Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Original Paper

Bioimage informatics

A Multi-Scale Convolutional Neural Network for Phenotyping High-Content Cellular Images

William J. Godinez [1,∗], **Imtiaz Hossain** [1], **Stanley E. Lazic** [1,3], **John W. Davies** [2] **and Xian Zhang** [1,∗]

[1] Novartis Institutes for BioMedical Research Inc., Basel, Switzerland.
[2] Novartis Institutes for BioMedical Research Inc., Cambridge, Massachusetts, USA.
[3] Current address: Quantitative Biology, AstraZeneca, Cambridge, CB4 0WG, UK.

∗To whom correspondence should be addressed.

Associate Editor: Prof. Robert Murphy

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Identifying phenotypes based on high-content cellular images is challenging. Conventional image analysis pipelines for phenotype identification comprise multiple independent steps, with each step requiring method customization and adjustment of multiple parameters.

Results: Here we present an approach based on a multi-scale convolutional neural network (M-CNN) that classifies, in a single cohesive step, cellular images into phenotypes by using directly and solely the images' pixel intensity values. The only parameters in the approach are the weights of the neural network, which are automatically optimized based on training images. The approach requires no a priori knowledge or manual customization, and is applicable to single- or multi-channel images displaying single or multiple cells. We evaluated the classification performance of the approach on eight diverse benchmark datasets. The approach yielded overall a higher classification accuracy compared to state-of-the-art results, including those of other deep CNN architectures. In addition to using the network to simply obtain a yes-or-no prediction for a given phenotype, we use the probability outputs calculated by the network to quantitatively describe the phenotypes. Our study shows that these probability values correlate with chemical treatment concentrations. This finding validates further our approach and enables chemical treatment potency estimation via convolutional neural networks.

Availability: The network specifications and solver definitions are provided in Supplementary Software 1.

Contact: william_jose.godinez_navarro@novartis.com, xian-1.zhang@novartis.com

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

High-content imaging has seen increased application in the fields of systems biology and drug discovery (Götte *et al.*, 2010; Liberali *et al.*, 2014; Boutros *et al.*, 2015). The images acquired through microscopy-based assays provide ample visual information that allows investigating cellular phenotypes induced by genetic or chemical treatments. Identifying phenotypes in such cellular images is challenging because of the inherent complexity of biological processes and the intrinsic variability of cellular assays. Conventional image analysis approaches for phenotype identification typically start by extracting features at either the cellular level via object segmentation (Carpenter *et al.*, 2006; Loo *et al.*, 2007; Matula *et al.*, 2009; Fuchs *et al.*, 2010; Ljosa *et al.*, 2013) or at the image level via image content descriptors requiring no segmentation (Huang and Murphy, 2004; Chebira *et al.*, 2007; Orlov *et al.*, 2008; Coelho *et al.*, 2013; Zhou *et al.*, 2013; Uhlmann *et al.*, 2016). Relevant features are subsequently selected, normalized, and summarized, and serve as input to a classification algorithm that predicts the phenotype (**Fig. 1a**; see also reviews in (Sommer and Gerlich, 2013; Finkbeiner *et al.*, 2015)). Although successfully applied in various studies, conventional image analysis approaches have certain limitations. For instance, several steps along the analysis pipeline, such as object segmentation, dimension reduction, and phenotype classification, typically require customization to each specific assay using *a priori* knowledge, such as the geometric properties of the expected phenotypes.

Further, a number of steps also involves adjustment of multiple parameters via an empirical strategy that ignores the performance of downstream or upstream steps (Sommer and Gerlich, 2013; Finkbeiner *et al.*, 2015). While several feature extraction methods (e.g., Haralick features (Haralick, 1979)) work well without any customization or parameter adjustments, the *joint* optimization of all parameters across the entire analysis pipeline remains challenging.

Conventional image analysis approaches essentially transform the image data into different levels of abstraction, starting from the pixel intensities and ending in the higher level semantics describing the data. *Deep learning* provides a joint framework for computing and representing such hierarchical abstractions within the data. Approaches using deep learning methods, such as deep neural networks, have recently achieved superior performance in domains such as computer vision (Krizhevsky *et al.*, 2012; Sermanet *et al.*, 2014; Simonyan and Zisserman, 2015) and genomics (Alipanahi *et al.*, 2015; Zhou and Troyanskaya, 2015; Chen *et al.*, 2016). Earlier analyses of cellular images via neural networks relied on manually defined features (Boland *et al.*, 1998; Weisser *et al.*, 1998). In comparison, convolutional neural networks (CNNs) learn and extract features from images automatically (LeCun *et al.*, 2015) and have therefore seen increased application to cellular image data. For example, CNN-based approaches have been used to segment cells or other subcellular entities with superior accuracy (Ning *et al.*, 2005; Ciresan *et al.*, 2012a; Helmstaedter *et al.*, 2013; Ronneberger *et al.*, 2015). Other studies first segment image regions displaying individual cells via conventional methods before using CNNs for classification (Buyssens *et al.*, 2012; Gao *et al.*, 2016). More recently, a CNN-based approach with a multiple instance learning scheme both segments and classifies individual cells before aggregating the per-cell results to generate a phenotypic prediction for the whole image (Kraus *et al.*, 2016).

Here we present a CNN-based approach that, without segmentation and in one cohesive step, classifies cellular images into phenotypes based directly and solely on the images' pixel intensity values. Multi-scale information has been found to be beneficial for phenotyping tasks with feed-forward neural networks (e.g., (Chebira *et al.*, 2007)). We have likewise adopted a multi-scale strategy and designed a multi-scale convolutional neural network (M-CNN) architecture that is applicable to diverse cellular imaging datasets without customization and manual parameter adjustments (**Fig. 1b**). We benchmarked our approach on eight publicly available image datasets involving six different cell lines with a broad range of phenotypes and assay setups. The proposed M-CNN approach using exactly the same network architecture on all datasets achieved better or equivalent classification accuracy compared to state-of-the-art results obtained with conventional methods. We also benchmarked other deep architectures, viz. AlexNet (Krizhevsky *et al.*, 2012) and GoogleNet (Szegedy *et al.*, 2015). The comparison with other deep architectures underlines the ability of the M-CNN architecture to obtain accurate classification results across small as well as large datasets. Analogous to previous image-based compound profiling schemes (e.g., (Bakal *et al.*, 2007; Loo *et al.*, 2007)), our study departs from the standard hard classification paradigm (i.e., yes-or-no prediction for a phenotype label) typically used in deep learning phenotyping studies (Buyssens *et al.*, 2012; Gao *et al.*, 2016; Kraus *et al.*, 2016), and instead proposes a soft classification approach where the probability outputs calculated by the network are used to quantitatively describe cellular phenotypes arising from perturbations induced by compounds at different concentrations. Based on these probability values, we computed concentration-response curves and potency values that were consistent with the literature. This result provides additional validation to our approach and demonstrates the ability of convolutional neural networks to obtain chemical treatment potency estimates from high-content cellular images.

