Project Report

Zhang Rongqi A0276566M

1. Project Background

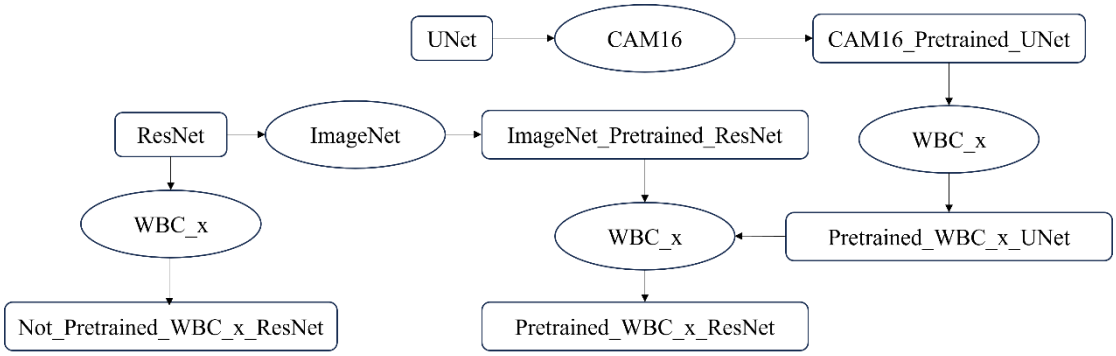
White blood cells are a vital part of the human immune system, classified into five types: eosinophils, basophils, lymphocytes, monocytes, and neutrophils. Different types have unique functions and reflect health. Count imbalances may signal medical conditions. Classification is critical for medical diagnosis and research. Traditional methods are time-consuming and subjective.

Advances in computer vision and machine learning enable automated classification, using additional data, e.g., segmentation masks, to improve accuracy. Three datasets (WBC, pRCC, CAM16) contain images of white blood cells, cancer cells, and lymph node metastases with partial masks. The project pretrains on pRCC and CAM16, fine-tunes on WBC, and compares performance with additional data.

This project is significant for white blood cell classification, benefiting medical diagnosis and research. Applications include medical image analysis, autonomous vehicles, satellite imagery, with deep learning enhancing classification. Challenges include limited morphological variation coverage and intraclass variations due to image capture and staining. Utilizing additional data, e.g., segmentation masks to customize loss function, is a key focus of in my implementation.

2. Method Description

My solution consists of two stages, using the UNet model to generate the corresponding mask based on the cell images in the WBC data set, and using the mask as an auxiliary image combined with the WBC data to train ResNet for classification tasks. The following image shows my model generation process.



In order to reproduce, please put the model files and codes in the same directory as the WBC_X folder, namely 'folder'/'xx_model.pth', 'folder'/'xx.ipynb', 'folder'/'WBC_x'. First run the Not_Pretrained_ResNet file, which will get four unpretrained ResNet_WBCx_model.pth for classification, then run pretrain_mask_generator, which will get the pretrained Unet model

UNet_CAM16_model.pth, then run mask_generator.ipynb, and modify the proportion in the code respectively For 1, 10, 50, 100, run four times, which will get four CustomLoss_UNet_WBCx_model.pth generated mask models, and finally run Pretrained_ResNet, which will get four final classification models Pretrained_ResNet34_WBCx_model.pth.

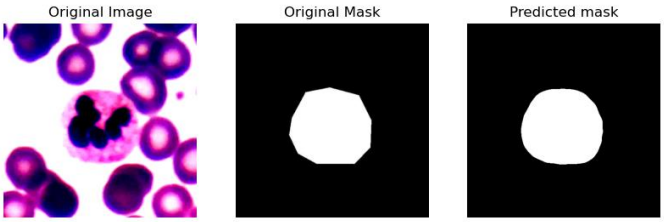
The models perform the classification task are four ResNet_WBCx_model.pth models, and four Pretrained_ResNet34_WBCx_model.pth models.

2.1 Mask generation

For the 10% mask data given, my first thought was to reproduce as much mask data as possible, so I considered automating the manual mask drawing process with a neural network. Soon, by looking up relevant information, I found a suitable model for this task.

