DEEP CNN-BASED DAMAGE CLASSIFICATION OF MILLED RICE USING A HIGH MAGNIFICATION DATASET

MS (Research) Thesis

By
BHUPENDRA
(Roll No.: 2004103002)

DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY INDORE

JUNE 2022

DEEP CNN-BASED DAMAGE CLASSIFICATION OF MILLED RICE USING A HIGH MAGNIFICATION DATASET

A THESIS

Submitted in fulfilment of the
requirements for the award of the degree
of

Master of Science (Research)

by

BHUPENDRA
(Roll No.: 2004103002)

DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY INDORE

JUNE 2022

INDIAN INSTITUTE OF TECHNOLOGY INDORE

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled **Deep CNN-based damage classification of milled rice using a high magnification dataset** in the partial fulfillment of the requirements for the award of the degree of **MASTER OF SCIENCE (RESEARCH)** and submitted in the **DEPARTMENT OF MECHANICAL ENGINEERING, Indian Institute of Technology Indore**, is an authentic record of my own work carried out during the time period from August 2020 to June 2022 under the supervision of Dr. Ankur Miglani, Assistant Professor, Department of Mechanical Engineering, Indian Institute of Technology, Indore

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other institute.

Bhupendra 28 July 2022

Signature of the student with date

(BHUPENDRA)

This is to certify that the above statement made by the candidate is correct to the best of my/our knowledge.

Signature of the Supervisor with date

MS (Research) thesis #1

Dr. Ankur Miglani

Dr. Ankur Miglani
Assistant Professor
Department of Mechanical Engineering
Indian Institute of Technology Indore
Simrol, Khandwa Road, Indore 453 552, India

28 July 2022

BHUPENDRA has successfully given his MS (Research) Oral Examination.

Dr. Sandeep Chaudhary
Professor
Department of Civil Engineering
Indian Institute of Technology Indore
Simrol, Indore 453552

28-07-2022

Signature of Chairperson (OEB) with date

Dr. Ankur Miglani
Assistant Professor
Department of Mechanical Engineering
Indian Institute of Technology Indore
Simrol, Khandwa Road, Indore 453 552, India

28 July 2022

Signature(s) of Thesis Supervisor(s) with date

29/7/2022

Signature of Convener, DPGC with date

Signature of Head of Discipline with date

ACKNOWLEDGEMENTS

It is a well-known fact that no matter how big or small the work is, it takes effort to get it to fruition. During this quest of project there are many people who are like steppingstones in the way to success. Their support is undeniably crucial, and I am grateful to have support of such great people during my research work.

To begin with, I would like to convey my sincere regards to my mentor Dr. Ankur Miglani for providing his support and guidance during this entire journey of research work. Dr. Ankur not only acted as guide to carry out the work but also provided a crucial support be it technically or emotionally. Despite being a person of tremendous technical acumen, Dr. Ankur was always available to have vital discussions, which not only helped me to gain technical knowledge but also it taught me various life skills. Therefore, I offer my regards to my mentor for being with me.

I am also grateful to **Mr. Rakesh Singh** (Owner at **RSKissan Foods (India) Pvt. Ltd.**) for helping me with grain sorting, identification and collection and providing access to sortex machine.

I would like to express my indebtedness to the PSPC members **Dr. Pavan Kumar Kankar** and **Dr. Abhishek Srivastava** for providing their crucial insights in assessment of this work during the research work.

I am also thankful to my lab mates **Mr. Kriz Moses, Mr. Janmejai Sharma, Mr. Rahul Suryawanshi, Mr. Prince Kaushik, and Mr. Rishabh Gupta** for their help during this journey.

I want to express my thanks to **Mr. Suresh Badgole**, Lab assistant, for aiding me in lab work.

Lastly, I am thankful to my family for their support and IIT Indore for giving me such an experience of lifetime.

-Bhupendra

DEDICATION

I would like to dedicate this work to my family, friends, and almighty God.

Abstract

Surface quality evaluation of pre-processed rice grains is a key factor in determining their market acceptance, storage stability, processing quality, and the overall customer approval. On one end the conventional methods of surface quality evaluation are time-intensive, subjective, and inconsistent. On the other end, the current methods are limited to either sorting of healthy rice grains from the damaged ones, without classifying the latter, or focusing on segregating the different types of rice. A detailed classification of damage in milled rice grains has been largely unexplored due to the lack of an extensive labelled image dataset and the application of advanced CNN models thereon; that enables quick, accurate, and precise classification by excelling at end-to-end tasks, minimizing pre-processing, and eliminating the need for manual feature extraction. In this study, a machine vision system is developed to first construct a dataset of 8048 high-magnification (4.5 x) images of damaged rice refractions, that are obtained through the on-field collection. The dataset spans across seven damage classes, namely, healthy, full chalky, chalky discolored, half chalky, broken, discolored, and normal damage. Subsequently, five different state-of-the-art memory efficient Deep-CNN models, namely, EfficientNet-B0, ResNet-50, InceptionV3, MobileNetV2, and MobileNetV3 are adopted and fine-tuned to enable damage classification of milled rice grains. Experimental results show that the EfficientNet-B0 is the best performing model in terms of the accuracy, average recall, precision, and F1-score. It achieves an individual class accuracy of 98.33%, 96.51%, 95.45%, 100%, 100%, 99.26%, and 98.72% for healthy, full chalky, chalky discolored, half chalky, broken, discolored, and normal damage class respectively. The EfficientNet-B0 architecture achieves an overall classification accuracy of 98.37 % with a significantly reduced model size (47 MB) and a small prediction time of 0.122 s and can sub-classify the chalky class further into 3 different classes i.e., full chalky, half chalky, and chalky discolored. Overall, this study demonstrates the Deep CNN architectures applied to a high-magnification image dataset enables the classification of damaged rice grains with high accuracy, which could be utilized as a tool for

better and more objective quality assessment of the damaged rice grains at market and trading locations.

Keywords: Rice Quality, Deep Convolutional Neural Networks (DCNN), EfficientNet, High-magnification image, Deep learning, Machine learning.

LIST OF PUBLICATIONS

1. Bhupendra, Kriz Moses, Ankur Miglani, Pavan Kumar Kankar “Deep CNN-Based Damage Classification of Milled Rice Grains Using a High-Magnification Image Dataset.” Computers and Electronics in Agriculture, vol. 195, Apr. 2022, p. 106811. DOI.org (Crossref), https://doi.org/10.1016/j.compag.2022.106811.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction and Literature Review

Rice is India's most important staple crop. It was planted on approximately 44 million hectares (27.16% of the global area) to generate an all-time high of 121 million tonnes (23.96% of global output) with a record average productivity of 4100 kg/ha (Area, 2021) in the financial year 2020. As the demand for high-quality, nutritious food grains grows with the increasing population, the need for a fast, reliable, and objective quality evaluation of food grains has become increasingly important. Surface quality of pre-processed food grains is a significant element in market acceptability, storage stability, processing quality, and overall consumer approval. Quality control of food grains is a complex and time-consuming process. At one end, the conventional practices of quality evaluation involve manual inspection by experienced personal, which is time-consuming, expensive, and inaccurate because of the human decision making in identifying quality factors such as appearance, taste, nutritional content, and texture, and therefore, are inconsistent, subjective, and slow (Patel, et al., 2012).

Another option for reliable quality evaluation is to conduct a lab test. However, this method is expensive, time-consuming, and highly dependent on the testing time and availability of labs in a nearby location. Other methods of quality evaluation such as the flatbed scanners depend on the image resolution (dpi), which is generally low, and therefore, results in a low accuracy (Paliwal, et al., 2003). On the other end, the state-of-the-art Sortex machines (Pearson, 2010) are extremely expensive and have limited functionality i.e., they sort the grains into healthy versus damaged without quantifying the degree of damage, and hence, unsuitable for sorting the grains into different classes based on the type of damage or its severity.

Since this quality inspection can be very effectively done by visual symptoms, it poses a very suitable task for the field of Machine vision (MV). Machine vision provides an automated, non-destructive, and cost-effective way of ascertaining the grain quality based on the visual symptoms (Rehman, et al.,

2019; Du and Sun, 2006; Chen, et al., 2002). A typical computer vision system for food classification involves feature extraction from the images by identifying different patterns in each class and building classification algorithms on top of them for quality evaluation. Therefore, several past efforts have focused on grain quality assessment using machine vision systems (Paliwal, et al., 2003; Wan, et al., 2002; Payman, et al., 2018; Kaur, H. and Singh B., 2013; Chen, et al., 2019; AKI, O., Güllü, A. and Uçar, E., 2015; Majumdar, S., and D. S. Jayas., 1999; Vithu and Moses, 2016).

For instance, Wan et al. (Wan, et al., 2002) developed an automated inspection system to classify brown rice into healthy, cracked, chalky, immature, dead, damaged, and broken classes, which were prepared artificially in a lab. Although they have achieved a good processing speed of 1200 kernels per minute. Accuracy varies from 87.1% to 99.6% for different classes. Whereas the present work classifies milled rice damages that were present naturally. Payman et al. (Payman, et al., 2018) developed a heuristic-based MV system to classify rice grains into four classes (broken, chalky, red-spotted, and black-spot) using a dataset of 200 images (40 for algorithm development and 160 for its assessment). The traditional approaches use image processing and manual feature extraction focusing majorly on machine learning since it has proven to be extremely effective. For instance, Kaur and Singh (Kaur, H. and Singh B., 2013) classified four different grades of rice with an accuracy of 86%. They applied a multi-class Support Vector Machine (SVM) on a dataset of 800 images, extracting features such as shape and chalkiness.

Chen et al. (Chen, et al., 2019) classified the flawed red indica rice kernels (different from milled rice) into four classes, namely, cracked, chalky, damaged, and spotted. The damaged rice kernels in the images were detected using a support vector machine (SVM) classifier. Another SVM performed the grey-level segmentation, which was subsequently used to extract the chalky areas. Damaged and spotted areas on the rice kernels were identified using edge detection and morphological methods such as dilation and morphological closing. The overall accuracy achieved was 96.4%. Aki et al. (Aki, Ozan &

Güllü, Aydın & Uçar, Erdem, 2015) developed a machine learning model using PLS-DA (partial least squares discriminant analysis) and SVM to achieve an accuracy of 91% for classifying the rice samples.

However, a key drawback with the conventional methods used in these studies is that they require manual feature extraction, which is highly domain-specific, tedious, time-consuming, and error-prone. With deep learning (LeCun, et al., 2015), the image-based automated recognition technology, in particular, the Deep Convolutional Neural Networks (CNNs) have found applications in many areas of computer vision (Applications of Deep Convolutional Neural Network in Computer Vision, 2022; Voulodimos, et al., 2018; Wu, et al., 2017; Li, et al., 2021). They have produced state-of-the-art results in the current research field on many well recognized datasets (Cires¸an, 2012; Lee, 2014) and winning different object recognition challenges (Krizhevsky, et al., 2017; Russakovsky, et al., 2015; Szegedy, 2014). Deep CNNs manage to represent the raw data with multiple levels of abstraction, which enables them to extract the relevant features from the input automatically and translate from one domain to another, thereby, eliminating the need for image processing and manual feature extraction (LeCun, et al., 2015; Sampaio, P.S., et al. "Identification of Rice Flour Types with Near Infrared Spectroscopy Associated with PLS-DA and SVM Methods." European Food Research and Technology, vol. 246, no. 3, Mar., 2020).

There has been noticeable development in the application of deep learning-based methods for rice categorization. Ibrahim, et al. (Ibrahim, et al., 2020) applied support vector machines (SVMs) and artificial neural networks (ANN) on a dataset of 600 images (90 test images) to categorize rice grains into three kinds. The ANN performed better with an accuracy of 93.34% compared to 92.2% of SVM. Lin et al. (Lin, et al., 2017) demonstrated the efficacy of CNNs in feature extraction, which enabled the classification of rice grains of different

varieties using a dataset of 3819 images (2854 calibration and 965 for validation) with an accuracy of 99.52%. Aukkapinyo et al. (Aukkapinyo, K., et al. "Localization and Classification of Rice-Grain Images Using Region Proposals Based Convolutional Neural Network." International Journal of Automation and Computing, vol. 17, no. 2, Apr., 2020) used a pre-trained RCNN model (on COCO dataset) with data augmentation to detect and classify rice grains into five types of Thai rice. I. Chatnuntawech et al. (Chatnuntawech, et al., 1805) utilized a hyperspectral imaging technology to simultaneously collect complementary spatial and spectral information of rice seeds. This collected spatio-spectral data is then utilized to identify rice types using a deep CNN. Two datasets were used - paddy (1656 datacubes) and processed (1392 datacubes). The CNN suggested in the article was ResNet-B. For the paddy rice dataset, the suggested approach achieved 91.09% mean classification accuracy compared to 79.23% obtained using SVM with both spatial and spectral information.

Despite the prior efforts that have demonstrated the classification of rice grains, it has been limited to segregating different grain types instead of classifying the different types of surface damages in rice grains. Identification of surface damage in rice grains and their subsequent classification would be key factor in determining the market price of damaged rice, which is particularly important for the Govt. of India's Ethanol Blended Petrol (EBP) program. Since damaged rice is one of the key raw material for bio-ethanol production it quality evaluation is of paramount importance in deciding its market price based on the damage type and severity (Miglani, A. et al., 2014,2015,2016,2017)

in these prior studies, either the image dataset is small (the largest reported dataset has 3819 images (Lin, et al., 2017)), or the image magnification is low (maximum reported resolution is 21 μm/ pixel (Lin, et al., 2018)), meaning that the information per pixel is limited. Furthermore, the image datasets used in these studies involve various classes of rice grains that are noticeably different, and therefore, easy to identify and sort (Payman, et al., 2018; Chen, et al., 2019; Ibrahim, et al., 2020; Lin, et al., 2017; Aukkapinyo, K., et al. "Localization and

Classification of Rice-Grain Images Using Region ProposalsBased Convolutional Neural Network." International Journal of Automation and Computing, vol. 17, no. 2, Apr., 2020; Chatnuntawech, et al., 1805). However, in practice, the dataset is complex and involves an overlap between different classes, as is demonstrated in this study.

A past few studies where damage classification is addressed (Payman, et al., 2018; Kaur, H. and Singh B. "Classification and Grading Rice Using Multi-Class SVM". International Journal of Scientific and Research Publications, Volume 3, Issue 4, April 2013; Chen, et al., 2019) have been based on the classical manual feature extraction-based modelling. Existing literature lacks a high-magnification large image dataset of different surface damages in milled rice (Payman, et al., 2018; Kaur, H. and Singh B. "Classification and Grading Rice Using Multi-Class SVM". International Journal of Scientific and Research Publications, Volume 3, Issue 4, April 2013; Wee, et al., 2009; Putri, et al., 2015;Yao, xxxx), and the application of advanced deep CNN models thereon to enable a detailed classification of damage in milled rice grains is unexplored.

In most of the relevant literature involving Deep CNNs, traditional state-of-the-art CNNs like VGGNet (Simonyan and Zisserman, 2015) and ResNets (He, 2015) are used, which have been quite popular for image classification. These models have extensively been used for the classification of food grains as well. They can be applied with transfer learning (Weiss, et al., 2016) or can be trained fully, given the dataset is large. They can achieve good accuracy and very well translate to our problem. However, it is observed that they can be quite bulky with a huge number of parameters leading to a large model size and high inference time. This is not suitable when you need real-time identification of rice grains where factors like time, model size, and power consumption also play a big role. The use of lightweight networks, with lesser size and low inference time, is also unexplored in this field.

Objectives and Scope

A primary goal of this study is to develop a high-magnification image dataset spread across seven different types of damages, namely, healthy, broken, normal damage, discolored, full chalky, half chalky, and chalky discolored. Subsequently, apply the state-of-the-art CNN architectures to enable damage classification of rice grains with high accuracy. Further a comparative study on the performance of lightweight CNNs (MobileNetV2 (Sandler, 2019), MobileNetV3 (Howard, et al., 1905), EfficientNet (Tan and Le, 2020)) along with traditional CNNs (ResNet-50 (He, 2015) and InceptionV3 (Szegedy, et al., 2016)) is presented.

The organization of the thesis is as follows:

Chapter 1: Introduction and Literature Review: This chapter discusses the previous studies along with their research gaps and thus sets the motivation for carrying out this research work.

Chapter 2: Materials and methods: This chapter discusses damage in rice grains, and how the dataset has been acquired. It also talks about the CNN models that have been used in this study along with the data augmentation technique.

Chapter 3: Experiments: It talks about pre-processing, training, and performance metrics.

Chapter 4: Results and Discussions: It talks about how the best model is chosen based on some parameters, and then it discusses the error analysis through a confusion matrix.

Chapter 5: Conclusion and Future scope: This chapter gives the concluding remarks for this study by mentioning the key conclusions followed by the future scope.

Chapter 2

Materials and methods

A typical machine vision system for food quality evaluation would follow five steps (as shown in Fig. 2.1): Image Acquisition, Pre-processing, Segmentation, Feature Extraction, Classification (Mery, et al., 2013). Since CNN-based models have been used in this study, it has eliminated the need for segmentation and feature extraction. Thus, the proposed machine vision system for the classification of rice grains consists of three steps: Image Acquisition, Pre-processing, and Classification. This is shown in Fig. 2.1.

2.1. Data acquisition

Building the database of damaged rice grains is the most crucial step in this study. Therefore, grains were collected and sorted on the field in preparation for constructing the database. Next, an in-house image database with 8048 images divided across 7 categories has been created using the imaging system.