2 Methods

2.1 Convolutional neural network

We used a deep convolutional neural network (CNN) to classify microscopy images into phenotypes. A convolutional neural network typically includes two types of computational layers: *convolutional layers* as well as *fully connected layers* as in a standard feed-forward neural network (LeCun *et al.*, 1998; Ciresan *et al.*, 2012b). A convolutional layer exploits the local 2D geometric information encoded in the image by computing *convolutions* between the layer's input (e.g., the original image or the output of a previous convolutional layer) and multiple 2D convolution kernels. Each kernel encodes a geometric pattern and convolution with each kernel results in a *kernel map* (or feature map) where image positions at which the pattern emerges are emphasized. The kernel maps are pixel-wise subjected to a non-linear activation function (e.g., a rectified linear function) and typically summarized spatially through a down-sampling operation known as *pooling*. The kernel maps are passed onto subsequent convolutional layers, and so kernels at deeper layers emphasize more complex patterns. Typically, the output of the last convolutional layer is fed onto a fully connected layer, whereupon the network operates in a standard feed-forward manner to generate a prediction on a given input. In our case, the network's output layer has N_p units representing the N_p phenotypes to be identified. For a given input image $\mathbf{x}$, the network computes an *activation level* $a_j(\mathbf{x})$ for each *j*-th unit at the output layer. Based on these activation levels, we compute a vector $\boldsymbol{\rho}$ where its elements ρ_k encode a probability mass function over the N_p phenotypes to be identified:

$$\rho_k := p(y = k|\mathbf{x}) = \frac{\exp(a_k(\mathbf{x}))}{\sum_j^{N_\mathrm{p}} \exp(a_j(\mathbf{x}))} \tag{1}$$

where k is an index representing a phenotype. From these probabilities, we obtain an estimate for the most probable phenotype:

$$\hat{y} = \underset{k}{\operatorname{argmax}}\ p(y = k|\mathbf{x}) \tag{2}$$

We refer to Equation 1 as the *soft-classification* predictions while we denote Equation 2 as the *hard-classification* prediction. All network parameters, including the patterns encoded in the convolution kernels, are determined automatically from training data. Concretely, given a training dataset consisting of N_t sample images annotated with ground-truth labels, the goal during training is to improve the performance of the network by minimizing the following error function:

$$\frac{1}{N_\mathrm{t}} \sum_{i=1}^{N_\mathrm{t}} f\left(\boldsymbol{\rho}^{(i)}, y_{\mathrm{true}}^{(i)}\right) + \lambda \|\mathbf{w}\|_2 \tag{3}$$

where $f(\cdot,\cdot)$ is a function (e.g., the cross-entropy error function) evaluating the agreement between the network's output $\boldsymbol{\rho}^{(i)}$ and the ground truth label $y_{\mathrm{true}}^{(i)}$ for the *i-th* training example, $\mathbf{w}$ is a vector including all weights of the network, $\|\cdot\|_2$ is the L2 norm, and λ is a factor regulating the influence of the magnitude of the weight vector on the error function (weight decay coefficient). To obtain an approximate solution we use the stochastic gradient descent (SGD) algorithm via backpropagation.

2.2 Multi-scale architecture

The network architecture (e.g., the number of layers, the number of kernels or units within each convolutional or fully connected layer, respectively, as well as the size of each kernel and the pooling factors) determines to a large extent the predictive performance of the method (Simonyan and Zisserman, 2015). In our case, because of biological reasons (e.g., different cell lines exhibit different morphologies) and technical reasons (e.g., images

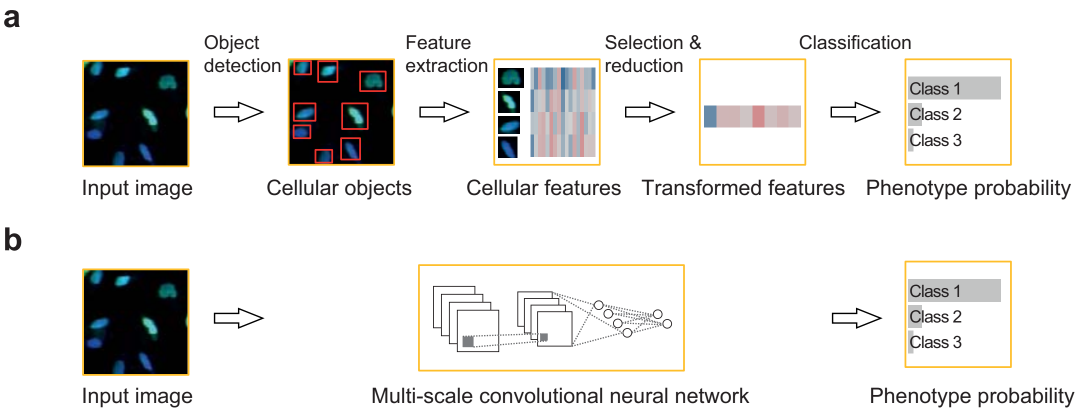


Fig. 1. Comparison between a conventional image analysis pipeline and the proposed approach based on a multi-scale convolutional neural network (M-CNN). (a) Starting from the raw image data, a conventional pipeline workflow carries out a series of independent data analysis steps that culminates with a prediction for the phenotype classes. Each step involves method customization as well as parameter adjustments. (b) The M-CNN approach instead classifies the raw image data into phenotypes in one unbiased and automatic step. The parameters in the approach correspond to the weights of the neural network and these are automatically optimized based on training images.

are acquired at different magnification factors), phenotypes are generally evident at different spatial levels in the image data. For the convolutional layers, if the size of the kernel does not cover the spatial extent of a relevant geometric pattern, the network might not learn that pattern. Typical sizes for the kernels at the first convolutional layer are relatively small (e.g., 3 × 3 pixels (Simonyan and Zisserman, 2015; Ronneberger *et al.*, 2015) and so these do not capture phenotypes observed over a large spatial region (e.g., elongated cells covering regions of about 200 × 200 pixels). In classical convolutional architectures, such as AlexNet (Krizhevsky *et al.*, 2012), the pooling layers effectively increase the spatial extent of the kernels at subsequent convolutional layers, with subsequent layers capturing patterns at increasingly coarse scales. This *sequential* multi-scale approach should in principle help the network to capture large scale patterns. However, either a large number of pooling steps or very strong pooling factors would be needed to capture phenotypes observed over a large spatial region. The number of pooling steps and pooling factors would also need adjustment based on the spatial extent of the phenotype. Further, the deeper convolutional layers do not operate on the original image data and so features at coarser scales directly from the original image data are not computed. Another strategy to cope with large-scale phenotypes is to increase the size of the kernels, possibly in parallel within a single convolutional layer (e.g., (Szegedy *et al.*, 2015)). This strategy however increases the number of the parameters and the computation time, as well as introduces severe artefacts at the borders of the kernel maps. Additionally, the kernel sizes might need to be adjusted based on the size of the relevant patterns of interest, which might be difficult to estimate accurately *a priori*. A final strategy to detect phenotypes observed over a large spatial region would be to downscale the original image prior to feeding it to the network, but this would discard the finer visual details of the phenotypes.

In order to capture cellular phenotypes at different spatial scales, we developed a multi-scale convolutional neural network (M-CNN) architecture (**Fig. 2**) that, in comparison to more classical architectures (e.g., (Krizhevsky *et al.*, 2012)), carries out a *parallel* multi-scale analysis of the image over a large number of scales. Previous parallel multi-scale approaches involve training several independent CNNs, with each network taking as input a version of the image at a different scale (Buyssens *et al.*, 2012). We, instead, developed a single network architecture that processes multiple scales of an image over parallel sequences of convolutional layers (Kamnitsas *et al.*, 2017; Farabet *et al.*, 2013). Given an original image with spatial dimensions $w \times h$, we obtain a scaled image with dimensions $(w/s \times h/s)$ via a subsampling operation. In total we consider seven values for s: 1, 2, 4, 8, 16, 32, and 64. The values are selected based on a geometric series that covers an exponential range of scales (Lindeberg, 1998). The network takes as input all seven scaled versions of the image and processes each scaled image with a sequence of three convolutional layers that constitute a *convolutional pathway*. Each convolutional pathway at each scale operates independently from the others and emphasizes patterns emerging at a particular scale. We use a fixed size (5 × 5 pixels) for the kernels in all pathways and in all layers. At the end of each pathway, the resulting kernel maps are scaled to the coarsest scale through a pooling step. To combine the information from the different scales, the pooled kernel maps from all pathways are concatenated and serve as input to a final convolutional layer that operates pixel-wise through the maps. The concatenation of the maps and subsequent convolution therefore allows the network to determine how fine and coarse features are spatially co-localized. The kernel maps from the last convolutional layer are passed to a fully connected layer with 512 units. The activation levels of these

