

## METHODS &amp; TECHNIQUES

# A fully automated deep learning pipeline for high-throughput colony segmentation and classification

Sarah H. Carl<sup>1,2,\*</sup>, Lea Duempelmann<sup>1,3</sup>, Yukiko Shimada<sup>1</sup> and Marc Bühler<sup>1,3,\*</sup>

## ABSTRACT

Adenine auxotrophy is a commonly used non-selective genetic marker in yeast research. It allows investigators to easily visualize and quantify various genetic and epigenetic events by simply reading out colony color. However, manual counting of large numbers of colonies is extremely time-consuming, difficult to reproduce and possibly inaccurate. Using cutting-edge neural networks, we have developed a fully automated pipeline for colony segmentation and classification, which speeds up white/red colony quantification 100-fold over manual counting by an experienced researcher. Our approach uses readily available training data and can be smoothly integrated into existing protocols, vastly speeding up screening assays and increasing the statistical power of experiments that employ adenine auxotrophy.

**KEY WORDS:** Deep learning, Neural networks, Adenine auxotrophy, Yeast, Growth assay

## INTRODUCTION

Auxotrophy is the inability of an organism to synthesize a particular organic compound required for its growth. For example, yeasts with mutations in the adenine biosynthetic pathway cannot grow on media lacking adenine. When grown under adenine-limiting conditions, adenine auxotrophs grow but accumulate a cell-limited red pigment in their vacuoles, whereas wild-type cells grow white. This red/white colony color assay has been widely used over the last few decades for the investigation of many biological processes, such as recombination, copy number, chromosome loss, plasmid stability, prion propagation, or epigenetic gene regulation in both budding and fission yeasts. However, adapting this assay to quantitative high-throughput applications has proven challenging, as it requires extensive scoring of colony phenotypes by eye. In addition to being time-consuming and tedious, manual colony scoring may suffer from inaccuracy and irreproducibility. Nonetheless, up to now manual scoring is a common practice in the yeast community. Modern machine-learning techniques such as deep learning have made huge strides in automated image classification in recent years and are beginning to be applied to

previously intractable problems in the biomedical imaging domain. We set out to leverage these recent developments to build a computational pipeline that would enable fully automated high-throughput adenine auxotrophy-based screening and quantification.

Typically, red/white colony color assays start with plating individual yeast cells on limiting adenine indicator plates and allowing them to grow until they form colonies large enough to be inspected by eye. Each plate may represent an independent condition or genotype, the penetrance of which can be assessed by quantifying the percentage of non-white colonies per plate. In order to create a pipeline that would fit into existing protocols as seamlessly as possible, we considered our input to be images of plates and our desired output to be percentages of white and non-white colonies per plate. Two major tasks are required to generate this output: separating the colonies from the plate background and classifying them individually as white or non-white.

These two tasks could conceivably be completed either in one step, as with a single-shot detector (Liu et al., 2016) or RetinaNet (Lin et al., 2017), or in two separate steps, such as with a semantic segmentation, where each pixel is assigned a label such as ‘foreground’ or ‘background’, followed by classification of cropped images. While a single-step approach may be preferable from the perspective of algorithmic efficiency and speed (Huang et al., 2016), the training data annotations are more complex, requiring both manually assigned labels and matched bounding boxes identifying the location of each colony on a plate. As insufficient training data is a common problem hampering efforts to apply deep learning in many biological domains (Hughes et al., 2018), we opted to use a pragmatic approach, treating the segmentation and classification steps as separate problems (Fig. 1A). This allowed us to use simpler and, when available, pre-existing annotations for training data: for the segmentation task, we used masks generated previously using the Ilastik image-processing toolkit (Sommer et al., 2011), while for the classification task, we relied on manual labels assigned by experienced biologists to cropped images of single colonies. All of our training data consisted of *Schizosaccharomyces pombe* colonies, with red pigment resulting from heterochromatin-mediated silencing of the *ade6<sup>+</sup>* gene.

## RESULTS

### Implementation of the pipeline


All analyses were performed in Python v. 3.6.5 inside a conda virtual environment (v. 4.5.4). Deep learning models were built and trained using the fast.ai library (<https://github.com/fastai/fastai>) (Howard and others, 2018). Image processing steps were performed using the scikit-image library (v. 0.13.1) (<https://scikit-image.org>) (van der Walt, 2014). Model training and prediction was run on a GeForce GTX 1080 GPU with CUDA v. 9.0 and CUDNN v. 7.1.2.

