THE ASIAN BULLETIN OF BIG DATA MANAGMENT
Vol. 5. Issue 4 (2025)
https://doi.org/10.62019/0t1qfy54

ASIAN BULLETIN OF BIG DATA MANAGEMENT
http://abbdm.com/
ISSN (Print): 2959-0795
ISSN (online): 2959-0809

Seed Quality Detection Using Deep Learning Though RGB Camera

Tehreem Fatima, Maleeha Saqlain, Sonia Raoof, Farwa Raoof*, Nimra Razzaq, Zartasha Kiran

Chronicle

Article history
Received: Aug 2, 2025
Received in the revised format: Sept 5, 2025
Accepted: sept 29 2025
Available online: Oct 20, 2025

Tehreem Fatima, Farwa Raoof*, & Zartasha Kiran, are currently affiliated with the Department of Software Engineering, The University of Lahore.
Email: tehreem.fatima@se.uol.edu.pk
Email: farwa.raoof@se.uol.edu.pk
Email: zartasha.kiran@se.uol.edu.pk

Maleeha Saqlain & Nimra Razzaq are currently affiliated with the Department of Computer Science University of Agriculture Faisalabad, Pakistan.
Email: maleehasaqlain@gmail.com
Email: csnimra@gmail.com

Sonia Raoof is currently affiliated with the Department of Computer science and IT, The university of Lahore, Pakistan.
Email: sonia.raoof@cs.uol.edu.pk

Corresponding Author*

Abstract

Maize plays a vital role in Pakistan's agriculture, but its seeds are susceptible to damage and mold during transportation and storage, making it essential to strengthen the quality identification of maize seeds to increase crop yield and quality. Conventionally, seed recognition has remained inefficient and subjective despite the improvement in machine vision technology and digital image processing method employed in agricultural context. However, seed testing is still the most important aspect of seed technologies. The project was to identify pure and broken seeds with the help of deep learning, with an RGB camera which had high precision, low error, and accurate outputs at minimum space and cost. In order to solve this issue, various CNN models were used to identify mixed maize seeds as pure or broken seeds whereby the accuracy rate of CNN Sequential Model, ResNet50 and VGG16 was 86%, 80%, and 93%, respectively. The seed identification with the CNN models was much better than the conventional methods, as the complex features of the seed images are extracted and this minimizes the subjectivity of the conventional methods. The achievements of the CNN models in the project demonstrate the potential of deep learning in the classification of mixed maize seeds as pure and broken. This advancement in technology of seed identification may be a major boost in the efficiency and reliability of seed tests in the Pak agriculture production sector.

Keywords: Seed Quality Detection, Deep Learning, Convolutional Neural Networks, Computer Vision, Precision Agriculture, VGG16, Maize Seed Classification, Agricultural Automation.

INTRODUCTION

Agriculture is an important element of the Pakistan economy with high Gross Domestic Product (GDP), employment and food security [1]. Maize (Zea mays L.) is an important crop in this sector since it is used as an essential food as well as animal food and industrial products. Quality of the seeds is also of utmost importance since this directly affects crop establishment, uniformity, and end yield [2]. Nevertheless, the maize seeds are very vulnerable to mechanical damage and fungal contamination during post harvesting, storage and transportation. The germination rates of planting stocks, seedling vigor, and the general crop productivity may be seriously impacted by the penetration of broken or damaged seeds into the planting stock. Conventionally, seed quality evaluation has been dependent on manual inspection and sorting a process that is subjective in nature, labor-intensive and inefficient at that. The traditional practices are highly subject to human mistakes and discrepancy and may result in inadvertent sorting of poor seeds or loss of healthy ones [3]. As a result, it is urgently necessary to have an automated, objective and quick system that can enhance seed quality control measures. Continuous use of manual techniques in identifying seed quality in the agriculture of Pakistan especially maize has a lot of challenges. Such techniques are not efficient, accurate, and scalable to ascertain the current farming activities, which in turn would slow down the productivity and economic gains of the farmers. An automated solution is urgently needed to overcome the limitations of low throughput, high subjectivity, and operational inefficiency.

The primary objective of this research is to develop and evaluate a deep learning-based automated system for the accurate classification of maize seeds into "pure" and "broken" categories. The specific aims are:

- To implement and train three distinct Convolutional Neural Network (CNN) architectures—a custom Sequential Model, ResNet50, and VGG16 for binary image classification.
- To perform a comparative analysis of these models based on performance metrics including accuracy, precision, and recall.
- To demonstrate the feasibility of a cost-effective and space-efficient system utilizing a standard RGB camera for high-precision seed quality assessment.

PROPOSED METHODOLOGY

This study leverages a dataset comprising over 30,000 annotated images of pure and broken maize seeds. We employed a comparative framework, training and evaluating three prominent CNN architectures:

CNN Sequential Model: A custom-designed network with a linear stack of convolutional, pooling, and dense layers for hierarchical feature extraction

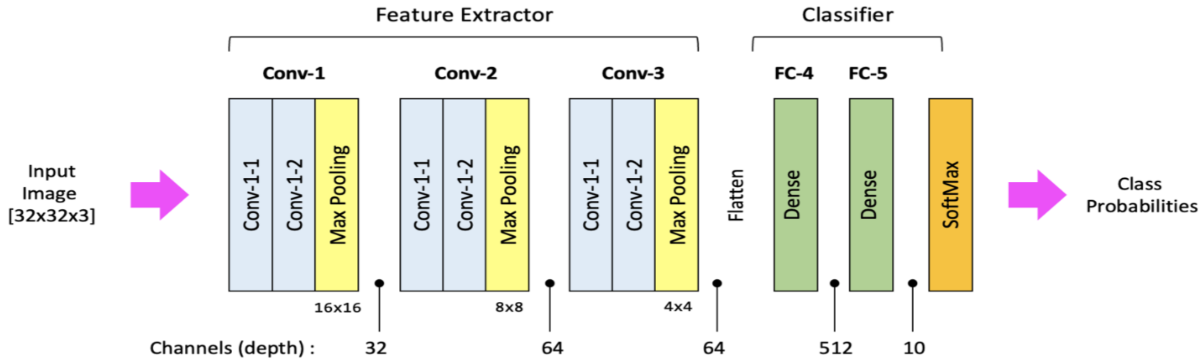

Figure 1.
Architecture of CNN sequential model

ResNet50: A deep 50-layer network utilizing residual learning blocks to facilitate the training of a very deep architecture and mitigate the vanishing gradient problem.

