UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Computer Vision (Curs 295II022)

Problem-Based learning project guideline

Group information (Maximum 4 students):

Student 1: Name: ______________________ ID: __________
Student 2: Name: ______________________ ID: __________
Student 3: Name: ______________________ ID: __________
Student 4: Name: ______________________ ID: __________

1 Project overview

This project represents **25% of the final grade** for the Computer Vision course and is part of **Block 2: Pattern Recognition**. Students will develop a deep learning model capable of classifying the five main types of **leukocytes (white blood cells)** using convolutional or transformer-based neural networks.

Leukocytes are components of the immune system. Their main function is to defend the body against infections and diseases by identifying and destroying pathogens such as bacteria, viruses, and parasites. There are five principal leukocyte types, each with distinct biological roles:

- **Lymphocytes:** Participate in immune responses. B lymphocytes produce antibodies, and T lymphocytes destroy infected or cancerous cells.
- **Monocytes:** Transform into macrophages and dendritic cells that remove pathogens and dead cells while presenting antigens to activate lymphocytes.
- **Basophils:** Release histamine and other mediators during allergic and inflammatory reactions.
- **Eosinophils:** Combat parasitic infections and regulate inflammatory responses in allergic or asthmatic conditions.
- **Neutrophils:** The most abundant leukocytes and the first to respond to bacterial infections, capable of phagocytizing pathogens and releasing enzymes.

Accurate classification of leukocytes supports medical diagnosis and monitoring of immune-related or hematological disorders.

The dataset provided contains **500 labeled images per class**, distributed as follows:

Label	Number of images
Lymphocyte	500
Neutrophil	500
Eosinophil	500
Basophil	500
Monocyte	500

Working modality: Groups of up to 4 students.

2 Main objective

To design, train, and evaluate a convolutional neural network (CNN) or a Vision Transformer (ViT) using **fastai**, capable of classifying leukocytes into their five categories. The model will later be tested with an independent dataset from a different patient to assess generalization performance.

3 Project tasks

3.1 Task overview

Students must train a deep learning model using the provided leukocyte dataset and evaluate its performance in classifying the five main leukocyte types. The work is divided into three stages, each to be implemented in a separate Jupyter notebook, ensuring reproducibility and allowing comparisons between experiments.

1. **Data preparation and partitioning.**
   - Load the dataset and verify that each leukocyte class (*lymphocyte, neutrophil, eosinophil, basophil, monocyte*) contains exactly **500 images**.
   - Create a partition into training, validation, and test subsets.
   - Save the resulting dataframe so that all later experiments use the same division.
   - This first notebook should focus exclusively on dataset exploration, preparation, and saving the split information, without training any model yet.
2. **Model training.**
   - Load the prepared training and validation subsets.
   - Design and train a model using the **fastai** library. You may implement a convolutional neural network (CNN) or a Vision Transformer (ViT), depending on your strategy and reasoning.
   - Explore different configurations, such as learning rate, optimizer, batch size, or number of epochs, and monitor their influence on performance.
   - Visualize training and validation accuracy/loss curves to evaluate model convergence and possible overfitting.
   - Save your best-performing model using:

     ```
     learn.export('model.pkl')
     ```

     This exported file will later be used for testing and external evaluation.
3. **Model evaluation.**
   - Load the test subset and evaluate the final model.
   - Compute the accuracy and generate the confusion matrix to analyze class-level behavior.
   - Include visualizations of correctly and incorrectly classified images to interpret the model's decisions.
   - Summarize the observed results and provide a short discussion in the PDF report regarding performance, class balance, and possible limitations.

3.2 External evaluation

Your submitted model will be tested on a new dataset from another patient. The goal is for the model to achieve similar accuracy to your test results. A significant accuracy drop will suggest overfitting, poor generalization, or suboptimal model selection.

- A global accuracy above **92%** on the external dataset will correspond to a grade of **10**.
- Lower accuracy values will be graded proportionally.

To support your preparation, two additional files are provided:

- **test_second_dataset.zip**: This file contains images from a second dataset. You may use these images to check how your model behaves on unseen samples. Run your model on these images and examine the predicted labels and their probabilities. If the predictions match the expected class with high probability, it indicates that your model is responding consistently. These samples can help you refine your model.

- **test your model.ipynb (mandatory use)**: This notebook contains the code that will be used to load and test your model during the evaluation. Before submitting your work, you must load your exported model using this exact procedure. The model must load exclusively with `load_learner`, without any additional libraries, custom functions, or manual modifications. If the file does not load with `load_learner` alone, it means the export was not done correctly, or the architecture was altered in a way that prevents loading. It is necessary that your model can be loaded exactly in this manner, as this will be the method used for external evaluation.

4 Deliverables (Atenea / Block 2)

- **1. PDF report (maximum 2 pages):**
  - Names and IDs of all group members.
  - Description of the chosen architecture and training strategy.
  - Training, validation, and test accuracy and loss plots.
  - Confusion matrix for the test set.
  - Link to a shared Google Drive folder if the model file exceeds Atenea's upload size limit.
- **2. Model file (.pkl):** Export your trained model from Colab using:

  ```
  learn.export('model.pkl')
  ```

  Submit the resulting file in `.pkl` format. If the file size exceeds Atenea's limit, include a Drive link (with access granted to `kevin.barrera@upc.edu`) inside your report. Only models that can be loaded correctly will be graded.

5 File naming

All files must be submitted together in a single ZIP folder named:

```
LastName_Student1_LastName_Student2_LastName_Student3_LastName_Student4.zip
```

This folder must contain:

- `model.pkl`
- `report.pdf`

6 Final considerations

- **Groups:** Up to 4 students.
- **Deadline:** December 11, 2025, 23:59 (CEST).
- **Late submissions:** 30% deduction per day after the deadline.
- **Report:** Maximum length of 2 pages (PDF), including plots and confusion matrix.
- **Submission:** All files must be uploaded to **Atenea** (or include a Drive link in the report if the model exceeds the size limit).
- **Plagiarism policy:** If two or more groups submit projects with identical code structure, architectures, training results, or figures, it will be considered **copying**. Since such coincidences are highly unlikely, **all involved students will receive a grade of 0**.

Good luck!