### Image segmentation

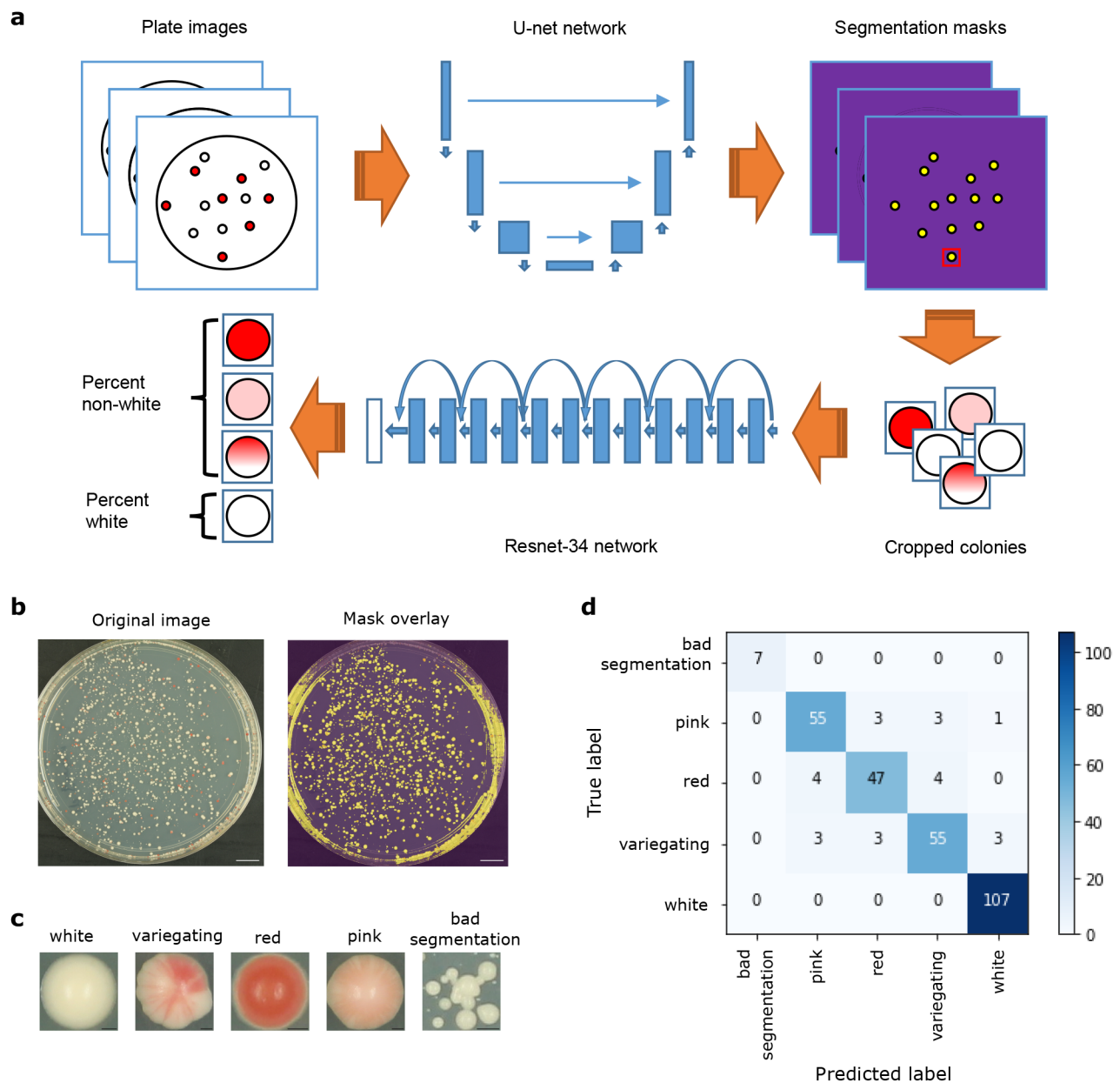
A modified U-net architecture using a Resnet-34 network pre-trained on ImageNet as the encoder was used as the network

<sup>1</sup>Friedrich Miescher Institute for Biomedical Research, Maulbeerstrasse 66, 4058 Basel, Switzerland. <sup>2</sup>SIB Swiss Institute of Bioinformatics Quartier Sorge - Batiment Amphipole 1015, Lausanne, Switzerland. <sup>3</sup>University of Basel, Petersplatz 10, 4003 Basel, Switzerland.

