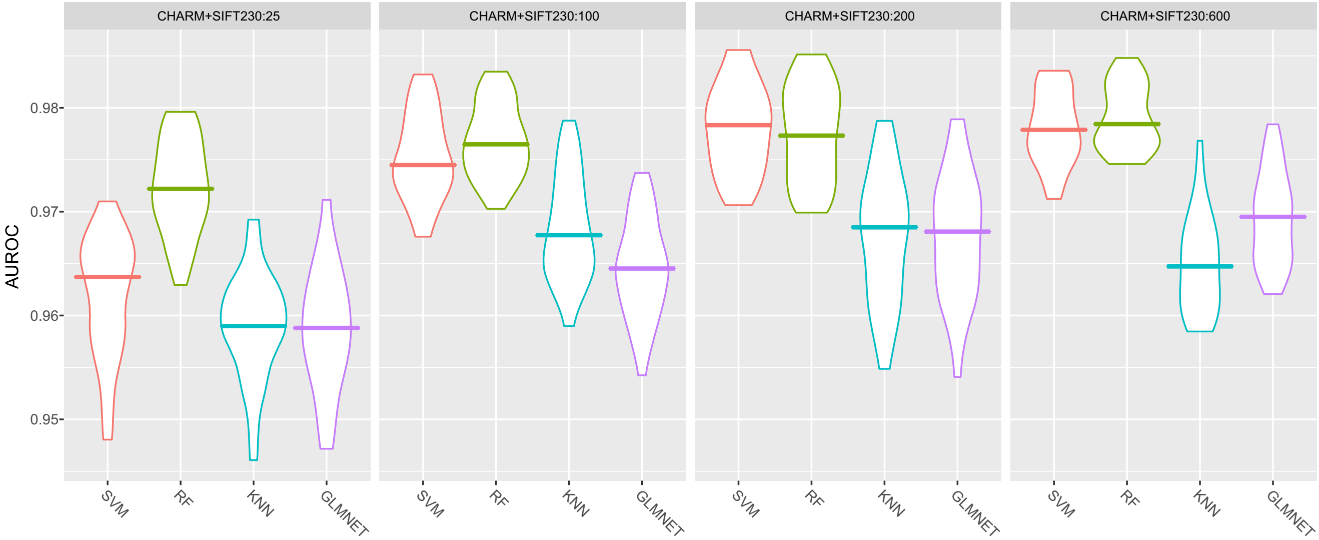


Fig. 6 Performance (AUROC) of the considered classifiers (SVM, RF, KNN, GLMNET) for different feature subsets (the top 25, 100, 200, and 600 features from the CHARM+SIFT230 set). The results are shown as violin plots, where the horizontal bar indicates the median value, the vertical extent is the interquartile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

