AUTOMATIC WHEAT GRAIN QUALITY ESTIMATION

Prakhar K, Anurendra Kumar, Satyam Dwivedi

EE604 Project Report, IIT Kanpur.

ABSTRACT

NAM (National Agriculture Market) is an online portal introduced by the government, to connect all the farmers online with the traders, so that farmers get the best price for their produce. Our project aims to facilitate the process by introducing automatic grain quality assessment from an image of spread out sample of grain. This is not a well explored problem so we do everything from scratch, in very constrained conditions. We segment out each particle from the image of spread out wheat grain sample and classify it as grain or impurity. Our model is able to distinguish between grain and impurities with a validation accuracy of 88%.

***Index Terms*—** Wheat grain, Segmentation, Classification.

1. INTRODUCTION

With the increasing demand of e-market in agriculture, there is considerable interest in automation of trading and farming. In the underdeveloped countries like India, traders often make much more money than farmers because there is no national standardized rules for rates and quality of the agricultural products. Currently the farmer takes his agricultural product to nearest 'Anaj Mandi' where a group of licensed traders instantly assess the grain quality and then bid for heap. Most of the traders practice cartelisation, due to which poor farmers get very less profit. We attempt to build a system which takes samples of different qualities of grains and gives a quality estimate of the grain which will later be utilized to predict the appropriate price. Currently, we have simplified the problem statement as Given an image of a fistfull of wheat grains spread evenly on a monocolor cloth, distinguish the grain from foreign particles to give a quality estimate of the sample.

2. DATASET

We couldn't find any existing grain dataset for this task, so we created our own dataset for the task. Initially we decided to focus on the 'wheat grain' only. So, we collected 8 samples of different qualities of wheat grain from the *Anaj Mandi, Kanpur*. Grains of each sample were manually separated into three categories: full grain, broken grain and foreign particles,

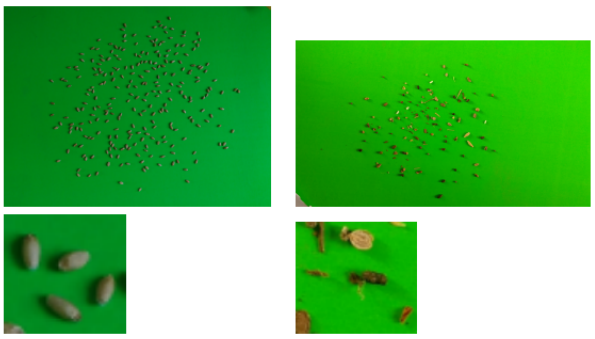

Fig. 1. A sample of full grain (left) and impurities(right) from our Wheat grain dataset

by the Mandi staff. For each of the 8 samples, high resolution pics (13-16mp) were clicked with the help of some students at IIT-Kanpur, by spreading the grains on a green background. 16 images of full grain (8 non-overlapping + 8 overlapping), 4 images for each kind of impurity, and 4 images of broken grain. In each set of 4 images, 1 is taken directly from above, and remaining are taken from random angles. The classification task is done on the overhead, non-overlapping grain images at first.

3. OVERVIEW OF PIPELINE

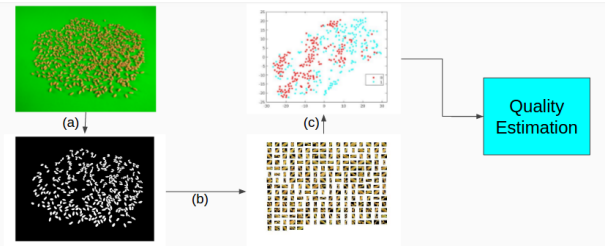

Fig. 2. Input image is first Pre-Processed(a) to give a binary image, which is then Segmented(b) to give different particles, which are then Classified(c) as grain/impurity.

We follow supervised machine learning approach to distinguish between grain particles and impurities. The first

step is to pre-process the acquired image to remove noise and make illumination invariant. It is followed by segmentation which extracts each grain/impurities as a separate image. These separate images are tested on our trained model to classify as grain/impurities. Finally we predict a quality estimate of the sample. It is assumed that the sample is representative of the heap. To model this assumption lot of different samples of the heap is to be taken and finally an average quality estimate of all the sampled pictures will be given.

4. IMAGE ACQUISITION AND PRE-PROCESSING

Fig. 3. Left one is the input image and right one is the binary image.