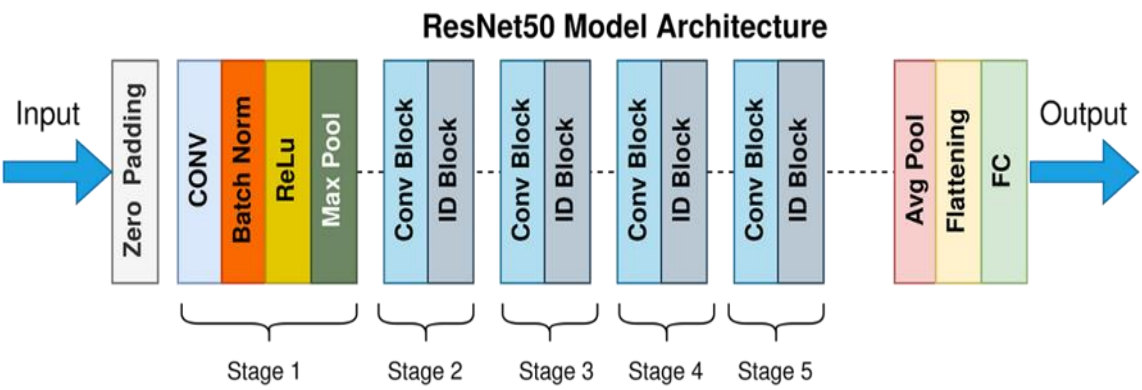

Figure2.
Architecture of ResNet50 model

VGG16: A well-established 16-layer network recognized for its uniform architecture using small receptive fields, providing a strong baseline for image classification tasks.

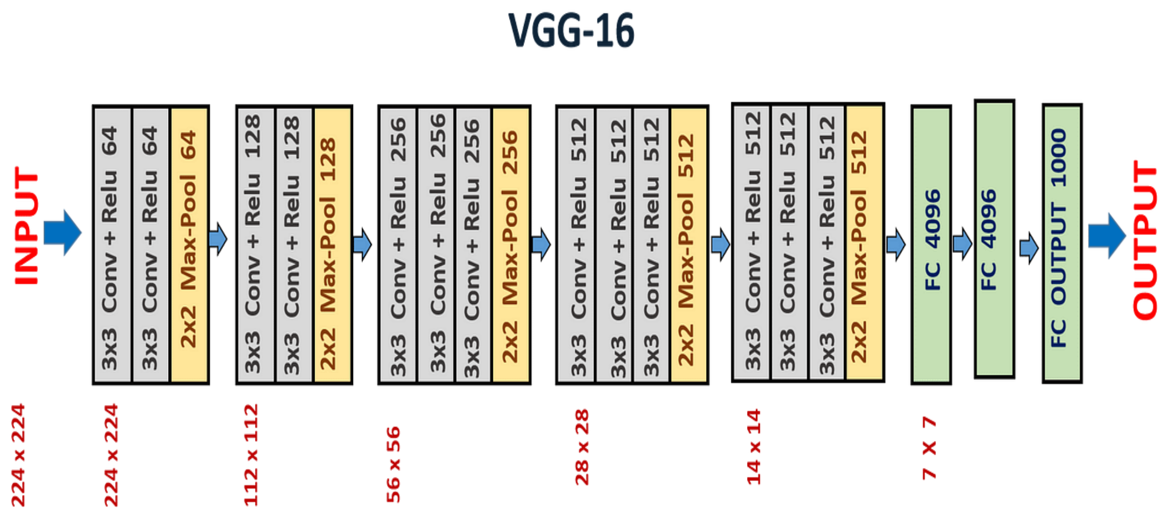

Figure 3.
Architecture of VGG16 model

Table 1.
Summary of Deep Learning Models Employed

Model	Depth	Key Characteristic
CNN Sequential	Custom	Linear stack of convolutional and pooling layers
ResNet50	50 layers	Residual connections with identity mapping
VGG16	16 layers	Uniform architecture with 3x3 convolutions

The models were trained end-to-end using the collected image dataset to learn discriminative features for robust classification [19]. This research contributes to the field of precision agriculture by demonstrating a practical application of deep learning for automated seed quality control [20]. The successful implementation of this system offers significant benefits, including Enhanced Agricultural Productivity: By ensuring the use of high-quality seeds, farmers can achieve better crop stands and higher yields. Economic Efficiency by Automation reduces reliance on manual labor, decreases operational costs, and minimizes losses from misclassified seeds. Technological Adoption research offers an assessed framework through which affordable, vision-based AI systems could be incorporated into the agricultural supply chain in developing nations such as Pakistan to assist in wider initiatives of food security and sustainable agriculture [21,22].

