

# Build an AI hematologist

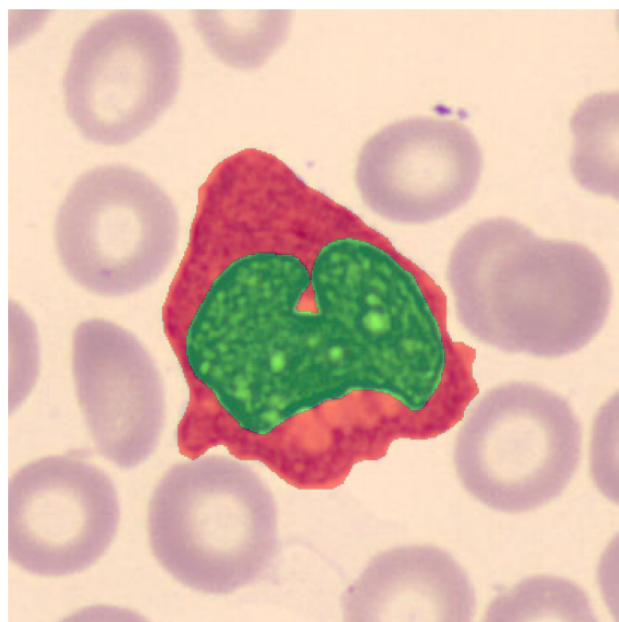
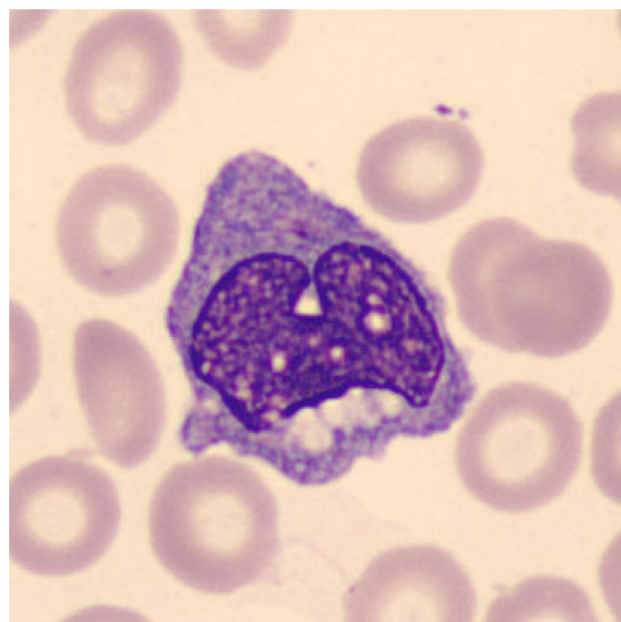
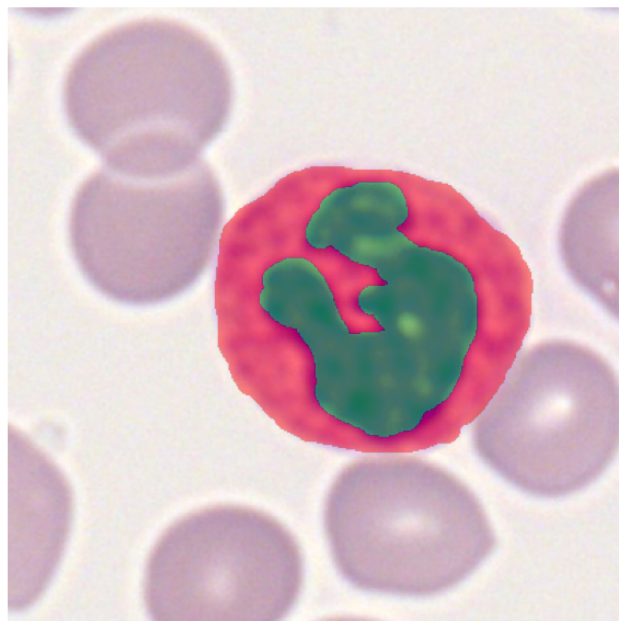
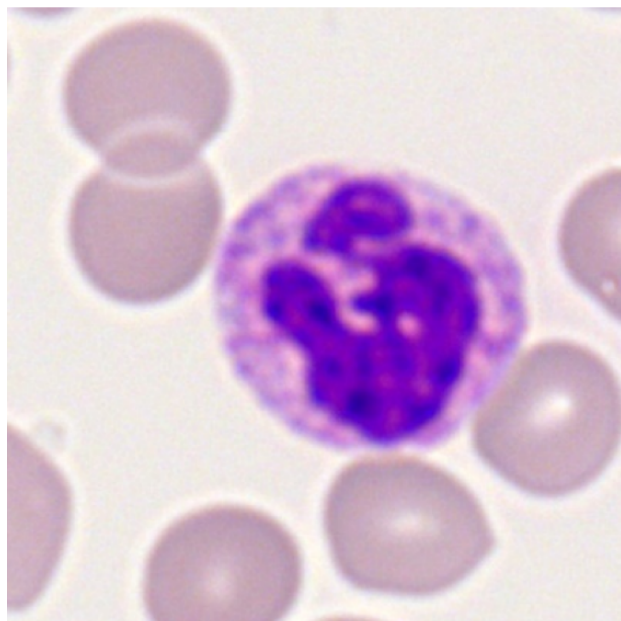
---

