Transfer learning with deep convolutional neural network for classifying cellular morphological changes

Alexander Kensert[1], Philip J Harrison[1] and Ola Spjuth[1]

[1]Department of Pharmaceutical Biosciences, Uppsala University, Uppsala, Sweden

June 2018

Abstract

Quantification and identification of cellular phenotypes from high content microscopy images have proven to be very useful for understanding biological activity in response to different drug treatments. The common approach has been to use classical image analysis to quantify changes in cell morphology, which requires several non-trivial and independent analysis steps. Recently convolutional neural networks have emerged as a compelling alternative, offering good predictive performance and the possibility to replace traditional workflows with a single network architecture. In this study we applied the pre-trained deep convolutional neural networks ResNet50, InceptionV3 and InceptionV2 to predict cell mechanisms of action in response to chemical perturbations for two cell profiling datasets from the Broad Bioimage Benchmark Collection. These networks were pre-trained on ImageNet which lead to much quicker model training. We obtain higher predictive accuracy than previously reported, between 94 and 97%. The ability to quickly and accurately distinguish between different cell morphologies from a scarce amount of labelled data illustrates the combined benefit of transfer learning and deep convolutional neural networks for interrogating cell-based images.

1 Introduction

High-content screening (HCS) has proven to be a useful and successful technique to identify and quantify cell phenotypes [1, 2]. Although conventional approaches for classification of phenotypes using cell-images have shown positive results [3, 4, 5, 6, 7], they require several non-trivial data analysis steps. An example is Ljosa et al. [7] and their pipeline workflows which include cellular segmentation, feature extraction, profiling methods (e.g. factor analysis) and a nearest neighbour classifier. Cell segmentation algorithms typically require

manual adjustments for each new experimental setup [4] and feature extraction tends to rely on "hand-crafted" features, such as those related to texture and shape (several of which are computationally expensive to measure). PCA, assuming a linear mapping, is then often used to reduce the dimensionality of these high dimensional (> 500) and highly correlated feature sets [8].

Convolutional neural networks (CNNs) have recently brought about breakthroughs in computer vision and image processing — CNNs automatically discover the features needed for classification of images based solely on the raw pixel intensity data [9]. This supervised feature learning technique has shown to be superior to using traditional hand-crafted features [10, 11], and the combination of segmentation and classification in a single framework [12] means that image classification can be performed without the need for prior cell segmentation. A recent survey shows a rapid growth in the application of deep learning to medical image analysis [13], with several studies outperforming medical expert classification. A convenient property of CNNs is that the pipeline workflow of the traditional methods are taken care of by the network itself; and by convolving layers with filters (or feature maps), local connectivity and parameter sharing keeps the number of parameters relatively low, even for a deeper network.

A major bottleneck when applying supervised CNNs to cell images is the scarcity of labelled data. Importantly, studies have shown that reusing models trained on different tasks reduced these problems [14, 15]. Jason et al. [16] notes that transferability of features depends on the distance between the base task and the target task. However, the features from distant tasks may still perform better than random features. The study also illustrated that initializing the network with pre-trained features improved the generalization even after considerable fine-tuning to the target dataset. Further, Zhang et al. [17] showed that features trained on natural images (ImageNet [18]) could be transferred to biological data. Neslihan et al. [19] used pre-trained models on natural images and facial images for cell nucleus classification where the performance of transfer learning and learning from scratch were compared. Their results showed that all their pre-trained models not only improved predictive performance, but also required less training time. Phan et al. [20] also successfully utilized transfer learning on bioimages outperforming all other methods on the mitosis detection dataset of the ICPR2012 contest. As the initial layers of CNNs capture low-level features, like edges and blobs — features usually shared between different types of images — transfer learning can be successfully applied to a different task.

The bbbc (broad bioimage benchmark collection) is an important publicly available collection of microscopy images intended for validating image-analysis algorithms [21]. Various algorithms have been tested and validated on these datasets — ranging from traditional pipeline workflows to deep learning techniques [5, 7, 22, 23, 24]. Pawlowski et al. [23] utilized transfer learning without fine-tuning to extract features, and Ando et al. [22] used a pre-trained model on consumer images and further transformation techniques to attain the top accuracy on this benchmark dataset of 96%. However, research on transfer learning and fine tuning of CNNs on these bbbc datasets is scarce — it is therefore important to investigate this technique and to compare with the various high

performing analysis tools that have already been applied to the bbbc datasets.

