CS755 final project: Human Protein Atlas Image Classification

Qian Zhang (PID:730186505), Peiran Guo (PID:730190851)

October 2018

1 Introduction

In this paper, we will discuss our algorithm for a kaggle competition, "Human Protein Atlas Image Classification" which is basically a multi-label classification task for 2D images [1]. There are 28 labels in total and each subject can have several labels at the same time. All subjects are equipped with four images: the protein of interest, the nucleus landmarks, the microtubules landmarks and the endoplasmic reticulum landmarks. There are 31072 subjects in the training set and 11702 subjects for testing.

The most challenging part of this competition is the data imbalance problem. As shown in Fig. 1, the label distribution is highly imbalanced. However, the evaluation method for this competition is Macro F-Score, which treat each label equally, so we have to make sure our algorithm works well for all labels. To address this issue, we divide our method into two components: 1). the DenseNet[2] based feature extracting component, and 2). the class specified refining component.

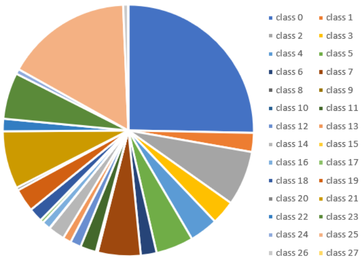
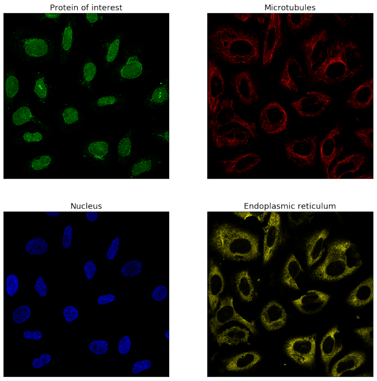


Figure 1: The label distribution


Protein of interest

Microtubules

Nucleus

Endoplasmic reticulum

Figure 2: Channels of a subject

2 Feature Extracting Component

Finetuning is a wilely used approach in computer vision task. However, the image for this task are very different from usual images. As shown in Fig. 2, the images for this task are given in four channels rather than the three channel images from ImageNet[3]. Thus, we have to use some modified methods for this task or just train the network from scratch with only the given images.

2.1 Drop one channel

To get a baseline method for this task, we just simply drop the yellow channel and test the finetuning method with a pretrained Densenet161 model. We also used some methods to maximize the performance of this model: 1). weighted loss(before applying sigmoid) 2). different linear layers as classifiers 3). use focal loss[4] for training. Fig. 3 shows the network architecture, the blocks in gray are using pretrained weights and the blocks in red are using randomized weights and are trained from scratch.

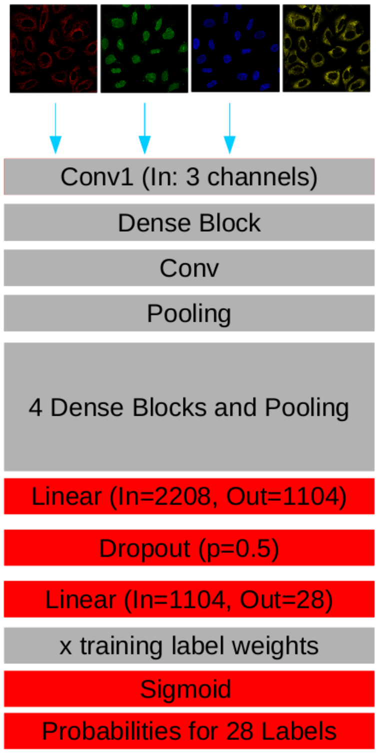


Figure 3: Original Densenet

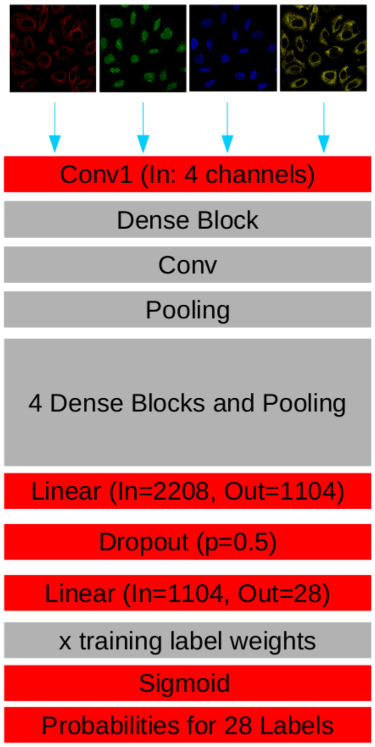


Figure 4: Modified Densenet

The best Macro F-Score we got using this strategy is 0.276.

2.2 Finetuning certain layers

To make full use of the 4 given channels, we modified the Densenet model by replacing the first convolution layer into a 4 channel layer. The architecture are shown in Fig. 4. However, we cannot rely on the outputs of the pretrained layers if we modified their base layer, so we trained this model in 2 steps: 1). we fix the weights of pretrained layers and train the first convolution layer and the classifier with a learning rate of 0.0003. 2). we release the weights of pretrained layers and train the whole model with a learning rate of 0.0001.

