DATS6203 – Machine Learning II, Fall 2020
Group 2 : Tran Hieu Le, Voratham Tiabrat, Aashish Nair
Individual Final Project Report : Voratham Tiabrat

Human Protein Atlas Image Classification

Introduction

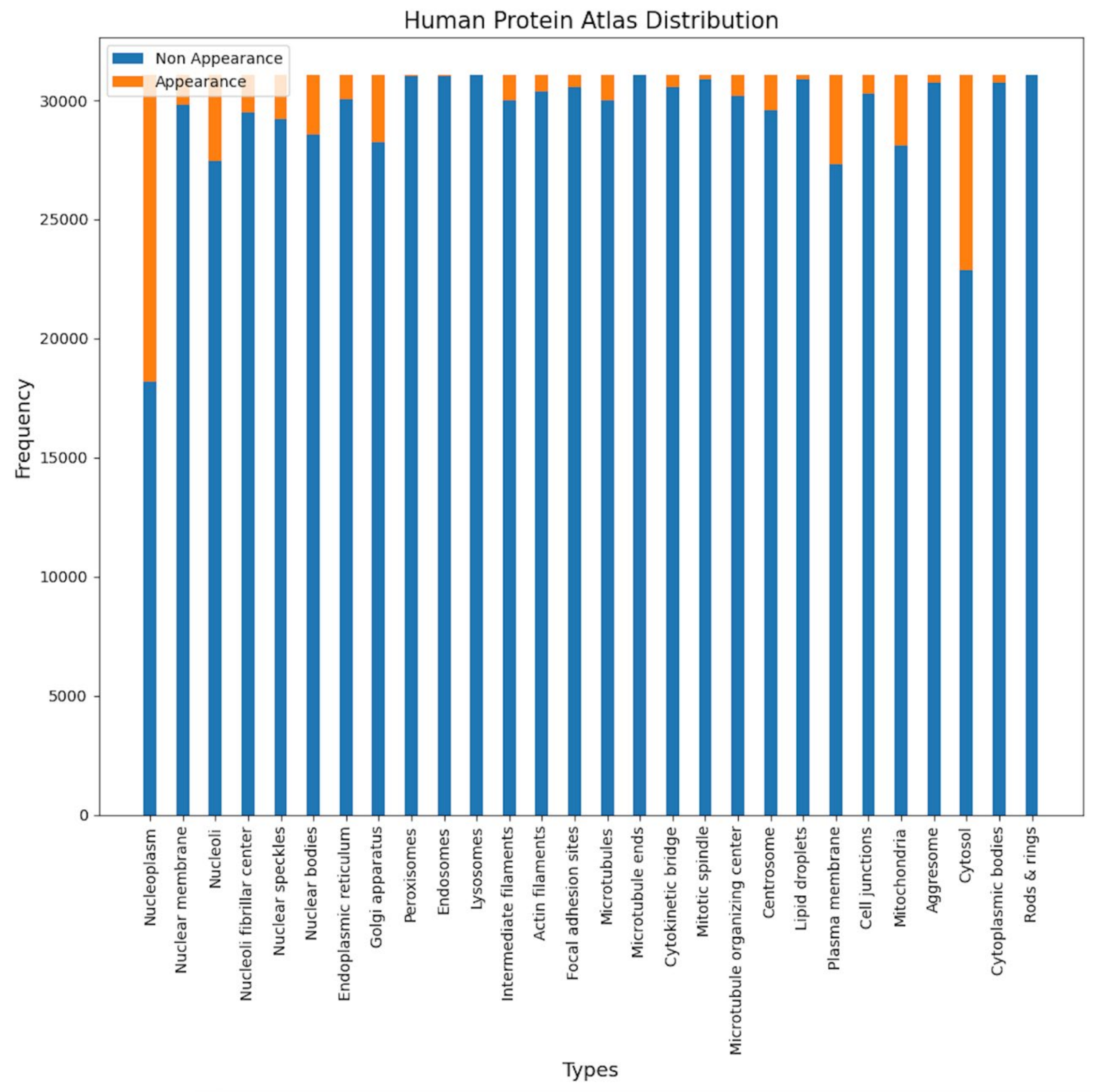
Proteins are the main component in the human body including cell. Also, they are the main key for executing function of human body. Classifying the proteins in cells would help to understand more insight of human cells which could be used to lead to the next breakthrough in effective medicine and treatment. Moreover, some disease could be diagnosed by observing the anomaly behavior or amount of protein in cell.

One of the main problem in the medical field is that manual classification of proteins has been limited to single patterns. The mixed patterns across different human cells should be able to be classified to fully understand the complexity of the cell or diagnose the disease. Therefore, automating biomedical image analysis using machine learning could significantly accelerate the efficiency in medical field one way or another.

