DEEP LEARNING BASED SEED QUALITY TESTER

Conference Paper · November 2019

6 authors, including:

Manish Bhurtel
Tribhuvan University

Jabeen Shrestha

Niten Lama
Purwanchal Campus

Sajan Bhattarai

DEEP LEARNING BASED SEED QUALITY TESTER

Manish Bhurtel[1], Jabeen Shrestha[2], Niten Lama[3], Sajan Bhattarai[4], Aashma Uprety[5], Manoj Kumar Guragain[6]

[1,2,3,4,5]Students of Department of Electronics and Computer Engineering, Institute of Engineering, Purwanchal Campus, Dharan Sub-Metropolitan City, Sunsari, Nepal

[1]manish.bhurtel09@gmail.com, [2]jabeenshrestha@gmail.com, [3]nitenlama46@gmail.com, [4]sajanbhatte@gmail.com, [5]aashmauprety99@gmail.com

[6]Lecturer at Department of Electronics and Computer Engineering, Institute of Engineering, Purwanchal Campus, Dharan Sub-Metropolitan City, Sunsari, Nepal

Email: [6]manojkguragai@ioepc.edu.np

Abstract

The paper presents a novel solution to the problem faced by the commercial farmers and seed packaging industries. It is very inconvenient to filter out every damaged seed and foreign elements by winnowing in industries and commercial farming. This issue can be minimized if the seeds are filtered in clusters. The paper presents an approach to enhance the efficiency in seed cultivation and seed packaging processes. We created a high-quality dataset which includes fine maize seeds, damaged maize seeds, and foreign elements. By using the Deep Learning technique, the system categorizes an input image as Excellent, Good, Average, Bad and Worst quality seed cluster. The Excellent and Good clusters (sometimes Average) can be cultivated or packaged, and the Bad and Worst clusters can be rejected. We also have recommended the use of object detection to detect and filter out damaged seeds and foreign elements from good quality seed clusters.

Keywords: *Seed Quality Tester, Deep Learning, Convolutional Neural Network, Optimizers, Dropout, Hidden Neuron*

1. Introduction

In the commercial seed cultivation sector, it is not feasible to filter out the individual damaged seeds and foreign elements in the case of the voluminous amount of seeds. Moreover, in the seed packaging industries, the seed filtering is done by the manual labor force which leads to poor quality seeds in the packets and waste of a large amount of fine quality seeds. In this context, efficient and automated seed testing [17] is the most important part of all other seeds technologies. Seed testing facilities need to evaluate tens of thousands of seed lots each year [17]. To make ease in all the sectors of seeds, there should be some efficiency and automation in the seed filtering and packaging sectors. All these problems motivated us to enhance efficiency and bring automation by the use of Deep Learning technique. This would help to cultivate the fine quality seeds and reject the bad quality seeds without much effort. On the other hand, packaging of only fine quality seeds in seed packaging industries can be ensured by the use of our system. In this regard, we came across various solutions but the deep learning technique was an ideal approach to implement the system efficiently. Furthermore, use of Artificial Intelligence and Machine Learning is quite rare in the context of agriculture and industry, especially in Nepal. We also aimed to synchronize the Artificial Intelligence, Agriculture and Seed Packaging Industry, which would be a novel solution in Nepal. Therefore, the purpose of the paper is to use the deep learning technique to generate a system that can enhance ease in the commercial seed cultivation and seed packaging industries. In this context, we are also motivated to carry out research on object detection in the future. By the use of object detection, we can detect few damaged seeds and foreign elements present in the cluster of good quality seeds, and detect the few fine quality seeds present in the bad quality cluster. In this way, we can easily separate the fine-quality cultivable seeds from the damaged seeds and foreign elements.

2. Literature Review

With the reference of the paper [16], we found the idea of processing the images of the seed which would also be highly practical in the context of Nepal. In this paper, Digital Image Processing and MATLAB were used for the processing of the image and classifying it as Bad Purity or Good Purity or Excellent Purity. But we felt that the cluster made for the image processing would not give accurate results as the seeds underneath would not be visible.

In the paper [3], they have used Computer Vision to detect the germination of Lettuce Seeds. They have created the image dataset by placing the seeds distinctly. They used the Edge Detection method and principle of counting to detect the germinated seeds. But still, the counting would not be feasible for implementing our concept.

Furthermore, in the paper [9], they have proposed the automation idea of seed testing using Image Analysis Software and Edge Detection but still, the dominating features in the images could not be extracted with just image analysis software.