Existing Literature

Deep learning architectures, most notably the Convolutional Neural Networks (CNNs) have radically altered the evolution of computer vision as it has been shown that they can automatically learn to produce hierarchical feature representations to raw pixel inputs. The initial pioneering research by Bayar and Stam made CNNs flexible to specialized visual tasks by detecting manipulation artifacts [1] and further studies expanded such functionality to various tasks such as cultural heritage preservation and medical imaging [2,3]. The architectural development of VGG16 into the revolutionary ResNet50, changing the deep stacks of small convolutional filters to the revolutionary residual connections, not only allowed more advanced feature-learning but also effectively addressed the issue of vanishing gradient in deep networks [4,5].In object detection, the YOLO framework has been a revolutionary development that transformed detection to a single regression problem, which led to the development of increased capabilities in inference speed and competitive accuracy [6]. Further work improved the performance of YOLO by improving its training strategies and architecture and comparative analyzes showed its benefits in real-time use in applications in various areas, including self-driving cars and field inspection systems [7,8]. The multi-purpose nature of the framework is also demonstrated by its application in other agricultural settings, such as seed classification and quality assessment tasks [9]. The introduction of deep learning to agriculture has prompted the development of precision agriculture technologies, with the first ones being large-scale monitoring tasks implemented by the use of remote sensing [10,11]. The application area of seed quality assessment has become one of the most promising areas, where hand inspection has been gradually replaced by use of automated visual inspection systems. Early solutions that employed standard machine vision and hyperspectral imaging [12,13] have developed to involve deep learning-based solutions that are proving highly effective in varietal classification, internal defects, and overall assessment of quality [14,15,16].

Notably, soybean seed inspection has been used as the experimental ground in developing advanced defect recognition, where researchers have come up with robust frameworks to identify internal defects, external defects on the whole surface, and quality on the whole with CNN architecture and have extended their capability to detect dynamic variants of seeds through assessment Such solutions show potential of automated control of quality in the seed processing activities and offer useful methodologies that can be applied to other crops [18]. On the basis of these well-known deep learning networks and agricultural computer vision concepts, this study fulfills this crucial requirement of high-performance, precise and scale able seed quality measurements in maize production systems via comparative study of various CNN networks of pure and broken seed classification [17].

Dataset Composition and Experimental Setup

Experimental Framework and Dataset

The experimental framework was designed to systematically evaluate deep learning models for maize seed classification. A comprehensive dataset comprising over 30,000 RGB images of maize seeds was utilized, with 26,000 images representing pure seeds and 6,000 images representing broken seeds. The dataset was carefully curated to ensure diverse representations of seed conditions, orientations, and lighting variations to enhance model generalization capabilities [23].

Development Environment Configuration

The experimental setup employed Python programming language within the Anaconda distribution environment, specifically tailored for data science applications. The development interface, shown in Figure 3.1, provided an integrated platform for package management and environment configuration. Jupyter Notebook served as the primary development environment, enabling interactive code execution and visualization, as illustrated in Figure 3.2.

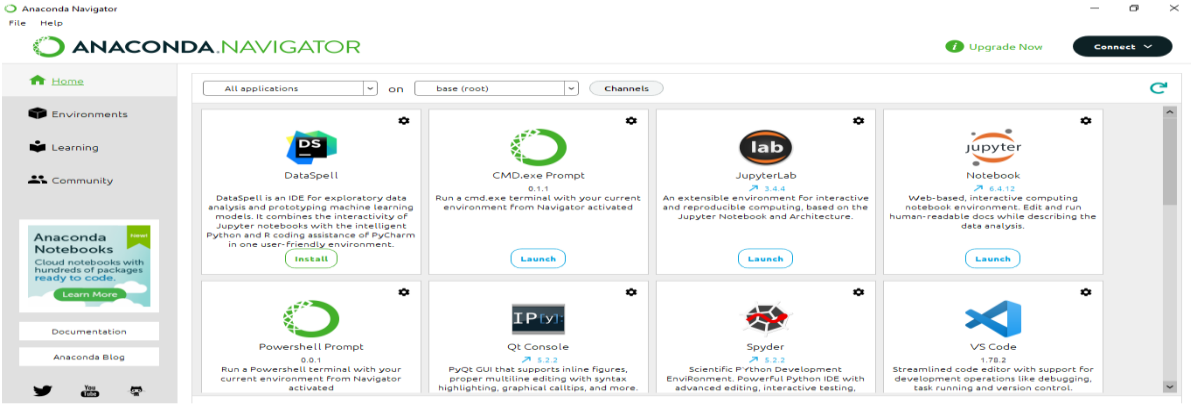

Figure 4.
Anaconda navigator interface

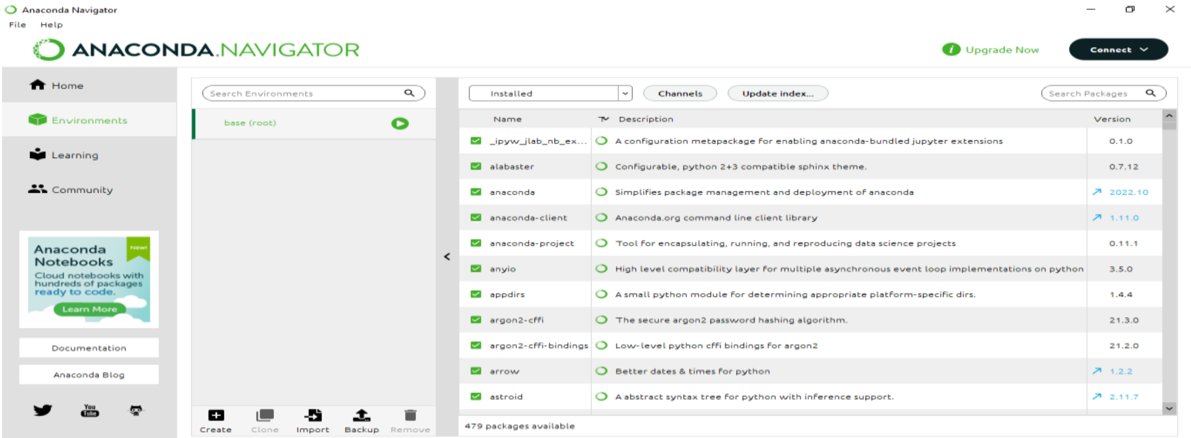

Figure 5.
Installed libraries in the anaconda navigator

COMPUTATIONAL LIBRARIES AND IMPLEMENTATION

Numerical Computing Infrastructure

NumPy provided the fundamental array operations and numerical computing capabilities essential for image data manipulation. The library was implemented using standard installation and import protocols (Figures 3.3,) with key arrays including

