Formulation of the problems

- **Problem 1** is the one blood cell image classification problem with three classes: `0-lymphocytes` , `1-neutrophils` and `2-misc` classes.

- **Problem 2** is the diagnosis problem which determines whether there is a pathology (`1-pathology` class) or not (`0-bening` class).

Training of the ConvNet

For training classifiers of both problem 1 and problem 2 we should use the following command:

```
python3 convnettrain.py --settings path/to/settings.json
```

where **settings.json** file has the following structure:

```
{
    "data_path": "path/to/data",
    "img_size": 50,
    "epochs": 25,
    "batch_size": 20,
    "model_path": "models",
    "labels": ["0-misc", "1-lymphocytes", "2-neutrophils"]      # for problem I
    "labels": ["0-benign", "1-pathology"]                      # for problem II
}
```

and the *data_path* is a path to a train data directory that has the following structure:

```
data /
    train /
        0-class_0 /
            img_1
            img_2
            ...
        1-class_1 /
            img_1
            img_2
            ...
        ...
    test /
        0-class_0 /
            img_1
            img_2
            ...
        1-class_1 /
            img_1
            img_2
            ...
        ...
```

If the training is finished successfully we get two files in *model_path* directory:

```
model_cell_is50_ep25_bs20_1                # model file
model_cell_is50_ep25_bs20_1_summary.csv    # performance of the model like a CSV table
```

The model file we can use to classify our input images.

Classification with ConvNet

To classify an image we should use the following command:

```
python3 convnet.py --settings path/to/settings.json --img_file path/to/classifying_
```

where the *–settings* file is similar to settings file of the training step.

After this command the program returns the class of the classifying image *–img_file*.