Project 2

Jiahui Zhu

October 2024

Abstract

This study investigates the prediction of cellular responses to drug treatments using a multi-modal deep learning approach applied to segmented microscopy images. Utilizing a subset of the CPJUMP1 dataset, the model combines EfficientNet-based branches for mask and flow images with a fully connected network for tabular data. An attention mechanism is employed to enhance feature selection, and a majority voting scheme aggregates cell-level predictions for robust image-level classification.

The findings demonstrate that training on segmented cell images allows the model to capture detailed cellular features, which are critical for accurate drug-response predictions at the image level. Despite limited computational resources and a constrained training duration, the model achieved promising image-level accuracy, illustrating the effectiveness of multi-input architectures in bioinformatics applications. Future work could focus on extended training and improved feature extraction methods, which hold potential for further enhancing predictive accuracy in complex cellular data analysis.

Introduction

This project explores the response of cells to various drug treatments using machine learning and image analysis techniques. The study utilizes a dataset derived from the CPJUMP1 experiments, as detailed in the work by Chandrasekaran et al. (2024), involving large-scale cellular imaging [1]. The original dataset comprises 22,000 images from eight fluorescently labeled channels. However, for this project, a filtered subset of 2,867 images was provided. These images were median-aggregated across the first three channels to enhance robustness and facilitate downstream segmentation tasks

The primary objective was to predict the type of drug treatment based on these aggregated microscopy images. To achieve this, a multi-modal approach was implemented, combining image data (mask and flow images) processed through modified EfficientNet-based convolutional neural networks (CNNs) and tabular features processed through a fully connected network. An attention mechanism was integrated to emphasize significant features, and the model was trained using label smoothing cross-entropy loss to improve generalization and mitigate overfitting, with optimization performed using Adam and a StepLR scheduler to adjust the learning rate every 5 epochs.

To enable image-level prediction, the training was conducted on segmented cell images obtained from Cellpose segmentation. This approach allowed for focused learning on individual cellular structures within each image, which were then aggregated to form an overall prediction. Final image-level predictions were made through a majority voting scheme, wherein the predictions from segmented cell instances within each image were combined, with the majority class determining the final label.

Methods

Part 1: CellPose Segmentation

Cell images were segmented using **Cellpose** (version 2.3.2) within a Python 3.11.9 and **PyTorch** 2.3.1 environment. Cell images were imported and formatted using `skimage.io` to ensure uniform channel structure. Segmentation was performed with `model.eval()`, using a flow threshold of 0.7 and a cell probability threshold of -5. The diameter was set to `None` for automatic cell size estimation. Masks and flow images were saved in TIFF format. The morphological and intensity features of each segmented cell were extracted using the `regionprops` function from `skimage.measure` and compiled into a CSV file.

Part 2: Extracted Cell Features and Metadata Cleaning

Extracted Cell Features Dataframe

A comprehensive set of tabular features for each cell in the image was extracted, including *area*, *convex area*, *perimeter*, *eccentricity*, *mean intensity*, *min intensity*, *max intensity*, *major axis length*, *minor axis length*, *orientation*, *solidity*, *extent*, and *centroid*. Additionally, *CoreFileName*, *label*, and *bounding box* (*bbox*) were recorded for downstream analyses. The resulting DataFrame had a shape of (491426 × 16). The *centroid* column was split into separate *centroid_x* and *centroid_y* columns, and the original column was removed. During data inspection, anomalous entries with a *perimeter* value of 0.0 were identified and removed due to their non-informative nature, leaving final Dataframe shape of (462008 x 17).

Metadata Dataframe

The *CoreFileName* was extracted from the *FileName_OrigRNA* column by retaining only the prefix (e.g., for *r01c01f01p01-ch3sk1fk1fl1.tiff*, only *r01c01f01* was retained as it shares a common suffix). Only the columns *metadata_target* and *metadata_pert_iname* were retained as they contained informative data. The *metadata_target* column provided information on the gene targeted by the compound, while *metadata_pert_iname* indicated the name of the chemical compound used as a perturbation. Rows with metadata entries not corresponding to image data were removed to ensure consistency.

Part 3: Model training

1. Data Preparation

Downsampling on 'DMSO' instances and Stratified Split Based on Metadata The dataset was found to be heavily dominated by 'DMSO' instances (509 in total), which represent the control group. To ensure balanced training and evaluation, downsampling was applied to the 'DMSO' instances to match the number of samples in the second-largest class (18 in total). A stratified split was then conducted using `StratifiedShuffleSplit` with one split (`n_splits=1`) and a test size of 20%, ensuring proportional representation of classes in both training and test sets. To prepare categorical metadata features for model input, `LabelEncoder` was applied to the `Metadata_pert_iname` and `Metadata_target` columns. Separate encoders were initialized for each column.

Transformations Image transformations are applied to both mask and flow images, which are resized to 224 × 224 pixels and normalized to a mean of $(0.5, 0.5, 0.5, 0.5)$ and a standard deviation of $(0.5, 0.5, 0.5, 0.5)$ for four-channel inputs. These transformations ensure consistency across images and improve convergence during training.