- data: Image data tensor containing resized seed image
- labels: Corresponding classification labels
- Training and validation splits (train_data, valid_data, train_labels, valid_labels)

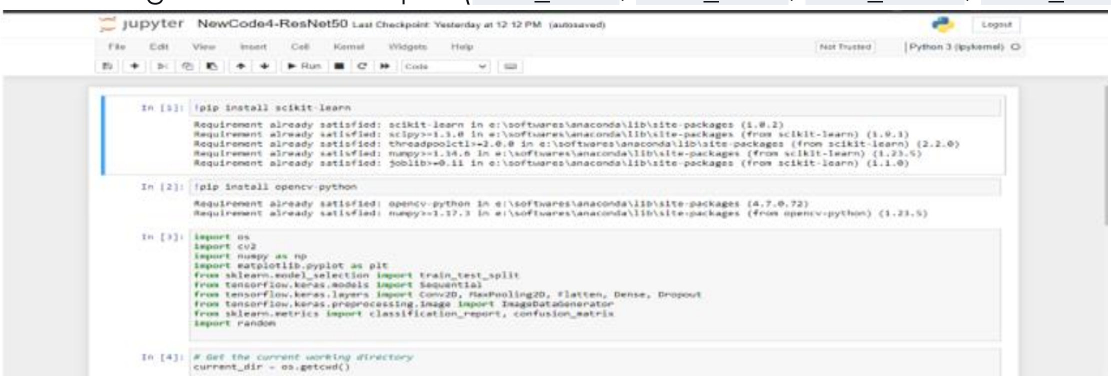

Figure 6.
Jupyter notebook interface

Deep Learning Framework

TensorFlow and Keras formed the core deep learning infrastructure, with the architectural relationship depicted in Figure 3.4 The installation and configuration followed established protocols (Figures 3.5, 3.6), utilizing the Sequential API for model construction (Figure 3.7). The layer hierarchy, illustrated in Figure 3.8, demonstrates the progressive feature extraction and transformation pipeline.