\*Authors for correspondence (marc.buehler@fmi.ch, sarahhcarl@gmail.com)

 S.H.C., 0000-0003-3927-2559; L.D., 0000-0002-5775-5186; M.B., 0000-0001-6661-9795

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0>), which permits unrestricted use, distribution and reproduction in any medium provided that the original work is properly attributed.



**Fig. 1. Overview of deep learning setup and results.** (A) Schematic of the entire automated colony classification pipeline. Plate images are given as input, then pass through a U-net network for segmentation prediction, resulting in cropped colony images. These are then fed into a Resnet-34 network for classification, followed by plate-level aggregation into white and non-white percentages. (B) An example input plate image (left), and the same image overlaid with the predicted segmentation mask (right). Scale bars: 1 cm. (C) Examples of cropped colonies classified into each of the five possible classes. Scale bars: 0.3 mm. (D) Confusion matrix showing the results of the classification step (Resnet-34 network) on the validation data. Numbers in each square indicate the number of colonies with each true (y-axis) and predicted (x-axis) label.

architecture for the plate image segmentation task (He et al., 2016; Ronneberger et al., 2015). A total of 492 pairs of plate images and corresponding segmentation masks generated by Ilastik were used as training data, with approximately 20% (95 pairs) set aside for validation. Binary cross-entropy with logits ('BCEWithLogitsLoss') was used as a loss function, and the dice coefficient was used as an evaluation metric. Data augmentation was applied to the training images with the following transformations: random rotation up to  $4^\circ$ , random horizontal flip, random adjustment of brightness and contrast. The same transformations, except for adjustment of brightness and contrast, were applied to the training masks.

The network was first trained with center-cropped masks and images resized to  $512 \times 512$  pixels, with a batch size of 4. A weight decay parameter of  $1e-7$  was used for all training. First, only the last layer was trained using stochastic gradient descent with restarts (SGDR) for 1 cycle of 8 epochs, using a circular learning rate scheduler with a maximum learning rate of  $4e-2$ , a minimum learning rate of  $8e-3$ , 1 epoch of increasing learning rate, and 7 epochs of decreasing learning rate (Huang et al., 2017; Smith, 2015). Next, all layers were trained using SGDR for 1 cycle of 20 epochs. Differential learning rates were applied across layers, with the first third of layers having a maximum learning rate of  $1e-4$ , the middle third having a maximum learning rate of  $1e-3$ , and the last

third having a maximum learning rate of  $1e-2$ . A circular learning rate scheduler was again used, with minimum learning rates of one-twentieth of respective maximum learning rates, 2 epochs of increasing learning rates, and 18 epochs of decreasing learning rates. The resulting weights were then saved and used as a starting point to train the network with larger,  $1024 \times 1024$  images.

Training images and masks were scaled up to  $1024 \times 1024$  pixels and the network was further trained. First, only the last layer was trained using SGDR for one cycle of 2 epochs, using a circular learning rate scheduler with a maximum learning rate of  $4e-2$ , a minimum learning rate of  $8e-3$ , 0.5 epochs of increasing learning rate, and 1.5 epochs of decreasing learning rate. Next, all layers were trained using SGDR for 20 cycles of 20 epochs. Differential learning rates were applied across layers, with the first third of layers having a maximum learning rate of  $4e-5$ , the middle third having a maximum learning rate of  $2e-4$ , and the last third having a maximum learning rate of  $4e-3$ . A circular learning rate scheduler was again used, with minimum learning rates of one-twentieth of respective maximum learning rates, 2.5 epochs of increasing learning rates per cycle, and 17.5 epochs of decreasing learning rates per cycle. The resulting weights were saved and used for prediction.

### Colony classification

A Resnet-34 network that had been pre-trained on ImageNet was used as the network architecture for the colony classification task. The final output layer was replaced with a layer predicting five classes ('white', 'red', 'pink', 'variegating' and 'bad segmentation'). A total of 1476 manually labeled, cropped images of individual colonies were used as training data, with approximately 20% (295 images) set aside for validation. Out of the total pool of images, 537 were labeled as white, 273 as red, 310 as pink, 318 as variegating, and 38 as bad segmentation. Validation images were chosen so as to have the same proportions among the five classes as in the total pool. For training, images were resized to  $50 \times 50$  pixels and a batch size of 128 was used. Stochastic gradient descent (SGD) with a Momentum of 0.9 was used as an optimization algorithm. Categorical cross-entropy was used as a loss function, and both log loss and accuracy were used as evaluation metrics. Data augmentation was applied to the training images with the following transformations: random rotation up to  $10^\circ$ , random rotation or reflection of dihedral group 8, random adjustment of brightness and contrast, random zoom up to  $1.1 \times$ .

