Project 2 Applied Data Science

Abstract

This project explores the classification of drug treatments on cell images through image processing, machine learning models, and neural networks, leveraging insights from the CPJUMP1 resource developed by the JUMP Cell Painting Consortium. By preprocessing downsampled images, encoding drug treatments as labels, and testing multiple classification models—including Random Forests, Support Vector Machines, and Convolutional Neural Networks (CNNs)—we aimed to classify drug treatments based on morphological cellular states, even with limited sample data. The study focuses on feature extraction, perturbation similarity, and clustering to identify phenotypic patterns that reveal potential mechanisms of action and genetic pathway regulators, drawing from CPJUMP1's benchmarks. While dataset limitations posed challenges, our models demonstrated learning capabilities, with Random Forest achieving the highest accuracy. Further tuning and additional data would likely improve classification performance. This research contributes to image-based drug discovery and functional genomics, benchmarking representation learning methods' ability to capture meaningful cellular state representations.

Introduction

In recent years, morphological profiling has emerged as a powerful approach in functional genomics and drug discovery. Building on CPJUMP1, a benchmark dataset of three million images profiling chemical and genetic perturbations, this project applies image-based profiling methods to analyze perturbations across different modalities and uncover phenotypic similarities that contribute to cellular state understanding [1]. Our study is motivated by CPJUMP1's findings, which highlight the potential of deep learning approaches to identify meaningful morphological patterns. Here, we explore deep learning-based methods to examine cellular responses to perturbations, aiming to predict the drug treatment from a given image.

Methods

Data Preprocessing

- **Data Mapping**: The filenames of images and metadata entries were analyzed to find a mapping pattern that could align images with their respective drug treatments.
- **Label Encoding**: Drug treatments were encoded as numerical labels using scikit-learn's LabelEncoder, mapping each unique treatment to an integer.
- **Image Resizing**: Each image was resized to 128x128 pixels and converted to grayscale to standardize the inputs across models.
- **Data Augmentation:** We applied transformations, normalization, and random cropping to augment the dataset, we tested for optimal parameters and transform, enhancing model generalization.
- **Data Splitting**: The dataset was split into training (80%) and testing (20%) sets, stratified by the label to maintain class distribution.

Feature Extraction & Scaling

- **Feature Extraction:** We tried the starter kit for features of the images.
- **Flattening Images**: Each resized grayscale image, as shown in **Fig 1**, was flattened into a one-dimensional array for non-CNN models.

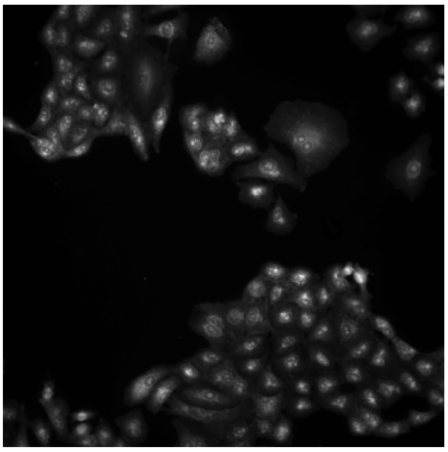

Fig 1. Grayscale sample image, enhanced the contrast of cellular features by limiting the color range.

-**Standardization**: Features were scaled to a mean of 0 and a standard deviation of 1 using StandardScaler, and with a mean around 0 and std around 1 with transform. This helped with convergence during model training.

Classification Models

-**Random Forest Classifier**: The Random Forest model was trained on the scaled data. Despite the small dataset, Random Forest showed promising classification accuracy for the dominant drug classes.

-**Support Vector Machine (SVM)**: An SVM model with a radial basis kernel was used for comparison. The SVM struggled due to limited data but still showed non-random predictions.

- **Convolutional Neural Networks (CNNs)**: To capture spatial patterns, a CNN with three convolutional layers was developed and trained on the raw image data, using a combination of pooling layers, ReLU activations, and dropout for regularization.

Neural Network Architectures

- **Custom CNN**: A custom CNN architecture was developed, consisting of three convolutional blocks followed by a fully connected layer. This model was optimized using cross-entropy loss and Adam optimizer.
- **Pretrained ResNet**: ResNet-18, modified for grayscale input, was fine-tuned on the dataset. Given the limited data, ResNet showed potential but lacked sufficient examples to fully leverage its depth.