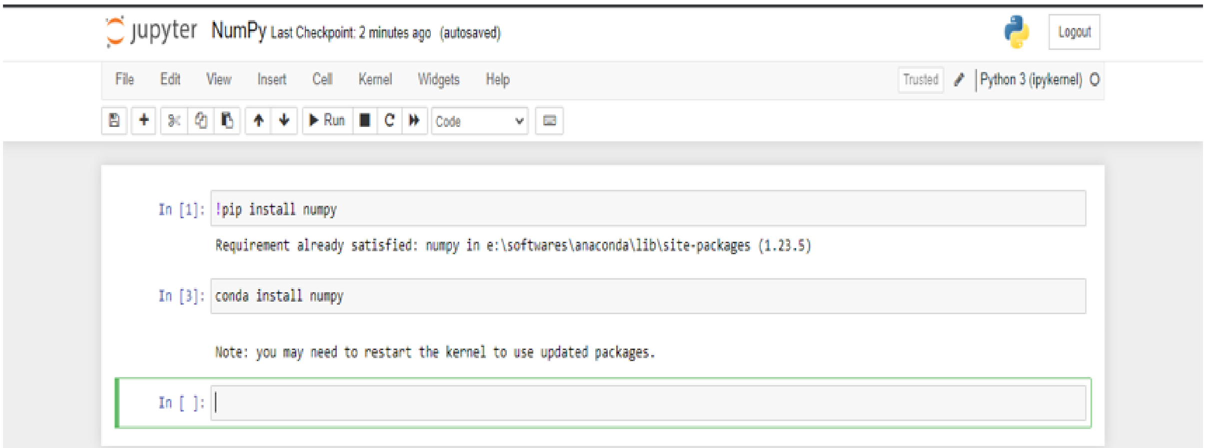

Figure 7.
NumPy installation

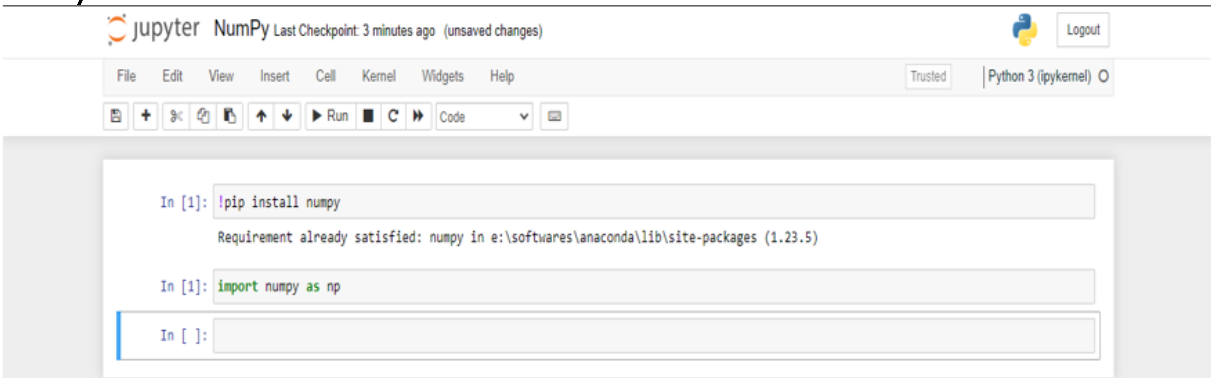

Figure 8.
NumPy importation

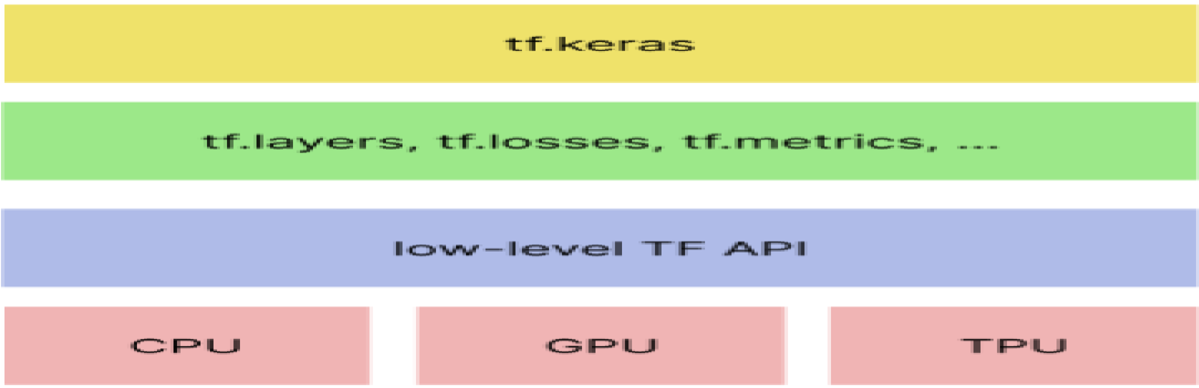

Figure 1.
Tensorflow & Keras architecture

```
!pip install tensorflow
```

Figure 9.

Installation of TensorFlow

```
!pip install keras
```

Figure 10.
Installation of Keras

Supporting Libraries

Scikit-learn provided essential machine learning utilities for dataset splitting, performance metrics, and confusion matrix generation (Figures 3.9, 3.10). Matplotlib enabled comprehensive visualization of training progress, results, and analytical metrics (Figures 3.11, 3.12), with detailed plot anatomy shown in Figure 3.13.

```python
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense

# Create a Sequential model
model = Sequential()

# Add Layers to the model
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(img_width, img_height, 3)))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dense(1, activation='sigmoid'))

# Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Train the model
history = model.fit(
    train_generator,
    steps_per_epoch=len(train_data) // batch_size,
    epochs=10,
    validation_data=valid_generator,
    validation_steps=len(valid_data) // batch_size
)
```

Figure 11.
Code snippet of keras sequential API

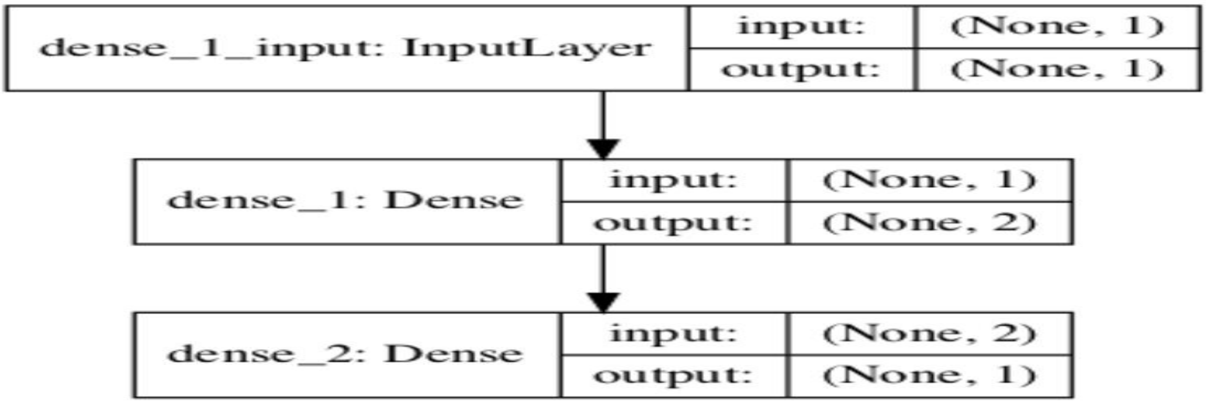

Figure 12.
Architecture of sequential model layers

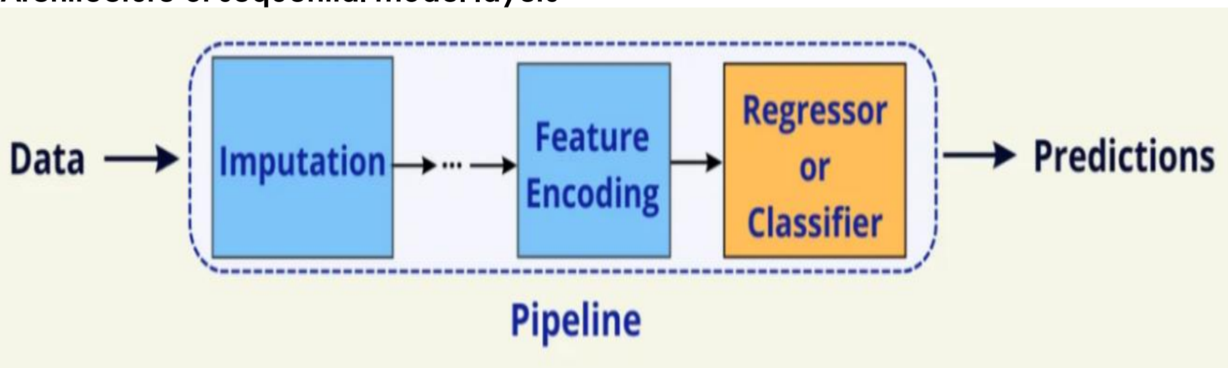

Figure 13.
Architecture pipeline of scikit-learn

```
pip install scikit-learn
```