I used the ResNet34-based UNet model to generate the masks in the WBC dataset. UNet, as a variant of the autoencoder, can be trained to output Mask images. Furthermore, the advantage of choosing this model is that UNet's network structure allows effective feature extraction and context information transfer. Through the structure of encoder and decoder, UNet can capture features at different scales, from lower-level image features to higher-level semantic information. This is important for medical image segmentation tasks, as they often require considering structures and features at different scales. In addition, medical image segmentation tasks often face small sample problems, and UNet performs well in this regard. This is extremely valuable in the medical field, as obtaining medical image labeling often requires expertise and time. On the other hand, medical images often contain complex structures and boundaries, and UNet's decoder can effectively capture the fine boundaries of objects.

In this task, I used the binary cross-entropy function as the loss function. After training, this model reached a pixel prediction accuracy of 96% for the Mask image in the WBC_10 training set. We cannot achieve 100% accuracy here because the masks given in the WBC training set are not accurate enough and they have very rough boundaries. If we achieve 100% accuracy, it means that we have overfitted. An example is as follows:



We noticed that in the WBC image, the identified cells are all concentrated in the center of the image. Therefore, we hope that the constructed mask can also be presented in the center of the image. In this regard, I modified the binary cross entropy. The main idea is to adjust the loss function according to the distance of each pixel in the image to the center of the image, so that pixels close to the center of the image contribute more to the loss, and pixels far from the center contribute less to the loss. The purpose of this design is to help the model better capture the outline of the target object, thereby improving the accuracy of segmentation. Specifically, the code calculates the L2 distance of each pixel to the center of the image, and then uses these distances as weights to adjust the binary cross-entropy loss. After applying the above Loss function, the prediction error rate on the WBC_1 data set dropped from 4.29% to 3.77%.

2.1.1 Pretraining

I trained UNet with mask on the CAM16 dataset. And based on the pretrained model, continue to train on the mask in the WBC training set. Because the amount of data in this part is too small, I did not divide the validation set and used all the data for training. At this point, we have obtained mask boundary information with high accuracy for each image. The accuracy is shown in the table below. Next, consider how to use the mask.

Dataset	Train Error Rate
WBC_1	3.77%
WBC_10	5.32%
WBC_50	5.69%
WBC_100	5.87%

2.2 Cell identification

I used ResNet34 as the image classification model. ResNet-34 is a variant of deep residual networks with deeper layers than traditional convolutional neural networks. In medical image classification tasks, deep models can better capture and learn complex features, helping to improve performance. ResNet-34 shares the parameters of the convolutional layers, allowing the model to effectively learn features at different scales and locations. In medical image classification, this is important for detecting and identifying lesions or structures, as they may appear in different sizes and locations. In addition, residual connections allow information to be transferred from one layer to distant layers, avoiding the vanishing gradient problem. This helps the model train and optimize better, especially in deep networks, which is important for complex tasks in medical image classification.

2.2.1 Pretraining

First, I called the ResNet model pretrained on ImageNet, which has sufficient cognitive capabilities for image elements (such as line segments, boundaries, etc.), but this is not enough because the images on ImageNet are not medical images. Further pretraining is required, First, I tried to use the model with all parameters unlocked and pretrained. The CAM16 data set reached an accuracy of 83%. The training process is shown in the figure.

I think this is crucial for a small training set (such as WBC_1), which can greatly improve its prediction accuracy. **However, this model only had a very low accuracy of 60% in the WBC data set. I don't know if this was the reason for the learning rate setting.** In the end, I did not use the model pretrained on CAM16.

2.2.2 Data balance

Due to the uneven distribution among data labels in the training set. This easily leads to the model being more likely to memorize the largest type of cells and ignore some of the smaller cell types because their impact in the Loss function is not obvious. Therefore, my approach is that uses the cross-entropy loss function. For each type of cell, cells with fewer samples have a higher weight (specifically, set to the reciprocal of the number of images of this type of cell). After modification, the probability of misidentification of each category is more balanced (the standard deviation is smaller), but we also notice that when we pay more attention to the categories of a few cells, in order to reduce their recognition error rate, we will increase other identification error rates for most cell classes. The final weighted average values more the error rate of the majority of cells, which leads to the application of data balance as a whole, which may lead to a decrease in recognition effect, that is, fairness is not free.

