Neural network fast-classifies biological images through features selecting to power automated microscopy

Maël Balluet, Florian Sizaire, Youssef El Habouz, Thomas Walter, Jérémy Pont, Baptiste Giroux, Otmane Bouchareb, Marc Tramier, Jacques Pecreaux

ORIGINAL ARTICLE

Neural network fast-classifies biological images through features selecting to power automated microscopy.

Maël Balluet[1,2] | Florian Sizaire[1,6*] | Youssef El Habouz[1] | Thomas Walter[3,5,7] | Jérémy Pont[2] | Baptiste Giroux[2] | Otmane Bouchareb[2] | Marc Tramier[1,4] | Jacques Pecreaux[1]

[1]CNRS, Univ Rennes, IGDR - UMR 6290, F-35043 Rennes, France

[2]Inscoper SAS, F-35510 Cesson-Sévigné, France

[3]Centre for Computational Biology (CBIO), MINES ParisTech, PSL University, F-75272 Paris, France

[5]Institut Curie, F-75248 Paris, France

[7]INSERM, U900, F-75248 Paris, France

[4]Univ Rennes, BIOSIT, UMS CNRS 3480, US INSERM 018, F-35000 Rennes, France

Correspondence
Jacques Pecreaux PhD, IGDR UMR 6290 CNRS / Univ Rennes, F35043, France
Email: jacques.pecreaux@univ-rennes1.fr

Present address
[*]Biologics Research, Sanofi R&D, F94400 Vitry sur Seine, France

Funding information
Rennes Métropole and Région Bretagne, PME 2018-2019/Roboscope; National Research Agency/ANR-19-CE45-0011; France-BioImaging infrastructure, National Research Agency/ANR-10-INBS-04; "Investissements d'avenir" program, PRAIRIE 3IA Institute/ANR-19-P3IA-0001; ANRT, CIFRE program/2017-1589

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 by making 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 a real-time image processing that is 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 to understand 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 significant accuracy loss, 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. 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.

KEYWORDS
Machine vision and scene understanding, Cell biology, Image processing, Embedded system, 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 [1]. 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 [2, 3, 4, 5]. However, the analysis of this data flood is performed posteriorly to the acquisition, limiting the information extracted [6]. 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)[7]. 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. It will 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 [8, 9, 10, 11]. The current systems that perform imaging and analysis in tandem alternate acquiring images and analysing them [12, 13]. We recently achieved efficient microscope driving [14, 15] and here investigate how to perform the real-time object's classification to feedback to it.

Searching for rare and brief events is a booming field beyond sole microscopy. They often carry significant information about normal or abnormal processes in a broad range of applications [16, 17]. Radiologists use such algorithms to assist the medical-doctor diagnosis interactively, calling for reduced image processing delay [18]. 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 [19, 20, 21]. Similarly, it can support detecting and diagnosing faults in construction or process industries [22, 23]. These situations may result in costly damages, human injuries and require rapid detection through real-time analysis. We here used a similar approach to detect rare and transient events in living biological samples.

The anaphase of cell division is very archetypal to these events 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) [24]. Cell division has received strong attention in fundamental research as its mechanisms are only partially known, and in applied research in particular to develop cancer therapies [25, 26, 27, 28, 29]. 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 [30, 31]. 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 [32]. 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 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 [25]. It implies the ability to fast screen across a library of compounds and quickly assess defects in mitosis and particularly deadlocked mitosis due to unsatisfied SAC [27]. Along a medical line, detecting mitosis in patient tissues is classically used for diagnosis as in breast cancer [33, 34, 35]. 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 identifying the hits in high content screening [36, 37, 38], 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 [39, 40, 12]. However, these classifiers may 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 [4]. 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 [41, 42, 6, 43, 4, 1]. Along that line, we here used a wide variety of features found in the library WND-CHARM [44]. The key to performing 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 numerical optimising onto a set of labelled images. In contrast to deep learning, it enabled identifying important features and even manually manipulating 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 [45]. This article proposed a strategy to optimise the selection of features of interest under the constraint of both accurate classification and fast performance. It implies to selecting 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 [40] software demonstration images. It comprises 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[40]. 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* [46]. Compared 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 [46], 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, according to the procedure in [46]. 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 | Feature 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 [44]. 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 the whole dataset. In particular, we ensured that the execution

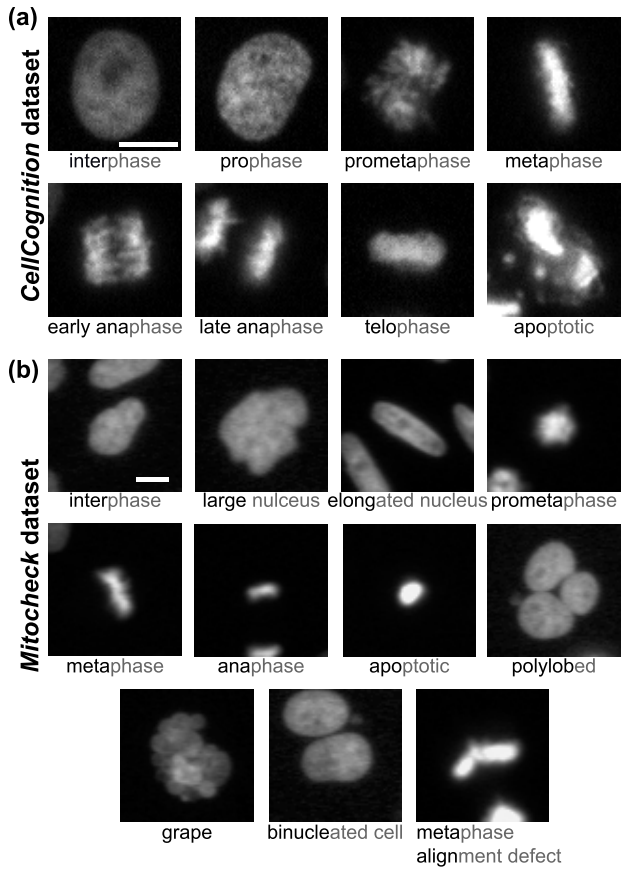
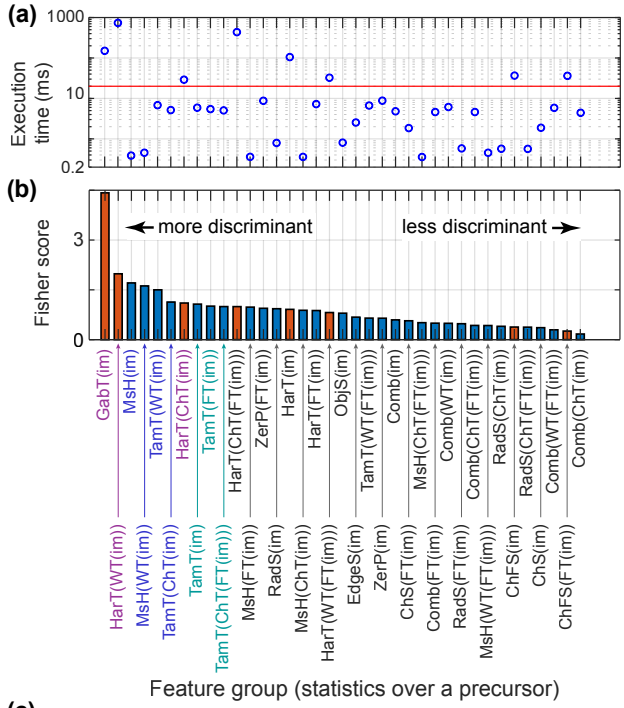


FIGURE 1 **Datasets used during numerical experiments. (a)** Exemplar vignettes upon 71×71 pixels cropping images from the CellCognition database. **(b)** Exemplar vignettes similarly cropped and extracted from the mitocheck database. Class names were abbreviated and written in black font, while the full name appeared 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.

was sequential on the CPU of the embedded system without using parallelism. When estimating the computing time of multiple features, we noticed that the features were not independent. Indeed, within a given group of features, they all correspond to statistics computed from 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 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).