With the paper published in 2018 [1], we were enlightened by the very convincing method of using deep learning technique in the agricultural purposes. They proposed a systematic approach of using a deep learning model for the prediction of a category of an image. Furthermore, with the reference of the handbook of Julianna Bányai [8], we came across the idea about creating a seed lot which could be a very novel concept and useful for the commercial farmers and seed packaging industries.

Deep Learning or Hierarchical Learning has been emerging as a new area of research on Machine Learning [2]. Deep learning discovers detailed structure in huge datasets by using the backpropagation [23] and updating the internal parameters that are used to predict certain categories.

With the amalgamation of all the concepts and techniques, we finally came across the idea of using Convolutional Neural Network (CNN) for detecting the quality of the image Seed Lot (16 seeds) as Excellent, Good, Average, Bad and Worst quality. CNN is an extension of the Artificial Neural Network (ANN) that is comprised of neurons that optimize itself through continuous learning [13]. In the paper [13], CNN composed of 3 layers viz. convolution layers, pooling layers and fully connected layers. They implemented it for the MNIST dataset for detecting the hand-written digits.

We were also inclined towards the concept cited in the paper [21] which compared the different results obtained from Transfer Learning of ResNet, Inception v3, Xception and VGG 19 [21] but we believed that it would be effective to adjust the hyperparameters and design the CNN architecture on our own as per our dataset and requirements.

3. Methodology

The paper proposes the system to detect the quality of a seed lot using Convolutional Neural Network (CNN). The system has been developed in four stages: Dataset Preparation, Image Pre - processing, Building CNN, and Compiling and Training CNN.

3.1 Dataset Preparation

To generate our dataset, we used the maize variety, Super 900 M - F1, which is largely available at the Eastern Terai Region of Nepal. We included fine seeds, damaged seeds and foreign elements in the dataset and categorized the seed lot in percentage basis (Table 1) as Excellent, Good, Average, Bad and Worst as in figure 1. First, we clicked the images of the seed lot with a normal camera with not so good resolution. But the detailing that the system require to distinguish between the fine seed and damaged seeds, were not good enough which could make our system vulnerable to inaccurate predictions. So we again clicked 3000 images with high-quality camera and generated high-quality dataset with sharp detailing. We tried to accommodate all the possible orientations of the seeds in the seed lot so that the model learns all the different positions of the seeds in the cluster. Finally, we separated the dataset as 2500 training data and 500 testing data belonging to five categories as in Table 1.


(a) Excellent Quality


(b) Good Quality


(c) Average Quality


(d) Bad Quality


(e) Worst Quality

Fig 1: Different Qualities of the Seed Lots based on the percentage of Fine Quality Seeds

Categories (Seed Lot)	**Fine Seeds (%)**	**Damaged Seeds or Foreign Elements (%)**
Excellent Quality	100	0
Good Quality	61 – 99	1 – 39
Average Quality	40 – 60	40 – 60
Bad Quality	1 – 39	61 – 99
Worst Quality	0	100

Table 1: Percentage of fine seeds, and damaged seeds or foreign elements in different seed lots

3.2 Image Pre-processing

The dataset should be processed before feeding it into the neural network. We augmented the dataset using the library function provided by Keras and processed all the images to uniform 256 *256 resolution.

Increasing the input shape beyond this resolution could give more accuracy but slows down the speed. This resolution keeps the distinguishing features of the damaged seeds which makes the machine easy to learn. The adjusted parameters are rescale, shear, zoom, width and height shifting, rotation, horizontal and vertical flip. After the dataset preparation, all the images are of uniform attributes and are ready to be fed into the Convolutional Neural Network.

3.3 Building Convolutional Neural Network (CNN)

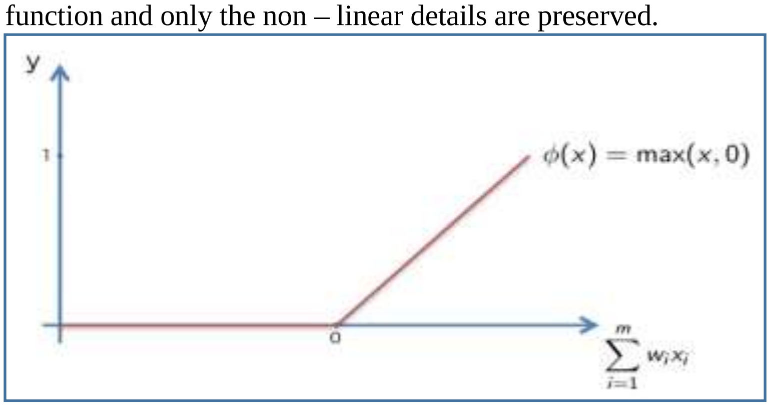
Building a neural network was a great challenge because we need to make sure that in any of the layers, the detailing of the damaged and fine seeds is not lost. So we built our CNN keeping the high resolution for the input images.