The last layer of the network was first trained without data augmentation for 1 epoch using a learning rate of  $1e-2$ . Data augmentation was then added, and it was trained with SGDR for three cycles of 1 epoch each using cosine annealing, with an initial learning rate of  $1e-2$ . Next, all layers were trained for 17 sets of three cycles of increasing length (1 epoch, followed by 2 epochs, followed by 4 epochs), for a total of 51 cycles and 119 epochs. Differential learning rates were applied across layers, with the first third of layers having a maximum learning rate of  $1.1e-4$ , the middle third having a maximum learning rate of  $3.3e-4$ , and the last third having a maximum learning rate of  $1e-3$ . Training was stopped when over-fitting was observed, and the resulting weights were saved and used for prediction.

### Mask prediction, post-processing and colony class prediction

The full prediction pipeline is implemented as follows: first, plate images are center-cropped and resized to  $1024 \times 1024$  pixels. Segmentation masks are predicted using the trained U-net

network, then resized to match the dimensions of the original (center-cropped) images. Border clearing and morphological opening are applied to the masks, reducing plate edge artefacts. Individual colonies are then labelled and a bounding box is drawn around each one, defining a region. Finally, colonies are selected if they have a regional eccentricity  $\leq 0.6$  and a regional area  $\geq 400$  pixels. Selected colonies are cropped and saved as individual .jpg images, retaining information about which plate each colony came from.

Classes are then predicted for individual colonies using the trained Resnet-34 network with test-time augmentation. This applies the same data augmentation transforms as were used during training to the test images, creating four randomly transformed versions of each image, then takes the average prediction for all four plus the original. The five colony class predictions are then aggregated per plate, and the percentage of non-white colonies is defined as the sum of predicted red, pink and variegating colonies divided by the sum of all properly segmented colonies (excluding those in the bad segmentation class). Segmentation and colony class prediction can also be performed separately, allowing for classification of previously-segmented images.

### Performance of the pipeline

To perform the segmentation task, we chose a U-net-like architecture implemented in the fast.ai library (Howard and others, 2018). U-net was developed specifically for semantic segmentation and has been successfully applied to complex biomedical images such as electron microscopy of neuronal structures and MRI or ultrasound images in breast cancer screening (Ronneberger et al., 2015; Kumar et al., 2018; Dalmış et al., 2017). We trained the U-net network using 492 plate images and corresponding Ilastik-generated masks, with 20% kept aside for validation (see Image segmentation for full training parameters). After training, visual inspection of predicted masks revealed an accurate segmentation of colonies from background, although some errors remained around plate edges (Fig. 1B). This was not unexpected, considering that the Ilastik-generated masks also often contained artefacts at the edges. In order to circumvent this problem, we applied post-processing on the predicted masks, which effectively removed artefacts. The vast majority of resulting cropped regions contained a single colony; however, a few regions still contained multiple small, overlapping colonies. To reduce possible bias that might result from counting multiple colonies as one, we filtered these out during the classification stage.

For the classification task, we fine-tuned a Resnet-34 architecture that was pre-trained on ImageNet (<http://www.image-net.org/>) (He et al., 2016; Deng et al., 2009), also implemented in the fast.ai library (Howard and others, 2018). We trained the network using 1476 images of individually cropped colonies, which were split into five manually-labeled classes: white, red, pink, variegating and multiple colonies. Again, 20% of colony images were kept aside for validation. After training, we achieved a validation accuracy of 91.8% across the five classes (Fig. 1D, Table 1). Further, aggregating the three classes of non-white colonies together (red, pink and variegating) yielded a much higher validation accuracy of 98.6%. This higher pooled accuracy was encouraging, considering our desired output of percentages of white and non-white colonies per plate. It also demonstrates that most classification errors occur within non-white classes rather than between white and non-white classes, an expected result given that the non-white classes often do not have clear distinctions and can be difficult to define even by eye.



