CS5242 Project Report

Background of the Project:

The project offers an opportunity to work and learn from real-world medical data and explore the complexities of transfer learning in the pursuit of improving hematologic cell identification. It aims to capture the difference in accuracies before and after pretraining on specific WBC datasets.

The primary goal of this project is to create a system that can accurately distinguish between five classes of white blood cells. The project utilizes advanced deep learning methods and image analysis techniques to assist in the precise identification of different types of white blood cells, aiming to resolve the challenges associated with hematologic cell recognition.

Background of the Dataset:

The project involves working on 3 datasets namely PRCC, Camelyon16 and WBC.

- The WBC dataset involves microscopic images of different white blood cell types, aiding in the classification of basophils, eosinophils, lymphocytes, monocytes, and neutrophils.
  The WBC dataset displays varying levels of data imbalance, with a significant difference in the number of images across different cell types, thus leading to reduced accuracies for the underrepresented classes.

- The PRCC dataset is all about studying a type of kidney cancer called Papillary Renal Cell Carcinoma. In the context of “Papillary Renal Cell Carcinoma subtyping”, the term "subtyping" refers to the process of categorizing or classifying different subtypes of this kidney cancer based on specific characteristics. Pretraining on this dataset helps develop a more comprehensive understanding of cell structures and textures leading to improved generalization and pattern recognition.

- Camelyon16 dataset involves images identifying whether cancer has spread to lymph nodes, which are part of the body's immune system. This task is crucial for determining the stage of cancer and planning appropriate treatment strategies. It contributes to the pre-training process, augmenting the model's understanding of intricate medical imaging. This dataset is labeled and supplemented with segmentation masks for accurate classification.

ML Task

This machine learning task involves classification of the WBC dataset into five classes. Pre-training is performed using the PRCC and Camelyon16 datasets, with a portion of the training set accompanied by masks. The main objective is to develop an algorithm that uses the additional information from the pre-training datasets to enhance the model's performance on WBC images classification. ML task involves experimenting with varying portions of the WBC training set (100%, 50%, 10%, and 1%) to evaluate how the addition of supplementary data from pre-training datasets affects both the training process and overall model performance.

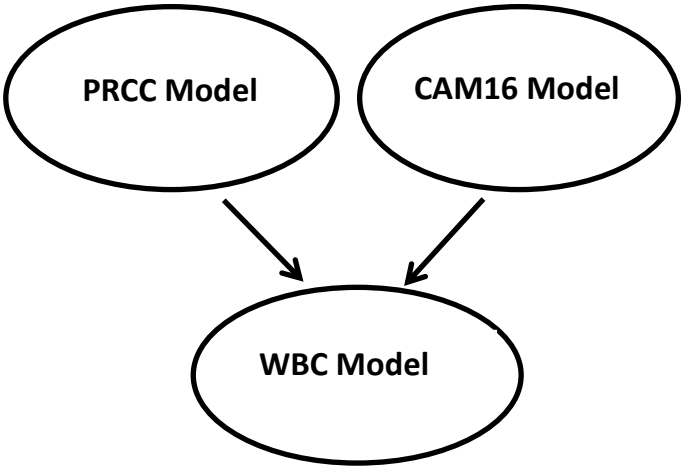
Impact to the world

The project aims to build a robust classification system that can aid medical professionals in

- precisely identifying blood cell types, leading to more precise and reliable diagnoses. This can be crucial for detecting and monitoring various hematological disorders and conditions.
- development of targeted and personalized treatment strategies, especially in cases where specific blood cell abnormalities are indicative of diseases or conditions.
- contributes to the discovery and management of conditions such as cancer, infections, allergies, and immune system disorders. Using the results, the doctors can prescribe treatment based on how far a cell has been infected by the disease.

Project Architecture:

In this project, I implemented a multi-step process that involved training the PRCC and CAM16 models on their respective datasets separately. Subsequently, I trained the white blood cell (WBC) dataset on various proportions of the data (1%, 10%, 50%, 100%). Next, I combined the features obtained from the PRCC and CAM16 models and used them to retrain the WBC dataset again on the same proportions of data (1%, 10%, 50%, 100%). This approach enabled the integration of knowledge from the pre-trained PRCC and CAM16 models to enhance the classification performance of the WBC dataset, thus increasing class wise and overall accuracies.



