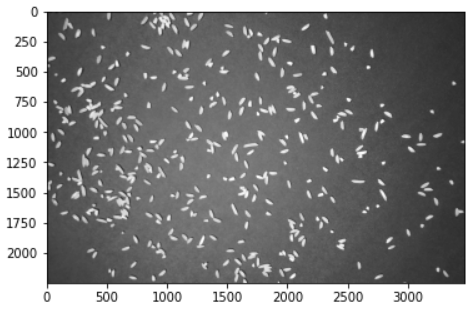
**Strategy:**

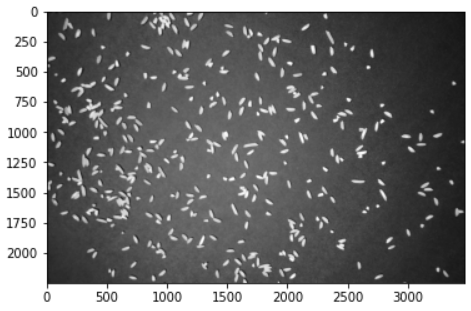
* From the train data, analyse the area of full grains and broken grains separately.
* Find the mean area of broken grains.
* On train data of mixed grains, computed the area of each grain and considered grains as broken if their area is area is less than the mean area of broken grains.
* If the results are satisfactory, apply the same on test data

**Steps followed:**

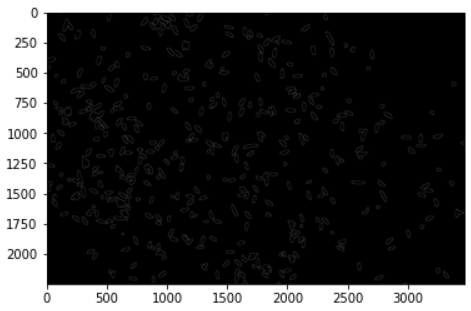
* To reduce the computational requirements and to simplify the algorithm being used, BGR image is converted to greyscale image using cv2.cvtColor() method from OpenCV.



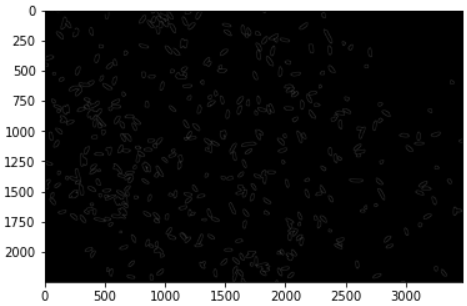
* To reduce the noise present on the input source image, cv2.GaussianBlur() function is used such that the sharp edges in images are smoothed while minimizing too much blurring.



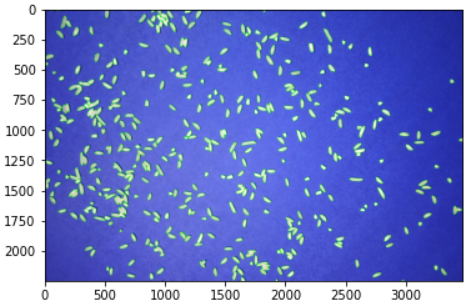
* To get the structural properties of an image, i.e., to find the boundaries of rice grains, edge detection is used so that it detects the discontinuities in brightness. Canny edge detector is used for the same.



* To add pixels to the boundaries of rice grains, cv2.dilate() function is used.



* To detect the rice grains and to find their count, we ned to get the line joining all the points along the boundaries of rice grains. So, to analyse the shape of the rice grains, contours are used. To find and draw contours, cv2.findContours() and cv2.drawContour() functions are used.



* **Count the number of broken grains:**
* Using cv2.contourAre() function, area of each contour is obtained.
* A minimum area threshold is set and if it is less than each contour area, then approxPlyDP() function is performed to approximate the shape of the contour.
* To highlight the region of polygons, cv2.boundingRect() function is used to draw an approximate rectangle around them.
* Dimensions (length and breadth) of all the bounding rectangles are stored separately and the mean dimensions are obtained to find the average area of rice grains. Average area = 3445
* For each bounding rectangle, (i.e., for each grain), if the area is less than the mean area, it is considered as a broken grain.
* Total number of broken grains = 271
* Total number of full grains = 214
* **Pseudocode:**

*Initialize a list to store the bounding rectangles*

*Initialize a variable to store the number of broken grains.*

*For all the contours:*

*Get counter area*

*Set a minimum threshold area*

*If contour area > minimum threshold area:*

*Approximate the contour to a polygon*

*Get its bounding rectangle.*

*Store the rectangles in a list*

*Get the dimensions of the bounding rectangle.*

*Set bounding rectangle*

*If area of bounding rect < mean area of bounding rect:*

*Update count of broken variables*

*Return count*

* Applied the same logic on the 5 test images.
* Recorded the count of total grains and broken grains for each image in the submission file.