**Table 1. Detailed performance metrics for colony classification in validation dataset**

Class	Sensitivity	Specificity	F1-score
White	1.0	0.98	0.99
Non-white	0.98	1.0	0.99
Pink	0.89	0.97	0.93
Red	0.86	0.97	0.91
Variegating	0.86	0.97	0.91

Class labels were predicted for  $n=295$  colonies and compared against manual annotations. Metrics are shown for individual class labels (white, pink, red and variegating), as well as for the aggregated non-white class, which includes pink, red and variegating colonies. For each class, true positives (TP) are considered to be colonies that were predicted to have a given label and also manually annotated with that label. False positives (FP) are considered to be colonies that were predicted to have a given label, but manually annotated with a different label. True negatives (TN) are considered to be colonies that were predicted to have any label besides a given label and also manually annotated to have any label besides that label. False negatives (FN) are considered to be colonies that were predicted to have any label besides a given label, but manually annotated with that label. Sensitivity was calculated as  $TP/(TP+FN)$ . Specificity was calculated as  $TN/(TN+FP)$ . The F1-score represents the harmonic mean of sensitivity and specificity and was calculated as  $2*(sensitivity * specificity)/(sensitivity+specificity)$ .

While we were encouraged by our high validation accuracy, we wanted to test the pipeline's performance against manual counting in a real-world, experimental context. To this end, we took data from two published experiments testing trans-generational inheritance of *ade6*<sup>+</sup> silencing in *Schizosaccharomyces pombe* (Duempelmann et al., 2019). In these experiments, *ade6*<sup>+</sup> silencing was first induced by expression of small interfering RNAs (siRNAs) that are complementary to the *ade6*<sup>+</sup> gene in a *paf1-Q264Stop* nonsense-mutant background, leading to red colonies. Paf1 is a subunit of the Paf1 complex (Paf1C), which represses siRNA-induced heterochromatin formation in *S. pombe* (Kowalik et al., 2015). In the presence of the *paf1-Q264Stop* allele, the silenced (red) phenotype was inherited through meiosis, even in the absence of the original siRNAs that have triggered *ade6*<sup>+</sup> repression. This was not the case if the progeny inherited a *paf1*<sup>+</sup> wild-type allele, i.e. the red silencing phenotype was lost. However, these white *paf1*<sup>+</sup> cells inherited a marked *ade6*<sup>+</sup> epiallele (*ade6*<sup>si3</sup>), which reinstated silencing when cells became mutant for Paf1 again in subsequent generations (Duempelmann et al., 2019). The following experiments were performed to quantify different aspects of this phenomenon.

In the first experiment, *paf1-Q264Stop* cells that inherited the *ade6*<sup>si3</sup> allele and re-established the red silencing phenotype were plated on limiting adenine indicator plates to quantify the maintenance of this re-established silencing through mitosis. Out of 59 plates derived from a red progenitor colony, our automated pipeline predicted a mean of 84.7% non-white colonies, indicating a high degree of maintenance of *ade6*<sup>+</sup> silencing. Ten of these plates were manually counted, and the mean difference in the percent of counted and predicted non-white colonies was 2.1 percentage points. In comparison, many red colonies (43%) were falsely classified as white when using the popular open-source software *CellProfiler* (github.com/CellProfiler/CellProfiler/tree/v3.1.9) (Fig. 2A). As a control, colonies derived from cells of white *paf1-Q264Stop* progenitor colonies were also quantified by both methods. Our automated pipeline predicted a mean of 0.57% non-white colonies across 60 plates, while manual counting of colonies on 12 plates detected 0 non-white colonies. Also, here the predictions made with *CellProfiler* were less accurate and misclassified 14.66% of the white colonies as non-white (Fig. 2B).

The second experiment was performed to assess mitotic stability of the *ade6*<sup>si3</sup> epiallele. White *paf1*<sup>+</sup> cells with an *ade6*<sup>si3</sup> epiallele were grown exponentially and a sample was crossed to white *paf1-Q264Stop* cells every 3 days (30–40 mitotic divisions) over 19 days total. Progeny of these crosses were also observed to re-establish silencing; however, the frequency of re-establishment declined with the number of mitotic divisions the cells had gone through. The percentage of non-white colonies on plates resulting from crosses at each time point was both counted manually (66,000 colonies total) and predicted using the automated pipeline. Both methods showed a near-exponential decrease in non-white colonies over time; the mean difference in the percent of non-white colonies between the two methods ranged from 0.26 to 5.1 percentage points (Fig. 2C).