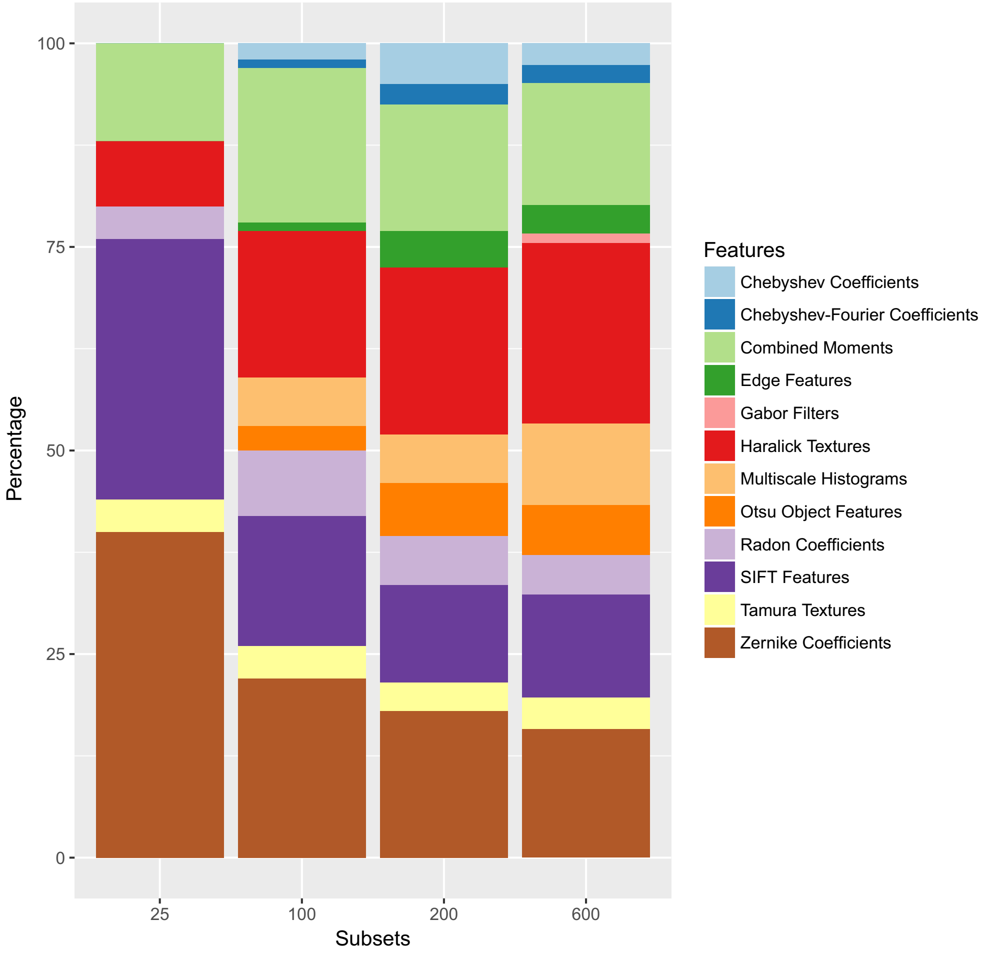


Fig. 7 Cumulative percentages of the different types of features contained in the four subsets (the top 25, 100, 200, and 600 features selected from the CHARM+SIFT230 set)

and datasets (García et al. 2010). We used the null hypothesis that all models yield the same performance on our data, and we rejected it with very significant values of $\chi^2 = 657$ and $p < 2.25 \cdot 10^{-10}$. 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 Finner test (Finner 1993), with the control model being the RF classifier using the CHARM+SIFT230:600 feature set, as it performed best in the feature selection experiment (Fig. 6).

The results (Fig. 10) show that several other models performed statistically similar to the control model. These include the SVM classifier using the SIFT230 feature set or the top 100, 200, or 600 features of the CHARM+SIFT230 set. Other statistically similar models include the RF classifier using the CHARM+SIFT230 feature set, or just the top 100 or 200 features of the latter. None of the models based on the KNN and GLMNET classifiers performed statistically similar to the control model.

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), SVM using CHARM+SIFT (the top 100, 200, or 600 features), and RF using CHARM+SIFT (the full 1,289 features or only the top 100, 200, or 600 features). In the course of our study we have also explored