I am training the PRCC and CAM16 models separately and then merging their output latent features from them to be integrated into the training of the WBC models with varied datasets.

```
for i, data in enumerate(train loader, 0):
        inputs, labels = data
        inputs, labels = inputs.to(device), labels.to(device)
        prcc_features_tensor = prcc_model_features(inputs)
        cam16_features_tensor=cam16_model_features(inputs)
        combined_features = prcc_features_tensor +
        cam16_features_tensor
        wbc_outputs= net(inputs)
        combined_features += wbc_outputs
```

Some common features of my implementation across all models:

1) I am using the image in its original size for all 3 datasets. I am not resizing/downsizing it, because I want to capture intricate details and maintain the original structure as down sampling may lead to information loss and degradation in the quality of the reconstructed image.

I am maintaining the same architecture across all WBC models, with only minor modifications for models with pretraining (extraction and usage of latent features from prcc and cam16). I believe that despite using similar code and architecture, the **variations in dataset size and pretraining have a significant impact on model performance.** Maintaining consistency allows me to HIGHLIGHT the effects of different dataset sizes and pretraining.

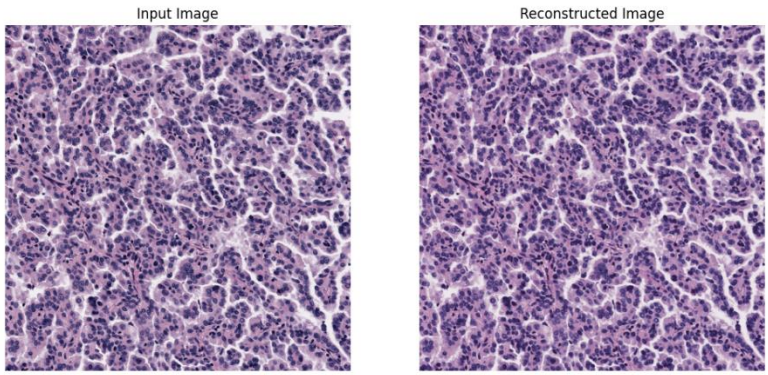
I am utilizing various features across the WBC and CAM16 models, some of which are as follows:

1. **CNN architecture**:
   Consisting of a fixed number of Convolutional layers with ReLU activation, Max Pooling layers, and Linear layers.
2. **Cross-entropy loss function.**
   Effectively penalizes the model based on the dissimilarity between predicted and actual class probabilities, encouraging accurate class assignments.
3. **Adam optimizer**
   It ensures adaptive learning rates and momentum for faster convergence and enhanced performance.
4. **weight decay (L2 regularization)**
   It adds a penalty term to the loss function, discouraging the model from excessively relying on a few specific weight for predictions.
5. **ReduceLROnPlateau scheduler** for learning rate adjustment, enabling finer convergence.
6. Data augmentation techniques, including RandomResizedCrop and RandomHorizontalFlip.
7. **Batch normalization**.
8. **Kaiming (He) initialization** of weights.
9. Evaluation of training and validation accuracy and loss over epochs.
10. Train and Test loader in all models, loads the training dataset in batches of size 8, shuffles the data for better generalization, and utilizes 2 workers for parallel data loading to expedite the process.

My model has approximately 719,000 parameters, which are capable of being fine-tuned during the training process, which implies the model can capture intricate patterns and features within the input images, allowing it to handle complex tasks that require the identification of detailed and nuanced visual characteristics.

Model Architecture of PRCC model

I implemented an autoencoder architecture with a complex encoder-decoder network for image reconstruction. The autoencoder is trained on PRCC dataset of unlabeled data. The model is trained using a Mean Squared Error (MSE) loss function with Adam optimizer and learning rate reduction on plateau. The code iteratively trains the autoencoder model for 10 epochs while evaluating the training and validation losses.