(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 [44, 47]. 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 the analysis result back to the microcontroller that drove the microscope and its attached devices. We embedded the processing on a microcontroller to ensure real-time processing 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 set to classify mitotic images within 8 classes using the CellCognition example set [40, 48] (see Methods §2.1). We especially 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 [44]. 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 [49]. 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 most 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 few 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 suited to our lack of a large training set and the 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 [50, 51], it requires a large training set. We first opted for a linear machine-learning algorithm, specifically Fisher's linear discriminant [52, 53]. Indeed such a kernel method, because linear, promised short execution times and was successful in similar problems [54, 55, 56, 57].

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) [58, 59, 60]. 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 detail, we assessed the classification quality through the area under the ROC curves (AUC) averaged over the eight classes of our dataset, a classical metric in machine learning [61]. 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 its 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, 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).

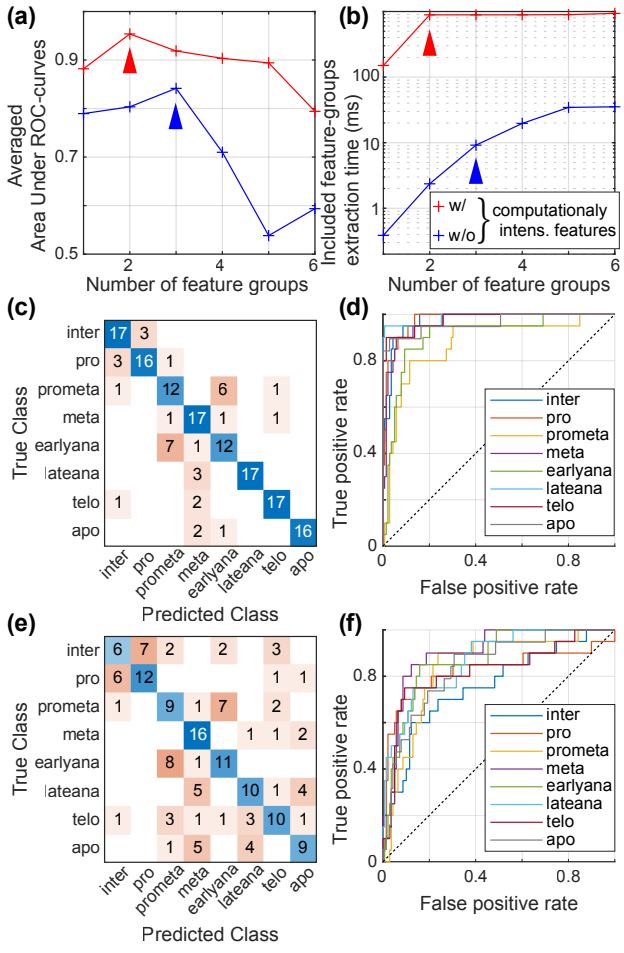


FIGURE 3 **Classification using Fisher's linear discriminant. (a)** Area Under Curve (AUC) averaged over the classes and **(b)** execution time for extracting the feature-groups included in the classification, both versus the number of feature-groups used in classification, including (red curve) all 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).

We noticed that the most discriminant feature-groups displayed a score neatly larger than 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 to prevent overfitting (named in light and dark blue Fig. 2b). As explained above, we then selected a subset of the groups, by excluding the least discriminant ones. 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 for the lack of the two most-discriminant groups and resulted in inaccurate classification that could not fit our applicative needs. 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. We observed a partial redundancy of the features in distinct groups and that classifying itself took a negligible time, provided that the features were already computed. It called for using the non-linear classification method.

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 by 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 [62, 63]. It is a machine learning algorithm based on an ensemble of decision trees that internally selects the most discriminant features, in line with our goal of using 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 the curvature test to select the best split predictor [64], 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, slightly better than Fisher's linear discriminant. All the classes were recovered at least as accurately or better by the random forests (Fig. S3). This 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.

The random forests offer a mechanism to assess the *importance* of each feature in the decision [63]. In a nutshell, it corresponds to the difference of the misclassification rate 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).

To perform a fast classification, we removed the least important feature groups again, computed the random-

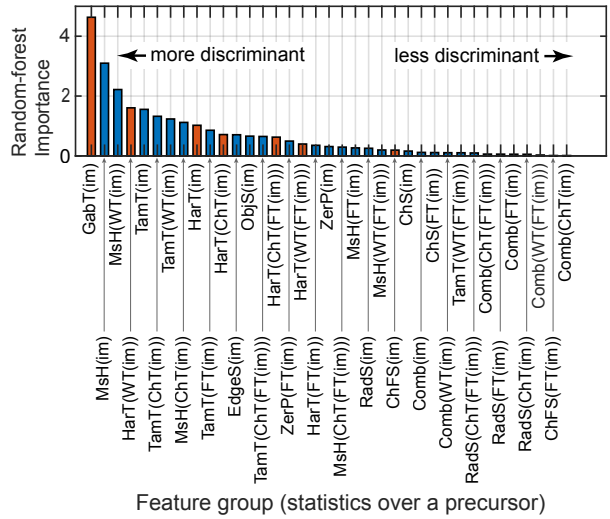


FIGURE 4 **Random forests using all the 1025 features** was trained and tested over 20% of the dataset, and we retrieved the importance of each feature-group (see main text). Red bars highlight the feature groups displaying an execution time greater than 20 ms. The execution times are reported in Fig.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

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 applying random forests to on-the-fly classification, we 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. Based on the trend obtained using Fisher's linear discriminant, we excluded the groups with execution time 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 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 a 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 looked 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 suggests redundancy of the features, at least in the measurements computed on the present images. Indeed, the random forests could spread the importance among the redundant features and thus compensate for removed redundant features [62, 65]. With the proposed feature-groups selection, such an ability could ensure fast execution on an embedded system and automated microscope.

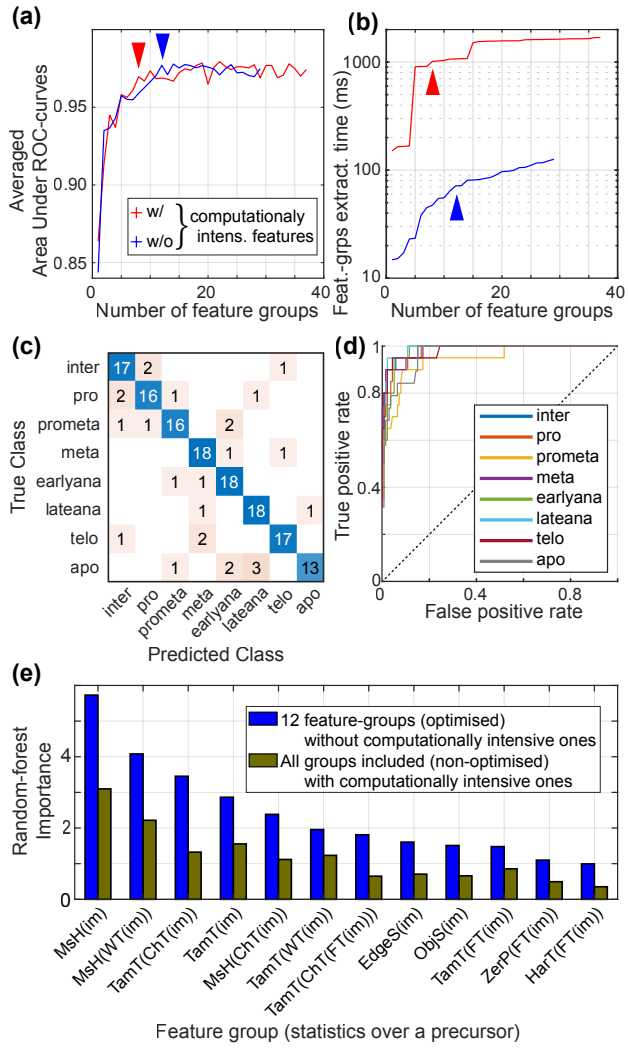


FIGURE 5 **Random-forests classification using a subset of feature-groups**. **(a)** Area Under Curve (AUC) averaged over the classes and **(b)** execution time for extracting the feature-groups included in the classification, both versus the number of feature-groups used in classification, including (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 optimal case of the twelve-feature-groups without computationally intensive ones using the CellCognition dataset (see §2.1). Class names are abbreviated after Fig. 1a.