3.3.1 Convolution and Pooling Layer

We used three convolution layers, each layer followed by a max-pooling Layer. In the convolution layer, feature detectors slide thoroughly and rapidly in the image searching for certain features within the image. The pooling function reduces the resolution of the feature maps to achieve spatial invariance [7]. We used the max-pooling function to extract the dominating or maximum features from the feature map. We used 64, 128 and 256 feature detectors in the corresponding three convolution layers and initialized the convolution layer to accept the image of 256 * 256 resolution and depth of 3 RGB color channels. The feature detector of the 3 * 3 matrix slides over the image and searches extensively for the features in the image. In max-pooling layers, the matrix of 2 * 2 pool size rolls over the feature map with a stride of 2 and collects the maximum values which are the respective dominating features in the image.

ReLU Activation

Rectified Linear Unit (ReLU) rectifies the linearities in the non – linear images. ReLU boosts up the training process [12] so as to converge to accurate predictions. The graphical representation of ReLU in figure 2 demonstrates how the linearity is removed from the image. The negative values are pruned by the rectifier function and only the non – linear details are preserved.



src: https://www.superdatascience.com

Fig 2: Rectified Linear Unit (ReLU) Activation Function

3.3.2 Dropout Layer

The dataset we created were high in quality but low in quantity. In the initial training procedure, our system was vulnerable to a high degree of overfitting. The input data must pass through a number of hidden layers which could cause the model to learn complicated relationships resulting in sampling noise and ultimately overfitting problem [19]. We overcome the issue by placing a dropout layer before the hidden layer. The dropout layer drops different units randomly from the neural architecture preventing the units from excessive co-adaptation [19]. With the placement of a dropout layer, there was no overfitting and the model predicted accurately.

3.3.3 Flattening Layer

The images after penetrating the convolution and max-pooling layers are in a two-dimensional pooled feature map. But the images should be fed into the series of artificial neurons which takes the input as a one-dimensional single vector representing the dominating features of the input image. This is achieved by performing the flattening operation to all the pooled feature maps. Finally, we have the flattened array of the input images ready to be fed into the fully connected layer.

3.3.4 Full Connection

The full connection is the addition of a general artificial neural network (ANN) with a fully connected hidden layer. With the reference of paper [10] and the number of experiments with hidden neurons, we assigned one fully connected layer with 512 hidden neurons for our system. The flattened data is passed through the number of hidden neurons and the predictive probability is passed to the final output layer. The output layer is assigned a softmax function whose output range is between 0 and 1 [4]. The softmax function designates the input image into five different output classes in our system. The softmax function with any input 'x' is calculated by using the following mathematical formula [6]:

$$f\,(x_i)= \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$

3.4 Compiling and Training CNN The compilation of the CNN architecture is done designating the following parameters:

Loss Function

In every iteration of the training and validation process, there arises a certain loss. The loss cannot be ignored as it is an important entity to determine the inconsistency between the predicted value and actual value. Since have used categorical cross-entropy function to calculate the errors and loss in the training of our multi-class classification approach. The error is calculated and then propagated backward from the network which updates the previous weights. The optimizer grabs a learning rate, propagates the error and updates the weights in each neuron. This helps to increase the accuracy of our training model. With the reference of the paper [20], to classify γ - dimensional input x_i to one of the C categories, then the Categorical Cross-Entropy (CCE) loss can be defined as [20]:

$$E_{CC} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}(p_{ic}\log(y_{ic})),$$

Where N is the number of training data, y is the output and p_{ic} is the binary indicator function that validates the accuracy of ith training pattern to cth category.

Optimizer

We experimented with our system on adam, adagrad and adadelta optimizers. Adam optimizer had big time complexity in our system. Furthermore, in adagrad [11], continuous decay of the learning rates throughout the training process was a major problem. Adadelta adapts the learning rate dynamically using only the first order information [24]. So we used adadelta optimizer which is the adaptive learning rate method [24] surpassing adam and adagrad optimizers.