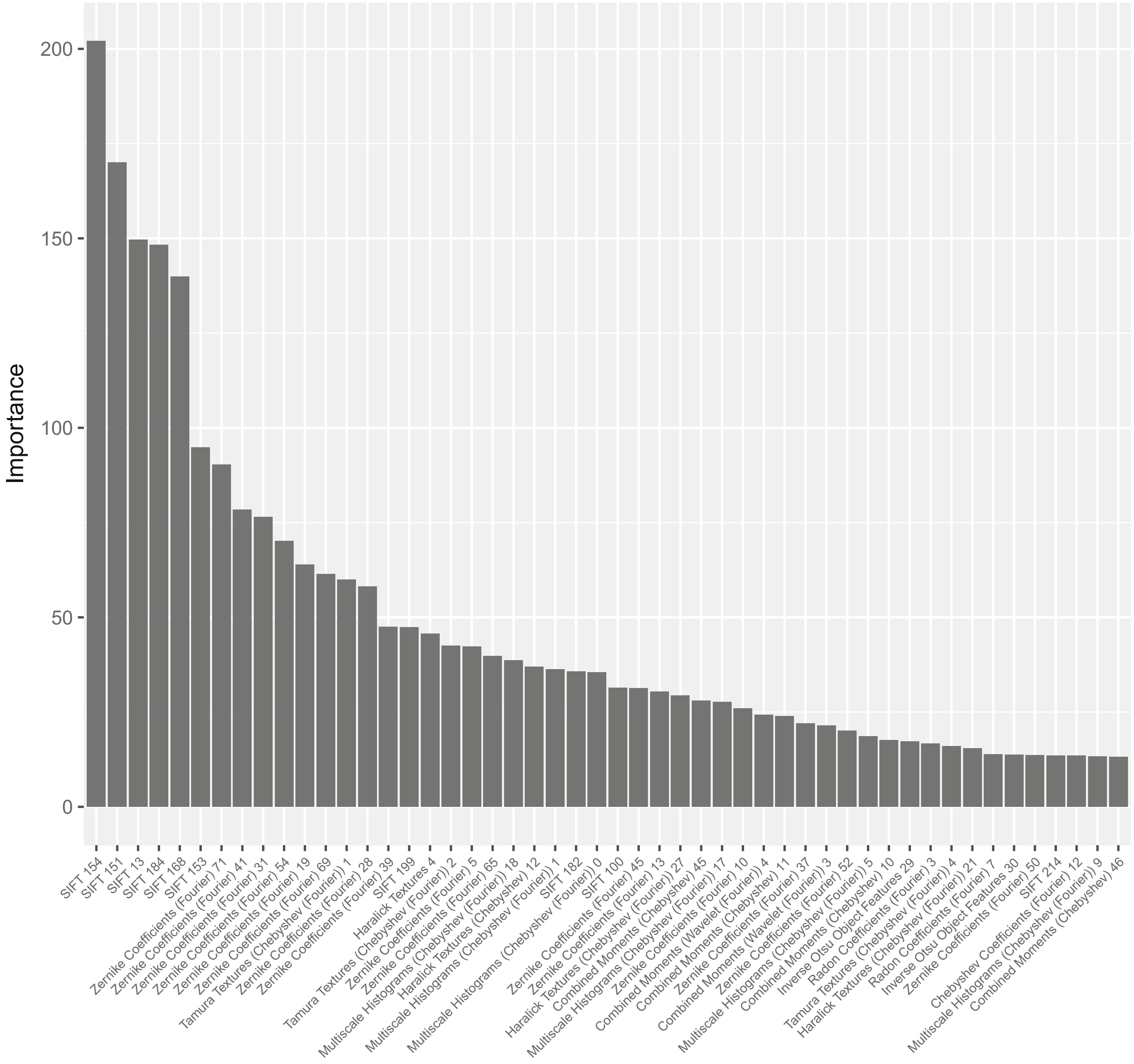


Fig. 8 The 50 most important features from the CHARM+SIFT230: 600 feature subset used by the best performing classifier. Importance was calculated according to the Gini index of the RF classifier. The importance value for each feature was averaged over all runs and folds of the cross-validation experiment

the potential of several alternative features, such as the histogram of oriented gradients (HOG) (Dalal and Triggs 2005) and spatial pyramid matching (SPM) (Lazebnik et al. 2006) based on sparse coding (ScSPM) (Yang et al. 2009), but the results were not as good.

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 expect deep learning approaches to be good and possibly superior alternatives, as they have