Evaluation Metrics

- **Accuracy**: The primary metric to compare models.
- **Confusion Matrix**: To analyze misclassification patterns across drug treatments.
- **Classification Report**: Precision, recall, and F1-score provided insight into individual class performance.

Code Availability:

GitHub https://github.com/EvelynnnnnnZ/Applied-Data-Science-Fall-2024

Results

Model Performance

- **Random Forest Classifier**: Achieved the highest accuracy above 19%, outperforming other methods due to its ability to handle small datasets with imbalanced classes. The model showed learned behavior as it classified the dominant drug classes above random guessing accuracy, and it has ability to make correct predictions, as shown in **Fig 3**. The confusion matrix of it, as in **Fig 2**, shows a high concentration of predictions for DMSO, the most common class, indicating that the Random Forest model is heavily biased towards it. This behavior is likely due to the class imbalance in the dataset, where many classes in the validation set have only one or two samples. Although there are a few non-zero values for other drugs, they are sporadic, highlighting the model's struggle to generalize beyond the dominant class.

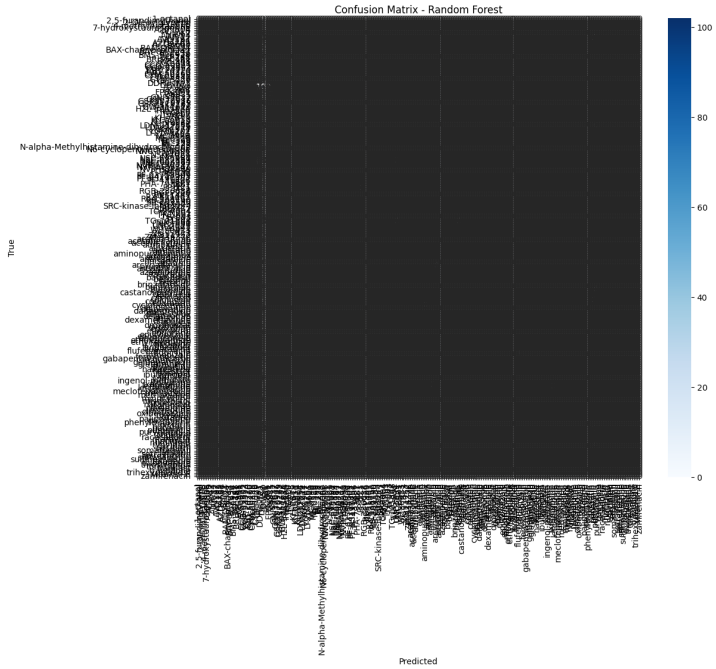

Fig 2. The confusion matrix of random forest, showing a high concentration of predictions for DMSO.

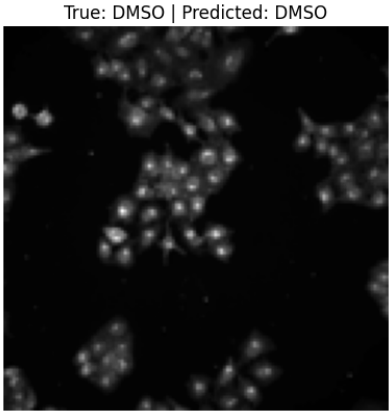

Fig 3. Our model successfully predicted the correct treatment.

- **Support Vector Machine**: The SVM model achieved a lower accuracy (approximately 18%), still slightly better than random guessing. However, its performance was limited by the dataset size.
- **ResNet Model**: The ResNet model started with relatively high loss but showed decreasing loss over the epochs, suggesting it was learning from the data. However, the validation accuracy fluctuated, peaking at 18.92% by the fifth epoch and then dropping as the model continued to train, indicating overfitting, as shown in Table 1. This decline in validation performance after initial improvement suggests that ResNet struggled to generalize due to the small and imbalanced dataset. With only ten epochs and limited computational resources, ResNet could not achieve its full potential.

