Neural network fast-classifies biological images using features selected after their random-forests-importance to power smart microscopy.

Maël Balluet[a,b], Florian Sizaire[a,g], Youssef El Habouz[a], Thomas Walter[c,d,e], Jérémy Pont[b], Baptiste Giroux[b], Otmane Bouchareb[b], Marc Tramier[a,f], Jacques Pecreaux[a,*]

[a] *CNRS, Univ Rennes, IGDR - UMR 6290, F-35043 Rennes, France*
[b] *Inscoper SAS, F-35510 Cesson-Sévigné, France*
[c] *Centre for Computational Biology (CBIO), MINES ParisTech, PSL University, F-75272 Paris, France*
[d] *Institut Curie, F-75248 Paris, France*
[e] *INSERM, U900, F-75248 Paris, France*
[f] *Univ Rennes, BIOSIT, UMS CNRS 3480, US INSERM 018, F-35000 Rennes, France*
[g] *Present address: Biologics Research, Sanofi R&D, F94400 Vitry sur Seine, France*

Abstract

Artificial intelligence is nowadays used for cell detection and classification in optical microscopy, during post-acquisition analysis. The microscopes are now fully automated and next expected to be smart, to make acquisition decisions based on the images. It calls for analysing them on the fly. Biology further imposes training on a reduced dataset due to cost and time to prepare the samples and have the datasets annotated by experts. We propose here a real-time image processing, compliant with these specifications by balancing accurate detection and execution performance. We characterised the images using a generic, high-dimensional feature extractor. We then classified the images using machine learning for the sake of understanding the contribution of each feature in decision and execution time. We found that the non-linear-classifier random forests outperformed Fisher's linear discriminant. More importantly, the most discriminant and time-consuming features could be excluded without any significant loss in accuracy, offering a substantial gain in execution time. It suggests a feature-group redundancy likely related to the biology of the observed cells. We offer a method to select fast and discriminant features. In our assay, a 79.6 ± 2.4 % accurate classification of a cell took 68.7 ± 3.5 ms (mean ± SD, 5-fold cross-validation nested in 10 bootstrap repeats), corresponding to 14 cells per second, dispatched into 8 phases of the cell cycle using 12 feature-groups and operating a consumer market ARM-based embedded system. Interestingly, a simple neural network offered similar performances paving the way to faster training and classification, using parallel execution on a general-purpose graphic processing unit. Finally, this strategy is also usable for deep neural networks paving the way to optimising these algorithms for smart microscopy.

1. Introduction

The optical microscope, after centuries as an advanced optical device, underwent significant evolutions during the last decades to become the motorised system now controlled by electronic signals. Its variegated modalities make it an unparalleled tool to investigate the living (Nketia et al., 2017). Beyond academic research, it can automatically image samples in large series, together with the appropriate robots, paving the way to live-cell high content screening (HCS) based on phenotypes (Esner et al., 2018; Peng, 2008; Sbalzarini, 2016; Chen et al., 2018). However, the analysis of this data flood is performed posteriorly to the acquisition, limiting the information extracted (Singh et al., 2014). A smart microscope, able to modify the imaging strategy in real-time by analysing images on the fly, is required to increase the number of images interesting for the biological question (so-called qualified images)(Scherf and Huisken, 2015). By autonomously acquiring rare objects and elusive events, it will not only ease basic-research imaging by saving fastidious searching and waiting for a cell of interest at the right stage but also increase the content of interest in HCS by selecting qualified images, up to become a standard tool of precision medicine similarly to next-generation sequencing (Hamilton et al., 2014; Leopold and Loscalzo, 2018; Klonoff, 2015; Djuric et al., 2017). The current systems that perform imaging and analysis in tandem alternate acquiring images and analysing them (Conrad et al., 2011; Tischer et al., 2014). We recently achieved efficient microscope driving (Sizaire et al., 2020; Roul et al., 2015) and here investigate how to perform the real-time object's classification to feedback to it.

Searching rare and brief events is a booming field beyond the sole microscopy. They often carry significant information about normal or abnormal processes in a broad range of applications (Ali et al., 2015; Kaushal et al., 2018). Radiologists use such algorithms to assist the medical-doctor diagnosis interactively, calling for reduced image processing delay

(Chartrand et al., 2017). Along a line more demanding of real-time processing, video can be processed to recognise the human activities, in particular, risky or abnormal situations like intrusions or dangerous behaviours (Bobick and Davis, 2001; Zhang et al., 2017; Sargano et al., 2017). Similarly, it can support detecting and diagnosing faults in construction or process industries (Koch et al., 2015; Duchesne et al., 2012). These situations may result in costly damages, human injuries and require rapid detecting through real-time analysis. We here used a similar approach to detect rare and transient events in living biological samples.

Very archetypal to these events is the anaphase of cell division when the sister chromatids are separated to be equally distributed to each daughter cell. In human cells, it lasts a few minutes or less in contrast with a cycle of 15 to 30 hours (the repetition time of mitosis) (Moran et al., 2010). Cell division has received strong attention in fundamental research as its mechanisms are only partially known, as well as in applied research in particular to develop cancer therapies (Manchado et al., 2012; Florian and Mitchison, 2016; McIntosh, 2017; Rieder and Khodjakov, 2003; Cireşan et al., 2013). Indeed, the spindle assembly checkpoint (SAC) secures the transition to anaphase by ensuring a correct attachment of the chromosomes, essential to their equal partitioning to daughter cells. However, this checkpoint may fail to detect errors or slip, paving the way to cancer (Potapova and Gorbsky, 2017; Sivakumar and Gorbsky, 2015). Unfortunately, the current techniques to investigate these phenomena are invasive, as blocking cultured (human) cells for a few hours at the entry in mitosis by drugs similar to antimitotic ones used in cancer therapies (Banfalvi, 2017). Doing so lets most of the cells reach the threshold of mitosis before the experimenter releases the block to observe all cells undergoing mitosis in a synchronised fashion. Although instrumental, this technique is perturbative, and we propose here to leap towards superseding it by detecting mitosis when they occur rather than triggering them artificially. Along an applied line, targeting mitosis is a cornerstone to designing drugs used in chemotherapy (Manchado et al., 2012). It implies the ability to fast screen across a library of compounds and quickly assess defect in mitosis and particularly deadlocked mitosis due to unsatisfied SAC (McIntosh, 2017). Along a medical line, detecting mitosis in patient tissues is classically used for diagnosis as in breast cancer (Wang et al., 2014; Hamidinekoo et al., 2018; Veta et al., 2015). Overall, it makes the automated detection of early anaphasic cells a highly relevant application case.

Beyond these applications, both fundamental and applied cell microscopy would need an approach to detect rare and short events to instruct the microscope some specific acquisition conditions. Such a system should exhibit three main specifications: perform fast enough to achieve real-time detection; being adaptive to a wide variety of problems (cell types, labellings or events of interest, e.g.) without re-programming or re-optimising; achieve this adapting (training) over a reduced exemplar dataset. While some dedicated image processings allow post-processing of the data and identification of the hits in high content screening (Wollmann et al., 2017; Fillbrunn et al., 2017; McQuin et al., 2018), each application resulted from a dedicated development. Furthermore, suitable performance often requires a detailed and long optimisation of the specific program. In particular, algorithms were developed to classify mitotic cells in distinct stages, along time and in live samples (Harder et al., 2009; Held et al., 2010; Conrad et al., 2011). These classifiers may, however, turn to be too slow for real-time since we aimed to acquire and classify images on the fly concurrently. Furthermore, these algorithms are specialised to a given biological situation while we aim at developing a single software adapted to a broad range of applications, i.e. generic. These latter approaches had used to result in poor classification as they involved one or a few generic features (Sbalzarini, 2016). In the last decades, the emergence of machine learning has been a real game-changer and allowed both generic and accurate analysis, and paved the way to new experiments (Moen et al., 2019; Nagao et al., 2020; Singh et al., 2014; Sommer and Gerlich, 2013; Sbalzarini, 2016; Nketia et al., 2017). Along that line, we here used a wide variety of features found in the library WND-CHARM (Orlov et al., 2008). Key to perform accurate and fast detecting was to select a subset of these features and combine them into an efficient discriminator. It enabled to optimise the code once and for all, without editing it again. The specifics of the application were encoded into a statistical model. Machine learning approaches addressed this need and could be trained easily to each application through a numerical optimising onto a set of labelled images. In contrast to deep learning, it enabled to identify important features and even manually manipulate their selected subset to improve execution time. We then embedded this classifier and adapted it to the case through its training to ensure real-time execution, paving the way to the autonomous microscope (Balluet et al., 2020). In this article, we proposed a strategy to optimise the selection of features of interest under the constraint of both accurate classification and fast performing. It implies to select features both quick to execute and discriminant. Amazingly, we found that highly discriminant features could be excluded, provided enough other features were available, without any loss in classification accuracy and with a strong gain in execution time on an ARM embedded system.

2. Materials and methods

2.1. Image database

We built a first image database (termed *CellCognition*) from the CellCognition (Held et al., 2010) software demonstration images. It is composed of wide-field fluorescence time-lapses of human Hela Kyoto cells, expressing histone H2B and α-tubulin markers, which revealed the chromosomes and the microtubules respectively. Images are acquired at three different positions with a 20x dry objective and taken with a time interval of 4.6 min. Each field contained 206 images of 1392×1040 pixels, including multiple cells. The corresponding annotations classified the cells between 8 classes, including the six mitotic phases and indicated the centre of the object(Held et al., 2010). We built a database of 71×71 pixels vignettes corresponding to classified cells extracted from the fields. Cells exemplary of each class are presented in Fig. 1a. We removed multiple instances of the same cell appearing at different stages and thus in distinct classes. We also discarded randomly chosen vignettes to equilibrate the dataset. We obtained 159 vignettes altogether, specifically 20 per class, except apoptosis showing 19 vignettes. This low number of cells was in line with our application in cell biology since large training sets are not achievable for experimental reasons.

To demonstrate that our classification method is generic, we used a second database, termed *mitocheck*. It is based on the class definitions published in (Neumann et al., 2010). With respect to this paper, we significantly increased the number of samples in each class. In addition, we added a second artefact class ”Focus”. For annotation, we preselected experiments that showed phenotypes according to the analysis in (Neumann et al., 2010), and we manually annotated individual nuclei in these movies without looking at the initial classification. For the dynamic phenotypes, such as prometaphase and metaphase, we sometimes used the time information to decide, in accordance to the procedure in (Neumann et al., 2010). In total, we annotated 5151 nuclei. It was composed of wide-field fluorescence time-lapses of Hela Kyoto cells, expressing chromatin GFP marker but no α-tubulin, acquired with a 10x dry objective on Olympus ScanR. Several mitotic phases and defect phenotypes were observed. After equilibration, we obtained 1100 vignettes of 64×64 pixels dispatched up into 11 classes (100 per class) (see Fig. 1b).

2.2. Features extraction

WND-CHARM is a multi-purpose image classifier developed in C++, generating a high-dimension features-vector and using Weighted Neighbour Distances for classification (Orlov et al., 2008). We used it to extract edges and objects statistics, multi-scale histograms, four first moments on images subdivision, polynomial decompositions (Chebyshev, Chebyshev-Fourier and Zernike), texture information (Haralick, Tamura and Gabor textures) and Radon transform statistics. In a first step, a transform like Fourier or wavelet could be applied to the raw vignettes to produce a so-called feature precursor, which is an image (Fig. 2c, right), on which statistics are extracted (Fig. 2c, left). Technically speaking, we gather in these statistics some computations that could involve the image (Otsu thresholding for Otsu object statistics case, e.g.) before computing scalar values as statistics (the bright segmented region area in Otsu statistics, e.g.). All features were scalar and were gathered in a 1025-valued vector. Importantly, we performed some optimisation of the WND-CHARM library to reduce its execution time.

