Human Protein Atlas Image Classification

Amita Patil, Rudra S Bandhu (amita2, rsbandhu@stanford.edu)

Stanford University

Objectives

The goal of this project is to develop models capable of classifying mixed patterns of proteins across range of different human cells in microscope images. These models could be used for localizing proteins from high throughput microscope images.

Introduction

Proteins are large complex molecules that play a critical role in functioning of the human body. For better understanding of the complexity of the human cell, models need to classify mixed protein patterns from a range of different human cells. High resolution images of proteins in human cells are being generated at far greater pace than what can be manually evaluated [1]. Therefore, the need to automate image analysis to accelerate understanding of human cells and disease. In this study we used deep CNN architectures (VGG, ResNet, and DenseNet) to predict 28 different protein organelle localization labels for each sample.

Dataset: Features and Preprocessing

Dataset from Kaggle contains 31072 samples. Each sample (size: 512 X 512) consists of 4 image files corresponding to 4 color filters RGBY (Protein of interest (G), Nucleus (B), Microtubules (R), Endoplasmic reticulum (Y)). The label for each sample can have multiple labels from the 27 different cell types that could be present in the sample. Nucleoplasm (Class0) is present in 12885 while rods and rings (Class27) is present in only 11 samples. Classes 10, 15 and 17 are present with other classes only. 5% of both unique and multilabel samples along with at least 5 samples from each class (1576) were held out for validation. All samples were downsampled to 224 X 224. Different augmentation were used:

- **Augment 6:** 3 rotations (90, 180, 270) and 2 flips (H and V)
- **Augment 10:** 7 rotations (45:45:315) and 2 flips (H and V)

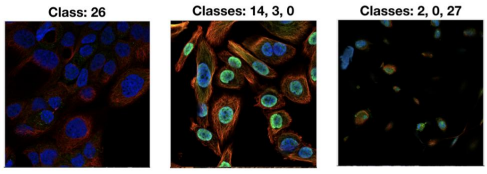


Figure 1: Representative Images with single label and multiple labels

Loss Function

BCELogitsloss functions with different weighting methods to address:

- Large number of negative labels (24 to 27 out of 28 labels are negative) in each sample
- Large imbalance of class sample count

$$\text{Loss} = -\frac{1}{N}\sum_{i=1}^{N} w_i[p_i y_{true,i}\log(y_{pred_i}) + (1 - y_{true,i})\log(1 - y_{pred_i})]$$

where $N = 28$ number of classes, w_i is the weight of class i and p_i is the weight for the positive label of class i

1. $w_i \propto 1/log(N_i)$, scaled to the class 27 (fewest labels)
2. $p_i \propto 1/N_i$
3. $p_i \propto 1/\sqrt{N_i}$

Models

Models trained (from simple to complex):

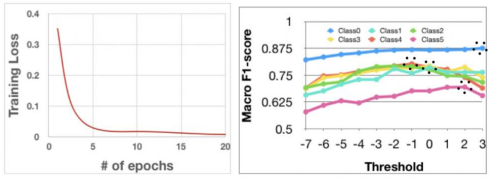
- Use features from pretrained deep CNN architectures (VGG19, Densenet161, Resnet18) and train the classifier layers only
- Use pretrained weights from Resnet18, Densenet161 and fine tune the entire network
- Tune from scratch Densenet73 architecture (3 dense blocks with 6,12,16 layers and initial feature size =64, growth rate = 12)

Results Summary Discussion

	Network	Densenet73	DenseNet161	DenseNet161	DenseNet161	DenseNet161 w/ dropout	DenseNet161	ResNet18
	Data Type	No data Augment	No data Augment	No data Augment	Augment6	Augment6	Augment10	Augment10
METRICS (Macro)	Loss Weight Type	1	2	3	3	3	3	3
Accuracy	Train	0.57	0.24	0.92	0.97	0.84	0.98	0.94
	Dev	0.36	0.19	0.43	0.49	0.46	0.56	0.47
Precision	Train	0.82	0.58	0.98	0.99	0.975	0.996	0.99
	Dev	0.62	0.53	0.68	0.78	0.73	0.83	0.76
Recall	Train	0.57	0.23	0.97	0.99	0.96	0.995	0.99
	Dev	0.38	0.5	0.62	0.7	0.73	0.73	0.69
F1 score	Train	0.65	0.31	0.98	0.99	0.967	0.995	0.99
	Dev	0.43	0.48	0.64	0.73	0.7	**0.76**	0.71

Figure 2: Performance Summary of key models

Trainging only classifier section of pretrained network achieves poor performance. Training and Validation performance improves with more complex network, fine tuning entire network and data augmentation. Best performing model: DenseNet161 with weighted loss type 3, per class thresholding and 10X data augmentation achieved a macro F1 score 0.76 and 56% accuracy on validation set. Overfitting is an issue across all trained models.



(a) (b)

Figure 3: (a): Training Loss vs. Epochs (b): Picking optimum threshold per class to maximize macro F1-score

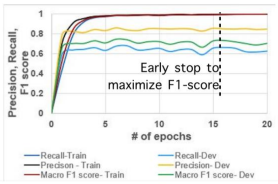


Figure 4: Macro Precision, Recall, F1-score for train/validation set vs. Epochs

Future Work

To address over-fitting issue and improve generalization capability of the models we suggest following approaches:

- Use the full resolution of the images (512 X 512) as input features to the network and augment dataset with rotations, flips and crops
- Train a light weight Densetnet type architecture
- Train separate networks with each network using features from one channel e.g. RRR or GGG.Use results from all 4 networks for final prediction.

References

[1] *https://www.kaggle.com/c/human-protein-atlas-image-classification.*

[2] Laurens van der Maaten Kilian Q. Weinberger Gao Huang, Zhuang Liu. Densely connected convolutional networks. *https://arxiv.org/*, 1608.06993, 201.

Link to youtube video — https://youtu.be/GFqlSV8YTMs

Acknowledgements

CS230 teaching staff, project sponsor David A Knowles.