CONVOLUTIONALNEURALNETWORKWITHTRANSFER LEARNINGFORRICETYPECLASSIFICATION

Vaibhav Amit Patel and Manjunath V. Joshi

Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar;

201401222@daiict.ac.in

mv joshi@daiict.ac.in

ABSTRACT

Presently, rice type is identified manually by humans, which is time consuming and error prone. Therefore, there is a need to do this by machine which makes it faster with greater accuracy. This paper proposes a deep learning-based method for classification of rice types. We propose two methods to classify the rice types. In the first method, we train a deep convolutional neural network (CNN) using the given segmented rice images. In the second method, we train a combination of a pretrained VGG16 network and the proposed method, while using transfer learning in which the weights of a pretrained network are used to achieve better accuracy. Our approach can also be used for classification of rice grain as broken or fine. We train a 5-class model for classifying rice types using 4000 training images and another 2-class model for the classification of broken and normal rice using 1600 training images. We observe that despite having distinct rice images, our architecture, pretrained on ImageNet data boosts classification accuracy significantly.

Keywords: Convolutional Neural Network, Computer Vision, Transfer Learning.

1. INTRODUCTION

Almost all the states in India produce rice and it is a very important food for Indian population. In the recent years, the demand for better quality of rice has been increasing. Manual classification of rice is neither practical nor economically feasible. Machine learning techniques are simple and fast in performing classification task even though we cannot expect the perfection of a human hand. These approaches are being widely used in many different areas such as agriculture and biological research areas, since they are easy to train and are cost effective.

Rice types can be categorized based on their lengths, edge details, color, shape, etc. There are majorly two types of problems prevalent in the rice grain markets. First being the traders adulterating a particular type of rice with poor quality type and the second being the mix may include broken rice, stones, damaged seeds, etc. Looking at these, we propose to solve the problem of classification of rice types and also classification of broken and normal rice. It may be of interest to note that distinguishing a rice grain from a damaged seed or a stone is relatively an easy task, but distinguishing two rice grain types is a hard task especially if these types come from a single rice group with multiple types such as Basmati, Jaimati.

In the past, researchers have attempted to solve the rice classification problem. Haung et al. [1] proposed a computer vision-based approach for classifying six rice types by using twenty-one features of the rice grains. They include seven features based on color and fourteen morphological features. They used neural networks for classification and achieved an accuracy of 88.3%. Kaur et al. [2] used multi-class support vector machine (SVM) for classifying the rice as premium, grade A, grade B, grade C and they achieved an accuracy of 86%. In [3], Guzman et al. applied neural networks to 4 groups of total 52 different types of Philippine rice grains and achieved an average accuracy of 70%, while classifying the rice group from the images of rice. Here, the individual rice grain images were obtained by placing the rice grains one by one on a scanner which is a cumbersome task. Liu et al. [4] proposed to use color and morphological features to classify six rice types once again by using neural networks. Their model achieved an average accuracy of 84.8%. In [5], Silva et al. classified 9 different rice types using thirteen morphological, fifteen texture and six color features and achieved 92% accuracy.

In our work, we classify types of Basmati rice using their images captured by a scanner. We obtained the data for our study from a company named INWEON [6]. The segmentation was carried out by placing a handful of rice grains on the scanner. To the best of our knowledge, the other researchers have not attempted classification of a specific kind of rice group e.g. Basmati, Jasmine, etc., rather they have worked classification of general rice groups, which is a simpler problem in comparison. Such rice grains have different sizes, colors and shapes making the classification task easier. However, if we consider Basmati, the problem becomes much harder (as described in the experimental results section), due to

similarity of rice grains in color, size and shape. Moreover, the methods used in the current literature do not use deep learning-based approaches to solve the problem. We propose to use deep convolutional networks with transfer learning for classifying Basmati rice types. We make use of deep convolutional neural network in our approach, which has shown much better classification accuracy in recent times [7] [8].