## Background

Microscopic examination and quantification of blood cells is a pivotal cornerstone of hematological diagnostics. This task, traditionally performed manually by hematologists, involves annotating blood cells, which can be both time-consuming and monotonous. This quiz is designed to challenge students to create an AI model capable of automating the segmentation of blood cells, thereby streamlining the diagnostic process.

## Task and Dataset

This is a semantic segmentation task and the annotation contains two classes: cytoplasm and nucleus. The provided input images are in standard RGB format (TIFF/JPG files), accompanied by single-channel, three-class masks. These masks use the values 0, 1, and 2 to represent the background, cytoplasm, and nucleus, respectively. Here are two randomly selected visualized examples.



The dataset is divided into three parts: 231 images for model training, 77 for internal validation, and 60 for external validation.

| Data Splits         | # Images | Notes  |
|---------------------|----------|--|
| Training            | 231      | Paired image and annotations   |
| Internal Validation | 77       | The images are from the same cell types as the training set.                           |
| External Validation | 60       | The images are from new cell types, aiming to test the model's generalization ability. |

The primary objective is to develop two distinct segmentation models using the training set. These models should be capable of automatically segmenting the cytoplasm and nucleus for the internal and external validation images. The first model should be based on a **convolutional neural network architecture**, while the second should utilize a **transformer-based network**. Please use the [nnU-Net](#) framework for this task.

Note: For each image, we only focus on segmenting one cell in the middle. You can exclude the segmentation outliers by using post-processing (e.g., [cc3d](#)) to extract the largest object.

## Evaluation

The segmentation quality will be evaluated using the Dice similarity coefficient score. This will be applied to two areas: the whole cell (label value: 1+2) and the nucleus (label value: 2). The implementation is publicly available at [medpy.metric.binary.dc](#)

```
from medpy import metric
from skimage import io
label_seg = io.imread('path to segmentation mask file')
label_gt = io.imread('path to ground-truth mask file')

whole_cell_dc = metric.binary.dc(label_seg>0, label_gt>0)
nucleus_dc = metric.binary.dc(label_seg==2, label_gt==2)
```

## Submission

Submissions should be organized into a zipped folder named `ML_quiz_submission_your_name.zip`, containing:

1. Report PDF: please detail the methodologies used in your model and include a GitHub link for replicating your results.
2. seg-Internal-UNet: please put the **internal validation set** segmentation results of U-Net (77 png mask files) inside this folder. File names should be saved as `image_name.split('.')[0] + '_label.png'`, such as `Internal_img_000_label.png`, ..., `Internal_img_076_label.png`.

3. seg-External-UNet: please put the **external validation set** segmentation results of U-Net (60 png mask files) inside this folder. File names should be saved as `image_name.split('.')[0] + '_label.png'`, such as `External_img_000_label.png`, ..., `External_img_059_label.png`.
4. seg-Internal-Trans: please put the **internal validation set** segmentation results of Transformers (77 png mask files) inside this folder. File names should be saved as `image_name.split('.')[0] + '_label.png'`, such as `Internal_img_000_label.png`, ..., `Internal_img_076_label.png`.
5. seg-External-Trans: please put the **external validation set** segmentation results of Transformers (60 png mask files) inside this folder. File names should be saved as `image_name.split('.')[0] + '_label.png'`, such as `External_img_000_label.png`, ..., `External_img_059_label.png`.