Cell Cycle Classification using Imaging Flow Cytometry and Deep Learning

Camilo Espinosa Bernal
PhD Program in Immunology
MS in Computer Science
Stanford University
camiloe@stanford.edu

Abstract

Imaging flow cytometry (IFC) enables the characterization of morphology and protein expression of thousands of cells at a single-cell resolution. However, fluorescent staining protocols can alter cell state and lead to visual artifacts. Thus, staining-free methods to quantify cell characteristics using only bright-field (BF) images are needed. Previous work has focused on extracting morphological features from BF images in a deterministic manner, with moderate success. Indeed, a key flaw of this approach is the bias introduced by the pre-defined features extracted from each image. Hence, the automated and agnostic feature extraction provided by deep learning architectures has the potential to improve on these results and provide superior performance.

In this work, I leveraged a dataset of 32,266 Jurkat cells in a proof-of-concept study to examine whether deep learning can classify cell cycle stage using only BF images. I separately trained a fully-connected (FC) 2-layer neural network, a 3-layer convolutional neural network (CNN), and a simple vision transformer (Simple ViT) and assessed their performance in a held-out test set. While the 2-layer FC architecture achieved the best balanced accuracy, the 3-layer CNN had the most interpretable embeddings. Thus, this work provides the first characterization of using deep learning for BF image analysis acquired using IFC.

1. Introduction

Imaging flow cytometry (IFC) enables the characterization of morphology and protein expression of thousands of cells at a single-cell resolution. Traditional IFC protocols involve staining cells with fluorescent antibodies against protein markers of interest followed by imaging of single cells with an IFC machine. These images contain a wealth of data, including bright-field (BF) and dark-field (DF) useful for determining cell morphology and specific channels for the quantification of desired protein markers.

There is a growing body of evidence which shows that disease initiation and progression of a broad range of pathologies is often reflected in changes of the immune state of patients. These immune cells, since they circulate in the blood, can easily be isolated, stained, and analyzed with IFC. Thus, IFC is a promising modality to capture these changes in morphology and protein expression representative of pathological mechanisms. However, it has been shown that staining protocols can alter cell state and morphology in a way which can obscure meaningful biological phenomena. Thus, there is a growing interest in computational tools which can quantify relevant cellular properties in unstained cells, i.e. using only BF and DF images. While machine learning methods to analyze IFC data have been developed, sophisticated tools to fully exploit this data are still lacking.

The NSF BBSRC Imaging Flow Cytometry Project from the Broad Institute at MIT and Harvard has established benchmark datasets together with a set of analytical approaches which they have published over the past 5 years. Of note, they have a publicly available dataset of 32,266 Jurkat cells, an immortalized T cell leukemia cell line, which has been widely used as a benchmark for stain-free computational methods. These cells are used in a multiclass classification task in which methods leverage the IFC images to determine what stage of the cell cycle a given cell is in. In this work, I use the BF images acquired of these 32,266 Jurkat cells as input to different deep learning architectures. A given BF image is processed either with a 2-layer fully-connected (FC) neural network, a 3-layer convolutional neural network (CNN), or a simple vision transformer (Simple ViT). The output of the networks are the predicted cell cycle stage for each cell.

Surprisingly, the 2-layer FC neural network achieved the best balanced accuracy (53.9%), with the best CNN model closely behind (52.6%), and the Simple ViT model performing relatively poorly (20.5%). Saliency maps of each model showed that the FC network and the CNN were able to seg-

ment the cells in the images to different degrees, with the transformer architecture failing to do so. Overall, this work provides an important step in leveraging machine learning for stain-free analysis of IFC images to infer cellular properties.

2. Related Work

Methods focusing on the computational analysis of microscopy images have usually focused on tissue histology slides, cells embedded in slides, or cells in tissue culture plates. Thus, these methods are not particularly suited for the single-cell imaging data provided by an IFC machine, but their application still yields some success on IFC data. All of these methods attempt to extract meaningful features from images, either in a pre-specified or agnostic fashion.

Pre-specified approaches quantify specific cellular characteristics - usually relating to morphology - and aim to use these features for the relevant downstream task. In the context of cells or tissue in slides, the first problem these methods need to solve is cell segmentation - i.e., finding masks which delineate individual cells. Using these masks, each cell can be separately processed to extract the relevant features and use these for downstream analysis [6]. In the context of IFC, cells are already usually in focus and centered, so that segmentation is a less relevant yet still important problem.

