

# My Approach (Solution)

---

## The Steps involved in solving the Objectives are as follows:

1. **Importing Required Libraries**
2. **Defining Required Function**
  - Defining custom "show" function for Image Visualization
2. **Image Pre-Processing**
  - GrayScale Conversion
  - Image Thresholding
  - Morphological Transformations (Noise Removal)
3. **Counting rice grains using the Contours method**
  - Working over Clear image to get insight into grain touching problem
3. **Applying Watershed Algorithm** (Solving grains touching problem)
  - Applying Watershed Algorithm for Solving Touching rice grains problem
4. **Counting total Rice grains and Broken Rice grains using the contour area**
  - For total rice grains counting: the Watershed method
  - For broken rice grains counting: A filter of an average area of broken rice grain.

## Explanation

The main Idea behind solving the objectives(Counting rice grains) is to make the provided image in the best possible format.

If there would be clarity in the image, and rice grains are well separated from the background image then there would be ease in counting them.

Then, Solving the Corner cases and hence building the solution.

**There are a total 3 major and challenging parts in building solutions:**

1. Image Preprocessing
2. Solving Grains touching problem
3. Counting broken rice

## Image Preprocessing

Image preprocessing is one of the vital parts of the solution, as, on this itself, Whole Ideology relies.

If the image is perfectly tuned as per the needs then it becomes easy to work further with the approaches.

The crucial part of Image preprocessing is tuning the Image, it takes a lot of trial and error to fix the parameter to the required value.

As part of Solution, I have used:

- Conversion of BGR Image to Grayscale Image
- Image Thresholding
- And removing noise from the thresholded image using morphologyEx (Opening)

After all the processes, the clear Image was ready for further use.

## **Solving Grains touching Problem**

After successfully Pre-processing the image, there comes the challenging part of the problem statement.

### **"Counting the rice grains that are touching each other"**

It is not even easy for a human eye in manual inspection process to count the rice grains. The small size and white color creates illusion.

Well for Machines,

Counting Rice grains are quite easy if they are well separated. The reason, is there are lots of algorithms out there and various techniques that can come in help.

But when there is an object touching or overlapping problem then there needs a lot of effort in grains classification.

In our case, it becomes more difficult as the rice grains are quite small in size.

Because of its small in size, We can't apply processes like erosion to get the touching corner part separated.

So, For Solving this problem, I have applied the Watershed Algorithm:

Watershed Algorithm is based on extracting sure background and foreground and then using markers will make the watershed algorithm run and detect the exact boundaries.

It is like, filling the valleys and then separating hills out of that.

## **Counting broken rice**

After using the Watershed algorithm, We will get the total number of rice present in the image, but counting broken rice grains are one of the typical tasks.

I used an area-based approach, where I put a threshold after several trial and errors that helps in classifying the rice grains into two categories.

**\*\*Either Broken rice grain or Full rice grain\*\*.**

If the area of the Image is below the provided threshold then it is counted in the broken rice category.

Thus with this, we are done with the solutions of both of the objectives .

Well,

I have added more approaches that came to my mind in the later section of this notebook. That too can be Implemented but I found this approach more Profound, thus it is part of the solution.