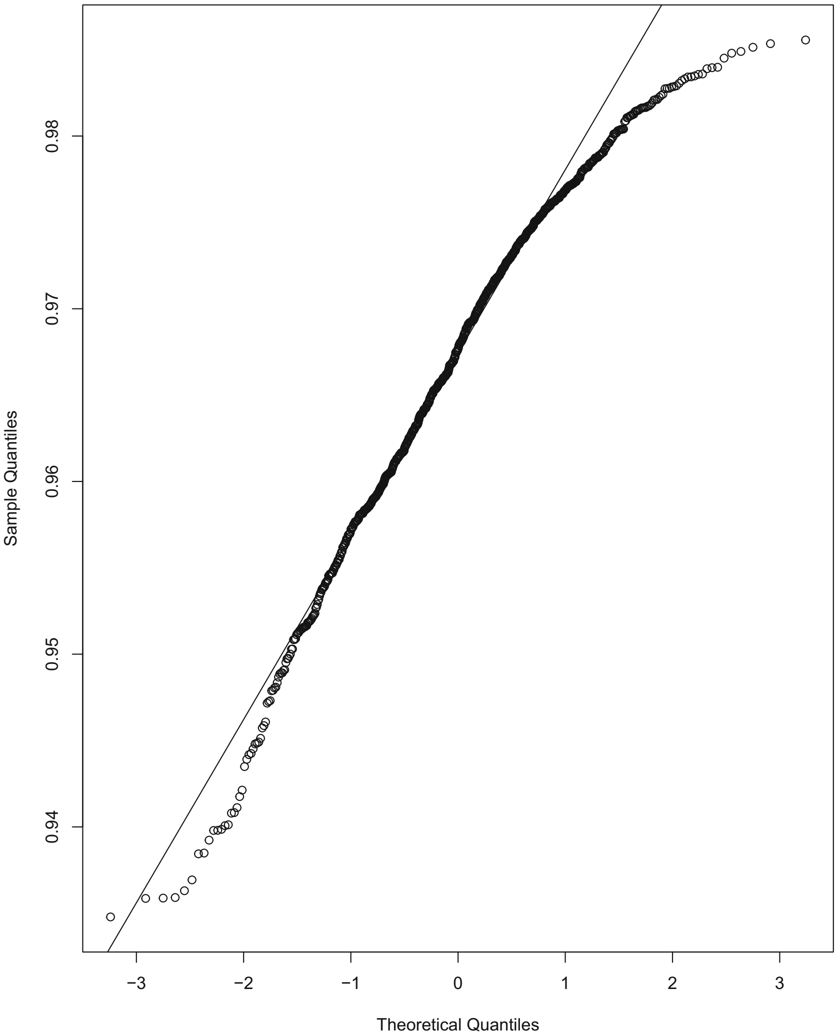
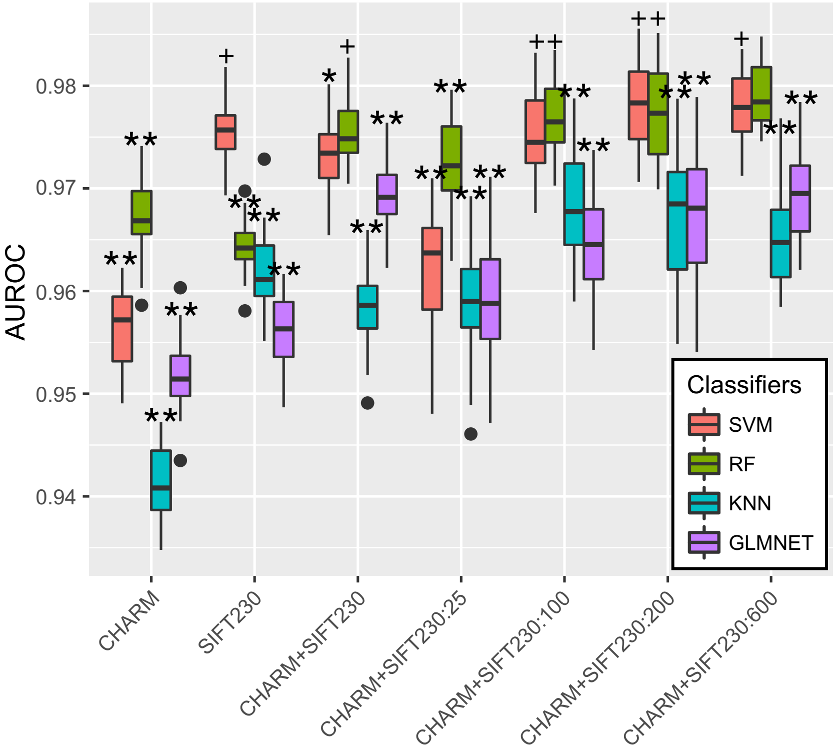


Fig. 9 Quantile-quantile (Q-Q) plot of the theoretical normal distribution and our data samples. Clearly, the computed values (small circles) deviate substantially from a straight line (the solid line is the least squares fit) and reveal a nonlinear relationship, leading to the conclusion that our data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