Custom Dataset Class A custom dataset class, `CellDataset`, is created to handle multi-modal data including mask images, flow images, and tabular features. Mask and flow images are read from specified folders, cropped according to bounding box information, and transformed as described above. Tabular features, extracted from the dataset and scaled using `StandardScaler`, are used to normalize the numerical data. The dataset class also retrieves labels and metadata necessary for training.

Prefetching and Data Loading To optimize data loading, a `PrefetchLoader` class is used to pre-load batches onto the GPU asynchronously, reducing data transfer latency. Data loaders are instantiated with batch size 512, using 20 worker processes and `pin_memory` for faster host-to-device data transfer. The `PrefetchLoader` wraps around the `DataLoader` to enhance efficiency during model training, improving GPU utilization.

2. Model Architecture

The proposed model is a multi-input convolutional neural network (CNN) designed for compound prediction tasks, integrating image-based and tabular data inputs. The model architecture consists of three main branches: two separate EfficientNet-based modules for mask and flow image processing and a fully connected module for tabular data. The outputs from these branches are aggregated through an attention mechanism to enhance feature representation before final classification.

Image Processing Branches The model utilizes two parallel branches based on EfficientNet-B0, each modified to accept four-channel input images. The mask and flow branches independently process mask and flow images, respectively. For each branch, the first convolutional layer is modified to handle four input channels by replacing the initial 3-channel convolutional layer with a 4-channel configuration. Each branch extracts features from the input images using the pre-trained EfficientNet model, where the spatial dimensions of the feature maps are subsequently reduced via global average pooling to produce compact feature representations.

Tabular Data Processing Tabular data is processed through a fully connected neural network. This module comprises a linear layer with ReLU activation, followed by batch normalization and dropout to reduce dimensionality and enhance generalization. The tabular input vector is transformed into a 32-dimensional feature representation, which is concatenated with the mask and flow image features for subsequent processing.

Attention Mechanism To capture salient information from the combined feature representations, an attention mechanism is applied to the concatenated features derived from the mask, flow, and tabular data branches. The attention module consists of a two-layer fully connected network with a ReLU activation, followed by a sigmoid activation to generate attention weights. These weights modulate the combined features through element-wise multiplication, amplifying informative features and suppressing redundant information.

Feature Projection and Classification The attended feature vector is then projected to a lower-dimensional space via a fully connected projection layer, applying a ReLU activation for feature extraction. Finally, a linear classification layer maps the projected features to the output class space, yielding the predicted compound class probabilities.

Loss Function The model employs **Label Smoothing Cross-Entropy Loss** as the objective function. By redistributing a small portion of the probability mass to all classes, this approach reduces overconfidence in predictions.

Optimization and Regularization The optimization of the model parameters is conducted using the Adam optimizer with a learning rate of 0.0001 and a weight decay of 1×10^{-4}, which helps mitigate overfitting by penalizing large weights. A learning rate scheduler, `StepLR`, is applied to adjust the learning rate dynamically during training. This scheduler reduces the learning rate by a factor of 0.5 every 5 epochs, allowing for finer adjustments as training progresses, thereby stabilizing convergence and enhancing model performance. Early stopping is used to monitor validation loss and prevent overfitting.

Evaluation During evaluation, predictions made at the cell level are aggregated to the image level via majority voting, and accuracy metrics are calculated to assess overall model performance.

Part 4: Code Availability

The code used for analysis is available at: GitHub Repository.

Results

Example of CellPose Segmentation results

Figure 1displays a sample segmentation result for cellular images. The panels illustrate (from left to right): the original image, the predicted cell outlines, the predicted segmentation masks, and the predicted cell poses. The segmentation accurately delineates individual cells, with distinct color-coded regions representing different cellular boundaries and poses. Additionally, the mask and flow images, along with bounding box coordinates, are saved for further analysis to facilitate downstream tasks

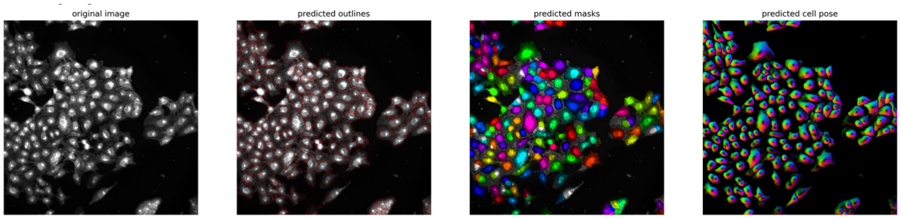

Figure 1: Segmentation result of sample r06c15f08 following treatment with compound BRL-50481

Training Results

The training and validation performance of the model across 13 epochs is summarized in Table 1. The training loss steadily decreased from an initial value of 5.3658 to 4.3886, reflecting progressive learning on the training data. Correspondingly, training accuracy improved from 1.68% to 13.25%, suggesting that the model gradually learned to fit the training dataset.