The feature extraction paradigm commonly used for IFC images is CellProfiler, a method which first tiles multiple single-cell images in order to leverage already-existing cell segmentation methods and proceeds to extract morphological features from each cell [2]. In the same dataset used in this work, CellProfiler achieved robust performance by using the extracted features followed by LSBoosting as a classification head. However, a key weakness of CellProfiler is that it is biased by the choice of morphological features extracted from cells, which might miss important yet more subtle cellular differences between classes.

A different approach refined CellProfiler's feature extraction by including a feature selection step and tried different classification head methods, such as random forest, yet these approaches suffer from the same fundamental problem of inherent feature bias [7]. In a different setting, CellProfiler was used to classify white blood cells (WBCs) such as T cells, B cells, and neutrophils from each other. In this task, CellProfiler once again achieved robust results [9].

Conversely, methods which extract features in an agnostic manner tend too use deep learning, and usually CNNs of some sort. By leveraging deep convolutional models, meaningful and subtle cellular features are extracted and used for downstream tasks. Such methods have two benefits: CNNs can perform their own segmentation, so that cellular segmentation is no longer necessary; and the embeddings generated by the CNN can be used for a variety of analysis which provide insight into what the model is doing.

In the context of IFC, multiple such approaches have been successfully developed to analyze single-cell imaging data. One approach leveraged a ResNet50 model to classify red blood cells (RBCs) based on their morphology in order to provide insight into the effect of storage time on blood bags [3]. The embeddings found by the model were able to reconstruct the order of RBC degradation, evidence of the potential of visual models in analyzing IFC data. A different study used an autoencoder architecture in which IFC images of T cells were compressed and reconstructed, with the latent space embeddings of the cells used afterwards to determine the health status of the patients participating in the study [10]. Finally, a study on the same dataset presented in this work leveraged an Inception-style architecture to achieve state-of-the-art results in the cell cycle stage classification task [5].

Given the rapid rise in attention, and in particular transformers, as small yet powerful deep learning architectures, leveraging transformers for IFC data is of high interest. Vision transformers have been shown to achieve state-of-the-art performance in visual recognition tasks comparable to deep CNN [4]. This is particularly impressive given the massive reduction in parameters and runtime needed to train transformer models.

3. Dataset

For this project, I will be leveraging the benchmark imaging dataset of 32,266 Jurkat T cells which was generated by Blasi et al. as part of the NSF BBSRC Imaging Flow Cytometry Project from the Broad Institute at MIT and Harvard [2]. Jurkat cells are an immortalized T cell leukemia cell line which is easily grown in cell culture, stained, and analyzed by IFC. These cells are in different stages of the cell cycle, posing a multiclass classification problem with 5 classes. However, this is also a highly imbalanced dataset, given that at any given time most cells in culture are in Interphase and not actively undergoing cell division. Specifically, 31,550 (97.78%) of images belonged to the Interphase (G1/S/G2) class, with Prophase, Metaphase, Anaphase, and Telophase having 606 (1.87%), 68 (0.21%), 15 (0.046%), and 27 (0.084%) cells respectively. For training, this dataset was split 70:10:20 into a training, validation, and test set, and the class distributions within each fold can be see in **Table 1**.

Each cell was imaged using 4 different microscopy channels: bright-field (BF), dark-field (DF), propidium iodide (PI) - a DNA stain - and MPM2 - a mitotic protein marker. Each of these channels produced an image of dimension (3, 66, 66), but only the BF images were used since the problem is particularly focused on stain-free classification. These cells have been previously manually annotated based on their morphology, their DNA content (assessed via the PI

Table 1. Distribution of Jurkat cells across data splits

Stage	Data Split		
	Training	Validation	Test
G1/S/G2	22084	3078	6388
Prophase	414	65	127
Metaphase	45	4	19
Anaphase	10	0	5
Telophase	17	2	8

channel), and their expression of mitotic proteins (assessed via the MPM2 channel). Examples of the images acquired for each channel and for each stage of the cell cycle are displayed in **Figure 1**. For training, images from the training set were randomly flipped vertically and/or horizontally in order to improve the model's generalizability.