In this study we present state-of-the-art deep convolutional neural networks pre-trained on natural images, with minor image pre-processing and without segmentation. These models are used to predict mechanisms of action (MoA) and nucleus translocation, based only on pixel intensities which automatically pass through the network to give final predictions. We used two different bbbc datasets: bbbc021v1 and bbbc014v1 [21], to evaluate the models' predictive performance as well as to visualize the feature maps throughout the network. This visualization was done to understand the different levels of abstraction processed and also to understand the transferability of the networks. After the parameter values were transferred we fine-tuned our network to fit the data, the transferred parameters could thus be thought of as good initial parameter values. Although no comparison with randomized initialization of parameter values was done, we hypothesized that the pre-trained parameters would improve performance both in terms of accuracy and learning time.

2 Method

2.1 Data

bbbc021v1 The first dataset, bbbc021v1, used in this study contains MCF-7 breast cancer images available from the Broad Bioimage Benchmark Collection [21] — wells were fixed, labeled for DNA, F-actin, B-tubulin, and imaged by fluorescence microscopy as described by Caie et al [25]. We used 38 out of 113 compounds (1-7 concentrations each) which were annotated with one MoA, resulting in a total of 103 treatments (compound-concentration pairs) and 12 different MoAs (for more information see `https://data.broadinstitute.org/bbbc/BBBC021/`).

bbbc014v1 The second dataset used in this study was the bbbc014v1 provided by Ilya Ravkin, and is also available from the Broad Bioimage Benchmark Collection [21]. The images are Human U2OS cells of cytoplasm to nucleus translocation of the MCF-7 and A549 (human alveolar basal epithelial) in response to TNFα concentrations. For each well there was one field with a nuclear counterstain (DAPI) and one field with a signal stain (FITC). A total of 96-wells with 12 concentration points and 4 replicate rows for each cell type (for more information see https://data.broadinstitute.org/bbbc/BBBC014/). In this study, the four highest (labelled positive) and four lowest concentrations (including 0 concentration; labelled negative) were used for each cell type.

Image preprocessing For the bbbc021v1 dataset, the three different channels (labelled for DNA, F-actin, B-tubulin) of 16-bit range were stacked into a three channelled image. The images were then normalized plate-wise, by subtracting the mean pixel intensities of DMSO images (the control samples) and then dividing by the standard deviation of their pixel intensities. After the

normalization an Anscombe transformation [26] was performed followed by a mapping to an 8-bit range.

Similarly, for the bbbc014v1 dataset, the different channels (in this case 2) of each sample were stacked with an addition of a zero matrix to create a three-channel input. These images were in 8-bit range and were variance stabilized by the Anscombe transformation, and then mapped back to an 8-bit range.

The resulting images for both datasets were cropped into 16 images (bbbc014v1) and 4 images (bbbc021v1) to increase the number of training samples.

2.2 CNN architectures

Three different state-of-the-art architectures were implemented in Keras [27]: Resnet50 [28], Inceptionv3 [29] and InceptionResnetv2 [30]. They were all pre-trained on the ImageNet dataset, containing 13 million natural images [18].

Residual Network Utilizing a very deep CNN can have a negative effect on model performance — arising from the difficulty in finding suitable parameters for the deeper layers. Adding further layers to a suitably deep model can lead to higher training error not caused by overfitting [28, 31, 32]. Residual networks use residual mapping $H(x) = F(x) + x$, where x is the original feature vector (identity mapping) added to the deeper version of the network $F(x)$ (output of the stacked layers). Importantly, if the mappings were optimal, it would be easier for the network to push the residuals to zero than fitting an identity mapping with stacks of nonlinear layers [28]. The implication of this is that although F(x) is not learning anything, the output will simply be an identity mapping x. Thus, the worst-case scenario is that output equals input, and the best-case scenario is that some important features are learned. Residual mappings therefore assist in avoiding the degradation problem that occurs for very deep CNNs. Another important aspect of residual networks is the intermediate normalization layers (also called batch normalization), which help to solve the problem of vanishing and exploding gradients.

The Residual network used in this study had 50 layers (49 convolutional layers and a final fully connected classification layer), based on ResNet50 from the paper "Deep Residual Learning for Image Recognition" [28].

