Classification of Subcellular Protein Localizations

Pradeep Mylavarapu

https://youtu.be/AK_NtSG1HKc

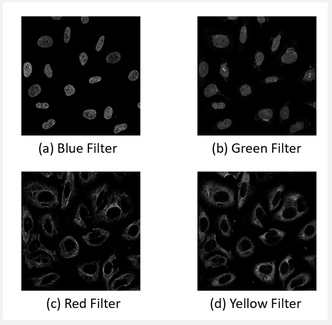
Introduction

Understanding how proteins are used in different types of cells in the human body, will help biomedical researchers better understand human cells & diseases.

- Developed a deeplearning based classifier to classify protein localizations from microscope images of different cells in the human body
- Inputs to the model were microscope images of various cells showing protein localizations in different parts of the cell
- Output for each example in dataset is a vector of size 28 that corresponds to 28 unique classes. Each class corresponds to an area within the cell where proteins are located

Dataset

Dataset obtained from the Kaggle Competition "Human Protein Atlas Image Classification", consists of 31,072 training examples. For each training example, the input is set of 4 greyscale images of size 512x512. The output is a vector of size 28 that corresponds to 28 unique classes. These classes indicate which part of the cell the proteins are localized in.



Features

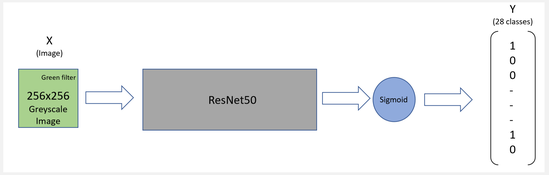
The input images are generated from post-processing microscope images using different filters. They are tagged as red/blue/green/yellow to indicate the filter used. The green filter images highlight the protein localizations while the images from other filters highlight different cell features. Since proteins are simultaneously used in multiple areas within the cell, each input can correspond to more than one output class.

Architecture

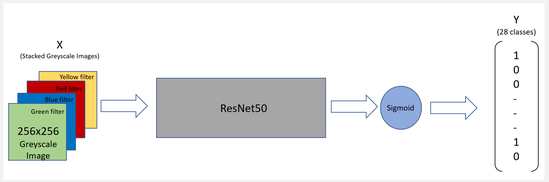
1. CNN Architecture: Resnet50 architecture was used with a sigmoid activation on output layer. This was later enhanced by adding additional CONV layer to Identity block.
2. Optimizer & Loss Function: Adam optimizer was used with Binary-crossentropy loss.
3. Train/Dev/Test Set split: 24000/3000/4072 examples

Models

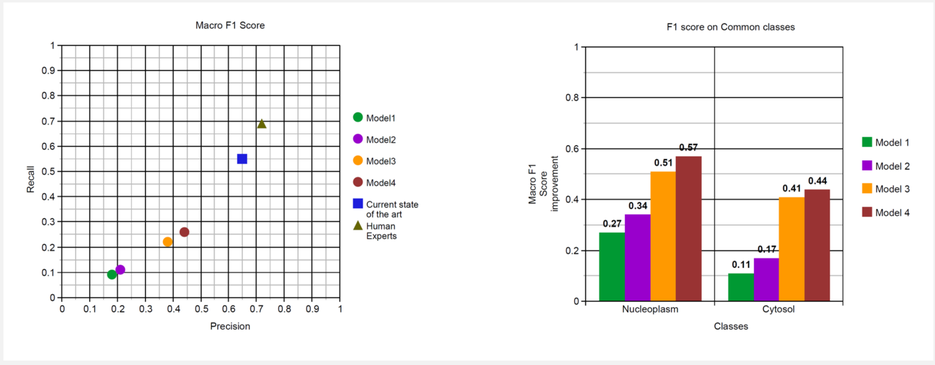
Initial Implementation I used Resnet50 implementation & used the green-filter images as inputs. The performance of this model on the test set produced a macro F1 score of 0.16.



Final Implementation After performing Error Analysis, I modified the architecture of the model to take a set of stacked images as input. For every training example, I stacked all 4 (green/blue/red/yellow filter) images and used them as input to the model. This improved the Macro F1 score to 0.32.



Results



Macro F1 score Improvements across 4 models. Model 1 used 128x128 Green-filter Images. Model 2 is the enhanced Resnet50 with 256x256 Green-filter Images as inputs. Model 3 used 3 Images (Green/Blue/Red Filters) stacked together. Model 4 used all 4 images (Green/Blue/Red/Yellow Filters) stacked together

Discussion

Moving from a single-image model to a stacked-image model resulted in a significant increase in terms of overall performance. Overall the models performed very well on the two common classes (Nucleoplasm & Cytosol) & struggled on rare classes that had very less data. Data augmentation improved the number of accurate predictions for some of the rare classes like Microtubule organizing center, Mitochondria. For some of the rare classes such as Endosomes, Lysosomes, Rods & rings, none of the models made accurate predictions on the test set. This was mainly due to lack of availability of data.

Future

Future steps to improve performance:

- Collect more data on the rare classes
- Use transfer learning by using a pretrained model as a starting point

References

[1] Wei Ouyang et al.
Analysis of the human protein atlas image classification competition.
Nature Methods, 16(10):1254–1261, 2019.

[2] Kaiming He et al.
Deep residual learning for image recognition.
2016 IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[3] BAOQI LI et al.
An improved resnet based on the adjustable shortcut connections.
2018 IEEE Access, pages 18967–18974, 2018.