2.1.1. Sample collection

Rskissan Foods (India) Private Limited, Mirzapur, Uttar Pradesh, assisted this study by permitting to obtain the rice grain samples (BPT 5204 variety) from the pile of rejection of a double color sortex machine (Milltec) for the 2019 growing year. The BPT 5204 is a medium-sized grain, mainly grown in Andhra Pradesh, Telangana, Karnataka, and some of the regions of Madhya Pradesh, Bihar, and Uttar Pradesh. The identification of different types of damages in rice grains is based on USDA visual reference manuals (Rice | Agricultural Marketing Service. https://www.ams.usda.gov/book/rice. Accessed 7 May 2021), and the Government of India Department of Food and Public Distribution (Refractions in Raw Parboiled Rice | Storage Research | Divisions | Department of Food and Public Distribution, Government of India. https://dfpd.gov.in/refractions-in-raw.htm. Accessed 7 May., 2021). Based on this, 8000 rice grains have been collected (including damaged, healthy and broken). A total of 8048 (for some grains, images are taken for both front and backside) color images of individual grain kernels were acquired as a dataset.

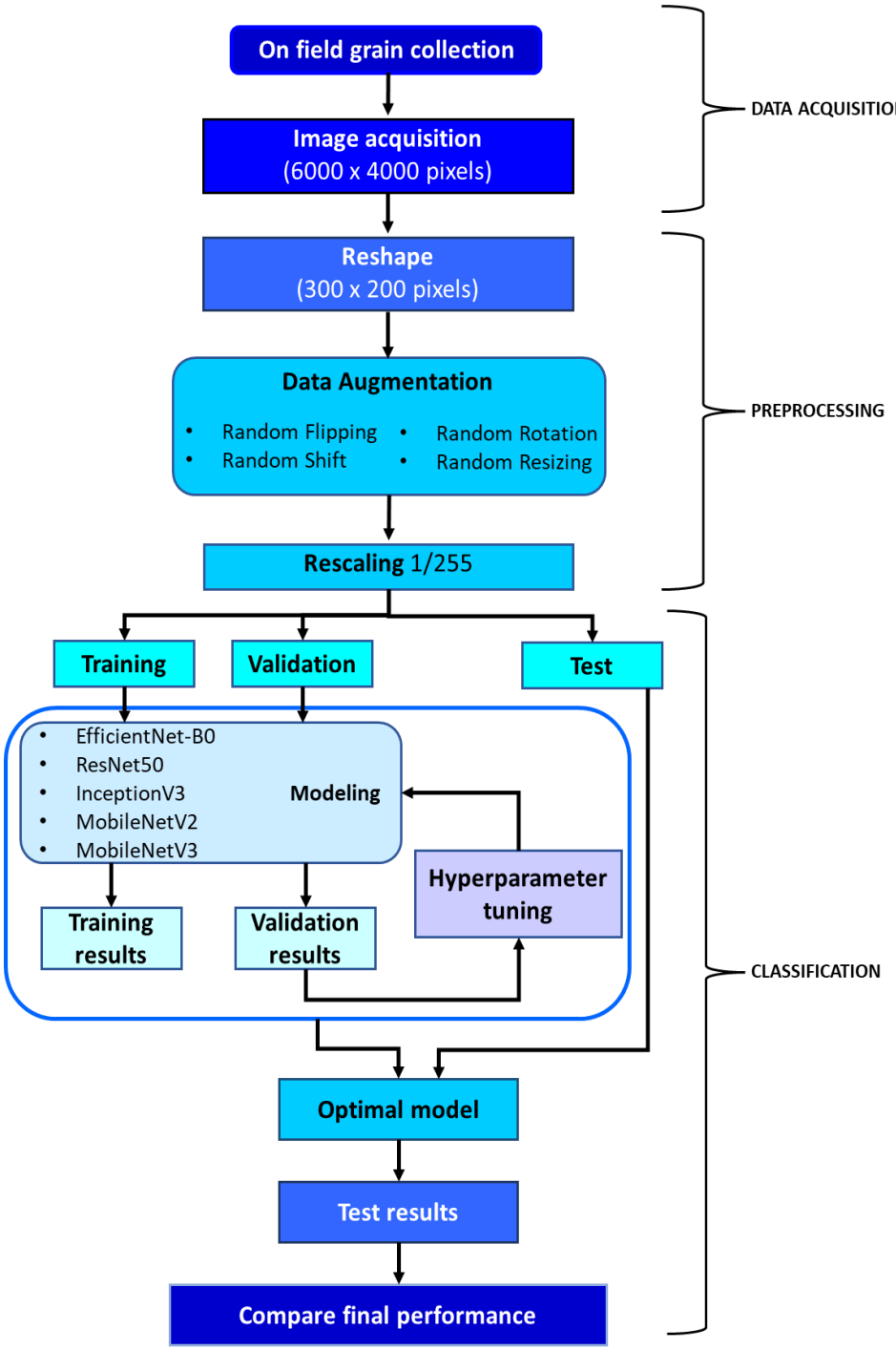


Fig 2.1: A flowchart showing the methodology.

2.1.2. Imaging system

Our computer vision system consists of an LED light source for illumination because of its benefits such as long life, energy-efficient, and no heat or UV emissions.

A mirrorless camera (Alpha 5100, Sony) with a complementary metal–oxidesemiconductor (CMOS) sensor and coupled with a microscopic zoom lens (Zoom 6000, Navitar) is used to obtain high-resolution (6000 × 4000 pixels), high-magnification (3.9 µm/pixel) images of rice kernels (The resolution mentioned is in height × width format). A zoom lens (Navitar) which gives us the working distance of 10 cm with 4.5x magnification so that we can easily pick and place the rice grain and get the most amount of rice in the image, a computer system (i5-9500 CPU @ 3.00 GHz, Dell OptiPlex 3070) and software (Sony, Imaging Edge Desktop).

The camera is mounted on a vertical stand, as illustrated in Fig. 2.2. To get the image, a 3-axis stage is positioned underneath the camera, on which rice grains are placed over a blue background. To get a clear image, the entire arrangement is enclosed in a rectangular box with a black inside coating to prevent stray reflections, as well as an LED panel and high-efficiency diffuser on its sides. As demonstrated by the red dotted line, the camera is connected to the PC and is managed by software to take and store images without disrupting the setup.

2.2. Dataset details

The dataset contains a total of 8048 images. The images have been labelled into seven categories, with the class names and their distribution as follows: healthy (1208), broken (1041), normal damage (1576), full chalky (883), chalky discolored (1107), half chalky (861), discolored (1372). The labelling has been done manually through visual symptoms/features, which are explained in detail in section 2.3.1. All the images are in RGB format with a size of 6000 × 4000 pixels i.e., a resolution of 24 MP (Megapixels). Although the size of the images has been reduced to 300 × 200 for the deep learning models, the acquisition of images with such a high resolution of 24 MP was necessary, since it helped in manual detection of features, and hence labelling. This proved to be difficult when dealing with images of lower resolution (like 300 × 200). This is because this study focuses on the fine-grained classification of damages, where feature detection and manual labelling for each class is not very trivial i.e. the features

separating each class are not simple to identify.

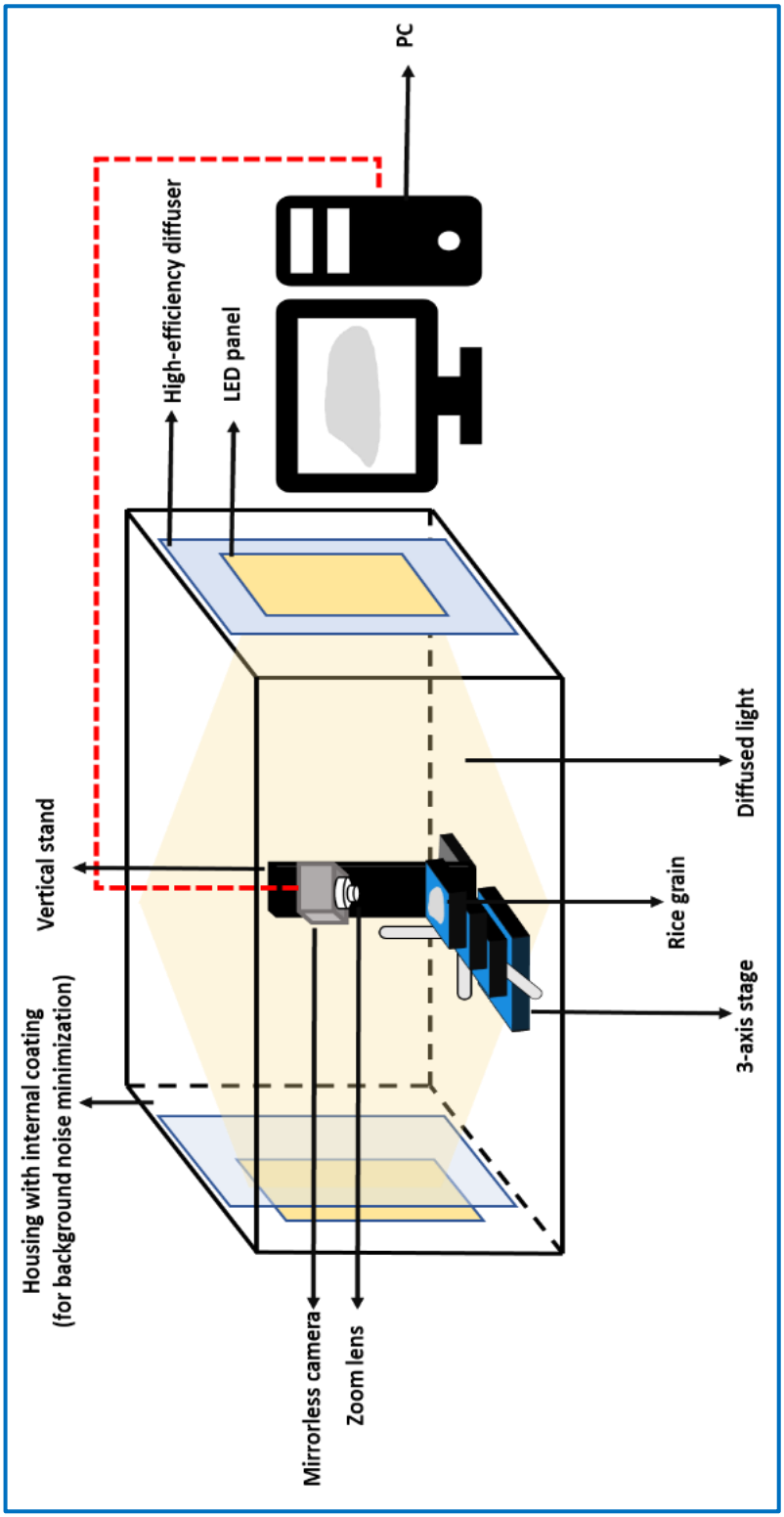


Fig 2.2: A schematic diagram of an image acquisition system.

2.2.1. Damage in rice grains

Milled rice grains are classified into seven categories: healthy, broken, normal damage, full chalky, half chalky, chalky discolored, and discolored. Healthy grains have not been damaged on the surface. Broken grains have a similar appearance to healthy grains, but they are broken. This may occur either after milling or during the milling process, when internal stress is generated in the healthy grains, causing them to shatter into tiny fragments. Heat-damaged grains have a black area, pin damaged grains have an indentation on the surface and are produced by insects, and watermark damaged grains have a brown circular ring and are caused by water or other means. Heat damaged, pin damaged, and watermark damaged grains are all included in the normal damage class. Chalky grains have a white color, are opaque, and are brittle. Chalkiness is produced by inadequate starch build-up during the grain filling process. Chalky grains were then divided into three groups based on how they appeared on the outside. They are termed complete chalky if the chalkiness covering the surface area is 90 percent or more; otherwise, they are classed as half chalky. Chalky discolored grains may be completely chalky or half chalky, but they will always have some discoloration, which varies in intensity from grain to grain. Paddy grains that have been exposed to damp conditions before drying and milling have a yellowish or brownish colored surface, consequently, these kinds of grains are termed discolored grains. The real damaged grain pictures, as well as their schematic, are shown in Fig. 2.3 to obtain a good idea of various sorts of damages found in rice grains. Some representative images of each damage type are shown in the appendix section.

In the present study, a 4.5x microscope is used to examine all the damaged grains, as well as broken and healthy grains in individuals, and take photographs. All the chalky classes have similarities among them. Chalky discolored has a dark husk spot and normal damage has a dark color spot. Sometimes healthy class has very light chalkiness in a small region which may conflict with half chalky. Light watermark damage that is included in normal

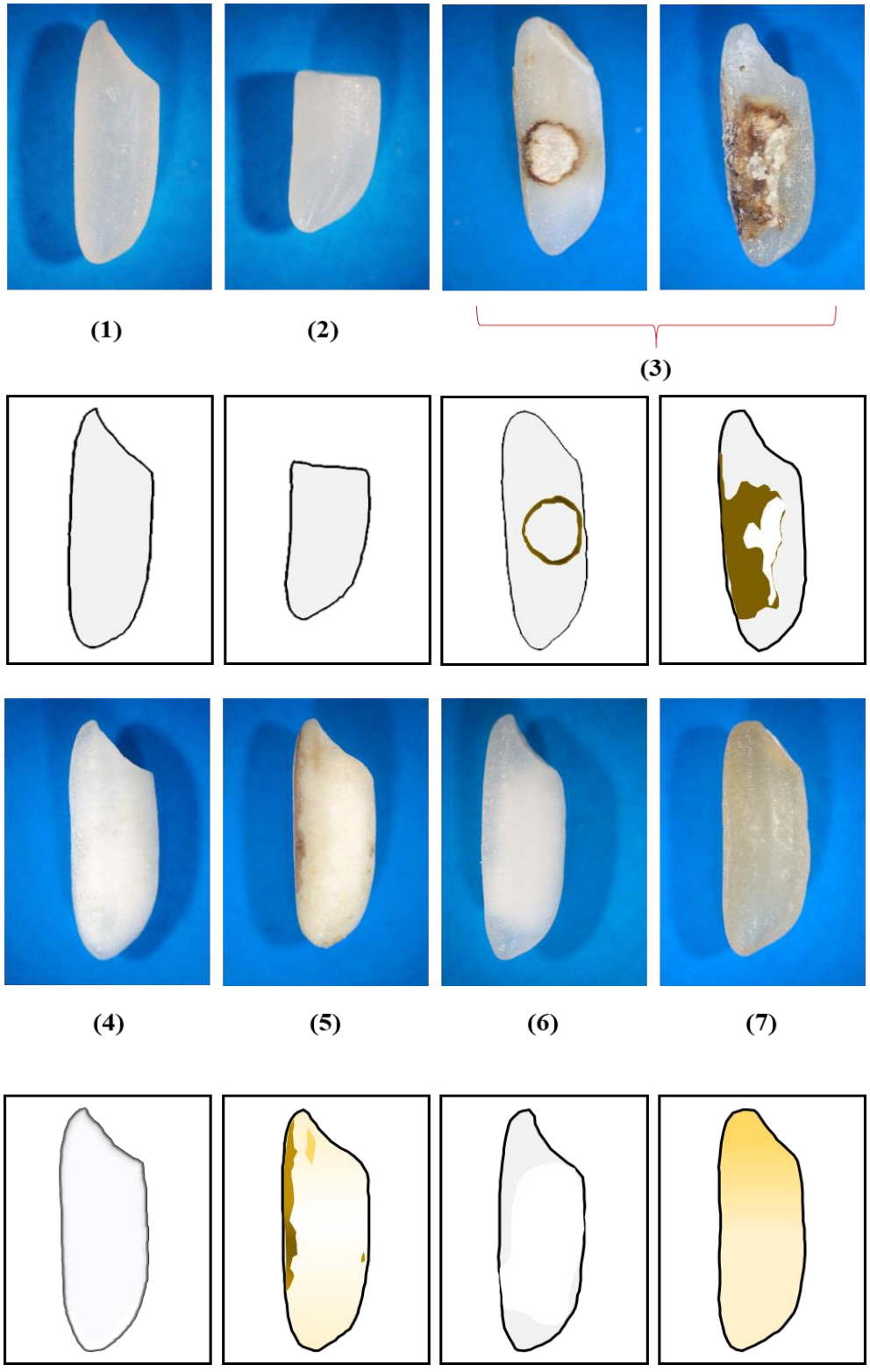


Fig 2.3: (1) Healthy, (2) Broken, (3) Normal damage, (4) Full chalky, (5) Chalky discolored, (6) Half chalky, (7) Discolored.

damage, may conflict with discolored class. The above discussion is based on observation and thus it makes the dataset complex.