2.3. Estimating the computing time of features extraction

To estimate the computing time of a single WND-CHARM feature, we computed it over the single-cell vignettes obtained for instance from CellCognition database, running on an NVIDIA Jetson AGX Xavier embedded system. We then averaged the results over the vignettes of all the dataset. In particular, we ensured that the execution was sequential on the CPU of the embedded system, without using parallelism. When it comes to estimating the computing time of multiple features, we noticed that the features were not independent. Indeed, for a given group of features, they all correspond to statistics computed on the same *feature precursor*. This latter was either the raw image or a transform computed from it. Several image-transforms could be composed together successively (Fig. 2c). Notably, the major part of computing time was spent in getting such feature precursors. We thus considered that features were computed by group deriving from the same-precursor. We thus summed up the execution times of all of them within a group, to get the group execution-time. For instance, in the case of the features based on the Haralick texture, the feature-precursor computation took 90% to 99% of the whole computing time (Fig. 2a).

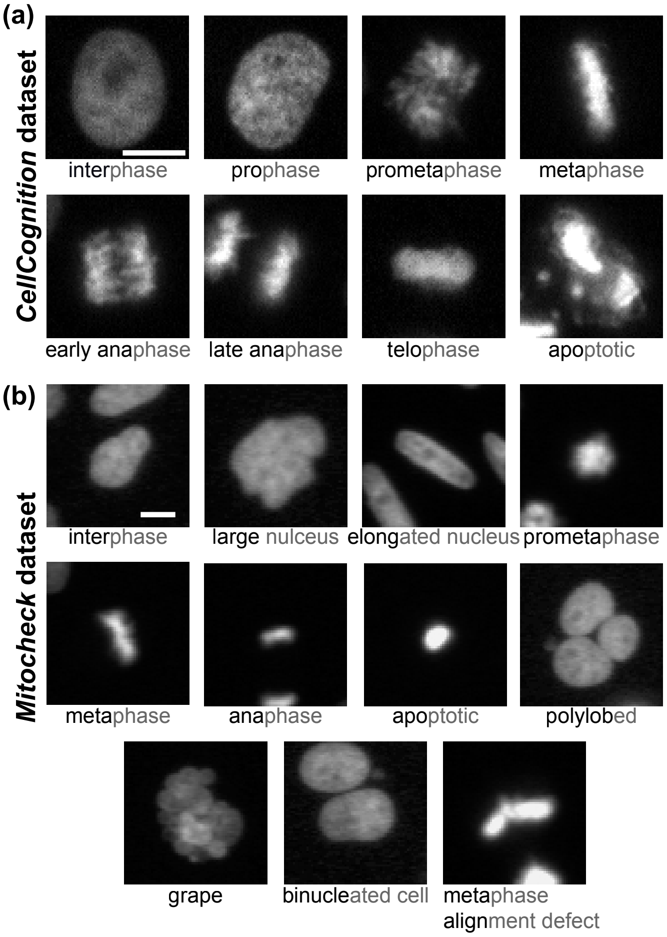
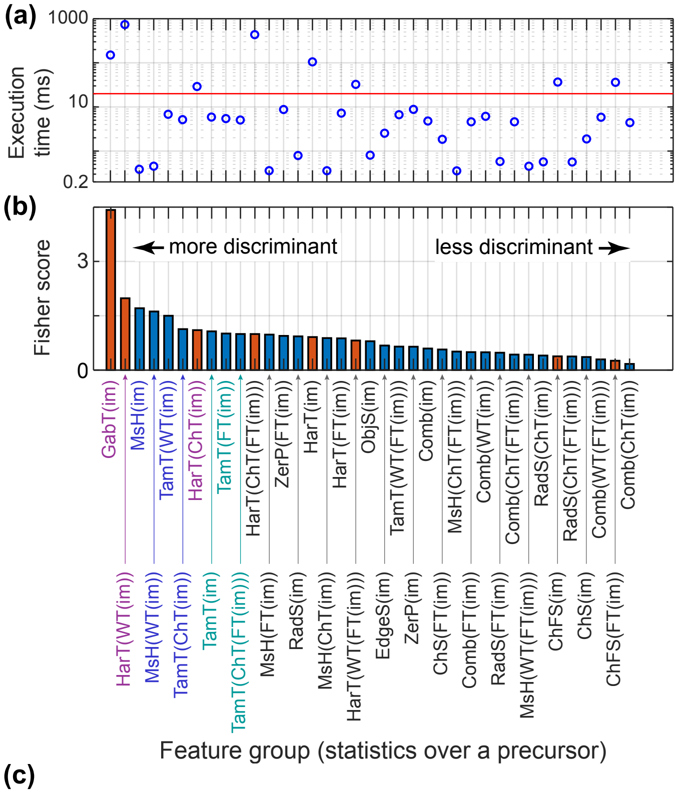


Figure 1: **Datasets used during numerical experiments.** **(a)** Exemplar vignettes upon 71×71 pixels cropping images from the CellCognition database. **(b)** Exemplar vignettes similarly cropped and extracted from the mitocheck database. Class names were abbreviated and written in black font, while the full name appears in grey. They correspond either to cell division phases or specific defects: cells whose nucleus display an elongated, polylobed or grapefruit-like shape, and nuclei reminiscent of apoptotic cells, binucleated ones (usually following a cytokinesis defect) or cells having an issue in aligning the chromosomes during metaphase, usually due to lagging chromosomes or multipolar spindles. A scale bar indicates 10 µm in the first frame, and all vignettes within a dataset are on the same scale.



(c)

Feature statistics	Abbrev.
Chebyshev-Fourier Statistics	ChFS(*TI*)
Chebyshev Statistics	ChS(*TI*)
Zernike Polynomials	ZerP(*TI*)
Edge Statistics	EdgeS(*TI*)
Gabor Textures	GabT(*TI*)
Otsu Object Statistics	ObjS(*TI*)
First Four Moments	Comb(*TI*)
Haralick Textures	HarT(*TI*)
Multi-scale Histogram	MsH(*TI*)
Tamura Textures	TamT(*TI*)
Radon Transform Statistics	RadS(*TI*)

Transformed image (*TI*)	Abbrev.
None (raw image)	im
Fourier Transform	FT(im)
Chebyshev Transform	ChT(im)
Wavelet Transform	WT(im)
Chebyshev of Fourier Transforms	ChT(FT(im))
Wavelet of Fourier Transforms	WT(FT(im))

Figure 2: **Feature-groups execution time and Fisher's score.** **(a)** Execution time summed up over feature groups, estimated on an NVIDIA Jetson AGX Xavier embedded system, and **(b)** the corresponding Fisher's score averaged over the same feature groups (see Methods, §2.3 and §2.4). **(c)** (left) Depicts the feature groups by statistics, computed over (right) various feature precursors, i.e. the raw image or its transform. Red bars highlight the feature groups displaying an execution time greater than 20 ms. A red line depicts this threshold time in panel (a). Feature-group labels written with colour depict the ones kept for assay using Fisher's linear discriminant (see §3.2), specifically the purple and dark blue when considering all feature-groups and the dark and light blue when excluding computationally intensive groups. . When excluding computationally intensive features, the blue ones are also used to complement to 7 groups. CellCognition dataset was used (see Methods §2.1).

2.4. Estimating the fisher score of features and feature-groups

The contribution of a feature to the classification was estimated using Fisher's score (Orlov et al., 2008; Bishop, 2006). For the feature groups as defined above (see §2.3), we averaged the score of the features over the whole group. Because various statistics within a group might display different scores, such an averaging strategy will favour groups with a majority of well-discriminant features.

3. Results

3.1. Classifying based on a single feature was not accurate enough.

We set to automatise the microscope by processing images on-the-fly and feeding back to the microcontroller that drove the microscope and its attached devices. To ensure real-time processing, we embedded the processing on a microcontroller as it was designed to execute only one or a few dedicated functions, with real-time constraints, by opposition to a general-purpose computer. It is widely used in fields requiring real-time applications and machine learning algorithms are now available on these platforms. To support the development, we classified mitotic images within 8 classes using the CellCognition example set (Held et al., 2010; CellCognition, 2010) (see Methods §2.1) and in particular detected the transition from metaphase to anaphase. We reckoned that the choice of the features could be essential for performance and precision. Therefore, we used the WND-CHARM framework that encompassed a large variety of features (Orlov et al., 2008). First, aiming at fast processing, we asked whether a single feature could be sufficient. We computed Fisher's score of each feature (see Methods §2.4) and found that the most discriminant one was the area of the segmented image with an Otsu static threshold (Otsu, 1979). The area of Otsu object was highly efficient to discriminate interphase from mitosis. However, this feature was unable to correctly detect anaphase onset since it was mostly sensitive to the surface of the bright objects (Fig. S1) It called for a multi-feature approach.

3.2. Selecting an optimal set of feature-groups using Fisher's linear discriminant.

Computing all the features offered by the WND-CHARM library for a 71 × 71 vignette on the ARM microcontroller, was too computationally intensive for several features (Fig. 2a), thus incompatible with real-time analysis. We foresaw that a small number of features could be combined into a discriminant score, sufficient to discriminate the different mitotic stages. To do so, we opted for a machine learning approach, to help to delineate important features, rather than a deep learning approach. Such an *a priori* choice appeared the most fitted to our lack of large training set and need for fast computation. Indeed, deep-learning-network convolutional layers are computationally intensive, and while optimisation strategies are available for embedded instances like pruning or quantisation (Jacob et al., 2018; Molchanov et al., 2016), it requires a large training set. We first opted for a linear machine-learning algorithm, specifically the Fisher's linear discriminant (Fisher, 1936; Duda and Hart, 1973). Indeed such a kernel method, because linear, promised short execution times and was successful in similar problems (Muller et al., 2001; Belhumeur et al., 1997; Liyang et al., 2005; Chiang et al., 2000).

We tested Fisher's linear-discriminant classification using the CellCognition dataset (see Methods §2.1), in particular, 80% of the vignettes for training and 20% for testing through a k-fold cross-validation process ($k = 5$). We ranked the feature groups by decreasing Fisher's score (see Methods 2.4). To avoid overfitting, we limited the number of features considered to less than the number of training images. We included the feature-groups in descending fisher score up to that limit. It led to the 7 feature-groups (named in purple and dark blue Fig. 2b) (Shaikhina et al., 2015; Kourou et al., 2015; Foster et al., 2014). To find an optimal number of features, we further pruned the feature groups by removing the least discriminant one iteratively until it harmed the overall classification. In further details, we assessed the quality of the classification through the area under the ROC curves (AUC) averaged over the eight classes of our dataset, a classical metric in machine learning (Fawcett, 2006). We measured the maximum AUC when removing the groups and conserved as many groups as needed so that the AUC is not decreased by more than 0.005 from this maximum. It could be achieved without re-training, taking advantage of the linearity (Fig. 3a). Such a reduction of the feature-groups number, beyond performance consideration, is essential to cope with the scarcity of labelled images, a commonplace in microscopy for biology and medicine. We obtained the best classification by considering only 2 groups, namely Gabor textures and Haralick calculated from Wavelet transform ones (Fig. 2b, Fig. 3a, red curve and arrowhead). While the classification could be satisfactory with a global accuracy of 78.0% (Fig. 3c and 3d), the execution time, 890 ms, was incompatible with the on-the-fly classification (Fig. 3b).

