Noname manuscript No.
(will be inserted by the editor)

Classification of Husk Species Using Convolutional Neural Networks for Improved Cattle Feed Formulation

Akshaj Pydimarri · Sanjeev Sharma

Received: date / Accepted: date

Abstract Automated cattle feed formulation is essential for optimizing the nutritional content of livestock feed while minimizing cost. This study aims to classify 8 different husk species from the BDHusk dataset, which are a crucial component of cattle feed, addressing a key aspect within the broader domain. This work employed a Convolution Neural Network (CNN) for classification tasks utilizing Xception, InceptionResNetV2, and ResNet152V2 pre-trained models. This work follows a systematic workflow to achieve optimum results. Data augmentation and preprocessing techniques are applied, and hyperparameters are tuned to our specific requirements. Performance is evaluated using Accuracy, Precision, Recall, and F1-score metrics. The InceptionResNetV2 model achieves impressive metrics of 99.17% accuracy, 99.18% precision, 99.17% recall, and 99.17% F1-score, demonstrating its robust capability in husk classification.

Keywords Husk Classification · Computer Vision · Transfer Learning · Convolution Neural Network · Xception · InceptionResNetV2 · ResnNet152V2 · Deep Learning

1 Introduction

Machine Learning and Deep Learning have revolutionized the livestock and animal sectors[38]. They have brought a paradigm shift in feed formulations

Akshaj Pydimarri
Indian Institute of Information Technology, Pune, India
E-mail: 112215144@cse.iiitp.ac.in

Sanjeev Sharma
Computer Science Engineering
Indian Institute of Information Technology, Pune, India
Survey No. -9/1/3. .Ambegaon Budruk Sinhgad Institute Road Pune, India 411041
E-mail: sanjeevsharma@iiitp.ac.in

by enabling precise, efficient, and cost-effective[5,40] nutritional strategies. ML models explore and evaluate alternative feed ingredients, such as insect protein or algae reducing the reliance on traditional feed sources. DL models can predict the nutrients within feed ingredients[31], ensuring balanced diets to meet precise dietary needs. They can also predict health issues[15] and recognize animal behaviour patterns[3]. ML/DL algorithms can be used for monitoring animal body condition through physical features such as internal-core and surface temperatures, respiration rate, sweating rate and weight[33].
Automated cattle feed formulation involves using computer algorithms to analyze the nutritional requirements of cattle, considering factors such as age, weight, breed, and production stage, and determining the optimal combination of ingredients and nutrients in livestock feed[46].
In previous studies, researchers have utilized linear programming [47] and stochastic programming procedures [45] to formulate minimal-cost cattle feed. This work focuses on classifying eight categories of husk species based on their visual appearance.
Husks, the outer layers of seeds or grains, are a significant component of many feed formulations[37,42,35]. They are fiber-rich, aid digestion, maintain rumen health, and serve as a cost-effective filler [13]. However, the nutritional value of husks can vary widely based on factors such as plant species, growing conditions, and processing methods. Also, their texture, color, and shape variability makes manual classification challenging and often leads to inaccuracies in feed formulation. To address this issue, this study explores the application of Convolutional Neural Networks (CNNs) to classify husks based on their visual appearance.
Convolutional Neural Networks (CNNs) excel at image classification because they can learn hierarchical features from images. They achieve this by identifying and sharing weights and processing small parts of images at a time, allowing them to detect patterns like edges and shapes [14]. This approach makes CNNs robust against changes in an object's position within an image. Furthermore, automating husk classification using Convolutional Neural Networks (CNNs) offers several advantages over manual classification, including speed, scalability, and the potential to uncover insights into husk characteristics that impact cattle nutrition.
In this study, data augmentation and data preprocessing plays a crucial role to prevent overfitting and enhance model's ability to generalize on unseen data. To construct an efficient and reliable automated method for husk classification, this study utilized three different transfer learning models named Xception, InceptionResNetV2, and ResNet152V2. These are highly effective pre-trained neural networks trained on the ImageNet dataset. By adding a custom head to this base model, these models were fine-tuned, specifically for our required task. Results were evaluated based on accuracy, precision, recall, and F1-Score, and relevant conclusions were drawn. The highest accuracy of 99.17% was achieved by the InceptionResNetV2 model.
The contributions of this paper are as follows:

- Preprocessing the data to improve generalization and prevent overfitting.
- Introducing three deep learning models: Xception, InceptionResNetV2, and ResNet152V2 for classifying husk species.
- Tuning various hyperparameters to optimize performance.
- Evaluating the models using accuracy, precision, recall, and F1-Score metrics.
- Analyzing the results and drawing meaningful conclusions.

This study represents a novel application of CNNs in agriculture, demonstrating the potential of artificial intelligence to revolutionize feed formulation practices. Automating husk classification using CNNs streamlines feed formulation processes, resulting in more consistent and balanced feed formulations, ensuring cattle and livestock's overall health and productivity.
The following sections detail this study. In the 2nd section, the Literature Review section, a brief history of pre-trained CNN models, along with their applications to different problem statements, especially in the agricultural sector, is provided. In the 3rd section, the Materials and Methods section, a thorough description of the dataset, pre-processing techniques, and a brief description of the architecture of the chosen pre-trained models - Xception, InceptionResNetV2, and ResNet152V2 are provided. In the 4th section, the Experiments and Results section, the detailed results and analysis of each model, along with loss and accuracy graphs and confusion matrices, are provided. In the 5th section, the Comparative Study section, the results of this study and previously conducted work on this data set are compared. In the 6th section, Conclusion and Future Scope section, the findings of this study are concluded, also specifying the future scope of this research.

2 Literature Review

The very first use of CNN for image classification dates back to 1989, when LeCun et al. proposed the first multilayered CNNs and successfully applied these large-scale networks to classify handwritten digits and zip codes [28].
The availability of open source data sets like ImageNet[11] along with advancements in GPU, activation functions(ReLu), Batch Normalization[23], optimization algorithms has led CNNs to deliver outstanding performance on image classification tasks. Deep convolutional neural networks like AlexNet[26], VGG[49], ResNet[20], Xception[10], MobileNet[21] etc with their exceptional accuracies have promoted researchers to find real-world application of image classifications and further employ transfer learning techniques.
The amount of literature in husk classification is near to zero, hence in this section, we specifically aim to study the research done in agricultural sector with the advancement of machine learning and deep learning with special emphasis on the use of CNN for image classification. The areas of usage of Machine Learning/Deep Learning include fruit counting and yield estimation,

plant recognition, disease detection, land cover classification, weed identification and crop type classification[25].
MT Chiu et al.[9] presented an agriculture-Vision, a large-scale aerial farmland image dataset for semantic segmentation of agricultural patterns.
H Lu et al.[30] proposed a custom Deep Convolutional Neural Network with 3 convolution layers and custom parameters for Land Cover classification at 2 regions - Pengzhou County and Guanghan County of Sichuan Province and concluded a best overall for the 3 experimental image 1, 2 and 3 being 91.7%, 88.1% and 88.2%.
M Carranza-García et al.[7] proposed a general CNN, with a fixed architecture and parametrization for Land Cover and Land Use Classification on hyperspectral and radar images and obtained an overall accuracy of 96.78% which outperforms SVM, KNN algorithm, Random Forest.
VK Shrivastava et al.[48] proposed as AlexNet archtecture as a feature extractor in combination with Support Vector Machine (SVM) as a classifier to rice plant disease classification on a smaller dataset of 619 images with an acuuracy of 91.93%.
R Ahila Priyadharshini et al.[2] proposed LeNet architecture for the maize leaf disease classification including 4 classes which is a part of the PlantVillage dataset and achieved an accuracy of 97.89%.
Ü Atila et al.[4] proposed EfficientNet B4 and B5 model to classify the plant leaf images of 39 classes in the PlantVillage dataset.The B5 model achieved 99.91% accuracy and 98.42% precision on the original dataset while the B4 model achieved 99.97% accuracy and 99.39% precision on the augmented dataset.
HS Abdullahi et al.[1] propsed a transfer learning based CNN architecture for estimating plant health on a maize plantation was used with an average prediction accuracy of 99.58%.
D Chen et al.[8] proposed ResNext101 architecture to classify 15 weed classes in the cotton production system and achieved a best overall F1-score of 98.93 $\pm$ 0.34%.
Mohd Anul Haq[19] proposed a CNN along with a LVQ (Learning Vector Quantization) layer for broadleaf and grass weeds detection in relation to soil and soybean crop and achieved an accuracy of 99.44%.
René A. Sørensen et al.[50] proposed a DenseNet model to detect the presence of Creeping thistle, perennial weed obtaining an accuracy of 94%.
Andres Milioto et al.[34] developed a classification system based on convolutional neural networks (CNNs) for weed detection on two datasets A - captured with the plants in an early growth stage and it is quite balanced in it's plants/weeds relation and B - which was captured 2 weeks later, in a more advanced growth stage and achieved an accuracy of 97.3% and 89.2% respectively on testing data.
A Nowakowski et al.[41] proposed two architectures VGG19 and GoogleNet for crop type mapping. The study was conducted on two datasets - Malawi and Mozambique representing two study areas. The crops chosen for Malawi dataset were Cassava, Groundnut, Maize, Sweet potatoes, and Tobacco where