We reckoned that these results represented one particular instance of database equilibration (see §2.1). To test the generality of our approach, we used bootstrap to randomly split data into balanced datasets 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 $\pm$ standard deviation), as found previously, although the optimum might vary by a few units. We 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). The slight changes in the set of selected feature-groups in each bootstrap iteration could account for such a variation. 11 feature-groups were present in all bootstrap instances, and the last one was drawn among 4 feature-groups (black and blue text, Tab. 1). We overall suggest that our method offers reproducibility upon using different training subsets.

Feature groups	In *n* iter.	Feature groups	In *n* 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 (and not when using *mitocheck*) were italicised (Tab. 2). The feature groups are described in Fig. 2c.

While an 80% accurate classification appeared suitable for automated microscopy, we investigated the reason for misclassifying some images. Firstly, the cell goes through cell division and interphase following a continuous evolution split into phases (Fig 1a, note the order of the phases). Between late interphase, prophase and even the beginning of prometaphase, the nucleus looked similar, leading to some confusion (Fig S7, green frames). Similarly, late metaphase and the beginning of early anaphase could be confused when the two sets of sister chromatids are not clearly separated, i.e. no dark region in-between was visible (Fig S7, blue frames). The proximity of other cells in a different stage was also a common source of misclassification (Fig S7, purple triangles). These limitations are more experimental than classifier-related, either due to either the continuous transition between classes or multiple cells in an image. We attributed the 5 remaining misclassified images to the variability of biological cells, which could lead to confusion between distant but related-looking classes as metaphase and telophase.

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: we performed a k-fold validation process ($k = 5$) followed by ten bootstrap iterations, randomly splitting data into balanced datasets without replacement (no duplicated image). Eight feature groups, excluding those whose execution time exceeded 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). It confirmed the robustness of the above procedure used to speed up image processing.

Feature groups	In *n* iter.	Feature groups	In *n* 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 (and not when using *CellCognition*) were italicised (Tab. 1). The feature groups are described in Fig. 2c.

We wondered how the set of selected features is linked to the very problem solved, or in other words, how generic is the trained network to select cells in different states. We tested using the two above databases, whose classes differ (Fig 1). It is noteworthy that the selected feature groups differ between (Tables 1 and 2). We tested the classification of *mitocheck* images with the network trained over *CellCognition*. We scaled and padded the images to get a similar resolution and size across databases. We kept all the features (no selection) to facilitate the task having in mind that random forests are not prone to overfitting. We obtained poor results, a 36% accuracy and AUC reading 0.47. The converse experiment, using a *mitocheck*-trained network to classify *CellCognition* images, was not better, displaying a 42% accuracy and AUC equals to 0.50. We concluded that the proposed strategy, by optimising execution time, prevent a direct application of a trained network to an other problem involving cells looking different and only-related classes.

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. We set to assess the classification time upon embedding the random forests. Indeed, the decision trees at the core of this algorithm could perform slowly. To do so, we used the RTrees module using the OpenCV library [66]. For the sake of simplicity, in a proof-of-concept perspective, we trained the algorithm using OpenCV on the embedded system. One could train on a general-purpose computer and embed only the classification. We 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. Overall, 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 ensure fast and accurate classification. We could envision using our feature-group optimised random forests together with the WND-CHARM features to enslave microscope driving to image classification.

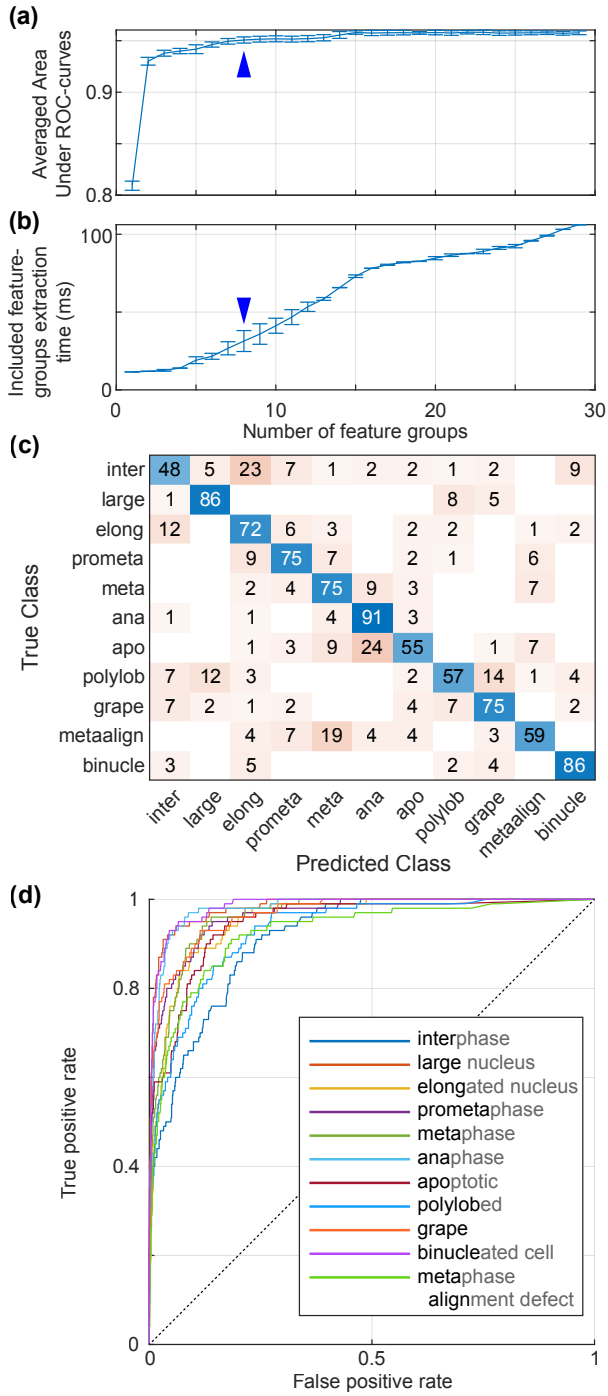


FIGURE 6 **Bootstrapping optimised random forests over *mitocheck* dataset.** **(a)** The Area Under Curve (AUC) was averaged over the classes, and **(b)** execution time for extracting the feature-groups included in classification was assessed (dependent of the selected feature-groups mildly variable 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.

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

Deep learning is the current paradigm in biological images analysis [42, 41]. We wondered whether the proposed approach discarding highly discriminant features for the sake of rapidity keeping accuracy could be used in that context to classify images faster. Indeed, fundamental research applications are more demanding about performances, requiring faster imaging frame-rate. Indeed, when studying mitotic events like metaphase-anaphase transition, the component dynamics are on the second- or even the tenth-of-a-second-scale [67]. 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. The time spent in the classification itself could also be improved. However, because of the high usage of conditional structures in such decision-tree-based methods, parallelising the random forests appeared difficult. We addressed this question in two steps: first, using a neural network as a classifier and second, extracting the features through the convolutional layers of a deep network classifier. However, these methods are more prone to overfitting [62, 68]. This issue is worsened by the large number of features, besides non-independent, correlated or poorly informative for some.

In the first case, we selected the optimal feature groups using random forests and used the neural network in "production context" to perform classification to safeguard against overfitting. In particular, 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. An L2 regularisation parameter was added to the loss function with a 0.1 ratio to avoid over-fitting,. These training parameters have been experimentally determined. We divided the dataset into three parts: training (70%), validation (20%) and test (10%). The validation subset allowed to stop training when the neural network started to overfit. We used again bootstrap to randomly split the whole data into a balanced dataset, without replacement (no duplicated image) (see §2.1). 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 network's weights and biases were initialised to the same values. Using the random forests, we determined that 15 non-computationally-intentive feature-groups allowed an optimal classification. Training and testing the neural network resulted in 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 of using a simple neural network with equal classification quality despite the small training set and many features.

We embedded our neural network using activation functions provided by the OpenCV library. After proper training, we executed the classification of 32 test vignettes. The execution time read 92 ± 15 µs, extrapolated to 28 ± 4 ms for an image containing 300 cells, comparable to the above random forests. The neural network could be further accelerated using GPU parallelisation. However, 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 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.