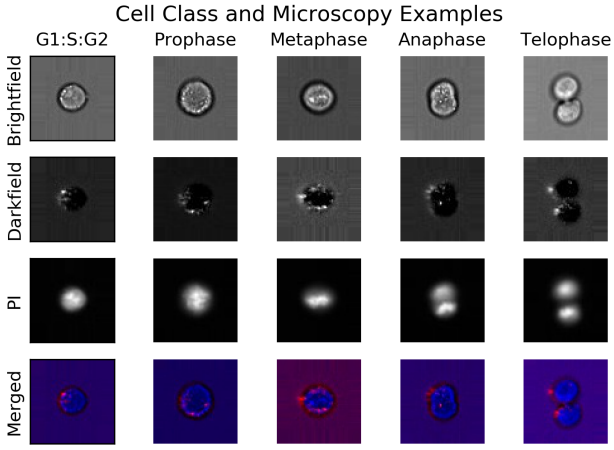

Figure 1. Example images for cells in each cell cycle stage and for each IFC channel acquired

4. Methods

Three different deep learning architectures were used to process the images and perform multiclass classification. Specifically, a FC neural network, a CNN, and a simple vision transformer. These three models specifically were chosen in order to assess the performance of visual learners in this task by having a baseline deep learning model (the FC architecture) and two different visual deep learning models (the CNN and the Simple ViT). Each model was also trained using three different loss functions in order to account for the highly imbalanced dataset.

4.1. Fully-Connected Neural Network

The FC neural network takes as input the un-augmented, flattened pixel values of the image and passes them through several fully-connected hidden layers before calculating the 5 raw class scores. Each of the hidden layers also leverages a batch normalization step and a non-linear activation in the form of a ReLU. After hyperparameter optimization, a 2-layer FC network with hidden dimensions of 100 and 50 was chosen as the final architecture for this model type. This model has 1312505 parameters.

4.2. Convolutional Neural Network

The CNN takes as input a random (64, 64)-pixel crop of the original image which has been randomly flipped vertically and/or horizontally with probability of 0.5. The architecture consists of 3 convolutional layers, with spatial batch normalization, non-linear activation with ReLU, and a max-pooling layer to reduce the dimensionality of the input at each convolution. With each convolution, we also add more filters to add expressivity to the model. The final architecture can be seen in **Figure 2**. This model has only 14181 parameters, making it a compact model with a 100-fold reduction in the number of parameters compared to the 2-layer FC model.

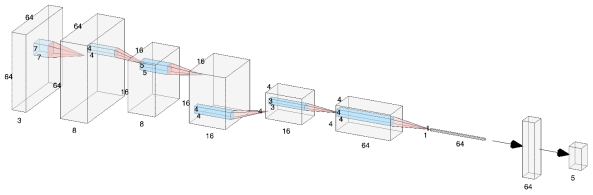

Figure 2. 3-layer CNN architecture

4.3. Vision Transformer

The simple transformer model breaks the image into patches of a given size, projects these patches with a linear encoder, and feeds them into a transformer encoder with positional encoding [4]. Inside the transformer encoder, multiple attention heads allow the network to attend to multiple parts of the image simultaneously. Finally, the output of the transformer encoder is fed to a multi-layer perceptron (MLP) classification head which outputs the scores associated to each class. For this work, I used the Simple ViT architecture outlined by Beyer et al. [1] [1]. After hyperparameter optimization, we used patches of size 16, a 32-dimensional patch projector and MLP classification head, 8 attention heads, and a depth of 2 for the encoder. This model had 160389 parameters, making it 10x smaller than the 2-layer FC model but 10x larger than the 3-layer CNN model.

[1] https://github.com/lucidrains/vit-pytorch

4.4. Loss Functions

Given the highly-imbalanced nature of the dataset, I trained all models with three different loss functions which would address the class imbalance to different degrees. The first loss function used was multiclass cross-entropy loss (CEL), which is defined as

$$L = \frac{1}{N}\sum_{i=1}^{N} -\log(\hat{y}_i)$$

where there are N training examples and $\hat{y}_{k_i}$ is the predicted probability that example i is given for its true label. In this work, we refer to this loss function as the unweighted CEL.

To account for the class imbalance, the loss calculated for each class is re-weighted in a manner inversely proportional to the number of examples for that class. Two different weights were used to penalize the class imbalance to different degrees. The second loss function utilized the inverse of the square root of the number of examples for that class, giving classes with a small number of examples a slightly larger relative contribution to the loss than in unweighted CEL. This loss function is referred to as sqrt_weighted in this work.