been very successful in many other applications (Bianchini and Scarselli 2014; LeCun et al. 2015; Greenspan et al. 2016; Tajbakhsh et al. 2016; Shaikhina and Khovanova 2017; Litjens et al. 2017; Shen et al. 2017). To get an impression of their performance on our current data, we performed a pilot experiment with three convolutional neural networks. The first was a home-built network (HBN17) with 17 convolutional layers, interspersed with six max-pooling layers, and followed by two fully connected layers outputting the two class probabilities (neuron versus background). The second network was VGG19 (Simonyan and Zisserman 2014), with one modification, because the image patch sizes in our study were more than four times larger than what VGG19 was originally designed for, which increased the number of network parameters and thus the memory usage to the point that we were not able to train the network on our available computers. Therefore we reduced the number of filters in the convolution layers by a factor of 16 and made the network return only two class probabilities to match our application. And the third network was ResNet50 (He et al. 2016) modified so as to return only two class probabilities. To train the networks we used categorical cross-entropy (Ghosh et al. 2017) as the loss function and Adam (Kingma and Ba 2014) as the optimizer. The networks were trained on the same balanced data set as the classifiers studied in this work and were tested using the same 3 × 10-fold cross-validation approach. The results showed that VGG19 performed best (median AUROC of 0.960), followed by ResNet50 (median AUROC of 0.947), and HBN17 (median AUROC of 0.936). Clearly the networks are as yet outperformed by the best classifiers considered in this study. Better results may be achieved not only by acquiring more data but also by applying stronger data augmentation than done here. Another direction for future research would be to reformulate the problem as a multiclass detection challenge, distinguishing not only between neurons and background, but also incomplete or out-of-focus neurons, astrocytes, and artifacts.

Achieving AUROC values between 0.97 and 0.98, the best models considered in the present study are already very

Fig. 10 Results of the Friedman-Finner 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 (RF using CHARM+SIFT230:600). Performance values (AUROC) of each model from all runs and folds of the cross-validation experiment are summarized using the ggplot2 box plot. Significance with respect to the control model is indicated for $p > 0.05$ (+), and $0.01 < p < 0.05$ (*), and $p < 0.01$ (**)



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 of our images (Fig. 11). In addition, to investigate generalizability, we also applied it to a new, "unseen" image from a new experiment. In that experiment, to introduce