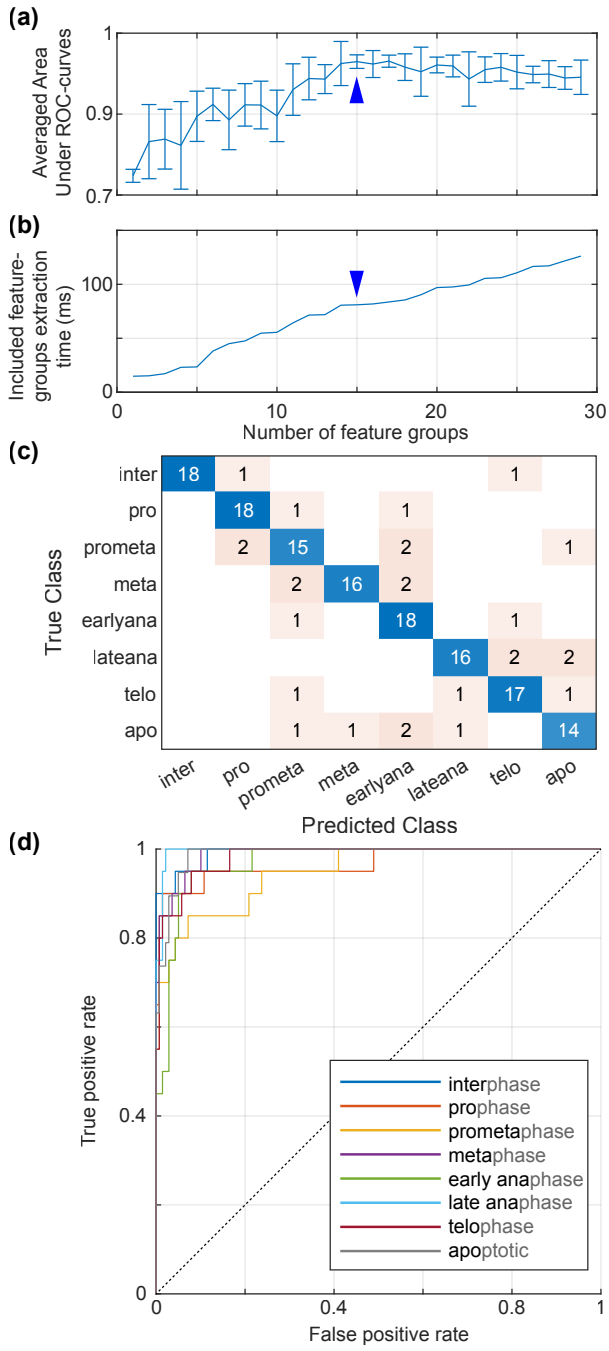


FIGURE 7 **Bootstrapping optimised neural network over *CellCognition* dataset. (a)** The Area Under Curve (AUC) was averaged over the classes, and **(b)** execution time for extracting the feature-groups included in the classification was assessed. Both quantities are plotted versus the number of feature-groups 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 fifteen-feature-groups optimal case. **(c)** The confusion matrix and **(d)** the ROC curves over the 5-fold cross-validation in a single bootstrap iteration. Note 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).

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 using Keras. We kept the same division of the dataset into three parts. We trained this network on the cell cognition dataset using a gradient descent optimiser with a 0.001 learning rate, a batch size of 8 and over 100 epochs. 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. We again performed 5-fold cross-validation followed by ten bootstrap iterations, randomly splitting data into balanced datasets without replacement (no duplicated image). We first included all the pseudo-features and iteratively reduced the number of features by discarding the less important ones. We obtained an optimal classification with 88 ± 48 pseudo-features (mean $\pm$ standard deviation) (Fig. 8a, red curve). 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 $72 \pm 2\,\%$.

We then tested whether the compensating mechanism previously observed was applicable here. We 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). We observed an equivalent variability of pseudo-features included compared to the all-pseudo-feature case: 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 also displayed a similar accuracy (Fig. 8a, compare red and blue curve tails and optimal pseudo-feature number marked by the arrowheads). In further detail, we found an averaged AUC of 0.945 ± 0.005 and global accuracy of $71 \pm 2\,\%$; 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 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.

Throughout this study, we adopted a machine learning approach. We asked how the result compared to the one of a deep network. We compared classification results using the entire deep network described above with the approach combining WND-CHARM and random forests, namely an average AUC of 0.95 ± 0.01 and global accuracy somewhat lower, $72 \pm 4\,\%$ over the 5-fold cross-validation (Fig. S6). We embedded this network on an NVIDIA Jetson AGX Xavier, in a similar fashion as the neural network. The classification time, averaged over the testing images across the 5-fold cross-validation, read 1.8 ms per vignette, which is about twenty times longer compared to optimised WND-CHARM and random forests. Indeed, the use of the entire deep network was equivalent to use all the features. This lukewarm result was furthermore obtained with a simple network. However, numerous recent developments aimed at making deep learning faster [69] and such limitations could be released in the future.

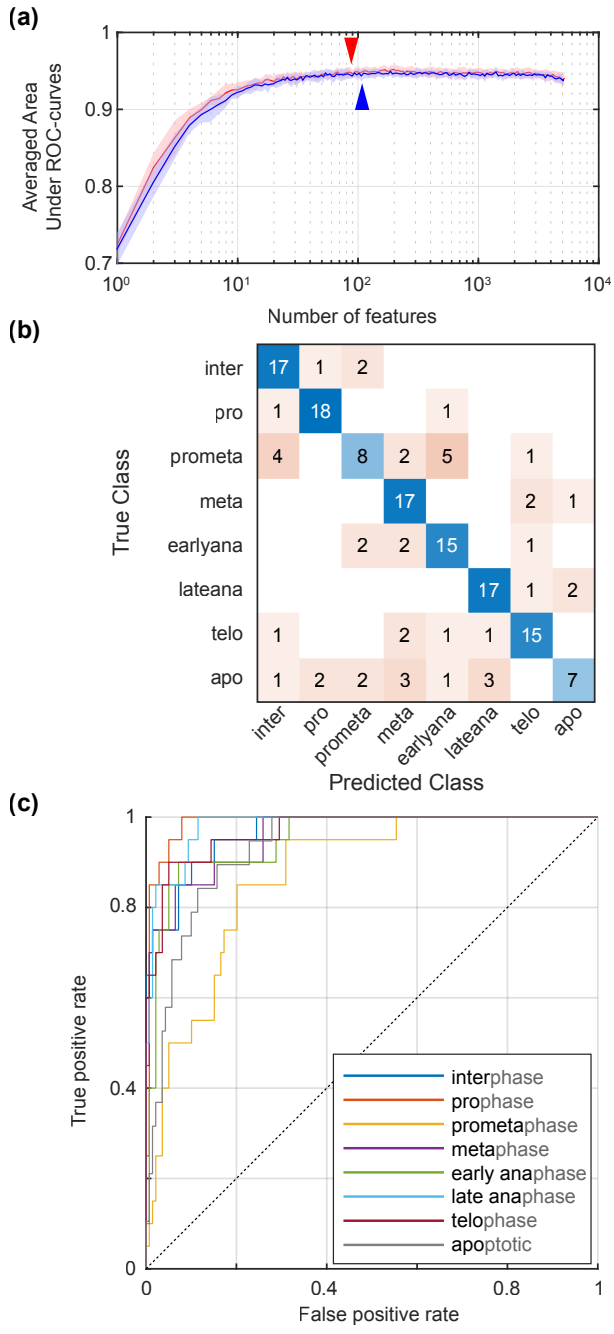


FIGURE 8 **Random forests classification extracting pseudo-features through a convolutional neural network and optimising the pseudo-feature number** over the *CellCognition* dataset. **(a)** Area Under Curve (AUC) averaged over the classes versus the number of pseudo-features used in classification, 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).

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 an automated 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 almost the same subset of features, we can train a small neural network and reach similar performances benefiting from a classifier easy to embed and optimise on GPU. Importantly, this approach is transferable to deep learning networks commonly used nowadays.