Inception network It is often difficult to determine the best filter sizes for your network, and whether to use pooling layers. To overcome this inception architectures use many different filter sizes and pooling layers in parallel (an inception block), the outputs of which are concatenated and inputted to the next block. In this way the network chooses which filter sizes or combinations thereof to use. To solve the problem of a large increase in computational cost, the inception networks utilize 1x1 convolutions to shrink the volume of the next layer. This network architecture was introduced by Szegedy et al. [33] to make a network deeper and wider, hence more powerful, but keeping the computational cost low. The Inception network could thus go very deep and,

like Resnet, utilizes intermediate normalization layers to avoid vanishing and exploding gradients.

The inception network used in this study was the InceptionV3 from the paper "Rethinking the Inception Architecture for Computer Vision" [29], excluding the auxiliary classifiers. This network had 95 layers in total, a number much larger than ResNet50 due to the width of each inception block.

Inception-Resnet network Szegedy et al. [30] evaluated a network combining inception blocks and residuals (similar to the ResNet50 residuals). They showed an improvement in training speed after introducing these residuals, making it possible to implement even deeper networks at a reasonable cost.

In this study, we implemented an inception-resnet architecture based on the InceptionResnetV2 from the paper "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning" [30]. This network is even deeper and bigger than ResNet50 and InceptionV3 respectively — totalling 245 layers.

Downsampling and data augmentation Before the images were inputted into the network, they were downsampled to the same dimensions as the images used for the pretrained network: 224x224x3 for ResNet50 and 299x299x3 for InceptionV3 and InceptionResnetV2. To increase the number of training examples the input images were randomly rotated and mirrored, as well as randomly blurred and shaded — the latter preventing the network from identifying noise as important features.

2.3 Model evaluation and deep visualization

Cross-validation To evaluate the models of the bbbc021v1 dataset we used a "leave-one-compound-out" cross-validation — resulting in a 38-fold cross-validation. In each fold predictions were made for all the treatments of the excluded compound. An element-wise median over the replicates was first calculated to obtain a prediction vector for each well. These vectors were then used to calculate the element-wise median over the wells, to obtain prediction vectors for each treatment. Finally, the highest values in the resulting 12-dimensional prediction vectors, containing the MoA predictions for the treatments, decided the models' final predictions for the treatments. This procedure was repeated for all cross-validation folds, resulting in a total of 103 final predictions.

For the bbbc014v1 dataset we used a 2-fold cross-validation where one cell-line was "left-out" as the test set, while the other was used for training. At test time, a prediction was made for each image (well) — resulting in 32 predictions for each fold.

Activation maximization To compare the pre-trained models and the fine-tuned models (fit to our MoA data), we contrast a selection of their feature maps. We used the high-level Keras visualization toolkit *keras-vis* [34] to do this and applied an activation maximization function to generate an input image

that maximizes certain filter output activations with a gradient descent method. This allows us to understand the input patterns that activate certain filters. At deeper levels within the network the filters learn more abstract representations and as a consequence show more complexly structured visualizations.

3 Results and Discussion

We illustrate the accuracy of the deep CNN models on the bbbc021v1 dataset using confusion matrices. ResNet50 attained a 97.1% accuracy (Figure 1), InceptionV3 and InceptionResnetV2 both attained an accuracy of 94.2% (Figure 2 and Figure 3). In terms of accuracy these models compare well with previous state-of-the-art algorithms, where our ResNet50 application reached greater accuracy than any model yet reported based on the Broad Bioimage Benchmark Collection [21].

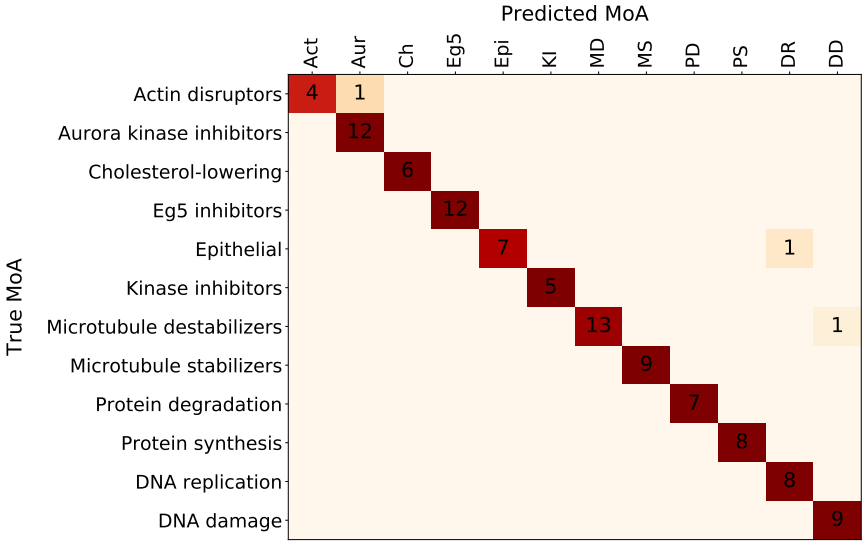

Figure 1: **ResNet50**. Confusion matrix for predictions of compound-concentration pairs, with a mean accuracy of 97.1%. Zeros are excluded for better visualization.