Evaluation Metric

The training of the neural network should be evaluated with a certain metric to improve the system performance. We have chosen accuracy metric to evaluate our CNN model. After running for 100 epochs, the model finally ensued 81% training accuracy and 80% testing accuracy.

4. Experiments and Results

We experimented with our model by using different numbers of hidden nodes and dropout layers with different probabilities (p).

4.1 Experiment with different number of hidden nodes

The number of hidden nodes is to be computed from the experiments. We took 5 sets of hidden nodes with 64, 128, 256, 512 and 1024 nodes, running for 100 epochs. The accuracy increased with an increase in the number of hidden nodes. There was no huge difference in using 512 and 1024 number nodes, so we decided to take 512 number of hidden nodes for our convolutional neural network design. The results are demonstrated in the following chart 1.

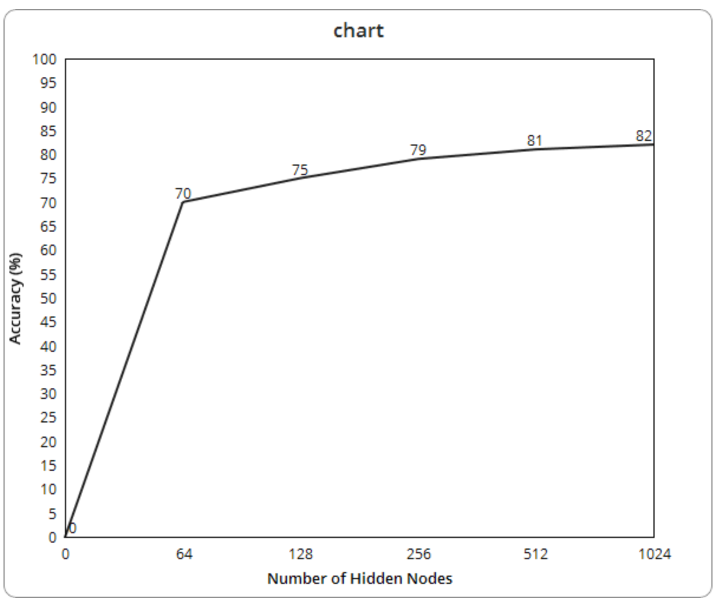


Chart 1: Change in Accuracy with increasing number of hidden nodes

4.2 Experiment with addition of dropout layer

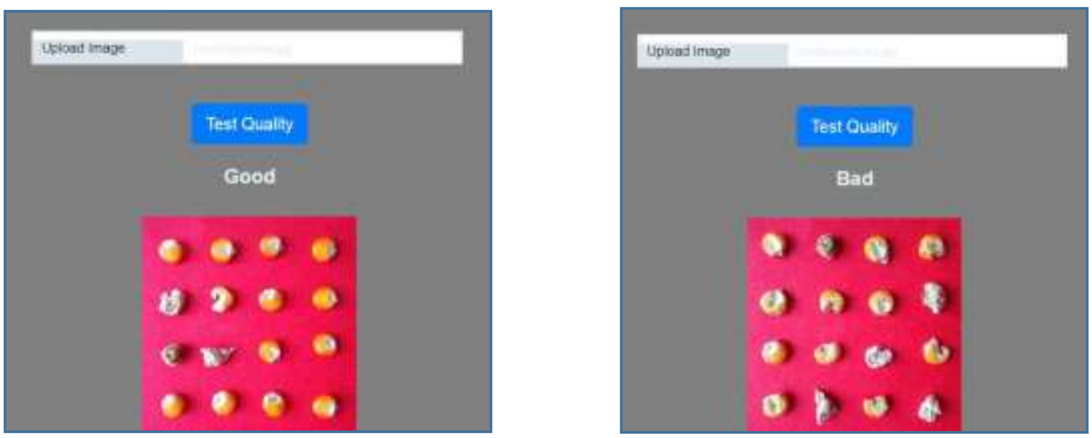
First, we tested the model without any dropout layer. We came across the issue of overfitting which must be addressed very carefully. Then we used the dropout layer before the hidden layer and gradually increased the probability ranging from 0.1 to 0.5. We got varying results which finally overcome the issue of overfitting and underfitting. The results are demonstrated in the following Table 2.

