Assignment 3 Instruction

Due Date: 30 November, 23:59

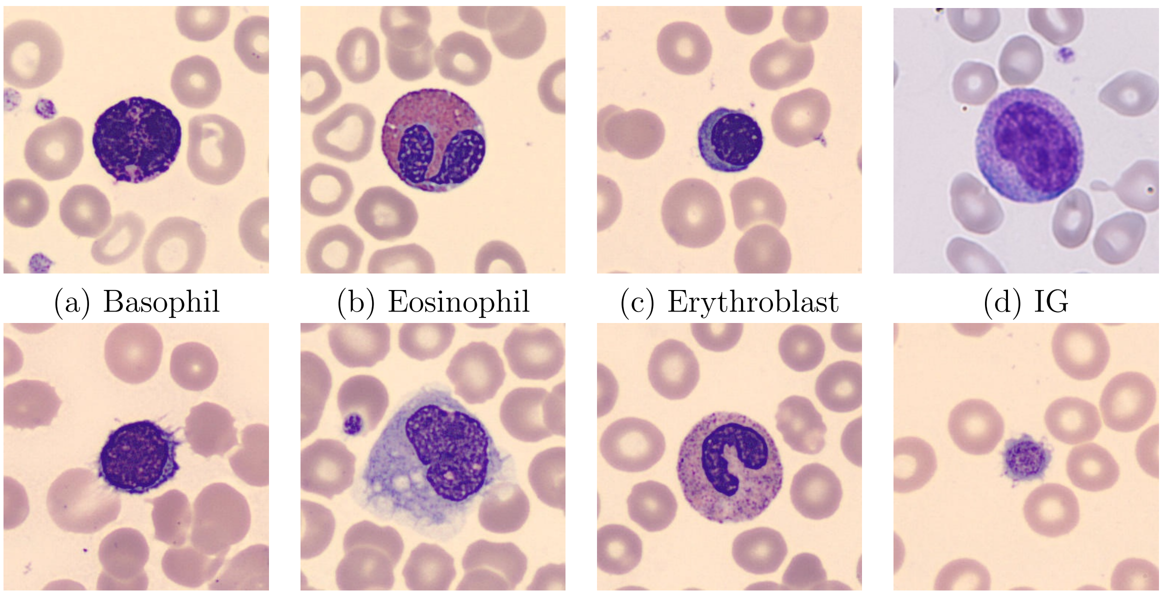
This is the final due date.

No extension will be granted due to the final grade release schedule.

Overview

The goal of this final assignment is to evaluate your ability to apply the deep learning knowledge and techniques learned throughout this semester to a real-world image classification task. You will be provided with a dataset containing microscopic images of different types of cells, and your objective is to design, implement, train, and evaluate a deep learning model capable of accurately classifying these images into their corresponding categories. You are encouraged to apply **any deep learning methods or architectures introduced during the semester**, selecting and justifying your approach based on its suitability for the given problem. Through this assignment, you will demonstrate your understanding of data preprocessing, model construction, training procedures, and performance evaluation. In addition to technical implementation, you will also be required to prepare a **comprehensive technical report (up to 6 pages)** that clearly presents the motivation, methodology, experimental design, and results analysis. This report should reflect your ability to reason about design choices, interpret experimental outcomes, and communicate findings in a structured and professional manner.

Background



(a) Basophil (b) Eosinophil (c) Erythroblast (d) IG

(e) Lymphocyte (f) Monocyte (g) Neutrophil (h) Platelet

Figure 1: Examples of the eight blood-cell categories used in this assignment. Each image has a resolution of 360 × 363 pixels.

Automatic recognition of blood cells is an important task in medical image analysis, as it supports clinical diagnosis and reduces the need for manual examination by experts. Deep learning methods have proven highly effective for such problems by learning discriminative features directly from raw images. In this assignment, you will apply the deep learning principles and models covered during the semester to build a system that classifies microscopic images of blood cells into their respective categories, demonstrating both technical understanding and practical implementation skills.

Dataset Description

The dataset consists of **real microscopic blood-cell images** collected from a clinical laboratory. Each image is of size **360 × 363 pixels** in JPG format. You are provided with **3,200 training images** and **1,000 testing images**, covering **eight major cell types** commonly observed in peripheral blood.

To provide visual intuition about the classification task, Figure 1 shows one representative image from each class. These examples illustrate the visual diversity of blood cells and the challenge of automatic classification.

Each image belongs to a **class**, which represents a specific category of blood cell. The mapping between class names and their corresponding numerical IDs is defined in the file `class_map.json` (see Table 1 for details). Please **do not modify** this file, as it is used for consistent label encoding across all submissions. In addition, note that the **file names themselves contain the cell-type abbreviation** (e.g., `BA_XXXX.jpg` for Basophil, `LY_XXXX.jpg` for Lymphocyte), which provides an intuitive hint of the ground-truth category in the training set.

