Annotation-free Quality Estimation of Food Grains using Deep Neural Network

Akankshya Kar*
ak.kar@samsung.com
Prakhar Kulshreshtha*
p.kulshresht@samsung.com
Ayush Agrawal*
ayu.agrawal@samsung.com
Sandeep Palakkal
sandeep.pal@samsung.com
Lokesh R. Boregowda
lokesh.rb@samsung.com

Samsung R&D Institute Bangalore
Bengaluru, India
* indicates equal contribution

Abstract

We propose a fast and accurate system for automatically estimating the quality of food grains on resource constrained portable devices using computer vision. We are motivated by an urgent need in India for grain quality estimation to ensure transparency in the agricultural supply chain and empower poor farmers to get the correct price for their crops. The system uses instance segmentation of touching grains, followed by classification of each grain according to E-NAM[1] parameters. To the best of our knowledge, this is the first attempt to use Deep Learning to estimate quality of cluttered sample of grains using only mobile phone. Samples are collected from various Agricultural Produce Market Committee (APMC) yards, which are used to generate synthetic data to simulate realistic clutter of grains for training our instance-segmentation network. Novel augmentation techniques while training make the system robust to illumination changes. Our system obtains the state-of-the-art performance and has been tested in various locations in India. At a mAP score of 0.74 and classification accuracy 92%, our system takes less than 100s as compared to 15 minutes of manual quality estimation.

1 Introduction

In a world driven by Artificial Intelligence (AI), large amount of work has been done to help in the field of agriculture. However, quality estimation of food grains using hand-held mobile device has remained an unexplored area. This is important because in developing nations like India, poor farmers do not have access to expensive machinery for grading their produce, hence a slight contamination in the produce results in heavy reductions of selling price. In this paper, we illustrate a computer vision based inexpensive mobile system for automatic quality estimation or assaying of food grains, which empowers farmers to get a fair

[1] Electronic National Agriculture Market (eNAM) is a scheme by Indian govt. aimed at linking all the APMC markets, and defining and enforcing pan-India quality parameters for various agricultural commodities [link]

price. We call this system CNN for Automatic Grains Assaying (**CNAGA**). CNAGA aims to counter the traditional subjective method of manual assaying of commodities. It brings trust and transparency in transactions and speeds up the manual process by 10x. CNAGA uses a single RGB image with On-Device AI to assay a sample of grains. The image is clicked after spreading a sample of grains on a green A4 sheet of paper. Green color has been chosen because particles like grains (full, broken, damaged, etc.), foreign matter, and other food grains appear distinct from this background color. The sheet is detected in the clicked image, and perspective of the sheet is corrected. Then CNAGA performs instance segmentation of the particles. Each extracted particle is classified into categories like full grain, damaged, weevilled, broken, immature/shrivelled, other food grains, inorganic foreign matter, organic foreign matter, etc. Hence the quality can be obtained in terms of percentages of various defects in the sample.

In this paper, techniques and practical steps taken to develop the system are described. An annotation free clutter segmentation network is proposed which utilises synthetic data generated from very few images of each category. A classifier is explained which is robust to light variation, as the application demands execution in varied environments. To benchmark this system, a hand annotated testing data for both segmentation and classifier consisting of 25 images and 1500 grains was created. The entire pipeline for data collection, labelling and generation is shown in Fig. 1 along with the inference pipeline. The paper follows the similar flow, Sec. 2 consists of the related study, Sec. 3 contains data collection strategy, Sec. 4 discusses the method of training the segmentation network, classification network and making the classifier robust to varying light conditions. Sec. 5 has the experimental results followed by Conclusion in Sec. 6.