Despite machine learning acts somewhat like a "black box", one could speculate on the use of each selected feature-group to perform the classification. For cell cognition database, the EdgeS (edge statistics) feature-group likely detected objects with clear border as in metaphase or telophase compared to dimmer objects in classes before or after in mitotic phases order (Table 1). The ObjS (object statistic) feature group is sensitive to the objects' intensity variations and helped distinguishing metaphase from telophase. However, most of the involved feature-groups correspond to texture analysis. They likely allowed to distinguish for instance, the patchy isotropic texture at prophase, when chromatid was still not fully condensed, from the one displaying lines at the next phase (prometaphase), when chromosomes arms became visible. Although one can speculate a posteriori on the use of each selected feature group, a manual selection of the feature-groups appears hardly possible.

Why suppressing the most discriminative features, for the sake of the execution time, did not degrade the classification accuracy? Although they be- long to different groups and use a distinct strategy, the various features act likely redundantly. It involved a non-linear combination of the available features, as suggested by the better accuracy achieved when using random forests. Thus, replacing features is 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 [62, 68]. This redundancy is unlikely to be mathematical, i.e. the different feature-groups do not rely on the same computations. Beyond this aspect, biological processes might also link some features by correlating different details of the images. For instance, metaphasic chromosomes organisation causes sharply defined filaments because of condensation (detected by edge statistics); these are brighter (object statistics) and mostly parallelly aligned (texture-related features).

In a broader take, using 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 [42]. 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 [70]. 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 [71]. Likewise, we recently analysed the mitotic-spindle length: we 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 controlled by few master regulators. Such a biological-originated correlation, modelled by our machine learning approach, further supports our strategy of reducing redundant features. While we used cell division in this study, a similar situation likely happens in other cell-biology processes.

From an automatic microscope's perspective, an 80% success rate is enough as it guarantees to retrieve most of the events of interest. For instance, we may wish to capture anaphasic cells at a high frame rate and with various wavelengths corresponding to different labelled proteins. The classification described here will instruct the microscope automaton where and when to perform these more in-depth acquisitions [45]. We can trigger it upon detecting metaphasic cells. We consider the class of interest as the most probable one in the machine learning classification, ensuring a balance between false-positive and false-negative. As an alternative, the envisaged automaton architecture leave room for adjusting the minimal probability for a class (metaphase, e.g.) to trigger in-depth acquisition and reduce the false positive at the expense of larger false negative. In a broader take, our 80% success rate corresponds to the expectation for such an autonomous microscope in the field and will outperform human selection as mostly reported in medical imaging [72, 73, 74, 75].

We opted for machine learning in the present work, although it required computing the features separately compared to deep learning. The WND-CHARM library contains a broad range of features, and we believe that most of the problems analysing biological images will find some appropriate. The approach proposed here will contribute to select them. We furthermore showed that the exact list of features used is flexible as we managed to remove the more computationally intensive without degrading classification. Therefore, machine learning appeared advantageous for three reasons: ((i) rather than using anonymous pseudo-features as in deep learning, we had meaningful ones. it allowed us to claim for compensating features across distinct groups using non-related algorithms. (ii) Performances were an essential aspect as we expect on-the-fly classifying of microscopy images. Indeed, the convolutional layers at the core of feature extracting in the deep networks are the most time-demanding. To remedy to this drawback, various approaches to prune and optimise a deep network were proposed [69]. However, it remains tedious, and in contrast, the approach appeared much simpler. For instance, the classical U-net, developed with performance in mind, required about 1 s to classify a 512x512 px frame [76], or recently YOLOv2 achieved about 5 frames per second embedded on a Jetson TX2, but with a 416×416 pixels frame [74]. In contrast, an automatic microscope should classify 10-30 frames per second with 2048×2048 pixels, broken into subframes. (iii) The low number of training images is a stringent constraint in applying our work to automated microscopy in biology, as annotating is highly time-consuming for experimenters. Deep networks often require a larger training dataset and are prone to overfitting [77], while random forests are robust to that issue by their design. Overall, future work may qualify the use of deep learning for automated microscopy, but it appeared interesting to demonstrate this compensating mechanism at first, using "classical" machine learning.

The proposed methodology was developed keeping in mind that it should apply to small datasets, a constraint in application to biomedical science [58, 59, 60]. Indeed, images are long to be produced and annotated. Furthermore, in 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. It is a constraint shared with all experimental sciences and engineering, leading to reduced numbers of degrees of freedom in the model, i.e. the number of used features and nodes in neural networks [78, 79, 58, 60]. 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). Decision-tree forests, particularly random forests, are known to cope well with this issue in the first place[63, 80]. Once this model is correctly trained, it helps to se-

lect 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 [81]. In conclusion, our approach offers both a feature selection strategy enabling us to directly decide the balance between execution time and accuracy, and enables us 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 removing 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 envision using deep learning, in particular, pruning the networks as we know that an optimal number of features could be found [51]. It will also benefit from the nowadays standard GPU acceleration of convolutional networks. By enabling accurate classification under the constraint of real-time execution, the proposed method paves the way towards smart microscopy. On top of making experiments on rare and brief phenomena achievable, this novel instrument 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 challenging the effect of drugs by investigating much more intimate cell processes. Finally, and in the 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. 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 the 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

references

[1] Nketia TA, Sailem H, Rohde G, Machiraju R, Rittscher J. Analysis of live cell images: Methods, tools and opportunities. Methods 2017;115:65–79.

[2] Esner M, Meyenhofer F, Bickle M. Live-Cell High Content Screening in Drug Development. Method Cell Biol 2018;1683:149–164.

[3] Peng H. Bioimage informatics: a new area of engineering biology. Bioinformatics 2008;24(17):1827–36.

[4] Sbalzarini IF. Seeing Is Believing: Quantifying Is Convincing: Computational Image Analysis in Biology. In: Focus on bio-image informatics New York, NY: Springer; 2016. p. 1–40.

[5] Chen W, Li W, Dong X, Pei J. A Review of Biological Image Analysis. Curr Bioinform 2018;13(4):337–343.

[6] Singh S, Carpenter AE, Genovesio A. Increasing the Content of High-Content Screening: An Overview. J Biomol Screen 2014;19(5):640–50.

[7] Scherf N, Huisken J. The smart and gentle microscope. Nat Biotechnol 2015;33(8):815–818.

[8] Hamilton PW, Bankhead P, Wang Y, Hutchinson R, Kieran D, McArt DG, et al. Digital pathology and image analysis in tissue biomarker research. Methods 2014;70(1):59–73.

[9] Leopold JA, Loscalzo J. Emerging Role of Precision Medicine in Cardiovascular Disease. Circ Res 2018;122(9):1302–1315.

[10] Klonoff DC. Precision medicine for managing diabetes. J Diabetes Sci and Technol 2015;9(1):3–7.

[11] Djuric U, Zadeh G, Aldape K, Diamandis P. Precision histology: how deep learning is poised to revitalize histomorphology for personalized cancer care. NPJ Precis Oncol 2017;1(1):22.

[12] Conrad C, Wünsche A, Tan TH, Bulkescher J, Sieckmann F, Verissimo F, et al. Micropilot: automation of fluorescence microscopy-based imaging for systems biology. Nat Methods 2011;8(3):246–249.

[13] Tischer C, Hilsenstein V, Hanson K, Pepperkok R. Adaptive fluorescence microscopy by online feedback image analysis. Method Cell Biol 2014;123:489–503.

[14] Sizaire F, Le Marchand G, Pecreaux J, Bouchareb O, Tramier M. Automated screening of AURKA activity based on a genetically encoded FRET biosensor using fluorescence lifetime imaging microscopy. Methods Appl Fluores 2020;8(2):024006.

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

[16] Ali A, Jalil A, Niu J, Zhao X, Rathore S, Ahmed J, et al. Visual object tracking—classical and contemporary approaches. Front Comput Sci 2015;10(1):167–188.

[17] Kaushal M, Khehra BS, Sharma A. Soft Computing based object detection and tracking approaches: State-of-the-Art survey. Appl Soft Comput 2018;70:423–464.

[18] Chartrand G, Cheng PM, Vorontsov E, Drozdzal M, Turcotte S, Pal CJ, et al. Deep Learning: A Primer for Radiologists. Radiographics 2017;37(7):2113–2131.

[19] Bobick AF, Davis JW. The recognition of human movement using temporal templates. IEEE T Pattern Anal 2001;23(3):257–267.