Methodology

The data contain 4 channels of images consist of Red, Green, Blue and Yellow (RGBY) which each of the channel is in the Gray scale images. The step to load the images is to read gray scale of each channels and resize it to 128x128, then stack it together. For the targets of each image, the dataset is multi-labels data with 28 classes totally in this dataset. So I create the dictionary for all classes then read the target file(train.csv) to get all targets for each image.

In the project, I split the data in to 3 set which are train(70%), validation(15%) and test set(15%). I create the figure to observe the imbalance of data and see which class is the minority class. As the dataset is imbalance, I search for the data that contain the minority classes which consist of Peroxisomes, Endosomes, Lysosomes, Microtubule ends, Mitotic spindle, Lipid droplets and Rods & rings in training set as the figure of the class distribution in the original data show. Then I use oversampling along with data augmentation technique to make the data more balance. I used Keras to generate the data augmentation as I couldn’t figure how to use Pytorch to generate 4 channels images, so I used Keras instead, but Pytorch is used in modeling.



Pytorch is used to use for modeling purpose. Starting with the created CNN model, I used 4 2DConvolute layers starting with 16 channels then 32, 64 and 128 then respectively with Batch Normalization and Dropout. Then I used 3 linear layers to flatten the output from Convolute layers. The activation function is Relu and then using BCEWithLogitsLoss for loss function. I created the save point and early stop by F1-score macro and samples on validation set then final evalution the model on test set.

Results

Keras framework:

Model description	Validation set		Testing set	
	Macro F1	Sample F1	Macro F1	Sample F1
CNN Baseline model CNN (4 Conv2d input 4 to 64 with (3x3) kernels size and 3 Linear input 64 * 6 * 6 to 28)	0.1979	0.4040	0.1582	0.3122
Resnet50: Baseline model	0.2866	0.4971	0.2783	0.4924
Resnet50_2: Oversampling Augmentation	0.5305	0.5795	0.4894	0.5774
Resnet152: Oversampling Augmentation	0.5700	0.6020	0.5133	0.5982

I start by using my own CNN model which consist of 4 2Dconvolute layer input from 4 dimension to 64 dimension and then flatten it by 3 linear layers to final output of 28. This model takes a lot of epoch to run as it is non-pretrained model. The result is that both macro F1-score on validation and test set is below 0.2. Even I tried to change the hyperparameters, but it wasn't improved that much so I try to implement the pre-trained model instead. The first pre-trained model I used was Resnet 50, I have to change the first 2Dconvolute layer input to 4 dimension and last linear layer to 28 for our dataset. I also add and early stop by F1-score when the model wasn't improved on validation set. The result is the model is better than mine CNN model, but both scores still below 0.3 so I use this model as a base model.

Even Resnet50 performance is better than the basic CNN but the F1-score still consider quite low. The reason is that there are some classes that appear only around 10 times in dataset (minority class) and the model could not be able to predict it. There are 7 classes considered the minority class which consist of Peroxisomes, Endosomes, Lysosomes, Microtubule ends, Mitotic spindle, Lipid droplets and Rods & rings. Therefore, oversampling by data augmentation of the minority classes is applied to improve the model. After using the generated data and tuning the hyperparameters, the macro F1-score of Resnet50 model is significantly improve to 0.5 for validation set, but still less than 0.5 for testing set.

Then I move on to another pre-trained, Resnet152, to see whether it would perform better performance than the Resnet50 or not. As a result, the Resnet152 performance is slightly better than the Resnet50 and also F1-score for both validation and testing set is above 0.5.Even data augment technique would be help to improve the model, but it is important to note that the dataset is multi-labels so when we generate more data for minority class, it is likely that we also increase the number of majority class too.

Conclusion

The main objective of this project is to classify mixed patterns of proteins by handling 4 channel image from microscope and be able to identify the minority class of proteins in the image. The main challenge is the minority classes data (strong imbalance) which is a critical problem in classification. To deal with the imbalance data, oversampling by data augmentation on minority classes are employed to increase the minority class input, which results into a much better improvement. As I mainly use the code from exam in training and evaluating part and also used the pre-trained model part, I didn't exactly copy and paste from the internet, but read for the idea instead. The percentage of the code that I found or copied from the internet would be around 20 percent which is mainly on the data loader file.

For future improvement, due to the insufficient memory of GPU, I cannot use the highest resolution of the raw image, but have to resize it which could impact the performance of the model. Moreover, the minority class is too low even using data augmentation but it also increase the majority class as the dataset is multi-labels so if we have more data of the minority class from external dataset. This would significantly improve the model performance.

References

https://machinelearningmastery.com/image-augmentation-deep-learning-keras/

https://www.kaggle.com/allunia/protein-atlas-exploration-and-baseline

https://www.kaggle.com/grg121/protein-atlas-cnn-with-pytorch