Furthermore, ResNet50 attained an accuracy of 100% on the bbbc014v1 dataset after just a single epoch of training. The quick learning is arguably a strong indication of transferability of the pretrained parameters.

To further analyze the transferability of the pretrained parameters, we illustrate the resulting images from the activation maximization (Figure 4). Notably, the early feature maps/filters of the two models showed similar patterns of activation, whereas the deeper filters, activated by higher level abstractions, were more dissimilar.

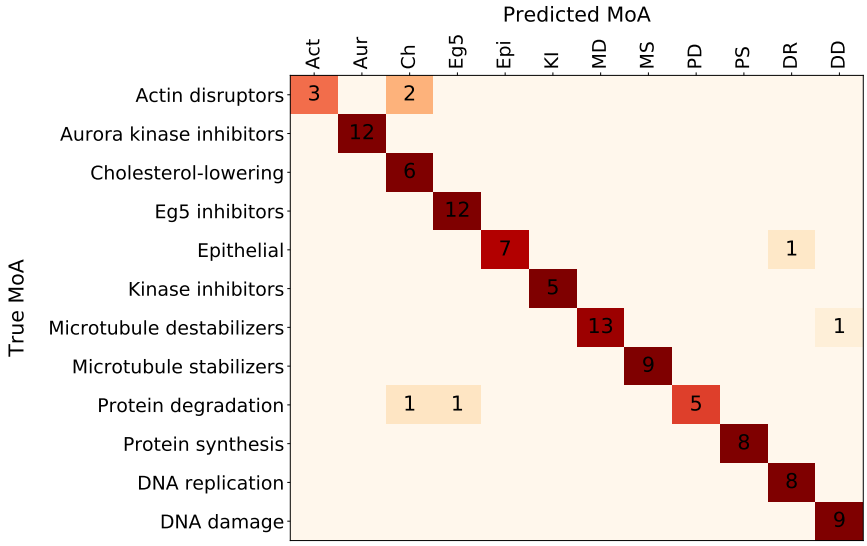

Figure 2: **InceptionV3**. Confusion matrix for predictions of compound-concentration pairs, with a mean accuracy of 94.2%. Zeros are excluded for better visualization.

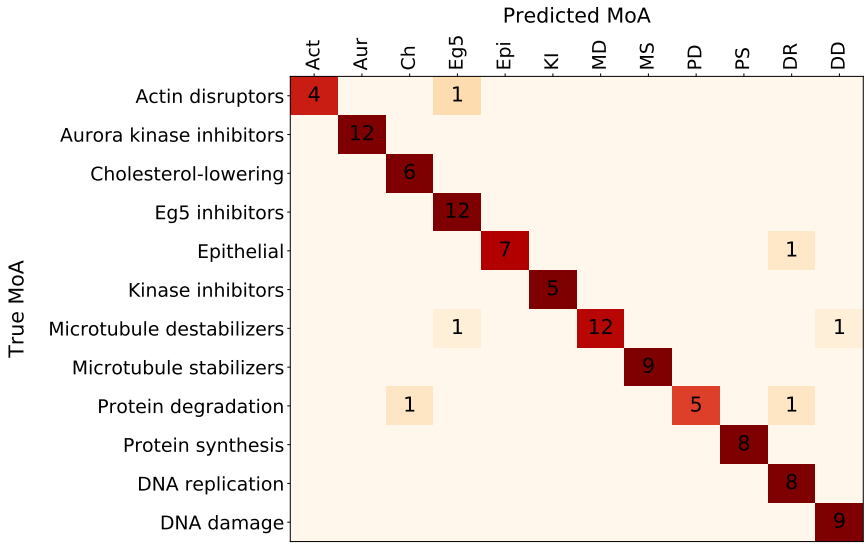

Figure 3: **InceptionResnetV2**. Confusion matrix for predictions of compound-concentration pairs, with a mean accuracy of 94.2%. Zeros are excluded for better visualization.