as for Mozambique dataset were the crop types chosen for the classification are Cassava, Maize, and Rice which represent the majority crops in the respective areas. Overall accuracy reach up to 83% for the Malawi dataset and up to 90% for the Mozambique dataset.
H Yalcin et al.[53] proposed a custom CNN with hyperparameters and weight decay for classifying 16 types of plant species with an accuracy of 97.47% and compared it to a general SVM with polynomial and RBF kernels.
UO Dorj et al.[12] proposed a citrus counting algorithm using distance transform and marker-controlled watershed algorithms to obtain a co-relation coefficient of 0.93.
P Nevavuori et al.[39] proposed a self made CNN architecture similar to AlexNet with 6 convolution layers, custom hyper parameters and Adadelta training algorithm for crop yield prediction with a minimum mean absolute percentage error (MAPE) of 8.8%.
K Moses et al.[36] analysed the ability of EfficientNet-B0, ResNet-50, InceptionV3, MobileNetV2, and MobileNetV3 for damage classification of milled rice grains on a high-magnification dataset of 8048 images consisting of seven types of rice grain damages. They concluded EfficientNet-B0 is the best-performing CNN model with an overall classification accuracy of 98.32%.
Z Liu et al.[29] analyzed the different YOLO architectures for broken corn detection. They proposed an adjusted YOLO model(based on YOLO-tiny) with the implementation of a focal loss function. Though YOLOv3 model showed a better accuracy of 90.24% compared to the accuracy of 89.77% of the proposed model, the detection speed of the proposed model is 3 times faster than the standard YOLOv3.
Osipov et al.[43] introduced YOLOv4 for detection and utilized two methods for classifying root damage in sugar beet crops during mechanical harvesting: bag of visual words (BoVW) with a support vector machine (SVM) and VGG-16. They achieved a precision of 74% and recall of 70% with YOLOv4-tiny. The CNN classification accuracy reached 92.6%

Table1 shows the results of the papers mentioned in the literature review.

3 Materials and methods

This study followed a systematic workflow, as illustrated in fig1. It began with loading and preprocessing the dataset through normalization and augmentation. The dataset was then divided into training, validation, and testing sets. Three pre-trained models, Xception, InceptionResNetV2, and ResNet152V2, were proposed and applied. Each pre-trained model was customized by adding specific softmax layers. Hyperparameters were fine-tuned to enhance performance. The models were evaluated using various metrics such as accuracy, precision, recall, and F1-Score. The performance of each model was assessed, and a comparison was made to determine the best-performing model.

Table 1: Literature Review

Sr. No.	Author Name	Accuracy (%)	Description	Model
1	M Carranza-García[48]	96.78	Land Cover and Land Use Classification	General CNN
2	VK Shrivastava[48]	91.93	Rice plant disease classification	AlexNet + SVM
3	R Ahila Priyadharshini[2]	97.89	Maize leaf disease classification	LeNet
4	U Atila[4]	99.91	Plant leaf classification	EfficientNet B4 and B5
5	HS Abdullahi[1]	99.58	Plant health estimation	Transfer learning based CNN
6	D Chen[8]	98.93(F1-score)	Weed classification in cotton production system	ResNext101
7	Mohd Anul Haq[19]	99.44	Weed classification based on soil and crop	CNN+LVQ
8	Ren e A. Sørensen[50]	94	Creeping Thistle weed detection	DenseNet
9	Andres Milioto[34]	89.2 - 97.3	Weed detection at two growth stages of a plant	General CNN
10	A Nowakowsk[41]	83 - 90	Crop type mapping	VGG19 and GoogLeNet
11	H Yalcin[41]	97.47	Plant type classification	Custom CNN
12	UO Dorj[12]	0.93(co-relation co-efficinet)	Citrus Counting Algorithm	Distance transform and Marker-controlled watershed algorithms
13	P Nevavuori[39]	8.8(MAPE)	Crop Yield Prediction	custom CNN similar to AlexNet
14	K Moses[36]	98.32	Damage classification of milled rice grains	EfficientNet B0(highest),ResNet-50, InceptionV3, MobileNetV2, MobileNetV3
15	Z Liu[29]	89.77	Broken Corn Detection	Custom YOLO, study of YOLOV3
16	Osipov et al.[43]	92.6%(blurred),99%(clean)	Detection and Classification of root damage	YOLOv4, bag of visual words (BoVW) with a support vector machine (SVM) and VGG-16