	Training Accuracy (%)	Testing Accuracy (%)	Remarks
No dropout layer	77	69	Overfitting
Dropout(0.1)	76	70	Overfitting
Dropout(0.2)	76	74	Decrease in Overfitting
Dropout(0.3)	77	75	Slight Overfitting
Dropout(0.4)	78	80	Slight Underfitting
Dropout(0.5)	81	80	Almost perfect Fit

Table 2: Use of Dropout Layer to overcome overfitting and underfitting

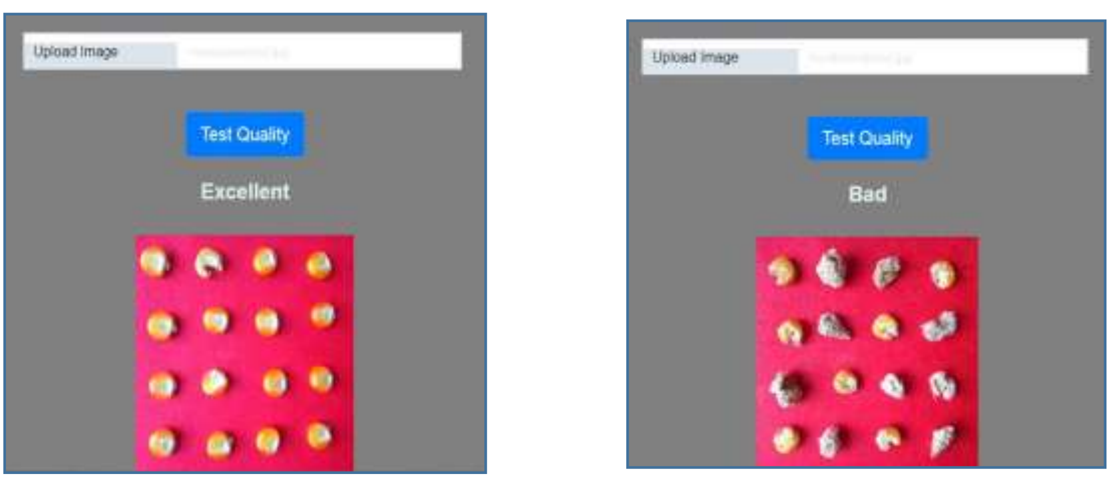
4.3 Output

After the model generation, we tested our system for the 20 test images in GUI developed using Django Framework. In the scenario, class prediction for 15 images were correct (figure 3) while the class prediction for 5 images were slightly different (figure 4) i.e. the CNN model predicted the category one step up or down.



(a) Model predicting Good accurately (b) Model predicting Bad accurately

Fig 3: Model showing accurate predictions



(c) Model predicting Good Quality as Excellent (d) Model predicting Worst Quality as Bad

Fig 4: Model predicting one – step up category

5. Conclusion

This paper presents an implementation of a seed quality testing system to detect the quality of a seed lot as excellent, good, average, bad and worst on the basis of the percentage of fine quality seeds in the seed lot. The decision of categorization is based on the flow of the dataset through the various neurons in the convolutional neural network (CNN). The summary of the CNN architecture used in our system is explained in the following table 3.

Keywords	**Details**	**Remarks**
Number of Convolution Layers	3	Feature Detectors: 64, 128, 256; Input Shape: 256 * 256; Activation: ReLU
Number of Pooling Layers	3	Use of Max – Pooling
Number of Hidden Layer	1	
Number of Hidden Neurons	512	Selected with multiple experiments
Dropout layer	p=0.5	Before the Hidden Layer
Optimizer	Adadelta	
Loss Function	Cross Entropy	Categorical
Performance Metric	Accuracy	81% training accuracy; 80% testing accuracy
Output Classes	5	Excellent, Good, Average, Bad, Worst; Use of Softmax Function

Table 3: Summary of the CNN Architecture in our system

With extensive research and experiments, we came to the conclusion that the model with small number of dataset can predict accurately if the images in the dataset are of high quality and all the hyperparameters like number of convolution layers, number of hidden neurons, use of dropout layers, etc. are properly adjusted as per the requirement of the system. In the initial phase, we did not have good accuracy, then we adjusted the hyperparameters but still, we were facing the problem of overfitting and underfitting. The problems were overcome by the use of dropout layers.

6. Suggestions and Recommendations

The final model of our system still ensued some inaccurate results. Those inaccuracies can also still be improved by adjusting the following parameters:

- Increasing the input shape in the convolution layer beyond 256 * 256.
- Training for higher number of epochs i.e. beyond 100.

We recommend the following future research activities to direct our project:

- Implementation using the emerging Rectified Adam optimizer (RAdam).
- Implementation and coverage of a wide variety of seeds.
- Implementation of Object Detection to detect damaged seeds and foreign elements in good quality seed lot, and fine seeds in the bad quality seed lot.
- Research to encompass different other criteria for seed quality detection like temperature, humidity, soil type and germinating nature of seed [18].