2. CONVOLUTIONAL NEURAL NETWORK AND TRANSFER LEARNING

Machine learning methods require careful feature engineering. Deep learning is a type of representation learning and generally works well for natural data [9]. Exploiting the computational power of Graphics Processing Units (GPUs), deep learning-based approaches have performed exceptionally well [10] in solving complex problems. In general, the input features to a neural network are considered to be independent of each other while solving many of the classification problems. However, when the input is an image, one has to exploit the spatial structure of the image for better classification and also for solving many of the regression problems. Use of convolutional neural network serves as a useful architecture for classification of images data, since it exploits spatial dependency of pixels while training. Researchers have explored the idea of using CNNs in handwritten digit recognition problem [11–14]. But CNNs became popular only after Alex et al. [7] proposed a CNN architecture that won 2012 ImageNet Large-Scale Visual Recognition Challenge (ILSVRC). Here, they used 2 GPUs in parallel for training the network. Training a CNN is computationally less expensive since it involves parameter sharing. A typical CNN consists of convolutional and pooling layers, non-linear activations, and fully connected layers. A convolutional layer finds the presence of a specific shape such as the edge and the pooling layer down samples the image which is necessary to reduce the number of parameters. Fully connected layers which simply represent the multi-layer perceptron are generally appended at the end. Non-linear activation functions are necessary to learn complex functions.

In practice, as a general rule, convolutional neural networks work better, if we have significant amount of data to train. Hence, if the training is done with limited dataset the architecture tries to remember the dataset itself. This is referred as over-fitting, in which the architecture does a good job in classifying the training data but fails to generalize. In [15], Srivastava et al. explain the problem of over-fitting in CNNs. In our work, we have a total of 5000 images that include training and testing images. Hence, with such a limited dataset, the network tends to over-fit and this motivates us to use transfer learning to overcome the problem of over-fitting.

In practice, deep learning architectures need significant amount of data for better results and the training and testing data should come from the same distribution. To this end, when we have a limited data for training and testing one can transfer the weights already learned from another architecture trained using a large dataset to the problem at hand and expect better results. This is known as transfer learning and we use it in our work. Oqubay. [16] demonstrated how weights trained on millions of images can be used while training using a small dataset. Sinno et al. [17] give a detailed survey of transfer learning and explain that one can use already learned features of one dataset to perform training on another dataset. If one does the transfer of weights in a proper way, we can get higher classification accuracy than the model that is trained using a limited dataset. In our work, we use transfer learning because of the availability of limited dataset.

The Image-net dataset [18] consists of a very large number of natural images for training with 1000 classes. Each class itself has 1000 training images giving a total of 1 million training images. A convolutional neural network trained on these images is very likely to capture required features of the natural images in its initial layers. As any convolutional neural network learns the content of the image layer by layer (lines, shapes etc.) [19], if we transfer these features of the initial layers to a network using limited data as input while training, we can detect the edges, shapes and other features better when compared to that when trained without using transfer learning.

