Hematologic Cell Identification - CS5242 Project

Name: Parashara Ramesh	**Student Id**: A0285647M	**NUSNETId**: E1216292	**Github Repo**: Link

I.Background

I.a Introduction

In the realm of medical diagnostics, the accurate classification of hematologic cells is a fundamental yet challenging task. This project aims to leverage computer vision-machine learning models to address the complexities associated with distinguishing five key white blood cell types: basophil, eosinophil, lymphocyte, monocyte, and neutrophil. While the results achieved in this project may not be groundbreaking or immediately deployable in a clinical setting, they serve as a valuable exercise in applying machine learning to a real-world medical scenario. By automating the classification of blood cells through image analysis, this work offers insights into the potential benefits of machine learning for medical professionals, showcasing its capacity to aid in the diagnostic process and improve precision. This project represents a modest step towards exploring the role of machine learning in hematologic cell identification, with room for future development and refinement in the medical field.

I.b Datasets

WBC Dataset: This dataset comprises microscopic images of different white blood cell types, including basophils, eosinophils, lymphocytes, monocytes, and neutrophils. While only a small subset of images is accompanied by masks, they provide valuable additional information for our classification task.

pRCC Dataset: The pRCC dataset is an unlabelled collection of medical images related to primary Renal Cell Carcinoma (pRCC), a type of kidney cancer, offering insights into kidney cancer diagnostics.

Camelyon16 Dataset: The Camelyon16 dataset primarily consists of histopathological images used in breast cancer diagnosis, and only a limited number of images are accompanied by masks, which can enhance the image analysis process.

I.c Experiments conducted

1. **Base WBC Model Development**: During this phase, a foundational deep convolutional classification model was established, serving as a baseline for the primary WBC dataset.
2. **Autoencoder for pRCC Dataset**: During this phase, a deep convolutional autoencoder was applied to extract insightful latent features from the unlabelled pRCC dataset. These learned features from the other models contributed to improved hematologic cell identification during the pre-training phase for the combined model.
3. **Camelyon16 Classification Model**: During this phase, a classification model was created utilizing the ResNet18 architecture for the Camelyon16 dataset, primarily focused on distinguishing between tumor and normal samples. Penultimate layer features were extracted for incorporation into the broader WBC model.
4. **Integration of Models**: In this phase, the weights of the pRCC and Camelyon16 models were frozen, and the same WBC model architecture from the base model was retrained from scratch.The WBC images were passed through both the Camelyon16 and pRCC models, facilitating the extraction of learnt features from the other models, contributing to improved hematologic cell identification.

I.d Impact

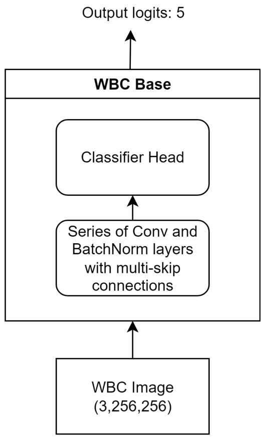
This project showcases the practical use of machine learning for more accurate medical image diagnosis. While not groundbreaking, it offers a tangible example of how technology can improve the accuracy of hematologic cell identification, contributing to the broader goal of enhancing medical image analysis.

II.Description

II.a Common Implementation Details

The codebase available on GitHub [link], draws inspiration from PyTorch Lightning hooks. This is particularly noticeable in the base trainer module [link] and demonstrates an organized and modular approach to model training. Key components of the implementation include the utilization of the weight decay Adam optimizer, Grad clipping to stabilize training, and the integration of the OneCycle learning rate scheduler. Additionally, the use of TQDM bars in the code enhances the visualization of epoch statistics, providing valuable insights into the training process.
Since the WBC and Cam16 datasets have **mask information** as well, data preprocessing was done to apply these masks[link] to the images to add it back into the dataset as new training data points thereby incorporating those mask details into the training loop as well.
While the WBC dataset exhibited class imbalance, the dataset balancing script involving augmentation [link] had to be omitted in accordance with explicit instructions from the TAs, who advised against balancing this specific dataset. Therefore all the WBC models are trained on the imbalanced datasets only.