2.3. Convolutional neural networks

Convolutional Neural Networks (Lecun et al., 1998; Nebauer, 1998; LeCun et al., 2010) are deep learning algorithms, specifically designed to work on images. They attempt to automatically extract the relevant features from an image by learning the spatial and temporal dependencies through the application of relevant filters. A filter is a square matrix with learnable parameters (weights and biases), which is used to capture a specific feature from the input image by convolving with the input feature map to produce an output feature map. In particular, for an input feature map of size (n × n × c), a (k × k) size kernel will perform these convolutions at each (k × k) patch and produce an output feature map of size (n + 1-k, n + 1-k, 1). The depth of each filter is set equal to the depth of the input map. The number of filters used decides the depth of the output feature map. A simple ConvNet is a sequence of layers connected through differentiable functions, to enable learning through backpropagation. Three main types of layers are used to build ConvNet architectures, namely, Convolutional Layer, Pooling Layer, and Fully Connected Layer. The convolutional layer tries to extract the important features from the image through the convolution operations containing multiple filters, where each tries to capture a different feature. The pooling layer (D. Cires¸an, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," in Proc. of 22nd Intl. Joint Conf. on Artificial Intelligence, 2011; Scherer et al., 2010) is used to down sample the feature map. In max pooling, the maximum value is calculated from the patch of each input feature map depending on the kernel size. It helps in learning complex invariances such as the scale and rotational invariance. Further, the fully connected layers are used to connect all the input features to the output categories. Fig.2.4 depicts a typical CNN with an input image size of 300 × 200 × 3 and the corresponding feature map sizes. The convolutions shown are with the padding 'same', which pads the input feature map to ensure that the

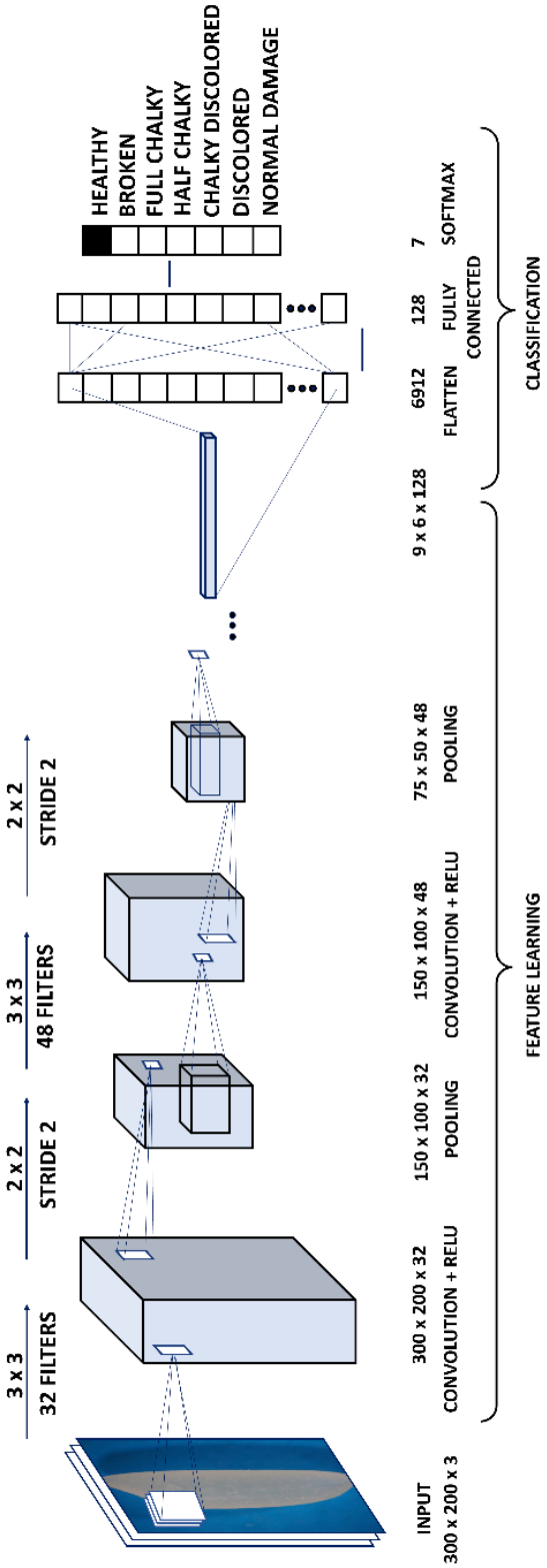


Fig 2.4: Architecture of a Standard Convolutional Neural Network with dimensions.

output is of the same resolution as the input. Strides represent the number of steps the filter takes before moving onto the next patch of the input map. Finally, the output feature map of size $9 \times 6 \times 128$ is flattened and connected with a fully connected layer of 128 neurons, which further connects with the output layer of 7 neurons with the SoftMax activation.

Since the introduction of AlexNet (Krizhevsky, et al., 2017), winning the ImageNet challenge: ILSVRC (Russakovsky, et al., 2015) in 2012, there has been significant development in the field of Computer Vision using Deep CNNs. Several models have been proposed since then. The state-of-the-art models include VGGNet (Simonyan and Zisserman, 2015) and Inception (Szegedy, 2014) in 2014, ResNets (He, 2015) in 2015 and DenseNets (Huang, 2018) in 2016. Over the years, the general trend has been to make deeper, more complex networks, and scaling up baseline ConvNets to achieve better accuracy: ResNets have been developed from ResNet-18 with 18 layers to ResNet-200 with 200 layers. Similarly, different versions of Inception (Szegedy, et al., 2016; Ioffe and Szegedy, 2015) and DenseNets have been proposed. However, it was observed that constructing deeper networks may increase accuracy but a lot of times, these models compromise on other factors like inference time, size of the model, and operations count. Over the past few years, there has been a lot of emphasis on developing fast and efficient lightweight architectures for usage in embedded and mobile devices. In 2017, a family of models called MobileNets (Sandler, 2019; Howard, et al., 1905) was developed with a focus on Mobile Applications. In 2018, NasNet (Zoph, 2018) was introduced, which aimed towards searching for optimal CNN architecture using reinforcement learning. Both these networks compared to state-of-the-art algorithms, with far fewer parameters and became the go-to networks for mobile applications. In 2019, a family of models named EfficientNets (Tan and Le, 2020) was introduced. The authors of the paper came up with an ingenious idea to efficiently scale CNN architectures using compound scaling, which lets you scale the network's depth, width, and input resolution together. These models

achieved state-of-the-art performance with far lesser parameters and are rapidly replacing ResNet as the backbone of choice for many Computer Vision Tasks. In this study, the performance of 5 state-of-the-art-CNN models (ResNet-50, InceptionV3, MobileNetV2, MobileNetV3, and EfficientNet) are compared on different metrics as mentioned in Section 3.4.

2.3.1. ResNet-50 and Inception V3

ResNet50 (He, 2015) won the ILSVRC-2015 competition in 2015. The architecture was specifically designed for tackling the accuracy degradation problem on increasing the depth of the CNNs. One reason attributed to this was the presence of multiple non-linear layers being unable to learn the identity mappings. ResNet introduced the skip connections (as shown in Fig. 2.5) in which the input feature map was added to the output feature map after a few weight layers. Hence, the output H(x) = F(x) + x. The stacked weight layers trying to fit another mapping: F(x) = H(x) – x, such that, even if in an extreme case, identity mapping was optimal, the residual can always be pushed to zero. ResNet50 is an architecture based on many such stacked residual units. For each residual function F, three stacked layers are used as shown in Fig. 2.5. 1x1 convolutions are used to reduce and increase the dimensions. This creates a bottleneck 3 × 3 layer with smaller input/output dimensions.

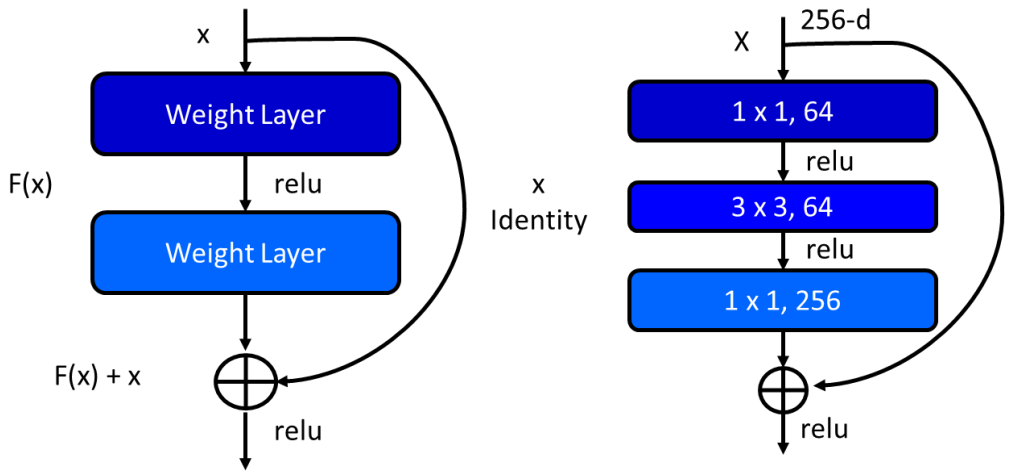


Fig 2.5: (Left) a residual building block with skip connection (Right) a "bottleneck" building block for ResNet-50.

Inception V3 (Szegedy, et al., 2016) developed by Google, which was the third release in the series of Inception networks, was the first Runner Up in the ILSVRC-2015 competition. Fig. 2.6 depicts one of the inception modules used in the architecture, which is built on top of the previous versions v2 and v3 (Szegedy, 2014; Ioffe and Szegedy, 2015). Firstly, the convolutions with varying kernel sizes are performed altogether for the previous input to extract more variety of features and stacked together again at the output. 1 × 1 convolutions are used for reducing the dimensions of the feature maps. This helps in reducing the computation time due to fewer convolution operations to be performed in the subsequent layers. Further, Batch Normalization is used from v2, where all the images are normalized using the batch statistics after a layer; as shown in equation below, where *x* represents the inputs to be normalized over a batch of size *m*.

Input: Values of *x* over a mini-batch: B = $\{x_1 \ldots m\}$;

Parameters to be learned: γ, β

Output: $\{ y_i = \mathrm{BN}_{\gamma, \beta}(x_i)\}$

$\mu_B \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i$ // mini-batch mean

$\sigma_B^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2$ // mini-batch variance

$\hat{x}_i \leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$ // normalize

$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \mathrm{BN}_{\gamma, \beta}(x_i)$ // scale and shift

Further, Inception v3 introduced the idea of factorizing convolutions to reduce the number of parameters without decreasing the network efficiency. The main idea was to factor the larger filters (e.g., 5 × 5 or 7 × 7) into multiple smaller ones (3 × 3 or 1 × 1), which can perform equally well in capturing the relevant information with the smaller size of the parameters.

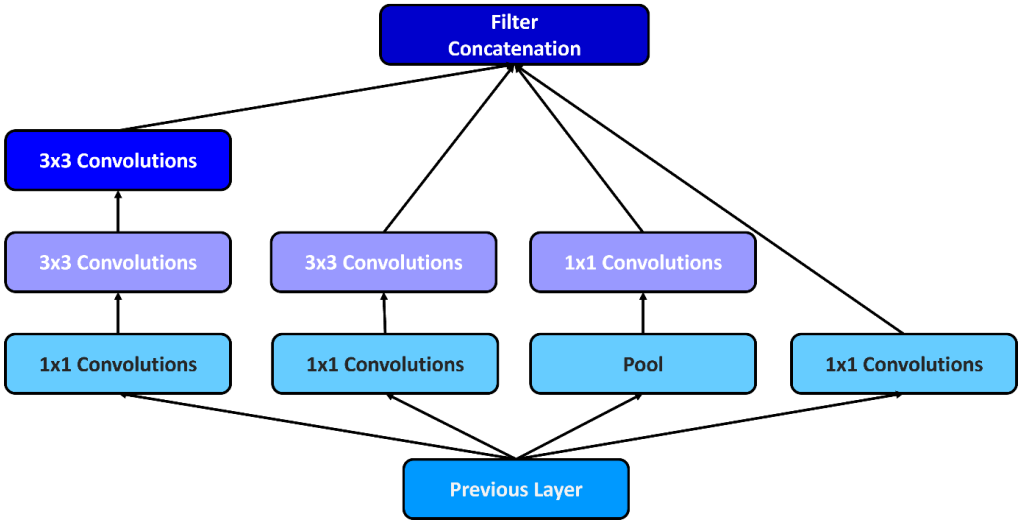


Fig 2.6: Inception Module with the factorization of 5 x 5 convolutions into 3 x 3 ones.

2.3.2. MobileNet and EfficientNet

Building deeper networks and scaling up of baseline ConvNets have worked well for achieving good accuracy but mostly at the cost of larger numbers of parameters and an increase in inference time. All this time, people used various techniques for scaling and there wasn't a uniform and more structured way for the same. The goal was to achieve maximum possible accuracy while using as few FLOPS (floating-point operations per second) as possible. In 2020, Tan and Le (Tan and Le, 2020) proposed the idea of compound scaling, which gave rise to a family of models called the EfficientNet. The authors of the paper show that the EfficientNet models outperform all previous models both in terms of the number of parameters and accuracy when applied to the ImageNet dataset.

EfficientNet achieves such results by scaling depth, width, and resolution uniformly while scaling down the model. In compound scaling, all the three dimensions of a network: the network's depth, width, and input resolution are systematically scaled together rather than conventional single-dimension scaling. For this they use a compound coefficient φ to uniformly scale network width, depth, and resolution in a principled way:

depth: $d = \alpha^{\varphi}$

width: $w = \beta^{\varphi}$

resolution: $r = \gamma^{\varphi}$ (2.1)

s.t. $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$ *and,* $\alpha \geq 1, \beta \geq 1, \gamma \geq 1$ (Tan and Le, 2020)

where α, β, γ are constants that can be determined by a small grid search. Intuitively, φ is a user-specified coefficient that controls how many more resources are available for model scaling, while α, β, γ specify how to assign these extra resources to network width, depth, and resolution respectively. Notably, the FLOPS (floating-point operations per second) of a regular convolution op is proportional to d, w^2, r^2, i.e., doubling network depth will double FLOPS, but doubling network width or resolution will increase FLOPS by four times. Since convolution ops usually dominate the computation cost in ConvNets, scaling a ConvNet with equation 2.1 will approximately increase total FLOPS by $(\alpha \cdot \beta^2 \cdot \gamma^2)^{\varphi}$. In paper Tan and Le, 2020, they constraint $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$ such that for any new φ, the total FLOPS will approximately increase by 2^{φ}. By using these equations and fixing φ = 1, the best values for EfficienNet-B0 were found to be α = 1.2, β = 1.1 and γ = 1.15.

For finding the baseline network, MnasNet (Tan, et al., 1807) is used as the Neural Architecture Search. The same search space is used and the main building block is the inverted bottleneck MBConv, first introduced in the MobileNet module (Howard, 2017), to which they add squeezeand-excitation optimization as well. In the paper, the EfficientNet-B0 is used since it is the most compact of all the models with only 4 million parameters and provides a good compromise between computation resources and accuracy. The schematic representation of EfficientNet-B0 is shown in Fig. 2.7. In the figure, GAP stands for Global Average Pooling and FC for Fully Connected Layer. The “Conv” blocks in the diagram stand for simple normal convolutions followed by batch normalization and swish activation. Note that the shapes of the feature maps in the figure are for an input image of size 300 × 200. However, in the training

process, rescaled images are randomized (under data augmentation) before entering the model, thus these shapes change accordingly. The final feature vector of dimension 1280 remains constant, before applying softmax. Fig. 2.8 shows an illustration of the MBConv6 block. As seen in MobileNets (Sandler, 2019), the main ideas of the network included depth-wise separable convolutions, inverted residual connections, and linear bottlenecks. On top of this, a squeeze and excitation block have been added as well.

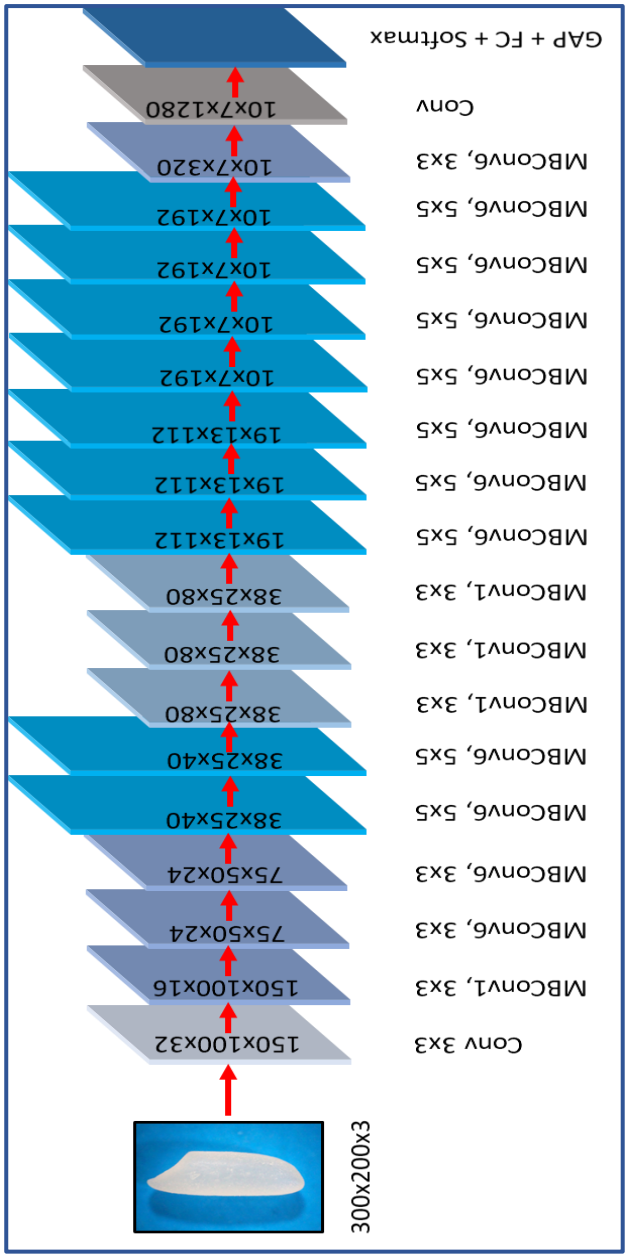


Fig 2.7: Schematic representation of EfficientNet-B0.