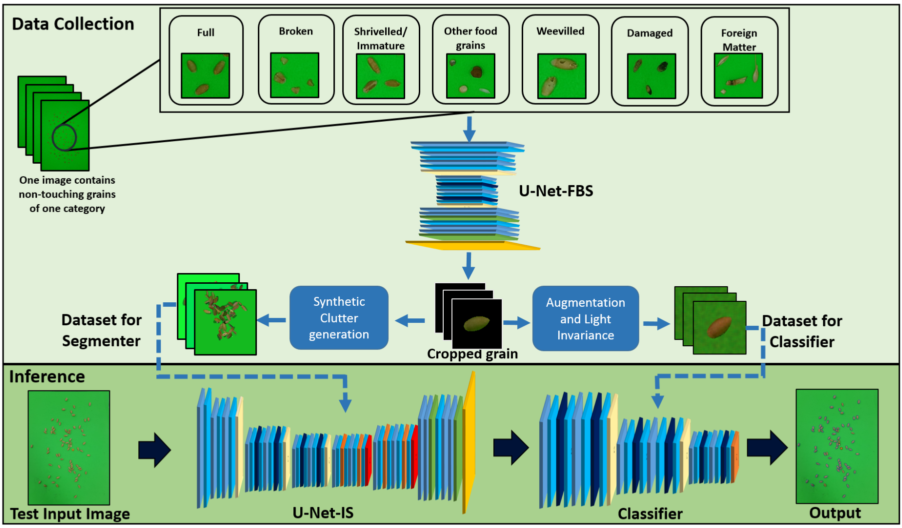

Figure 1: Overall pipeline of our approach. Images of samples of grains spread in a non-touching manner are collected, and then U-Net-FBS is used to crop out particles from the green background. These particles are used to train an instance segmentation network (U-Net-IS) and the MobileNetV2 classifier, both of which together form the inference engine.

2 Related Work

Instance Segmentation is the problem of segmenting out each instance of an object belonging to certain category. The two most popular networks for Instance Segmentation are U-Net [13] and the Mask-RCNN [5]. There is large amount of recent work which improves upon these models [3, 7, 10]. Despite the advances in the networks, a major problem in the Instance Segmentation, and in general in Semantic Segmentation is the lack of availability of annotated training images. While the training images can be increased via augmentations [12, 13], many other works have utilized synthetically placed copies of objects on a background to create synthetic training data [17, 18, 19]. Generative Adverserial Networks (GANs) also seem to be an attractive option, however this framework cannot generate images with pixel-level annotation [18]. [8] maybe used, but contains artifacts and random noise in generated images. It was observed that a spread out grain sample looks similar to touching cells in a microscope image, so cell population simulators [9, 16] could be used for generating the training data.

For the classification task, while there are a large number of CNN-based classifiers [4, 6, 21], MobileNetV2 [15] has become the top choice for on-device classifiers. This network is very fast for computation and has pre-trained models available. Transfer Learning [20] on this network is useful for many tasks. For the task of automatic grain assaying from images, the works include sorting of grains with specialized hardwares in a constraint environment [11] or cover only the case of non-touching particles [14]. To the best of our knowledge this is the pioneer work which uses a hand-held mobile device to assay wheat grains in an unconstrained manner.

3 Data Collection

Different samples of wheat grain were collected from various APMC yards in India. Three volunteers were trained and given different samples to manually separate each grain based on E-NAM parameters. For each sample separated, the other two volunteers cross-verified the correctness. Finally, all the grains were classified into 8 categories derived from E-NAM - Full, Broken, Shrivelled/Immature, Foreign Matter-Organic, Foreign Matter-Inorganic, Damaged, Weevilled, Other Grains. The cropped out grain images of these particles were required for two tasks. First, was to generate synthetic images for training the instance segmentation network(U-Net-IS) ,and second, to train the classification network. To get these images, for each category, around 50 grains were spread on a green A4 sheet such that none of the grains touched any other grain. The images were captured using a 12MP mobile phone camera. All particles were extracted from the background and assigned a single label, since they belonged to the same category. The sheet was defined as background, and particles as foreground. A mere foreground-background threshold value should have worked but effects like the illumination conditions, shadows of particles etc. did not allow simple threshold methods to get the whole particles. Hence, basic U-Net was trained for binary classification for each pixel: 0 for background, and 1 for foreground. The network was trained on 5k synthetic images of size 512x512pixels, with 30-50 randomly placed ellipses and rectangles of various colours, orientations and sizes with their shadows of various shades of green, augmented by Gaussian noise (see Fig. 2). This network is called U-Net for Foreground Background Separation (U-Net-FBS). Using this network, the particles were extracted from the background and labeled in one go. Since the images were not labelled manually, therefore the method is annotation-free.