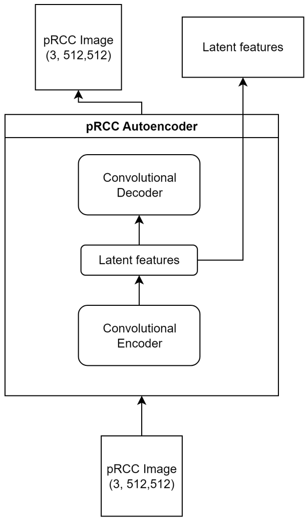
II.a Base WBC Classifier



Output logits: 5
WBC Base
Classifier Head
Series of Conv and BatchNorm layers with multi-skip connections
WBC Image (3,256,256)

This is a rough diagram of the model architecture, for the full architecture with 175k params, refer to this [link].
Since this is a classification task, the WBC input image is resized to (3,256,256) and then trained using cross-entropy loss.
The model itself is composed of a series of conv2d and batch norm layers along with skip-connections & multi-skip connections inspired by Resnet-9 architecture.
This model was then trained on each of the WBC datasets (WBC-1, WBC-10, WBC-50, WBC-100) for 10 epochs each.

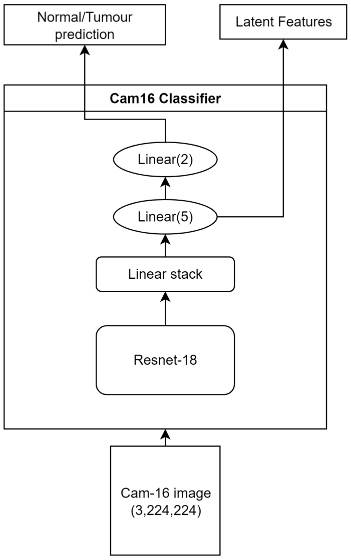
II.b pRCC autoencoder



This is a rough diagram of the model architecture**,**for the full architecture with 135k params, refer to this [link].
Since this dataset has no labels, it made sense to train an autoencoder after resizing the original image to dimensions (3,512,512). The model itself is a deep convolutional network where the encoder consists of conv2d and maxpool2d operations while the decoder consists of convtranspose operations.
The dimension of the latent features is of shape (128,64,64) which is used in the combined-pretrained WBC model mentioned later. The training dataset size was increased by adding augmented images obtained as a result of random rotation and flips which helped the model to learn from more data.
Instead of using regular MSE loss, SSIM loss is used instead as 'Structural Similarity Index loss'(SSIM), measures image quality by considering structural information, making it more effective than Mean Squared Error (MSE) loss for autoencoders as it captures perceptual differences and enhances visual fidelity.
This model was trained on the entire dataset for 20 epochs and resulted in decent reconstructions of the original image.

II.c Cam16 classifier

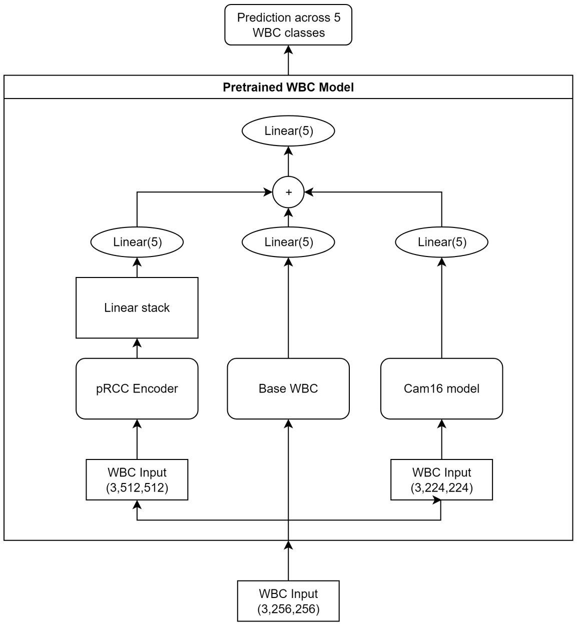
This model builds on top of the pretrained Resnet-18 weights (**note**: resnet18 is frozen for training here) and performs binary classification on the last layer and it is trained using Cross entropy loss. However we also return the features from the penultimate layer which has the shape of (5) so as to directly use it in the combined model during inference.