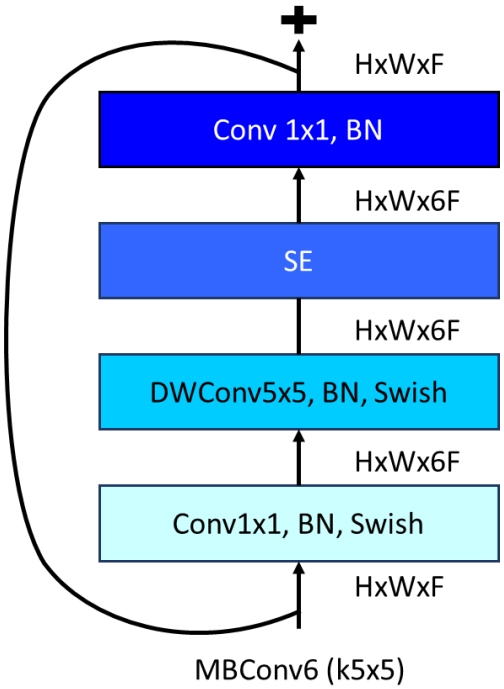


Fig 2.8: Illustration of MBConv6 block. SE (Squeeze Excitation), BN (Batch Normalization), DWConv (Depthwise convolution)

Depthwise separable convolution refers to depthwise convolution followed by a pointwise convolution. The first depthwise convolution performs channel-wise spatial convolutions, where a single convolutional filter is applied for each input channel. Pointwise convolution is 1× 1 convolution to change the dimension of the feature map. The depthwise convolution proves to reduce the computational time considerably as compared to normal convolutions. The reduction in computation comes out to be equal to = 1/N +1/Dk2 where N represents the number of output channels and Dk, the feature map size. Fig. 2.9 illustrates this by taking an example of an input feature map of size 6x6x3 convolved to produce an output feature map of size 4x4x5.

The two other concepts, taken from MobileNetV2 include inverted residuals and linear bottleneck. Residual blocks use a skip connection to connect the beginning and end of a convolutional block, which has shown to be quite successful in constructing deeper networks. In the original residual blocks, skip

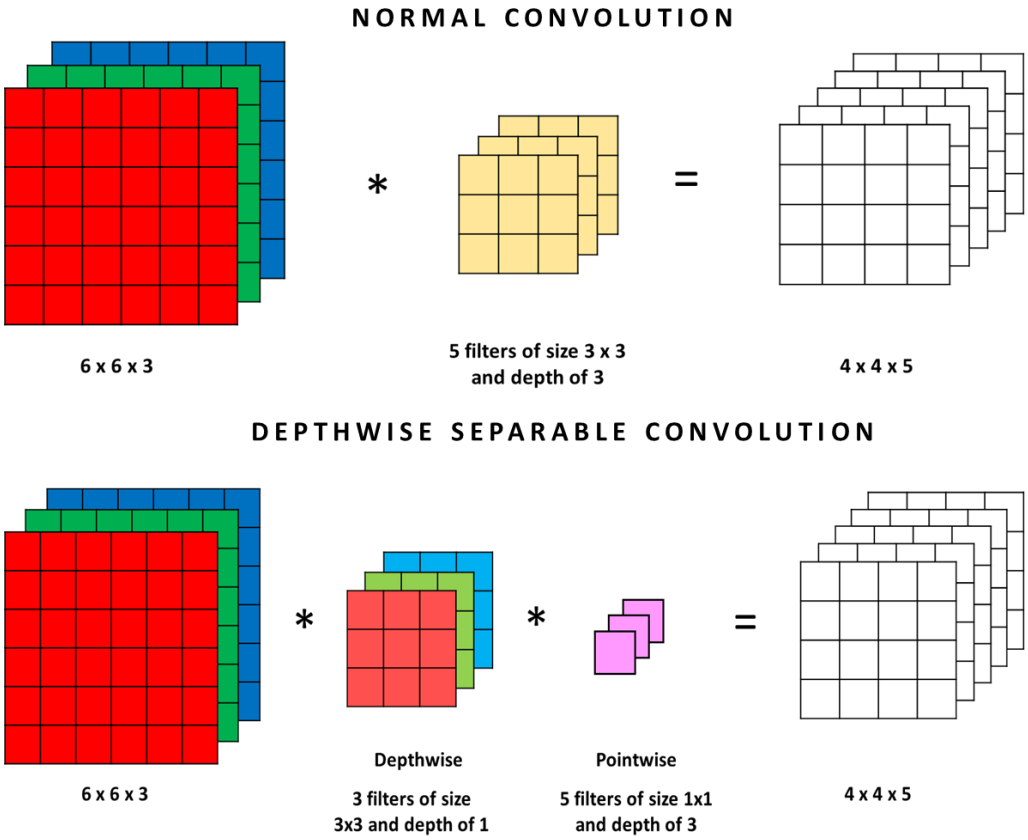


Fig 2.9: Represents normal convolution and depthwise separable convolution.

connections exist between the wider parts of the network, whereas MobileNetV2 follows the opposite approach. It first widens the network using 1x1 convolution, followed by a 3x3 depthwise convolution. After, another 1x1 convolution squeezes the network to match the initial number of channels (Fig. 2.10). Hence, skip connections exist between the narrow parts of the network. The idea behind the expansion was to have a computationally rich region where feature extraction can take place. Depthwise convolution makes this possible since it greatly reduces the number of parameters. This is followed by contraction to make the final output of lower size for memory efficiency. In practice, the inverted residual block has far fewer parameters than the residual block. The other concept was linear bottlenecks, where the last convolution of a residual block has a linear output before being added to the initial activations. According to the authors of MobileNetV2, the ReLU activation function, which

is widely used in CNN architectures, does not perform well with inverted residual blocks since it discards values less than zero. Hence the network used linear activations, which produced a better performance for the bottleneck channels. Additionally, the network uses Swish activation: (Tan and Le, 2020)

$$f_{swish}(x) = x / (1 + e^{-\beta x}) \quad (2.2)$$

here, $\beta >= 0$ is a parameter that can be learned during the model training. For β = 0, the function turns into a scaled linear function $f(x) = x/2$, whereas for $\beta \rightarrow \infty$, behaves more like a smooth ReLU function. Further, the concept of Squeeze and Excitation (SE) Block, improved the performance of the model. The SE block gives different weightage to each channel of its input feature map instead of treating them equally. The output shape of the SE block is of shape (1 x 1 x channels), specifying the weightage for each channel which can be learned during training.

2.4 Data Augmentation

Data Augmentation has been shown to produce promising ways to increase the accuracy of classification tasks, by increasing the desired invariance and robustness properties. In this work, extensive data augmentation has been used to increase the amount of training data without the need to acquire more data. A separate pipeline was constructed for data augmentation. In this pipeline, images undergo random transformations before being fed to the main model. Random flipping is used in both horizontal and vertical directions. Then the images are shifted randomly in the horizontal and vertical space. The maximum amount of shift was set to 5% for the vertical and 10% for the horizontal. Images are then rotated by a maximum amount of 0.025x2π radians or 9 degrees in either direction. On top of this, the shape of the images was also altered by an amount of 10% for both height and width. Fig. 2.11 shows the various data augmentation operations that have been performed on a half chalky rice grain.

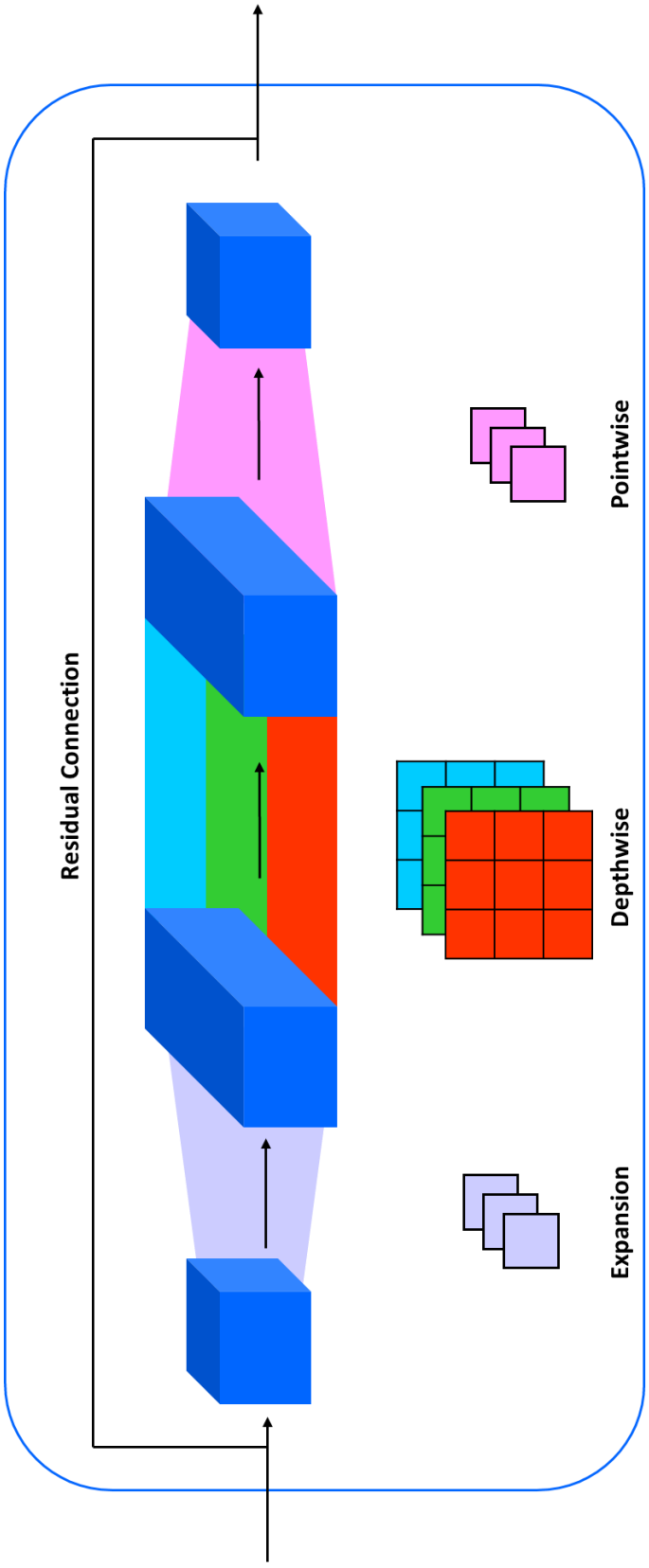


Fig 2.10: Representation of residual connection.

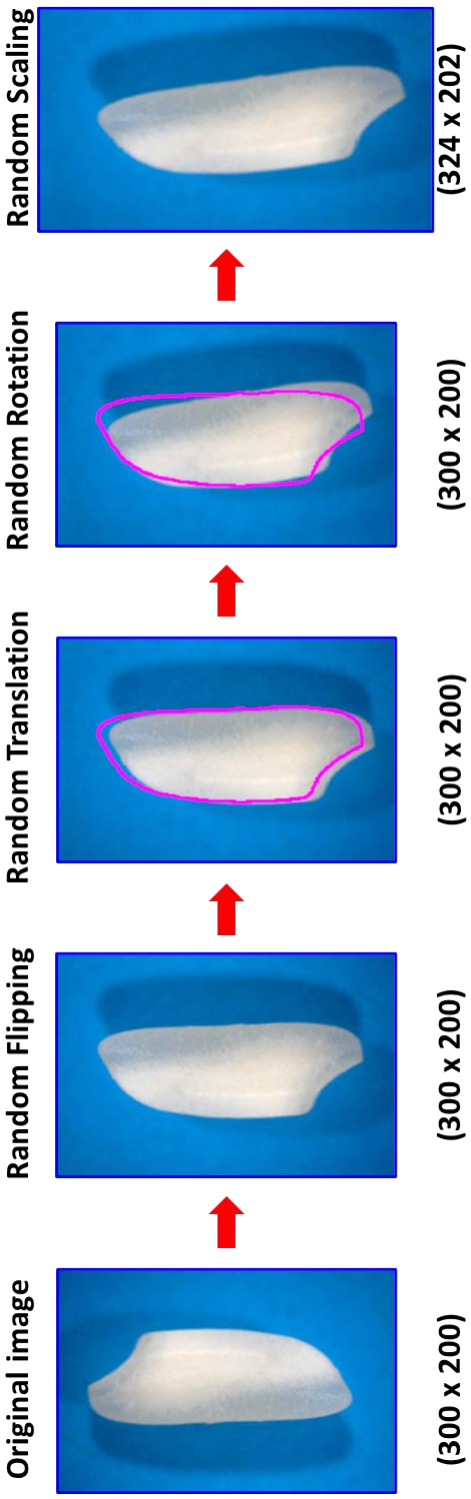


Fig 2.11: Representative images of half chalky with various data augmentation techniques applied to it.

Chapter 3

Experiments

3.1. Experimental setup

All the experimental studies have been conducted in Google cloud environment on a 64-bit Debian GNU/Linux operating system running on Intel (R) Xeon (R) CPU @ 2.20 GHz and 26 GB RAM with NVIDIA Tesla P100 having 16 GB memory. All the code is implemented in Python. Keras framework Keras, 2021 with the Tensorflow (TensorFlow, https://www.tensorflow.org/. Accessed 11 Sept., 2021) backend was used for the proposed method.

3.2. Pre-processing

The total dataset of 8048 images is randomly split into train, validation, and test sets using stratified sampling without replacement, by 70%, 20%, and 10% respectively. The training dataset has the distribution of classes as follows: healthy (845), broken (729), normal damage (1103), full chalky (619), chalky discolored (775), half chalky (602), discolored (960).

The training data is imbalanced and data imbalance (Chawla, et al., 2004) has proven to be a problem in training deep learning models, making them biased towards the majority class. To tackle this, random oversampling has been used without replacement, making the number of points in all classes equal 1103. Finally, after oversampling, 7721 train images, 1618 validation images, and 797 test images were obtained. In the study, all the images are first resized to 300 × 200 pixels to reduce the training time of the models and be able to conduct feasible experiments. As part of pre-processing, these images first go through the data augmentation pipeline as mentioned in Section 2.4. Then, the pixel values of the images are rescaled by a factor of 1/255.

3.3. Training

All the models used in the study were pre-trained on the ImageNet (Russakovsky, et al., 2015) dataset. The pre-trained weights were only used as initializations and all the networks were trained fully on data used in this study.

The last fully-connected layer in each network was changed from 1000 to 7 since the images were being classified into seven categories. SoftMax activation was used in the final layer. The models were trained with categorical cross-entropy as the loss using the Adam Optimizer (Kingma and Ba, 2017). The parameters for the optimizer except the learning rate were set to the default ones as mentioned in the Keras documentation: $\beta1 = 0.9$, $\beta2 = 0.999$, $\varepsilon = 10^{-7}$. The batch size was set to 32 unless it did not fit into memory in which case it was reduced to 16. For every model, dropout was used before the last layer. The dropout rate used was 0.3 for ResNet50, and 0.2 for InceptionV2, MobileNetV2 and MobileNetV3. For EfficientNet, a drop-connect rate of 0.45 was used. The initial learning was set to 0.0001. The learning rate and the dropout rates were optimized using the validation set. On top of this we used a learning rate scheduler as well: For the first 10 epochs the learning rate was kept constant and then after the 10th epoch, it followed a step decay schedule, i.e. the learning rate reduced by a factor every epoch. The factor was set to $e^{-0.1}$ which rounds to 0.905. All the models were run for a maximum of 50 epochs. Early stopping was also used, monitoring validation score, with the patience of 25 epochs. Finally, for each model, the best epoch was selected as the one with the lowest validation loss. The corresponding instance of the model was taken to be the best instance and used for the final evaluation on the test set.

3.4. Performance metrics

A key objective of the study is to identify a model which can classify the surface damages in milled rice grains with a high accuracy, in a fast and cost-efficient manner. Since this is a multi-class classification task, as far as the correctness is concerned, the metrics used to compare the performance of different models are cross-entropy (loss function), accuracy, macro-averaged precision (Pr), macro-averaged recall (Re), and macro-averaged F1-score (F1). Accuracy represents the ratio of correctly classified images to the total number of images. Precision, Recall, and F1-score for each class have been described in equations (3.1) to (3.6) using the indices such as True Positive (TP), False Positive (FP), and False Negative (FN). Here, TP for a class k i.e., TP(k) represents the total

number of images correctly classified as class k. FN(k) gives the number of misclassified images belonging to class k. FP(k) gives the number of misclassified images predicted to be in class k. Precision (P_r) is defined as the number of true positives over the number of true positives plus the number of false positives and is represented by equation 3.1. Recall (R_e) is defined as the number of true positives over the number of true positives plus the number of false negatives and is represented by equation 3.2. F1-score is defined as the harmonic mean of precision and recall and is represented by equation 3.3. The macro-averages of metrics represent the average of all these metrics calculated over all the classes. The macro-averages of Precision, Recall, and F1-score have been described in equations 3.4, 3.5 and 3.6 respectively. A detailed discussion on these metrics can be found in Sokolova, et al. (Sokolova and Lapalme, 2009). Scikit-learn library (Scikit-Learn: Machine Learning in Python — Scikit-Learn 1.0.2 Documentation. https://scikit-learn.org/stable/. Accessed 27 Jan., 2022) has been used for the metric implementations. Besides these, the model size, number of parameters, and the average time for prediction are also compared for practical feasibility. Further, a confusion matrix with precision and recall for each class is presented for the best performing model.

$$P_r(k) = \frac{TP(k)}{TP(k)+FP(k)} \tag{3.1}$$

$$R_e(k) = \frac{TP(k)}{TP(k) + FN(k)} \tag{3.2}$$

$$F1(k) = \frac{2\ Pr(k)\ .\ Re(k)}{\mathrm{Pr}(k) + Re(k)} \tag{3.3}$$

$$Macro\ Avg\ Pr = \frac{1}{\mathrm{N}}\sum_{k=1}^{N} Pr(k) \tag{3.4}$$

$$Macro\ Avg\ Re = \frac{1}{\mathrm{N}}\sum_{k=1}^{N} Re(k) \tag{3.5}$$

$$Macro\ Avg\ F1 = \frac{1}{\mathrm{N}} \sum_{k=1}^{N} F1(k) \qquad (3.6)$$

Chapter 4

Results and Discussion

This section discusses the results of the experiments conducted in detail. First, a comparison of all the models is presented based on metrics discussed in section 3.4. Further, an error analysis section is presented discussing the misclassified images as well as the images which were predicted with very low confidence.