The final loss function utilize the inverse of the number of examples for each class, giving all classes equal contribution to the total loss. Given the stark class imbalance, this means the few examples for each of the small classes have a strong guiding effect on the training of the model, while individual examples of the largest class (Interphase) have very little say in guiding model training. This loss is referred to as inv_weighted in this work.

4.5. Other Parameters

All models were trained with dropout with p = 0.25; no weight decay; 5 minibatches per epoch for 10 epochs; an Adam optimizer; and early stopping using balanced accuracy on the validation set. Specifically, if the balanced accuracy in the validation set did not improve after 20 steps (where each minibatch is a single step), the model with the best balanced accuracy on the validation set was kept as the final model. The learning rate of the FC neural network and the Simple ViT was 0.001, while the CNN used a learning rate of 0.01.

5. Results and Discussion

5.1. Model Training

For each of the model architectures used, a model was trained on the training data set using each of the three loss functions as described in **Methods**. As expected given the class imbalance, the vanilla multi-class CEL resulted in models which assigned all images to the Interphase class given the strong dominance that these images had on the loss given their huge numbers. Interestingly, the 2-layer FC model achieved the best performance with the sqrt_weighted, while the CNN and the Simple ViT achived their best performance with the inv_weighted loss. All models stopped via early stopping, with most models stopping training before epoch 7/minibatch 35. The loss and accuracy curves for the best models in each architecture type and with its respective best loss function can be seen in **Figure 3**. Based on the learning curves, none of the models had issues with overfitting on the training set, since the curves for loss and accuracy for the validation and test sets mostly follow those for the trianing set across all three models.

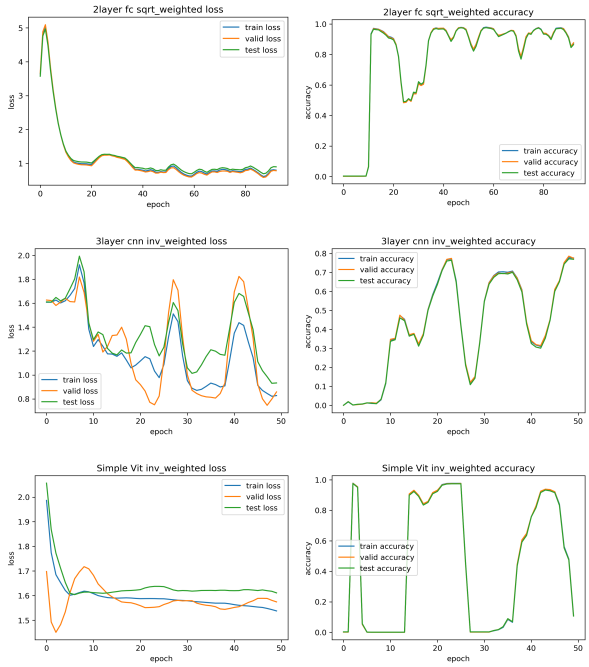

Figure 3. Loss and accuracy across epochs for each model

5.2. Model Performance

Surprisingly, the 2-layer FC model achieved the best performance in the test set, as can be seen in **Table 2**.

Table 2. Results for cell cycle classification task

Method	Balanced Accuracy (%)		
	Training	Validation	Test
2-layer FC	53.64	54.50	**53.89**
3-layer CNN	**55.73**	**57.88**	48.27
Simple ViT	31.86	33.23	20.49

However this performance was closely matched by the best 3-layer CNN model, which had better performance on

the training and validation sets. Disappointingly, the Simple ViT architecture had very poor performance, due to the fact that it predicted most cells were either in Interphase or Telophase. Overall, the performances of the 2-layer FC and 3-layer CNN models, despite their simplicity, is close to the state-of-the-art from Eulenberg et al., which achieves a 57.4% balanced accuracy [5]. However, the Simple ViT architecture gets stuck in the trivial solution.

The confusion matrices for the best models in each architecture in both the training and the test sets can be seen in **Figure 4**. Despite the 2-layer FC network achieving a better balanced accuracy in the test set, the confusion matrices show that the model doesn't predict Metaphase or Anaphase for any cell image, which is a flaw of the model. However, the 3-layer CNN has more varied predictions, and does in fact succeed in correctly classifying at least one image for each class and. On the other hand, as previously mentioned, the Simple ViT model predicts every image as either Interphase or Telophase.