Even though this model is only trained to perform binary classification on normal/tumor images, the features learnt by the model proved to be effective in helping the combined-pre trained wbc model perform classification better than the base model as we shall see in the later sections.

Even in this model, the training data size was considerably increased using the masks provided along with random rotations and flips added as new augmented data points thereby enabling the model to learn from more data.

II.d **Combined Pretrained WBC Classifier**



This model combines all the models trained before by freezing the weights of the pRCC autoencoder and the Cam16 classifier while still retaining the same architecture used in the base WBC model.

The encoder of the pRCC autoencoder returns latent features of shape (128,64,64) it is passed through a linear stack to get a tensor of shape 5. Similarly the penultimate layer of the Cam16 model gives us another tensor of shape 5. Since the WBC base model also returns a tensor of shape 5 we can combine all the output vectors at the last layer to form another final tensor of shape 5 using which we can train the classification model using the Cross entropy loss.

Since each model accepts inputs of different dimensions, the same WBC image is reshaped in 3 different ways to feed into the different models using a custom dataset.

This model was then trained on each of the WBC datasets (WBC-1, WBC-10, WBC-50, WBC-100) for 14 epochs. Each model here showed some improvement over the base model which was trained suggesting that pretraining does indeed help the model to learn and converge better.

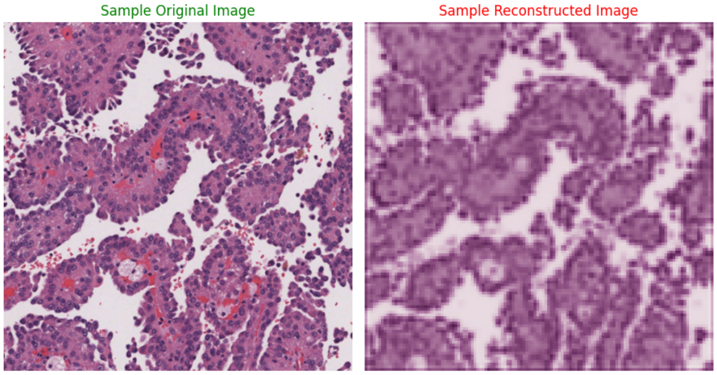
III.Results

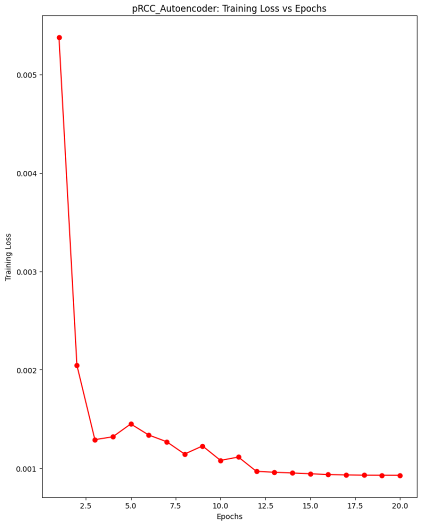
III.a **Final Test Metrics**

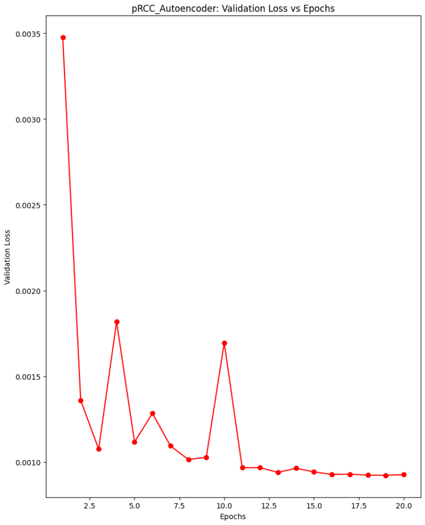
Model	**Test Loss**	**Test Accuracy**
pRCC Autoencoder	9.3208e-4	NA
WBC-1	0.8754	59.5%
WBC-10	0.38016	87.025%
WBC-50	0.25016	93.14%
WBC-100	0.24113	92.56%