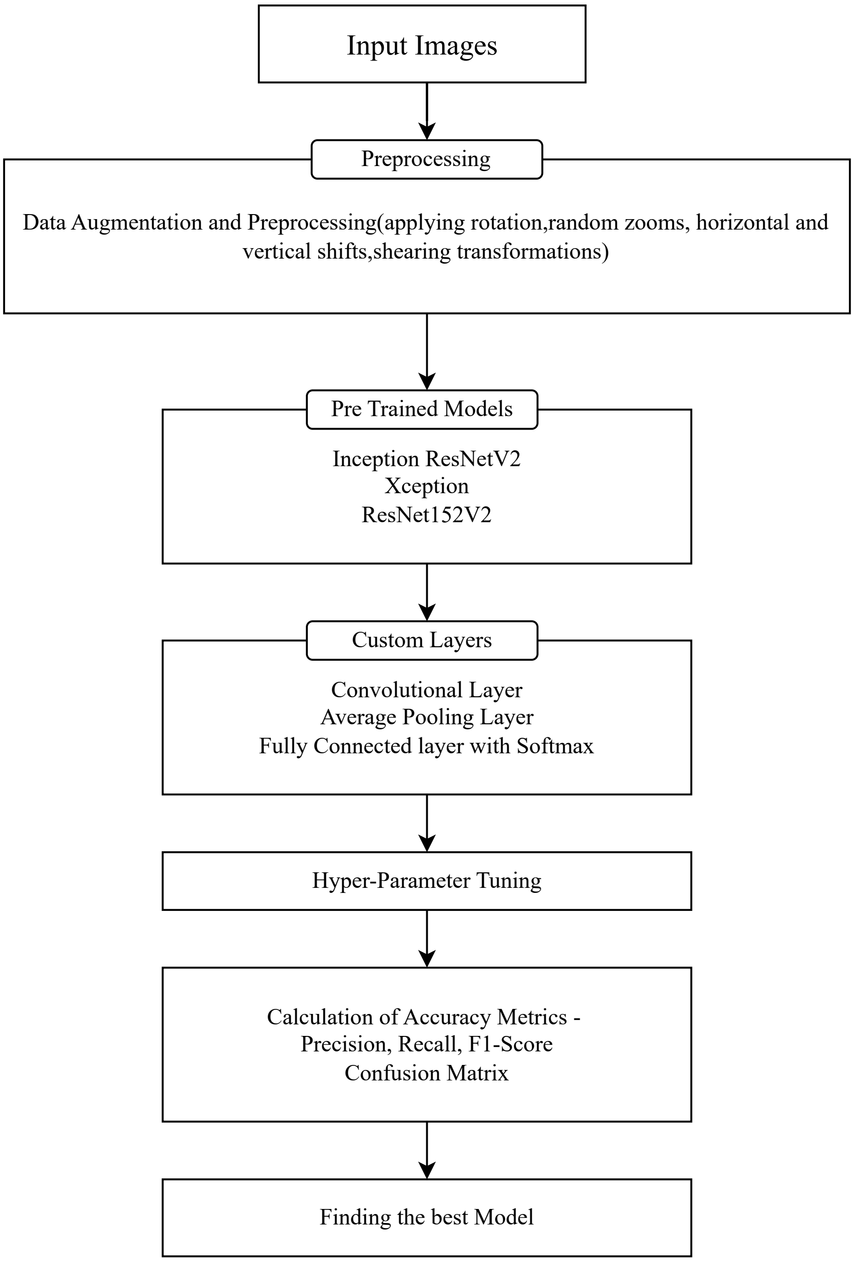


Fig. 1: Workflow Graph of this Study

3.1 Dataset Description

This study uses the dataset "BDHusk" which is publicly available[24]. The dataset consists of images collected from multiple locations in Sirajganj, Bangladesh. The dataset comprises 2,400 original images and 9,280 augmented images in two different folders containing 8 classes of images - Rice Husk (Oryza sativa), Corn Husk (Zea mays), Wheat Husk (Triticum aestivum), Chickpea Husk (Cicer arietinum), Lentil Husk (Lens culinaris), Soybean Husk (Glycine max), Grass Pea Husk (Lathyrus sativus) and Field Pea Husk (Pisum Sativum var. arvense L. Poiret). The augmented directory contains class-specific augmented images for researchers to use based on their specific requirements.
The original dataset comprises 300 images for each of the 8 classes. The augmented dataset comprises varying amounts of images each dataset, ranging from a minimum of 910 images in the Wheat Husk (Triticum aestivum) class to a maximum of 1360 images in the Soybean Husk (Glycine max) class.
Each image has varying dimensions represented by a tuple - (height of the image in pixels,width of the image in pixels,the number of color channels in the image).
The follwing image2 contains sub images belonging to 8 different husk classes.

3.2 Dataset Preprocessing

Data preprocessing is one of the most crucial steps for preparing raw data before feeding it to deep learning models. Proper data preprocessing helps improve model performance, reduces computational overhead and ensures the model's ability to generalize well to new, unseen data, i.e., prevent overfitting to training data through regularization[32]. This study has normalized and augmented data as a part of preprocessing. Data normalization is the process of rescaling the features of a dataset to have a mean of 0 and a standard deviation of 1. Data augmentation is manually creating images by modifying original images, generally by cropping, filtering, rotating, or flipping images. Data augmentation and normalization are applied to the images using ImageDataGenerator.
Data Normalization :- Pixel values in images have an 8-bit representation(ranging from 0-255 in decimal system). Therefore, the pixel values are rescaled to [0,1] by dividing with 255 in all training, validation and testing datasets.
Data augmentation :- Traditional transformations consist of using a combination of affine transformations to manipulate the training data[6].
A range of random rotations is applied as specified by the rotation range(up to 30 degrees). A range of random zooms applied as defined by zoom range (up to 0.15).Random horizontal and vertical shifts with the range controlled by width shift range and height shift range(up to 0.2 of image width and height, respectively). Random shearing transformations as determined by shear range(up to 0.15). Random horizontal flipping creates mirrored versions of the images

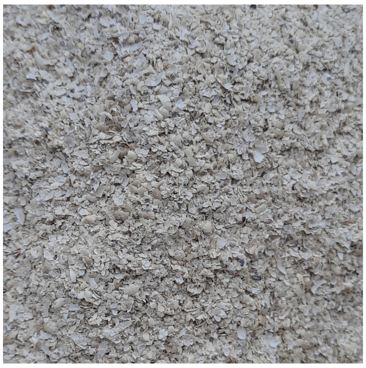
((a)) Rice Husk

((b)) Corn Husk

((c)) Wheat Husk

((d)) Chick Pea Husk

((e)) Lentil Husk



((f)) Soybean Husk

((g)) Grass Pea Husk

((h)) Field Pea Husk

Fig. 2: 8 different classes of husk

(as specified by horizontalflip=True). Empty areas created by transformations will be filled using the nearest available pixel value, with the filling mode set to "nearest" as specified.

3.3 Proposed Model

A Convolutional Neural Network (CNN) is a type of deep learning neural network architecture commonly used in Computer Vision. A CNN generally contains Convulutional layer, Pooling layer and a Fully Connected(Dense) layer. Convolutional layer applies a filter to the input data to identify features such as edges and textures. A filter is a two-dimensional array of weights that represents part of a 2-dimensional image. The filter applied to an input image calculates a dot product between the pixels[22], which is fed to an output array. The final output of all the filter processes is called the feature map. The CNN applies the ReLU (Rectified Linear Unit) transformation to each feature map after every convolution to introduce nonlinearity to the model.
Pooling layer reduces the dimensionality of input by applying a pooling operation (either a Max Pooling or Average Pooling)and increase the robustness of feature extraction[18]. Flattening Layer converts the 2D arrays from the pooling layer into a 1D vector, preparing the data for input into the fully connected layers.
Fully Connected layer performs classification tasks using the features that the previous layers and filters extracted. Instead of ReLu functions, the FC layer typically uses a softmax function that classifies inputs more appropriately and produces a probability score between 0 and 1.
Transfer learning in image classification involves using pre-trained CNN models on large datasets like ImageNet. By utilizing previously trained weights and biases[44], models are fine-tuned to achieve better performance with much less training time and computational resources. In this work, Xception, Inception-ResNetV2 and ResNet152V2 architectures have been used.

3.3.1 Xception

Xception, denoting "Extreme Inception," is an architecture known for its utilization of refined depth-wise separable convolutions. It features 36 convolutional layers which form the feature extraction base of the network. These layers are organized into 14 modules, each with linear residual connections around them, except for the initial and final modules. The main idea behind Xception is to split the convolution process into two stages: a depthwise convolution that applies a single filter to each input channel separately, followed by a pointwise convolution that combines the outputs of the depthwise convolution with a 1x1 convolution. This approach differs from Inception, which performs the 1x1 convolution first followed by channel wise spatial convolutions. Notably, depthwise separable convolutions in Xception are typically implemented without non-linearities[10].Fig.3 and Fig.4 show the architecture and model parameters of Xception architecture.