3. PROPOSED APPROACH FOR CLASSIFICATION

The images of the rice grains were captured by placing them on a scanner. A highly accurate labeled data was created using classification by human experts and good quality rice grains. In this study, we use five different types of Basmati rice and classify them into five different classes. The segmentation was done by an industry called GRAMS [6] and our approach
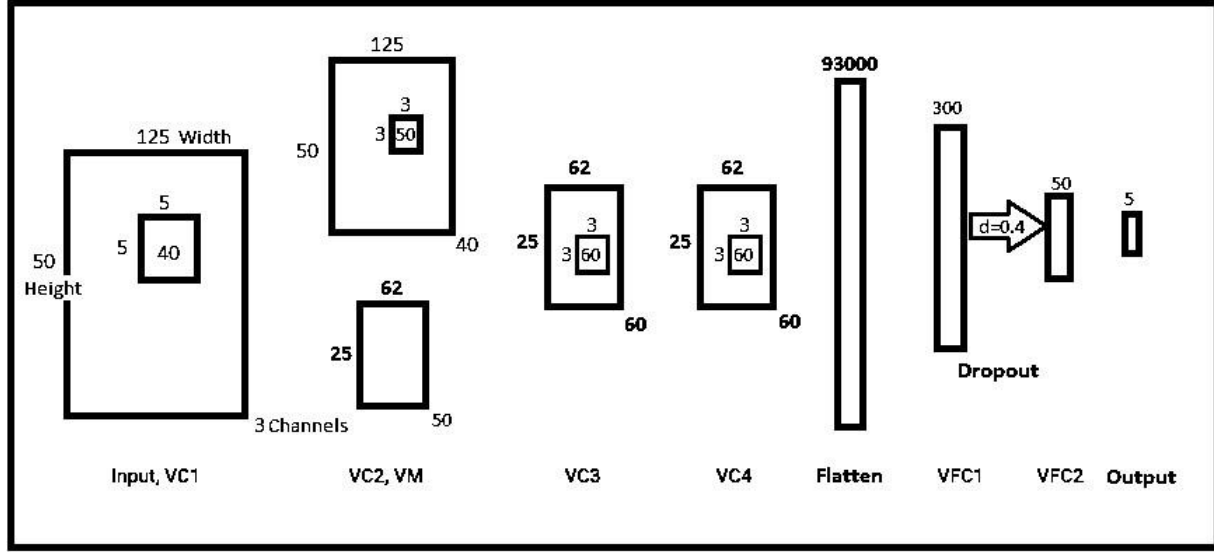
Figure 1: Proposed CNN architecture which is appended to the VGG16's architecture.

uses these segmented images for classification. We would like to mention here that, once we obtain the scanned images, any suitable segmentation algorithm can be used in order to get the segmented grains. After acquiring 1000 segmented images per class (total 1000×5 = 5000 images), they were divided into training, validation and test data in proportion of 80%, 10% and 10%, respectively.

In our work, we first train the proposed CNN architecture only using the data consisting 4000 training images (80% of the 5000 images). We will refer to this approach as proposed approach without transfer learning. We then use transfer learning in which we append our architecture to the pretrained network VGG16 [8] that is trained on image-net classification dataset. The combined network is then trained again. We will refer to this approach as the proposed approach with transfer learning or the approach using combined architecture. We will use these terms interchangeably.

3.1 PROPOSED ARCHITECTURE WITHOUT TRANSFER LEARNING

The proposed architecture without transfer learning is shown in Figure 1. The order of the layers in which the input data is passed through the architecture is VC_1, VC_3, VC_3, VC_4, Flatten, VFC_1, VFC_2 and finally the output. As seen from the figure, our input images are of size 125×50. Here, VC and VM represent convolutional and max pooling layer, respectively. Flatten represents unraveling of a 3-dimensional matrix channels, width and height into a 1-dimensional array. VFC_1 and VFC_2 represent the fully connected layers that correspond to simple multilayer perceptron's. Our output is a 5×1 array representing the output corresponding to 5 rice types. The first convolutional layer VC_1 has a filter size of 5×5, while the remaining 3 convolutional layers have filters of size 3×3. Thus, our proposed architecture is made up of 4 convolutional layers (VC_1-VC_4), 3 fully connected layers (VFC_1, VFC_2 and output) and one max pooling layer (VM). We use Rectified Linear Unit (ReLU) as the activation function and a dropout of 0.4 between two fully connected layers VFC_1 and VFC_2 of size 300 and 50, respectively. Prior to performing the convolution operation in VC_1 - VC_4, zero padding of size $(f-1)/2$ is applied, where f corresponds to filter size of that convolutional layer. This is done in order to keep the spatial output size same as the input size after performing the convolution operation. Thus, VC_1 has a zero padding of two ($(5-1)/2 = 2$) while VC_2-VC_4 have zero paddings of one. For max pooling, the stride size is fixed as two. Stride in convolutional layers is set to 1 since there is already reduction in features size due to max pooling (VM) after the convolutional layer (VC_1).