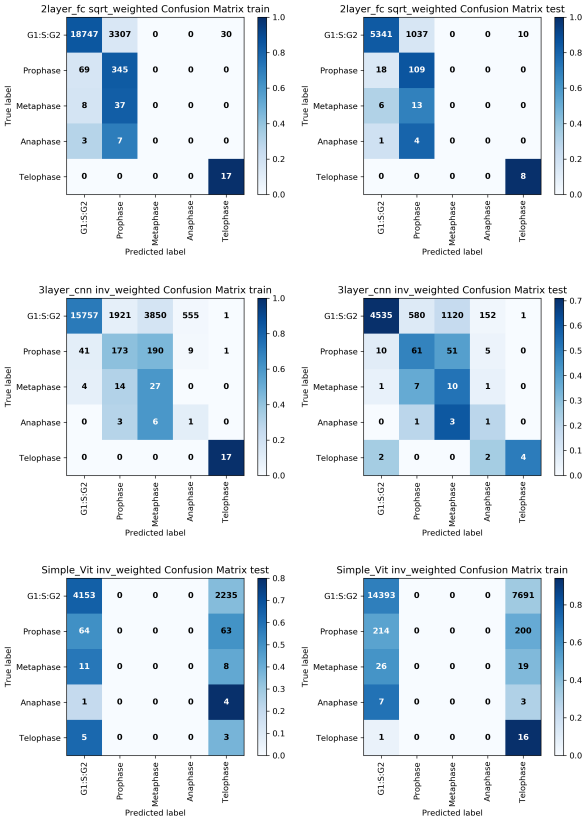

Figure 4. Train and test set confusion matrices for the best model in each model type

5.3. Saliency Map Analysis

In order to better understand the manner through which each of these models was making classification decisions, I generated a saliency map for 4 example cell images. Specifically, given that Interphase and Telophase were two classes with highly-different morphology, I visualized a test set cell from each of these classes in which the 3-layer CNN either correctly or incorrectly classified the image [2].

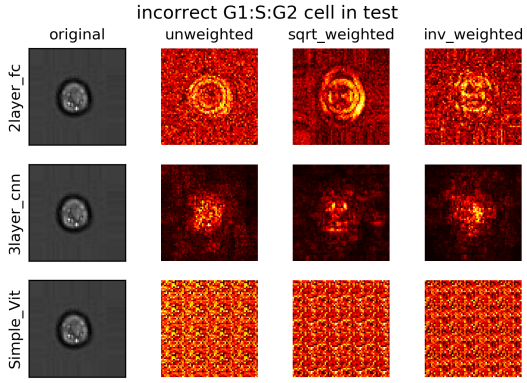

Figure 5. Saliency map for a cell in Interphase (G1/S/G2) incorrectly classified by the 3-layer CNN model

The saliency map for the incorrectly and correctly classified cells in Interphase can be seen in **Figure 5** and **Figure 6** respectively. Immediately it is apparent that the Simple ViT model, which operates via patches, is not learning meaningful features from the input image, explaining its poor performance. However, it is also clear from the maps for the 2-layer FC and 3-layer CNN models that these models are trying to segment the cells, given that the pixels which are contributing the most to the prediction are roughly located where the cell is in the image.

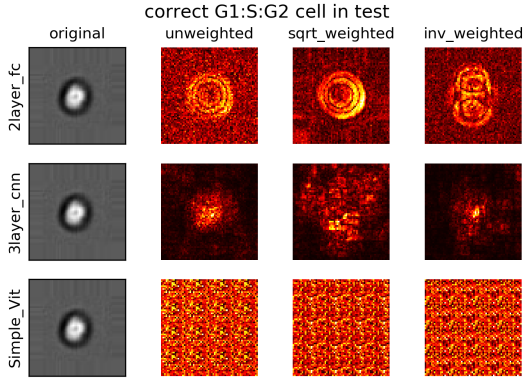

Figure 6. Saliency map for a cell in Interphase (G1/S/G2) correctly classified by the 3-layer CNN model

Interestingly, the 2-layer FC saliency maps show a very clear depiction of the outline and interior of the cell. On the

[2]Saliency map code taken from CS231N assignment 2

other hand, the 3-layer CNN has a fuzzier segmentation, possibly due to the reduced depth of this model translating to coarser visual features. For the 3-layer CNN, the saliency map segmentation in the incorrectly-classified cell shows a broader signal than truly corresponds to the cell in the image. This might be the reason for the classification error, since larger cells tend to be undergoing the cell cycle and thus belong to a different class, as can be seen in **Figure 1**. The correctly-classified cell shows a much more concentrated set of pixels relevant for the final score.