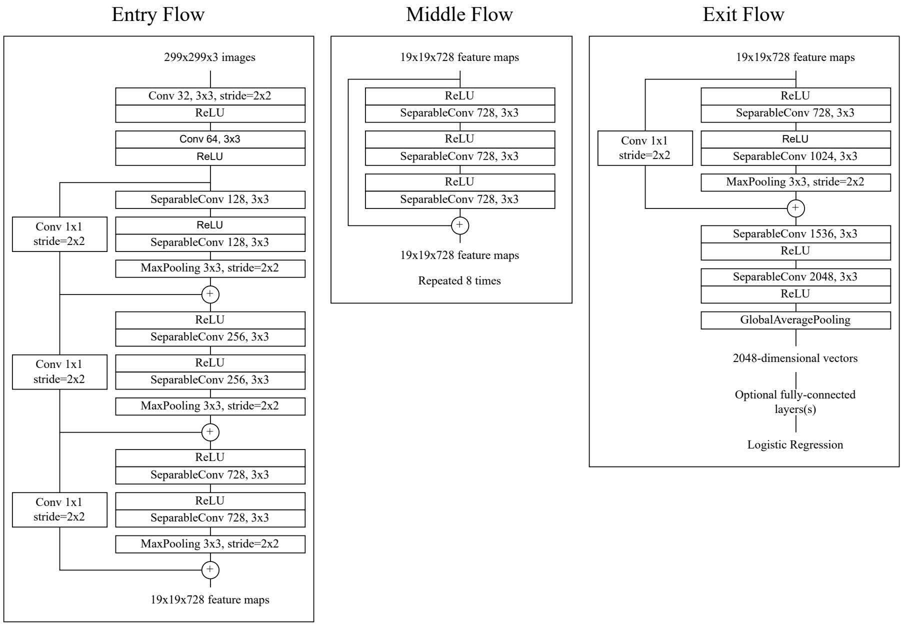


Fig. 3: Xception Architecture[10]

3.3.2 InceptionResNetV2

The InceptionResNetV2 architecture combines the concepts of the Inception architecture and residual learning from ResNet[20]. The key idea behind InceptionResNetV2 is to use the deep and wide inception modules while also incorporating residual connections.
The inception modules allow the network to capture complex features at different scales while ensuring efficient training. The architecture is a combination of all 1x1, 3x3 and 5x5 layers with their output filter banks concatenated into a single output vector forming the input of the next stage with an additional parallel pooling layer[52]. 1x1 convolutions are added whenever necessary to decrease the computational cost. Residual skip connections allow the network to bypass layers, helping to mitigate the vanishing gradient problem and enable the training of very deep networks.Fig.5 and Fig.6 show the architecture and model parameters of InceptionResNetV2 architecture.

Model: "functional_1"

Layer (type)	Output Shape	Param #
input_layer_1 (InputLayer)	(None, 128, 128, 3)	0
xception (Functional)	(None, 4, 4, 2048)	20,861,480
global_average_pooling2d (GlobalAveragePooling2D)	(None, 2048)	0
dense (Dense)	(None, 1024)	2,098,176
dense_1 (Dense)	(None, 8)	8,200

Total params: 22,967,856 (87.62 MB)
Trainable params: 22,913,328 (87.41 MB)
Non-trainable params: 54,528 (213.00 KB)

Fig. 4: Xception Model Parameters

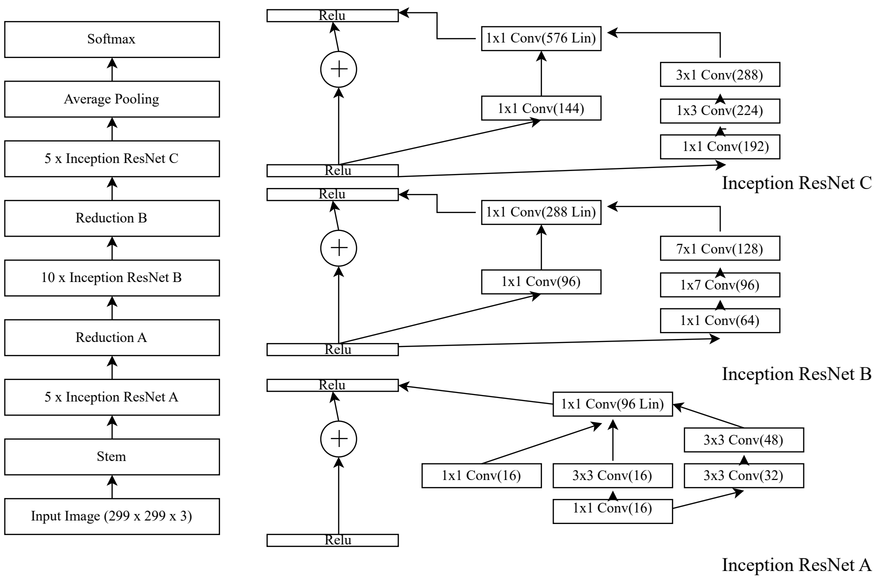


Fig. 5: InceptionResNetV2 Architecture

3.3.3 ResNet152V2

ResNet152V2 architecture belongs to the ResNet (Residual Neural Network) family which introduced the residual blocks idea to address the issue of the vanishing/exploding gradient(gradient becoming zero or being overly large)[16]. Residual blocks which contain skip connections allow layer activations to by-

Model: "functional_1"

Layer (type)	Output Shape	Param #
input_layer_1 (InputLayer)	(None, 128, 128, 3)	0
inception_resnet_v2 (Functional)	(None, 2, 2, 1536)	54,336,736
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1536)	0
dense (Dense)	(None, 1024)	1,573,888
dense_1 (Dense)	(None, 8)	8,200

Total params: 55,918,824 (213.31 MB)
Trainable params: 55,858,280 (213.08 MB)
Non-trainable params: 60,544 (236.50 KB)

Fig. 6: InceptionResNetV2 Model Parameters

pass certain levels and connect directly to subsequent layers. This creates residual or "leftover" blocks that are stacked together to form ResNets. The strategy behind this network is to let the network fit the residual mapping rather than have layers learn the underlying mapping[20].
ResNet152V2 specifically has 152 layers, making it one of the deepest variants of the ResNet architecture. ResNet-152v2 employs bottleneck building blocks, which consist of three convolutional layers: 1x1, 3x3, and 1x1 convolutions. The 1x1 convolutions are used to reduce and then restore dimensions, reducing computational complexity while maintaining representational power.Fig.7 and Fig.8 show the architecture and model parameters of InceptionResNetV2 architecture.

4 Experiments and Results

4.1 Hardware and Software Setup

NVIDIA P100 and 13GB RAM is used for training along with TensorFlow, Keras, and Scikit-learn libraries in Kaggle, coded in Python 3.10.12.

4.2 Training, Validation and Testing data

The dataset used in this study named BDHusk is spilt into 3 different sections. The first section is named training images, which consist of 80% of data. The second section is named validation images, which consist of 10% of data. The third section is named testing images, which consist of 10% of data. Table

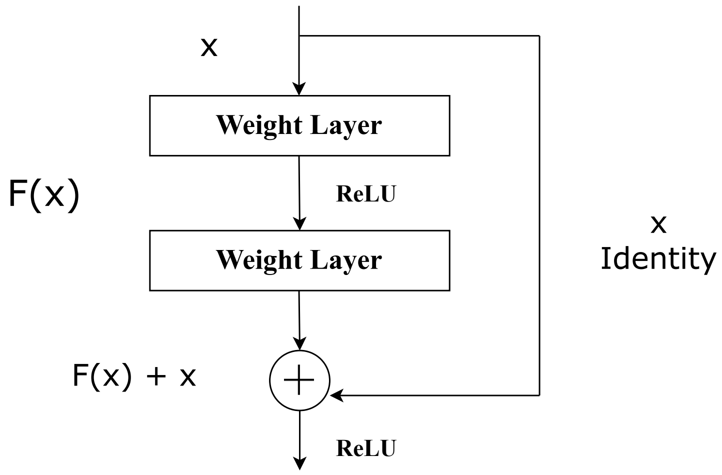


Fig. 7: Residual skip connections of the ResNet model[20]

Model: "functional_3"

Layer (type)	Output Shape	Param #
input_layer_3 (InputLayer)	(None, 128, 128, 3)	0
resnet152v2 (Functional)	(None, 4, 4, 2048)	58,331,648
global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 2048)	0
dense_2 (Dense)	(None, 1024)	2,098,176
dense_3 (Dense)	(None, 8)	8,200

Total params: 60,438,024 (230.55 MB)
Trainable params: 60,294,280 (230.00 MB)
Non-trainable params: 143,744 (561.50 KB)

Fig. 8: ResNet152V2 Model Parameters

2 shows the distribution of images of each species into training, testing, and validation datasets.