The image taken should have a mono-color background and sufficiently illuminated. In our dataset we have chosen green as background color.
i)To remove the shadow effect of grains we take the red channel of the image which doesn't have shadows in case when background is green.
ii)We then employ gaussian filter to remove noise and smoothen it.
iii)The image is further sharpened to enhance the edges of overlapping grains.
iv) It is then converted to binary image using threshold.
v) The binarized image has lots of dots and overlapping particles. We use morphological opening to remove stray dots and open up slightly overlapping particles. (insert image)

5. SEGMENTATION

We propose two level segmentation to segment each particle from the given image.

5.1. First Level Segmentation

This step takes binary image as input and gives the clusters of grains as output.Te steps are
i) Find all the connected components in the binary image.
ii) Remove all components with pixel area less than a threshold.
iii) Each remaining component is a particle segment.
The above steps extract segments from the image. Most of the

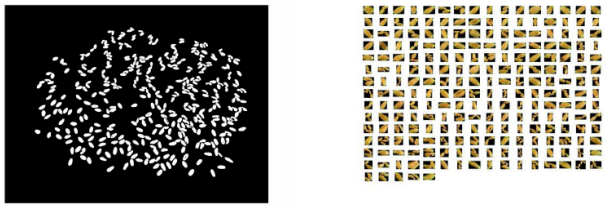

Fig. 4. Left one is the binary image and right one are the obtained segments.

segments have single grain, some have more than one grain. The segments are therefore sent to second level segmentation to separate each grain.

5.2. Second Level Segmentation

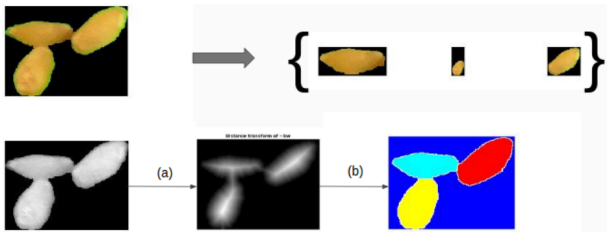

Fig. 5. Second Level Segmentation: Take the binary image corresponding to segment, take its Distance Transform(a), invert it then use watershed segmentation (b) after minima suppression.

The second level segmentation takes segments of grains as input and gives the separated image as output. We try two methods to further segment the clusters of grains into individual grains.

5.2.1. Watershed Transform and Segmentation

The term "Watershed" means the ridges that separate water flowing to different basin. In such scenario water in each basin travels downward towards it's local minima. A grayscale image can be thought of as a surface whose height at each point is proportional to the grayscale value at each point. The lighter pixels are near peaks while darker pixels are near catchment basin. To construct such type of surface, the distance transform of the image is found. The distance transform calculates distance of nearest pixel with non-zero value for each point. The distance transform is further inverted to construct catchment basin instead of peaks. We use matlab's inbuilt function that uses Fernand Meyer algorithm [1] to find watershed segmentation of the image. The steps of second level segmentation can be summarised as
i) Take the binary image of segment in question.

ii) Obtain its distance transform.
iii) Invert the Distance transform and remove unwanted minimas.
iv) Apply watershed segmentation.

5.2.2. Segmentation using EM Algorithm

Even though the previous method was able to further segment out touching grains to some extend but it was still giving incorrect results in many cases where we get more number of segments as expected due to occurence of more local minima in the image. Therefore we tried another method based on Expectation Maximization (EM) Algorithm for second level segmentation (Figure 6). In this method we try to cluster pixels by fitting Gaussian Mixture Model (GMM) on the result of the first level segments. In this way we are basically taking advantage of the ellipse like shape of the grains and try to fit ellipses to get the required segments. Steps corresponding to this process are summarized below:
i) Erode the binary image to separate islands in the image.
ii) Find connected components and Initialize the EM algorithm with the means of these components and total number of gaussians equal to the total number of the components.
iii) Perform EM algorithm on the non-eroded binary image to get means and covariances and indices corresponding to different grains.
The process of finding connected components may give false components or may not extract all the components, so in future we will experiment with different components finding algorithms like BIC, AIC etc.