Model	**Test Loss**	**Test Accuracy**
Cam16 classifier	0.33018	87.5%
WBC-1+	0.64201	80%
WBC-10+	0.41802	90.578%
WBC-50+	0.2717	94.95%
WBC-100+	0.2145	94.79%

III.b **pRCC Model Evaluation**

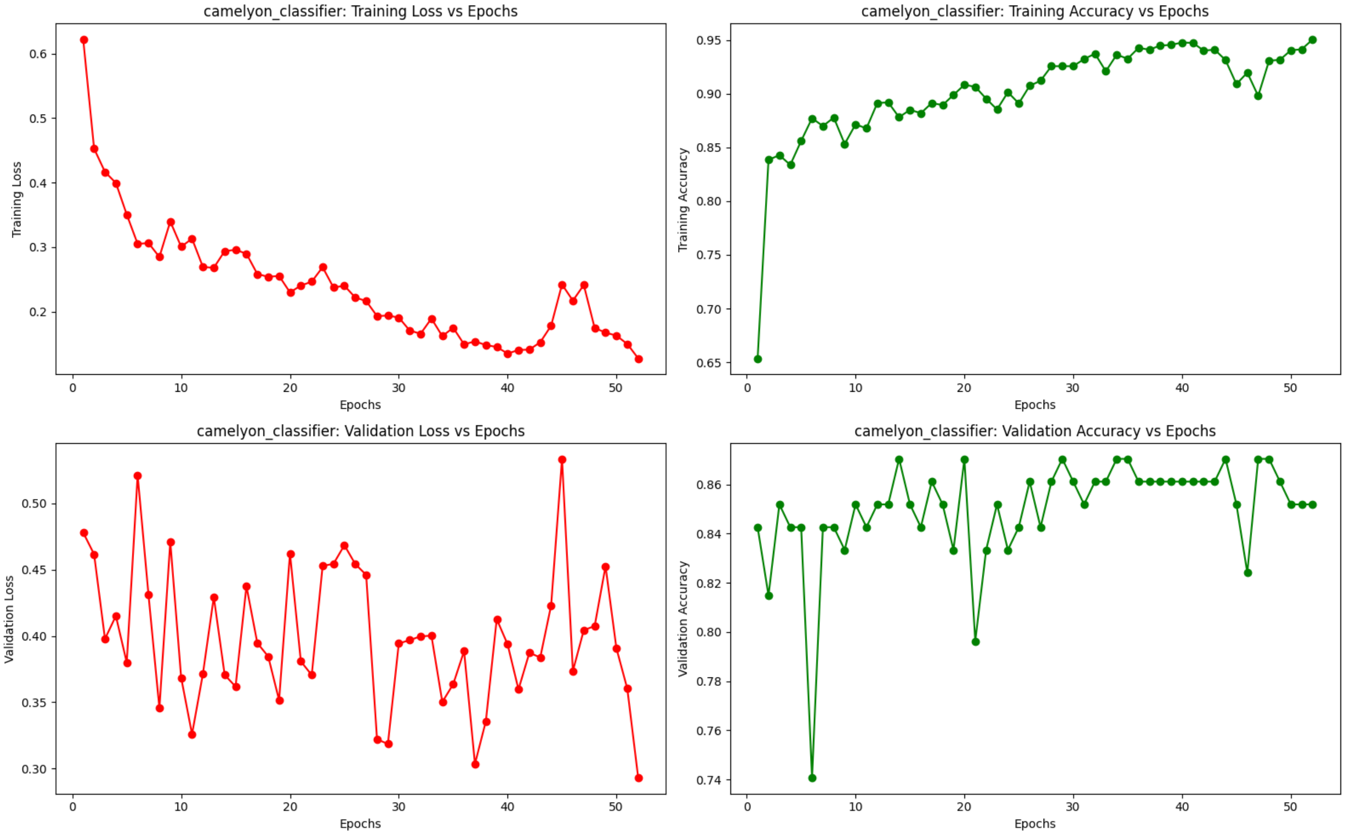






As we can see the pRCC autoencoder reconstruction is happening decently and the plots on the right show how the SSIM loss decreases over epochs for both the training and validation respectively. The depth of the model along with SSIM loss is the main reason for the good reconstructions as my initial attempts in building an autoencoder gave poor results overall.