For training, the weights of the architecture are initialized randomly. Using the above specifications for the architecture, we train and validate our network. The final parameters of the network were decided after analyzing the training and the validation accuracy over each epoch. The number of epochs were decided based on the change in training and validation accuracies. The network parameters were changed after analyzing the confusion matrix and whenever we found over-fitting we reduced the number of epochs. Note that while training a network the training and the validation accuracy should not differ much. This is because if the training and validation accuracy differs much, its over-fits the data. Hence, we stopped the training process when the desired accuracy is obtained and this happens when the validation accuracy starts deviating
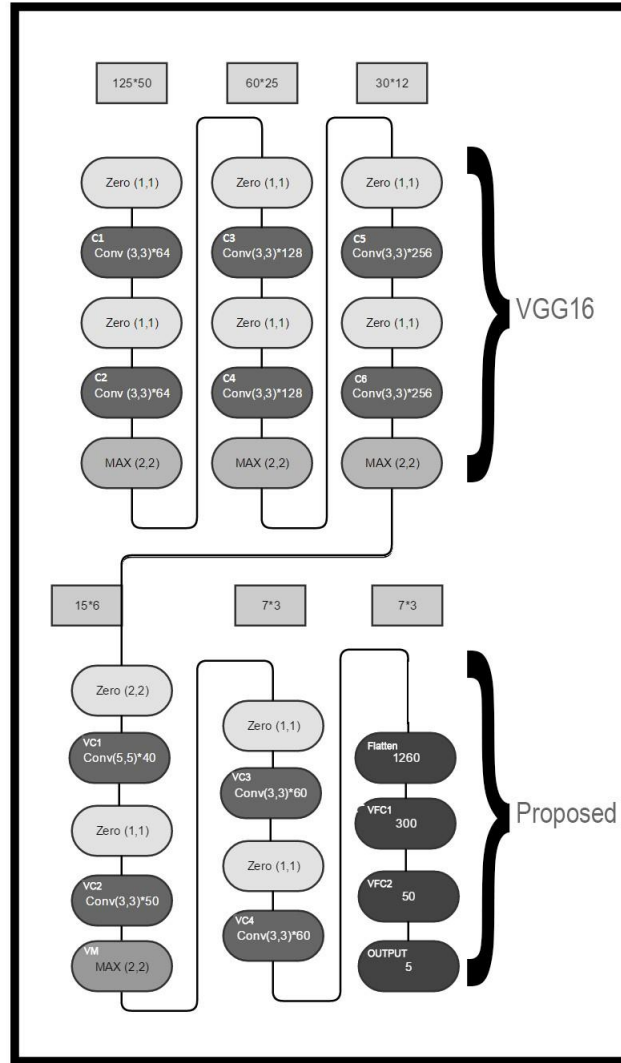
Figure 2: The combined architecture of VGG16 and proposed.

from training accuracy. Other thumb rule used was that of the filter size of the first convolutional layer which is chosen larger when compared to the other convolutional layers. In the next sub section, the proposed architecture with transfer learning is explained.

3.2 PROPOSED ARCHITECTURE WITH TRANSFER LEARNING

The combined architecture in which the proposed architecture without transfer learning shown in Figure 1 is appended is shown in Figure 2. This is divided into 6 stacks in which the first 3 stacks correspond to VGG16. Here, each stack corresponds to the blocks shown in one half column. The last three stacks belong to the proposed architecture without transfer learning and this corresponds to the same as shown in the proposed architecture given in Figure 1. In this work, we are using the first 6 convolutional layers of the VGG16 spanned in the first 3 stacks as C_1-C_6. Here, Conv (x, y) represents convolutional layer of filter size (x, y), Zero (x, y) represents the zero-padding layer with padding of x rows of zeros at top and bottom and y columns of zeros at left and right and MAX (x, y) represents max pooling layer of widow size (x, y).

In order to train the combined architecture, we proceed as follows. The weights of the last 3 stacks corresponding to the proposed architecture without transfer learning are randomly initialized. Then the combined architecture with the pretrained weights of VGG16 is trained using our training images with the weights of C_1 - C_6 fixed. For training, the Algorithm 1 Steps for learning weights using transfer learning.