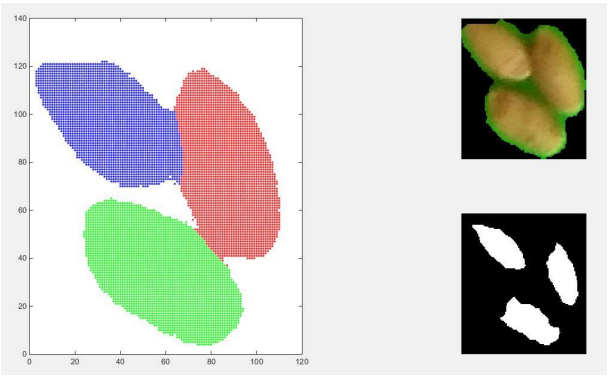

Fig. 6. Segmentation II using EM: The binary image is highly eroded, and no. of islands are set as components of GMM. Their means and variances are used to initialize the EM algorithm. After EM predictions are made on original binary to get segments.

6. FEATURE EXTRACTION

At first, we calculate the average pixel area A_{avg} of all the segments of a single image. This will be used to as a normalization parameter for scale invariance against the camera distance from grain-sheet. Each segment is taken and the following basic features are extracted from it:

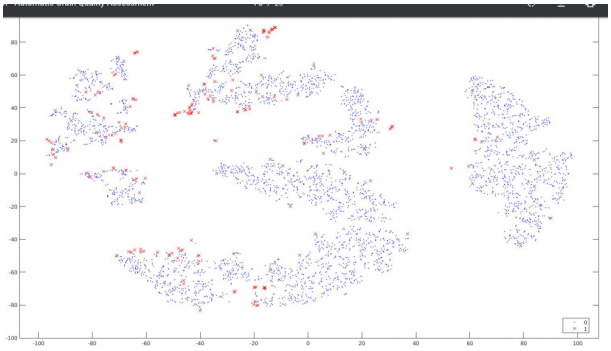

Fig. 7. TSne plot of the complete data in feature space. Red crosses are the impurities and blue dots are grains.

1. **color**: average pixel intensity for [R G B] colors
2. **size**: $\frac{A_{seg}}{A_{avg}}$, where A_seg is the pixel area of segment.
3. **axes**: $[\frac{\lambda_1}{A_{avg}}, \frac{\lambda_2}{A_{avg}}]$, the major axis length and minor axis length of the segment.
4. **eccentricity**: $\frac{\lambda_1}{\lambda_2}$

These seven features are concatenated and used for classification.

7. CLASSIFICATION

Best non-overlapping sample images are handpicked from the dataset, and segments from those are used for training our classification model. Features are extracted from these segments. We plotted the TSne plot of the feature space and found that the distribution of the impurities is highly non linear. Hence we use the following Non-linear classifiers:

- Support Vector Machine (**SVM**) with (rbf) Kernel
- K Nearest Neighbor (**KNN**) with nNeighbors $= 5$.
- Random Forest Classifier (**RF**) with no. of trees $= 50$.

Now, since we have 4038 segments of full grains and 350 segments of impurities [1], training a model on full data will make a skewed model more inclined towards 'grain' class than 'impurity' class. So we take a random sample of 350 points from full grains, and hence we have equal data from

both the classes. A 5-fold cross-validation is done on this data to obtain the validation accuracy along with average confusion matrix. The model is then tested on all the 4038 grains to see if it classifies all of them as 'grains'. All these results are reported in Table 1.

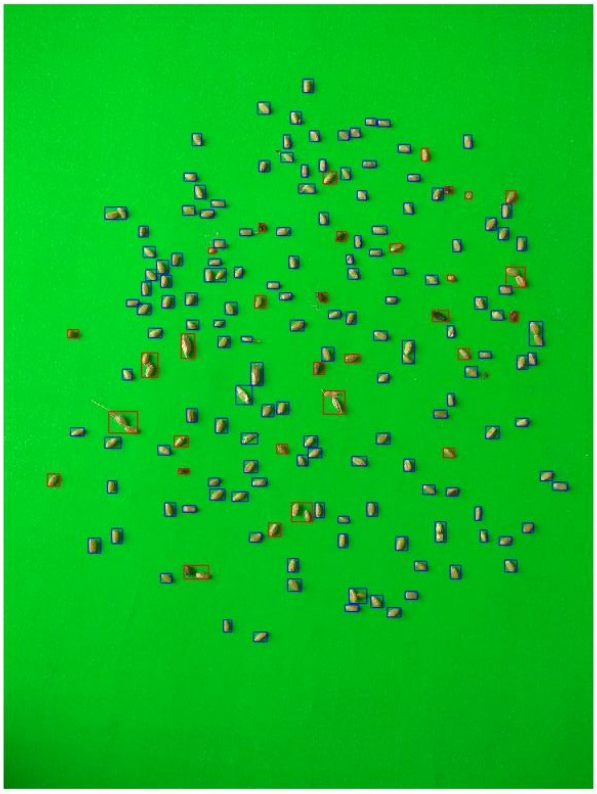

Fig. 8. Complete pipeline tested on a sample image. Particles enclosed in blue boxes are predicted as 'full grain' while those in red boxes as 'impurities'.