[20] Zhang S, Wei Z, Nie J, Huang L, Wang S, Li Z. A Review on Human Activity Recognition Using Vision-Based Method. J Healthc Eng 2017;p. 3090343.

[21] Sargano A, Angelov P, Habib Z. A Comprehensive Review on Handcrafted and Learning-Based Action Representation Approaches for Human Activity Recognition. Appl Sci 2017;7(1).

[22] Koch C, Georgieva K, Kasireddy V, Akinci B, Fieguth P. A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. Adv Eng Inform 2015;29(2):196–210.

[23] Duchesne C, Liu JJ, MacGregor JF. Multivariate image analysis in the process industries: A review. Chemom Intell Lab Syst 2012;117:116–128.

[24] Moran U, Phillips R, Milo R. SnapShot: key numbers in biology. Cell 2010;141(7):1262–1262 e1.

[25] Manchado E, Guillamot M, Malumbres M. Killing cells by targeting mitosis. Cell Death Differ 2012;19(3):369–77.

[26] Florian S, Mitchison TJ. Anti-Microtubule Drugs. Method Mol Biol 2016;1413:403–21.

[27] McIntosh JR, editor. Special Issue "Mechanisms of Mitotic Chromosome Segregation". Biology, MDPI; 2017.

[28] Rieder CL, Khodjakov A. Mitosis through the microscope: advances in seeing inside live dividing cells. Science 2003;300(5616):91–6.

[29] Cireşan DC, Giusti A, Gambardella LM, Schmidhuber J. Mitosis Detection in Breast Cancer Histology Images with Deep Neural Networks. In: Mori K, Sakuma I, Sato Y, Barillot C, Navab N, editors. Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013 Springer Berlin Heidelberg; 2013. p. 411–418.

[30] Potapova T, Gorbsky GJ. The Consequences of Chromosome Segregation Errors in Mitosis and Meiosis. Biology (Basel) 2017;6(1).

[31] Sivakumar S, Gorbsky GJ. Spatiotemporal regulation of the anaphase-promoting complex in mitosis. Nat Rev Mol Cell Biol 2015;16(2):82–94.

[32] Banfalvi G. Overview of Cell Synchronization. Method Mol Biol 2017;1524:3–27.

[33] Wang H, Roa AC, Basavanhally AN, Gilmore HL, Shih N, Feldman M, et al. Mitosis detection in breast cancer pathology images by combining handcrafted and convolutional neural network features. J Med Imaging 2014;1(3):034003.

[34] Hamidinekoo A, Denton E, Rampun A, Honnor K, Zwiggelaar R. Deep learning in mammography and breast histology, an overview and future trends. Med Image Anal 2018;47:45–67.

[35] Veta M, van Diest PJ, Willems SM, Wang H, Madabhushi A, Cruz-Roa A, et al. Assessment of algorithms for mitosis detection in breast cancer histopathology images. Med Image Anal 2015;20(1):237–48.

[36] Wollmann T, Erfle H, Eils R, Rohr K, Gunkel M. Workflows for microscopy image analysis and cellular phenotyping. J Biotechnol 2017;261:70–75.

[37] Fillbrunn A, Dietz C, Pfeuffer J, Rahn R, Landrum GA, Berthold MR. KNIME for reproducible cross-domain analysis of life science data. J Biotechnol 2017;261:149–156.

[38] McQuin C, Goodman A, Chernyshev V, Kamentsky L, Cimini BA, Karhohs KW, et al. CellProfiler 3.0: Next-generation image processing for biology. PLoS Biol 2018;16(7):e2005970.

[39] Harder N, Mora-Bermúdez F, Godinez WJ, Wünsche A, Eils R, Ellenberg J, et al. Automatic analysis of dividing cells in live cell movies to detect mitotic delays and correlate phenotypes in time. Genome Res 2009;19(11):2113–2124.

[40] Held M, Schmitz MHA, Fischer B, Walter T, Neumann B, Olma MH, et al. CellCognition: time-resolved phenotype annotation in high-throughput live cell imaging. Nat Methods 2010;7(9):747–754.

[41] Moen E, Bannon D, Kudo T, Graf W, Covert M, Van Valen D. Deep learning for cellular image analysis. Nat Methods 2019;16(12):1233–1246.

[42] Nagao Y, Sakamoto M, Chinen T, Okada Y, Takao D. Robust classification of cell cycle phase and biological feature extraction by image-based deep learning. Mol Biol Cell 2020;31(13):1346–1354.

[43] Sommer C, Gerlich DW. Machine learning in cell biology - teaching computers to recognize phenotypes. J Cell Sci 2013;126(Pt 24):5529–39.

[44] Orlov N, Shamir L, Macura T, Johnston J, Eckley DM, Goldberg IG. WND-CHARM: Multi-purpose image classification using compound image transforms. Pattern Recogn Lett 2008;29(11):1684–1693.

[45] Balluet M, Pont J, Giroux B, Bouchareb O, Chanteux O, Tramier M, et al., Method for managing command blocks for a microscopy imaging system, corresponding computer program, storage means and device; 2020. Patent WO2021170565 A1.

[46] Neumann B, Walter T, Heriche JK, Bulkescher J, Erfle H, Conrad C, et al. Phenotypic profiling of the human genome by time-lapse microscopy reveals cell division genes. Nature 2010;464(7289):721–7.

[47] Bishop CM. Pattern recognition and machine learning. Information science and statistics, New York: Springer; 2006.

[48] CellCognition, Demo data "Chromatin + Microtubles"; 2010. https://cellcognition-project.org/demo_data.html.

[49] Otsu N. Threshold selection method from gray-level histograms. IEEE T Syst Man Cyb 1979;9(1):62–66.

[50] Jacob B, Kligys S, Chen B, Zhu M, Tang M, Howard A, et al. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2018. p. 2704–2713.

[51] Molchanov P, Tyree S, Karras T, Aila T, Kautz J, Pruning Convolutional Neural Networks for Resource Efficient Inference; 2016. ArXiv. 1611.06440.

[52] Fisher RA. The Use of Multiple Measurements in Taxonomic Problems. Ann Eugenic 1936;7:179–188.

[53] Duda R, Hart P. Pattern classification and scene analysis. Philadelphia: Wiley; 1973.

[54] Muller K, Mika S, Ratsch G, Tsuda K, Scholkopf B. An introduction to kernel-based learning algorithms. IEEE T Neural Networ 2001;12(2):181–201.

[55] Belhumeur PN, Hespanha JP, Kriegman DJ. Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. IEEE T Pattern Anal 1997;19(7):711–720.

[56] Liyang W, Yongyi Y, Nishikawa RM, Yulei J. A study on several Machine-learning methods for classification of Malignant and benign clustered microcalcifications. IEEE T Neural Networ 2005;24(3):371–380.

[57] Chiang LH, Russell EL, Braatz RD. Fault diagnosis in chemical processes using Fisher discriminant analysis, discriminant partial least squares, and principal component analysis. Chemom Intell Lab Syst 2000;50(2):243–252.

[58] Shaikhina T, Lowe D, Daga S, Briggs D, Higgins R, Khovanova N. Machine Learning for Predictive Modelling based on Small Data in Biomedical Engineering. In: 9th IFAC Symposium on Biological and Medical Systems BMS 2015, vol. 48(20); 2015. p. 469–474.

[59] Kourou K, Exarchos TP, Exarchos KP, Karamouzis MV, Fotiadis DI. Machine learning applications in cancer prognosis and prediction. Comput Struct Biotechnol J 2015;13:8–17.

[60] Foster KR, Koprowski R, Skufca JD. Machine learning, medical diagnosis, and biomedical engineering research - commentary. BioMed Eng OnLine 2014;13:94.

[61] Fawcett T. An introduction to ROC analysis. Pattern Recogn Lett 2006;27(8):861–874.

[62] Tuv E, Borisov A, Runger G, Torkkola K. Feature Selection with Ensembles, Artificial Variables, and Redundancy Elimination. J Mach Learn Res 2009;10(45):1341–1366.

[63] Breiman L. Random Forests. Mach Learn 2001;45(1):5–32.

[64] Loh WY, Shih YS. Split selection methods for classification trees. Stat Sinica 1997;7(4):815–840.