Despite these improvements in training performance, the validation results demonstrated limited generalization. Validation loss showed only a modest reduction from 5.4119 to 5.0762 and fluctuated slightly throughout the training

Table 1: Training and Validation Performance over 13 Epochs

Epoch	**Train Loss**	**Train Accuracy**	**Validation Loss**	**Validation Accuracy**
1	5.3658	0.0168	5.4119	0.0129
2	5.1366	0.0354	5.2110	0.0278
3	4.9924	0.0502	5.1600	0.0298
4	4.8862	0.0620	5.0783	0.0394
5	4.8017	0.0715	5.0700	0.0381
6	4.7045	0.0849	5.0696	0.0405
7	4.6549	0.0908	5.0623	0.0407
8	4.6068	0.0981	5.0316	0.0431
9	4.5602	0.1044	5.0397	0.0450
10	4.5122	0.1121	5.0451	0.0449
11	4.4465	0.1230	5.0539	0.0445
12	4.4177	0.1268	5.0720	0.0449
13	4.3886	0.1325	5.0762	0.0451

Note: Early stopping was applied at epoch 13.

process. Validation accuracy remained low, increasing from 1.29% to 4.51%, which highlights the model's challenge in generalizing beyond the training set.

The learning rate scheduler, which reduced the learning rate by half every 5 epochs, may have constrained further improvements, particularly in validation performance. While the reduction in learning rate typically aids convergence, the scheduler's fixed-step decay might have prematurely limited the learning capacity.

Due to limited computational resources, the training process was restricted to 13 epochs, and early stopping was applied at this point due to plateauing validation performance. This limitation may have constrained the model's ability to improve further on validation accuracy. Future work with extended training epochs and refined hyperparameters, potentially including a more adaptive learning rate strategy, could be beneficial for improved generalization and validation accuracy.

Image-Level Prediction

In the final image-level prediction task, the model achieved an accuracy of 17.44% after only 13 epochs. This task involved training on segmented cell images obtained from Cellpose segmentation, allowing the model to focus on learning distinctive cellular structures within each image.

For the final image-level predictions, a majority voting scheme was employed, wherein predictions from each segmented cell instance within an image were aggregated, and the majority class determined the final label. This approach leveraged the diversity of cell-level predictions to enhance the robustness of the overall image-level classification. Although cell-level prediction accuracy on the validation set was relatively low at 4.51%, the individual cell predictions provided a nuanced representation that, when aggregated, improved the final image-level accuracy.

In summary, while individual cell predictions posed classification challenges, their aggregation through majority voting allowed the model to produce a stronger, more consistent image-level prediction. This method demonstrates how cell-level segmentation and focused learning can contribute positively to image-level tasks, suggesting that further refinement of cell segmentation and feature extraction could enhance both cell- and image-level prediction accuracy.

Conclusion

This project addresses the challenging task of predicting cellular responses to various drug treatments based on microscopy images. The model implemented is a multi-input CNN that integrates EfficientNet-based branches for processing mask and flow images, alongside a fully connected network for tabular data. An attention mechanism enhances key features from these inputs, which are then combined in a final classification layer. To improve generalization, the model utilizes Label Smoothing Cross-Entropy Loss and is optimized using Adam with a StepLR learning rate scheduler.

Training on segmented cell images allows the model to capture specific cellular features that form the basis for reliable image-level classification. By employing a majority voting scheme, predictions from individual cell segments are aggregated to determine the final label for each image, thereby leveraging diverse cellular perspectives within each image. This approach effectively strengthens image-level predictions by incorporating detailed cell-level insights.

This work highlights the potential of multi-modal deep learning for complex cellular data analysis, even within computational limits. Future work, including extended training and adaptive learning rate strategies, could further enhance model performance. Improvements in cell segmentation and feature extraction would also likely increase cell-level accuracy, contributing to more precise and robust image-level predictions.

References

[1] Srinivas Niranj Chandrasekaran, Beth A. Cimini, Amy Goodale, Lisa Miller, Maria Kost-Alimova, Nasim Jamali, John Doench, Briana Fritchman, Adam Skepner, Michelle Melanson, John Arevalo, Juan C. Caicedo, Daniel Kuhn, Desiree Hernandez, Jim Berstler, Hamdah Shafqat-Abbasi, David Root, Sussane Swalley, Shantanu Singh, and Anne E. Carpenter. Three million images and morphological profiles of cells treated with matched chemical and genetic perturbations. *bioRxiv*, 2022.

Supplementary Materials

Sanity Check: Assessing Morphological Differences Between the Control Group (DMSO) and Compounds

To evaluate whether there were significant morphological differences between the control group (DMSO) and compound-treated samples, a simple logistic regression classifier was trained using the extracted morphological features. The model achieved an accuracy of 0.79, suggesting a reasonable ability to distinguish between DMSO-treated and compound-treated samples based on the extracted features. This result indicates that the morphological characteristics captured in the feature set provide meaningful distinctions between the control and treatment groups.