We noticed that the most discriminant feature-groups displayed a score neatly larger compared to the others (Fig. 2b). However, the two most discriminant groups used for optimal classification were too computationally intensive for our application. We reckoned that they could be removed, keeping a reasonable classification accuracy. In a broader take, we censored all the feature groups, which required more than 20 ms to be computed (Fig. 2a, red line). We again considered 7 feature-groups only to prevent overfitting (named in ligh and dark blue Fig. 2b). We then selected a subset of the groups, by excluding the least discriminant ones, as explained above. We obtained the best classification using 3 feature groups (Fig. 3a, blue curve): multi-scale histograms calculated from raw vignettes, multi-scale histograms from Wavelet transform of the vignettes and Tamura textures from Wavelet transform. However, while the transition from metaphase to anaphase was still correctly detected, the confusion matrix and the ROC curves, on early and late mitotic phases, showed a clear degradation of the classification (compare Fig. 3ef with Fig. 3cd). Overall, the accuracy read 52.2% and class-averaged AUC 0.842 for the three-groups case, compared to 78% and 0.954, respectively, for the two-groups case including the computationally-intensive features. Using three non-computationally-intensive feature-groups only partially compensated the lack of the two most-discriminant groups and resulted in classification so inaccurate that it could not fit our applicative needs. However and importantly, the feature extraction took only 9 ms in the three-groups case, compared to 890 ms in the two-groups one, in line with embedded on-the-fly processing.

Overall, using multiple feature-groups in classification needed a tedious balance between accuracy and execution time, unworkable by a linear machine learning approach. However, we observed a partial redundancy of the features in distinct groups. Importantly, we noticed that the classifying itself took a negligible time, provided that the features were already computed. It called for using non linear classification method to combine the features at the expense of computing time.

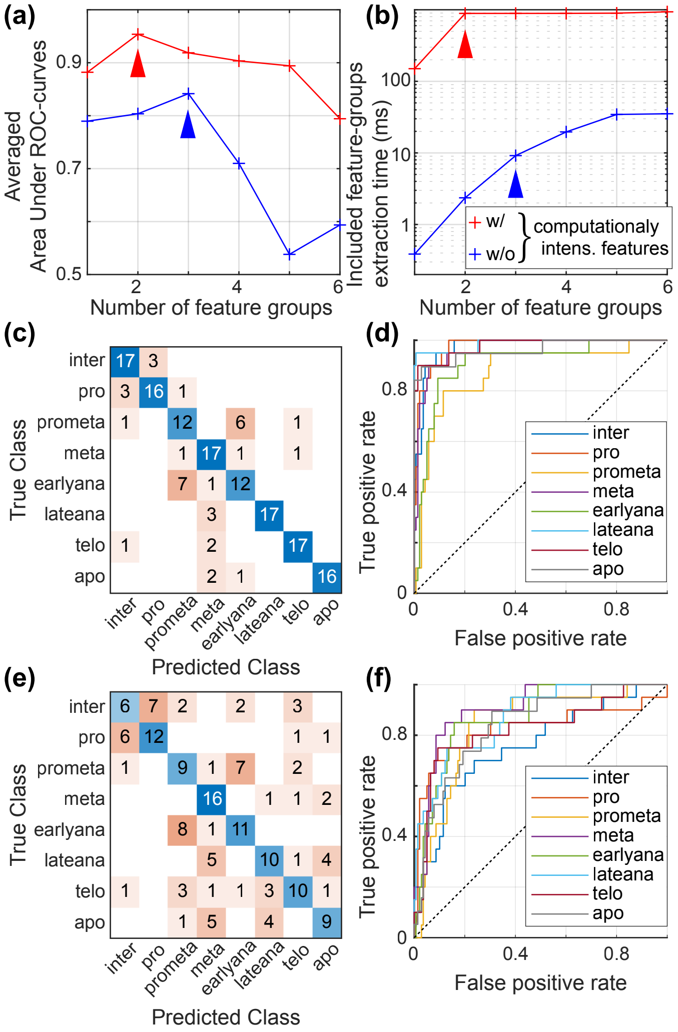


Figure 3: **Classification using Fisher's linear discriminant.** **(a)** Area Under Curve (AUC) averaged over the classes and **(b)** execution time for extracting the feature-groups included in the classification, both versus the number of feature-groups used in classification, including (red curve) all available features or (blue curve) only groups with an execution time below 20 ms (not computationally intensive). Arrowheads of the corresponding colour depict their optimal number (see §3.2). **(c)** and **(e)** report the corresponding confusion matrix for these two-groups (Gabor textures and Haralick over wavelet transform ones), and three-groups (multi-scale histograms over raw vignettes, multi-scale histograms over wavelet transform, and Tamura textures over wavelet transform) optimal cases, respectively, and **(d)** and **(f)** are the corresponding ROC curves. Class names are abbreviated after Fig. 1a. We used the 5-fold cross-validation over the CellCognition dataset (see Methods §2.1).

3.3. Revealing the feature-groups redundancy using random forests.

We pursued searching for a feature-group subset, fast enough to be used in our real-time application and using a non-linear classifier. We set to use a decision-tree based method as it copes well with the large number of features coupled to the reduced training dataset. We specifically chose the random forests algorithm (Tuv et al., 2009; Breiman, 2001). It is a machine learning algorithm based on an ensemble of decision trees, that furthermore internally selects the most discriminant features, in line with our goal to use a subset of feature groups. Compared to other non-linear methods, random forests, by this selection process, better avoids over-fitting problems. Practically, we trained 300 decision trees using curvature test to select the best split predictor (Loh and Shih, 1997), and we validated this model using k-fold cross-validation with $k = 5$. We empirically determined the number of trees, measuring that more than 300 trees would not improve the classification accuracy (Fig. S2). We first performed the classification using all the 1025 features, and the algorithm training converged. The global accuracy read 81.8% and AUC 0.974, which is slightly better compared to Fisher's linear discriminant. All the classes were recovered at least as accurately or better by the random forests (Fig. S3). This encouraging result confirmed the suitability of the random forests to our problem. However, extracting all the features from the image remained too computationally intensive for our application.

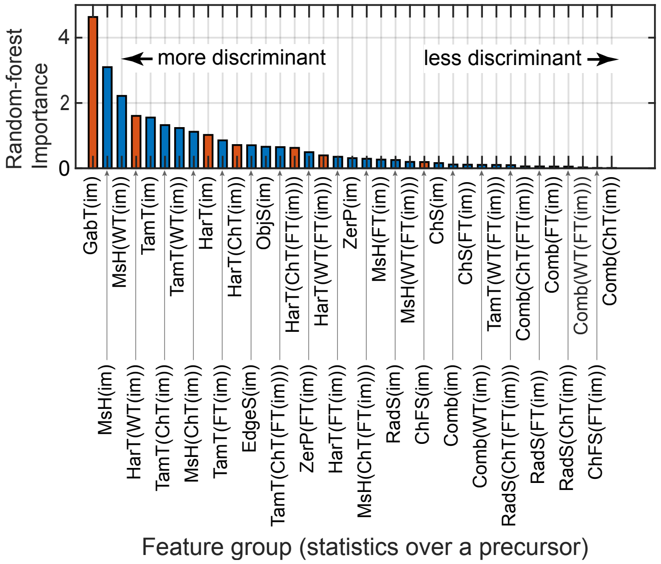


Figure 4: **Random forests using all the 1025 features** was trained and tested over 20% of the dataset, and we retrieved the importance of each feature-group (see main text). Red bars highlight the feature groups displaying an execution time greater than 20 ms. The execution times are reported in Fig. 2a. The feature groups are described in Fig. 2c.

The random forests offer a mechanism to assess the *importance* of each feature in the decision (Breiman, 2001). In a nutshell, it corresponds to the difference of the rate of misclassification of the "out-of-bag" samples (i.e. the labelled images not used for training a given tree because of the internal bootstrap mechanism), when randomly shuffling the values of a given feature. Hence, the importance of features is directly related to the performed classification, in contrast to Fisher's discriminant criterion used above. We summarised the feature importances as previously, by taking the average over their values within a group. We then averaged over the five forests generated in the k-fold validation process (Fig. 4).

We aimed to perform a fast and precise classification, thus as for the Fisher's linear discriminant, we removed the least important feature groups and computed the random-forests importances again over the training vignettes and averaged over 5-fold cross-validation. Unlike the case of Fisher's discriminant, the approach was iterative, requiring a re-training upon each change of the feature-group subset. The classification quality, measured by the mean AUC, decreased when using less than 8 groups (Fig. 5a, red curve). These 8 groups represented 147 features out of 1025. The global accuracy obtained with 8 groups was 75.5% and the mean AUC 0.970, so very close to the results obtained with all features, suggesting that we could reduce the execution time by excluding features, without decreasing the classification quality (Fig. S4).

When it came to applying random forests to on-the-fly classification, we yet noticed that some computationally intensive feature-groups (red in Fig. 4) displayed large importance like Gabor-on-raw-image and Haralick-on-wavelet-transform textures. On the ground of the trend obtained using Fisher's linear discriminant, we excluded the groups, which execution time was greater than 20 ms. We then iteratively removed the least-important features until it degraded the classification (Fig. 5a, blue curve). It showed an optimum with 12 feature groups (264 features out of 1025). In that latter case, AUC read 0.977 and global accuracy 83.6%, which was again very similar to the case using all 1025 features. We also obtained similar confusion matrix and the ROC curves (Fig. 5cd) but the execution time was considerably reduced (divided by more than 50). This result validated the feasibility of our embedded classification by reducing the number of features and censoring the computationally intensive ones (Fig. 5b).

We then took a closer look at the feature importance when reducing the number of features, to get clues of this compensating mechanism. We compared the importance of the 12 feature-groups used in the optimised classification, with the importance of the same groups upon classifying over all the features (Fig. 5e). We observed that the importance of these groups increased. It is suggestive of redundancy of the features, at least in their significance for the present classification if not in general. Notably, it was proposed that the random forests spread the importance among the redundant features. As expected, compensation of removed redundant features similar to what we observed was seen in other studies (Tuv et al., 2009; Zhao et al., 2019). We here took advantage of this ability of random forests to ensure fast execution on an embedded system, compatible with real-time classification in an automated microscope.