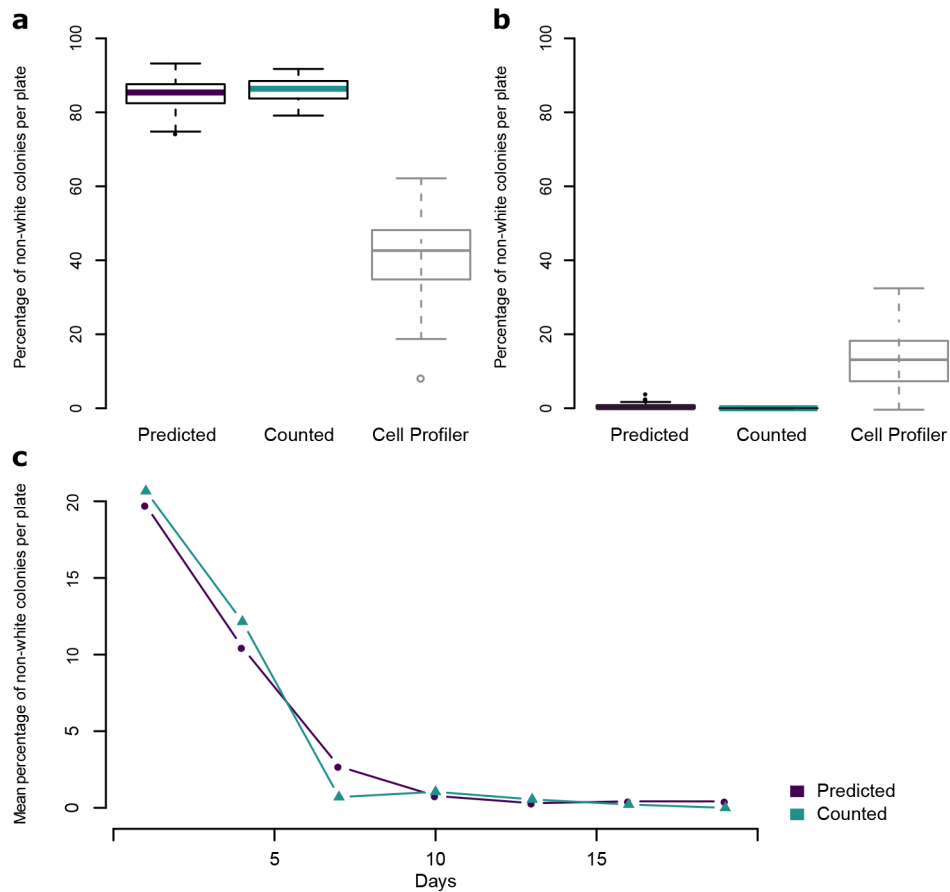
## DISCUSSION

Here we introduce a novel computational pipeline for colony segmentation and classification. It allows us to achieve accuracy comparable to human performance, which was difficult to achieve with existing colony segmentation and classification programs. For example, many red colonies were falsely classified as white when analyzing our data with *CellProfiler* (Fig. 2 and data not shown). *CellProfiler* also cannot distinguish variegating, pink and red colonies.

We observed several factors that contributed to the accuracy of the automated pipeline predictions versus manual counting. In general, the pipeline performed with very high accuracy on plates with a low (<5%) percentage of non-white colonies; however, on some plates that had been grown for over 2 weeks, white colonies formed an irregular ring-like morphology and tended to be mis-classified as pink. The accuracy of the pipeline was also decreased for plates with very dense, small colonies. This may be partly due to the difficulty in segmenting individual colonies when they are touching one another, leading to many colonies being excluded from the analysis. Very small colonies are also more likely to be mistakenly filtered out during post-processing. Since cells with a silenced *ade6*<sup>+</sup> gene tend to grow more slowly, this may lead to bias, as very small colonies are more likely to be non-white. However, plating cells at a controlled density and imaging the plates after an appropriate amount of time can counteract this potential bias. We have posted a full suggested protocol describing plating and imaging for most accurate prediction on Protocol Exchange (<https://protocolexchange.researchsquare.com/article/nprot-7305/v1>).

Our pipeline generates as output the numbers of predicted colonies per plate for both the aggregated non-white versus white categories as well as for the more granular classes of red, pink and variegating colonies. As detailed in Table 1, we observed a higher overall performance for white versus non-white predictions; however, in some cases researchers may prefer to use the more granular predictions despite their lower accuracy. This decrease in accuracy is largely due to reduced sensitivity for individual class predictions, while a high level of specificity is maintained.