4.1. Comparison of methods

In comparison to accuracy, loss is not a percentage. It represents the total number of mistakes/errors committed for every example in the training or validation set. The performance of the model is determined by the loss, which is calculated on the training and validation set. A deep learning model's performance on the validation set is measured using a metric called validation loss. The validation set is a part of the data set aside to check the model’s performance. After every epoch, the validation loss is also measured. This indicates whether or not the model needs additional fine-tuning or modifications. A learning curve plot for the validation loss is employed to do this. Ideally, one would anticipate a decrease in loss after each, or several iterations.

A model's accuracy is often assessed after its parameters have been learned, fixed, and no further learning is occurring. Then the test sets are passed to the model. After comparing to the actual targets, the number of errors the model makes are reported. The rate of misclassification is then determined. For instance, if there are 100 test samples and the model properly classifies 95 of them, then its accuracy is 95%.

As discussed before, all the models were trained for a maximum of 50 epochs. Fig. 4.1 and Fig. 4.2 show the graph of validation loss and validation accuracy w.r.t the epoch for various models.

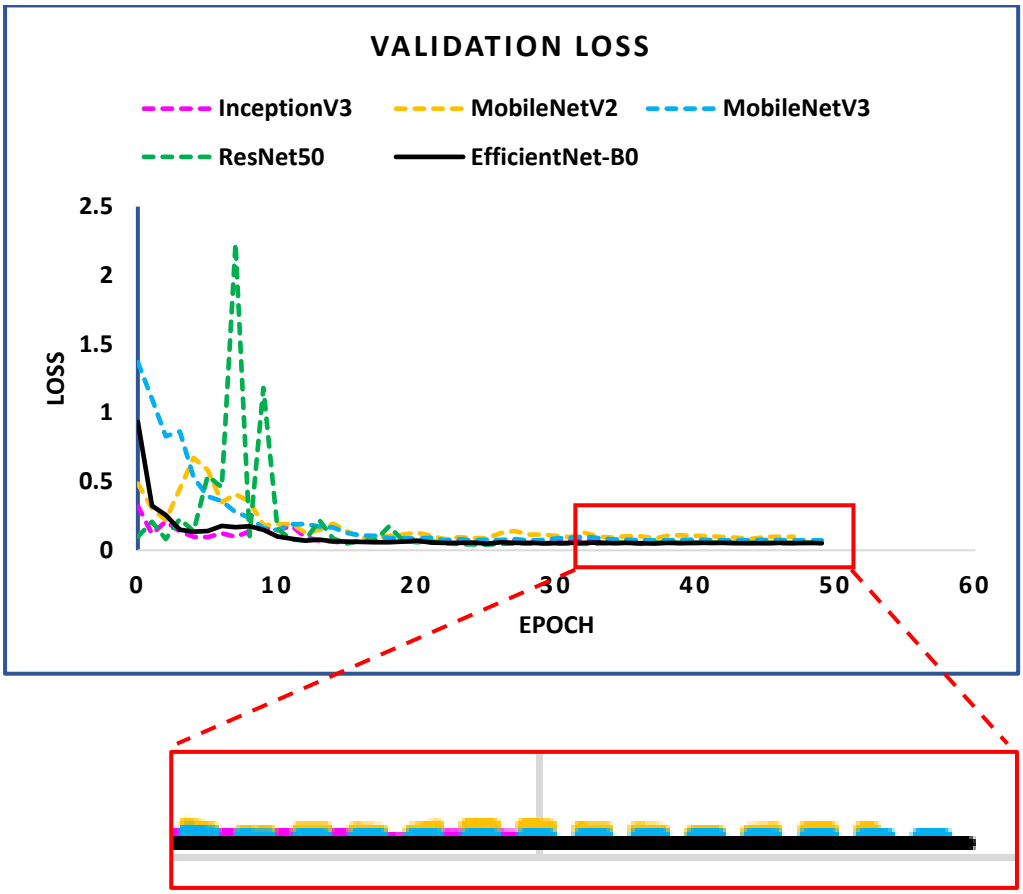


Fig 4.1: Graph of validation loss for different models.

ResNet-50 suffers from a couple of peaks because of a higher initial learning rate, but it becomes flat as the number of epochs increases. In case of validation loss, the enlarged view clearly shows that EfficientNet has minimum loss among all the models, followed by ResNet. In the case of validation accuracy, ResNet50 is having the highest accuracy, closely followed by EfficientNet. Further, all the metrics are reported on the test dataset containing 797 total images. Fig. 4.3 lists accuracy, macro- averaged precision, macro-averaged recall, and macro-averaged F1-score for all the five models used. The chart clearly represents the case where the best value was obtained for the respective performance criteria. As seen in Fig. 4.3, every model was able to achieve an accuracy of over 97.37%, which was quite decent given the complexity of the task. EfficientNet models outperform all previous models both in terms of the number of parameters and accuracy when applied to the ImageNet dataset. The

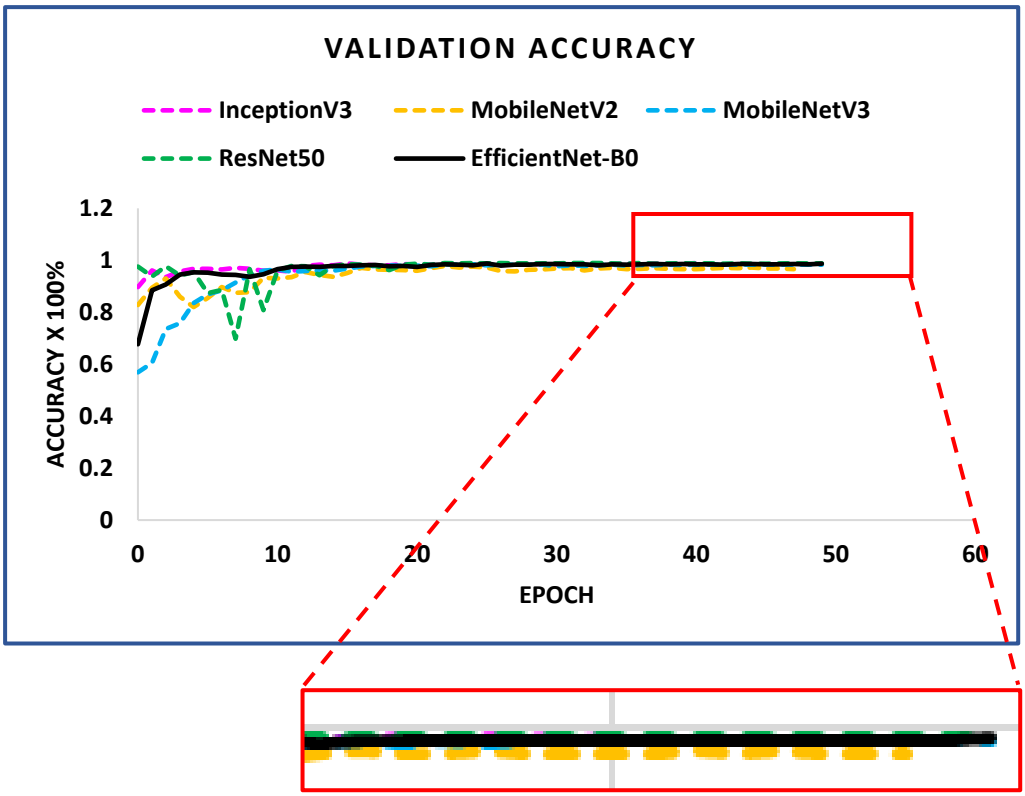


Fig 4.2: Graph of validation accuracy for different models.

same can be seen in present work, the best performing model out of all of them was EfficientNet-B0, and that too in all respects. The next best model was ResNet50, which achieved the same accuracy of 98.37% as EfficientNet but could not match it in all the other metrics. MobileNetV2, MobileNetV3 and InceptionV3 performed decently with 97.62%, 97.37%, and 97.62% as their respective accuracies. MobileNetV3 had the lowest accuracy of them all.

Fig. 4.4 compares the models in terms of the model size, number of parameters, and average inference time. The inference time reported is the time taken by the model for prediction, averaged over 1000 iterations. For inference time we disabled GPU support and it was measured on CPU. MobileNetV2 is the most compact model of them all with 2.3 M parameters and model size of 26 MB. After that, MobileNetV3 and EfficientNet have very similar model sizes of 49 MB and 47 MB respectively. The model size increases considerably when we move to the last two models: 250 MB for InceptionV3 and 270 MB for ResNet50. ResNet50 is 5.74 times larger than EfficientNet in size. As far as

inference time is concerned, MobileNetV2 and MobileNetV3 performed the best with 0.093 s and 0.098 s respectively. For EfficientNet, the inference time came out to 0.122 s. ResNet50 had the slowest time of 0.227 s.

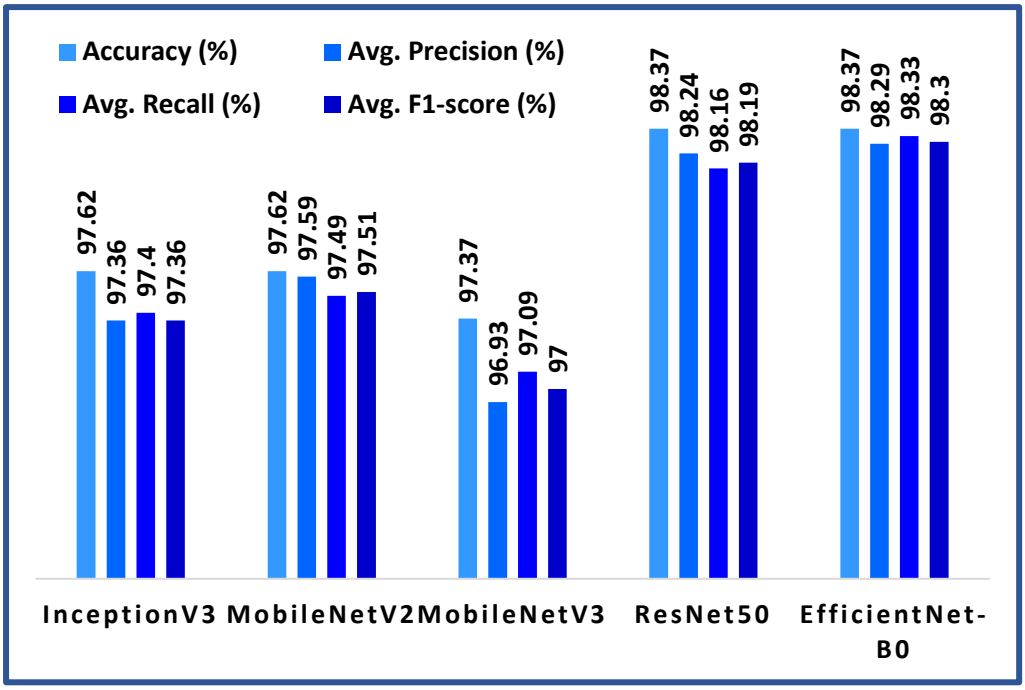


Fig 4.3: Model comparison based on some parameters.

So, in terms of compactness and inference time, MobileNetV2 performed the best. EfficientNet is not far off and has a compact size of 47 MB comparable to MobileNetV3 and a decent inference time as well. ResNet50 proved to be very bulky with the highest inference time. So overall in terms of performance on the classification task, EfficientNet was the best performing model with ResNet50 as the next best. Then in terms of model compactness and inference time, the best model was MobileNetV2, but it was compromised in terms of performance. And the study aimed to find an efficient, more compact, and faster model without any compromise on accuracy. Comparing EfficientNet with ResNet, both having similar accuracy, we see that although they have the same accuracy, EfficientNet has better precision, recall, f1-score, and cross-entropy loss. Then it's 5.74 times smaller than ResNet in terms of size. And it's almost twice as fast as ResNet. Thus, out of all the models we find EfficientNet to be the best performing one and being efficient at the same time.

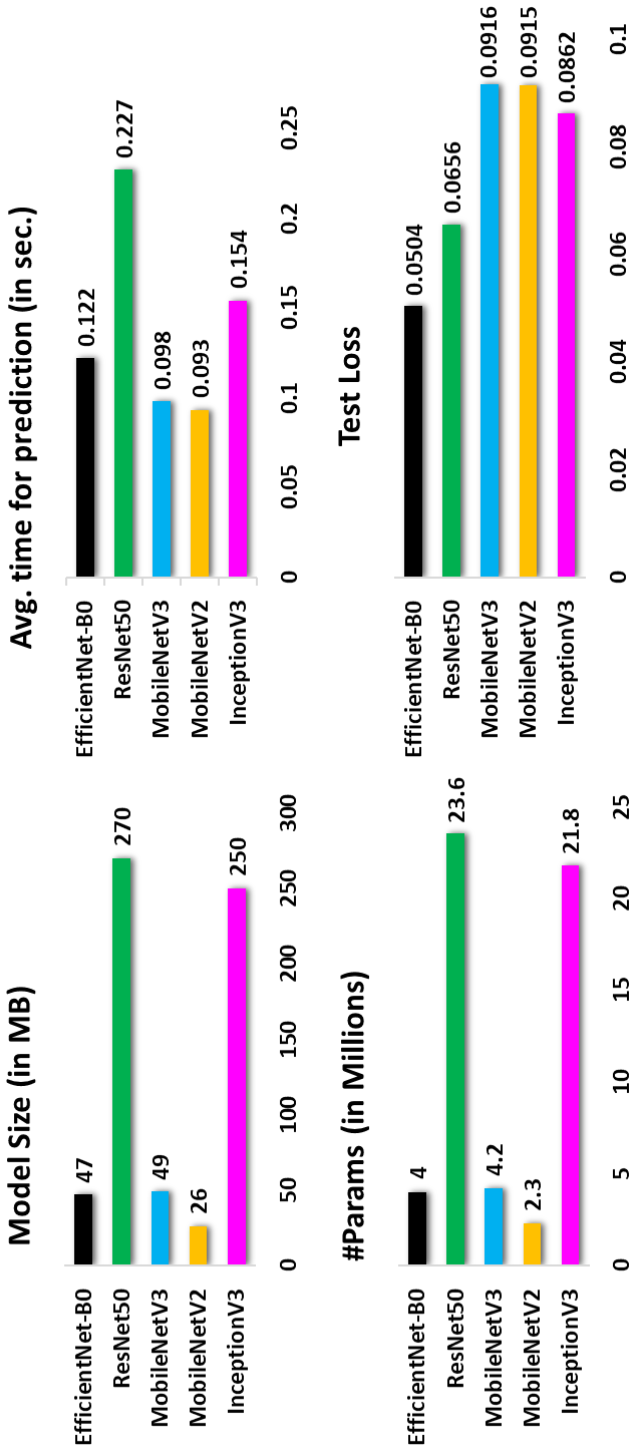


Fig 4.4: Model comparison based on other parameters.

In this study, the image resolution was fixed at 300 × 200. This number was determined experimentally through optimization by considering the trade-off between the improvement in performance versus an increase in the training/inference time. Models trained on images of lower resolution (<300 × 200) compromise accuracy, while the higher resolution images increase the inference time without a noticeable improvement in accuracy. One reason for this may be the overlapping damages in a single grain: there were always a few grains that the models could not predict accurately due to multiple damages existing in a single grain, each in prominence. As far as the confusion matrix for lower resolution images is concerned, the degradation in performance was mainly observed for the chalky classes i.e., full chalky, half chalky, and chalky discolored. The higher resolution images also took much longer to train and were not feasible for further experimentations as already the scope of improvement on the current dataset was very low. Hence, the shape 300 × 200 was chosen as it provided the best performance and enabled the conduction of feasible experiments.

4.2. Error analysis

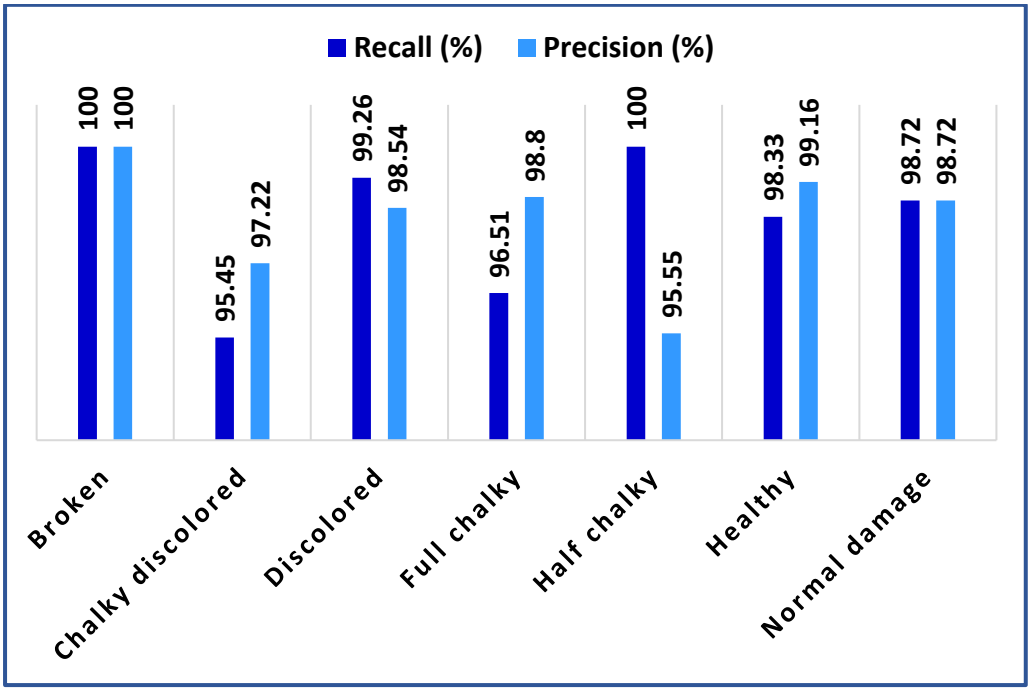


Fig 4.5: Precision and Recall values for each class.