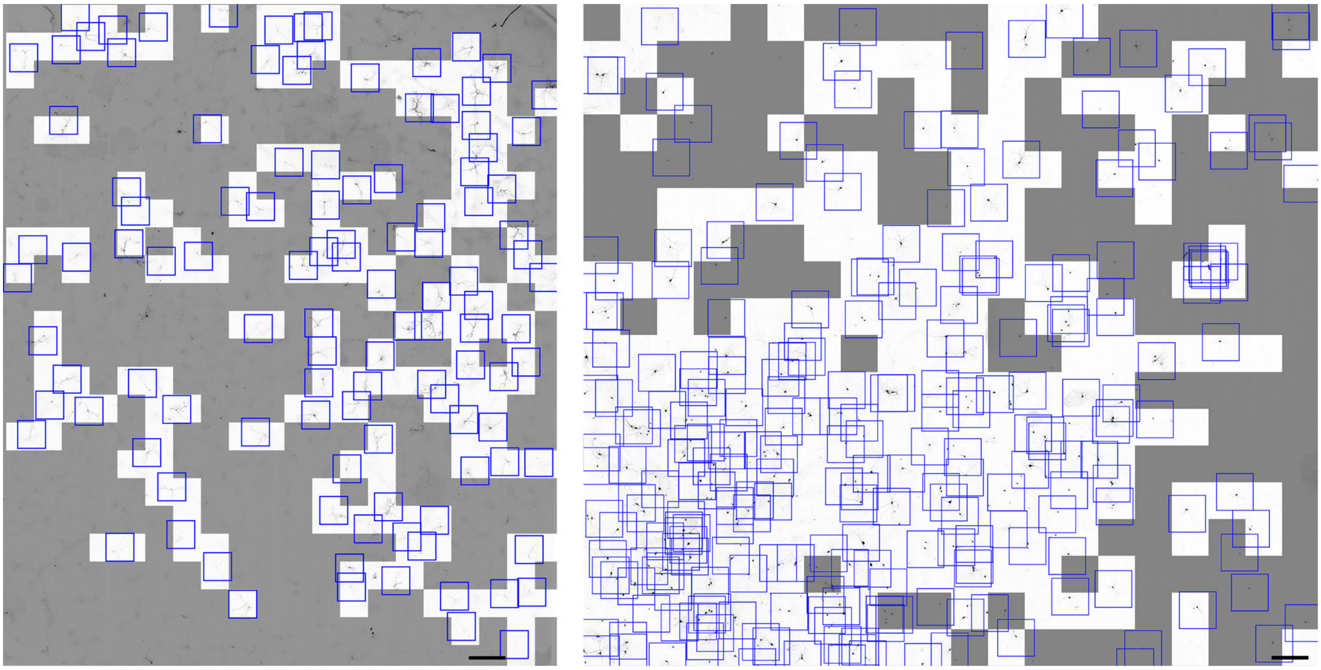


Fig. 11 Example of neuron detection in high-content fluorescence microscopy images. The images are shown with inverted intensities (dark grayscale parts) compared to the original. Left: One of the eight images used in the cross-validation experiment. Right: A new image acquired in a later experiment and not used in the cross-validation experiment. Here we used the SVM classifier with the SIFT230 feature set to classify square patches from a superimposed grid as neuron (bright grayscale) versus non-neuron (dark grayscale). The detected neuron regions correspond very well with the expert human annotations (blue squares). Scale bars: 500 μm

some variability, we used a transfection method with higher efficiency (Bredenbeek et al. 1993), resulting in higher intensities and larger numbers of neurons in the field of view. In both images, to detect the neurons, a very simple and low-cost detection approach was used, where square patches (same patch size as used throughout this study) 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. But in our work, detection is only the first step in a much more comprehensive pipeline we are developing for fully automated neuron screening, where the actual neuron reconstruction and downstream morphological analysis is based on much higher-resolution images taken at the locations detected in the low-resolution high-content images. From the results presented in this study we conclude that machine learning approaches are very suitable for this initial detection task and can drastically reduce the high-resolution scan time and analysis.

Information Sharing Statement

All presented data and code are available from http://www.unirioja.es/cu/jurubio/ANDHCFMIUML/.

Acknowledgments This work was partially supported by the Spanish Ministry of Economy, Industry and Competitiveness (project numbers MTM2014-54151-P, UNLC08-1E-002, UNLC13-13-3503), 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. Carlos Fernandez-Lozano was supported by a Juan de la Cierva postdoctoral fellowship grant (Spanish Ministry of Economy, Industry and Competitiveness, FJCI-2015-26071).

References

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

Antony, P.M.A., Trefois, C., Stojanovic, A., Baumuratov, A.S., 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, K.W., Schindelin, J., Cardona, A., Seung, H.S. (2017). Trainable Weka Segmentation: a machine learning tool for microscopy pixel classification. *Bioinformatics*, *33*(15), 2424–2426.

Ascoli, G.A. (2015). *Trees of the brain, roots of the mind*. Cambridge: MIT Press.

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. https://CRAN.R-project.org/package=mlr.

Bishop, C.M. (2006). *Pattern recognition and machine learning*. New York: Springer.

Boser, B.E., Guyon, I.M., Vapnik, V.N. (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, S.Y., Lee, H.K., Loo, L.H. (2017). Large-scale image-based screening and profiling of cellular phenotypes. *Cytometry Part A*, *91*(2), 115–125.

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

Bredenbeek, P.J., Frolov, I., Rice, C.M., Schlesinger, S. (1993). Sindbis virus expression vectors: packaging of RNA, replicons by using defective helper RNAs. *Journal of Virology*, *67*(11), 6439–6446.

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

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

Charoenkwan, P., Hwang, E., Cutler, R.W., Lee, H.C., Ko, L.W., Huang, H.L., Ho, S.Y. (2013). HCS-Neurons: identifying phenotypic changes in multi-neuron images upon drug treatments of high-content screening. *BMC Bioinformatics*, *14*(S16), S12.

Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*(1), 321–357.

Chawla, N.V., 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.

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.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 886–893).

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 morphogenesis. *BMC Neuroscience*, *12*(100), 1–13.

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

Ebrahimpour, M.K., 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., 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.

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.

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

Fei-Fei, L., & Perona, P. (2005). A Bayesian hierarchical model for learning natural scene categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (Vol. 2 pp. 524–531).

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

Finner, H. (1993). On a monotonicity problem in step-down multiple test procedures. *Journal of the American Statistical Association*, *88*(423), 920–923.

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

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

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

García, S., Fernández, A., Luengo, J., Herrera, F. (2010). Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power. *Information Sciences*, *180*(10), 2044–2064.

García, V., Mollineda, R.A., Sànchez, J.S. (2014). A bias correction function for classification performance assessment in two-class imbalanced problems. *Knowledge-Based Systems*, *59*, 66–74.