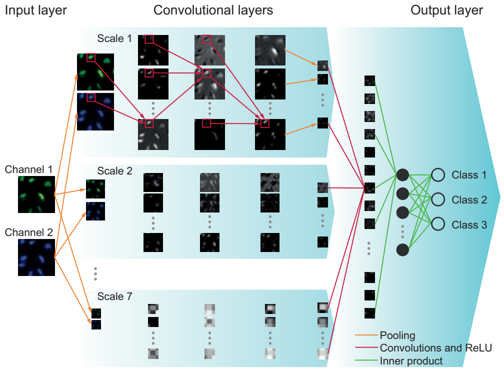


Fig. 2. Schematic overview the multi-scale convolutional neural network (M-CNN) architecture. Starting from a multi-channel input image, the approach subsamples the image into seven scales (pooling operation). Each scaled version of the image serves as input to a convolution layer, where the network emphasizes a geometric pattern, which is learned during a prior training stage (convolution operation). Each convolution operation results in a kernel map, where a specific geometric pattern is further highlighted by applying a rectification step (ReLU; rectified linear unit). Kernel maps computed across all scales are mapped to a common scale and combined through a final convolutional layer. The resulting kernel maps are passed onto a fully connected layer (inner product operation) whereupon the output layer generates a probability value for each phenotype class. For simplicity, only example scales, kernel maps, and operations are shown.

units are propagated onto the final output layer with N_p units representing the N_p phenotypes to be identified. From the output layer we obtain an estimate for the most probable phenotype in the image (see Equation 2). In total, this deep and wide architecture comprises 24 layers including 22 convolutional layers and two fully connected layers. The complete architecture is shown in **Supplementary Table 1**. Learning details and further empirical validation of our architectural choices are described in **Supplementary Note**.

2.3 Image Data

We evaluated the performance of our multi-scale approach on eight publically available cellular imaging datasets with ground-truth labels, namely five datasets from the Broad Bioimage Benchmark Collection (Ljosa *et al.*, 2012) (viz. BBBC013, BBBC014, BBBC015, BBBC016, and BBBC021), two datasets from the WND-CHARM collection (Orlov *et al.*, 2008) (viz. HeLa and CHO datasets), and the Human Protein Atlas (HPA) dataset (Barbe *et al.*, 2008; Li *et al.*, 2012). The datasets cover a wide range of biological questions, cellular systems, labeling strategies, and imaging settings (see **Supplementary Table 2** for details). In all eight datasets, we used exactly the same architecture, the same training algorithm, and the same values for the training parameters (see **Supplementary Note** for details).

2.4 Evaluation strategy

To evaluate the accuracy of the hard classification predictions generated by our approach on each dataset (bar BBBC021), we repeatedly divide randomly the images of each class into a training dataset and a test dataset. The proportion of images of each class assigned to the training dataset is 90%. We train our network on the training dataset over 18 epochs. We then apply the network to the test dataset and compute a confusion matrix. The classification accuracy is computed based on the trace of this confusion matrix. We repeat this process 50 times. The classification accuracy over all repeats is summarized using the mean, standard deviation (SD), median, and median absolute deviation (MAD). On the BBBC013 dataset we use the images from the positive and negative controls plus images from the highest and lowest concentrations to evaluate the accuracy. On the BBBC014-16 datasets we evaluate the performance using images corresponding to the two highest and two lowest concentrations of the treatments. To avoid biasing the classifier with “batch” effects (Shamir, 2011) (e.g., non-biological image artefacts such as illumination differences across batches), in the BBBC015 and BBBC016 dataset we use only one field per well during cross-validation. The images in the WND-CHRM HeLa and CHO datasets come presumably from the same batch. Nevertheless, analogous to previous publications (Chebira *et al.*, 2007; Orlov *et al.*, 2008; Coelho *et al.*, 2013; Uhlmann *et al.*, 2016), we use all images for evaluation. In the HPA dataset, we use images of proteins that localize to a single sub-cellular location. The number of images used for validation for each dataset is listed in **Suppplementary Table 2**.

On the BBBC021 dataset the aim is to evaluate the accuracy of the network for predicting the mechanism of action of individual treatments (compound-concentration pairs). The image acquisition protocol for this dataset already accounts for batch effects (Ljosa *et al.*, 2013). Here we adopt a leave-one-compound-out used in previous benchmark studies (Ljosa *et al.*, 2013; Kandaswamy *et al.*, 2016). We split the annotated subset of images covering 38 compounds into a training set and a test set. The training set includes images corresponding to 37 compounds, while the test set includes images of the compound excluded from the training dataset (‘left-out’ compound). The network is trained and tested on these datasets, respectively. Testing results in a prediction vector $\boldsymbol{\rho}$ for each image of the left-out compound (see Equation 1). We summarize the predictions over the fields of view and replicates. We compute the element-wise median of the probability vectors $\boldsymbol{\rho}$ computed for each of the replicate’s field-of-view images. We take the resulting median vectors for each replicate and compute the element-wise median to summarize the predictions over the replicates of a given concentration. These steps (partitioning the dataset, training, and testing) are repeated for each of the 38 compounds in the annotated dataset, so that each compound is excluded from the training set once. Afterwards, the predictions for each compound-concentration pair obtained during testing are compared with the ground-truth annotations and aggregated on a per-MoA basis to evaluate the accuracy.

Table 1. Classification accuracies on all evaluated datasets. Standard deviation values are in parentheses. The values are given in percentages [%]. The accuracy for the benchmark methods is reported in the literature using different performance measures. The performance measure used in each dataset to describe the accuracy of the benchmark method as well as the CNN architectures is specified below.

Dataset	**Benchmark**	**AlexNet**	**GoogleNet**	**M-CNN**
BBBC013[1]	99 (1)	50 (0)	50 (7)	100 (0)
BBBC014[1]	84 (3)	50 (4)	50 (4)	100 (14)
BBBC015[1]	99 (0.8)	50 (0)	50 (7)	100 (0)
BBBC016[1]	81 (7)	50 (0)	50 (24)	100 (0)
HeLa[2]	95 (0)	11 (0)	92 (3)	91 (3)
CHO[1]	99 (0.3)	29 (0)	91 (5)	94 (4)
HPA[3]	83	63 (5)	79 (3)	77 (3)
BBBC021[4]	97 (9)	65 (31)	86 (30)	100 (13)

[1] (Uhlmann *et al.*, 2016) Median
[2] (Chebira *et al.*, 2007) Mean
[3] (Coelho *et al.*, 2013) Mean
[4] (Ljosa *et al.*, 2013) Median

To compare the performance of our approach with other deep CNN architectures, we assess the accuracy of the AlexNet (Krizhevsky *et al.*, 2012) and GoogleNet architectures (Szegedy *et al.*, 2015) on all datasets using exactly the same evaluation strategy and learning parameters (see **Supplementary Note** for details).

3 Results and Discussion

The classification accuracy statistics for the M-CNN architecture on all eight datasets are shown in **Table 1**, which also includes the statistics reported in the literature for state-of-the-art results as well as the statistics for two other deep CNN architectures. All statistics for all evaluated architectures on all datasets are shown in **Supplementary Table 6.** Without customization or parameter adjustment for any particular dataset, the M-CNN architecture achieved better or similar classification accuracy compared to state-of-the-art results, including those of other deep CNN architectures. Below we discuss in detail the results on each dataset.

3.1 Binary phenotype classification

In each of the first four datasets (viz. BBBC013-16), only two phenotype classes are expected - namely the neutral control and the positive control phenotypes. The number of units at the output layer of the neural network was thus set to two ($N_\mathrm{p} = 2$). For each dataset, using neutral and positive control images only, we evaluated the classification performance via cross-validation (see **Section 2.4**). After training, for each image in the test dataset, the network computes a probability value for the neutral or positive phenotype, with a value of 0 indicating a low likelihood and a value of 1 indicating a high likelihood for a given phenotype. If the probability for the positive phenotype was larger than 0.5, then the image was predicted as a positive phenotype, otherwise a neutral phenotype (hard classification prediction, see Equation 2). Over the 50 validation repeats, the M-CNN architecture achieved equal or better accuracy in all datasets in comparison to the benchmark CP-CHARM method that by design classifies only the GFP channel (see **Table 1**; the M-CNN yields exactly the same statistics when classifying only the GFP channel). The neutral and positive phenotypes in the BBBC013 and BBBC015 datasets are visually distinct and thus were captured correctly by the CP-CHARM method. Our method classifies these phenotypes equally well, with 100% median accuracy over all validation runs in both cases. The phenotypes in the BBBC014 and BBBC016 datasets are more heterogeneous and thus more challenging to distinguish. The BBBC014 dataset contains images of two cell lines. The benchmark CP-CHARM method achieves a median accuracy of 84% and 81% on these two datasets, respectively. In comparison, the M-CNN architecture yields a median accuracy of 100% in both cases, with a larger standard deviation value in the BBBC014 dataset indicating variability over the cross-validation repeats.