[65] Zhao Z, Anand R, Wang M, Maximum Relevance and Minimum Redundancy Feature Selection Methods for a Marketing Machine Learning Platform; 2019. ArXiv. 1908.05376.

[66] Itseez, Open Source Computer Vision Library; 2015. https://github.com/itseez/opencv.

[67] Elting MW, Suresh P, Dumont S. The Spindle: Integrating Architecture and Mechanics across Scales. Trends Cell Biol 2018;28(11):896–910.

[68] Bolón-Canedo V, Sánchez-Maroño N, Alonso-Betanzos A. A review of feature selection methods on synthetic data. Knowl Inf Syst 2012;34(3):483–519.

[69] Berthelier A, Chateau T, Duffner S, Garcia C, Blanc C. Deep Model Compression and Architecture Optimization for Embedded Systems: A Survey. J Signal Process Sys 2020;93(8):863–878.

[70] Pollard TD, Earnshaw WC. Cell biology. Philadelphia: Saunders; 2002.

[71] Farhadifar R, Ponciano JM, Andersen EC, Needleman DJ, Baer CF. Mutation is a sufficient and robust predictor of genetic variation for mitotic spindle traits in Caenorhabditis elegans. Genetics 2016;203(4):1859–70.

[72] Esteva A, Chou K, Yeung S, Naik N, Madani A, Mottaghi A, et al. Deep learning-enabled medical computer vision. NPJ Digit Med 2021;4(1):5. https://www.ncbi.nlm.nih.gov/pubmed/33420381.

[73] Lau RP, Kim TH, Rao J. Advances in Imaging Modalities, Artificial Intelligence, and Single Cell Biomarker Analysis, and Their Applications in Cytopathology. Front Med (Lausanne) 2021;8:689954. https://www.ncbi.nlm.nih.gov/pubmed/34277664.

[74] Waithe D, Brown JM, Reglinski K, Diez-Sevilla I, Roberts D, Eggeling C. Object detection networks and augmented reality for cellular detection in fluorescence microscopy. J Cell Biol 2020;219(10). https://www.ncbi.nlm.nih.gov/pubmed/32854116.

[75] Eulenberg P, Kohler N, Blasi T, Filby A, Carpenter AE, Rees P, et al. Reconstructing cell cycle and disease progression using deep learning. Nat Commun 2017;8(1):463. https://www.ncbi.nlm.nih.gov/pubmed/28878212.

[76] Ronneberger O, Fischer P, Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab N, Hornegger J, Wells WM, Frangi AF, editors. Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015 Springer International Publishing; 2015. p. 234–241.

[77] Ying X. An Overview of Overfitting and its Solutions. J Phys: Conf Ser 2019;1168.

[78] Feng S, Zhou H, Dong H. Using deep neural network with small dataset to predict material defects. Mater Design 2019;162:300–310.

[79] Pasupa K, Sunhem W. A comparison between shallow and deep architecture classifiers on small dataset. In: 8th International Conference on Information Technology and Electrical Engineering (ICITEE); 2016. p. 1–6.

[80] Azar AT, El-Metwally SM. Decision tree classifiers for automated medical diagnosis. Neural Comput Appl 2012;23(7-8):2387–2403.

[81] Borisov A, Eruhimov V, Tuv E. Tree-Based Ensembles with Dynamic Soft Feature Selection. In: Feature Extraction: Foundations and Applications Berlin, Heidelberg: Springer Berlin Heidelberg; 2006. p. 359–374.

Supplemental figures

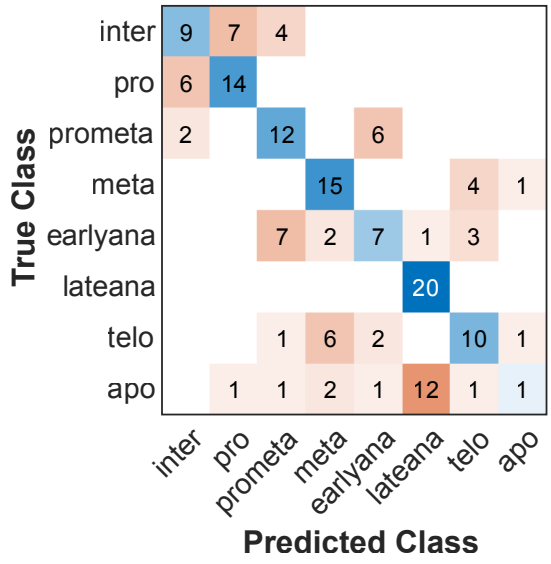


FIGURE S1 Classification using a single feature (Otsu-segmented-region area) resulted in a poor confusion matrix. Class names are abbreviated after Fig. 1a. CellCognition dataset was used (see Methods §2.1).

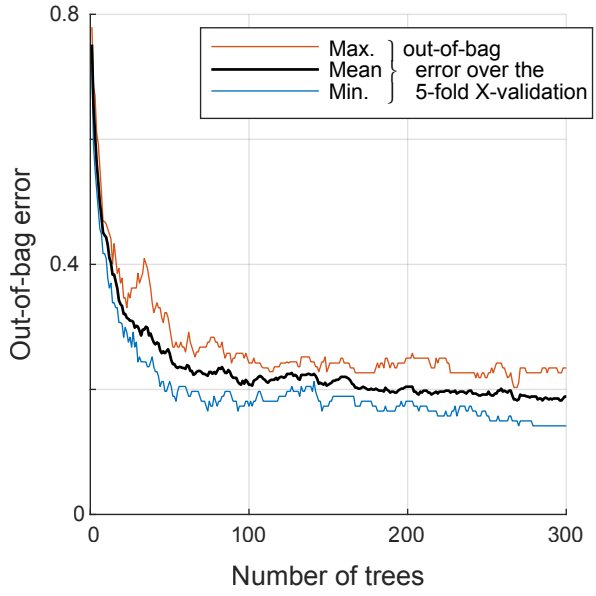


FIGURE S2 Selecting the number of trees in random forests classifier by plotting the out-of-bag error versus the number of trees. The black, blue and red lines depict the average, minimum and maximum out-of-bag errors, respectively, over the 5-fold iterations of the cross-validation. CellCognition dataset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

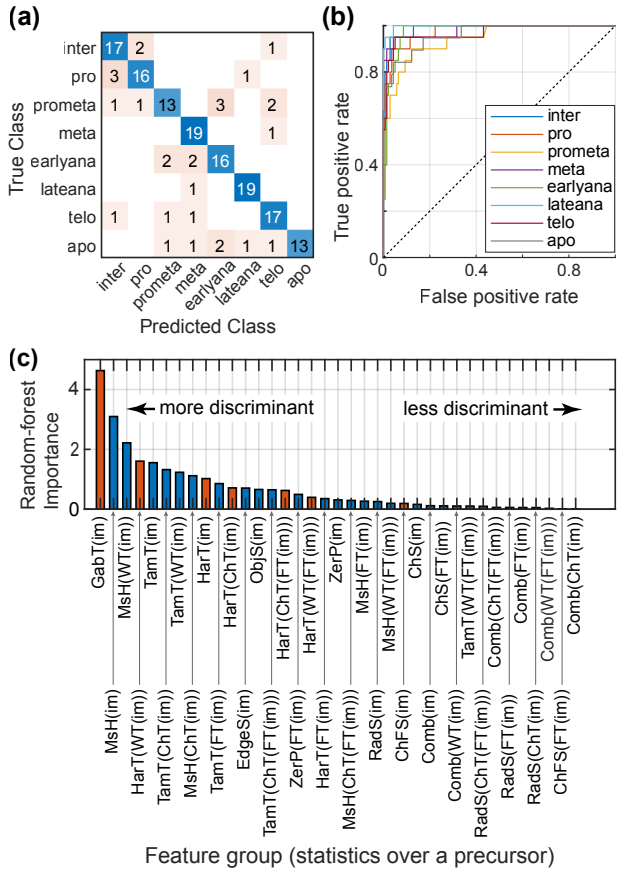


FIGURE S3 **Random forests using all the 1025 features** was trained and tested over 20% of the dataset to get **(a)** the confusion matrix and **(b)** the ROC curves over the 5-fold cross-validation using the CellCognition dataset (see Methods §2.1). Class names are abbreviated after Fig. 1a.