We reckoned that these results represented one particular instance of database equilibration (see §2.1). To test how general was our approach, we used bootstrap to randomly split data into balanced datasets, however without replacement (no duplicated image). We performed ten bootstrap iterations. Within each of them, we performed a 5-fold validation and repeated the optimisation process as described above, excluding computationally intensive feature-groups. On average, 12 feature-groups were the optimal balance between performance and accuracy (precisely 11.6 ± 2.4, mean ± standard deviation), as found previously, although variations of a few units were observed. We

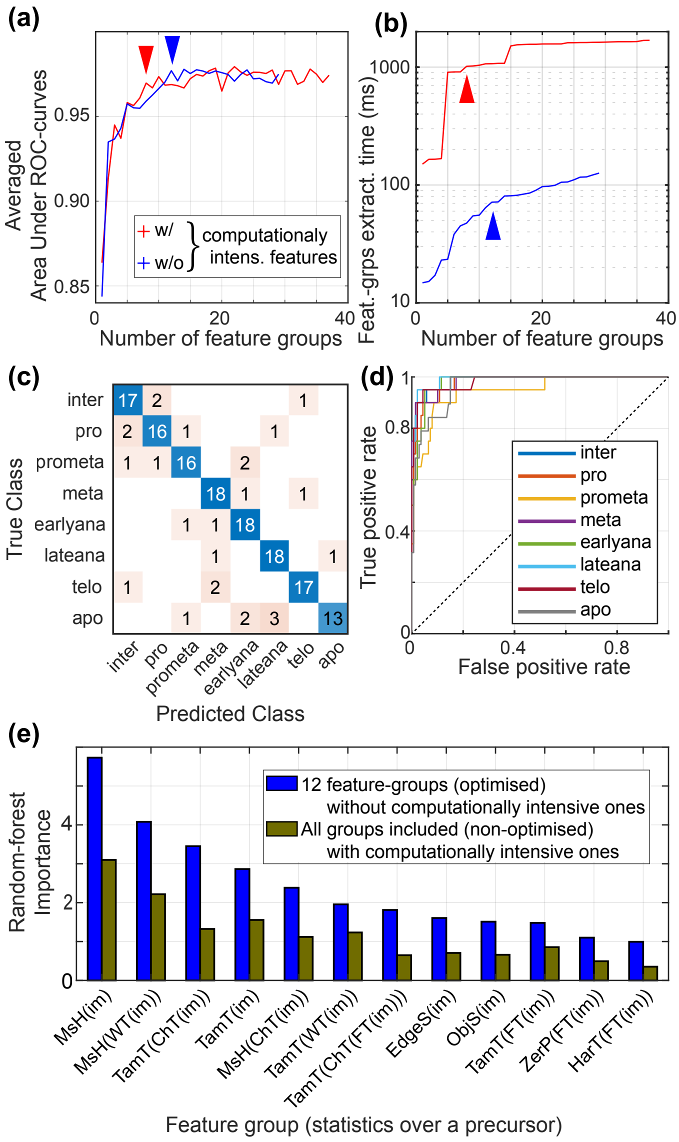


Figure 5: **Random-forests classification using a subset of feature-groups.** **(a)** Area Under Curve (AUC) averaged over the classes and **(b)** execution time for extracting the feature-groups included in the classification, both versus the number of feature-groups used in classification, including (red curve) all available features or (blue curve) only feature groups with an execution time below 20 ms (not computationally intensive). Arrowheads of the corresponding colour depict their optimal number (see §3.3). **(c)** Random forests importance (blue) in the twelve-groups case, optimal when excluding computationally intensive feature-groups, and (brown) the all-feature-case (non optimised, reported Fig. 4 and S3). We averaged over the 5-fold cross-validation and used the CellCognition dataset (see Methods §2.1). **(d)** The confusion matrix and **(e)** the ROC curves averaged, using the 5-fold cross-validation in the twelve-feature-groups optimal case without computationally intensive feature-groups in the optimal case using the CellCognition dataset (see §2.1). Class names are abbreviated after Fig. 1a.

observed a 79.6 ± 2.4 % accurate classification lasting overall (feature extracting and vignette classification) 68.7 ± 3.5 ms. Furthermore, the variations of classification accuracy and total execution time between bootstrap-iterations were reduced (Fig. S5). However, the execution time varied mildly between each iteration of the bootstrap, in particular, because the selected feature-groups changed marginally in each bootstrap iteration. Indeed, we observed that 11 feature-groups are present in all these instances, while 1 is drawn in four other groups (black and blue text, Tab. 1). The low difference between bootstrap iterations showed the reproducibility of our method when different training subsets are used.

Feature groups	**In *n* bootstrap iter.**	**Feature groups**	**In *n* bootstrap iter.**
EdgeS(im)	10	MsH(ChT(im))	10
MsH(WT(im))	10	MsH(im)	10
ObjS(im)	10	*TamT(ChT(FT(im)))*	10
TamT(ChT(im))	10	*TamT(FT(im))*	10
TamT(WT(im))	10	*TamT(im)*	10
ZerP(FT(im))	10		
HarT(FT(im))	5	MsH(ChT(FT(im)))	3
MsH(FT(im))	1	*ZerP(im)*	1

Table 1: **Bootstrapping random forests optimal feature-groups-number classification** over the *CellCognition* dataset. (black) 11/12 groups were always present in the 10 bootstrap iterations while (blue) the last group was taken among four other groups. The feature groups appearing only in the optimal cases using this dataset compared to *mitocheck* one were italicised (Tab. 2). The feature groups are described in Fig. 2c.

To further confirm this result, we repeated the approach using the second dataset *mitocheck* (see §2.1). In this case, images were classified between 11 classes, with 100 vignettes per class. We followed the same method as above and performed a *k*-fold validation process ($k = 5$) followed by ten bootstrap iterations, randomly spliting data into balanced datasets, however without replacement (no duplicated image). Eight feature groups, excluding the ones which execution time exceed 20 ms, were enough to achieve an optimal classification (Fig. 6ab). All classes were correctly recovered (Fig. 6cd). The feature-groups finally used in classification vary in the different instances of the bootstrap as with the CellCognition dataset without considerably impacting the execution time and the classification quality (Fig. 6). Interestingly, the selected feature-groups are mostly the same as with *mitocheck* dataset (compare Tab. 2 and 2). Overall, it confirmed the robustness of the above procedure used to embed image processing. Interestingly, most of the feature-groups were conserved, suggesting some possible generalisation.

Feature groups	**In *n* bootstrap iter.**	**Feature groups**	**In *n* bootstrap iter.**
MsH(ChT(im)	10	MsH(im)	10
MsH(WT(im))	10	ObjS(im)	10
TamT(ChT(im))	10	TamT(WT(im))	10
MsH(WT(FT(im)))	6	HarT(FT(im))	4
MsH(ChT(FT(im)))	4	MsH(FT(im))	4
TamT(WT(FT(im)))	2		

Table 2: **Bootstrapping random forests optimal feature-groups-number classification** over the *mitocheck* dataset. (black) 6/8 groups were always present in the 10 bootstrap iterations while (blue) the two other groups were taken among five other groups. The feature groups appearing only in the optimal cases using this dataset compared to *CellCognition* one, were italicised (Tab. 1). The feature groups are described in Fig. 2c.

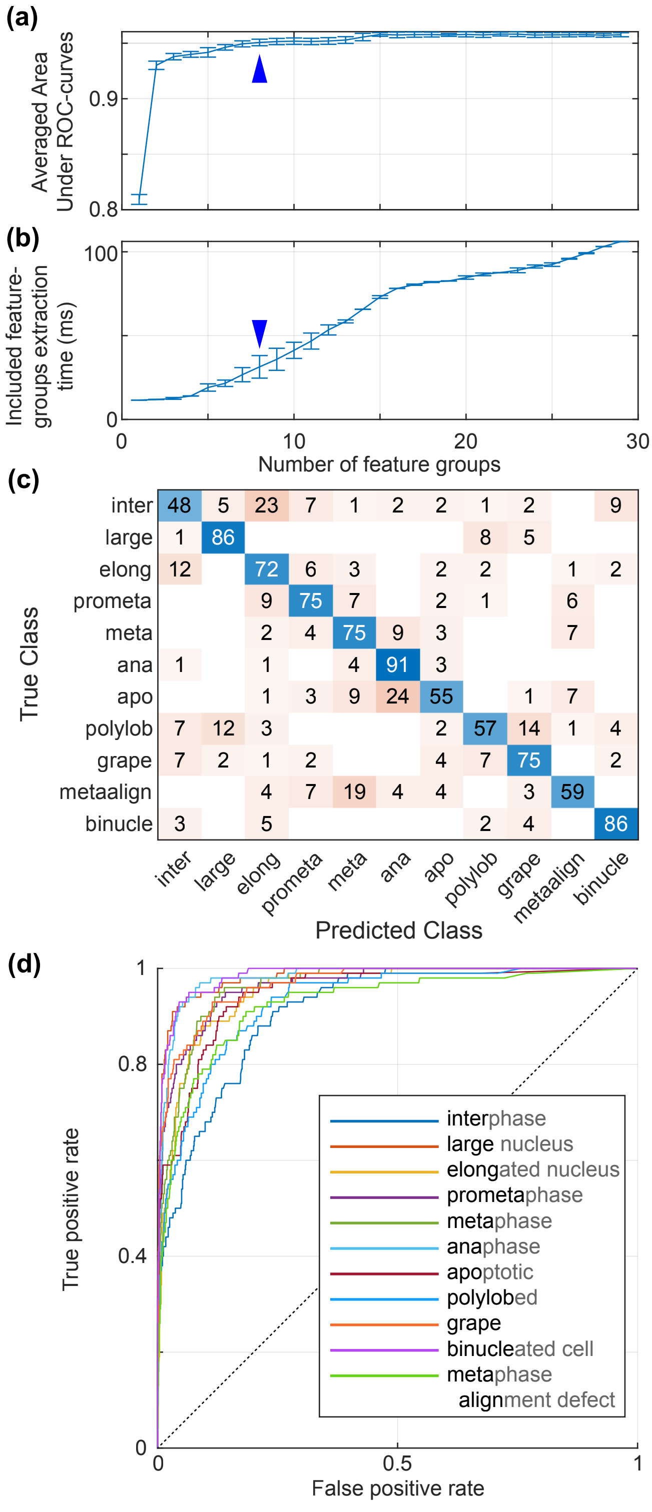


Figure 6: **Bootstrapping optimised random forests over *mitocheck* dataset.** **(a)** The Area Under Curve (AUC) was averaged over the classes, and **(b)** execution time for extracting the feature-groups included in classification was assessed (dependent of the selected feature-groups mildly variable between bootstrap iterations, see §3.3). Both quantities are plotted versus the number of feature-groups used in classification and were computed in the 5-fold cross-validation repeats. This approach was repeated 10 times in the bootstrap approach, where the vignettes included in the balanced dataset were selected differently (see Methods §2.1). We thus obtained the standard deviations reported by the error bars. Arrowheads depict the 8 feature groups optimal case. **(c)** The confusion matrix and **(d)** the ROC curves over the 5-fold cross-validation in a single bootstrap iteration.

In the perspective of classifying vignettes on-the-fly, we had focused on the feature-extraction time by analogy to Fisher's linear discriminant, where this task took the vast majority of the execution time and where classification execution time was insignificant. Therefore, we had only embedded the feature extraction up to now to assess these times when using the random forests. We here reevaluated whether random forests might take significant time as it used decision trees. To do so, we ran the random forests classification on the embedded system using the RTrees module using the OpenCV library (Itseez, 2015). For the sake of simplicity, in a proof-of-concept perspective, we trained the algorithm using OpenCV on the embedded system. However, one could train on a general-purpose computer and embed only the classification. We then assessed the classification performance using 32 test vignettes (20% of the whole *CellCognition* dataset) in the optimal twelve-feature-groups case, excluding computationally intensive ones. With 300 trees, the execution time to classify these vignettes read 89 ± 20 µs (mean ± standard deviation), extrapolated to 27 ± 6 ms for a 300 cells picture. It should be compared to feature extraction over the same picture, lasting 21.6 s. Because feature extraction is performed independently on each vignette, this latter time could be scaled down by parallelising the features extraction since the NVIDIA Jetson AGX Xavier that we used here had 8 CPU cores. Finally, segmenting the image on one CPU core to extract the vignettes took a not noticeable time, about 132 ± 5 ms (mean ± standard deviation) for the whole picture, in comparison to features extracting. In any cases, the classification itself took a lightweight time compared to the feature extraction.