On these four datasets, which include a small number of images (see **Supplementary Table 2** for the number of images), the optimization algorithm (viz. stochastic gradient descent, SGD) used during training exhibits poor convergence for both the AlexNet and GoogleNet architectures, and thus we observe a lower classification accuracy. In contrast, the same optimization algorithm finds highly-accurate solutions in parameter space for the M-CNN architecture.

In our multi-scale architecture (see **Supplementary Table 1**), 624 kernel maps are computed throughout the 21 convolutional layers that are distributed over the seven convolutional pathways (three layers per pathway). **Fig. 3a** presents cropped examples of the neutral and positive phenotype images and example kernel maps from the first layer of the first convolutional pathway (finest scale). Additional examples of kernel maps at coarser scales are shown in **Supplementary Fig. 1**. In the BBBC013 and BBBC014 datasets, the positive phenotype is reflected visually through co-localization of the GFP and nuclear fluorescent signals, while in the BBBC015 and BBBC016 datasets, the positive phenotype is reflected visually via vesicle-like spots (Ljosa *et al.*, 2012). The example kernel maps provide an indication that the network is able to automatically learn kernels that enhance the visual features (e.g., spots) that differentiate the controls in each dataset. It should be noted, however, that the interpretation of the learned features in a neural network, as reflected by these kernel maps, remains an open research question within the deep learning community (Zeiler and Fergus, 2014; Simonyan *et al.*, 2014; Yosinski *et al.*, 2015).

In these four datasets, cells were subjected to serial dilutions of different treatments (see **Supplementary Table 2**) to investigate the response induced at different concentrations. In each dataset, we trained the M-CNN architecture using all images from the neutral and positive controls (viz. images from the two lowest and two highest concentrations, respectively). We applied the trained network to all images of all other concentrations in the respective data set. For each image, the network computes a probability value for the neutral and positive phenotypes (soft classification predictions, see Equation 1). Since the probability values for the two phenotypes add up to one, in the following we only discuss the probability for the positive phenotype. We take the median of the probability values computed for images corresponding to replicates of a given concentration. We then plot these median values as a function of the concentration (see **Fig. 3b**). The plots show that the phenotype probability quantitatively correlates with treatment concentrations. At low concentrations, the treatments have no detectable effects and the predicted probability is close to 0. At high concentrations the probability increases to 1. At the intermediate concentrations, the variability over the experimental replicates is relatively high. As the original experiments were designed with previous knowledge of the treatment potencies, one can fit a sigmoid function describing the relation between concentration and phenotype probability to estimate EC_{50} values (**Fig. 3b**). For the disclosed treatments (viz. Wortmannin, LY294002, TNFα, and Isoproterenol), the EC_{50} values calculated from the output of the M-CNN architecture are consistent with previous studies (Carpenter *et al.*, 2006; Arnett *et al.*, 1978; Kirk *et al.*, 1982; Vlahos *et al.*, 1994).

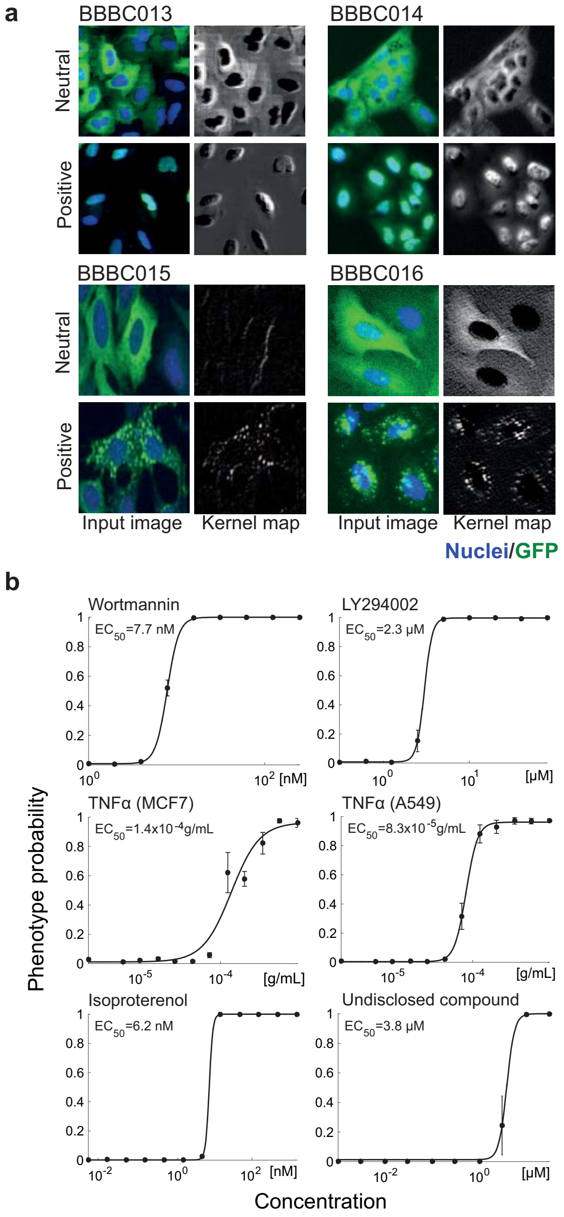


Fig. 3. Example kernel maps and concentration response curves for the BBBC013-16 datasets computed with the M-CNN architecture. (a) For each data set, example two-channel images from the neutral and positive controls are shown in their original (finest) scale. For visualization purposes, only a representative region instead of the full image is displayed (Nuclei, blue; GFP, green). Next to the images we show the representative kernel maps for a single convolution operation at the first layer and at the finest scale within the M-CNN architecture. (b) Concentration response curves for all treatments included in the four datasets computed based on the network's soft-classification predictions. The predicted probability for the positive phenotype (y-axis) is plotted against the concentration (x-axis, log scale). The dots and error bars represent the median and median absolute deviation over the experimental replicates (n = 4 for Wortmannin, LY294002, TNFα in MCF7 cells, and TNFα in A549 cells, n= 3 for Isoproterenol, and n= 2 for the undisclosed compound). A sigmoid function (black line) is fitted to the median values and used to estimate the EC_{50} values.

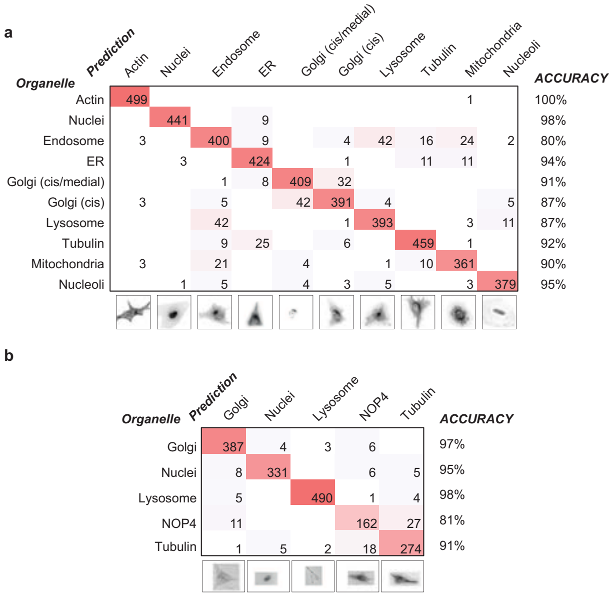
3.2 Multi-label classification of sub-cellular organelles

To evaluate the performance of our approach on datasets involving more than two classes, we applied the M-CNN architecture to three datasets where HeLa, CHO, and A-431 cells, respectively, were stained with various organelle-specific fluorescent dyes (Boland *et al.*, 1998; Boland and Murphy, 2001; Barbe *et al.*, 2008). These datasets are known to be challenging for classification, as some classes have similar patterns that can be confusing even for human experts (Boland and Murphy, 2001; Li *et al.*, 2012).