Error rate	Basophil	Eosinophil	Lymphocyte	Monocyte	Neutrophil	Weighted Mean	Standard Deviation
Before	0%	15.08%	1.46%	12.63%	0.94%	2.72%	6.45%
After	0%	10.32%	4.61%	8.42%	2.36%	3.76%	3.79%

2.2.3 Data augmentation

As a way to enhance the generalization ability of the model, when applying data augmentation, we also need to consider whether the enhanced data will affect the classification judgment of human experts. In this task, I used random cropping, random vertical flipping, random horizontal flipping, and color enhancement on the training set. Applying data augmentation makes it more difficult for the model to classify samples. Experimental results also confirm this phenomenon. When the accuracy of the verification set reaches more than 95%, the accuracy of the training set remains around 90%. This reflects that the model has certain generalization capabilities.

2.2.4 Hyperparameter settings

We can use smaller epochs for a larger training set, and we can treat the additional data as additional enhanced data. Therefore, multiplying the epoch times the data set size equals a constant, which roughly achieves the same effect. I used lr_scheduler. Since the value of epoch has been confirmed, I set step=epoch/7, gamma=0.5. If the initial learning rate is 3e-3, the learning rate will gradually decrease during the training process, and eventually shrink to 2e- 5. Due to the large size of the images given in the data set, I uniformly reduced them to the size of 224*224 to reduce computational complexity and speed up training.

2.2.5 Some attempts at mask usage

First, I tried the simplest way: passing the mask as the fourth channel input. However, the effect of this method is not good. When training the WBC_50 data set on ResNet that has not been pretrained, using the above method reduces the accuracy of the original 83% verification set by 80%. My guess about this is that although we give all the information to the model and let it learn on its own. However, this method will confuse the model, because the mask is a very special kind of data. If it is input into the channel normally, the model cannot make good use of this data.

Another way is to use a mask to weight the loss function, so that the loss can be made as small as possible through the training algorithm, so that the model can learn to determine the cell area. Specifically, multiplying the normally calculated loss function by $(\alpha_1 * mask + \alpha_2)$ is equivalent to $\big((\alpha_1 + \alpha_2) * mask + \alpha_2 * nonmask\big)$, that is, weight the inner and outer areas of the mask respectively. I tested some α combinations and the results are as follows.

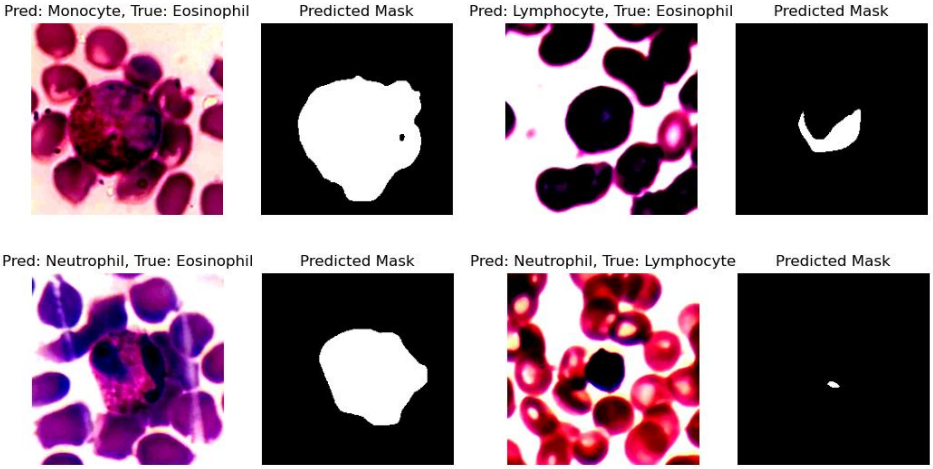
	α_1	α_2	Validation Accuracy			
			WBC_100	WBC_50	WBC_10	WBC_1
Pretrained	0	1	98.03%	98.20%	96.29%	92.07%
	1	0	96.41%	96.75%	96.29%	92.76%
	-1	1	98.37%	98.14%	97.10%	92.48%
Not Pretrained			86.40%	87.50%	86.40%	86.57%