To conclude, we showed that using a non-linear method allowed us to find a much better time-performance compromise than the linear method, to both ensure fast and accurate classification. Therefore, we could envision using our feature-group optimised random forests together with the WND-CHARM features to enslave microscope driving to image classification.

3.4. Neural-network classification also benefits from feature-groups redundancy.

Deep learning is the current paradigm in biological images analysis (Nagao et al., 2020; Moen et al., 2019). We wondered whether the proposed approach discarding highly discriminant features for the sake of rapidity keeping accuracy could be used in that context. We addressed this question in two steps firstly using a neural network as classifier and secondly extracting the features through the convolutional layers of a deep network classifier. Furthermore, fundamental research applications are more demanding about performances, requiring faster cycle time. Indeed, when it comes to studying mitotic events like metaphase-anaphase transition, the dynamics of the components are on the scale of the second or even the tenth of a second (Elting et al., 2018). To reach such fast processing, we could speed up the feature extraction through GPU-parallelisation, although it was out of the scope of the present paper. In such a context, the time spent in the classification itself became as well important. However, because of the high usage of conditional structures in such decision-tree-

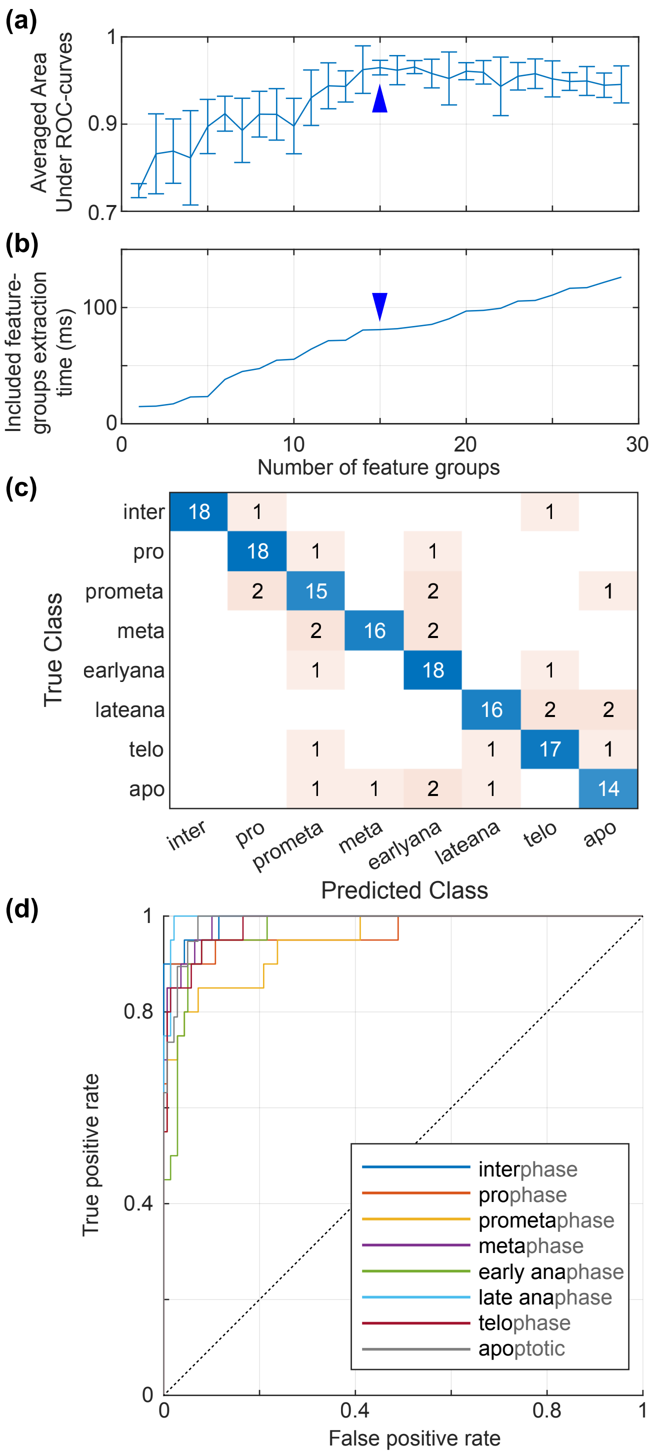


Figure 7: **Bootstrapping optimised neural network over *CellCognition* dataset.** **(a)** The Area Under Curve (AUC) was averaged over the classes, and **(b)** execution time for extracting the feature-groups included in the classification was assessed. Both quantities are plotted versus the number of feature-groups used in classification and were computed in the 5-fold cross-validation repeats. This approach was repeated 20 times in the bootstrap approach, where the vignettes included in the balanced dataset were selected differently (see Methods §2.1). We thus obtained the standard deviations reported by the error bars. Arrowheads depict the fiveteen-feature-groups optimal case. **(c)** The confusion matrix and **(d)** the ROC curves over the 5-fold cross-validation in a single bootstrap iteration. It is noteworthy that no error bar can be computed on execution time as the features are always ranked in the same order of importance (see main text §3.4).

based methods, parallelising the random forests appeared difficult. We reckoned that we could use neural-network-based machine learning over the selected feature-groups. However, this method is more prone to overfitting (Tuv et al., 2009; Bolón-Canedo et al., 2012). In our case, this issue is worsened when the number of features is large, when they are non-independent, correlated or poorly informative as observed in our case. We, therefore, kept using the random forests to select the optimal feature groups, while we used the neural network in ”production context” to perform classification.

We trained a one-hidden-layer network with 64 neurons, using the gradient descent backpropagation algorithm with an adaptative learning rate starting from 0.01, a momentum of 0.1 and a mean squared error (MSE) loss function. To avoid overfitting, an L2 regularisation parameter was added to the loss function with a 0.1 ratio. These training parameters have been experimentally determined. The dataset was divided into three parts: training (70%), validation (20%) and test (10%). The validation subset was used to stop training when the neural network started to overfit. As previously done, we used bootstrap to randomly split the whole data into a balanced dataset (see §2.1), however without replacement (no duplicated image). We performed twenty bootstrap iterations. Within each of them, we used k-fold cross-validation, with $k = 10$. For each instance of the k-fold process, the weights and biases of the networks were initialised to the same values. As previously, we tested the optimal number of feature-groups while excluding the most computationally intensive ones and assuming that group importances were ranked in the same order as in the case of random forests. The optimal classification was found with 15 feature groups and showed comparable accuracy with random forests (Fig. 7cd), reading an AUC of 0.979 and global accuracy of 83.0 %. However, the quality was more variable than with Random forests (Fig. 7a) across the twenty bootstrap iterations. In a broader take, it validated the possibility to use a simple neural network with equal classification quality despite the small training set and a large number of features.

We embedded our neural network using activation functions provided by the OpenCV library. As in the case of random forests, after proper training, we executed the classification of 32 test vignettes. The execution time read 92 ± 15 µs. It could be extrapolated to 28 ± 4 ms for an image containing 300 cells. It has to be compared to the time taken by the random forests to perform a similar task, 27 ± 6 ms. The neural network performed similarly to the random forests when run on CPU. However, it could be further accelerated in the specific case of the neural network using GPU parallelisation. These times remained small compared to the ones needed for feature extraction §(see 3.3). Notably, neural networks used more features groups to perform classification with similar quality than random forests (15 versus 12), which can diminish neural networks interest for execution-time optimisation (Fig. 7b). Conversely, Random forests were much slower than the neural network to be trained: training 300 decision trees using Random forests with 127 samples (80 % of the whole dataset) and 264 features (the 12 best feature-groups) took 21 s on Matlab using one CPU while training our neural network needed between 1 s to 6 s. The need for random forests to rank feature-groups by importance for each new category of images mitigated this advantage of the neural networks. Overall, the neural networks are more promising, but feature extraction will have to be parallelised to realise this pledge.

3.5. Features extracted through a convolutional neural network also show redundancy.

We finally assessed whether the observed redundancy of biological images could be used to discard discriminant features in a deep neural network context. To do so, we built a simple convolutional neural network, including 3 convolutional layers separated by relu activation layers and trained it on the CellCognition images. We retrieved the outputs of the last layer before the fully-connected one and used them as pseudo-features. They are 5184, and we classified them using a 1000-trees random forests algorithm, to avoid over-fitting issues. We again performed 5-fold cross-validation followed by ten bootstrap iterations, randomly splitting data into balanced datasets, however without replacement (no duplicated image). We first included all the pseudo-features and iteratively reduced the number of feature by discarding the less important ones. We obtained an optimal classification with 88 ± 48 pseudo-features (mean ± standard deviation) (Fig. 8a, red curve). Fixing the number of pseudo-features to that number, we observed a larger variability of the pseudo-features included in the set among the bootstrap iterations. We might attribute it to observing single pseudo-features rather than groups; grouping would require a detailed analysis of the network out of the scope of this study. Consistently, among 275 pseudo-features appearing in one optimal set at least out of the ten bootstrap iterations, 18 are present in all sets and 71 in half of them at least. Overall, the optimal classification showed comparable accuracy with random forests, reading an averaged AUC of 0.948 ± 0.006 and global accuracy of 72 ± 2 %.

We then tested whether the compensating mechanism previously observed was applicable here. We thus suppressed the 100 most discriminant pseudo-features, i.e. reported as the most important by the random forests and selected in the optimal pseudo-feature set in at least 4/10 bootstrap iterations above. We repeated a similar analysis and obtained an optimal classification with 108 ± 124 pseudo-features (Fig. 8a, blue curve). Fixing the set to 108 pseudo-features, we observed an equivalent variability of the used pseudo-features as the case with all pseudo-features included: among 303 pseudo-features appearing in one optimal set at least out of the ten bootstrap iterations, 22 are present in all sets and 91 in half of them at least. The optimal classification displayed an accuracy similar to the case with all pseudo-features or with an optimal set among them not discarding important ones (Fig. 8a, compare red and blue curve tails and optimal pseudo-feature number marked by the arrowheads). In further details, we found an averaged AUC of 0.945 ± 0.005 and global accuracy of 71 ± 2 %; the class-wise precisions were similar to the one obtained by classifying WND-CHARM features with random forests (Fig. 8bc). We concluded that pseudo-features based on deep-neural-networks convolutional layers were also redundant, allowing the most