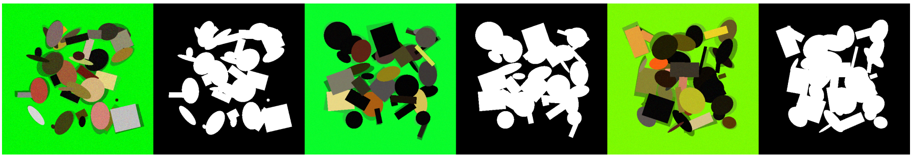

Figure 2: A sample of image-mask pairs from dataset used for training U-Net-FBS

4 Method

4.1 Perspective Correction

In order to take care of the varying poses with which a user can click the picture the quadrilateral boundary of the sheet was detected, using edge-detection followed by contour selection [1]. Then the sheet was cropped and corrected based on perspective into a rectangle. The aspect ratio of the image is kept same as that of the sheet (see Fig. 3).

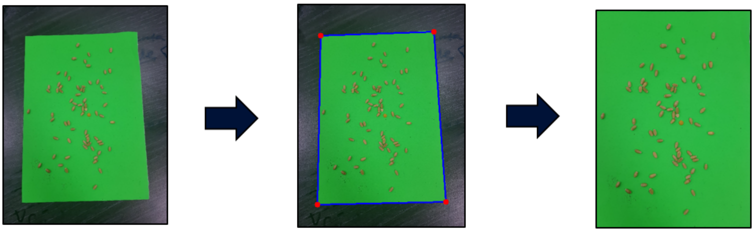

Figure 3: Steps of perspective correction

4.2 Instance Segmentation Network

After perspective correction of the image, each instance of the spread out grains was supposed to be segmented and sent to the classifier for categorization. The primary task was to ensure that each grain has its entire boundary. This was not a trivial task for cluttered grains, because the network needed to learn to predict the boundary along with whole particles accurately. The traditional method for training an instance segmentation network for this problem needs thousands of hand-annotated images of touching grains spread on a sheet. However, the background in this problem was almost uniform green (with slight lighting variations and shadows), hence copies of grains were placed to replicate the real images of spread out samples of grains. Three methods were experimented for placing copies of grains, as shown in Fig. 4. In the first one, copies of grains were placed randomly. In the second experiment, copies were placed randomly but it was ensured that the overlapping distance of a new particle was less than a desired threshold. This method was named **Random+OC** (Random with Overlap Constraints). The third method used was a cell-population simulator TRAGEN [16], as the spread out grain samples mimic the behaviour of cluttered cells in a microscope slide. However, TRAGEN algorithm was around 30x slower, because instead of directly placing the grains, it requires to first place the grains at random locations and then simulate the interactions of various forces between particles for around 300 iterations. In the later iterations the particles come closer, and create dense clutter (Fig. 4). A large amount of time was taken by TRAGEN to generate thousands of images. The developed algorithm Random+OC gave similar performance and was considerably faster than TRAGEN (Table 1). To maintain the sanctity of the inference environment, test data and validation data images were manually collected and annotated by five volunteers. There were 25 images and each image contained 200 to 500 grains evenly spread on a sheet, such that they may be touching each other but not occluding one another. Each image took 1.5 hours to annotate. Out of these, 4 were used for validation, and 21 were used for testing. The generated images were used to train the instance segmentation network. U-Net and Mask-RCNN were experimented as they are widely used for segmentation task. U-Net had better results than Mask-RCNN. Mask-RCNN took more time to compute on device and the model size was significantly larger. Therefore, preference was given to U-Net over Mask-RCNN and was selected for the segmentation network. The original U-Net was modified to have two decoders. One decoder terminates in a simple softmax mask for foreground-background separation, and the other one to identify the common boundary between touching particles. This mask was a binary softmax mask, 0 for background or grain, and 1 for the common boundary between touching grains, similar to [7]. The boundaries obtained were often thicker than the actual boundary hence boundary mask was skeletonized and then subtracted from the foreground mask. Sometimes the boundaries were not complete, so Watershed algorithm was applied on the distance transform of binary mask of each of the connected component, as shown in Fig 5. We call this entire engine **U-Net-IS**, or U-Net for Instance Segmentation.