Class ID	**Cell Type**
0	Basophil (BA)
1	Eosinophil (EO)
2	Erythroblast (ERB)
3	Immature Granulocyte (IG)
4	Lymphocyte (LY)
5	Monocyte (MO)
6	Neutrophil (BNE)
7	Platelet

Table 1: Mapping between class ID and blood-cell type.

General Guidance and Requirements

This section outlines the general rules and recommendations for completing this assignment. Please read carefully before you start model implementation and experimentation. *Failure to comply with these guidelines may result in penalties or disqualification of the submission.*

Do's

- You may use **any model architecture** introduced during this trimester's Deep Learning Fundamentals (DLF) course as your main model, such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Transformers, Autoencoders, or even large language models, if appropriate to your design.

- You are encouraged to explore and compare multiple network structures, loss functions, or training strategies as part of your experimental analysis.
- You may use any training techniques covered in this course, including various optimizers (e.g., Adam, SGD), learning-rate scheduling, normalization methods (e.g., BatchNorm, LayerNorm), and regularization techniques (e.g., dropout, weight decay).
- Data augmentation is allowed and encouraged to improve model generalization.
- You may use open-source libraries such as `PyTorch`, `TensorFlow`, `scikit-learn`, `NumPy`, or `Pandas` for model development and evaluation.
- You are encouraged to maintain clean, modular, and reproducible code using Jupyter Notebook, following good programming practices.

Don'ts

- You must NOT use any pre-trained models or weights, including but not limited to classifiers pre-trained on ImageNet or other external datasets.
- You must NOT use any external labeled datasets beyond the provided training data. The only acceptable modification to the data is through data augmentation or data cleaning.
- You must NOT manually label, alter, or annotate any of the provided images in the testing set.
- You must NOT hard-code any results or predictions in your code or report. All results must be generated from actual model inference.
- You must NOT use any third-party APIs, pre-built AI services, or auto-training frameworks that automate model selection or hyperparameter tuning.

These rules are designed to ensure fairness and academic integrity while allowing you to fully demonstrate your understanding of deep learning model design, implementation, and analysis. If you have any questions that are not addressed in this section, please post them in the MyUni Discussion Board for clarification.

Task 1: Technical Report (15%)

You **must** present your report using one of the official conference LaTeX templates: either CVPR (two-column style) or ICLR (one-column style). You are required to select **only one** of these two templates for your submission. You may use Overleaf, a free online LaTeX editor, to edit and compile your report. To ensure fairness in page limits and formatting, **you must not modify any default settings of the chosen template**, including but not limited to font size, line spacing, margins, or column width. *Submissions that do not follow the CVPR or ICLR official formatting rules may receive mark deductions.*

The report must not exceed **6 pages** (including figures and tables but excluding references). Any content beyond this page limit **will not be marked**, *e.g.*, if key sections such as the conclusion or results appear outside the 6-page limit, they will receive **no credit**. In addition, each extra page beyond the limit will incur a **1-mark deduction**.

A single PDF file must be submitted to GradeScope (link available on the Assignment 3 page). All content in your report must be your own. You may use generative AI tools for brainstorming ideas, checking grammar, or proofreading text. However, **you must not use AI tools to generate full paragraphs, technical explanations, or any substantive part of your report.** Doing so will be considered a breach of academic integrity and may be reported under the University's Academic Misconduct Policy. You are fully responsible for the originality, correctness, and authenticity of all submitted content.

The report should contain the following sections:

- **Abstract & Introduction (2%)** Begin your report with a short **Abstract** that summarizes the problem, motivation, dataset, main approach, and key findings. Follow this with an **Introduction** section that clearly describes the problem definition, motivation, and relevant background context. Outline your proposed method, dataset usage, and experimental setup at a high level. Conclude with the main insights or contributions from your work. All background sources should be appropriately cited.

- **Related Work (1%)** Summarize previous studies and established methods relevant to image classification or blood-cell analysis. Compare these approaches with your own, highlighting differences and justifying your design choices.

- **Method (5%)** Describe your model architecture, algorithm, and training process in detail. You are required to select **one Deep Learning architecture** covered in this trimester's DLF course (e.g., CNNs, RNNs, Transformers, Autoencoders, or other valid architectures) as your **main model**. You should apply optimization or improvement techniques to this model and justify your design choices. Clearly explain the structure, components, and training procedure so that a technically skilled reader can reproduce your results. If you propose enhancements or variations, describe their motivation and expected benefits.

- **Experiments and Results (5%)** Describe your experimental setup, dataset division, evaluation metrics, and parameter settings. Present your results and include a **comparison with at least two baseline methods**. These baselines are not limited to deep learning models — they may include traditional machine learning methods or simpler neural architectures. Baseline models do not need to be highly optimized; they can also represent preliminary experiments or earlier trials you conducted before deciding on your main architecture. You should briefly explain why these models were chosen and what insights their results provide. In addition, conduct a **comprehensive ablation study** to evaluate the contribution of individual components, training settings, or design choices in your main model. The goal is to demonstrate a solid understanding of experimental methodology and result interpretation, rather than simply achieving the highest performance.

- **Discussion (2%)** Reflect on your findings, discuss challenges and limitations, and suggest possible improvements or directions for future work.