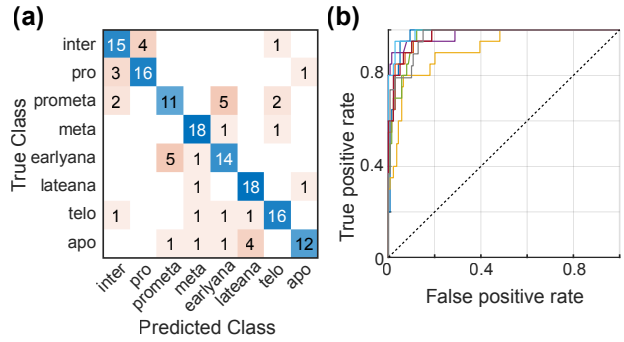


FIGURE S4 **Random forests with computationally intensive features** optimised by removing low importance feature groups. The algorithm was trained and tested over 20% of the dataset to get **(a)** the confusion matrix and **(b)** the ROC curves over the 5-fold cross-validation using the CellCognition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

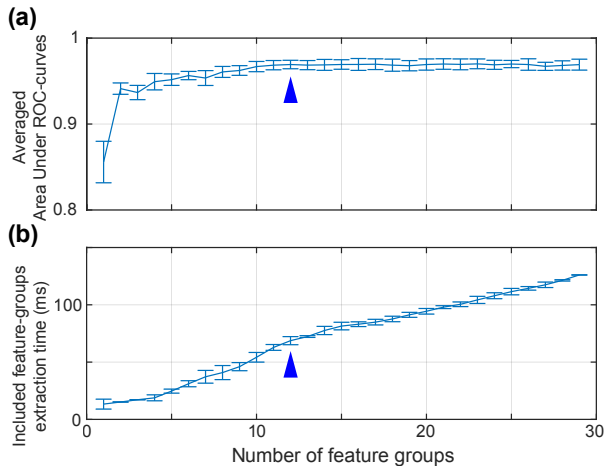


FIGURE S5 Bootstrapping the random forests optimised with only non-computationally-intensive feature-groups. **(a)** The Area Under Curve (AUC) was averaged over the classes, and **(b)** execution time for extracting the feature-groups included in the classification was assessed. Both quantities are 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.

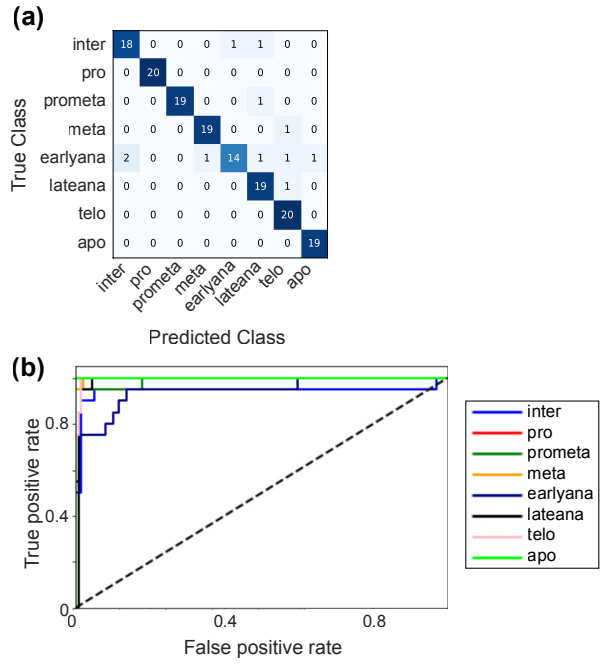


FIGURE S6 **Classification using Deep learning. (a)** Confusion matrix and **(B)** corresponding ROC curves, averaged over the 5-fold cross-validation for the classification over the *CellCognition* dataset (see Methods §2.1). Class names are abbreviated after Fig. 1a.

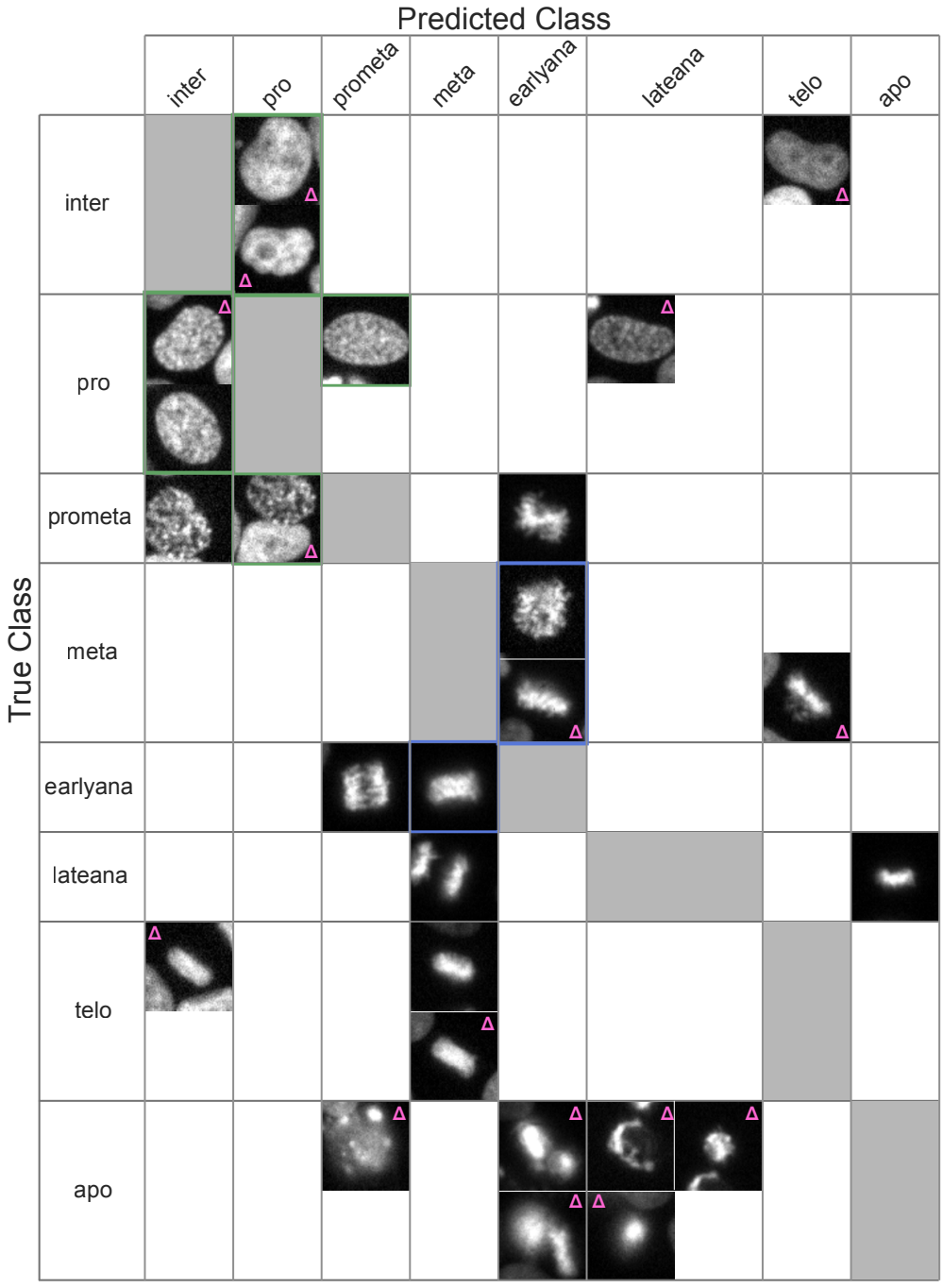


FIGURE S7 Mis-classified images using optimised random forests over Cell cognition. Cell pictures misclassified were reported for the 5-fold cross-validation over the CellCognition dataset (see Methods §2.1). The table mimics the confusion matrix (Fig. 5c). Rows correspond to true classes while columns to predicted classes. Green and blue frames highlight images misclassified to the class just before or after the real one in the order of the cycle or mitotic phases. Purple triangles depict frames where neighbouring cells appear, likely confusing the classifier. Class names are abbreviated after Fig. 1a.