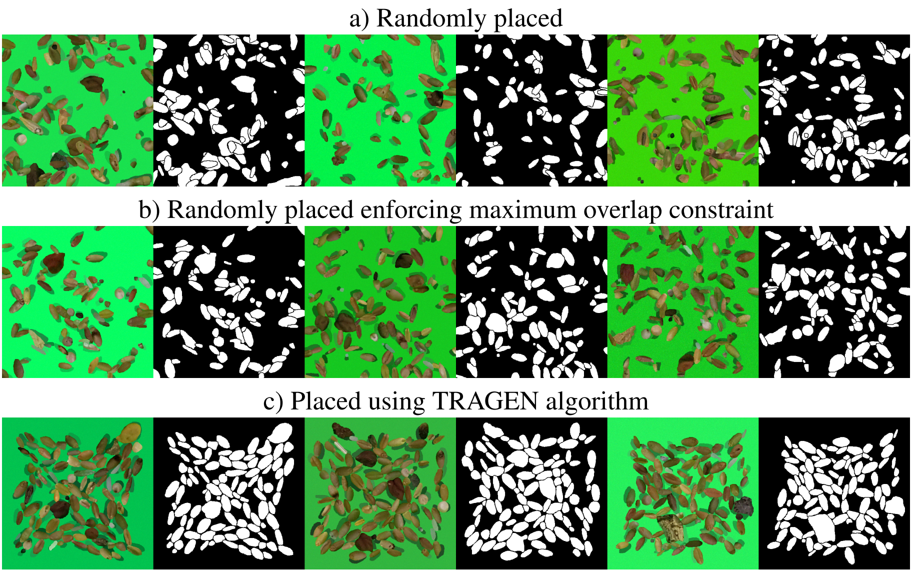

Figure 4: Samples of image-mask pairs generated from three different ways of placing copies of grains for generating data for instance segmentation

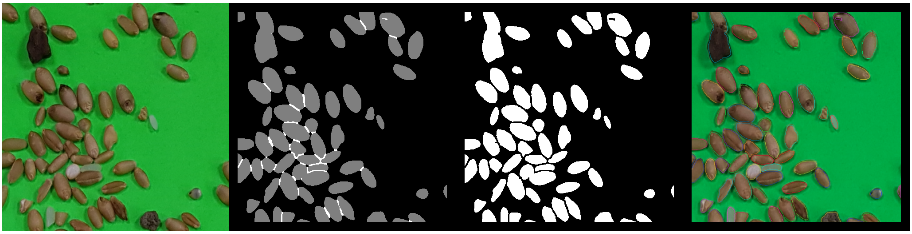

Figure 5: The steps in the instance segmentation engine. First image from left is a sample patch which is fed to U-Net-IS to get foreground-boundary-background mask shown in second image. Third image is the output after post processing steps described in Sec. 4.2, and the final instances detected after component-wise watershed transform are shown in fourth image

