Project 2 of Appied Data Science

Zihan Tang

November 2024

Abstract

This project focuses on classifying chemical perturbations among 2,867 bio-images to uncover patterns in cell morphology. Two classification approaches were explored: (1) a Simple CNN applied directly to the original, labeled images and (2) a two-step model involving image segmentation followed by classification with a Simple CNN on the segmented images. As anticipated, the segmented-image model outperformed the direct classification approach, effectively distinguishing chemical perturbations. This method offers valuable insights into cell morphology and enhances our ability to detect meaningful patterns and relationships influenced by chemical perturbations.

1 Introduction

Background: The study ”Three Million Images and Morphological Profiles of Cells Treated with Matched Chemical and Genetic Perturbations” used machine learning and image processing to analyze microscopy images, identifying patterns and relationships between genetic modifications and drug treatments. This work sets a benchmark for measuring similarities and effects of cellular perturbations, showcasing the potential of image-based profiling for applications like drug mechanism discovery and functional genomics.

Goal: Building upon this foundational work, our project aims to replicate machine learning methodologies for image classification and analysis from scratch in Python.We explore two approaches: a Simple CNN applied to raw images and a refined pipeline using image segmentation followed by classification to enhance accuracy in detecting chemical perturbations.

Data: We used a subset of the original three-million-image dataset, focusing on one batch and extracting 2,867 median-aggregated grays images, a robust dataset suited for downstream segmentation tasks. For testing, we created a smaller dataset of five images with specified chemical perturbation or non-perturbation labels, stored in the metadata.

2 Methods

2.1 Load and Normalize Dataset

The training and testing images are stored in separate files without being categorized by label (perturbation status). To facilitate data handling and ensure efficient loading, we created a DataFrame that contains the image names, file paths, and labels from the metadata.In preparing the dataset for model training, we used stratified sampling to reserve 20% of the training images as a validation set, preserving the distribution of perturbation labels.

Finally, we applied transformations to the images, resizing them to 256*256 pixels and normalizing the pixel values to a mean of 0.5 and a standard deviation of 0.5. These transformations help standardize the input data for the model, improving training stability and performance.

2.2 Define a Convolutional Neural Network

We implemented two Convolutional Neural Network (CNN) pipelines for classification: one without segmentation, which directly classifies the original images (referred to as SimpleCNN), and one that segments the images before classification (referred to as SegCNN).

2.2.1 SimpleCNN

- **Loss Function:** Binary Cross-Entropy Loss (BCELoss) was chosen to handle the binary classification of perturbation presence.

- **Optimizer:** The Adam optimizer was employed to train the network and adjust the weights efficiently.

2.2.2 SegCNN

- **Segmentation:** We utilized the OpenCV library to identify cell contours in each image. The segmented regions were assigned the same label as the original image.

- **Loss Function:** Binary Cross-Entropy Loss (BCELoss) was chosen to handle the binary classification of perturbation presence.

- **Optimizer:** The Adam optimizer was employed to train the network and adjust the weights efficiently.

2.3 Train and Test the Network

2.3.1 Train process

The models were trained with a batch size of 32. For training, we used Binary Cross-Entropy Loss (BCELoss) as the loss function and the Adam optimizer to improve convergence.

2.3.2 Test process

Due to the limited size of the test dataset, we loaded the test data along with the training data and evaluated accuracy simultaneously. Model performance was primarily assessed based on classification accuracy.

3 Results

3.1 SimpleCNN

The validation and test results for SimpleCNN are presented in Table **??**
Overall, we observe that the classification performance of SimpleCNN is suboptimal, suggesting limitations in directly classifying perturbations from raw images without segmentation.

3.2 SegCNN

The validation and test results for SegCNN are presented in Table **??**
We can see the results are good, and improve a lot compared with SimpleCNN.

4 Conclusion

This study compared two CNN pipelines for classifying chemical perturbations in bio-images: SimpleCNN, which directly classified raw images, and SegCNN, which incorporated a segmentation step. Our findings show that SegCNN, by focusing on cell morphology, significantly improved classification accuracy over SimpleCNN. This highlights the value of segmentation in enhancing model performance for bio-imaging tasks. These results suggest that segmentation-based pipelines could be valuable in applications such as drug discovery and functional genomics, where accurate morphological analysis is essential.

5 Code Availability

The code used for this analysis is available on a public GitHub repository: `https://github.com/Zihan-Hazel/Cellimage_CNN_ADS`