1: Load the pretrained first 6 (C_1, C_2, C_3, C_4, C_5, C_6) convolutional layers of the VGG16 network.

2: Append the proposed architecture given in Figure 1 at the end of the pretrained VGG16 with 6 convolutional layers.

3: Train the combined architecture by updating only the weights of appended network.

4: Change the learning rate to a small value and once again train the network. This time allow the update of the last two convolutional layers of VGG16 and the last 4 layers of the appended network. i.e., C_5, C_6, VC_1, VC_2, VC_3, VC_4.

---

weights of the proposed architecture are allowed to update. Once the entire network is trained, fine tuning is done by allowing only the convolutional layers C_5 and C_6 (Stack 3 of VGG16 architecture) to train. The steps for learning the weights using transfer learning is given in Algorithm 1. The same training procedure as explained earlier for proposed architecture is followed here as well.
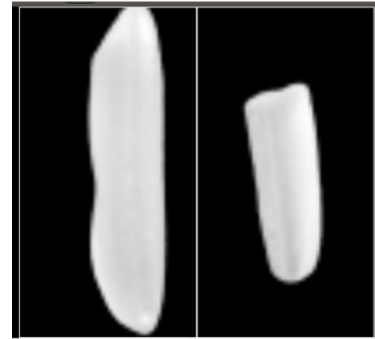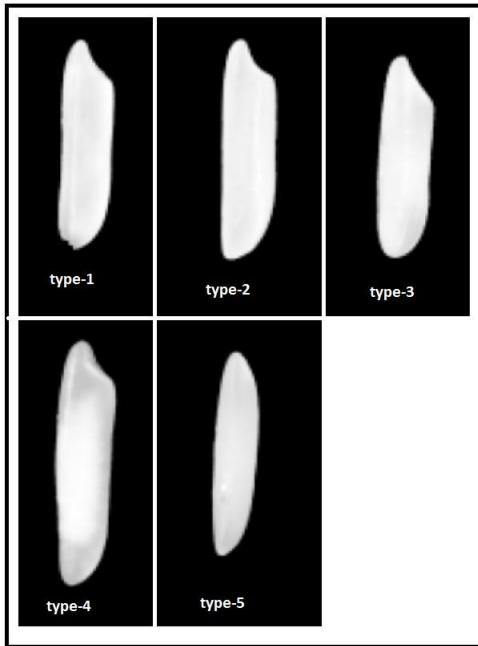

Figure 4: Normal and Broken rice.

Figure 3: Five types of rice grains used in the study.

Note that we are considering only the first 6 convolutional layers of the VGG16 pre-trained network. This is because these 6 layers learn the low-level features in these few layers i.e., the required details are learned from the available larger training dataset which is the ImageNet dataset. But, after these first 6 layers, the VGG16 learns some specific details of the large dataset, which are not required in the problem considered in the study. We want these specific details to be learned from our training set only even if it is of limited size (4000 images). This results in improvement in the overall accuracy of classification [19]. In effect we are obtaining the weights for learning the low-level features from VGG16 which is trained using a large data set of natural images and these features are common for all kinds of natural scenes. This is reflected in our learning process which is given in Algorithm 1. As seen from the algorithm we do not allow the training of the first 4 convolutional layers (C_1- C_4) of the VGG16. Even if we allow these layers also to learn, there exists a strong possibility that the important features already learned using millions of images may get distorted because of the limited size of our dataset. This causes the weights to change from their already learned optimal values to the other values which now corresponds to the training dataset of small size, resulting in over-fitting. While training, we allow the weights of the last two convolutional layers of VGG16 to update but with a very slow learning rate. The reason for this small learning rate is that it avoids the alteration of the features in these layers by a large amount which otherwise may cause over-fitting. In one of our experiments, we allowed the network to learn all the 6 convolutional layers of VGG16 and we observed significant over-fitting in that case.