4.3 Classifier and Light Invariance

The task at hand had many categories which had minimal inter-class variation. For example, weevilled grain is a full grain with a hole, discoloured is a full grain with brownish-black colour on one side. In order to classify grains into E-NAM categories, MobileNetV2 was taken as it is the most preferred classifier for constrained environment computations. MobileNetV2 trained on ImageNet dataset was used as base network. Its first 37 layers were frozen, which generally learn the corners, edges, basic shapes etc. and the remaining layers were fine-tuned to learn domain specific features. In the trials, it proved to be fast and achieved good accuracy. For our first experiment, each segmented grain was taken, pasted on a black background and fed into the classifier which ensured that the grain was salient, and the network learned its features. This resulted in 89.95% accuracy when tested on normal light testset. However in the dim light setting, test accuracy dropped by 15-20%. Even after applying various augmentation techniques there was a difference of approx. 8% in test accuracies of normal and dim lighting conditions. The main reason for the drop was speculated to be the change in colour of the particles, which is one of the distinguishing features in our case. Dim lighting or very bright lighting can cause the colour spectrum of a category to overlap with other categories. For example, full grain in darker light, would look very similar to a discoloured grain in normal light (Fig. 6). Shadows also contributed to this problem, as one part of the sheet would be bright, but the other dull, as it would contain the user's or some near by object's shadow. One possible solution to this problem would be to take the background color into account. It can be observed from the first row of Fig. 6 that the color of the background sheet also changes with the changing lighting.

This observation inspired our approach for the second experiment in which the cropped grain was taken and its surrounding background was also extracted. The mean and variance of the background was calculated and used to generate a normalized green background for the grain instead of the black background. See Fig. 7 for details. This ensured that the network learned, for example, that in a dark background, full grains look darker and vice versa. This made the network robust to varied lighting conditions and shadows. The results are shown in the Table 2.

4.4 Mobile Application

An image of the green A4 sheet with the grains spread is captured, by holding the phone directly above the sheet. The user should try to keep it almost parallel to the surface. Per-

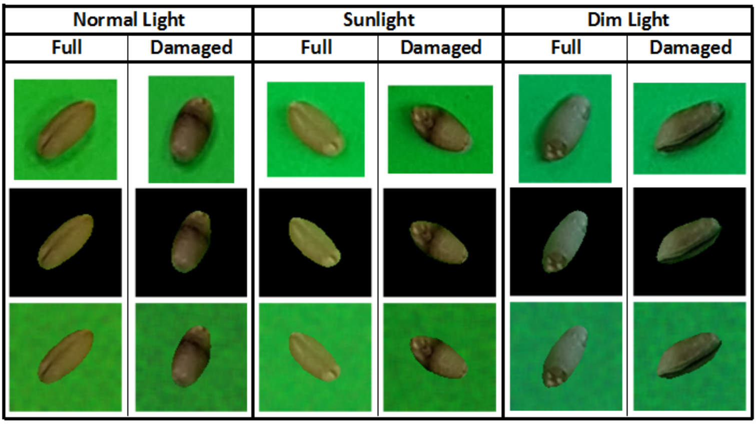

Figure 6: Full versus damaged grain in various light conditions. First row shows original cropped image of particle kept on the green sheet. Second row shows the 96x96 image obtained by pasting segmented out grain on a black background. Third row shows the grain pasted on algorithmically generated green background, as described in Sec. 4.3

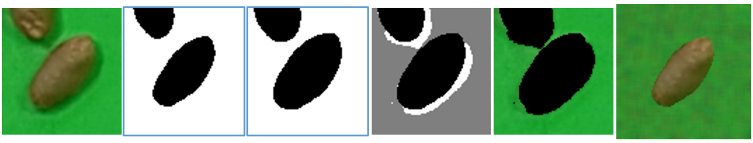

Figure 7: Steps involved in obtaining the green background image. From left to right, first image shows a patch of original image of a grain. Second image is the background obtained from U-Net-IS. Erosion is performed so that slight grain boundaries around segmented out crop is removed. Still the background may contain shadows, therefore Otsu's thresholding is applied on the background to get output as shown in fourth image (black- dilated particle mask, white- pixels below threshold, grey- pixels above threshold). Then mean and variance of pixels above threshold (shown in fifth image) is used to create a green background, in which the cropped out grain is pasted at its center, shown in final image.

spective transformation is applied on this image as discussed in Sec. 4.1 and then this image divided into patches of size 500x500. As all computations are on-device, the segmentation network is run on each patch to extract mask for each grain. Once all the masks are obtained, each mask is used to crop the corresponding grain from the actual image and this cropped grain image is pasted on the background, which is discussed in Sec. 4.3. The cropped images are then fed into the classification network to obtain their class label. The results are shown as absolute count and percentage by count for each category.