In the first dataset, HeLa cells were stained with fluorescent dyes targeting proteins in 10 organelles: Actin, Nuclei, Endosomes, ER, Golgi (cis/medial), Golgi (cis), Lysosomes, Tubulin, Mitochondria, and Nucleoli (Boland and Murphy, 2001). Accordingly, we set the number of units at the output layer of the neural network to 10 ($N_\mathrm{p} = 10$). Thus the network computes a probability value for each of the ten organelle classes, and these ten probability values add up to unity (or 100%). The class with the largest probability is considered as the predicted class (hard classification prediction). Mean classification accuracies of up to 95% have been reported for this dataset (Chebira *et al.*, 2007; Coelho *et al.*, 2013). For the AlexNet architecture, we again observe convergence issues leading to a lower mean classification accuracy (11%) over the cross-validation repeats. The GoogleNet architecture achieves a mean classification accuracy of 92%. Similarly, the M-CNN architecture yields a mean accuracy of 91%. The confusion matrix aggregating the results obtained with the M-CNN architecture over all 50 cross-validation repeats is shown in **Fig. 4a**. The diagonal of the matrix shows the number of correctly classified images for each organelle class. The non-diagonal entries show the number of incorrectly classified images per class and all zero values are omitted for simplicity. The classification accuracies for each organelle class are summarized to the right and sample images for each class are shown at the bottom of the matrix. While the classification accuracy for the whole dataset as well as for most of the organelle classes is higher than 90%, the M-CNN architecture tends to confuse images from the Endosome and Lysosome classes, as well as images from the two Golgi classes. These classification errors are consistent with results from conventional methods, and caused by the fact that the staining patterns within these classes are highly similar (Boland and Murphy, 2001).

In the CHO dataset, there are five staining classes: Golgi, Nuclei, Lysosomes, NOP4, and Tubulin (Boland *et al.*, 1998). The number of units at the output layer of the network is set to 5 ($N_\mathrm{p} = 5$). The CP-CHARM method with manually predefined features yields a median classification accuracy of 99%. Because of convergence issues, the AlexNet architecture yields a median accuracy of 29%. The GoogleNet architecture obtains a median accuracy of 91% while the M-CNN architecture yields a median accuracy of 94%. The confusion matrix for the M-CNN architecture is shown in **Fig. 4b**, with sample images shown at the bottom of the table. Here the classification accuracy for the NOP4 class is the lowest, as the network tends to confuse images from that class with images from the Golgi and Tubulin classes. Similar errors have been observed in previous studies (Boland *et al.*, 1998), which suggest overlap among these staining patterns.

In the HPA dataset, images of fluorescently labelled proteins in A-531 cells are assigned to one of thirteen subcellular location classes: Centrosome, Cytoplasm, Actin filaments, Intermediate filaments, Microtubules, Endoplasmic reticulum (ER), Golgi, Mitochondria, Nuclei, Nuclei but not nucleoli, Nucleoli, Plasma membrane, and Vesicles. The images include four channels: protein, nucleus, microtubuli, and endoplasmatic reticulum (ER). The benchmark approach yields an overall accuracy of 83%. Using all four channels as input, AlexNet obtains an accuracy of 63%, GoogleNet yields 79% while the M-CNN architecture obtains an accuracy of 77%. The confusion matrix for the M-CNN architecture is shown in **Supplementary Table 7**. Here the errors are similar to those obtained with conventional approaches (e.g., the Nuclei class is confused with the Nuclei but not nucleoli class, (Li *et al.*, 2012)). On this large dataset with multiple channels, the accuracy of the M-CNN architecture is partially curtailed by the relatively small number of filters included in the architecture. We obtain more competitive results (accuracy of 80%) with a simple three-fold increase in the number of filters that helps the architecture to cope with the larger number of channels and patterns. Further architectural changes (e.g., introduction of residual layers, (He *et al.*, 2016)) might be needed to obtain a higher accuracy on this dataset.



a

Organelle \ Prediction	Actin	Nuclei	Endosome	ER	Golgi (cis/medial)	Golgi (cis)	Lysosome	Tubulin	Mitochondria	Nucleoli	ACCURACY
Actin	499								1		100%
Nuclei		441		9							98%
Endosome	3		400	9		4	42	16	24	2	80%
ER		3		424		1		11	11		94%
Golgi (cis/medial)			1	8	409	32					91%
Golgi (cis)	3		5		42	391	4			5	87%
Lysosome			42			1	393		3	11	87%
Tubulin			9	25		6		459	1		92%
Mitochondria	3		21		4		1	10	361		90%
Nucleoli		1	5		4	3	5		3	379	95%

b

Organelle \ Prediction	Golgi	Nuclei	Lysosome	NOP4	Tubulin	ACCURACY
Golgi	387	4	3	6		97%
Nuclei	8	331		6	5	95%
Lysosome	5		490	1	4	98%
NOP4	11			162	27	81%
Tubulin	1	5	2	18	274	91%

Fig. 4. Confusion matrices for the HeLa and CHO datasets obtained via cross-validation for the M-CNN architecture. (a) HeLa. (b) CHO. For each matrix, the rows show the true organelle class while the columns show the prediction from the M-CNN architecture. The values are the aggregated results over 50 cross-validation runs. Entries are shaded in red and the shading intensity correlates with the relative magnitude of the values. Entries without numbers indicate values of zero. The per-class accuracies are summarized next to each row while example images from each class are shown below each table.

3.3 Multi-label classification and prediction of compound MoA

The BBBC021 dataset is a large-scale compound profiling effort, where MCF-7 breast cancer cells were treated with 113 compounds at eight concentrations in triplicate, and labeled for DNA, F-actin and β-tubulin. The compound collection was designed to induce a range of cellular phenotypes at different spatial scales (Caie *et al.*, 2010). A subset of the data, covering 38 compounds or 103 compound-concentration pairs, has been manually annotated with 12 mechanism-of-action (MoA) classes (Ljosa *et al.*, 2013). Previous studies have used this “ground-truth” annotation to evaluate the ability of conventional pipelines (Ljosa *et al.*, 2013) (including a pipeline with a classification step using a deep-learning method (Kandaswamy *et al.*, 2016)) for predicting compound MoA.