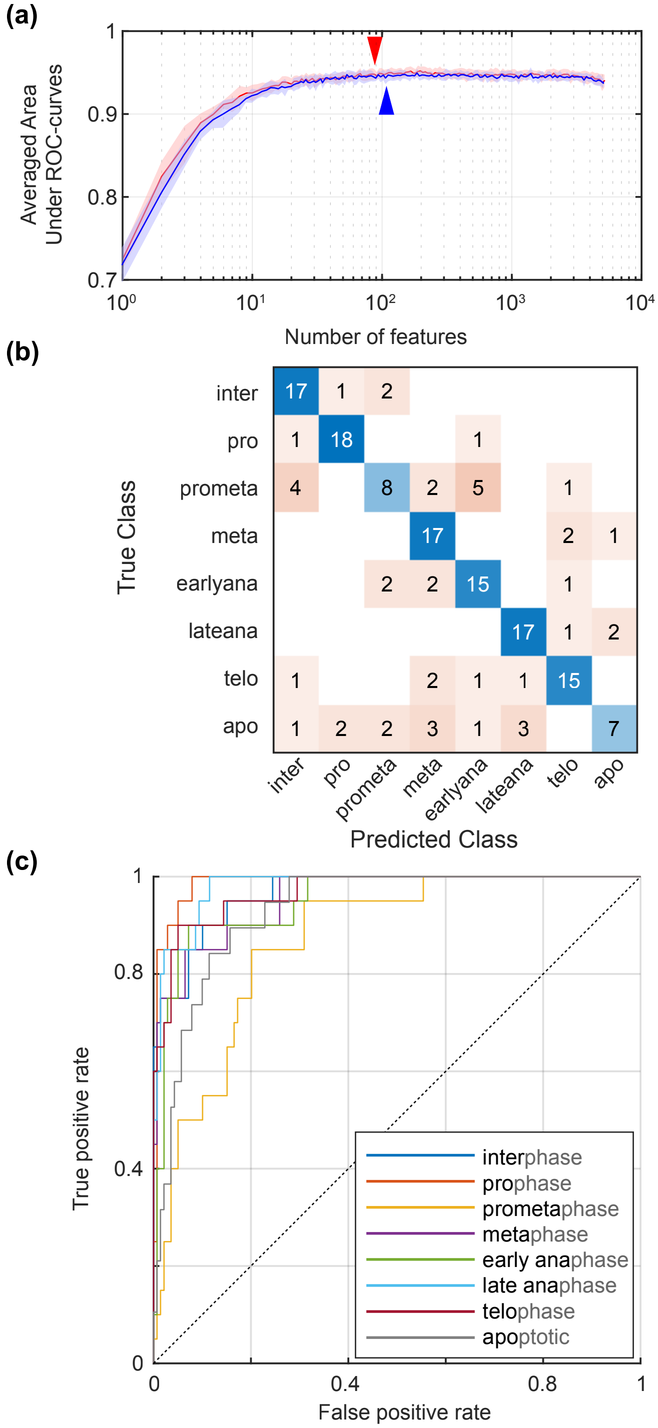


Figure 8: **Random forests classification extracting pseudo-features through a convolutional neural network and optimising the pseudo-feature number** over the *CellCognition* dataset. **(a)** Area Under Curve (AUC) averaged over the classes versus the number of pseudo-features used in classification, including (red curve) all available pseudo-features or (blue curve) discarding the 100 most significant ones. Arrowheads of the corresponding colour depict their optimal number. **(b)** The confusion matrix and **(c)** the ROC curves averaged over the 5-fold cross-validation and ten bootstrap iterations, randomly splitting data into balanced datasets without duplicates (see §2.1).

discriminant ones to be discarded. It proves that such a network could be pruned for the sake of computing time disregarding the importance of the nodes in classification.

4. Discussion and conclusion

In this study, we proposed a method to embed and execute cell-image classification in real-time as an essential module to create a smart microscope used for cell biology at large. In line with the reduced number of images available for training, a peculiar trait of our envisioned application, we used an existing general-purpose image feature extractor coupled with a machine learning algorithm. We analysed the contribution in the classifying decision of each feature, grouped by the image transforms from which they are computed. We took advantage of the machine learning algorithm that was able to report the feature importances. Doing so, we selected a subset of features best discriminating the various mitotic phases. Interestingly, censoring the most computationally intensive features did not degrade the classification upon re-training and selecting a new feature-subset. We could obtain excellent accuracy, suitable for the targeted application, by using a non-linear Machine Leaning method, combined with high execution performance on an embedded system to ensure analysis on-the-fly. In our example, we could classify about 14 cells per second into 8 phases of the cell cycle, with an accuracy greater than 80% using Random forests classification. Using the almost the same subset of features, we can train a small neural network and reach similar performances benefiting of a classifier easy to embed and optimise on GPU. Importantly, this approach is transferable to deep learning network commonly used nowadays.

Why suppressing the most discriminative features, for the sake of the execution time, did not degrade the classification accuracy? The various features, despite they belong to different groups and use a distinct strategy, are likely redundant. However, the quantity redundant to a censored feature is a non-linear combination of the available features as suggested by the better accuracy achieved when using a non-linear method. The replacing features are thus non-intuitive and likely not easily accessible by direct programming, outside of statistical modelling. Indeed, a large set of features as the one offered by WND-CHARM are expected to be redundant, and the use of decision trees appears well appropriate to decrease this redundancy (Tuv et al., 2009; Bolón-Canedo et al., 2012). Beyond this aspect, biological processes might also correlate some features independent mathematically-speaking. For instance, in the context of deep learning and larger image datasets, Nagao and co-authors found that additional markers on top of chromosomes did not improve the classification between the mitotic phases (Nagao et al., 2020). Indeed, the mitotic-phase changes involve numerous modifications of the sub-cellular structures, all under the control of the cell cycle regulation. It translates into various feature evolutions (Pollard and Earnshaw, 2002). Along a similar line, measuring the mitotic spindle – the essential structure tasked to dispatch the chromosomes to daughter cells correctly – suggested that various features are correlated (Farhadifar et al., 2016). Similarly, we recently analysed the mitotic-spindle length and found that only three components, out of a principal component analysis, are enough to account for 95% of inter-individual variability across more than 100 conditions obtained by involved protein depletion (Y. Le Cunff et al., data to be published). Overall, the variegated appearances of the sub-cellular structures as revealed by fluorescence microscopy are under the control of one or a few master regulators. Such a biological-originated correlation, modelled by our machine learning approach, further supports our strategy of reducing redundant features. While we investigated it on cell division, a similar situation likely happens in other cell-biology processes.

The proposed methodology was developed keeping in mind that it should apply to small datasets, a constraint in application to biomedical science (Shaikhina et al., 2015; Kourou et al., 2015; Foster et al., 2014). Indeed, images are long to be produced and annotated. Furthermore, in the case of biological research, each experiment corresponds to a particular dataset: training with images from a distinct experiment (labelling other structures, e.g.) appears a poor option. As a result, only small datasets are available for training. This is a constraint shared with all experimental sciences and engineering, leading to reduce the number of degrees of freedom in the model, i.e. the number of used features and nodes in neural network (Feng et al., 2019; Pasupa and Sunhem, 2016; Shaikhina et al., 2015; Foster et al., 2014). The major risk is overtraining, leading the statistical model to learn details of the training set, failing to extract the general aspects, and *in fine* causing low accuracy on real-data classification (testing). This is also why we opted here for the decision-tree forests and in particular, random forests, known to cope well with this issue at the first place(Breiman, 2001; Azar and El-Metwally, 2012). Once this model is correctly trained, it helps to select features. Indeed, reducing the number of features, discarding the poorly-informative ones, not only improves the execution time but also limits the risk of overfitting in an approach similar to classical dropout technique used in deep learning (Srivastava et al., 2014; Borisov et al., 2006). In conclusion, our approach offers both a feature selection strategy enabling to decide the balance between execution time and accuracy directly, but also enables to use a neural network in a second time, when in the production set-up.

We obtained the presented results using machine learning. We also showed that the removal of the most significant pseudo-features of a deep neural network, i.e. the nodes of the last layer before the fully connected one, does not preclude an accurate classification. On this ground, one can now envision using deep learning, in particular, pruning the networks as we know that an optimal number of features could be found (Molchanov et al., 2016). It will also benefit from the nowadays standard GPU acceleration of convolutional networks. The proposed method, by enabling accurate classification under the constraint of real-time execution, paves the way towards smart microscopy. This novel instrument, beyond making feasible experiments on rare and brief phenomena, will extend the HCS towards High Throughput Experimenting: beyond the bare observation of the sample, it will enable deeper imaging and in the future photo-perturbations. For example, this will enable to challenge the effect of drugs by investigating much more intimate processes of the cell. Finally and in a shorter term, medicine and biology are currently restricted to analyse data *a posteriori*, requiring to acquire a huge amount of images to sort them afterwards because most of them are information-scarce. The smart microscopy promises a more parsimonious approach.

Acknowledgments

We thank Drs. Hélène Bouvrais, Youssef El Habouz, Sébastien Huet, Yann Le Cunff and Sébastien Le Nours for discussions about the project. JP was supported by a Centre National de la Recherche Scientifique (CNRS) ATIP starting grant and La Ligue nationale contre le cancer. We acknowledge the support from Rennes Métropole and Région Bretagne through the program PME 2018-2019 under the reference roboscope, and from the national research agency (ANR-19-CE45-0011). MT and JP acknowledge France-BioImaging infrastructure supported by the French National Research Agency (ANR-10-INBS-04). This work was funded in part by the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute). MB fellowship was partially funded by the ANRT CIFRE program #2017-1589. The University of Rennes 1 and Région Bretagne funded the fellowship of FS.

References

Ali, A., Jalil, A., Niu, J., Zhao, X., Rathore, S., Ahmed, J., Aksam Iftikhar, M., 2015. Visual object tracking—classical and contemporary approaches. Frontiers of Computer Science 10, 167–188. doi:10.1007/s11704-015-4246-3.

Azar, A.T., El-Metwally, S.M., 2012. Decision tree classifiers for automated medical diagnosis. Neural Computing and Applications 23, 2387–2403. doi:10.1007/s00521-012-1196-7.

Balluet, M., Pont, J., Giroux, B., Bouchareb, O., Chanteux, O., Tramier, M., Pécréaux, J., 2020. Method for managing command blocks for a microscopy imaging system, corresponding computer program, storage means and device.

Banfalvi, G., 2017. Overview of cell synchronization. Methods Mol Biol 1524, 3–27. doi:10.1007/978-1-4939-6603-5_1.

Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J., 1997. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence 19, 711–720. doi:10.1109/34.598228.

Bishop, C.M., 2006. Pattern recognition and machine learning. Information science and statistics, Springer, New York.

Bobick, A.F., Davis, J.W., 2001. The recognition of human movement using temporal templates. IEEE Transactions on Pattern Analysis and Machine Intelligence 23, 257–267. doi:10.1109/34.910878.

Bolón-Canedo, V., Sánchez-Maroño, N., Alonso-Betanzos, A., 2012. A review of feature selection methods on synthetic data. Knowledge and Information Systems 34, 483–519. doi:10.1007/s10115-012-0487-8.

Borisov, A., Eruhimov, V., Tuv, E., 2006. Tree-Based Ensembles with Dynamic Soft Feature Selection. Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 359–374. doi:10.1007/978-3-540-35488-8_16.

Breiman, L., 2001. Random Forests. Machine Learning 45, 5–32. doi:10.1023/A:1010933404324.

CellCognition, 2010. Demo data "chromatin + microtubles". https://cellcognition-project.org/demo_data.html.

Chartrand, G., Cheng, P.M., Vorontsov, E., Drozdzal, M., Turcotte, S., Pal, C.J., Kadoury, S., Tang, A., 2017. Deep learning: A primer for radiologists. Radiographics 37, 2113–2131. doi:10.1148/rg.2017170077.

Chen, W., Li, W., Dong, X., Pei, J., 2018. A review of biological image analysis. Current Bioinformatics 13, 337–343. doi:10.2174/1574893612666170718153316.