$(\alpha_1, \alpha_2) = (0,1)$ means that we do not use mask to modify the loss function, $(\alpha_1, \alpha_2) = (1,0)$ means that we only want Loss to calculate the area inside the cell , $(\alpha_1, \alpha_2) = (-1,1)$ means that we only want Loss to calculate the area outside the cell. Each of them performed better than the untrained model. In my implementation, I chose $(-1,1)$, which is quiet counterintuitive that only focus on the environment will lead to a better performance. But at least, we can say that the additional information is useful.

3. Conclusions and findings

- By weighting the Loss function, the model can focus more on a certain area and achieve better performance overall. Using ImageNet pretrained image classification models can improve a lot of prediction accuracy.
- Without pretraining, as the training set increases, the model prediction accuracy will gradually increase. In the case of pretraining, the increment will decrease, which is because WBC accounts for a smaller proportion of the data used for model training.
- Applying too strong data augmentation will often result in the accuracy of the validation set being higher than the accuracy of the training set, because this makes model training more difficult and enhances generalization.
- Since the WBC data set is unbalanced, training on categories with less data is prone to overfitting and lower prediction accuracy. By modifying the loss function weight for each category, the data can be made more balanced, which embodies the idea of fairness, but this fairness is not without cost.
- Instead of inputting all the information into the model and letting the model fit a very complex function, the effect is not as good as our manual intervention in the training process of the model and the effect of letting the model perform specified behaviors.

- Although using mask to modify the weight of the loss function can help the model better focus on the cells themselves, this method will also affect the upper limit of the accuracy of the classification task: the accuracy of the final classification task is affected by this Mask generation model. The following images are some misclassifications cells with their generated masks, where the generated masks have more severe errors. We can see that providing the mask incorrectly affects the judgment of the classification model. Additional information is helpful for training, but the quality of the additional information can also seriously affect the training effect.



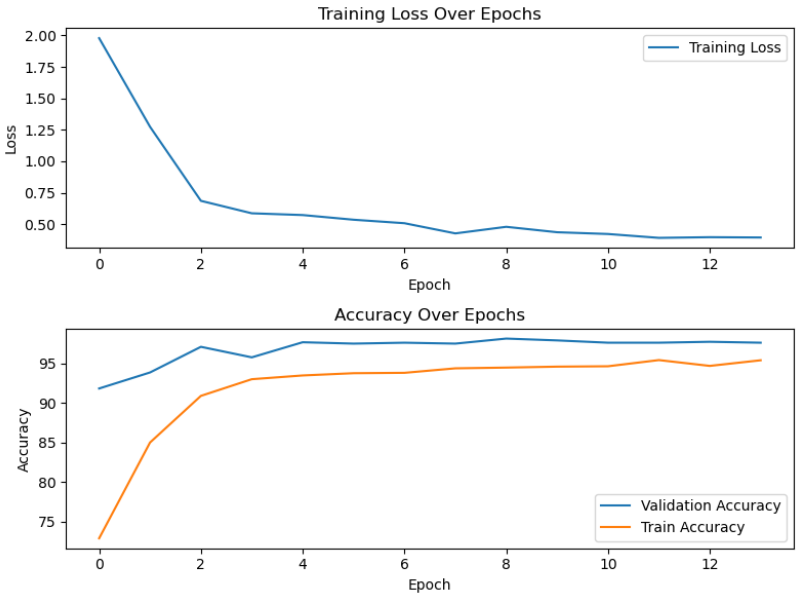
4. Improvements and shortcomings

Due to time conflicts with several projects in other courses, I only had six days (18 Oct.- 23 Oct.) to complete this project. The design only took me two days, but due to the need for adjustments There are too many parameters, and after permuting and combining them, I spent a lot of time on training to test my ideas. There is still much room for improvement in my solution, but I don't have time to design and implement it, such as exploring more possible ways to use masks, using attention models that are more targeted at such tasks, such as Transformer, and using the pRCC data set to pretrain (for example: cut the data in the pRCC data set into several small images, and use contrastive learning to self-supervised learning).