- **References (no marks; deductions for issues)** Include all cited works using a consistent academic format. While correct referencing does *not* provide additional marks, **missing or incorrect citations may result in mark deductions**. Ensure that all non-original content is properly attributed to maintain academic integrity.

- **Template and Formatting (no marks; deductions for issues)** You must prepare your report using either the official **CVPR** (two-column) or **ICLR** (one-column) LaTeX. Follow the default font size, line spacing, and margin settings in the chosen template. While correct formatting does not provide additional marks, **failure to comply with the required template or any manual modification of formatting parameters (e.g., font size, spacing, margins)** may result in mark deductions to ensure fairness in the page limit policy.

Task 2: Code (10%)

In this section, students will be assessed on their ability to prepare data, implement a deep learning model from scratch, and produce reproducible experiments, while ensuring that their code is well-structured and of high quality. Readability and clarity of the code are essential. All results reported in the report must be traceable to the code implementation. It is not necessary to include additional results or plots within the code itself, as these will not contribute to the report. All analysis and discussion should be presented in the written report rather than embedded in the code.

- **Preprocessing (2%)** Implement appropriate techniques for data preparation and visualization, such as data cleaning, normalization, and dataset splitting. Brief visualizations are encouraged to assist understanding of the dataset (e.g., class distribution, feature statistics, or correlations). These are intended to demonstrate your understanding of the data rather than to serve as part of the report's analysis.

- **Model Implementation (4%)** Implement your **main proposed method from scratch** using Python OOP in either **PyTorch** or **TensorFlow**. The implementation should demonstrate clear modular structure and good coding practices. Each major component should include a short (1–2 line) description explaining its purpose. You must also include at least **two baseline models**, which can be either traditional Statistical ML or Deep Learning methods. These baseline models may be implemented using built-in or pre-trained modules and do not need to be optimized — they can represent preliminary trials or simple benchmarks for comparison. The proposed main model, however, **must be implemented from scratch** by PyTorch or TensorFlow. All implementations must align with the description and experiments presented in your report.

- **Experiments (4%)** Your experiment code must align with the experiments and results presented in your report. The code should be able to **reproduce all reported results** and show evidence that they were generated from your actual implementation. Experimental design should demonstrate understanding of model evaluation and effectiveness, including a well-organized **ablation study** or controlled comparisons. While brief comments describing the purpose of each experiment are encouraged, you should not include detailed analysis or discussion within the code itself — these belong in your report. All experiment results and configurations must be consistent between your code submission and your written report.

- **Code Structure and Organization (no marks; deductions for issues)** Your code should be cleanly structured, logically organized, and easy to navigate. Clear separation of model, training, data processing, and utility components is expected.

While this section carries no direct marks, **poor structure, lack of readability, or disorganized code that prevents the marker from identifying core functionality may result in mark deductions**. Ensure that filenames, variable names, and comments are meaningful and consistent with your report.

Task 3: Model Performance (10%)

In this task, your trained model will be evaluated on a **confidential test set**, whose ground truth labels are not provided. You are required to generate predictions on the test set and submit them in the specified format for automatic evaluation on GradeScope. The performance of your model will be ranked among all students based on test accuracy. Higher ranks will receive higher marks according to the following criteria:

- Top 10% of students by accuracy: **10 marks**
- Top 20% of students by accuracy: **8 marks**
- Top 50% of students by accuracy: **6 marks**
- All other students achieving accuracy above the minimum threshold: **5 marks**
- Submissions below the minimum threshold (50% accuracy): **0 marks**

Your submission will be collected in **JSON format**. The JSON file should follow the structure shown in the provided `predict_labels.json` file, where the key is the image filename (sample ID) and the value is the predicted class label represented by a number. An example format is shown below:

```
{
    "img_0.jpg": 0,
    "img_1.jpg": 1,
    "img_2.jpg": 2,
    ...
}
```

You must name your submission file exactly as **prediction_labels.json** and upload it to GradeScope under `Assignment 3 - Task 3: Cell Classification Prediction Results`. The autograder will automatically compare your predictions against the hidden ground truth labels and return your accuracy score. **Submissions that use an incorrect filename or deviate from the required format will not be evaluated.**

Important Notes (MUST READ BEFORE SUBMISSION):

- The provided **prediction_labels.json** is only an example to illustrate the expected format. You must generate and submit your own prediction file following this structure.
- To prevent overfitting on the confidential test set, you are allowed to submit at most **three times** throughout the assignment period. Please use these submission opportunities wisely and only submit when your model has been thoroughly validated on your local validation set. Any additional submissions beyond the third attempt will not be recorded on the leaderboard and will incur mark deductions. Any attempt to

bypass this limit, such as creating or using multiple accounts to gain additional submission opportunities, will be treated as a serious academic misconduct and will result in mark deductions or further disciplinary actions. If your submission fails to follow the required format (for example, incorrect filename or invalid JSON structure) and therefore cannot be evaluated, **the attempt may still be counted** and no additional submission allowance will be granted.