Chiang, L.H., Russell, E.L., Braatz, R.D., 2000. Fault diagnosis in chemical processes using fisher discriminant analysis, discriminant partial least squares, and principal component analysis. Chemometrics and Intelligent Laboratory Systems 50, 243–252. doi:10.1016/S0169-7439(99)00061-1.

Cireşan, D.C., Giusti, A., Gambardella, L.M., Schmidhuber, J., 2013. Mitosis detection in breast cancer histology images with deep neural networks, in: Mori, K., Sakuma, I., Sato, Y., Barillot, C., Navab, N. (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013, Springer Berlin Heidelberg. pp. 411–418. doi:10.1007/978-3-642-40763-5_51.

Conrad, C., Wünsche, A., Tan, T.H., Bulkescher, J., Sieckmann, F., Verissimo, F., Edelstein, A., Walter, T., Liebel, U., Pepperkok, R., Ellenberg, J., 2011. Micropilot: automation of fluorescence microscopy-based imaging for systems biology. Nature Methods 8, 246–249. doi:10.1038/nmeth.1558.

Djuric, U., Zadeh, G., Aldape, K., Diamandis, P., 2017. Precision histology: how deep learning is poised to revitalize histomorphology for personalized cancer care. NPJ Precis Oncol 1, 22. doi:10.1038/s41698-017-0022-1.

Duchesne, C., Liu, J.J., MacGregor, J.F., 2012. Multivariate image analysis in the process industries: A review. Chemometrics and Intelligent Laboratory Systems 117, 116–128. doi:10.1016/j.chemolab.2012.04.003.

Duda, R., Hart, P., 1973. Pattern classification and scene analysis. Wiley, Philadelphia.

Elting, M.W., Suresh, P., Dumont, S., 2018. The spindle: Integrating architecture and mechanics across scales. Trends Cell Biol 28, 896–910. doi:10.1016/j.tcb.2018.07.003.

Esner, M., Meyenhofer, F., Bickle, M., 2018. Live-cell high content screening in drug development. Methods Mol Biol 1683, 149–164. doi:10.1007/978-1-4939-7357-6_10.

Farhadifar, R., Ponciano, J.M., Andersen, E.C., Needleman, D.J., Baer, C.F., 2016. Mutation is a sufficient and robust predictor of genetic variation for mitotic spindle traits in caenorhabditis elegans. Genetics 203, 1859–70. doi:10.1534/genetics.115.185736.

Fawcett, T., 2006. An introduction to ROC analysis. Pattern Recognition Letters 27, 861–874. doi:10.1016/j.patrec.2005.10.010.

Feng, S., Zhou, H., Dong, H., 2019. Using deep neural network with small dataset to predict material defects. Materials & Design 162, 300–310. doi:10.1016/j.matdes.2018.11.060.

Fillbrunn, A., Dietz, C., Pfeuffer, J., Rahn, R., Landrum, G.A., Berthold, M.R., 2017. Knime for reproducible cross-domain analysis of life science data. J Biotechnol 261, 149–156. doi:10.1016/j.jbiotec.2017.07.028.

Fisher, R.A., 1936. The use of multiple measurements in taxonomic problems. Annals of Eugenics 7, 179–188. doi:10.1111/j.1469-1809.1936.tb02137.x.

Florian, S., Mitchison, T.J., 2016. Anti-microtubule drugs. Methods Mol Biol 1413, 403–21. doi:10.1007/978-1-4939-3542-0_25.

Foster, K.R., Koprowski, R., Skufca, J.D., 2014. Machine learning, medical diagnosis, and biomedical engineering research - commentary. Biomed Eng Online 13, 94. doi:10.1186/1475-925X-13-94.

Hamidinekoo, A., Denton, E., Rampun, A., Honnor, K., Zwiggelaar, R., 2018. Deep learning in mammography and breast histology, an overview and future trends. Medical Image Analysis 47, 45–67. doi:10.1016/j.media.2018.03.006.

Hamilton, P.W., Bankhead, P., Wang, Y., Hutchinson, R., Kieran, D., McArt, D.G., James, J., Salto-Tellez, M., 2014. Digital pathology and image analysis in tissue biomarker research. Methods 70, 59–73. doi:10.1016/j.ymeth.2014.06.015.

Harder, N., Mora-Bermúdez, F., Godinez, W.J., Wünsche, A., Eils, R., Ellenberg, J., Rohr, K., 2009. Automatic analysis of dividing cells in live cell movies to detect mitotic delays and correlate phenotypes in time. Genome Research 19, 2113–2124. doi:10.1101/gr.092494.109.

Held, M., Schmitz, M.H.A., Fischer, B., Walter, T., Neumann, B., Olma, M.H., Peter, M., Ellenberg, J., Gerlich, D.W., 2010. CellCognition: time-resolved phenotype annotation in high-throughput live cell imaging. Nature Methods 7, 747–754. doi:10.1038/nmeth.1486.

Itseez, 2015. Open source computer vision library. https://github.com/itseez/opencv.

Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., Kalenichenko, D., 2018. Quantization and training of neural networks for efficient integer-arithmetic-only inference, pp. 2704–2713. doi:10.1109/cvpr.2018.00286.

Kaushal, M., Khehra, B.S., Sharma, A., 2018. Soft computing based object detection and tracking approaches: State-of-the-art survey. Applied Soft Computing 70, 423–464. doi:10.1016/j.asoc.2018.05.023.

Klonoff, D.C., 2015. Precision medicine for managing diabetes. J Diabetes Sci Technol 9, 3–7. doi:10.1177/1932296814563643.

Koch, C., Georgieva, K., Kasireddy, V., Akinci, B., Fieguth, P., 2015. A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. Advanced Engineering Informatics 29, 196–210. doi:10.1016/j.aei.2015.01.008.

Kourou, K., Exarchos, T.P., Exarchos, K.P., Karamouzis, M.V., Fotiadis, D.I., 2015. Machine learning applications in cancer prognosis and prediction. Comput Struct Biotechnol J 13, 8–17. doi:10.1016/j.csbj.2014.11.005.

Leopold, J.A., Loscalzo, J., 2018. Emerging role of precision medicine in cardiovascular disease. Circ Res 122, 1302–1315. doi:10.1161/CIRCRESAHA.117.310782.

Liyang, W., Yongyi, Y., Nishikawa, R.M., Yulei, J., 2005. A study on several machine-learning methods for classification of malignant and benign clustered microcalcifications. IEEE Transactions on Medical Imaging 24, 371–380. doi:10.1109/TMI.2004.842457.

Loh, W.Y., Shih, Y.S., 1997. Split selection methods for classification trees. Statistica Sinica 7, 815–840. Publisher: Institute of Statistical Science, Academia Sinica.

Manchado, E., Guillamot, M., Malumbres, M., 2012. Killing cells by targeting mitosis. Cell Death Differ 19, 369–77. doi:10.1038/cdd.2011.197.

McIntosh, J.R., 2017. Special Issue ”Mechanisms of Mitotic Chromosome Segregation”. MDPI. Biology.

McQuin, C., Goodman, A., Chernyshev, V., Kamentsky, L., Cimini, B.A., Karhohs, K.W., Doan, M., Ding, L., Rafelski, S.M., Thirstrup, D., Wiegraebe, W., Singh, S., Becker, T., Caicedo, J.C., Carpenter, A.E., 2018. Cellprofiler 3.0: Next-generation image processing for biology. PLoS Biol 16, e2005970. doi:10.1371/journal.pbio.2005970.

Moen, E., Bannon, D., Kudo, T., Graf, W., Covert, M., Van Valen, D., 2019. Deep learning for cellular image analysis. Nat Methods 16, 1233–1246. doi:10.1038/s41592-019-0403-1.

Molchanov, P., Tyree, S., Karras, T., Aila, T., Kautz, J., 2016. Pruning convolutional neural networks for resource efficient inference. arXiv:1611.06440 .

Moran, U., Phillips, R., Milo, R., 2010. Snapshot: key numbers in biology. Cell 141, 1262–1262 e1. doi:10.1016/j.cell.2010.06.019.

Muller, K., Mika, S., Ratsch, G., Tsuda, K., Scholkopf, B., 2001. An introduction to kernel-based learning algorithms. IEEE Transactions on Neural Networks 12, 181–201. doi:10.1109/72.914517.

Nagao, Y., Sakamoto, M., Chinen, T., Okada, Y., Takao, D., 2020. Robust classification of cell cycle phase and biological feature extraction by image-based deep learning. Mol Biol Cell 31, 1346–1354. doi:10.1091/mbc.E20-03-0187.

Neumann, B., Walter, T., Heriche, J.K., Bulkescher, J., Erfle, H., Conrad, C., Rogers, P., Poser, I., Held, M., Liebel, U., Cetin, C., Sieckmann, F., Pau, G., Kabbe, R., Wunsche, A., Satagopam, V., Schmitz, M.H., Chapuis, C., Gerlich, D.W., Schneider, R., Eils, R., Huber, W., Peters, J.M., Hyman, A.A., Durbin, R., Pepperkok, R., Ellenberg, J., 2010. Phenotypic profiling of the human genome by time-lapse microscopy reveals cell division genes. Nature 464, 721–7. doi:10.1038/nature08869.

Nketia, T.A., Sailem, H., Rohde, G., Machiraju, R., Rittscher, J., 2017. Analysis of live cell images: Methods, tools and opportunities. Methods 115, 65–79. doi:10.1016/j.ymeth.2017.02.007.

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, 1684–1693. doi:10.1016/j.patrec.2008.04.013.

Otsu, N., 1979. Threshold selection method from gray-level histograms. Ieee Transactions on Systems Man and Cybernetics 9, 62–66. doi:10.1109/TSMC.1979.4310076.

Pasupa, K., Sunhem, W., 2016. A comparison between shallow and deep architecture classifiers on small dataset, in: 2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE), pp. 1–6. doi:10.1109/ICITEED.2016.7863293.

Peng, H., 2008. Bioimage informatics: a new area of engineering biology. Bioinformatics 24, 1827–36. doi:10.1093/bioinformatics/btn346.

Pollard, T.D., Earnshaw, W.C., 2002. Cell biology. Saunders, Philadelphia.

Potapova, T., Gorbsky, G.J., 2017. The consequences of chromosome segregation errors in mitosis and meiosis. Biology (Basel) 6. doi:10.3390/biology6010012.

Rieder, C.L., Khodjakov, A., 2003. Mitosis through the microscope: advances in seeing inside live dividing cells. Science 300, 91–6. doi:10.1126/science.1082177.

Roul, J., Pecreaux, J., M., T., 2015. Method for controlling a plurality of functional modules including a multi-wavelength imaging device, and corresponding control system. Patent WO2015144650 A1.

Sargano, A., Angelov, P., Habib, Z., 2017. A comprehensive review on handcrafted and learning-based action representation approaches for human activity recognition. Applied Sciences 7. doi:10.3390/app7010110.

Sbalzarini, I.F., 2016. Seeing Is Believing: Quantifying Is Convincing: Computational Image Analysis in Biology. Springer, New York, NY.

Scherf, N., Huisken, J., 2015. The smart and gentle microscope. Nature Biotechnology 33, 815–818. doi:10.1038/nbt.3310.

Shaikhina, T., Lowe, D., Daga, S., Briggs, D., Higgins, R., Khovanova, N., 2015. Machine learning for predictive modelling based on small data in biomedical engineering. IFAC-PapersOnLine 48, 469–474. doi:10.1016/j.ifacol.2015.10.185.