As the dataset contained 55 plates, we normalized the pixel intensity values of each fluorescence channel to the intensity statistics of the images of the DMSO mock treatments of the same plate (see **Supplementary Note**). The number of units at the output layer of the neural network was set to 12 (one per MoA class). Thus the network yielded 12 probability values for the 12 MoA classes (which summed up to 100%), and the class with the highest probability was selected as the predicted MoA

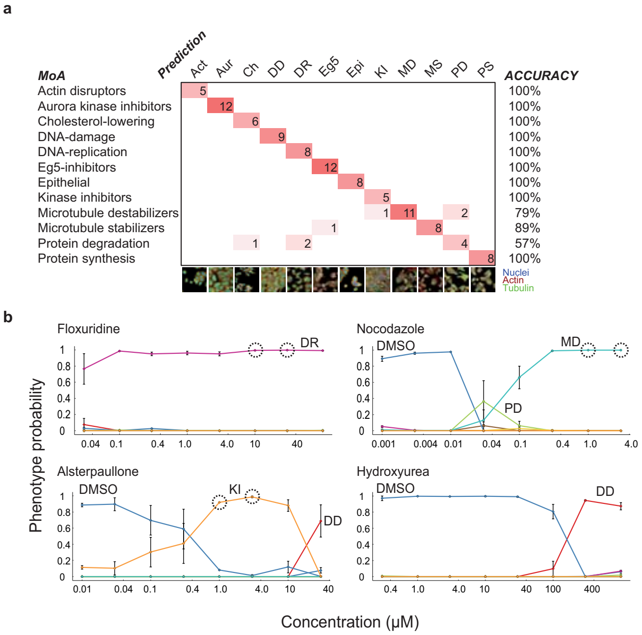


Fig. 5. Confusion matrix and example concentration response curves for the BBBC021 data. (a) Performance of the M-CNN architecture for classifying previously unseen test images into 12 MoA classes. Each row of the confusion matrix shows the true MoA annotation while the columns show the predictions from the M-CNN architecture. Entries are shaded in red and the shading intensity correlates with the relative magnitude of the values. Entries without numbers indicate values of zero. The accuracies for each MoA class over cross-validation are shown next to each row. Example three-channel images (Actin, red; Tubulin, green; Nuclei, blue) from each class are shown below the table. (b) Concentration response curves for four example compounds computed based on the network's soft-classification predictions. The predicted probability (y-axis) for 12 MoA classes plus the DMSO class (each class represented by a colored line) is plotted against the concentration (x-axis, log scale). Most classes are not visible since they are close to zero and overlap with each other. The more prominent classes are labeled with the corresponding DMSO or MoA abbreviations (cf. Fig. 5a). The dots and error bars represent the median and median absolute deviation over the experimental replicates (n = 2 for Alsterpaullone and n = 3 for the other three compounds). Images corresponding to data points marked by dashed circles are part of the training data.

(hard classification prediction). The predicted MoA classes of the held-out compound at different concentrations were compared with the original annotations to evaluate the accuracy (see **Section 2.4** and **Supplementary Table 3**).

Fig. 5a displays the computed confusion matrix based on the aggregated results over all 38 compounds. The classification accuracies for each MoA class are displayed on the right, the number of correctly classified treatments (compound-concentration pairs) is shown on the diagonal, and all zero values are omitted for simplicity. The M-CNN architecture achieved perfect classification for 9 out of 12 MoA classes. For the remaining three MoA classes (microtubule destabilizers, microtubule stabilizers and protein degradation), the accuracies were 79%, 89% and 57% respectively, with only a handful of misclassified treatments. In six of the seven misclassified treatments, the MoA with the second-best probability corresponds to the true MoA (see **Supplementary Table 3**). These results are similar to the best method evaluated in a previous benchmark study (Ljosa *et al.*, 2013), which deployed a customized and manually adjusted analysis pipeline (viz. segment cellular objects, extract pre-defined features, perform factor analysis, and apply a nearest-neighbor classifier for phenotype prediction). Further, our approach outperformed a conventional pipeline feeding feature vectors to a deep learning method (viz. deep auto-encoders) for classification (Kandaswamy *et al.*, 2016), which achieved an overall classification accuracy of 77%, as well as other deep CNN architectures, namely AlexNet and GoogleNet, which achieve accuracies of 65% and 86%, respectively. The results highlight the importance of analyzing the image data directly via a parallel multi-scale convolutional strategy as performed in the M-CNN architecture as well as the ability of the proposed architecture to generalize to compounds left out from the training process.

We then proceeded to describe the MoA class of compounds and concentrations without any annotation using the soft classification predictions of the M-CNN architecture. We first trained a single multi-scale convolutional neural network with 1684 original images (approximately 13% of the data) coming from all annotated data (without holding out any compounds) plus images from the DMSO mock treatments. The network generated predictions for the 12 annotated MoA classes and a DMSO class. The number of units at the output layer of the neural network was therefore set to 13. We then applied the trained model to the whole data collection of 113 compounds at eight concentrations and three replicates to predict their mechanism of action. The computed probability values representing the soft-classification predictions for all treatments are provided in **Supplementary Table 4**.

We selected four example compounds and plotted the phenotype probability for the 13 classes against the compound concentration (**Fig. 5b).** Multiple fields and replicates were summarized by calculating the median and the median absolute deviation. The first example, Floxuridine, was tested at eight concentrations ranging from 0.03 to 100 μM. Concentrations 10 and 30 μM were annotated as DNA replication (DR) in the training data (marked with dashed circles). The neural network predicted the other concentrations to have the same phenotype with high probabilities and low replicate variability except for the lowest concentration at 0.03 μM (**Fig. 5b**). This result is consistent with the reported EC_{50} of Floxuridine around 0.01 μM (Raić-Malić *et al.*, 2000). In the second example, Nocodazole was tested at eight concentrations between 0.001 and 3 μM (**Fig. 5b**), with concentrations 1 and 3 μM labeled as microtubule destabilizers (MD) in the training data (marked with dashed circles). At the lowest concentrations (viz. 0.001, 0.003 and 0.01 μM) DMSO was predicted as the dominant class, indicating that at these concentrations the compound did not induce any distinguishable effect. At 0.03 μM, the protein degradation (PD) phenotype was slightly prevalent. As the concentration increased, the microtubule destabilizers (MD) MoA became gradually dominant. The concentration at which we observe the half maximal response for the MD MoA also agrees with previously published EC_{50} values (Kiselyov *et al.*, 2010). In the third example, Alsterpaullone was tested at eight concentrations between 0.03 and 30 μM (**Fig. 5b**). Concentrations 1 and 3 μM were labeled as kinase inhibitor (KI) in the training data. The model was able to predict a concentration-dependent response for the kinase inhibitor MoA class over the first seven concentrations. Interestingly, at the highest concentration (30 μM) the neural network predicted DNA damage (DD) as the dominant MoA. This is likely due to the apoptosis-inducing effect of this compound at this concentration (Lahusen *et al.*, 2003; Faria *et al.*, 2015). For the final example, Hydroxyurea, none of the concentrations were included in the training dataset. Here the model predicted a concentration-dependent MoA of DNA damage (DD), with an EC_{50} between 100 and 400 μM, which is consistent with previous studies (Šimunović *et al.*, 2009; Banh and Hales, 2013).

4 Conclusion

We have developed an approach based on a multi-scale convolutional neural network (M-CNN) to analyze high-content cellular images. The approach yields, in a single and cohesive step, a phenotype prediction for an input image using solely the image's pixel intensity values. The proposed multi-scale architecture considers in parallel the input image at different spatial scales, and is able to identify complex and diverse phenotypic patterns emerging at various spatial levels requiring no *a priori* knowledge about the expected imaging phenotype. Analogous to previous image-based compound profiling schemes (Bakal *et al.*, 2007; Loo *et al.*, 2007), our approach uses the probability outputs computed by the network to quantitatively describe cellular phenotypes. Our study showed that these soft-classification predictions behave in a concentration-dependent manner, thus describing quantitatively the phenotypic variations arising from compound treatments, and consequently enabling chemical treatment potency estimation from high-content cellular images via convolutional neural networks. The resulting concentration response curves and estimated EC_{50} values further validate our approach, since these results suggest that the network is capturing biologically relevant information and generalizing well to images not used during training.

We demonstrated the capabilities of the M-CNN architecture by applying it to eight datasets that cover a range of cell lines, staining reagents, treatments as well as image acquisition settings (e.g., instrument types and magnifications). Compared with previous state-of-the-art results, including those of other deep CNN architectures, the M-CNN architecture achieved similar or better performance. Certainly, our comparison is limited to the available experimental scenarios as well as approaches, and as such, we do not claim that our approach is universally better. Already on more complex datasets, such as the HPA dataset, we observe some of the performance limitations of the approach. The comparison nevertheless shows that the M-CNN architecture performs well across diverse datasets without any customization or manual parameter adjustments.