Figure 14.
Installation of scikit-learn

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix

# Split the data into training and validation sets
train_data, valid_data, train_labels, valid_labels = train_test_split(data, labels, test_size=0.2, stratify=labels, random_state=

# Randomly select images for visualization
random_indices = random.sample(range(len(valid_data)), 10)
random_images = valid_data[random_indices]
random_labels = valid_labels[random_indices]

# Make predictions on the random images
random_images = random_images / 255.0  # Normalize the images
predictions = model.predict(random_images)
predicted_labels = np.round(predictions)

# Calculate classification metrics
true_labels = random_labels
class_names = ['broken', 'pure']

print('Classification Report:')
print(classification_report(true_labels, predicted_labels, target_names=class_names))

# Calculate the confusion matrix
cm = confusion_matrix(true_labels, predicted_labels)
print('Confusion Matrix:')
print(cm)
```

Figure 15.
Code snippet of scikit-learn

Deep Learning Architectures

Convolutional Neural Network Fundamentals

The study employed CNN architectures based on their proven efficacy in image classification tasks. The fundamental CNN architecture, illustrated in Figure 3.14, processes input images through successive convolutional and pooling layers to extract hierarchical features, followed by fully connected layers for classification. The complete experimental workflow is summarized in Figure 3.15.

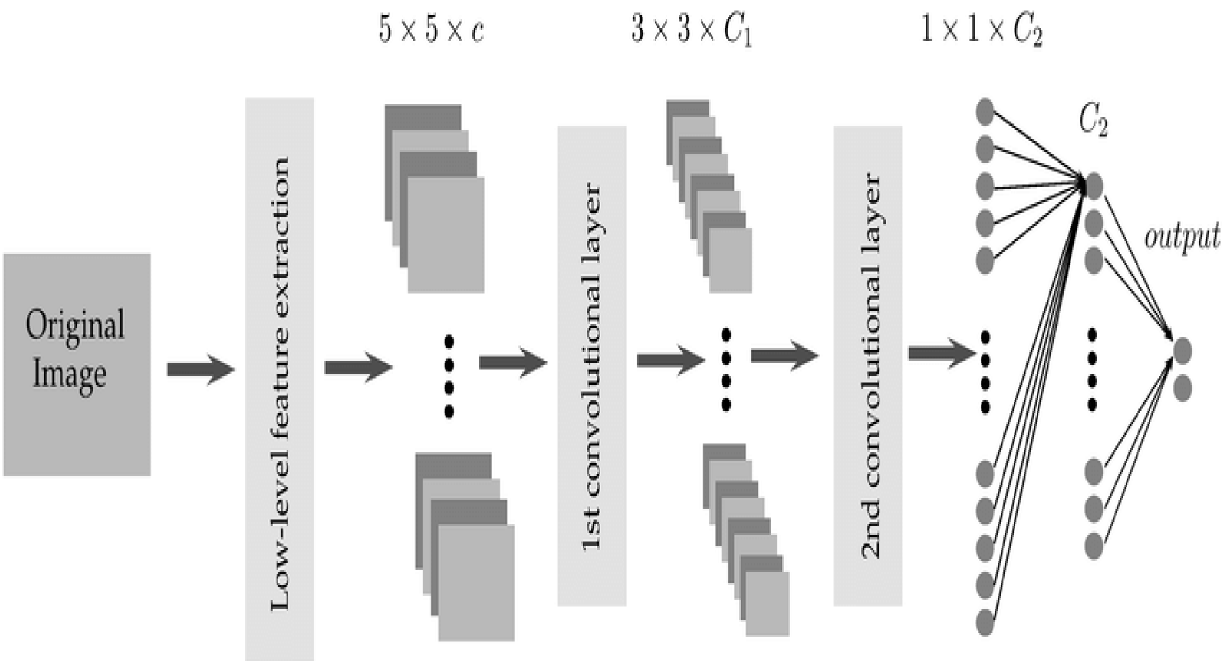

Figure 16.
Architecture of deep learning

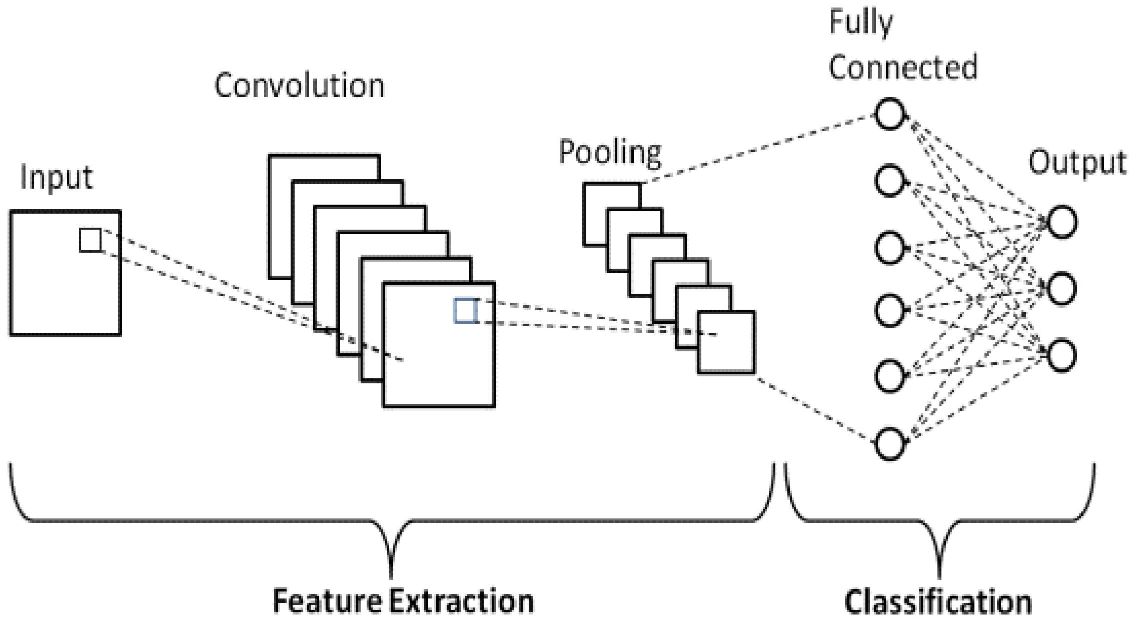

Figure 17.
Architecture of CNN model

Custom Sequential Model