4.3 Evaluation Criteria

This study uses loss and accuracy evaluation metrics for the training process. Loss indicates how far the model's predictions are from the actual values. The lower the loss, the better the model. While, accuracy measures the proportion

Table 2: Dataset Splitting

Class Name	**Total Images**	**Training Images**	**Validation Images**	**Testing Images**
Rice Husk	300	240	30	30
Corn Husk	300	240	30	30
Wheat Husk	300	240	30	30
Chickpea Husk	300	240	30	30
Lentil Husk	300	240	30	30
Soybean Husk	300	240	30	30
Grass Pea Husk	300	240	30	30
Field Pea Husk	300	240	30	30

of correct predictions made by the model. It is calculated as the number of correct predictions divided by the total number of predictions.
The results in this study are measured on accuracy, precision, recall and F1 - score. Precision is the proportion of true positives (relevant items) among all the items returned by the model. Recall is the proportion of true positives (relevant items) successfully retrieved by the model[17]. The F1-score is a harmonic mean of precision and recall, providing a balanced measure of a model's performance. The formulas for precision, recall, F1-score, and accuracy use the following definitions: TP (True Positive), TN (True Negative), FP (False Positive), and FN (False Negative) and are mentioned below.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + TN + FP} \tag{4}$$

4.4 Hyperparameter Tuning

Two parameters exist in machine learning models: model parameters and hyperparameters. Model parameters are initialized and updated through the data learning process (e.g., the weights of neurons in neural networks). While hyperparameters are set before training an ML model because they define the model architecture[27]. This study tweaks batch size, learning rate, epochs, optimizer, and loss function hyperparameters.

- The model was run on 50 epochs with a batch size of 32.

- Optimizations algorithms improve the performance of the CNN-based models by optimizing the gradient descent algorithm. Adam Optimizer which is proved to have better performance is used[51]
- Softmax function is used in last layer as optimization function.
- Categorical Cross Entropy has been used to calculate the loss values of different models.

The respective values are specified in the below table3.

Hyperparameter	**Value**
Batch Size	32
Learning Rate	0.00001
Epochs	50
Optimizer	Adam
Loss	Categorical Cross Entropy

Table 3: Hyperparameters

4.5 Performance Analysis of Proposed Model

On training the model on the training data set and validating it on the validation data set, we have obtained certain results that are described in the table below4.

Table 4: Model Performance Metrics on Test Dataset

Sr. No.	**Model Name**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1 Score (%)**
1	Xception	97.92	97.98	97.92	97.90
2	InceptionResNetV2	99.17	99.18	99.17	99.17
3	ResNet152V2	97.92	98.00	97.92	97.91

On performing analysis in the proposed models, highest precision, recall and F1 - score are shown by InceptionResNetV2 while Xception and ResNet152V2 behave similarly with minimal changes.

4.5.1 Results for Xception

Fig. 9 and Fig. 10 show the training and validation accuracy and loss respectively over 50 epochs. Both the accuracy and loss graphs are fairly smooth, with loss decreasing and accuracy increasing. On completion of 50 epochs, maximum training accuracy and minimum training loss came out to be 98.26% and 0.0465, respectively, while maximum validation accuracy and minimum validation loss came out to be 99.58% and 0.0254, respectively. Fig11 depicts the confusion matrix for all the husk classes. Upon calculation on the test data, the accuracy came out to be 97.92%. The weighted average precision value for this model turned out to be 97.98%, the weighted average recall value for the same model was 97.92% and weighted average F1-Score of the same model was 97.90%.

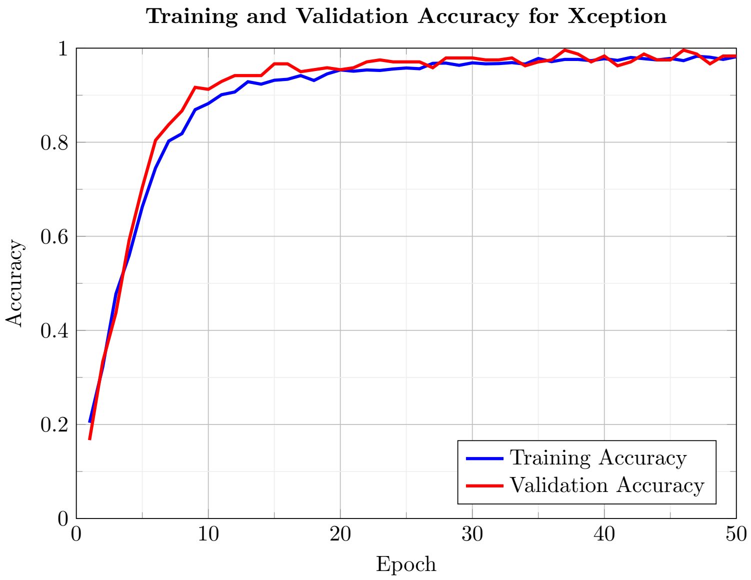


Fig. 9: Xception Accuracy

4.5.2 Results for InceptionResNetV2

Fig. 12 and Fig. 13 show the training and validation accuracy and loss respectively over 50 epochs. The accuracy graph is relatively smooth, showing an increase in accuracy with each epoch. The training loss graph is also smooth, consistently decreasing over time. However, the validation graph displays occasional spikes and dips. On completion of 50 epochs, maximum training accuracy and minimum training loss came out to be 98.52% and 0.0485, respectively, while maximum validation accuracy and minimum validation loss came out to be 99.17% and 0.0828, respectively. Fig14 depicts the confusion matrix for all the husk classes. Upon calculation on the test data, the accuracy came out to be 99.17%. The weighted average precision value for this model turned out to be 99.18%, the weighted average recall value for the same model was 99.17% and weighted average F1-Score of the same model was 99.17%.

