Report exercise 4 - Binary vision

In this exercise you will use OpenCV's methods to program the function

```
int countNumberOfRice(cv::Mat input)
```

which counts the number of rice in the provided image. This is done by performing the following three step:

a. **Threshold image**: Choose an automatic threshold method from class, find the corresponding OpenCV implementation, and apply it to the image. Hint: read the documentation for the methods *cv::threshold* and *cv::adaptiveThreshold*.

b. **Apply morphological operations**: Remove noise in the binary image using morphological operations. Hint: read the documentation for the methods *cv::erode* and *cv::dilate*.

c. **Connected components**: Apply the connected components algorithm and return the number of classes. Hint: read the documentation for the method *connectedComponents*.

Your method should be able to run using the main method below. Hint: Do not over engineer the solution. The exercise can be solved with only 6 lines of code.

```
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc/imgproc.hpp>
#include <iostream>

int main(int argc, char* argv[])
{

    if(argc != 2){
        std::cout << "Usage: ./main <imageFile>" << std::endl;
        return -1;
    }

    cv::Mat src = cv::imread(argv[1], cv::IMREAD_GRAYSCALE);
    cv::namedWindow("Input image",cv::WINDOW_FULLSCREEN);
    cv::imshow("Input image", src);
    cv::waitKey(0);


    //There are N-1 rice grains since one class is background
    std::cout << "Number of objects: ";
    std::cout << countNumberOfRice1(src) - 1;
    std::cout << std::endl; }
}
```

The follwing is the image to be used in the exercise. It will be provided as a png-file on itsLearning.