Furthermore, I didn't realize until the end of the experiment that for the same method and parameters, I needed to experiment with multiple different random number seeds and measure their mean and variance. However, due to time constraints, training a single model often takes more than half an hour, and I was unable to conduct further experiments.

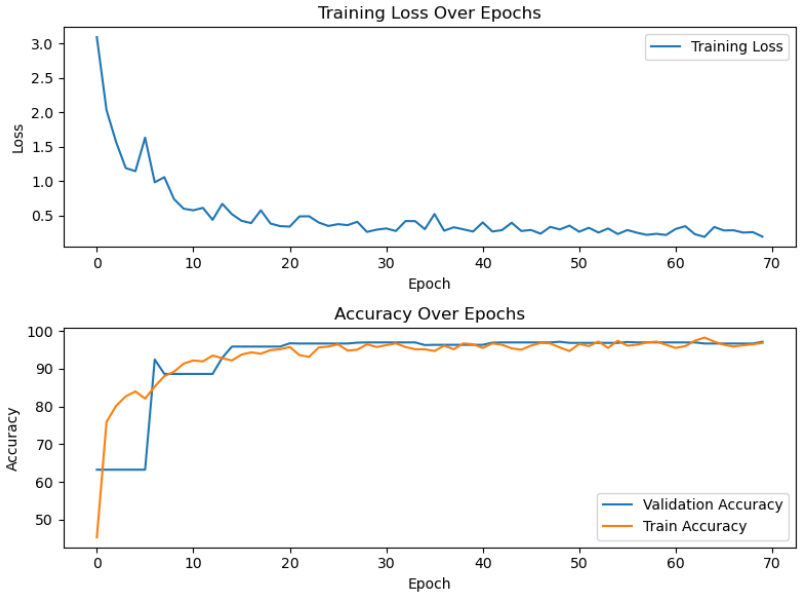
Appendix A Train process of pretrained ResNet model with $(\alpha_1, \alpha_2) = (-1,1)$

WBC_100

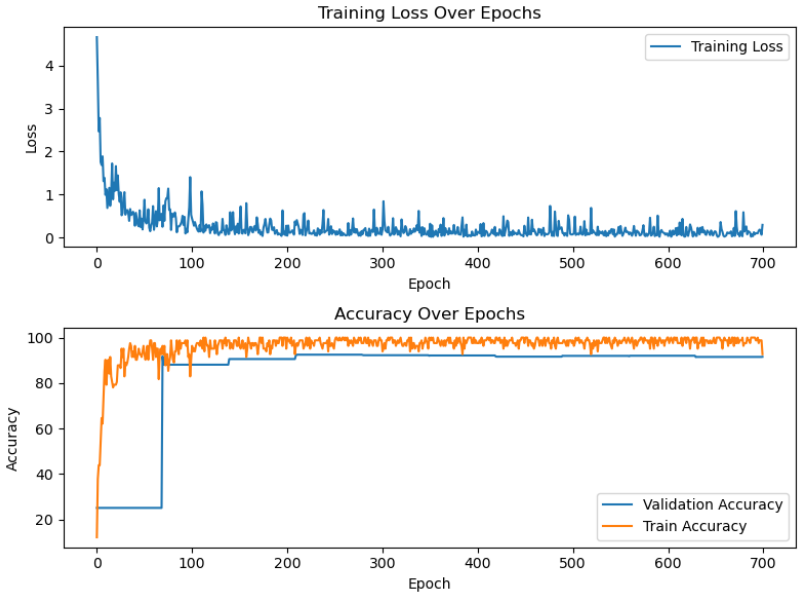


WBC_50

WBC_10


Training Loss Over Epochs
Training Loss
Loss
Epoch
Accuracy Over Epochs
Accuracy
Validation Accuracy
Train Accuracy
Epoch

WBC_1


Training Loss Over Epochs
Training Loss
Loss
Epoch
Accuracy Over Epochs
Accuracy
Validation Accuracy
Train Accuracy
Epoch