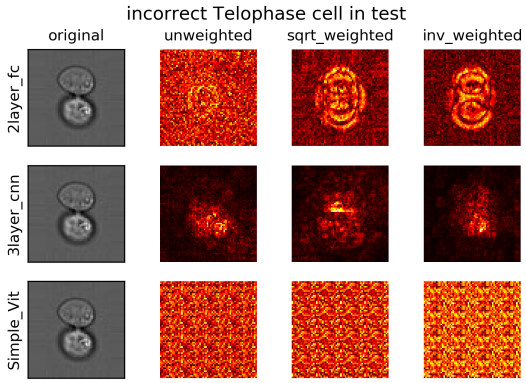

Figure 7. Saliency map for a cell in Telophase incorrectly classified by the 3-layer CNN model

When looking at two Telophase cells which were incorrectly and correctly classified by the 3-layer CNN in **Figure 7** and **Figure 8**, a similar segmentation phenomena is observed. Again, the Simple ViT model doesn't achieve meaningful feature extraction, as can be seen by the uninterpretable saliency map. Similarly to the previous analysis, the 2-layer FC model achieves a very clear segmentation of the two budding daughter cells in the image.

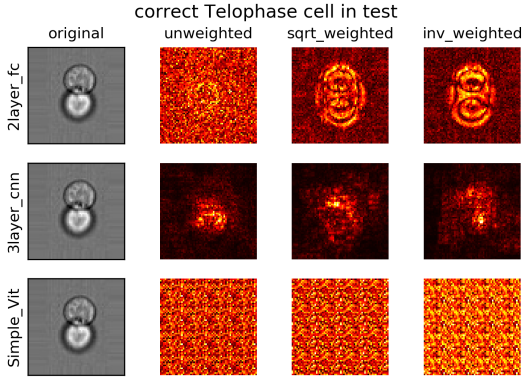

Figure 8. Saliency map for a cell in Telophase correctly classified by the 3-layer CNN model

The 3-layer CNN fails to segment the cells into two daughter cells as can be seen in the saliency maps for the incorrect classification example, where there is only 1 peak of signal intensity. However, in the correctly classified example, the 3-layer CNN has two different peaks of pixel intensity in the saliency map, suggesting that it has identified that there are in fact two cells and that therefore the image must be of cells in Telophase.

5.4. Embedding Analysis

In order to understand the structure of the data generated by the final embeddings of the images by the 3-layer CNN model, I extracted the final 64-dimensional layer of this model and used the UMAP dimensionality reduction algorithm to visualize the cell embeddings in 3-dimensional space [8]. The structure of the data can be seen in **Figure 9**, where Interphase cells have been randomly down-sampled after dimensionality reduction to improve the visualization quality.

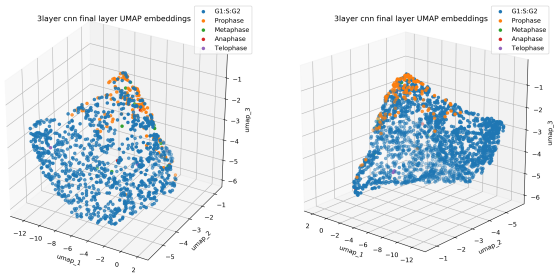

Figure 9. UMAP 3-dimensional representation of the test set image embeddings of the final layer of the 3-layer CNN

From the UMAP visualization, it is clear that the final layer embeddings provide a robust separation of the two largest classes (Interphase and Prophase, in blue and orange respectively). There is also a cluster of Telophase cells, explaining why the Telophase class, despite its low numbers, was fairly accurately predicted by this model. Overall, this visualization confirms that the 3-layer CNN extracted meaningful cellular features useful for separating the different cell cycle stages.

6. Conclusions and Future Work

Overall, this work shows that deep learning can be used to predict cell cycle stage in Jurkat cells using only BF images acquired by an IFC. Furthermore, this work shows that simple models can achieve close to state-of-the-art performance in this task, given that the 2-layer FC model and the 3-layer CNN model were both close in balanced accuracy to the 42-layer deep model from Eulenberg et al. Particularly, all the models used in this work were trained in under 30 minutes by my personal computer, where the model

from Eulenberg et al trains in 7 hours with higher compute power. Disappointingly, the vision transformer architecture from Simple ViT was not able to make meaningful progress in this task, possibly due to the reduced compute power and time available.