5 Experiments and Results

In this section, first the variants of generated data for training instance segmentation network are discussed. Then performance of classifier for different training strategies is discussed. Finally, end-to-end performance on 5 sample images is evaluated.

A sample of cropped images of grains are taken, as described in Sec. 3, and individually augmented via resizing, rotation, and color transformation. Shadow is simulated by placing a silhouette of the particle in its proximity (distance and size decided randomly via Normal Distribution). Color of the shadow is created by reducing the Saturation and Value component from the background and Gaussian noise is added to the final image.

Table 1 illustrates three different ways of placing the kernels, as described in Sec. 4. Our comparisons are based on the widely used mean average precision (mAP), which is the mean of AP, from IOU 0.5 to 0.95 in increments of 0.05, and average precision at 0.5 IOU threshold. Model sizes and the generation time per synthetic image were also compared.

In each case, 6000 images were generated for training and the network was trained for 2 epochs, using a batch size of 5. Timings were calculated for a single thread on an Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz. **Random+UNet** is the U-Net trained on randomly placed particles. This model had the lowest mAP and AP@0.5 amongst the three variants, but its per image generation time was lowest. **Random+OC+UNet** is the U-Net trained on particles placed randomly but with the overlap constraint. **TRAGEN+UNet**, which is the U-Net trained on images generated using TRAGEN[16] method, gives the highest mAP, but its generation time per image is around 31s which is around 19x slower than Random+OC, with only slight reduction in mAP. Hence, Random+OC was chosen for expanding to various other grains.

The U-Net was also compared with Mask RCNN,for which Matterport's implementation[2] was used. In our case, even though AP@0.5 is highest for MaskRCNN, its model size and overall mAP is inferior to the U-Net. Hence U-Net was chosen over Mask-RCNN.

Method	AP@0.5	mAP	Gen. time per image(s)	Model Size(MB)
Random+UNet	0.890	0.729	0.450	2.6
Random+OC+UNet (ours)	**0.899**	**0.740**	**1.680**	**2.6**
TRAGEN+UNet	0.901	0.754	31.432	2.6
Random+OC+MRCNN	0.952	0.684	1.680	179.2

Table 1: Comparison of different synthetic data generation methods against Instance Segmentation networks

For evaluating the classifier, three test sets were prepared, in three different lighting conditions: normal light, dim light, and the sunlight. Proportion of the particles belonging to different categories was kept same in all the three test cases. MobileNetV2 models trained on four different kinds of training dataset were evaluated and results are given in Table 2. Around 22k training images were collected in normal lighting (See Table 3). Model trained on images of grains with just rotation augmentation, placed on a black background, is denoted in the table as **MV2+Black**. Second model was trained on images of grains augmented using rotation, channel multiplication, additive Gaussian noise, blur, mask erosion and mask dilation. This network is denoted as **MV2+Black+Aug**. **MV2+Green** is the network trained on images of grain placed on green background, as described in Sec. 4. Finally, **MV2+Green+Aug** is the network trained on natural image patches which were augmented in a similar manner as for MV2+Black+Aug, followed by placing the particle on extracted green background. In this case, augmentations, which simulate the varying lighting conditions, affect both the background and foreground simultaneously, unlike MV2+Black+Aug where the augmentations only affected the particles since background was fixed to black. This resulted in the best performance across all testsets, as depicted in Table 2, which indicates that MV2+Green+Aug is the most robust model in varying lighting conditions.