7. Acknowledgements

We would like to express our deep gratitude to our project supervisor Mr. Manoj Kumar Guragain for his relentless guidance and encouragement during our research project. We would also like to thank our friends Ujjawal Poudel and Saroj Raj Sharma who helped us with our research project without a frown.

We would also like to extend our thanks to Mr. Heros Silva Araujo, who is working as a Data Scientist at Gupy, Sao Paulo, Brazil, for his solicitous assistance and positive encouragement.

8. References

1. Andreas Kamilaris, F. X.-B. (2018). Deep Learning in Agriculture: A Survey. *Computers and Electronics in Agriculture, 147*(1). doi: 10.1016/j.compag.2018.02.016
2. Bengio, Y. (2009, January). Learning Deep Architectures for AI. *Foundations and Trends® in Machine Learning, 2*(1), 1-127 .
3. Chao Li, A. R. (2012). Application of Computer Vision for Lettuce Seeds Germination Detection.
4. Chigozie Enyinna Nwankpa, W. I. (2018). Activation Functions: Comparison of Trends in Practice and Research for Deep Learning. *ArXiv*.
5. Chollet, F. (2017). *Deep Learning with Python.* Manning Publications.
6. Courville, I. G. (2016). *Deep Learning.* MIT Press. Retrieved from http://www.deeplearningbook.org
7. Dominik Scherer, A. M. (2010). Evaluation of pooling operations in convolutional architectures for object recognition. *ICANN'10 Proceedings of the 20th international conference on Artificial neural networks: Part III* (pp. 92-101). Thessaloniki, Greece: Springer-Verlag Berlin, Heidelberg.
8. Dr. Julianna Bányai, D. J. (2002). *Handbook On Statistics In Seed Testing.*
9. Hemender, S. S. (2018, April). Image Analysis: A Modern Approach to Seed Quality Testing. *Current Journal of Applied Science and Technology*. doi:10.9734/CJAST/2018/40945
10. Jha, G. (2019). Artificial Neural Networks and Its Applications.
11. John Duchi, E. H. (2011). Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. (T. Zhang, Ed.) *Journal of Machine Learning Research, 12*, 2121-2159.
12. Kaiming He, X. Z. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *2015 IEEE International Conference on Computer Vision (ICCV)*, 1026-1034.
13. Keiron Teilo O'Shea, R. N. (2015). An Introduction to Convolutional Neural Networks. *ArXiv e-prints*.
14. Kuo, C.-C. J. (2016). Understanding Convolutional Neural Networks with A Mathematical Model. *Journal of Visual Communication and Image Representation, 41*, 406 - 413. doi:DOI: 10.1016/j.jvcir.2016.11.003
15. Li Deng, D. Y. (2014, June 30). Deep Learning: Methods and Applications. *Foundations and Trends® in Signal Processing, 7*(3-4), 197 - 387. doi:10.1561/2000000039

16. Ms. Mrinal Sawarkar, D. S. (2017 , May). Digital Image Processing Applied to Seed Purity Test . *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering, 5*(5).

17. Ms. Prajakta Pradip Belsare, P. M. (2013, February). Application Of Image Processing For Seed Quality Assessment: A Survey. *International Journal of Engineering Research & Technology (IJERT), 2*(2).

18. Nathaniel Narra, P. T. (2018). Deep Learning in Agriculture – what's happening.

19. Nitish Srivastava, G. H. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research, 15*, 1929-1958. Retrieved from http://jmlr.org/papers/v15/srivastava14a.html

20. Rusiecki, A. (2019, March 21). Trimmed categorical cross-entropy for deep learning with label noise. *Electronics Letters, 55*(6), 319 – 320. doi:10.1049/el.2018.7980

21. Samer Alashhab, A.-J. G. (2019). Hand Gesture Detection with Convolutional Neural Network. *Distributed Computing and Artificial Intelligence, 15th International Conference.* Springer International Publishing.

22. Y. Lecun, L. B. (1998, Nov). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*, 2278-2324. doi:10.1109/5.726791

23. Yann LeCun, Y. B. (2015, 5). Deep learning. *Nature, 521*(7553). doi:10.1038/nature14539

24. Zeiler, M. D. (2012). ADADELTA: An Adaptive Learning Rate Method. *ArXiv, abs/1212.5701*.