Typical deep learning architectures, such as AlexNet and GoogleNet, require a large amount of labeled data for training. Here we have showed that, together with a data augmentation strategy, relatively few images are required to train the M-CNN architecture. Certainly, the proposed deep learning approach is more expensive, from a purely computational perspective, than conventional methods. On an NVIDIA Tesla K80 GPU with 11.5 GB of memory, training the neural network on all annotated images from the BBBC021 dataset together with the images generated by the data augmentation scheme takes approximately 1.5 days. Generating the predictions shown in **Supplementary Table 4** for all 13200 three-channel images takes ca. 25 minutes on the same hardware. Although requiring substantially more computational resources, the M-CNN approach saves time and effort that would be otherwise employed to manually adjust the parameters, such as segmentation and feature selection parameters, involved in conventional image analysis pipelines. The current computational performance is already sufficient for analyzing large-scale experiments and can be further improved via advanced hardware and distributed computing (Dean *et al.*, 2012).

The proposed approach computes a phenotypic prediction for a whole image and does not quantify explicitly single cell features such as cell area or nuclear ellipticity. As such, it does not replace more traditional single-cell analyses (Carpenter *et al.*, 2006; Matula *et al.*, 2009; Fuchs *et al.*, 2010) that provide such explicit cellular-level information with which detailed measurements of, e.g., phenotype heterogeneity over space and time (Gut *et al.*, 2015) may be obtained.

The current study applies a supervised deep learning approach, where training data with class labels are used to build a classification model that can be applied to test data for prediction and evaluation. The model is limited to the phenotype classes within the training dataset, and therefore cannot detect novel phenotypes. Unsupervised learning approaches, requiring no manual annotations, are an on-going research effort in the deep learning field (LeCun *et al.*, 2015) and would address this limitation. Consequently, we are exploring the application of unsupervised deep learning methods for cellular imaging data analysis and the potential to discover novel phenotypes.

Acknowledgements

We would like to thank Stephan Reiling, Enrico Schmidt, and Pierre Farmer for fruitful discussions. We acknowledge the datasets provided by Ilya Ravkin and Peter D. Caie via the Broad Bioimage Benchmark Collection, Robert F. Murphy via the WND-CHARM collection, as well as the HPA Project Team.

Funding

W.J.G. is supported by a postdoctoral fellowship from the Education Office of the Novartis Institutes for Biomedical Research.

References

Alipanahi, B., Delong, A., Weirauch, M. T., and Frey, B. J. (2015). Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nat. Biotechnol.*, **33**(8), 831–838.

Arnett, C. D., Wright, J., and Zenker, N. (1978). Synthesis and adrenergic activity of benzimidazole bioisosteres of norepinephrine and isoproterenol. *J. Med. Chem.*, **21**(1), 72–78.

Bakal, C., Church, G., and Perrimon, N. (2007). Quantitative Morphological Signatures Define Local Signaling Networks Regulating Cell Morphology. *Science*, **316**(5832), 1753–1756.

Banh, S. and Hales, B. F. (2013). Hydroxyurea exposure triggers tissue-specific activation of p38 mitogen-activated protein kinase signaling and the DNA damage response in organogenesis-stage mouse embryos. *Toxicol. Sci.*, **133**(2), 298–208.

Barbe, L., Lundberg, E., Oksvold, P., Stenius, A., Lewin, E., Björling, E., Asplund, A., Pontén, F., Brismar, H., Uhlén, M., and Andersson-Svahn, H. (2008). Toward a confocal subcellular atlas of the human proteome. *Mol. Cell. Proteomics*, **7**(3), 499–508.

Boland, M. V. and Murphy, R. F. (2001). A neural network classifier capable of recognizing the patterns of all major subcellular structures in fluorescence microscope images of HeLa cells. *Bioinformatics*, **17**(12), 1213–1223.

Boland, M. V., Markey, M. K., and Murphy, R. F. (1998). Automated Recognition of Patterns Characteristic of Subcellular Structures in Fluorescence Microscopy Images. *Cytometry*, **33**(3), 366–375.

Boutros, M., Heigwer, F., and Laufer, C. (2015). Microscopy-Based High-Content Screening. *Cell*, **163**(6), 1314–1325.

Buyssens, P., Elmoataz, A., and Lézoray, O. (2012). Multiscale Convolutional Neural Networks for Vision - based Classification of Cells. In *Proc. Asian Conf. Comput. Vis.*, pages 342–352.

Caie, P. D., Walls, R. E., Ingleston-Orme, A., Daya, S., Houslay, T., Eagle, R., Roberts, M. E., and Carragher, N. O. (2010). High-content phenotypic profiling of drug response signatures across distinct cancer cells. *Mol. Cancer. Ther.*, **9**(6), 1913–1926.

Carpenter, A. E., Jones, T. R., Lamprecht, M. R., Clarke, C., Kang, I. H., Friman, O., Guertin, D. a., Chang, J. H., Lindquist, R. a., Moffat, J., Golland, P., and Sabatini, D. M. (2006). CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biol.*, **7**(10), R100.

Chebira, A., Barbotin, Y., Jackson, C., Merryman, T., Srinivasa, G., Murphy, R. F., and Kovacević, J. (2007). A multiresolution approach to automated classification of protein subcellular location images. *BMC Bioinformatics*, **8**, 210.

Chen, Y., Li, Y., Narayan, R., Subramanian, A., and Xie, X. (2016). Gene expression inference with deep learning. *Bioinformatics*, **32**(12), 1832–1839.

Ciresan, D., Giusti, A., Gambardella, L., and Schmidhuber, J. (2012a). Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images. In *Proc. Advances in Neural Information Processing Systems 25*, pages 2843–2851.

Ciresan, D., Meier, U., and Schmidhuber, J. (2012b). Multi-column Deep Neural Networks for Image Classification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, number February, pages 3642–3649.

Coelho, L. P., Kangas, J. D., Naik, A. W., Osuna-Highley, E., Glory-Afshar, E., Fuhrman, M., Simha, R., Berget, P. B., Jarvik, J. W., and Murphy, R. F. (2013). Determining the subcellular location of new proteins from microscope images using local features. *Bioinformatics*, **29**(18), 2343–2349.

Dean, J., Corrado, G. S., Monga, R., Chen, K., Devin, M., Le, Q. V., Mao, M. Z., Ranzato, M. A., Senior, A., Tucker, P., Yang, K., and Ng, A. Y. (2012). Large Scale

Distributed Deep Networks. In *Proc. Advances in Neural Information Processing Systems 25*, pages 1223–1231.

Farabet, C., Couprie, C., Najman, L., and LeCun, Y. (2013). Learning Hierarchical Features for Scene Labeling. *IEEE Trans. Pattern Anal. Mach. Intell.*, **35**(8), 1915–1929.

Faria, C. C., Agnihotri, S., Mack, S. C., Golbourn, B. J., Diaz, R. J., Olsen, S., Bryant, M., Bebenek, M., Wang, X., Bertrand, K. C., Kushida, M., Head, R., Clark, I., Dirks, P., Smith, C. A., Taylor, M. D., and Rutka, J. T. (2015). Identification of alsterpaullone as a novel small molecule inhibitor to target group 3 medulloblastoma. *Oncotarget*, **6**(25), 21718–29.

Finkbeiner, S., Frumkin, M., and Kassner, P. (2015). Cell-Based Screening: Extracting Meaning from Complex Data. *Neuron*, **86**(1), 160–174.

Fuchs, F., Pau, G., Kranz, D., Sklyar, O., Budjan, C., Steinbrink, S., Horn, T., Pedal, A., Huber, W., and Boutros, M. (2010). Clustering phenotype populations by genome-wide RNAi and multiparametric imaging. *Mol. Syst. Biol.*, **6**, 370.

Gao, Z., Wang, L., Zhou, L., and Zhang, J. (2016). HEp-2 Cell Image Classification with Deep Convolutional Neural Networks. *IEEE J. Biomed. Heal. Informatics*, **PP**, –.

Götte, M., Hofmann, G., Michou-Gallani, A. I., Glickman, J. F., Wishart, W., and Gabriel, D. (2010). An imaging assay to analyze primary neurons for cellular neurotoxicity. *J. Neurosci. Methods*, **192**(1), 7–16.