Finally, end-to-end performance of CNAGA was tested on 5 separate sample images

	Normal Light	Sunlight	Dim Light
No of particles	5057	3806	3607
MV2+Black	89.95	87.94	60.52
MV2+Black+aug	90.55	88.81	83.50
MV2+Green	90.31	65.00	61.90
MV2+Green+Aug	**92.19**	**89.30**	**86.00**

Table 2: Accuracies of the MobileNetV2 models on three testsets with different lighting conditions. Although architecture for all cases is kept same, each model is trained on differently augmented training data

Image Count	Normal Light (Train)	Normal Light (Test)	Sunlight (Test)	Dim Light (Test)
Broken	5074	1016	812	769
damaged	1024	185	128	130
Foreign Matter (Inorganic)	496	42	31	30
Foreign Matter (Organic)	953	206	98	119
Full	5437	2032	1715	1538
Other Grain	5092	393	274	270
Shrivelled/Immature	3180	1016	660	663
Weevilled	882	167	88	88
Total	22138	5057	3806	3607

Table 3: Category-wise distribution of the collected grain images for training set in normal lighting conditions, and testset in normal lighting, sunlight and dim lighting

which were hand annotated for both instance boundaries and instance labels. The predicted masks and labels after inference were obtained. mAP would not have been a good metric since for the end-to-end task, count of particles is more important. Hence, category-wise predicted/actual count, along with intersection over union (IOU) is reported in Table 4.

S No.	total	B	D	FMI	FMO	F	OG	SI	W
1	614	0.557	0.418	0.000	0.500	0.681	0.801	0.502	0.131
		131/102	21/38	3/0	38/21	273/255	37/38	100/119	24/30
2	414	0.549	0.349	0.117	0.350	0.687	0.781	0.468	0.343
		100/66	12/19	4/2	23/10	174/174	24/30	67/89	9/20
3	406	0.587	0.557	0.000	0.748	0.816	0.904	0.530	0.415
		80/64	17/18	0/0	31/21	187/200	25/25	59/49	19/21
4	462	0.524	0.679	0.000	0.407	0.729	0.781	0.188	0.095
		80/70	27/19	0/1	23/7	203/281	41/37	82/27	8/6
5	428	0.513	0.538	0.549	0.378	0.675	0.833	0.531	0.237
		98/80	20/14	1/1	25/8	141/179	30/35	83/83	17/17

Table 4: IOU and the count of predicted/actual over different categories for sample test images

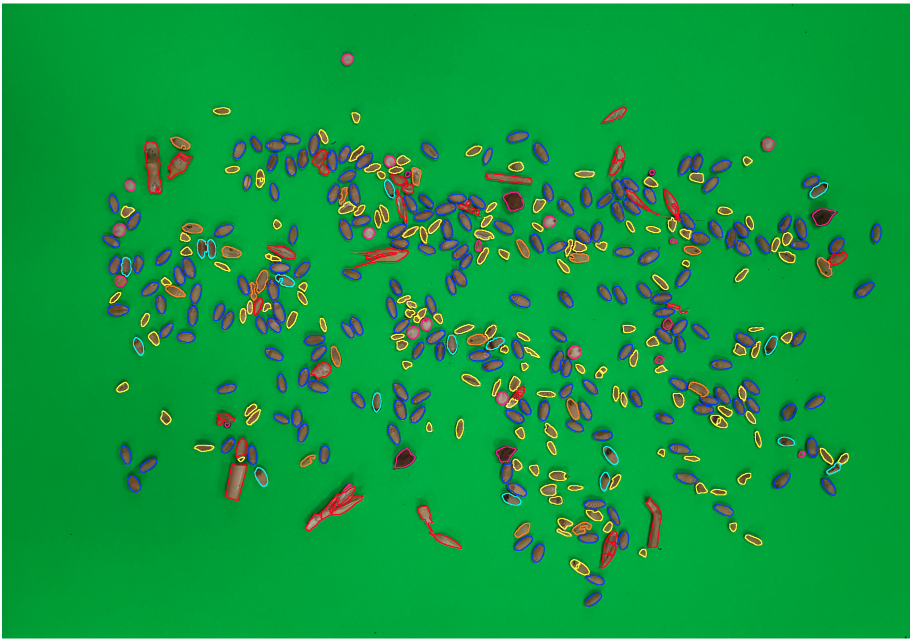

Figure 8: Output of CNAGA on a sample image. Color coding: Navy Blue - Full, Red - Foreign Matter(Organic/Inorganic), Orange - Weevilled, Yellow - Broken/Immature/Shrivelled, Magenta - Other Grains, Cyan - Damaged.