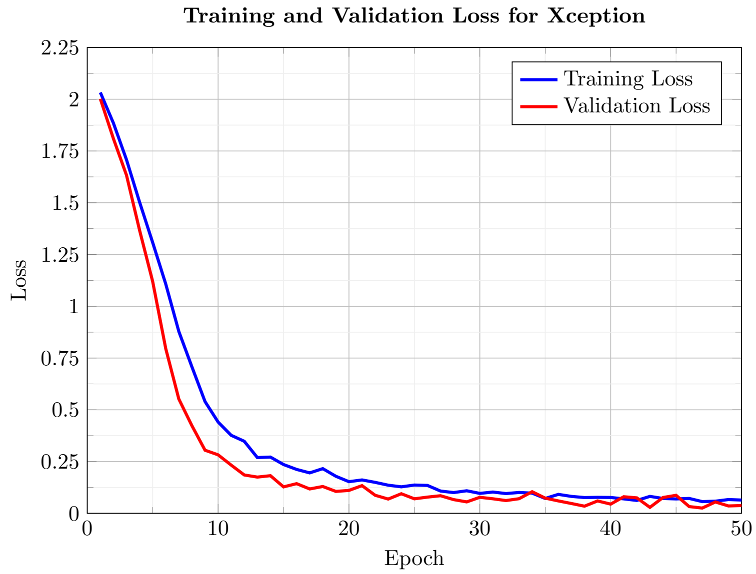


Fig. 10: Xception Loss

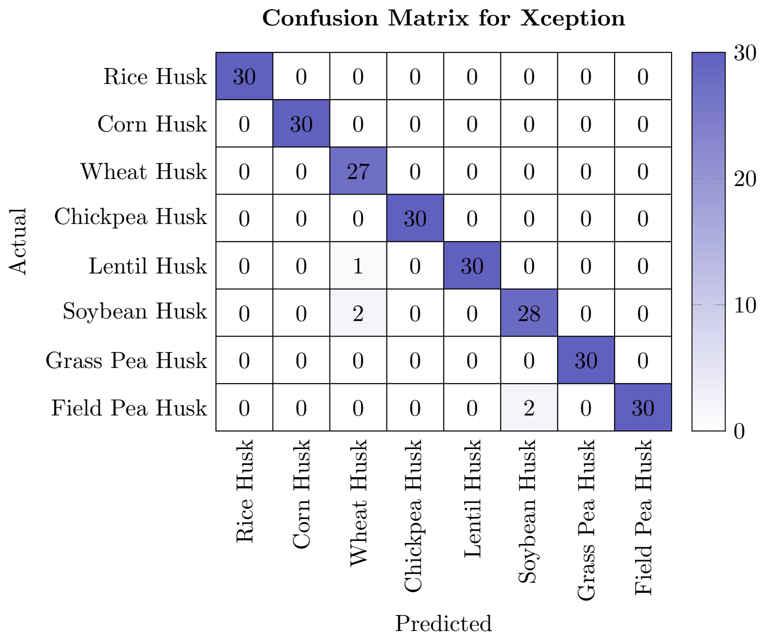


Fig. 11: Xception Confusion Matrix

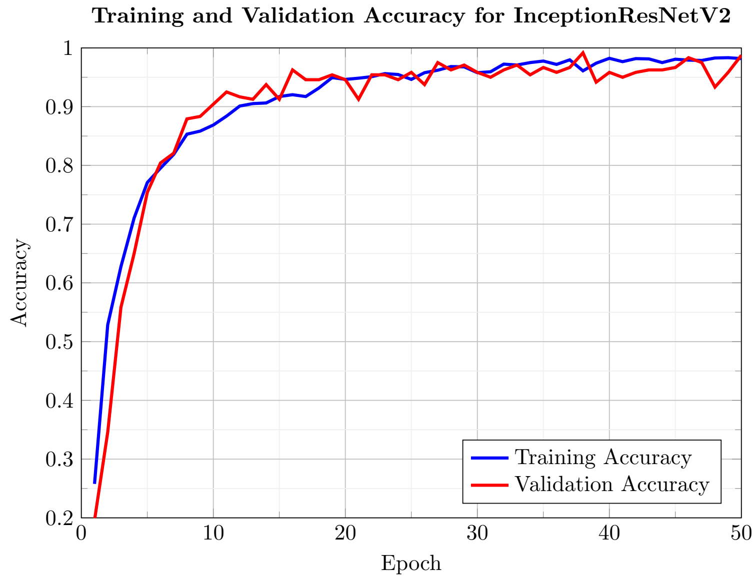


Fig. 12: InceptionResNetV2 Accuracy

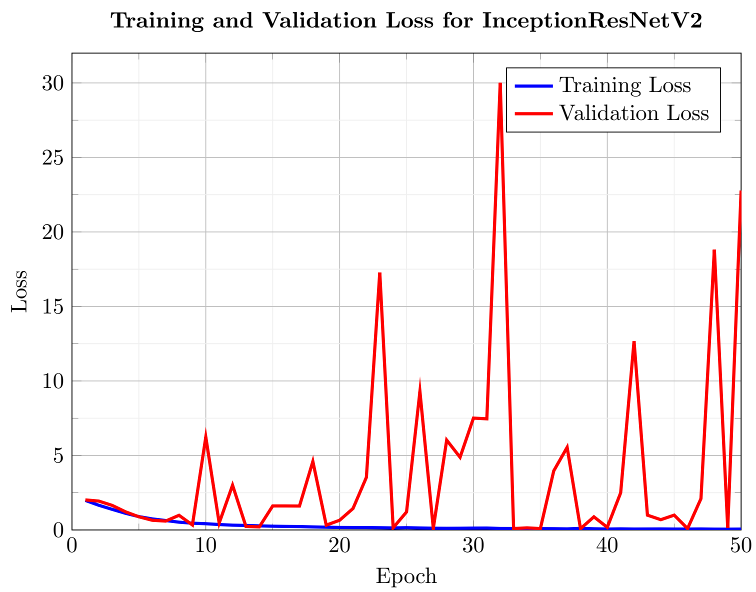


Fig. 13: InceptionResNetV2 Loss

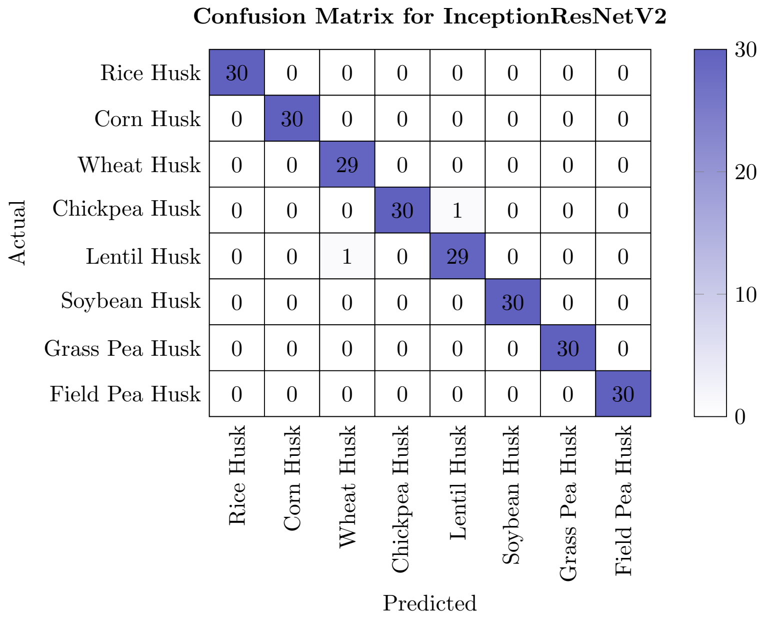


Fig. 14: InceptionResNetV2 Confusion Matrix

4.5.3 Results for ResNet152V2

Fig. 15 and Fig. 16 show the training and validation accuracy and loss respectively over 50 epochs. Both the accuracy and loss graphs are fairly smooth with loss decreasing and accuracy increasing. On completion of 50 epochs maximum training accuracy and minimum training loss came out to be 98.50% and 0.0502 respectively while maximum validation accuracy and minimum validation loss came out to be 99.17% and 0.0365 respectively. Fig17 depicts the confusion matrix for all the husk classes. Upon calculation on the test data, the accuracy came out to be 97.92%. The weighted average precision value for this model turned out to be 98.00%, the weighted average recall value for the same model was 97.92% and weighted average F1-Score of the same model was 97.91%.

5 Comparative Study

The papers reviewed in this study were trained on the original images from the "BDHusk" dataset. Similar studies can be conducted using other datasets and models, allowing for comparisons with additional research.
II Jahin et al.[24] proposed two models, ResNet50 and DenseNet201, both achieving a highest testing accuracy of 96.25%. In contrast, this study explores three models—Xception, InceptionResNetV2, and ResNet152V2—using the specified hyperparameters3. The proposed models demonstrated improved