Note:

I have saved the PRCC model as "autoencoder_model_complex.pth" in my final submission of models, as I have used the same name in the other model's code when pretraining.

Model Architecture of CAM16 model

This model implements a convolutional neural network architecture tailored for image classification tasks. The model utilizes a residual network, having multiple layers of basic residual blocks for effective feature extraction and classification. It employs the Adam optimizer with a learning rate of 0.001 and weight decay for regularization. The model's performance is measured based on the training and validation accuracy, as well as the corresponding loss values, all of which are printed and tracked during the training process.

- By independently training PRCC and CAM16, I have enabled each model to concentrate exclusively on its designated task, devoid of any potential interference from the other. This specialized approach facilitates a more refined and concentrated understanding of the unique patterns and features within their respective datasets.
- From the PRCC dataset, the model can potentially learn about cellular structures, abnormal cell morphology, and various characteristics relevant to cancerous cells. This knowledge aids in recognizing intricate patterns and improving generalization for complex cell structure analysis.
- From the CAM16 dataset, the model can learn about color grading, perception of light, and color adaptation mechanisms. This understanding can contribute to robust color feature extraction, enhancing the model's ability to differentiate subtle color variations and detect anomalies or tumors more effectively.
- As the inherent nature of PRCC and CAM16 are different, I am not training them in a linear fashion.(not training PRCC first, then use it to train CAM16 and then use CAM16 to train WBCs. Instead training PRCC and WBC independently and combine their learnings for WBC).

Here are the problems I have faced and how I have overcome those problems in my implementation:

1. **Data Imbalance:**
   The dataset exhibits an uneven distribution of different WBC classes. Thus, the model might struggle to learn from the minority classes, leading to biased predictions and reduced accuracy for those classes.
   I have not added more training data or balanced the dataset. But I used "RandomResizedCrop" and "RandomHorizontalFlip" during train data augmentation, these processes introduce variations in the training data by presenting different perspectives of the same image, thereby diversifying the existing training samples.
2. **Overfitting:**
   I used weight decay (L2 regularization) and Kaiming (He) initialization, which have helped mitigate overfitting by penalizing large weights and ensuring proper initialization of the network layers making it more stable.

Results

Final validation accuracies (in %)of all WBC models.

Model	Without Pretraining	With Pretraining
WBC_1	63.48	74.13
WBC_10	78.70	84
WBC_50	90	91.09
WBC_100	90.34	93.04

My Inferences:

I observed that we can see increase in validation accuracies when,

i) when progressing from left to right across the last 2 columns.

There is a significant increase in values for all rows(before and after pretraining), irrespective of the dataset size which can be **attributed to transfer learning.** Both the PRCC and Camelyon16 datasets contain diverse medical images that expose the model to various patterns and structures. Although these datasets are not directly related to hematologic cells, the features learned from them can help the model recognize patterns in cell structures, draw outlines or borders for cancer cell, distinguish between cancer and normal cells based on color, shape texture.

ii) when moving from the top to bottom across all rows individually in the last 2 columns.

The increase in values across the rows as the dataset size increases can be **attributed to the availability of more data for training the model.** When more data is used for training, the model can learn from a richer and more diverse set of examples, which enables it to capture complex patterns and nuances within the data more effectively.

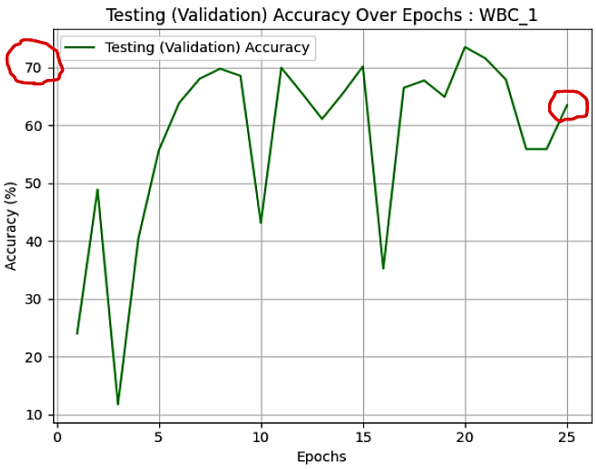
Specifically, when using a larger percentage of the WBC training set (e.g., 100% or 50%), the model has access to a more comprehensive representation of the various white blood cell types, their characteristics, and their distinguishing features.