Gut, G., Tadmor, M. D., Pe'er, D., Pelkmans, L., and Liberali, P. (2015). Trajectories of cell-cycle progression from fixed cell populations. *Nat. Meth.*, **12**(10), 951–954.

Haralick, R. M. (1979). Statistical and structural approaches to texture. *Proc. IEEE*, (5), 786–804.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'16)*, pages 770–778.

Helmstaedter, M., Briggman, K. L., Turaga, S. C., Jain, V., Seung, H. S., and Denk, W. (2013). Connectomic reconstruction of the inner plexiform layer in the mouse retina. *Nature*, **500**(7461), 168–74.

Huang, K. and Murphy, R. (2004). Automated classification of subcellular patterns in multicell images without segmentation into single cells. In *Proc. IEEE Int. Symp. on Biomedical Imaging: From Nano to Macro (ISBI'04)*, pages 1139–1142.

Kamnitsas, K., Ledig, C., Newcombe, V. F., Simpson, J. P., Kane, A. D., Menon, D. K., Rueckert, D., and Glocker, B. (2017). Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. *Medical Image Analysis*, **36**, 61 – 78.

Kandaswamy, C., Silva, L. M., Alexandre, L. A., and Santos, J. M. (2016). High-Content Analysis of Breast Cancer Using Single-Cell Deep Transfer Learning. *J. Biomol. Screen.*, **21**(3), 252–259.

Kirk, K. L., Cantacuzène, D., Collins, B., Chen, G. T., Nimit, Y., and Creveling, C. R. (1982). Syntheses and adrenergic agonist properties of ring-fluorinated isoproterenols. *J. Med. Chem.*, **25**(6), 680–684.

Kiselyov, A. S., Semenova, M. N., Chernyshova, N. B., Leitao, A., Samet, A. V., Kislyi, K. A., Raihstat, M. M., Oprea, T., Lemcke, H., Lantow, M., Weiss, D. G., Ikizalp, N. N., Kuznetsov, S. A., and Semenov, V. V. (2010). Novel derivatives of 1,3,4-oxadiazoles are potent mitostatic agents featuring strong microtubule depolymerizing activity in the sea urchin embryo and cell culture assays. *Eur. J. Med. Chem.*, **45**(5), 1683–1697.

Kraus, O. Z., Ba, J. L., and Frey, B. J. (2016). Classifying and segmenting microscopy images with deep multiple instance learning. *Bioinformatics*, **32**(12), i52–i59.

Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In *Proc. Advances in Neural Information Processing Systems 25*, pages 1097–1105.

Lahusen, T., De Siervi, A., Kunick, C., and Senderowicz, A. M. (2003). Alsterpaullone, a novel cyclin-dependent kinase inhibitor, induces apoptosis by activation of caspase-9 due to perturbation in mitochondrial membrane potential. *Mol. Carcinog.*, **36**(4), 183–194.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, **86**(11), 2278–2324.

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

Li, J., Newberg, J. Y., Uhlén, M., Lundberg, E., and Murphy, R. F. (2012). Automated Analysis and Reannotation of Subcellular Locations in Confocal Images from the Human Protein Atlas. *PLoS ONE*, **7**(11), e50514.

Liberali, P., Snijder, B., and Pelkmans, L. (2014). Single-cell and multivariate approaches in genetic perturbation screens. *Nat. Rev. Genet.*, **16**(1), 18–32.

Lindeberg, T. (1998). Feature Detection with Automatic Scale Selection. *Int. J. Comput. Vis.*, **30**(2), 79 – 116.

Ljosa, V., Sokolnicki, K. L., and Carpenter, A. E. (2012). Annotated high-throughput microscopy image sets for validation. *Nat. Methods*, **9**(7), 637–637.

Ljosa, V., Caie, P. D., ter Horst, R., Sokolnicki, K. L., Jenkins, E. L., Daya, S., Roberts, M. E., Jones, T. R., Singh, S., Genovesio, a., Clemons, P. a., Carragher, N. O., and Carpenter, a. E. (2013). Comparison of Methods for Image-Based Profiling of Cellular Morphological Responses to Small-Molecule Treatment. *J. Biomol. Screen.*, **18**(10), 1321–1329.

Loo, L.-H., Wu, L. F., and Altschuler, S. J. (2007). Image-based multivariate profiling of drug responses from single cells. *Nat. Meth.*, **4**(5), 445–453.

Matula, P., Kumar, A., Wörz, I., Erfle, H., Bartenschlager, R., Eils, R., and Rohr, K. (2009). Single-cell-based image analysis of high-throughput cell array screens for quantification of viral infection. *Cytometry Part A*, **75A**(4), 309–318.

Ning, F., Delhomme, D., LeCun, Y., Piano, F., Bottou, L., and Barbano, P. E. (2005). Toward automatic phenotyping of developing embryos from videos. *IEEE Transs. Image Process.*, **14**(9), 1360–1371.

Orlov, N., Shamir, L., Macura, T., Johnston, J., Eckley, D. M., and Goldberg, I. G. (2008). WND-CHARM: Multi-purpose image classification using compound image transforms. *Pattern Recognit. Lett.*, **29**(11), 1684–1693.

Raić-Malić, S., Svedruzić, D., Gazivoda, T., Marunović, a., Hergold-Brundić, a., Nagl, a., Balzarini, J., De Clercq, E., and Mintas, M. (2000). Synthesis and antitumor activities of novel pyrimidine derivatives of 2,3-O,O-dibenzyl-6-deoxy-L-ascorbic acid and 4,5-didehydro-5,6- dideoxy-L-ascorbic acid. *J. Med. Chem.*, **43**(25), 4806–4811.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, volume 9350, pages 238–245.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y. (2014). OverFeat : Integrated Recognition , Localization and Detection using Convolutional Networks. In *Proc. International Conference on Learning Representations*.

Shamir, L. (2011). Assessing the efficacy of low-level image content descriptors for computer-based fluorescence microscopy image analysis. *J. Microsc.*, **243**(3), 284–292.

Simonyan, K. and Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. In *Proc. International Conference on Learning Representations*.

Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. In *Proc. International Conferences on Learning Representations Workshop*.

Šimunović, M., Perković, I., Zorc, B., Ester, K., Kralj, M., Hadjipavlou-Litina, D., and Pontiki, E. (2009). Urea and carbamate derivatives of primaquine: Synthesis, cytostatic and antioxidant activities. *Bioorganic Med. Chem.*, **17**(15), 5605–5613.

Sommer, C. and Gerlich, D. W. (2013). Machine learning in cell biology - teaching computers to recognize phenotypes. *J. Cell Sci.*, **126**(Pt 24), 5529–39.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going Deeper with Convolutions. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9.

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

Vlahos, C. J., Matter, W. F., Hui, K. Y., and Brown, R. F. (1994). A specific inhibitor of phosphatidylinositol 3-kinase, 2-(4-morpholinyl)-8-phenyl-4H-1-benzopyran-4-one (LY294002). *J. Biol. Chem.*, **269**(7), 5241–5248.

Weisser, M., Tiegs, G., Wendel, A., and Uhlig, S. (1998). Quantification of apoptotic and lytic cell death by video microscopy in combination with artificial neural networks. *Cytometry*, **31**(1), 20–28.

Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., and Lipson, H. (2015). Understanding Neural Networks Through Deep Visualization. In *Proc. ICML Deep Learning Workshop*.

Zeiler, M. D. and Fergus, R. (2014). Visualizing and Understanding Convolutional Networks. In *Proc. European Conference on Computer Vision*, pages 818–833.

Zhou, J. and Troyanskaya, O. G. (2015). Predicting effects of noncoding variants with deep learning.based sequence model. *Nat. Methods*, **12**(10), 931–934.

Zhou, J., Lamichhane, S., Sterne, G., Ye, B., and Peng, H. (2013). BIOCAT: a pattern recognition platform for customizable biological image classification and annotation. *BMC Bioinformatics*, **14**(1), 291.