A key limitation of this work is the highly imbalanced dataset used for the multiclass classification task, which severely constrained the expressivity of the models and made training harder. This limitation could be addressed by gathering more IFC images, given that these are cheap to generate. The existing images could also be augmented more thoroughly. Alternatively, a pretrained ResNet model or a model pretrained on other BF IFC datasets could be leveraged with a transfer learning approach.

Future work should focus on exploring whether state-of-the-art performance can truly be achieved with similarly-size small deep learning models, and, if not, what the ideal depth is for models trained for this task. Similarly, deeper models and more compute could be useful for achieving non-trivial results with the vision transformer.

References

[1] Lucas Beyer, Xiaohua Zhai, and Alexander Kolesnikov. Better plain vit baselines for imagenet-1k, 2022. 3

[2] Thomas Blasi, Holger Hennig, Huw D. Summers, Fabian J. Theis, Joana Cerveira, James O. Patterson, Derek Davies, Andrew Filby, Anne E. Carpenter, and Paul Rees. Label-free cell cycle analysis for high-throughput imaging flow cytometry. *Nature Communications*, 7(1):10256, Jan. 2016. 2

[3] Doan Minh, Sebastian Joseph A., Caicedo Juan C., Siegert Stefanie, Roch Aline, Turner Tracey R., Mykhailova Olga, Pinto Ruben N., McQuin Claire, Goodman Allen, Parsons Michael J., Wolkenhauer Olaf, Hennig Holger, Singh Shantanu, Wilson Anne, Acker Jason P., Rees Paul, Kolios Michael C., and Carpenter Anne E. Objective assessment of stored blood quality by deep learning. *Proceedings of the National Academy of Sciences*, 117(35):21381–21390, Sept. 2020. Publisher: Proceedings of the National Academy of Sciences. 2

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020. 2, 3

[5] Philipp Eulenberg, Niklas Köhler, Thomas Blasi, Andrew Filby, Anne E. Carpenter, Paul Rees, Fabian J. Theis, and F. Alexander Wolf. Reconstructing cell cycle and disease progression using deep learning. *Nature Communications*, 8(1):463, Sept. 2017. 2, 5

[6] Noah F. Greenwald, Geneva Miller, Erick Moen, Alex Kong, Adam Kagel, Thomas Dougherty, Christine Camacho Fullaway, Brianna J. McIntosh, Ke Xuan Leow, Morgan Sarah Schwartz, Cole Pavelchek, Sunny Cui, Isabella Camplisson, Omer Bar-Tal, Jaiveer Singh, Mara Fong, Gautam Chaudhry, Zion Abraham, Jackson Moseley, Shiri Warshawsky, Erin Soon, Shirley Greenbaum, Tyler Risom, Travis Hollmann, Sean C. Bendall, Leeat Keren, William Graf, Michael Angelo, and David Van Valen. Whole-cell segmentation of tissue images with human-level performance using large-scale data annotation and deep learning. *Nature Biotechnology*, 40(4):555–565, Apr. 2022. 2

[7] Holger Hennig, Paul Rees, Thomas Blasi, Lee Kamentsky, Jane Hung, David Dao, Anne E. Carpenter, and Andrew Filby. An open-source solution for advanced imaging flow cytometry data analysis using machine learning. *Flow Cytometry*, 112:201–210, Jan. 2017. 2

[8] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2018. 6

[9] Mariam Nassar, Minh Doan, Andrew Filby, Olaf Wolkenhauer, Darin K. Fogg, Justyna Piasecka, Catherine A. Thornton, Anne E. Carpenter, Huw D. Summers, Paul Rees, and Holger Hennig. Label-Free Identification of White Blood Cells Using Machine Learning. *Cytometry Part A*, 95(8):836–842, Aug. 2019. Publisher: John Wiley & Sons, Ltd. 2

[10] Corin F. Otesteanu, Martina Ugrinic, Gregor Holzner, Yun-Tsan Chang, Christina Fassnacht, Emmanuella Guenova, Stavros Stavrakis, Andrew deMello, and Manfred Claassen. A weakly supervised deep learning approach for label-free imaging flow-cytometry-based blood diagnostics. *Cell Reports Methods*, 1(6):100094, Oct. 2021. 2