6 Conclusion

We have presented a simple and effective application of well-known CNNs (U-Net and MobileNetV2) for grain assaying. Since CNNs require large amount of labelled data and creating hand-annotated images is a cumbersome task, we proposed an annotation free method to generate thousands of labelled images for instance segmentation as well as for classification. Images of randomly placed copies of grains with maximum overlap constraint was effective for training the U-NET-IS. Further, in order to make the classification system robust to varying lighting conditions, intuitive and novel method was designed which takes brighter background pixels into account, before augmenting the training data for classification. In future, we would work to expand the commodities. We would also like to further explore methods to calculate weight of each grain, to accurately give percentage by weight of the defects, along with the count. We believe that this work has significance in future applications. We plan to collect more data in the future and release it for the research community. For this work we limit ourselves to one grain - Wheat, which is the most widely consumed food grain in North India. However our solution is easily extendable to other food grains. This application has been tested in the Agricultural Produce Market Committee (APMC) markets by government technicians and a beta version of the Android-app will be released soon.

References

[1] Pyimagesearch tutorial: How to build a mobile document scanner [url here].

[2] Waleed Abdulla. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. https://github.com/matterport/Mask_RCNN, 2017.

[3] Fidel A Guerrero-Pena, Pedro D Marrero Fernandez, Tsang Ing Ren, Mary Yui, Ellen Rothenberg, and Alexandre Cunha. Multiclass weighted loss for instance segmentation of cluttered cells. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 2451–2455. IEEE, 2018.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[5] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[6] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[7] Vladimir Iglovikov, Selim Seferbekov, Alexander Buslaev, and Alexey Shvets. Ternausnetv2: Fully convolutional network for instance segmentation. *arXiv preprint arXiv:1806.00844*, 2018.

[8] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arxiv*, 2016.

[9] Antti Lehmussola, Pekka Ruusuvuori, Jyrki Selinummi, Heikki Huttunen, and Olli Yli-Harja. Computational framework for simulating fluorescence microscope images with cell populations. *IEEE transactions on medical imaging*, 26(7):1010–1016, 2007.

[10] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8759–8768, 2018.

[11] Thomas Pearson. High-speed sorting of grains by color and surface texture. *Applied engineering in agriculture*, 26(3):499–505, 2010.

[12] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

[13] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[14] Kadir Sabanci, Abdurrahim Toktas, and Ahmet Kayabasi. Grain classifier with computer vision using adaptive neuro-fuzzy inference system. *Journal of the Science of Food and Agriculture*, 97(12):3994–4000, 2017.

[15] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[16] Vladimír Ulman, Zoltán Orémuš, and David Svoboda. Tragen: a tool for generation of synthetic time-lapse image sequences of living cells. In *International Conference on Image Analysis and Processing*, pages 623–634. Springer, 2015.

[17] Daniel Ward, Peyman Moghadam, and Nicolas Hudson. Deep leaf segmentation using synthetic data. *arXiv preprint arXiv:1807.10931*, 2018.

[18] Zheng Wu, Ruiheng Chang, Jiaxu Ma, Cewu Lu, and Chi-Keung Tang. Annotation-free and one-shot learning for instance segmentation of homogeneous object clusters. *arXiv preprint arXiv:1802.00383*, 2018.

[19] Wenqiang Xu, Yonglu Li, and Cewu Lu. Srda: Generating instance segmentation annotation via scanning, reasoning and domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 120–136, 2018.

[20] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3320–3328. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5347-how-transferable-are-features-in-deep-neural-networks.pdf.

[21] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018.