```
Model: "sequential"
_________________________________________________________________
Layer (type)              Output Shape             Param #
=================================================================
conv2d (Conv2D)            (None, 198, 198, 32)      896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 99, 99, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)          (None, 97, 97, 64)        18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2D) (None, 48, 48, 64)      0
_________________________________________________________________
conv2d_2 (Conv2D)          (None, 46, 46, 64)        36928
_________________________________________________________________
flatten (Flatten)          (None, 135424)            0
_________________________________________________________________
dense (Dense)              (None, 64)                8667200
_________________________________________________________________
dense_1 (Dense)            (None, 2)                 130
=================================================================
Total params: 8,723,650
Trainable params: 8,723,650
Non-trainable params: 0
```

ResNet50 Implementation
The ResNet50 architecture was implemented using transfer learning principles. The model loading and configuration followed the structure shown in Figure 3.16. Data augmentation techniques (Figure 3.17) were employed to enhance model robustness, including:

- Random rotations (±20 degrees)
- Horizontal and vertical flipping
- Zoom and shear transformations
- Brightness and contrast adjustments

The base ResNet50 model was supplemented with custom classification layers:
base_model = ResNet50(weights='imagenet', include_top=False, input_shape=(224, 224, 3))
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(1024, activation='relu')(x)
predictions = Dense(2, activation='softmax')(x)
model = Model(inputs=base_model.input, outputs=predictions)

```
# Load the ResNet50 model
base_model = ResNet50(weights='imagenet', include_top=False, input_shape=(img_width, img_height, 3))
```

```
# Create the model
model = Sequential()
model.add(base_model)
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid'))
```

Figure 18.
ResNet50 model loading

```
# Data augmentation and normalization for training and validation
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=0.1,
    zoom_range=0.1,
    horizontal_flip=True,
    vertical_flip=True
)

valid_datagen = ImageDataGenerator(rescale=1./255)
```

Figure 19.
Data augmentation

VGG16 Architecture

The VGG16 model was implemented with its characteristic uniform architecture of 3×3 convolutional filters (Figure 3.18). The model shown in Figure 3.19, maintaining the original architectural principles while adapting the final layers for binary classification.

```
!pip install tensorflow
!pip install numpy
!pip install matplotlib
!pip install scikit-learn
!pip install opencv-python