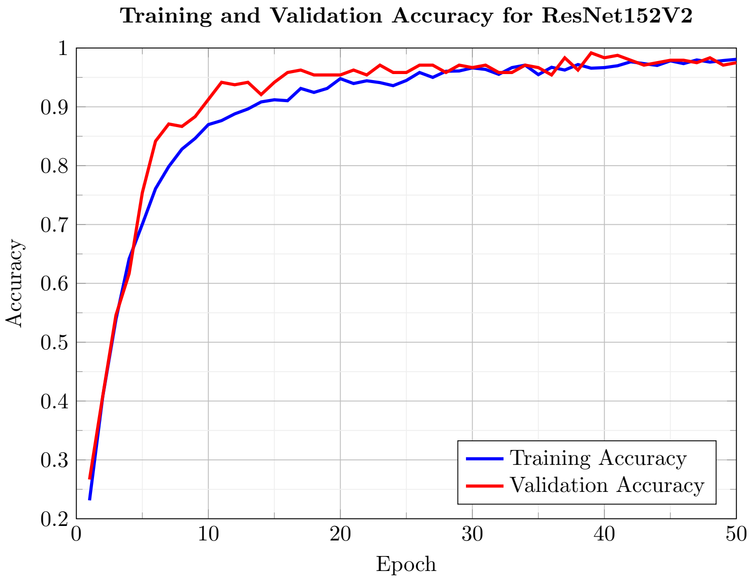


Fig. 15: ResNet152V2 Accuracy

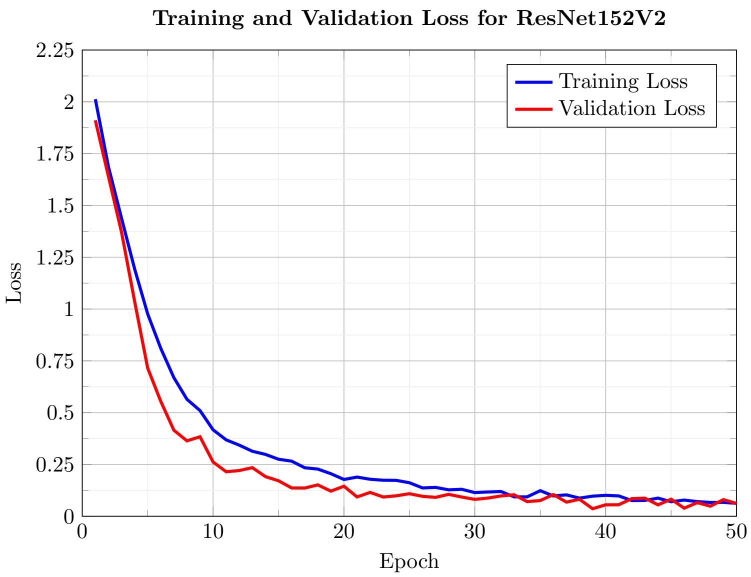


Fig. 16: ResNet152V2 Loss

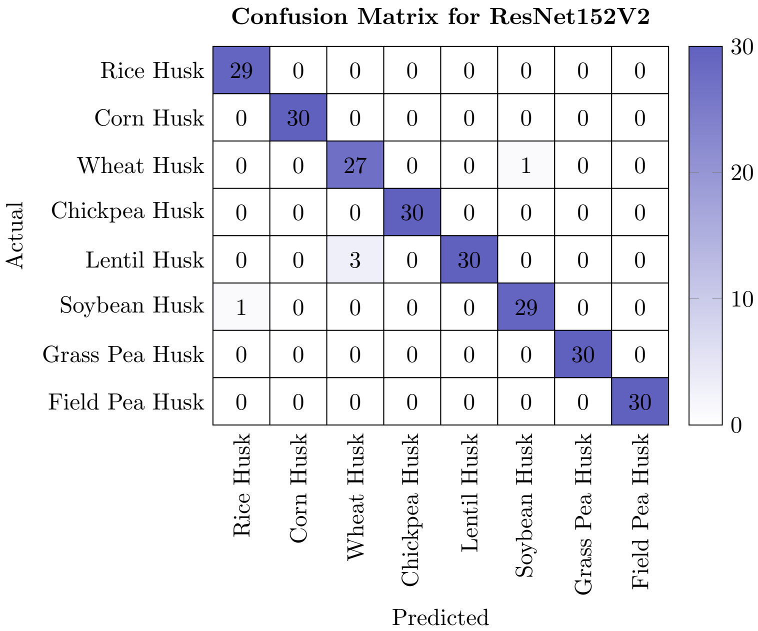


Fig. 17: ResNet 152V2 Confusion Matrix

performance on the testing dataset, achieving accuracies of 97.92%, 99.17%, and 97.92%, respectively. Notably, InceptionResNetV2 delivered the best accuracy, with the corresponding weights and biases recorded. The higher testing accuracies across all three models indicate superior generalization capabilities and performance on unseen data.

Table5 presents a detailed comparative analysis of our proposed models against previous work conducted on the same BDHusk dataset.

Sr. No.	Model Name	Testing Accuracy (%)
1	ResNet50[24]	96.25
2	DenseNet201[24]	96.25
3	Xception(proposed)	97.92
4	InceptionResNetV2(proposed)	99.17
5	ResNet152V2(proposed)	97.92

Table 5: Testing accuracy of different models

6 Conclusion and Future Scope

This study automates husk classification to enhance cattle and livestock feed formulation using convolutional neural networks (CNNs), demonstrating their efficiency in image recognition tasks. Utilizing the BDHusk dataset, which includes high-quality images of eight different types of husk, the study adopts three pre-trained models: Xception, InceptionResNetV2, and ResNet152V2. These models are tailored for husk classification by integrating custom-designed heads.

Through meticulous training and fine-tuning over 50 epochs using the BDHusk dataset, the models' performance was evaluated based on key metrics such as loss and accuracy. The culmination of the efforts in this study resulted in an impressive accuracy of 99.17% on the testing dataset using the InceptionResNetV2 model. This achievement underscores the strong predictive capabilities of the model in identifying and predicting husk species. By combining advanced computer vision and image processing techniques, supported by the BDHusk dataset and the refined InceptionResNetV2 architecture, this study has established a solid foundation for developing effective and reliable automated systems for husk classification to improve cattle feed formulation.

Furthermore, this study highlights the promising future of deep learning in addressing diverse challenges and transforming sectors like agriculture. By extracting advanced insights from visual data, deep learning can significantly enhance existing processes and improve outcomes. As algorithms and models continue to advance, we are moving towards faster and more efficient integration of deep learning into our daily lives.

Future research directions include broadening the BDHusk dataset to include more husk types and conditions, improving the model's generalization and accuracy. There is also potential to predict the nutritional content of husks, further enhancing the formulation of balanced and nutritious feed for livestock also considering cost as a parameter. Developing real-time husk recognition systems with IoT devices for integration into existing feed formulation processes could provide immediate benefits to farmers and feed manufacturers. Collaboration with agricultural institutions can further refine the model and ensure its practicality and effectiveness in real-world agricultural settings, ultimately benefiting the agriculture industry as a whole.

7 Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Funding No funding was received to assist with the preparation of this manuscript.

Data Availability The paper uses the publicly available dataset for husk classification. The dataset is openly available at https://data.mendeley.com/datasets/h754ntdtfx/1

References

1. Abdullahi, H.S., Sheriff, R., Mahieddine, F.: Convolution neural network in precision agriculture for plant image recognition and classification. In: 2017 Seventh International Conference on Innovative Computing Technology (INTECH), vol. 10, pp. 256–272. Ieee New York (2017)
2. Ahila Priyadharshini, R., Arivazhagan, S., Arun, M., Mirnalini, A.: Maize leaf disease classification using deep convolutional neural networks. Neural Computing and Applications **31**, 8887–8895 (2019)
3. Ahrendt, P., Gregersen, T., Karstoft, H.: Development of a real-time computer vision system for tracking loose-housed pigs. Computers and Electronics in Agriculture **76**(2), 169–174 (2011)
4. Atila, Ü., Uçar, M., Akyol, K., Uçar, E.: Plant leaf disease classification using efficientnet deep learning model. Ecological Informatics **61**, 101182 (2021)
5. Bezen, R., Edan, Y., Halachmi, I.: Computer vision system for measuring individual cow feed intake using rgb-d camera and deep learning algorithms. Computers and electronics in agriculture **172**, 105345 (2020)
6. Bjerrum, E.J.: Smiles enumeration as data augmentation for neural network modeling of molecules. arXiv preprint arXiv:1703.07076 (2017)
7. Carranza-García, M., García-Gutiérrez, J., Riquelme, J.C.: A framework for evaluating land use and land cover classification using convolutional neural networks. Remote Sensing **11**(3), 274 (2019)
8. Chen, D., Lu, Y., Li, Z., Young, S.: Performance evaluation of deep transfer learning on multi-class identification of common weed species in cotton production systems. Computers and Electronics in Agriculture **198**, 107091 (2022)
9. Chiu, M.T., Xu, X., Wei, Y., Huang, Z., Schwing, A.G., Brunner, R., Khachatrian, H., Karapetyan, H., Dozier, I., Rose, G., et al.: Agriculture-vision: A large aerial image database for agricultural pattern analysis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2828–2838 (2020)
10. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1251–1258 (2017)
11. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition, pp. 248–255. Ieee (2009)
12. Dorj, U.O., Lee, M., Yun, S.s.: An yield estimation in citrus orchards via fruit detection and counting using image processing. Computers and electronics in agriculture **140**, 103–112 (2017)
13. Eastridge, M.: Major advances in applied dairy cattle nutrition. Journal of dairy science **89**(4), 1311–1323 (2006)
14. Elngar, A.A., Arafa, M., Fathy, A., Moustafa, B., Mahmoud, O., Shaban, M., Fawzy, N.: Image classification based on cnn: a survey. Journal of Cybersecurity and Information Management **6**(1), 18–50 (2021)
15. Fernández-Carrión, E., Martínez-Avilés, M., Ivorra, B., Martínez-López, B., Ramos, Á.M., Sánchez-Vizcaíno, J.M.: Motion-based video monitoring for early detection of livestock diseases: The case of african swine fever. PloS one **12**(9), e0183793 (2017)
16. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics, pp. 249–256. JMLR Workshop and Conference Proceedings (2010)
17. Goutte, C., Gaussier, E.: A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In: European conference on information retrieval, pp. 345–359. Springer (2005)