Singh, S., Carpenter, A.E., Genovesio, A., 2014. Increasing the content of high-content screening: An overview. J Biomol Screen 19, 640–50. doi:10.1177/1087057114528537.

Sivakumar, S., Gorbsky, G.J., 2015. Spatiotemporal regulation of the anaphase-promoting complex in mitosis. Nat Rev Mol Cell Biol 16, 82–94. doi:10.1038/nrm3934.

Sizaire, F., Le Marchand, G., Pecreaux, J., Bouchareb, O., Tramier, M., 2020. Automated screening of aurka activity based on a genetically encoded fret biosensor using fluorescence lifetime imaging microscopy. Methods Appl Fluoresc 8, 024006. doi:10.1088/2050-6120/ab73f5.

Sommer, C., Gerlich, D.W., 2013. Machine learning in cell biology - teaching computers to recognize phenotypes. J Cell Sci 126, 5529–39. doi:10.1242/jcs.123604.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15, 1929–1958.

Tischer, C., Hilsenstein, V., Hanson, K., Pepperkok, R., 2014. Adaptive fluorescence microscopy by online feedback image analysis. Methods Cell Biol 123, 489–503. doi:10.1016/B978-0-12-420138-5.00026-4.

Tuv, E., Borisov, A., Runger, G., Torkkola, K., 2009. Feature Selection with Ensembles, Artificial Variables, and Redundancy Elimination. Journal of Machine Learning Research 10, 1341–1366.

Veta, M., van Diest, P.J., Willems, S.M., Wang, H., Madabhushi, A., Cruz-Roa, A., Gonzalez, F., Larsen, A.B., Vestergaard, J.S., Dahl, A.B., Ciresan, D.C., Schmidhuber, J., Giusti, A., Gambardella, L.M., Tek, F.B., Walter, T., Wang, C.W., Kondo, S., Matuszewski, B.J., Precioso, F., Snell, V., Kittler, J., de Campos, T.E., Khan, A.M., Rajpoot, N.M., Arkoumani, E., Lacle, M.M., Viergever, M.A., Pluim, J.P., 2015. Assessment of algorithms for mitosis detection in breast cancer histopathology images. Med Image Anal 20, 237–48. doi:10.1016/j.media.2014.11.010.

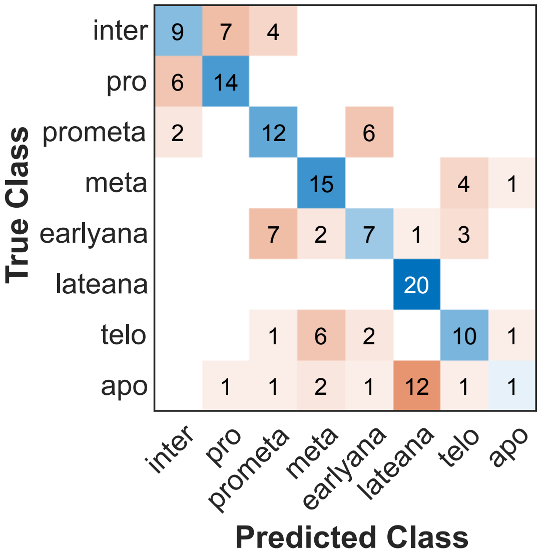
Wang, H., Roa, A.C., Basavanhally, A.N., Gilmore, H.L., Shih, N., Feldman, M., Tomaszewski, J., Gonzalez, F., Madabhushi, A., 2014. Mitosis detection in breast cancer pathology images by combining handcrafted and convolutional neural network features. Journal of Medical Imaging 1, 034003. doi:10.1117/1.JMI.1.3.034003.

Wollmann, T., Erfle, H., Eils, R., Rohr, K., Gunkel, M., 2017. Workflows for microscopy image analysis and cellular phenotyping. J Biotechnol 261, 70–75. doi:10.1016/j.jbiotec.2017.07.019.

Zhang, S., Wei, Z., Nie, J., Huang, L., Wang, S., Li, Z., 2017. A review on human activity recognition using vision-based method. J Healthc Eng 2017, 3090343. doi:10.1155/2017/3090343.

Zhao, Z., Anand, R., Wang, M., 2019. Maximum Relevance and Minimum Redundancy Feature Selection Methods for a Marketing Machine Learning Platform. arXiv:1908.05376 .

Supplemental figures



True Class \ Predicted Class	inter	pro	prometa	meta	earlyana	lateana	telo	apo
inter	9	7	4					
pro	6	14						
prometa	2		12		6			
meta				15			4	1
earlyana			7	2	7	1	3	
lateana						20		
telo			1	6	2		10	1
apo		1	1	2	1	12	1	1

Figure S1: **Classification using a single feature (Otsu-segmented-region area)** resulted in a poor confusion matrix. Class names are abbreviated after Fig. 1a. CellCognition dataset was used (see Methods §2.1).

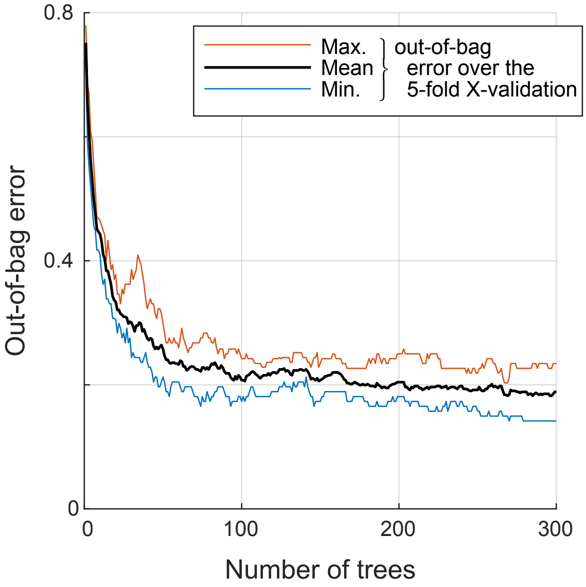


Figure S2: **Selecting the number of trees in random forests classifier** by plotting the out-of-bag error versus the number of trees. The black, blue and red lines depict the average, minimum and maximum out-of-bag errors, respectively, over the 5-fold iterations of the cross-validation. CellCognition dataset was used (see Methods §2.1).

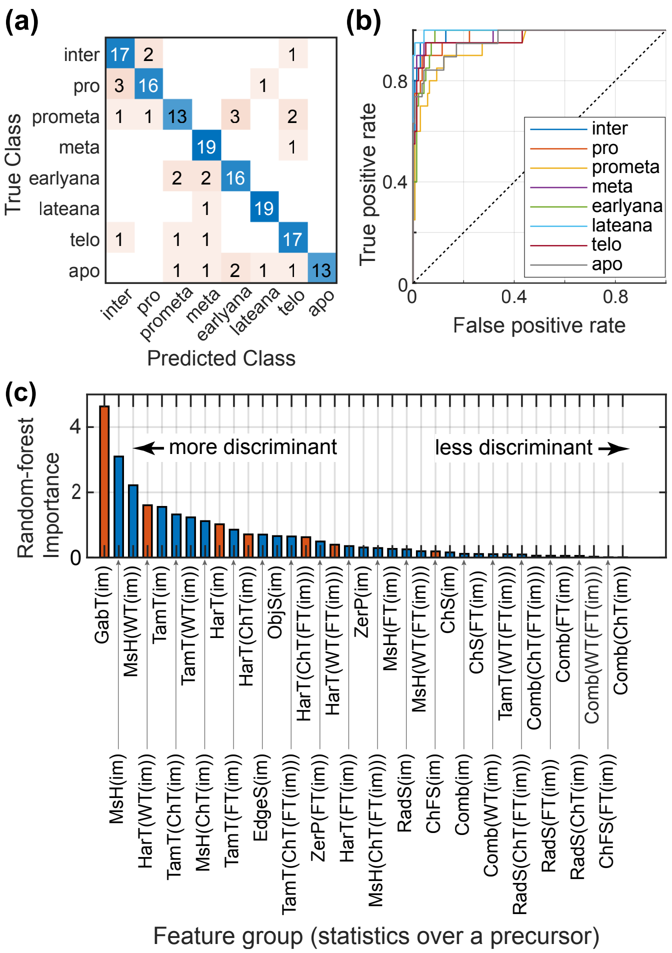


Figure S3: **Random forests using all the 1025 features** was trained and tested over 20% of the dataset to get **(a)** the confusion matrix and **(b)** the ROC curves over the 5-fold cross-validation using the CellCognition dataset (see Methods §2.1). Class names are abbreviated after Fig. 1a.

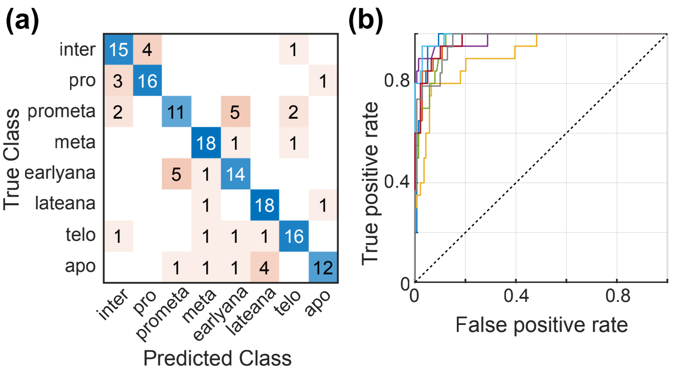


Figure S4: **Random forests with computationally intensive features** optimised by removing low importance feature groups. The algorithm was trained and tested over 20% of the dataset to get **(a)** the confusion matrix and **(b)** the ROC curves over the 5-fold cross-validation using the CellCognition dataset (see Methods §2.1). Class names are abbreviated after Fig. 1a.

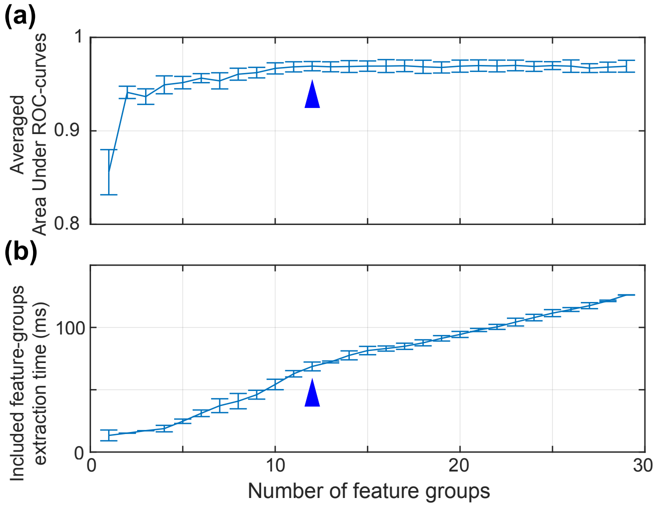


Figure S5: **Bootstrapping the random forests optimised with only non-computationally-intensive feature-groups**. **(a)** The Area Under Curve (AUC) was averaged over the classes, and **(b)** execution time for extracting the feature-groups included in the classification was assessed. Both quantities are plotted versus the number of feature-groups used in classification and were computed in the 5-fold cross-validation repeats. This approach was repeated 10 times in the bootstrap approach, where the vignettes included in the balanced dataset were selected differently from the CellCognition (see Methods §2.1). We thus obtained the standard deviations reported by the error bars. Fig. 5ab report results in the same conditions for a single bootstrap iteration. Arrowheads depict the 12 feature groups optimal case.