Method	Dataset type	Classification accuracy
Bayes decision theory (morphological features) [1]	Six generic rice types	88.30%
Multi-class SVM [2]	Four rice grades	86.00%
Multi-layer perceptron (Morphological features) [3]	Four rice groups	70.00%
Multi-layer perceptron (Morphological features) [4]	Six rice groups	84.80%

Multi-layer perceptron (Morphological features) [5]	Nine rice groups	92.00%
Proposed Without transfer learning	Five Basmati rice types	86.80%
Proposed With transfer learning	Five Basmati rice types	94.20%

Table 1: Comparison of the proposed approach with the other works.

4. EXPERIMENTAL RESULTS

In this section, we conduct several experiments to illustrate the efficiency of our approach. The entire training process is carried out using NVIDIA Titan X GPU card for faster training. The Titan X GPU card has 3584 Cuda cores and 12 GB of standard GDDR5X memory. It is interesting to observe that, it is easier to distinguish among different grain types such as rice, wheat, corn, etc. However, distinguishing two different rice types is difficult. In the rice specie itself there are tens of different rice types available in the market and classifying some of these types is easier than others. This indicates that no meaningful comparison can be done between our work and the work done by other researchers. In TABLE 1, we list the comparison of our approach with four other works. One can see that even when we consider a dataset which has rice grains that are similar, the proposed method works better when compared to other approaches. Note that rice groups and grades are different from rice types. Rice type is an individual rice variety while rice groups are a cluster of similar rice types. Rice grades are based on quality and adulteration. Note that all the results listed in this section are averaged over 10 partitions of the training data. We have used 10-fold cross validation method to report the results.

Rice type	No. of samples used in testing	Classification accuracy without transfer learning	Classification accuracy with transfer learning
type 1	112	84%	91%
type 2	125	87%	92%
type 3	95	82%	96%
type 4	102	89%	94%
type 5	135	92%	98%

Table 2: Result of classifying different rice types with and without transfer learning.

	True label 1	True label 2	True label 3	True label 4	True label 5
Predicted label 1	102	4	0	3	0
Predicted label 2	3	115	0	2	0
Predicted label 3	3	2	91	0	0
Predicted label 4	1	3	2	96	3
Predicted label 5	3	1	2	1	132

Table 3: Confusion matrix of the classification between 5 types obtained with transfer learning

One can see from Figure 3 that the rice grains we have chosen are very similar in length and shape, making our problem more difficult. In TABLE 2, we list the classification accuracy by considering different rice types of the proposed method with and without transfer learning. The overall accuracy of the model with 5-class classification without and with transfer leaning are 86.80% and 94.20%, respectively. These figures indicate that the use of transfer learning significantly improves the classification accuracy. The confusion matrix or the precision and recall of the test results between these 5 types is listed in TABLE 3.

Rice 1	Rice 2	No. of samples used in testing	Classification accuracy without	Classification accuracy with

			transfer learning	transfer learning
type 1	type 2	237	85%	90%
type 1	type 3	207	89%	94%
type 1	type 4	214	85%	89%
type 1	type 5	247	92%	95%
type 2	type 3	220	82%	91%
type 2	type 4	227	89%	97%
type 2	type 5	260	83%	91%
type 3	type 4	197	94%	99%
type 3	type 5	230	84%	93%
type 4	type 5	237	94%	99%

Table 4: Binary classification with and without transfer learning.

Observe that type 1 and 2 as well as type 1 and 4 are hard to distinguish (see Figure 3). We see that the use of transfer learning helps us in improving the accuracy significantly for such classes as well. Generally, the traders mix a particular rice type with another type for illegal trade i.e., cheap one with costly one. Hence, we train architecture for $\binom{5}{2}=10$ different binary classes and the results are shown in TABLE 4. Note that these results are better than these given in TABLE 2. This is because classifying fewer number of classes is relatively an easy task since in this case our architecture has to distinguish between only two classes.

Rice condition	Samples used in testing	No. of samples classified correctly	Accuracy with transfer learning
Normal	83	83	100%
Broken	63	62	98%

Table 5: Results of Normal and Broken rice.