This section presents the error analysis on the predictions made by the best performing model i.e., EfficientNet-B0. The confusion matrix is presented in Fig. 4.6 that provides an overview of all the class predictions. The true labels are marked on the vertical axes and the predicted labels on the horizontal axes, which were used to calculate the precision and recall values for the individual classes. Fig. 4.5 compares the precision and recall values for each class. The recall values of individual classes signify the individual class accuracy. The recall for a class k represents the proportion of images that the model captured (or classified) correctly out of all the images that belonged to class k. Whereas, precision for a class k is the proportion of the correctly classified images out of all the images that the model predicted to be in class k. It can be seen that full chalky and chalky discolored had the least accuracy (or recall) of 96.51% and 95.45% respectively. Half-chalky and broken had a perfect accuracy of 100%. Healthy, discolored, and normal damaged categories had an accuracy of 98.33%, 99.26%, and 98.72% respectively.

With regards to the precision values, half chalky had the lowest precision of 95.55%. This means that out of the misclassified points, a large proportion was classified as half chalky. Broken had a perfect precision of 100%, meaning that none of the other categories of grains were misclassified as healthy or broken.

The Deep CNN models that were trained; all use a SoftMax activation in the last layer. The SoftMax layer provides a probability distribution over all the classes and represents the confidence with which the model makes the prediction for each class. Table 1 shows the number of points that the model predicts for each class with confidence greater than a particular threshold. For instance, in the second row, for chalky discolored damage, 105 predictions were correct out of a total of 108 total predictions. Then the number of images classified as chalky discolored with a confidence > 0.75 was 105. Similarly, for thresholds 0.9, 0.95, and 0.99, the number of predicted images with confidence > threshold is 101, 96, and 85. It is observed that most of the images were predicted with a confidence of > 0.75.

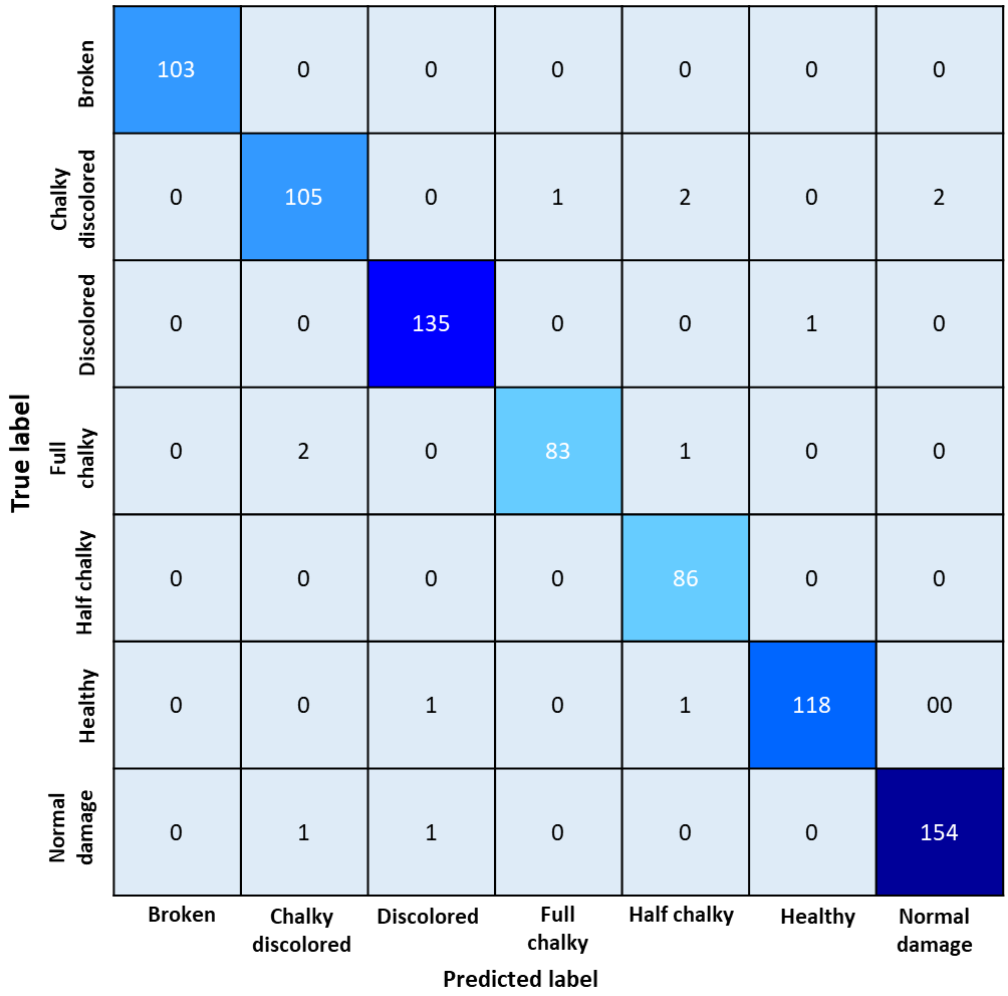


Fig 4.6: Confusion matrix for the test image dataset.

Further, as we increase the threshold, it is observed that the number of predictions reduce in number. This is expected because an increase in threshold demands more confidence from the model in its prediction. However, this is not uniform across all the damages. In broken, healthy, pin-damage, and discolored, the reduction in number is comparatively lesser i.e., the model is still able to predict a high percentage of images with a confidence > 0.99. However, in full-chalky, half-chalky, and chalkydiscolored the proportion of predictions with confidence > 0.99 are 84.5%, 78.9%, and 78.7%. This shows that the model is not as confident when predicting the chalky classes as it is in other ones.

Classifying various kinds of damages in rice is a tough job, there are borderline instances that are not simple to detect manually and anticipate the type of

damage, and these borderline cases are often misclassified. Chalky discolored may be misclassified with full chalky, half chalky, and normal damage. When discoloration is extremely light then it can be misclassified as full chalky or half chalky (Fig. 4.7-g), depending upon the kind of damage. Chalky discolored grains which are having darker patches are misclassified with normal damage (Fig. 4.7-b, c). Discolored grains which are having very little chalkiness but can't be seen by the human eye may be misclassified as chalky discolored. Full chalky may be misclassified with chalky discolored (Fig. 4.7-e) and half chalky (Fig. 4.7-a) for borderline instances. Half chalky may be misclassified with chalky discolored and full chalky. Healthy grains which are exhibiting very little chalkiness in a limited area may be misclassified as half chalky (Fig. 4.7-h). Normal damaged grains which are having a tiny area of damage on chalky discolored may be misclassified as chalky discolored (Fig. 4.7-d) and if the same occurs with discolored then it can be misclassified as discolored grains (Fig. 4.7-f). This explanation is validated by the help of the confusion matrix, where the diagonal elements indicate the correctly recognized grains and other numbers in that row indicate misclassified grains. For a given cell, true class is stated on the y-axis, and predicted class is mentioned on the x-axis. For EfficientNet-B0, the Confusion Matrix for the predictions on the test set is shown in Fig. 4.6.

Table 1: The number of images predicted for each damage with different confidence thresholds. Here "P" represents the output softmax activation for that particular damage class and signifies the confidence with which the model makes the prediction.

Predicted Damage	Total predictions	Correct predictions	P > 0.75	P > 0.9	P > 0.95	P > 0.99	Proportion of predictions with P > 0.99
Broken	103	103	102	102	102	101	98.1
Chalky discolored	108	105	105	101	96	85	78.7
Discolored	137	135	132	131	131	127	92.7
Full chalky	84	83	81	78	76	71	84.5
Half chalky	90	86	88	83	81	71	78.9
Healthy	119	118	119	118	117	116	97.5
Normal damage	156	154	156	155	153	148	94.9

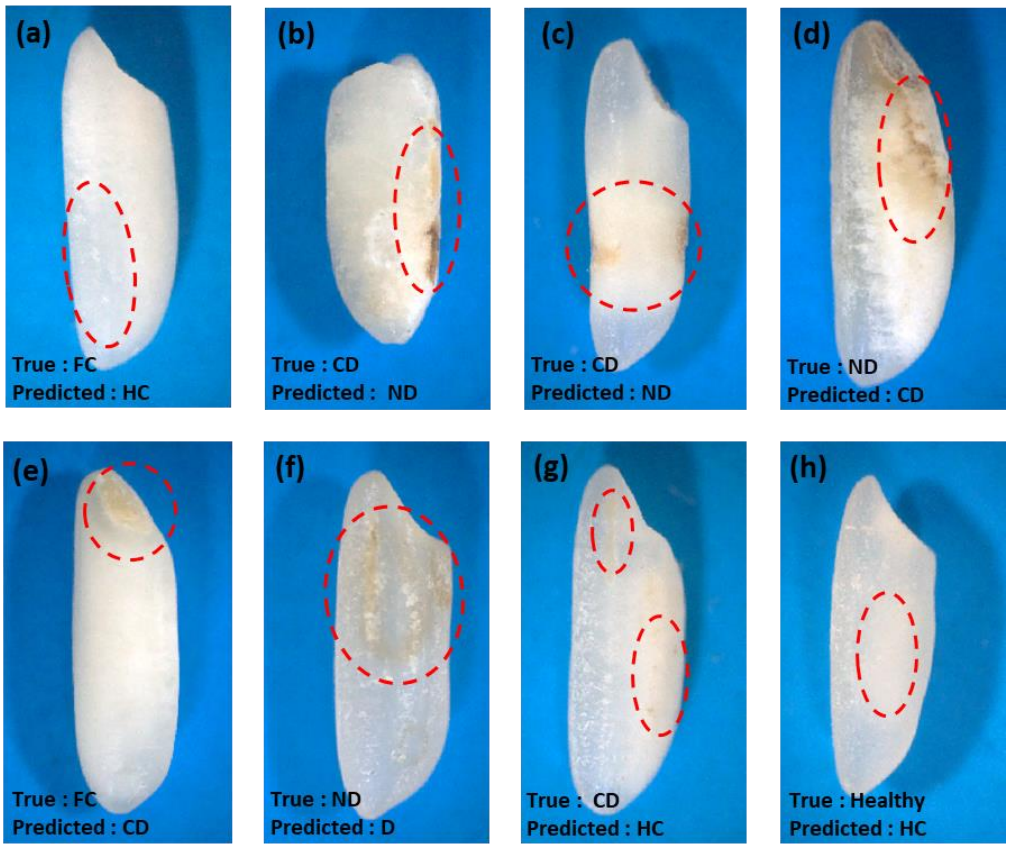


Fig 4.7: Misclassified images marked with the region of interest. FC (Full Chalky), HC (Half Chalky), CD (Chalky Discolored), ND (Normal Damage), D (Discolored).

Chapter 5

Conclusion and Future scope

Due to an unprecedented increase in the production of food grains to meet an ever-growing population, it is imperative to have a grain quality assessment tool that offers fast, reliable, and objective decision-making while being cost-effective. Therefore, a significant effort has been directed for developing machine vision systems (MVs), however, these efforts have been primarily focused on either sorting the healthy grains from the damaged ones or differentiating the rice grains based on their type. While from a point-of-view of nutrition, the focus remains on detecting and segregating healthy rice grains, in developing countries such as India the damaged rice grains constitute a significant amount. In this context, the use and distribution of damaged rice grains remain unstructured due to a lack of quality-assessment tools that can ascertain their market acceptability and quality with high accuracy. To this end, this study demonstrates the application of Deep CNN models that are fine-tuned to enable the classification of damage in rice grains with high accuracy. The following key conclusions can be drawn from this study:

1. A machine vision system is developed to create an in-house database of 8048 high-magnification (4.5 x) images of milled rice grains that are damaged and spread across seven damage types, namely, healthy, broken, full chalky, half chalky, chalky discolored, discolored, and normal damage.
2. The state-of-the-art Deep-learning based CNN algorithms, namely, EfficientNet-B0, ResNet-50, InceptionV3, MobileNetV2, and MobileNetV3 are fine-tuned and applied on the image dataset to enable damage classification across the seven classes. The performance of these five algorithms is compared in terms of recall, F1-score, precision, and their size and prediction time. It is demonstrated that EfficientNet-B0 is the best performing algorithm which provides an overall classification accuracy of 98.37 %, and the individual class accuracies of 96.51%, 95.45%, 100%,

100%, 98.33%, 99.26%, and 98.72% for normal damage full chalky, chalky discolored, half chalky, broken, healthy, discolored and normal damage, respectively.

3. The image dataset being a real dataset is reasonably complex due to the possibility of occurrence of more than one damage on a single rice grain, which complicates the process of damage classification. For instance, in half chalky class, the region with chalkiness varies from 10% to 90%, while in the chalky discolored grains, the discoloration intensity and its location vary from one grain to another. Similarly, in the normal damage class, the region with heat damage varies from 10% to 90%, while pin damage and watermark damages that fall under normal damage may conflict with the discolored damage class.
4. Despite the complexity of the dataset, the tuned EfficientNet-B0 model achieves a high overall classification accuracy of 98.37% at a nominally small model size of 47 MB and a prediction time of 0.122s. This demonstrates the resilience and fidelity of the model for damage classification, as well as the quality of the constructed image dataset.
5. The EfficientNet-B0 model can successfully categorize the chalky damage further into three subcategories i.e., half chalky, full chalky, and chalky discolored.

In most places, the quality of milled rice grains is being determined by trained personnel, but this is slow and inconsistent. In some places, flatbed scanners are used to predict the quality, but they don't give high fidelity results. Moreover, the conventional methods used nowadays are not enough to identify and quantify each damage type. Since the quality evaluation of rice grains is based on visuals of surface characteristics, it becomes a very good problem to solve through a machine vision system. To date, not a simple machine vision system has been used in a practical application for determining the rice quality, be it in terms of size, shape, length, or damage. This study can be used in the practical application of machine vision systems based on supervised learning, which can

be helpful in determining the quality and deciding its market value at different rice trading locations. The following work can be done as an extension of this study:

- To date, no one has used unsupervised learning to classify rice damages. So one can try using unsupervised learning to make clusters of these damages based on similarity and then classify them as supervised learning takes a lot of time and effort to create this labelled data.
- One can further sub-classify the normal damage class and hence it can help to decide its market value based on damage severity.
- A quality assessment product can be made that can sort some handful of grain samples and give a clear understanding of that particular lot.

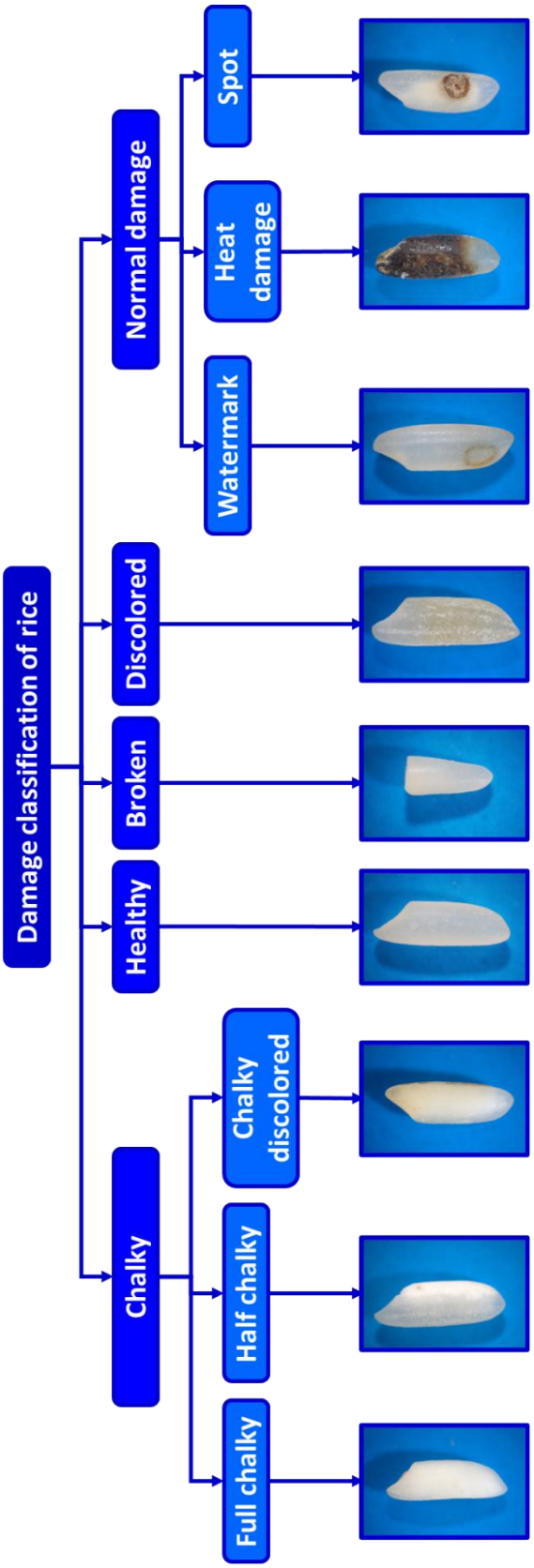


Fig 5.1: Classification chart of milled rice based on different damage types.