The best Macro F-Score we got using this strategy is 0.311, and we decide to use the features extracted by this method as the input for our refining component.

2.3 Each channel as a gray images

This is a ”crazy” model we tried. A common approach for applying pretrained models to gray scale images is to duplicate the image three times and feed them into the pretrained model. Inspired by this strategy, we treat each channel as a gray image, feed them into the pretrained model and only combine them together in the first linear layer.

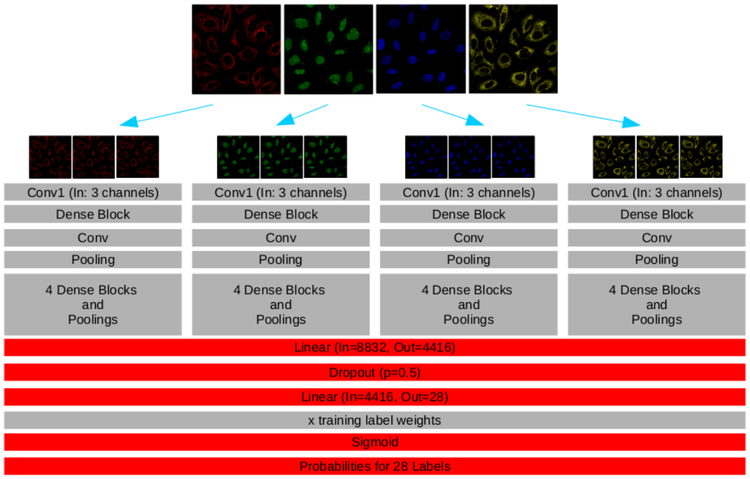


Figure 5: Each channel as a gray image

The best Macro F-Score we got using this strategy is 0.294, and it’s possible to further improve the accuracy by fineturning the whole model like we did in section 2.2. However, this model is four times bigger than original densenet and our GPU don’t have enough memory to train the whole model, so we don’t use features extracted by this model.

3 Refining Component

The F-Scores for each class using our feature extracting component are shown in table. 1. We can see that classes with very small number of positive samples have a low F-Score, which strongly eliminate our overall scorce, so we need to apply a specified refining method for each different class.

3.1 Dimensionality Reduction

Having feature of high dimensionality is problematic: a) multicollinearity of feature leads to weak generalization of model b) High dimension features are sparse and redundant. Using PCA[5] for most classes, and LDA[6] for classes with small number of positive samples due to high feature dimensionality and imbalance labels. We reduced the feature dimensionality to 15 which explained 90.22 percent variance in PCA. The features with high collinearity were excluded before using LDA.

3.2 Multi-label K Nearest Neighbor

The Multi-label K Nearest Neighbor[7] method was applied to the updated features for classification. The parameters with the minimum test error from 5-fold cross-validation(k=7, s=0.5) were used in this method.

3.3 Random Forest

The random forest method[8] was applied to the updated features for classification. We redefined the class weight by increasing the weights for the samples in the minority classes to lessen the impact of imbalance problem. The parameter with the minimum test error from 5-fold cross-validation(number of trees=43) was used in this method.

The Macro F-Score were 0.430 and 0.470 in knn and in random forest, respectively.

4 Result and disscussion

The macro F-score we get using our CNN classifier is 0.311. And we further improve this score to 0.470 using our refining component, and by mixing the two results, we achieve a final score of 0.488. The detailed score for each label are shown in Fig. 6. We can see that CNN model performs better when there are enough positive samples, and our random forest based refining component works better in classes with very small number of positive samples.

All our training and testing code are available on Github:
https://github.com/qzane/Human-Protein-Atlas-ImageClassification.

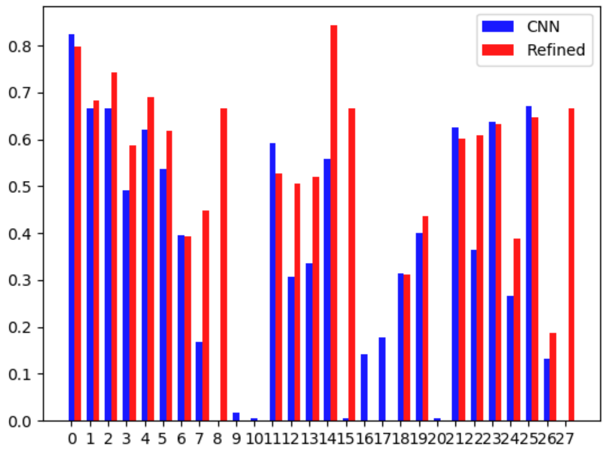


Figure 6: The f1 score of CNN model and refined model

References

[1] Sullivan et al. *Human Protein Atlas Image Classification.* Kaggle, 2018.

[2] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, volume 1, page 3, 2017.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009.

[4] Tsung-Yi Lin, Priyal Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[5] Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

[6] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[7] Min-Ling Zhang and Zhi-Hua Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern recognition*, 40(7):2038–2048, 2007.

[8] Andy Liaw, Matthew Wiener, et al. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.