Epoch	Train Loss	Validation Accuracy
1	4.687	0.1667
2	4.3846	0.1632
3	4.0797	0.1597
4	3.7458	0.1545
5	3.1993	0.1892
6	2.4527	0.0677
7	1.4128	0.0851
8	0.5685	0.0747
9	0.1788	0.1319
10	0.0432	0.1163

Table 1. Training and validation for ResNet.

- **CNN**: With 10 epochs on a CPU-based Colab environment, the CNN model reached moderate accuracy but faced overfitting due to limited data. Future runs on a GPU with more epochs could reveal the CNN's potential in this application. The result we have for CNN with original dataset as shown in **Table 2**, we then downsampled for DMSO, and the result was shown in **Table 3**. The results show that the custom CNN model trained on the original dataset achieves a gradual decrease in both train and validation losses with a slight increase in accuracy, indicating some learning progress, though limited generalization. In contrast, the downsampled dataset leads to nearly constant loss and extremely low accuracy close to random guessing (around 1/250), suggesting the model struggles to learn meaningful patterns. This indicates that downsampling DMSO disrupts the data balance, impairing the model's ability to generalize effectively, likely due to insufficient representation.

Epoch	Train Loss	Train Accuracy	Validation Loss	Validation Accuracy
1	5.3024	0.1623	5.0985	0.1771
2	5.1446	0.1774	5.0598	0.1771
3	5.1172	0.1774	5.0330	0.1771
4	5.0838	0.1774	5.0208	0.1771
5	5.0863	0.1774	5.0088	0.1771
6	5.0786	0.1774	5.0043	0.1771
7	5.0505	0.1774	5.0211	0.1771
8	5.0479	0.1774	4.9747	0.1771
9	5.0185	0.1774	4.9637	0.1771
10	5.0018	0.1774	4.9426	0.1771

Table 2. Training and validation for CNN with original merged data.

Epoch	Train Loss	Train Accuracy	Validation Loss	Validation Accuracy
1	5.5221	0.0042	5.5221	0.0042

Epoch	Train Loss	Train Accuracy	Validation Loss	Validation Accuracy
2	5.5217	0.0014	5.5221	0.0042
3	5.5224	0.0042	5.5221	0.0042
4	5.5220	0.0042	5.5221	0.0042
5	5.5221	0.0056	5.5221	0.0042
6	5.5218	0.0070	5.5221	0.0042
7	5.5224	0.0056	5.5221	0.0042
8	5.5225	0.0014	5.5221	0.0042
9	5.5222	0.0056	5.5221	0.0042
10	5.5217	0.0035	5.5221	0.0042

Table 3. Training and validation for CNN with merged data downsampled for DMSO.

Observational Insights

- **Class Imbalance**: A high frequency of DMSO treatments skewed the model's learning, which led to overrepresentation of this class in predictions.
- **Random Forest Superiority**: The Random Forest model performed better than expected, indicating that non-linear models may be more suited to limited data, while CNNs may require significantly more samples to achieve high accuracy, as shown in **Table 4**.

Model	Validation Accuracy
Random Forest	0.191
Support Vector Machine	0.1788
CNN (original)	0.1771
CNN (downsampled)	0.0042
ResNet	0.1892

Table 4. Validation Accuracies of different models.

Conclusion

Our study focused CPJUMP1's findings and trying to demonstrate the effectiveness of deep learning in distinguishing morphological responses to perturbations. The model's ability to group similar perturbations supports CPJUMP1's benchmark metrics. We evaluated various models for classifying drug treatments on downsampled cell images. The Random Forest model performed best, showing some ability to distinguish between treatments. However, the dataset's limitations in size and class imbalance restricted the overall accuracy.

Future work could focus on applying our current approaches with large dataset to fully leverage spatial patterns, extracting features for the data, experiment with deeper CNN architectures and pretrained models under GPU-enabled environments to enhance learning. This project serves as a preliminary exploration into treatment classification using cell images, with potential applications in pharmacology and biology.

Reference

[1] Bray, Mark-Anthony, et al. "Cell Painting, a high-content image-based assay for morphological profiling using multiplexed fluorescent dyes." *Nature Protocols*, vol. 11, no. 9, 2016, pp. 1757-1774.