III.b **Camelyon-16 Model Evaluation**



Since this model was built on top of Resnet-18 , the number of epochs needed to get to a decent accuracy was pretty high as the Resnet-18 model weights were frozen. When attempting to train it end to end without freezing the accuracy and loss worsened epoch after epoch as the Resnet-18 essentially has to 'unlearn' the features it learnt from ImageNet dataset which is why my architecture has a few linear layers on top of the Resnet-18 to learn only the bare minimum features in order to classify the cancer cell correctly.

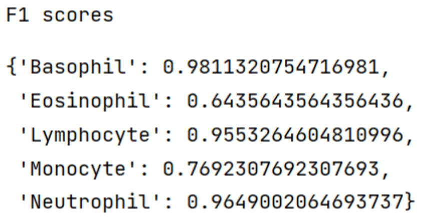
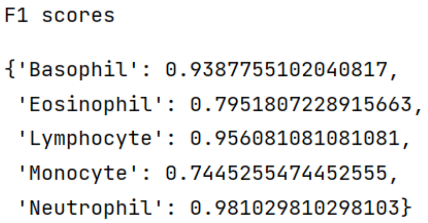
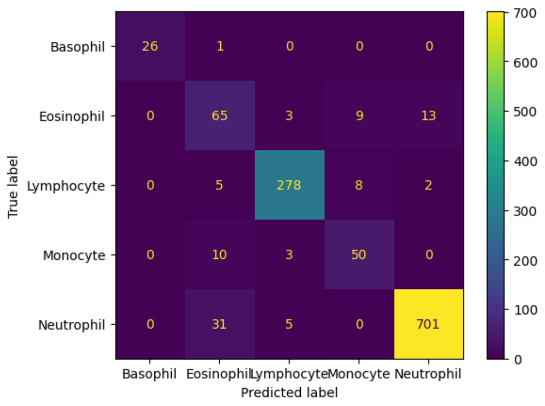
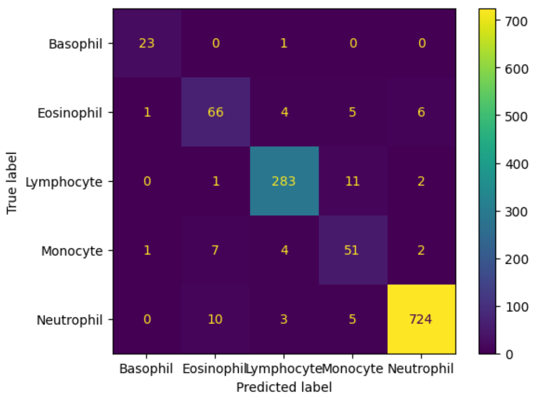
From the accuracy graphs on the right we can see that the training accuracy reaches a high of ~95% while the validation accuracy reaches a high of ~86% which certainly makes it a more than decent model which can be used for pretraining the final combined WBC model.

III.c **WBC & Pretrained WBC Test metric results**

Note: each cell is a link to my github repository as it is easier to show the entire image there. WBC_n+ refers to the pretrained model

Model	**Training Graphs**	**Precision**	**Recall**	**F1 scores**	**Confusion Matrix**
WBC-1	link	link	link	link	link
WBC-1+	link	link	link	link	link
WBC-10	link	link	link	link	link
WBC-10+	link	link	link	link	link
WBC-50	link	link	link	link	link
WBC-50+	link	link	link	link	link
WBC-100	link	link	link	link	link
WBC-100+	link	link	link	link	link

III.d **Comparing F1 scores and Confusion matrix only for WBC-100 (Base model vs Pretrained Model)**

	WBC-100	**WBC-100+**
F1-scores	 F1 scores {'Basophil': 0.9811320754716981, 'Eosinophil': 0.6435643564356436, 'Lymphocyte': 0.9553264604810996, 'Monocyte': 0.7692307692307693, 'Neutrophil': 0.9649002064693737}	 F1 scores {'Basophil': 0.9387755102040817, 'Eosinophil': 0.7951807228915663, 'Lymphocyte': 0.956081081081081, 'Monocyte': 0.7445255474452555, 'Neutrophil': 0.981029810298103}
Confusion Matrix		