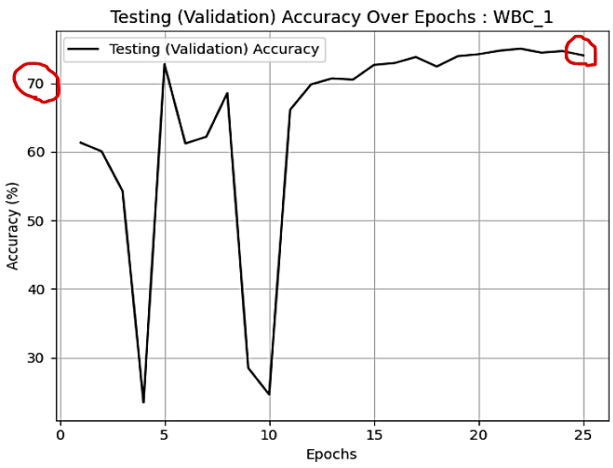
Conversely, when only a small percentage of the WBC training set is used (e.g., 10% or 1%), the model's training data is limited, which can restrict its ability to learn the intricate details and subtle differences between different white blood cell types.

I noticed a significant increase (5-10% increase) in final validation accuracies scores before and after validation on small datasets (WBC1 and WBC10). Comparatively the increase in validation accuracies started diminishing (1-3% increase) as dataset size increased (WBC50 and WBC100). This is due to the size of dataset again, as model generalizes less and learns more in bigger datasets which it cannot do in smaller datasets.

I have plotted the graph of loss and accuracies vs epochs individually on both training and validation data for all models. I will take an example of an output graph before and after pretraining to understand the effects of transfer learning. I can explain it for this example case of WBC1, but the graphs might differ for each model.

Graph for validation accuracy of WBC_1 model with and without pretraining:





We can see considerable difference in values at beginning and final epoch for both cases.

Note : As the WBC_1 dataset contains high variability in terms of image quality, lighting conditions and variations in cell appearances, it leads to inconsistencies in the model's ability to generalize effectively, resulting in varying validation accuracies. That is why in some graphs of validation data, we could observe steep increases and decreases. As the model is complex and the dataset is also very small, there is significant variation in WBC_1 graph.

Total F1 score of all WBC models

Model	Without Pretraining	With Pretraining
WBC_1	0.5450	0.699
WBC_10	0.784	0.843
WBC_50	0.901	0.899
WBC_100	0.899	0.900

*The same pattern as I noticed in above validation accuracies also applies here.

The F1 score is related to the dataset and transfer learning in the following ways:

i) **Dataset Size and F1 Score:** A larger dataset generally provides more mixed coverage of the possible variations within the data, enabling the model to learn more robust representations and perform better on the classification task. With a larger dataset, the model has more data to learn from, which can lead to high recall and precision, and it can strike a balance between both, leading to a higher F1 score.

ii) **Transfer Learning and F1 Score**: When additional information is incorporated during training, the model can benefit from supplementary insights that may enhance its understanding of the data, By pretraining on diverse datasets, the model can acquire diverse knowledge that can be fine-tuned for specific tasks. This enables the model to better understand and classify the different white blood cell types in the WBC dataset, ultimately leading to improved F1 scores.

In summary, both dataset size and the utilization of transfer learning through pretraining play crucial roles in improving the model's ability to capture complex patterns and features, ultimately **leading to higher F1 scores** in the WBC classification task.

Class-wise Accuracies of all WBC models

As my primary concern is the model's performance in correctly identifying instances belonging to a particular class, I am considering only true positives. This metric gives me a clear understanding of how well the model can accurately classify instances within each specific class.

I noticed across all class-wise accuracies, some classes had comparatively lesser accuracy scores despite pre training and across different datasets. Example : Eosinopil and Monocyte. (I will explain why these 2 classes performed poorly when compared to Basophil which had less data when compared to these 2 classes, in Scientific discovery section).