References:

- Aki, Ozan & Güllü, Aydın & Uçar, Erdem. "Classification of Rice Grains using Image Processing and Machine Learning Techniques". International Scientific Conference 20 – 21 November 2015, Gabrovo Bulgaria
- Applications of Deep Convolutional Neural Network in Computer Vision--《Journal of Data Acquisition and Processing》2016年01期. http://en.cnki.com.cn/Article_en/CJFDTOTAL-SJCJ201601001.htm. Accessed 27 Jan. 2022
- *Area, Yield, and Production*. https://ipad.fas.usda.gov/cropexplorer/util/new_get_psd_data.aspx?regionid=sasia. Accessed 7 May 2021.
- Aukkapinyo, K., et al. "Localization and Classification of Rice-Grain Images Using Region Proposals-Based Convolutional Neural Network." International Journal of Automation and Computing, vol. 17, no. 2, Apr. 2020, pp. 233–46. DOI.org (Crossref), https://doi.org/10.1007/s11633-019-1207-6
- Basu, S, and Miglani, A. 'Combustion and Heat Transfer Characteristics of Nanofluid Fuel Droplets: A Short Review'. International Journal of Heat and Mass Transfer, vol. 96, May 2016, pp. 482–503. DOI.org (Crossref), https://doi.org/10.1016/j.ijheatmasstransfer.2016.01.053.
- Chatnuntawech, Itthi, et al. "Rice Classification Using Spatio-Spectral Deep Convolutional Neural Network." ArXiv:1805.11491 [Cs], June 2019. arXiv.org, http://arxiv.org/abs/1805.11491
- Chawla, Nitesh V., et al. "Editorial: Special Issue on Learning from Imbalanced Data Sets." ACM SIGKDD Explorations Newsletter, vol. 6, no. 1, June 2004, pp. 1–6. June 2004, https://doi.org/10.1145/1007730.1007733
- Chen, Shumian, et al. "Colored Rice Quality Inspection System Using Machine Vision." Journal of Cereal Science, vol. 88, July 2019, pp. 87–95. DOI.org (Crossref), https://doi.org/10.1016/j.jcs.2019.05.010

- Chen, Yud-Ren, et al. “Machine Vision Technology for Agricultural Applications.” Computers and Electronics in Agriculture, vol. 36, no. 2–3, Nov. 2002, pp. 173–91. DOI.org (Crossref), https://doi.org/10.1016/S0168-1699(02)00100-X
- Ciresan D. C., Meier U., Masci J., Gambardella L. M., and Schmidhuber J. “Flexible, high performance convolutional neural networks for image classification,” in Proc. of 22nd Intl. Joint Conf. on Artificial Intelligence, 2011, pp. 1237–1242.
- Cireşan, Dan, et al. “Multi-Column Deep Neural Networks for Image Classification.” ArXiv:1202.2745 [Cs], Feb. 2012. arXiv.org, http://arxiv.org/abs/1202.2745
- Du, Cheng-Jin, and Da-Wen Sun. “Learning Techniques Used in Computer Vision for Food Quality Evaluation: A Review.” Journal of Food Engineering, vol. 72, no. 1, Jan. 2006, pp. 39–55. DOI.org (Crossref), https://doi.org/10.1016/j.jfoodeng.2004.11.017
- He, Kaiming, et al. “Deep Residual Learning for Image Recognition.” ArXiv:1512.03385 [Cs], Dec. 2015. arXiv.org, http://arxiv.org/abs/1512.03385
- Howard, Andrew, et al. “Searching for MobileNetV3.” ArXiv:1905.02244 [Cs], Nov. 2019. arXiv.org, http://arxiv.org/abs/1905.02244
- Howard, Andrew G., et al. “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.” ArXiv:1704.04861 [Cs], Apr. 2017. arXiv.org, http://arxiv.org/abs/1704.04861
- Huang, Gao, et al. “Densely Connected Convolutional Networks.” ArXiv:1608.06993 [Cs], Jan. 2018. arXiv.org, http://arxiv.org/abs/1608.06993
- Ibrahim, S., et al. “Contrastive Analysis of Rice Grain Classification Techniques: Multi-Class Support Vector Machine vs Artificial Neural Network.” IAES International Journal of Artificial Intelligence (IJ-AI), vol. 9, no. 4, Dec. 2020, p. 616. DOI.org (Crossref), https://doi.org/10.11591/ijai.v9.i4.pp616-622

- Idahosa, U., Basu, S., and Miglani, A. (April 11, 2014). "System Level Analysis of Acoustically Forced Nonpremixed Swirling Flames." ASME. J. Thermal Sci. Eng. Appl. September 2014; 6(3): 031015. https://doi.org/10.1115/1.4027297
- Idahosa, U., Santhosh, R., Miglani, A., and Basu, S. (November 11, 2015). "Response Dynamics of Recirculation Structures in Coaxial Nonpremixed Swirl-Stabilized Flames Subjected to Acoustic Forcing." ASME. J. Thermal Sci. Eng. Appl. March 2016; 8(1): 011008. https://doi.org/10.1115/1.4030728
- Ioffe, Sergey, and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift." ArXiv:1502.03167 [Cs], Mar. 2015. arXiv.org, http://arxiv.org/abs/1502.03167
- John, Jerin, et al. 'Rheology of Solid-like Ethanol Fuel for Hybrid Rockets: Effect of Type and Concentration of Gellants'. Fuel, vol. 209, Dec. 2017, pp. 96–108. DOI.org (Crossref), https://doi.org/10.1016/j.fuel.2017.06.124.
- Kaur, H. and Singh B. "Classification and Grading Rice Using Multi-Class SVM". International Journal of Scientific and Research Publications, Volume 3, Issue 4, April 2013 ISSN 2250-3153
- Keras: The Python Deep Learning API. https://keras.io/. Accessed 11 Sept. 2021
- Kingma, Diederik P., and Jimmy Ba. "Adam: A Method for Stochastic Optimization." ArXiv:1412.6980 [Cs], Jan. 2017. arXiv.org, http://arxiv.org/abs/1412.6980
- Krizhevsky, Alex, et al. "ImageNet Classification with Deep Convolutional Neural Networks." Communications of the ACM, vol. 60, no. 6, May 2017, pp. 84–90. DOI.org (Crossref), https://doi.org/10.1145/3065386
- LeCun, Y., et al. "Deep Learning." Nature, vol. 521, no. 7553, May 2015, pp. 436–44. www.nature.com, https://doi.org/10.1038/nature14539

- LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. (1998). Gradient-Based Learning Applied to Document Recognition. Proceedings of the IEEE. 86. 2278 - 2324. 10.1109/5.726791.
- LeCun, Y., Kavukcuoglu, K., Farabet, C., "Convolutional networks and applications in vision," in Proc. of the IEEE Intl. Symp. on Circuits and Systems, Jun. 2010, pp. 253–226.
- Lee, Chen-Yu, et al. "Deeply-Supervised Nets." ArXiv:1409.5185 [Cs, Stat], Sept. 2014. arXiv.org, http://arxiv.org/abs/1409.5185
- Li, Zewen, et al. "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects." IEEE Transactions on Neural Networks and Learning Systems, 2021, pp. 1–21. DOI.org (Crossref), https://doi.org/10.1109/TNNLS.2021.3084827
- Lin, P., et al. "Determination of the Varieties of Rice Kernels Based on Machine Vision and Deep Learning Technology." 2017 10th International Symposium on Computational Intelligence and Design (ISCID), IEEE, 2017, pp. 169–72. DOI.org (Crossref), https://doi.org/10.1109/ISCID.2017.208
- Lin, P., et al. "A Deep Convolutional Neural Network Architecture for Boosting Image Discrimination Accuracy of Rice Species." Food and Bioprocess Technology, vol. 11, no. 4, Apr. 2018, pp. 765–73. DOI.org (Crossref), https://doi.org/10.1007/s11947-017-2050-9
- Majumdar, S., and D. S. Jayas. "Classification of Bulk Samples of Cereal Grains Using Machine Vision." Journal of Agricultural Engineering Research, vol. 73, no. 1, May 1999, pp. 35–47. DOI.org (Crossref), https://doi.org/10.1006/jaer.1998.0388
- Mery, Domingo, et al. "Automated Design of a Computer Vision System for Visual Food Quality Evaluation." *Food and Bioprocess Technology*, vol. 6, no. 8, Aug. 2013, pp. 2093–108. *DOI.org (Crossref)*, doi:10.1007/s11947-012-0934-2.
- Miglani, A, et al. 'Insight into Instabilities in Burning Droplets'. Physics of Fluids, vol. 26, no. 3, Mar. 2014, p. 032101. DOI.org (Crossref), https://doi.org/10.1063/1.4866866.

- Miglani, A, et al. ‘Suppression of Instabilities in Burning Droplets Using Preferential Acoustic Perturbations’. Combustion and Flame, vol. 161, no. 12, Dec. 2014, pp. 3181–90. DOI.org (Crossref), https://doi.org/10.1016/j.combustflame.2014.06.010.
- Miglani, A, and Basu, S. ‘Coupled Mechanisms of Precipitation and Atomization in Burning Nanofluid Fuel Droplets’. Scientific Reports, vol. 5, no. 1, Dec. 2015, p. 15008. DOI.org (Crossref), https://doi.org/10.1038/srep15008.
- Miglani, A., and Basu, S. (October 1, 2015). "Effect of Particle Concentration on Shape Deformation and Secondary Atomization Characteristics of a Burning Nanotitania Dispersion Droplet." ASME. J. Heat Transfer. October 2015; 137(10): 102001. https://doi.org/10.1115/1.4030394
- Miglani, A & Nandagopalan, P & John, J & Baek, S. (2016). Disruptive Combustion Behavior of Gelled Ethanol Fuel droplets.
- Miglani, A, et al. ‘Oscillatory Bursting of Gel Fuel Droplets in a Reacting Environment’. Scientific Reports, vol. 7, no. 1, Dec. 2017, p. 3088. DOI.org (Crossref), https://doi.org/10.1038/s41598-017-03221-x.
- Nandagopalan, P, et al. ‘Shear-Flow Rheology and Viscoelastic Instabilities of Ethanol Gel Fuels’. Experimental Thermal and Fluid Science, vol. 99, Dec. 2018, pp. 181–89. DOI.org (Crossref), https://doi.org/10.1016/j.expthermflusci.2018.07.024.
- Nebauer, C. “Evaluation of convolutional neural networks for visual recognition,” IEEE Trans. on Neur. Netw., vol. 9, no. 4, pp. 685–595, 1998.
- Paliwal, J. et al. “Classification of Cereal Grains Using a Flatbed Scanner.” *2003, Las Vegas, NV July 27-30, 2003*, American Society of Agricultural and Biological Engineers, 2003. *DOI.org (Crossref)*, doi:10.13031/2013.15408
- Patel, Krishna Kumar, et al. “Machine Vision System: A Tool for Quality Inspection of Food and Agricultural Products.” *Journal of Food Science and Technology*, vol. 49, no. 2, Apr. 2012, pp. 123–41. *DOI.org (Crossref)*, doi:10.1007/s13197-011-0321-4

- Payman, S. H., et al. “Development of an Expert Vision-Based System for Inspecting Rice Quality Indices.” Quality Assurance and Safety of Crops & Foods, vol. 10, no. 1, Mar. 2018, pp. 103–14. DOI.org (Crossref), https://doi.org/10.3920/QAS2017.1109
- Pearson, T. “High-Speed Sorting of Grains by Color and Surface Texture.” Applied Engineering in Agriculture, vol. 26, no. 3, 2010, pp. 499–505. DOI.org (Crossref), https://doi.org/10.13031/2013.29948
- Prakash, Jatin, and Pavan Kumar Kankar. "Health prediction of hydraulic cooling circuit using deep neural network with ensemble feature ranking technique." Measurement 151 (2020): 107225.
- Prakash, Jatin, and Pavan Kumar Kankar. "Determining the working behaviour of hydraulic system using support vector machine." Advances in systems engineering. Springer, Singapore, 2021. 781-791.
- Prakash, Jatin, P. K. Kankar, and Ankur Miglani. "Internal Leakage Detection in a Hydraulic Pump using Exhaustive Feature Selection and Ensemble Learning." 2021 International Conference on Maintenance and Intelligent Asset Management (ICMIAM). IEEE, 2021.
- Putri, Tahir, et al. “Rice Grading using Image Processing.” *ARPN Journal of Engineering and Applied Sciences*, vol. 10, no 21, Nov 2015.
- Ranawat, Nagendra Singh, Pavan Kumar Kankar, and Ankur Miglani. "Fault Diagnosis in Centrifugal Pump using Support Vector Machine and Artificial Neural Network." Journal of Engg. Research EMSME Special Issue pp 99 (2021): 111.
- *Refractions in Raw & Parboiled Rice | Storage & Research | Divisions | Department of Food and Public Distribution, Government of India*. https://dfpd.gov.in/refractions-in-raw.htm. Accessed 7 May. 2021.
- Rehman, Tanzeel U., et al. “Current and Future Applications of Statistical Machine Learning Algorithms for Agricultural Machine Vision Systems.” Computers and Electronics in Agriculture, vol. 156, Jan. 2019, pp. 585–605. DOI.org (Crossref), https://doi.org/10.1016/j.compag.2018.12.006
- *Rice | Agricultural Marketing Service*. https://www.ams.usda.gov/book/rice. Accessed 7 May 2021.

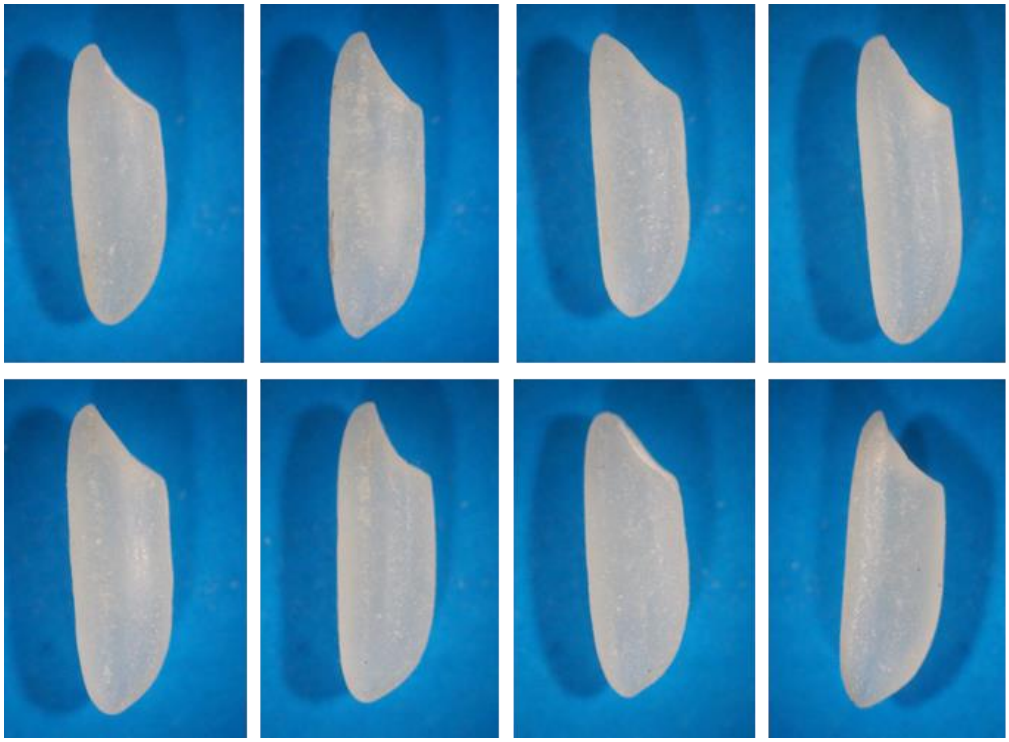
- Russakovsky, Olga, et al. “ImageNet Large Scale Visual Recognition Challenge.” International Journal of Computer Vision, vol. 115, no. 3, Dec. 2015, pp. 211–52. DOI.org (Crossref), https://doi.org/10.1007/s11263-015-0816-y
- Sampaio, P.S., et al. “Identification of Rice Flour Types with Near-Infrared Spectroscopy Associated with PLS-DA and SVM Methods.” European Food Research and Technology, vol. 246, no. 3, Mar. 2020, pp. 527–37. DOI.org (Crossref), https://doi.org/10.1007/s00217-019-03419-5
- Santhosh, R., et al. ‘Transition and Acoustic Response of Recirculation Structures in an Unconfined Co-Axial Isothermal Swirling Flow’. Physics of Fluids, vol. 25, no. 8, Aug. 2013, p. 083603. DOI.org (Crossref), https://doi.org/10.1063/1.4817665.
- Santhosh, R., et al. ‘Transition in Vortex Breakdown Modes in a Coaxial Isothermal Unconfined Swirling Jet’. Physics of Fluids, vol. 26, no. 4, Apr. 2014, p. 043601. DOI.org (Crossref), https://doi.org/10.1063/1.4870016.
- Sandler, Mark, et al. “MobileNetV2: Inverted Residuals and Linear Bottlenecks.” ArXiv:1801.04381 [Cs], Mar. 2019. arXiv.org, http://arxiv.org/abs/1801.04381
- Scherer, D., Muller, A., and Behnke, S. “Evaluation of pooling operations in convolutional architectures for object recognition,” in Proc. of the Intl. Conf. on Artificial Neural Networks, 2010, pp. 92–101.
- Scikit-Learn: Machine Learning in Python — Scikit-Learn 1.0.2 Documentation. https://scikit-learn.org/stable/. Accessed 27 Jan. 2022.
- Simonyan, Karen, and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition.” ArXiv:1409.1556 [Cs], Apr. 2015. arXiv.org, http://arxiv.org/abs/1409.1556
- Sokolova, Marina, and Guy Lapalme. “A Systematic Analysis of Performance Measures for Classification Tasks.” Information Processing & Management, vol. 45, no. 4, July 2009, pp. 427–37. DOI.org (Crossref), https://doi.org/10.1016/j.ipm.2009.03.002