Our fully automated pipeline can be run on a CUDA-enabled CPU or GPU. On a GeForce GTX 1080 GPU, it took approximately 45 minutes to process 245 plate images, representing an 80–100x speedup over manual counting by an experienced biologist. In addition to saving time and manual labor, the pipeline has the potential to increase reproducibility by removing variations between individual researchers or computer monitors. Our pragmatic approach, combining transfer learning with readily available annotations, allows us to achieve accuracy comparable to human performance with relatively little training data. Our work should



**Fig. 2. Comparison of automated colony classification and manual counting on experimental data.** (A) Boxplots showing the predicted or manually counted percentage of non-white colonies per plate on plates resulting from red *paf1-Q264Stop; ade6<sup>si3</sup>* cells.  $n=59$  (predicted) or  $n=10$  (counted). For comparison, percentage of non-white colonies per plate were also predicted with *CellProfiler*, with a threshold value of 0.055 ( $n=59$ ). (B) Boxplots showing the predicted or manually counted percentage of non-white colonies per plate on plates resulting from white *paf1-Q264Stop* cells.  $n=60$  (predicted) or  $n=12$  (counted). For comparison, percentage of non-white colonies per plate were also predicted with *CellProfiler*, with a threshold value of 0.055 ( $n=60$ ). For boxplots, center line is median, bottom and top hinges are first and third quartiles, whiskers show the most extreme points within 1.5 times the interquartile range, and more extreme values are plotted as individual points. (C) Mean predicted or manually counted percentages of non-white colonies per plate across a time course of mitotically dividing white *paf1<sup>+</sup> ade6<sup>si3</sup>* cells crossed to white *paf1-Q264Stop* cells every 3 days. The time course was repeated with 11 independent biological replicates, and six plates per replicate per timepoint were quantified by automated prediction ( $n=66$  per time point). The mean of all 66 plates is reported for each timepoint. For manual counting, one plate per replicate per timepoint was counted ( $n=11$  per time point); the mean of 11 plates is reported for each timepoint.

thus enable larger-scale experiments and higher statistical power, unlocking a true quantitative use of the red/white color assay in yeast research. We have made the code and trained network weights freely available to the community at <https://github.com/fmi-basel/buehler-colonyclassification>.

## MATERIALS AND METHODS

### Protocol for plating and imaging of yeast

We have posted a full suggested protocol describing plating and imaging for most accurate prediction on Protocol Exchange (<https://protocolexchange.researchsquare.com/article/nprot-7305/v1>).

#### Pipeline requirements

- Operating system(s): Platform independent
- Programming language: Python
- Other requirements: Python 3.6 or higher, fastai library v. 0.7
- License: GNU GPL v3.0
- Any restrictions to use by non-academics: None

### Predictions with CellProfiler

Analyses with *CellProfiler* were performed as described previously (Bray et al., 2015) with few changes. The step ‘Align’ was removed due to inaccurate alignment, therefore individual PlateTemplate.png files were

made for each image. Because of the high memory usage and memory leaks we reduced the image resolution from 5184×3456 pixels to 1728×1152 pixels. The range of typical diameter of objects was set to 6–33 pixel units and the threshold value which separates white from non-white colonies was optimized to 0.055.

### Acknowledgements

We thank M. Rempfler for discussions and for suggesting ideas for the automated segmentation, and K. Volkman and S. van Eeden for help with image acquisition and processing.

### Competing interests

The Friedrich Miescher Institute for Biomedical Research (FMI) receives significant financial contributions from the Novartis Research Foundation. Published research reagents from the FMI are shared with the academic community under a Material Transfer Agreement (MTA) having terms and conditions corresponding to those of the UBMTA (Uniform Biological Material Transfer Agreement).

### Author contributions

Conceptualization: L.D., S.H.C.; Methodology: L.D., S.H.C., Y.S.; Software: S.H.C.; Validation: L.D., S.H.C., Y.S.; Formal analysis: L.D., S.H.C.; Resources: M.B.; Data curation: L.D., S.H.C.; Writing - original draft: S.H.C.; Writing - review & editing: L.D., S.H.C., M.B.; Visualization: L.D., S.H.C., M.B.; Supervision: M.B.; Project administration: M.B.; Funding acquisition: M.B.

**Funding**

The Schweizerischer Nationalfonds zur Förderung der Wissenschaftlichen Forschung NCCR RNA & Disease [grant no. 141735], the European Research Council [ERC-CoG-REpiReg-681213], and the Friedrich Miescher Institute for Biomedical Research, which is supported by the Novartis Stiftung für Medizinisch-Biologische Forschung, provided funds to cover salary and infrastructure costs. Design of the study and collection, analysis, and interpretation of data and writing of the manuscript was not influenced by the funding bodies.