8. RESULTS

The classification accuracies of different models are reported in Table 1. We can see that all of them are performing similarly. The model is then tested on a sample image containing both grain and impurity. As we can see, almost all single particles are correctly classified. But the overlapping grains are being interpreted by the model as some combined shape and it call it 'impurity'. This problem would have been solved after we plugin our Level-2 Segmentation in the pipeline. We haven't incorporated it yet because it is not generalized enough to work for all the overlapping-grain cases. The purity of sample is calculated as ratio of sum of areas of grains to sum of areas of all the particles. Purity for this sample comes out to be 80.86%.

[1]Please note that these numbers are slighly different from those given in presentation, because earlier due to a bug we were picking only a subset of the segments. It has been corrected and the report contains the updated results.

Classifier	**Valid. Accu(%)**	**Avg. Confusion Matrix**	**All Grain Accu(%)**
SVM (rbf kernel)	87.29	g: 60.2, 9.8 / i: 8.0, 62.0	84.71
KNN (NN=5)	81.86	g: 63.2, 6.8 / i: 18.6, 51.4	90.40
RF (50 Trees)	88.43	g: 59.4, 10.6 / i: 5.6, 64.4	85.76

Table 1. Table containing the results for various Classification Models. In the Confusion Matrix 'g' represents 'grain' class and 'i' represents 'impurity'.

9. RELATED WORK

While we did implement everything from scratch, we don't claim anything done by us to be an entirely new innovation. [[2]] have tried impurity detection in a spread out wheat grain sample, but they distinguish between wheat and impurity using just an area threshold, below which everything is considered as impurity. No classification model is learnt. [[3]] use morphological operations for simulated, overlapping grains separation. [[4]] try to separate overlapping ellipsoid cells by fitting ellipse models to each of the cells.

10. CHALLENGES

Since we have been designing our pipeline from scratch, we had to work on each component and at the same time ensure the sync between them all. In this project we had two major objectives:

- To initiate each component, even if by using a simple model.
- To create the complete, flowing pipeline.

The initial objectives have been achieved, but a few challenges still remain to be overcome, like:

- Automating the fine-tuning of binarization threshold, which currently we do overselves for every different image due to illumination variations. Otsu's Method [[5]] is often used for dynamic global thresholding, but in our dataset it was failing in some images.

- Automating the Segmentation-II for overlapping grain segmentation and then plugging it into our pipeline.
- Using better feature representations for contour representation, such as mean distance from a fitted ellipse, etc.
- Improving the classification accuracy.
- Distinguishing broken grain from full grain.

11. ACKNOWLEDGEMENT

The project was granted to us by Mr. Gaurav Agrawal [6]. We had long discussions about pipeline and the method. He has been very patient with us, and has provided us with all the logistic/resource support we have asked. We also thank Kanpur Anaj Mandi and the staff there, who helped us in collecting the samples and separating the particles in each sample. We thank our mentor Prof. Tanaya Guha for her enthusiastic support and her guidance at each and every step. Last but not the least, we thank our TA Mr. Saurabh Kataria for encouraging us to take up this project and being always available whenever needed.

12. REFERENCES

[1] Fernand. Meyer, "Topographic distance and watershed lines.," Signal processing 38.1, 1994, pp. 113–125.

[2] Amandeep Singh. Harshwardhan Kakkar, Jaspreet Kaur, "Detection of good quality wheat grains using image processing.," ResearchCell: An International Journalof Engineering Sciences, 2016.

[3] AndrRS. Maral, "Alternative methods for counting overlapping grains in digital images.," International Conference on Image Analysis and Recognition. Springer Berlin Heidelberg, 2008, pp. 1051–1060.

[4] Wesley Nunes Gonalves and Odemir Martinez Bruno., "Automatic system for counting cells with elliptical shape.," arXiv preprint arXiv:1201.3109, 2012.

[5] Nobuyuki. Otsu, "A threshold selection method from gray-level histograms.," Automatica 11.285-296, 1975, pp. 23–27.11.

[6] Gaurav Agrawal, "Machine learning for grain assaying," pp. (https://github.com/dhishku/Machine–Learning–for–Grain–Assaying).