18. Guo, T., Dong, J., Li, H., Gao, Y.: Simple convolutional neural network on image classification. In: 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA), pp. 721–724. IEEE (2017)
19. Haq, M.A.: Cnn based automated weed detection system using uav imagery. Computer Systems Science & Engineering **42**(2) (2022)
20. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778 (2016)
21. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
22. Hussain, M., Bird, J.J., Faria, D.R.: A study on cnn transfer learning for image classification. In: Advances in Computational Intelligence Systems: Contributions Presented at the 18th UK Workshop on Computational Intelligence, September 5-7, 2018, Nottingham, UK, pp. 191–202. Springer (2019)
23. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning, pp. 448–456. pmlr (2015)
24. Jahin, I.I., Khatun, M., Islam, M.T., Rahman, M.W., Raka, I.Z.: Bdhusk: A comprehensive dataset of different husk species images as a component of cattle feed from different regions of bangladesh. Data in Brief **52**, 110018 (2024)
25. Kamilaris, A., Prenafeta-Boldú, F.X.: Deep learning in agriculture: A survey. Computers and electronics in agriculture **147**, 70–90 (2018)
26. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25** (2012)
27. Kuhn, M., Johnson, K., et al.: Applied predictive modeling, vol. 26. Springer (2013)
28. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. Neural Computation **1**(4), 541–551 (1989). DOI 10.1162/neco.1989.1.4.541
29. Liu, Z., Wang, S.: Broken corn detection based on an adjusted yolo with focal loss. IEEE Access **7**, 68281–68289 (2019)
30. Lu, H., Fu, X., Liu, C., Li, L.g., He, Y.x., Li, N.w.: Cultivated land information extraction in uav imagery based on deep convolutional neural network and transfer learning. Journal of Mountain Science **14**, 731–741 (2017)
31. Ma, P., Li, A., Yu, N., Li, Y., Bahadur, R., Wang, Q., Ahuja, J.K.: Application of machine learning for estimating label nutrients using usda global branded food products database,(bfpd). Journal of Food Composition and analysis **100**, 103857 (2021)
32. Maharana, K., Mondal, S., Nemade, B.: A review: Data pre-processing and data augmentation techniques. Global Transitions Proceedings **3**(1), 91–99 (2022)
33. Milan, H.F., Perano, K.M., Gebremedhin, K.G.: Survey and future prospects in precision dairy farming. In: 10th International Livestock Environment Symposium (ILES X), p. 1. American Society of Agricultural and Biological Engineers (2018)
34. Milioto, A., Lottes, P., Stachniss, C.: Real-time blob-wise sugar beets vs weeds classification for monitoring fields using convolutional neural networks. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences **4**, 41–48 (2017)
35. Millam, J., Kosontyav, J., Mohammed, I., Abbaya, H., Bello, S., John, P.: Development of feeding regimes with some crop residues: maize offal, cowpea husk, poultry litter and groundnut haulms in a semi-arid environment of nigeria. Nigerian Journal of Animal Science **20**(3), 213–222 (2018)
36. Moses, K., Miglani, A., Kankar, P.K., et al.: Deep cnn-based damage classification of milled rice grains using a high-magnification image dataset. Computers and Electronics in Agriculture **195**, 106811 (2022)
37. Naseer, R., Hashmi, A.S., Rehman, H., Naveed, S., Masood, F., Tayyab, M., et al.: Assessment of feeding value of processed rice husk for lohi sheep in growing phase. Pakistan Journal of Zoology **49**(5) (2017)
38. Neethirajan, S., Kemp, B.: Digital livestock farming. Sensing and Bio-Sensing Research **32**, 100408 (2021)
39. Nevavuori, P., Narra, N., Lipping, T.: Crop yield prediction with deep convolutional neural networks. Computers and electronics in agriculture **163**, 104859 (2019)

40. Nikoloski, S., Murphy, P., Kocev, D., Džeroski, S., Wall, D.P.: Using machine learning to estimate herbage production and nutrient uptake on irish dairy farms. Journal of dairy science **102**(11), 10639–10656 (2019)
41. Nowakowski, A., Mrziglod, J., Spiller, D., Bonifacio, R., Ferrari, I., Mathieu, P.P., Garcia-Herranz, M., Kim, D.H.: Crop type mapping by using transfer learning. International Journal of Applied Earth Observation and Geoinformation **98**, 102313 (2021)
42. Oduguwa, O.O., Edema, M.O., Ayeni, A.O.: Physico-chemical and microbiological analyses of fermented corn cob, rice bran and cowpea husk for use in composite rabbit feed. Bioresource technology **99**(6), 1816–1820 (2008)
43. Osipov, A., Shumaev, V., Ekielski, A., Gataullin, T., Suvorov, S., Mishurov, S., Gataullin, S.: Identification and classification of mechanical damage during continuous harvesting of root crops using computer vision methods. IEEE Access **10**, 28885–28894 (2022)
44. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on knowledge and data engineering **22**(10), 1345–1359 (2009)
45. Van de Panne, C., Popp, W.: Minimum-cost cattle feed under probabilistic protein constraints. Management Science **9**(3), 405–430 (1963)
46. Piles, M., Fernandez-Lozano, C., Velasco-Galilea, M., González-Rodríguez, O., Sánchez, J.P., Torrallardona, D., Ballester, M., Quintanilla, R.: Machine learning applied to transcriptomic data to identify genes associated with feed efficiency in pigs. Genetics Selection Evolution **51**, 1–15 (2019)
47. Scott Jr, J.T., Broadbent, E.E.: A computerized cattle feeding program for replacement and ration formulation. Illinois Agricultural Economics pp. 16–25 (1972)
48. Shrivastava, V.K., Pradhan, M.K., Minz, S., Thakur, M.P.: Rice plant disease classification using transfer learning of deep convolution neural network. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences **42**, 631–635 (2019)
49. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
50. Sørensen, R.A., Rasmussen, J., Nielsen, J., Jørgensen, R.N.: Thistle detection using convolutional neural networks. In: EFITA WCCA 2017 Conference, Montpellier Supagro, Montpellier, France, pp. 2–6 (2017)
51. Sun, Y., Zhang, W., Gu, H., Liu, C., Hong, S., Xu, W., Yang, J., Gui, G.: Convolutional neural network based models for improving super-resolution imaging. Ieee Access **7**, 43042–43051 (2019)
52. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1–9 (2015)
53. Yalcin, H., Razavi, S.: Plant classification using convolutional neural networks. In: 2016 Fifth International Conference on Agro-Geoinformatics (Agro-Geoinformatics), pp. 1–5. IEEE (2016)