**Data availability**

Code and trained network weights are freely available to the community at <https://github.com/fmi-basel/buehler-colonyclassification>. Raw images of plates with yeast colonies used in Fig. 2 are available from the Zenodo data archive (<https://zenodo.org/>). DOI: 10.5281/zenodo.3779863.

**References**

- Bray, M.-A., Vokes, M. S. and Carpenter, A. E.** (2015). Using CellProfiler for automatic identification and measurement of biological objects in images. *Curr. Protoc. Mol. Biol.* **109**, 14.17.1-14.17.13. doi:10.1002/0471142727.mb1417s109
- Dalmış, M. U., Litjens, G., Holland, K., Setio, A., Mann, R., Karssemeijer, N. and Gubern-Mérida, A.** (2017). Using deep learning to segment breast and fibroglandular tissue in MRI volumes. *Med. Phys.* **44**, 533-546. doi:10.1002/mp.12079
- Deng, J., Dong, W., Socher, R., Li, L.-J., Kai Li, R., Li Fei-Fei, N.** (2009). ImageNet: a large-scale hierarchical image database. *IEEE Computer Vision and Pattern Recognition (CVPR)*, pp. 248-255. doi:10.1109/CVPR.2009.5206848
- Duempelmann, L., Mohn, F., Shimada, Y., Oberti, D., Andriollo, A., Lochs, S. and Bühler, M.** (2019). Inheritance of a phenotypically neutral epimutation evokes gene silencing in later generations. *Mol. Cell* **74**, 534-541.e4. doi:10.1016/j.molcel.2019.02.009
- He, K., Zhang, X., Ren, S. and Sun, J.** (2016). Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 770-778, doi:10.1109/CVPR.2016.90
- Howard, J. and others.** (2018). *fastai*. GitHub.
- Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S. et al.** (2016). Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 3296-3297. doi:10.1109/CVPR.2016.351
- Huang, G., Li, Y., Pleiss, G., Liu, Z., Hopcroft, J. E. and Weinberger, K. Q.** (2017). Snapshot Ensembles: Train 1, get M for free. arXiv:1704.00109v1 [cs.LG].
- Hughes, A. J., Mornin, J. D., Biswas, S. K., Beck, L. E., Bauer, D. P., Raj, A., Bianco, S. and Gartner, Z. J.** (2018). Quanti.us: a tool for rapid, flexible, crowd-based annotation of images. *Nat. Methods* **15**, 587-590. doi:10.1038/s41592-018-0069-0
- Kowalik, K. M., Shimada, Y., Flury, V., Stadler, M. B., Batki, J. and Bühler, M.** (2015). The Paf1 complex represses small-RNA-mediated epigenetic gene silencing. *Nature*. doi:10.1038/nature14337
- Kumar, V., Webb, J. M., Gregory, A., Denis, M., Meixner, D. D., Bayat, M., Whaley, D. H., Fatemi, M. and Alizad, A.** (2018). Automated and real-time segmentation of suspicious breast masses using convolutional neural network. *PLoS ONE* **13**, e0195816. doi:10.1371/journal.pone.0195816
- Lin, T.-Y., Goyal, P., Girshick, R., He, K. and Dollár, P.** (2017). IEEE International Conference on Computer Vision (ICCV), Venice, 2999-3007, doi:10.1109/ICCV.2017.324
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y. and Berg, A. C.** (2016). SSD: Single Shot MultiBox Detector. In: Leibe B., Matas J., Sebe N., Welling M. (eds) Computer Vision - ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, vol.9905. Springer, Cham. doi:10.1007/978-3-319-46448-0\_2
- Ronneberger, O., Fischer, P. and Brox, T.** (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab N., Hornegger J., Wells W., Frangi A. (eds) Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science, vol. 9351. Springer, Cham. doi:10.1007/978-3-319-24574-4\_28
- Smith, L. N.** (2015). Cyclical Learning Rates for Training Neural Networks. arXiv:1506.01186 [cs.CV].
- Sommer, C., Straehle, C., Köthe, U. and Hamprecht, F. A.** (2011). Ilastik: Interactive learning and segmentation toolkit. in 2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro 230-233. doi:10.1109/ISBI.2011.5872394
- Van Der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E. and Yu, T.** (2014). Scikit-image: image processing in Python. *PeerJ* **2**, e453. doi:10.7717/peerj.453