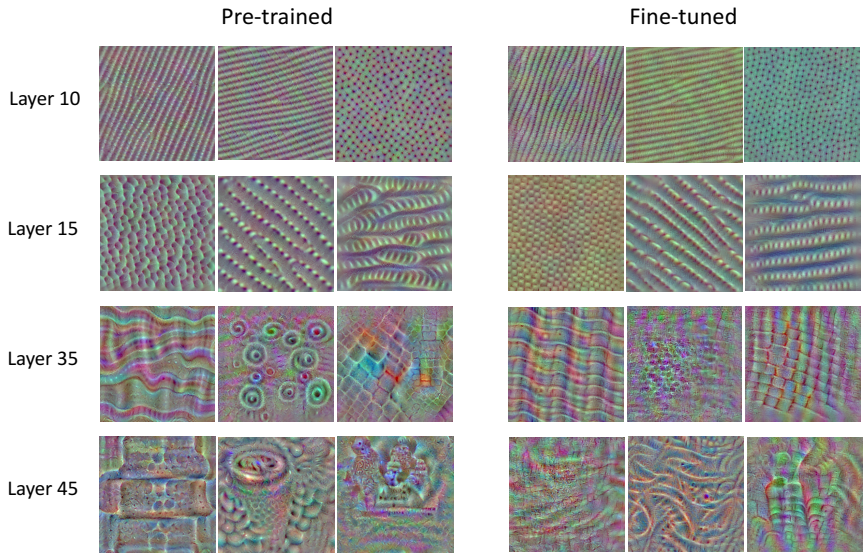

Figure 4: **Activation maximization**. Images that maximize certain filter output activations in the different layers of ResNet50 using keras-vis toolkit. A comparison between the pre-trained model and the fine-tuned model.

Limitations Although studies have been done on the topic, and techniques have been developed, the interpretation of deep neural networks are still challenging. It is not always clear what the machine has learned and how it learns, however it evidently performs well on computer vision tasks.

Finally, although the bbbc021v1 dataset is one of the very few good benchmarking datasets publicly available, it no longer presents significant challenges for many of the current state-of-the-art models, many of which have already reported accuracies of 90% and above. It would therefore be interesting to evaluate these models on more difficult classification tasks of MoAs, and evaluate them further in the field of high content imaging.

4 Conclusions

Transfer learning and deep CNNs, when used in combination, produce highly accurate classification of MoAs. These models were able to quickly distinguish the different cell phenotypes despite a limited quantity of labelled data.

Additional Files

Additional file 1 — GitHub Repository

Code used for this study can be found at `https://github.com/pharmbio/kensert_CNN`.

References

[1] Carpenter A, R Jones T, Lamprecht M, Clarke C, Han Kang I, Friman O, et al. CellProfiler: Image analysis software for identifying and quantifying cell phenotypes. 2006 02;7:R100.

[2] Liberali P, Snijder B, Pelkmans L. Single-cell and multivariate approaches in genetic perturbation screens. Nature Reviews Genetics. 2015;16:18–32.

[3] Sommer C, Gerlich DW. Machine learning in cell biology – teaching computers to recognize phenotypes. Journal of Cell Science. 2013;126(24):5529–5539. Available from: `http://jcs.biologists.org/content/126/24/5529`.

[4] C Caicedo J, Cooper S, Heigwer F, Warchal S, Qiu P, Molnar C, et al. Data-analysis strategies for image-based cell profiling. 2017 09;14:849–863.

[5] Singh S, Bray MA, Jones TR, Carpenter AE. Pipeline for illumination correction of images for high-throughput microscopy. Journal of Microscopy;256(3):231–236. Available from: `https://onlinelibrary.wiley.com/doi/abs/10.1111/jmi.12178`.

[6] Uhlmann V, Singh S, Carpenter AE. CP-CHARM: segmentation-free image classification made accessible. In: BMC Bioinformatics; 2016. .

[7] Ljosa V, Caie PD, ter Horst R, Sokolnicki KL, Jenkins EL, Daya S, et al. Comparison of Methods for Image-Based Profiling of Cellular Morphological Responses to Small-Molecule Treatment. Journal of Biomolecular Screening. 2013;18(10):1321–1329. PMID: 24045582.