Formula I used:

Accuracy=
Number of True Positives/
Total Number of Instances

Not considering True negatives

WBC_10 (without Pretraining)

```
Accuracy of class Basophil: 100.0%
Accuracy of class Eosinophil: 25.396825396825395%
Accuracy of class Lymphocyte: 96.11650485436893%
Accuracy of class Monocyte: 44.21052631578947%
Accuracy of class Neutrophil: 80.64211520302172%
```

Increased Dataset size case-1

```
Accuracy of class Basophil: 100.0%
Accuracy of class Eosinophil: 42.06349206349206%
Accuracy of class Lymphocyte: 94.66019417475728%
Accuracy of class Monocyte: 35.78947368421053%
Accuracy of class Neutrophil: 90.08498583569406%
```

WBC_50 (without Pretraining)

with pretraining case-2

```
Accuracy of class Basophil: 100.0%
Accuracy of class Eosinophil: 42.06349206349206%
Accuracy of class Lymphocyte: 94.66019417475728%
Accuracy of class Monocyte: 35.78947368421053%
Accuracy of class Neutrophil: 90.08498583569406%
```

WBC_10 (with Pretraining)

Scientific Discoveries

The class wise accuracies, total validation accuracies and overall F1-score improved in 2 cases :

1. When the dataset size increased.
2. After Pretraining (after transfer learning was implemented)

1.) I observed that **Basophil had better accuracy** when compared to Eosinophil and Monocyte because:

i) **Distinctive Features**: Basophils possess relatively distinct and easily recognizable morphological or structural features that differentiate them from other types of white blood cells. This distinctiveness makes it easier for the model to learn and accurately classify Basophils, even with a smaller amount of training data.

ii) **Representative Training Data:** Although the number of training instances for Basophils is comparatively small, the available data might be highly representative of the characteristics of Basophils, allowing the model to learn effectively and generalize well to unseen instances.

iii) **Lack of Complex Variations:** The Basophil class has fewer variations or complexities compared to other classes, making it relatively easier for the model to distinguish and classify Basophils accurately, leading to the high accuracy score.

iv) **Challenges in Feature Extraction:** Eosinophils and Monocytes possess intricate and subtle features that are difficult to capture effectively, especially with limited training data. Inadequate representation of these complex features might have hindered the model's ability to make broad predictions well, resulting in lower accuracies for these classes.

2.) Class Accuracy of **Monocyte** decreased even when dataset size increased and with pretraining:

Challenges with Monocyte Class: The decrease in accuracy for the Monocyte class, especially in the pretraining scenario, suggests the complexity of identifying Monocytes accurately. Identifying Monocytes accurately is particularly complex due to their subtle structural variations and overlaps with other cell types, making it challenging for the model to distinguish them effectively.

3.) Accuracy of classes **Lymphocyte** had the least increase in accuracies among all classes.

The consistently high accuracy of the Lymphocyte class, both with and without pretraining, indicates the effectiveness of the model in distinguishing this class accurately. It implies that the Lymphocyte class is easily distinguishable by the model. This undermines the potential of ML tasks in precisely identifying Lymphocytes.

Conclusion

The integration of advanced machine learning models such transfer learning, as demonstrated in this project, holds significant promise for revolutionizing the study and treatment of hematologic analysis and diagnosis in the medical field. By accurately categorizing these cells, doctors can quickly identify specific problems like cancer, allergies, infections, and issues with the immune system. Depending upon the infected level, then can curate the medication to be taken and this also helps them give the right treatments faster.

Furthermore, the demonstrated effectiveness of pretraining the model with diverse medical imaging datasets (of PRCC and CAM16) highlights the potential of transfer learning in enhancing the model's overall performance. Incorporating pretraining methodologies can equip the model with a broader understanding of complex cellular structures and patterns, enabling it to make more nuanced and informed classifications.

Additionally, continued research and collaboration between the fields of technology and medicine are essential to ensure the seamless integration of these innovative techniques into real-world clinical settings.