Ghosh, A., Kumar, H., Sastry, P.S. (2017). Robust loss functions under label noise for deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence* (pp. 1919–1925).

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

Gosain, A., & Sardana, S. (2017). Handling class imbalance problem using oversampling techniques: a review, In *Proceedings of the International Conference on Advances in Computing, Communications and Informatics* (pp. 79–85).

Gradshteyn, I.S., & Ryzhik, I.M. (1994). *Table of integrals, series and products*. New York: Academic Press.

Greenspan, H., van Ginneken, B., Summers, R.M. (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, S.S., 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, R.M., Shanmugam, K., Dinstein, I. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, *3*(6), 610–621.

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

He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770–778).

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, C.J. (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, Y.S., Lim, M.H. (2012). Ockham's razor in memetic computing: three stage optimal memetic exploration. *Information Sciences*, *188*, 17–43.

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

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

Kingma, D.P., & Ba, J. (2014). Adam: a method for stochastic optimization, Computing Research Repository arXiv:1412.6980.

Kraus, O.Z., & Frey, B.J. (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.

Lazebnik, S., Schmid, C., Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (Vol. 2 pp. 2169–2178).

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

Lee, D.H., Lee, D.W., Han, B.S. (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, R.K., Chu, V.W. (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, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., van der Laak, J.A.W.M., van Ginneken, B., Sánchez, C.I. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis*, *42*, 60–88.

Lowe, D.G. (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* (pp. 281–297). Berkeley: University of California Press.

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)*. Natick: MA.

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

Meijering, E., Carpenter, A.E., Peng, H., Hamprecht, F.A., Olivo-Marin, J.C. (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. 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.

Ni, D., Chui, Y.P., Qu, Y., Yang, X.S., Qin, J., Wong, T.T., Ho, S.S.H., Heng, P.A. (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, D.M., Goldberg, I.G. (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, J.M.S. (1970). Object enhancement and extraction. In *Picture Processing and psychopictorics* (pp. 75–149). New York: Academic Press.

R Core Team (2016). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, 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. (2007). Histología del sistema nervioso del hombre y de los vertebrados. CSIC Madrid reprinted in.

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

Sáez, J.A., Luengo, J., Stefanowski, J., Herrera, F. (2015). SMOTE-IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. *Information Sciences*, *291*, 184–203.

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

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

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

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

Shamir, L., Delaney, J.D., Orlov, N., Eckley, D.M., Goldberg, I.G. (2010). Pattern recognition software and techniques for biological image analysis. *PLOS Computational Biology*, *6*(11), e1000974.

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

Shamir, L., & Tarakhovsky, J.A. (2012b). Computer analysis of art. *Journal on Computing and Cultural Heritage*, *5*(2), 7.

Shapiro, S.S., & Wilk, M.B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, *52*(3-4), 591–611.

Shen, D., Wu, G., Suk, H.I. (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* (pp. 173–186). Boston: Springer.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. Computing Research Repository arXiv:1409.1556.

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

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

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

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

Strobl, C., Hothorn, T., Zeileis, A. (2009). A new, conditional variable importance measure for random forests available in the party package. *The R Journal*, *1*(2), 14–17.

Tajbakhsh, N., Shin, J.Y., Gurudu, S.R., Hurst, R.T., Kendall, C.B., Gotway, M.B., 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, A.E. (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, M.D., Tan, S.S., Gunnersen, J.M. (2007). Automated analysis of neurite branching in cultured cortical neurons using HCA-Vision. *Cytometry Part A*, *71*(10), 889–895.

Vapnik, V.N. (1998). *Statistical learning theory*. New York: Wiley.

Vapnik, V.N. (1999). *The nature of statistical learning theory*. New York: Springer-Verlag.

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

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

Wickham, H. (2009). *Ggplot2: Elegant Graphics for Data Analysis*. New York: Springer.

Wu, C., Schulte, J., Sepp, K.J., Littleton, J.T., 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, S.T.C. (2012). Concise review: a high-content screening approach to stem cell research and drug discovery. *Stem Cells*, *30*(9), 1800–1807.

Yang, J., Yu, K., Gong, Y., Huang, T. (2009). Linear spatial pyramid matching using sparse coding for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1794–1801).

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

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.