[8] Kraus OZ, Frey BJ. Computer vision for high content screening. Critical Reviews in Biochemistry and Molecular Biology. 2016;51(2):102–109. PMID: 26806341. Available from: `https://doi.org/10.3109/10409238.2015.1135868`.

[9] LeCun Y, Bengio Y, Hinton G. Deep Learning. 2015 05;521:436–44.

[10] Xu Y, Mo T, Feng Q, Zhong P, Lai M, Chang E. Deep learning of feature representation with multiple instance learning for medical image analysis; 2014.

[11] Pärnamaa T, Parts L. Accurate Classification of Protein Subcellular Localization from High Throughput Microscopy Images Using Deep Learning. 2017 04;7.

[12] Z Kraus O, Jimmy Ba L, Frey B. Classifying and Segmenting Microscopy Images Using Convolutional Multiple Instance Learning. 2015 11;.

[13] Litjens G, Kooi T, Bejnordi BE, Setio AAA, Ciompi F, Ghafoorian M, et al. A survey on deep learning in medical image analysis. Medical Image Analysis. 2017;42:60 – 88. Available from: http://www.sciencedirect.com/science/article/pii/S1361841517301135.

[14] Razavian A, Azizpour H, Sullivan J, Carlsson S. CNN Features Off-the-Shelf: An Astounding Baseline for Recognition; 2014.

[15] Donahue J, Jia Y, Vinyals O, Hoffman J, Zhang N, Tzeng E, et al. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. 2013 10;32.

[16] Yosinski J, Clune J, Bengio Y, Lipson H. How transferable are features in deep neural networks?; 2014.

[17] Zhang W, Li R, Zeng T, Sun Q, Kumar S, Ye J, et al. Deep Model Based Transfer and Multi-Task Learning for Biological Image Analysis. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '15. New York, NY, USA: ACM; 2015. p. 1475–1484. Available from: http://doi.acm.org/10.1145/2783258.2783304.

[18] Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, et al. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision. 2015 Dec;115(3):211–252. Available from: https://doi.org/10.1007/s11263-015-0816-y.

[19] Bayramoglu N, Heikkilä J. Transfer Learning for Cell Nuclei Classification in Histopathology Images. In: Hua G, Jégou H, editors. Computer Vision – ECCV 2016 Workshops. Cham: Springer International Publishing; 2016. p. 532–539.

[20] Tran Hong Phan H, Kumar A, Kim J, Feng DDF. Transfer learning of a convolutional neural network for HEp-2 cell image classification; 2016.

[21] Vebjorn L, L SK, E CA. Annotated high-throughput microscopy image sets for validation. Nature Methods. 2012;9(10):637 EP.

[22] Ando DM, McLean C, Berndl M. Improving Phenotypic Measurements in High-Content Imaging Screens. bioRxiv. 2017;Available from: https://doi.org/10.1101/161422.

[23] Pawlowski N, Caicedo JC, Singh S, Carpenter AE, Storkey A. Automating Morphological Profiling with Generic Deep Convolutional Networks. bioRxiv. 2016;Available from: https://www.biorxiv.org/content/early/2016/11/02/085118.

[24] Godinez W, Hossain I, E Lazic S, W Davies J, Zhang X. A Multi-Scale Convolutional Neural Network for Phenotyping High-Content Cellular Images. 2017 02;33.

[25] Caie PD, Walls RE, Ingleston-Orme A, Daya S, Houslay T, Eagle R, et al. High-Content Phenotypic Profiling of Drug Response Signatures across Distinct Cancer Cells. Molecular Cancer Therapeutics. 2010;9(6):1913–1926. Available from: http://mct.aacrjournals.org/content/9/6/1913.

[26] Anscombe FJ. Biometrika. 1948 12;35(4-5):246–254.

[27] Chollet F, et al.. Keras; 2015. https://keras.io.

[28] He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. 2015 12;7.

[29] Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the Inception Architecture for Computer Vision; 2016.

[30] Szegedy C, Ioffe S, Vanhoucke V. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. 2016 02;.

[31] He K, Sun J. Convolutional neural networks at constrained time cost; 2015.

[32] Kumar Srivastava R, Greff K, Schmidhuber J. Highway Networks. 2015 05;.

[33] Christian S, Liu W, Jia YQ. Going deeper with convolutions. 2015 01;p. 1–9.

[34] Kotikalapudi R, contributors. keras-vis. GitHub; 2017. https://github.com/raghakot/keras-vis.