- Szegedy, Christian, et al. “Going Deeper with Convolutions.” ArXiv:1409.4842 [Cs], Sept. 2014. arXiv.org, http://arxiv.org/abs/1409.4842
- Szegedy, Christian, et al. “Rethinking the Inception Architecture for Computer Vision.” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016, pp. 2818–26. DOI.org (Crossref), https://doi.org/10.1109/CVPR.2016.308
- Tan, Mingxing, et al. “MnasNet: Platform-Aware Neural Architecture Search for Mobile.” ArXiv:1807.11626 [Cs], May 2019. arXiv.org, http://arxiv.org/abs/1807.11626
- Tan, Mingxing, and Le, Quoc V. “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.” ArXiv:1905.11946 [Cs, Stat], Sept. 2020. arXiv.org, http://arxiv.org/abs/1905.11946
- TensorFlow, https://www.tensorflow.org/. Accessed 11 Sept. 2021.
- Vithu, P., and J. A. Moses. “Machine Vision System for Food Grain Quality Evaluation: A Review.” *Trends in Food Science & Technology*, vol. 56, Oct. 2016, pp. 13–20. *DOI.org (Crossref)*, https://doi.org/10.1016/j.tifs.2016.07.011
- Voulodimos, Athanasios, et al. “Deep Learning for Computer Vision: A Brief Review.” Computational Intelligence and Neuroscience, vol. 2018, Feb. 2018, p. e7068349. www.hindawi.com, https://doi.org/10.1155/2018/7068349
- Wan, Y.N. et al. “RICE QUALITY CLASSIFICATION USING AN AUTOMATIC GRAIN QUALITY INSPECTION SYSTEM.” Transactions of the ASAE, vol. 45, no. 2, 2002. DOI.org (Crossref), https://doi.org/10.13031/2013.8509
- Wee, Chong-Yaw, et al. “Sorting of Rice Grains Using Zernike Moments.” *Journal of Real-Time Image Processing*, vol. 4, no. 4, Nov. 2009, pp. 353–63. *DOI.org (Crossref)*, https://doi.org/10.1007/s11554-009-0117-1
- Weiss, Karl, et al. “A Survey of Transfer Learning.” Journal of Big Data, vol. 3, no. 1, May 2016, p. 9. BioMed Central, https://doi.org/10.1186/s40537-016-0043-6

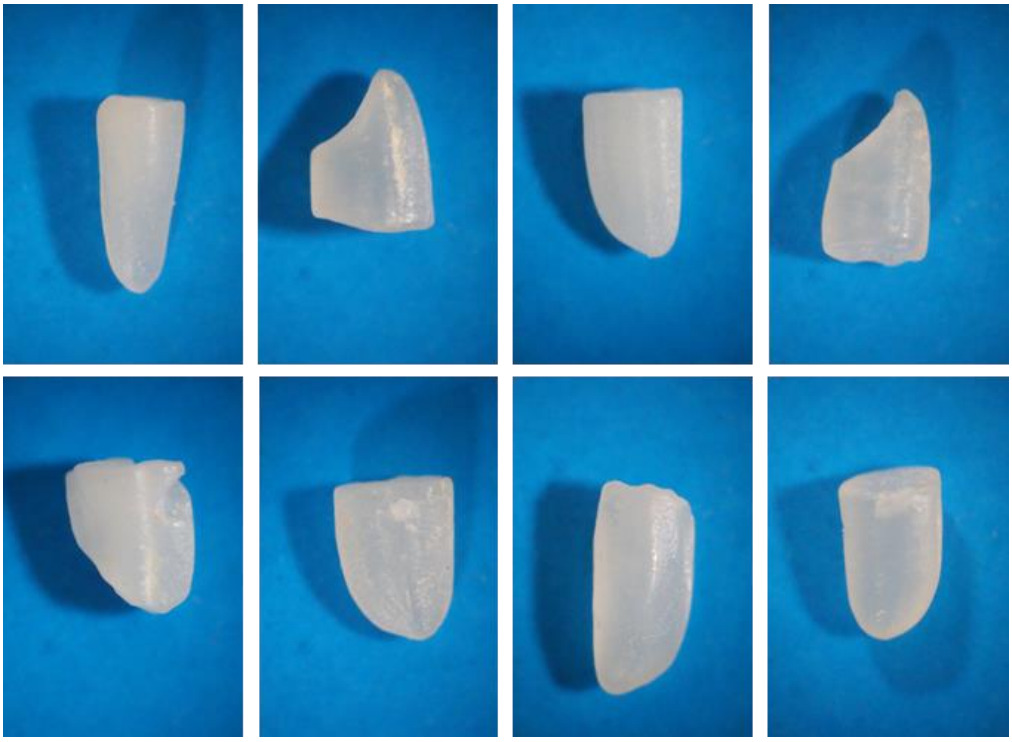
- Wu, Qing, et al. “The Application of Deep Learning in Computer Vision.” 2017 Chinese Automation Congress (CAC), IEEE, 2017, pp. 6522–27. DOI.org (Crossref), https://doi.org/10.1109/CAC.2017.8243952
- Yao, Sun, et al. “Inspection of rice appearance quality using machine vision.” Global Congress on Intelligent Systems.
- Zoph, Barret, et al. “Learning Transferable Architectures for Scalable Image Recognition.” ArXiv:1707.07012 [Cs, Stat], Apr. 2018. arXiv.org, http://arxiv.org/abs/1707.07012.

Appendix:

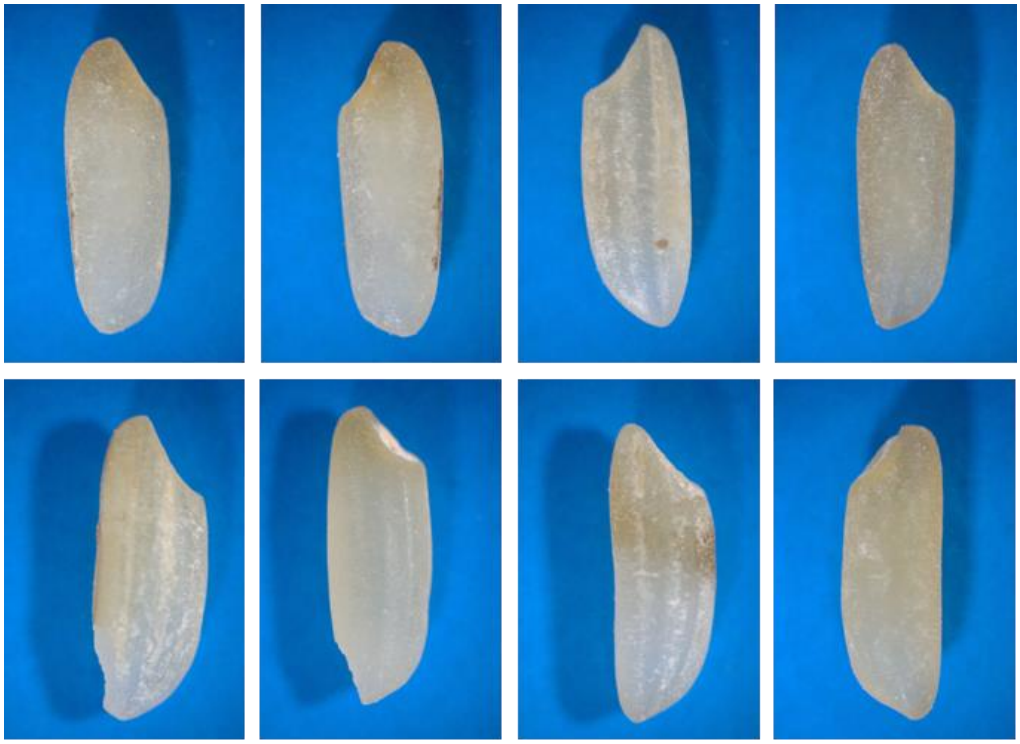
Class 1: Healthy



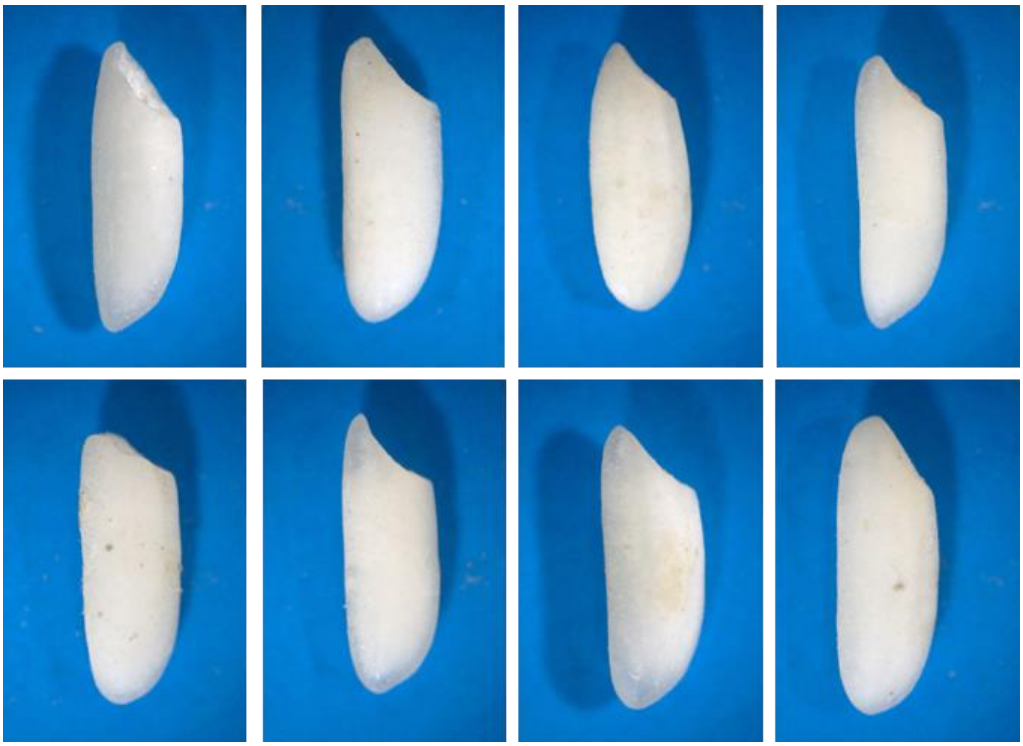
Class 2: Broken



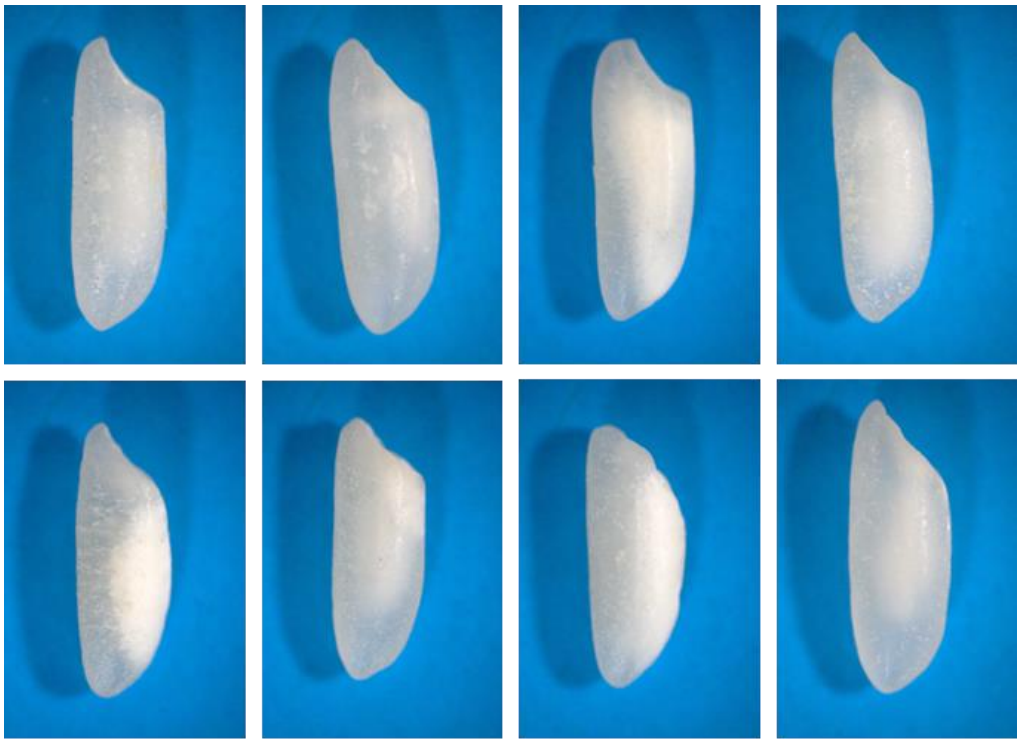
Class 3: Discolored



Class 4: Full chalky

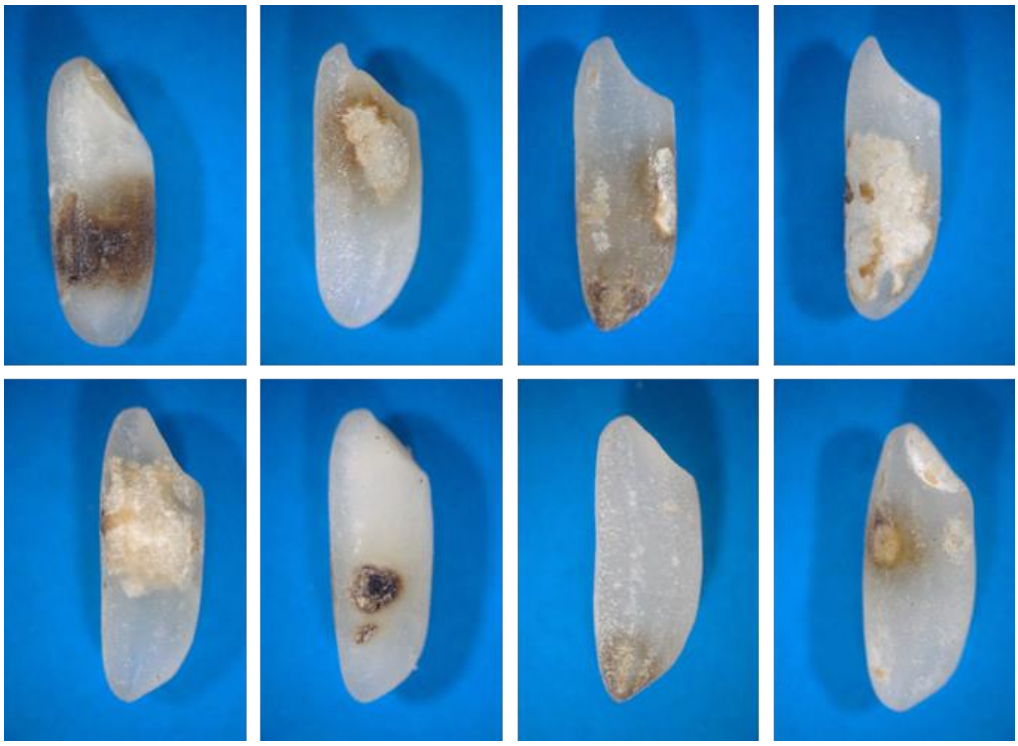


Class 5: Half chalky



Class 6: Chalky discolored

Class 7: Normal damage



Basic CNN architecture:

CNNs are a subclass of Deep Neural Networks that are frequently used for visual image analysis. CNNs can identify and categorise certain characteristics from images. The CNN is made up of three different kinds of layers: convolutional, pooling, and fully connected (FC) layers. A CNN architecture is created when these layers are layered. The dropout layer and the activation function, which are detailed below, are two additional crucial factors in addition to these three layers.

1. Convolutional Layer

This is the first layer, applied to extract the different characteristics from the input images. Convolution is a mathematical process that is carried out at this layer between the input image and a filter of a specific size, mxm. The dot product is obtained between the filter and the input image's components with regard to the filter's size by sliding the filter over the input image (mxm). The result is known as the feature map, and it provides details about the image, including its corners and edges. This feature map is later supplied to further layers to teach them more features from the input image.

2. Pooling Layer

A Pooling Layer often comes after a Convolutional Layer. Its main goal is to lower the convolved feature map's size in order to save on computational expenses. This is done individually on each feature map and by reducing the links between layers. There are several sorts of pooling operations, depending on the mechanism applied. The greatest component in Max Pooling is obtained from the feature map. The average of the components in a predetermined sized Image portion is determined via average pooling. Sum Pooling computes the total sum of the components in the designated section. Typically, the Pooling Layer acts as a link between the FC Layer and the Convolutional Layer.

3. Fully Connected Layer

To link the neurons between two layers, the Fully Connected (FC) layer, which also includes weights and biases, is utilised. These layers make up the final few levels of a CNN architecture and are often positioned before the output layer.

This process flattens the input image from the preceding layers and feeds it to the FC layer. The flattened vector is then sent through a few additional FC layers, where the standard operations on mathematical functions happen. The classification procedure starts to take place at this point.

4. Dropout

Normally, overfitting in the training dataset might result from all features being linked to the FC layer. When a given model performs so well on training data that it has a detrimental effect on the model's performance when applied to fresh data, this is known as overfitting. To solve this issue, a dropout layer is used, in which a small number of neurons are removed from the neural network during training, reducing the size of the model. A dropout of 0.3 causes 30% of the nodes in the neural network to be randomly removed.

5. Activation Functions

The activation function is one of the most crucial elements of the CNN model. They are employed to discover and approximate any type of continuous and complicated link between network variables. In layman's terms, it determines which model information should shoot ahead and which should not at the network's end. The network gains nonlinearity as a result. The ReLU, Softmax, tanH, and Sigmoid functions are a few examples of regularly used activation functions. Each of these operations has a particular use. Sigmoid and softmax functions are recommended for a CNN model for binary classification, while softmax is typically employed for multi-class classification.