Good thing about our proposed architecture is that one can easily extend it to train using other types of rice as input and also add more classes to the problem at hand. Results of classifying the broken and normal rice grains where we used 2000 images for training with 1000 images for each class is shown in the TABLE 5 and the two sample images are shown in Figure 4. The number of epochs used in classification of 5 rice types correspond to 50 and 10 in the training and fine-tuning stage, respectively. While in the classification of broken and normal rice grains the number of epochs were selected as 17 and 3. Hence, the number of epochs were reduced in order to prevent the over-fitting. Since the binary classification problem is relatively easy, we require fewer number of epochs in this case. We can see from the Figure 4 that classifying normal and broken rice is relatively a simple problem, since there is significant difference in lengths of the two groups. As shown in TABLE 5, the results also support this claim with overall accuracy of 99.31% ((83+62)/(83+63) = 0.9931).

5. CONCLUSION

We have proposed a transfer learning-based approach for Basmati rice type classification. Though the rice type classification for Basmati rice group is a harder task than classification of rice types in general or rice groups, our proposed network with pretrained weights of VGG16 perform better than the other approaches. The experimental results suggest that proposed architecture is better than the previous works.

References

[1] Xing-yi Huang, Jian Li, and Song Jiang. "Study on identification of rice varieties using computer vision [J]". In: *Journal of Jiangsu University (National Science Edition)* 2.003 (2004).

[2] Harpreet Kaur and Baljit Singh. "Classification and grading rice using multi-class SVM". In: *International Journal of Scientific and Research Publications* 3.4 (2013), pp. 1–5.

[3] JD Guzman and EK Peralta. "Classification of Philippine rice grains using machine vision and artificial neural networks." In: *World conference on agricultural information and IT, IAALD AFITA WCCA 2008, Tokyo University of Agriculture, Tokyo, Japan, 24-27 August, 2008*. Tokyo University of Agriculture. 2008, pp. 41–48.

[4] Zhao-yan Liu, Fang Cheng, Yi-bin Ying, and Xiu-qin Rao. "Identification of rice seed varieties using neural network". In: *Journal of Zhejiang University. Science. B* 6.11 (2005), p. 1095.

[5] Chathurika Sewwandi Silva and Upul Sonnadara. "Classification of Rice Grains Using Neural Networks". In: *Proceedings of Technical Sessions*. Vol. 29. 2013, pp. 9–14.

[6] *INWEON, GRAMS, Rice analysis software*. URL: http://www.inweon.com/.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[8] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[9] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *Nature* 521.7553 (2015), pp. 436–444.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778.

[11] Yann LeCun and Yoshua Bengio. "Convolutional networks for images, speech, and time series". In: *The handbook of brain theory and neural networks* 3361.10 (1995), p. 1995.

[12] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. "Backpropagation applied to handwritten zip code recognition". In: *Neural computation* 1.4 (1989), pp. 541–551.

[13] Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. "Handwritten digit recognition with a back-propagation network". In: *Advances in neural information processing systems*. 1990, pp. 396–404.

[14] Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-based learning applied to document´ recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[15] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting." In: *Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958.

[16] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. "Learning and transferring mid-level image representations using convolutional neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 1717–1724.

[17] Sinno Jialin Pan and Qiang Yang. "A survey on transfer learning". In: *IEEE Transactions on knowledge and data engineering* 22.10 (2010), pp. 1345–1359.

[18] *ImageNet dataset*. URL: http://image-net.org/index.

[19] Matthew D Zeiler and Rob Fergus. "Visualizing and understanding convolutional networks". In: *European Conference on Computer Vision*. Springer. 2014, pp. 818–833.

AUTHORS' BACKGROUND

Name	Title	Research Field	Personal Website
Manjunath V. Joshi	Full Professor	Computer Vision and Deep Learning	https://sites.google.com/site/manjunathvjoshi/
Vaibhav Amit Patel	Student	Computer Vision and Deep Learning	https://sites.google.com/view/vaibhavpatel