III.e **Interpretation of results (WBC vs WBC+)**

- **Training & Validation Graphs:** (refer to the images in the table of links)
  - *Training Loss:* The training loss decreases more quickly in the pretrained model due to the additional insights from the other models. However, in the WBC-1 split the training loss is more erratic for the pretrained model as there is not enough data for the model to learn insights that quickly. In all splits however, the final training loss is much lower in the pretrained model as opposed to the base model.
  - *Training Accuracy:* The training accuracy increases more quickly in the pretrained model when compared with the base model and also is able to reach a really high value in all splits (~100% accuracy).

- *Validation Loss:* The loss curve in both models across all splits are very erratic, however it is not as erratic in the pretrained model, indicating that the pretrained models are able to perform much better on unseen data with the additional insights gained from the Cam16 and the pRCC model.
- *Validation Accuracy:* The pretrained models across all splits consistently perform better than the base models, which indicates that the pretrained models are able to perform much better on unseen data with the additional insights gained from the Cam16 and the pRCC model.

- **F1 Scores:** Due to the class imbalance in the training datasets, for the WBC-1 and WBC-10 splits there were few classes which had an F1 score of 0. This pattern of lower f1 score also shows up for the WBC-50 and WBC-100 splits as well which is purely only because of the class imbalance. The key takeaway point however is that the pretrained models all have f1 scores consistently better than the class wise f1 scores of its corresponding base model (despite the class imbalance) which seems to suggest that the pretrained models are better at classification prediction than their base model counterparts.
- **Confusion Matrix:** We can see that the pretrained models are able to make more predictions across varied classes when compared with their base model counterparts where many of the cells have a value of 0. Additionally, we can see that as the training dataset split size increases there are more predictions across all classes as there are more examples per class during the training phase. In a nutshell, the training dataset imbalance is clearly reflected in these predictions but in all cases the pretrained model performs better.

IV.Conclusion

From the results, it can be clearly seen that the pretrained model clearly performs better in classifying WBC cells when compared with the base model. Even though the WBC model architecture has not changed in both approaches, the differentiating factor is that the generic insights learnt by both the pRCC model and the Cam16 model aid the WBC model in learning the underlying latent representations much better and in some cases much faster as well. The pRCC autoencoder model's encoder is able to capture the most minimal latent representations needed for any medical image in order to reconstruct it while the Cam16 model is able to capture the latent representations needed to perform discriminative tasks both of which certainly aid the WBC model to learn much better. In one of my experiments, when I experimented by not freezing the pRCC model and the Cam16 model the accuracy and loss kept worsening for some time. This is because training it end to end forces the other two learnt models to unlearn most of what it knows to better optimize itself for predicting WBC cells. However, by freezing those two models we are ensuring that the WBC model is guided along a path where the other two models do not need to unlearn or relearn anything new specific to the WBC classification problem.

In order to further improve the accuracy of this problem here are some avenues which can be further explored as future work:

1. Balancing the WBC dataset to ensure that all classes have an equal representation using data augmentation. (this approach was discarded based on the suggestions of the TA's in the discussion forum)
2. Using SOTA model architectures which are known to work extremely well with medical datasets (e.g. UNets) (**Note**: this was another approach I tried as well but since I was not able to train those models in a stable manner because of which I had to resort to more simpler models)
3. Using pretrained medical vision models (e.g. DenseNet) instead of typical computer vision models like (Resnet , VGG etc). This is mainly because the preexisting computer vision models are all trained on a generic ImageNet dataset which has nothing to do with medical images in general. Therefore using a pretrained computer vision model which has already established a high level of accuracy in medical images would be an ideal starting point for performing transfer learning.
4. Finding/ Training more medical ML tasks while also training WBC classification by framing it as a multi-task problem thereby enabling the WBC classification model to gain insights from a wide variety of medical models.

In summary, this project showcases the potential of machine learning in the medical field. Approaches like these can potentially offer healthcare professionals better diagnostic tools and opportunities for research. While this project may not revolutionize the medical field entirely, it certainly highlights the transformative impact of machine learning on medical diagnosis and treatment.