from tensorflow.keras.models import Sequential
from tensorflow.keras.applications import VGG16
from tensorflow.keras.layers import Flatten, Dense, Dropout
from tensorflow.keras.optimizers import Adam
from sklearn.model_selection import train_test_split
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np
import matplotlib.pyplot as plt
import os
import cv2
```

Figure 20.
Libraries used in VGG16 model

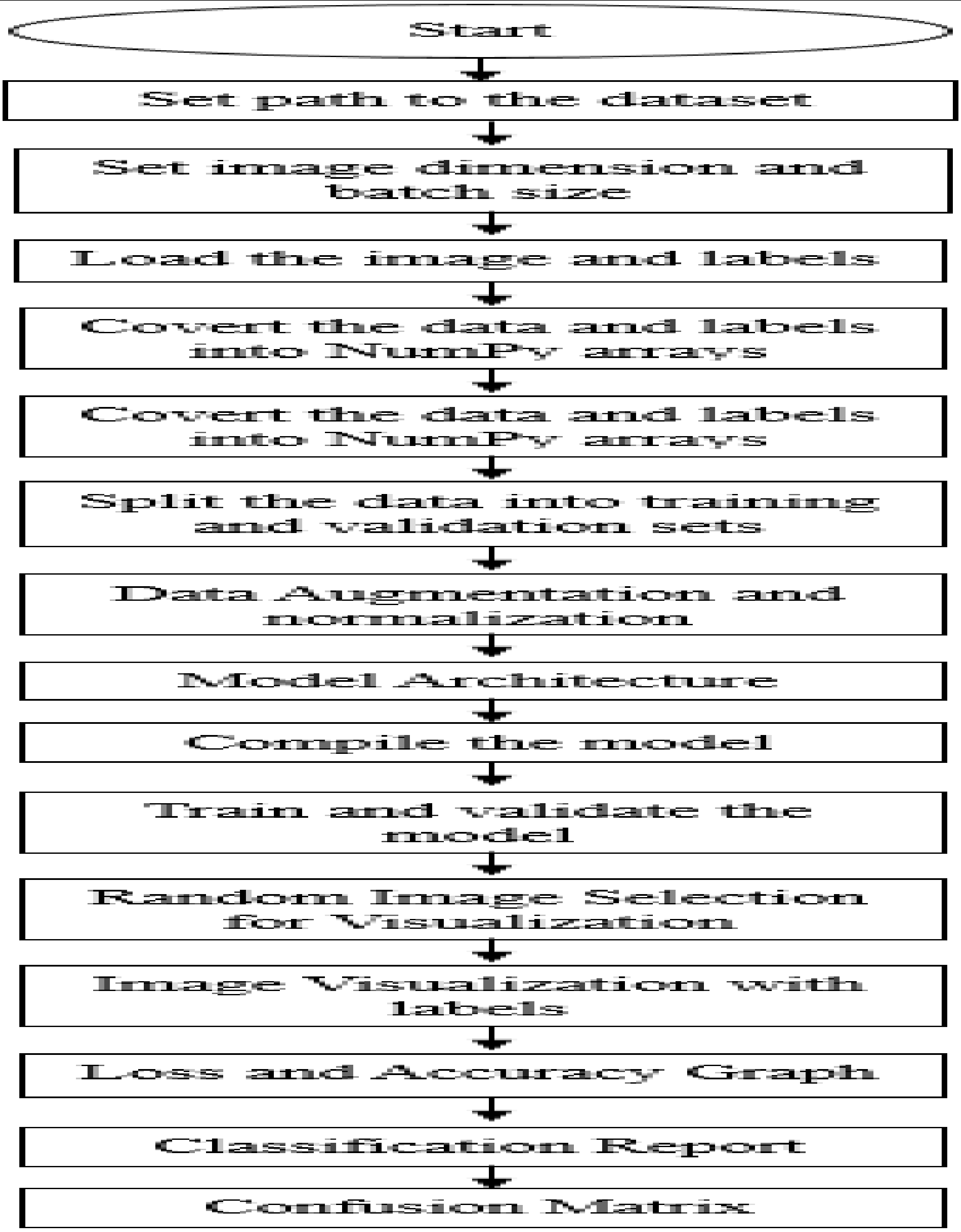

Figure 21.
Flowchart of the model

Training Configuration

All models were trained with consistent hyperparameters to ensure fair comparison:

- Epochs: 10
- Batch size: 3
- Optimization algorithm: Adam
- Learning rate: 0.00
- Loss function: Categorical Cross entropy

- Train-validation split: 80-20%

The training process monitored key metrics including accuracy, precision, recall, and F1-score, with early stopping implemented to prevent overfitting. The comprehensive experimental design ensured rigorous evaluation of each architecture's capability for maize seed quality classification.

RESULTS AND DISCUSSION

This study evaluated the efficacy of three convolutional neural network (CNN) architectures for the binary classification of maize seeds into "pure" and "broken" categories. The models were trained and validated on a dataset of over 30,000 RGB images. The performance of the custom CNN Sequential Model, ResNet50, and VGG16 is summarized below, with a comparative analysis provided in Table 2.

Table 2.

Model	Epochs	Batch Size	Accuracy	Key Observation
CNN Sequential	10	32	86%	Robust performance with a simpler architecture.
ResNet50	10	32	80%	Struggled with convergence, leading to higher false negatives.
VGG16	10	32	**93%**	Superior feature extraction and generalization, achieving the best performance.

Custom CNN Sequential Model

The custom Sequential model demonstrated a strong learning capability, achieving a final accuracy of 86%. The training and validation curves (Fig. 20) show a steady convergence indicating effective learning without significant overfitting [1]. The classification report (not shown) revealed balanced precision and recall for both classes. The confusion matrix (Fig. 21) confirms a reliable performance with a low and balanced rate of misclassification between the two categories [24].

```
Epoch 1/10
482/482 [==============================] - 252s 521ms/step - loss: 0.5772 - accuracy: 0.7015 - val_loss: 0.4753 - val_accuracy:
0.7839
Epoch 2/10
482/482 [==============================] - 275s 570ms/step - loss: 0.4652 - accuracy: 0.7894 - val_loss: 0.4048 - val_accuracy:
0.8307
Epoch 3/10
482/482 [==============================] - 266s 552ms/step - loss: 0.3689 - accuracy: 0.8466 - val_loss: 0.2992 - val_accuracy:
0.8792
Epoch 4/10
482/482 [==============================] - 275s 571ms/step - loss: 0.3483 - accuracy: 0.8588 - val_loss: 0.3516 - val_accuracy:
0.8526
Epoch 5/10
482/482 [==============================] - 274s 568ms/step - loss: 0.3332 - accuracy: 0.8634 - val_loss: 0.3136 - val_accuracy:
0.8727
Epoch 6/10
482/482 [==============================] - 277s 575ms/step - loss: 0.3190 - accuracy: 0.8713 - val_loss: 0.2637 - val_accuracy:
0.9005
Epoch 7/10
482/482 [==============================] - 3913s 8s/step - loss: 0.3037 - accuracy: 0.8794 - val_loss: 0.2525 - val_accuracy:
0.8984
Epoch 8/10
482/482 [==============================] - 275s 571ms/step - loss: 0.3024 - accuracy: 0.8785 - val_loss: 0.2602 - val_accuracy:
0.8865
Epoch 9/10
482/482 [==============================] - 282s 584ms/step - loss: 0.3162 - accuracy: 0.8726 - val_loss: 0.2413 - val_accuracy:
0.9049
Epoch 10/10
482/482 [==============================] - 314s 650ms/step - loss: 0.2850 - accuracy: 0.8865 - val_loss: 0.3240 - val_accuracy:
0.8685
```

Figure 22.
Training epochs of CNN sequential model

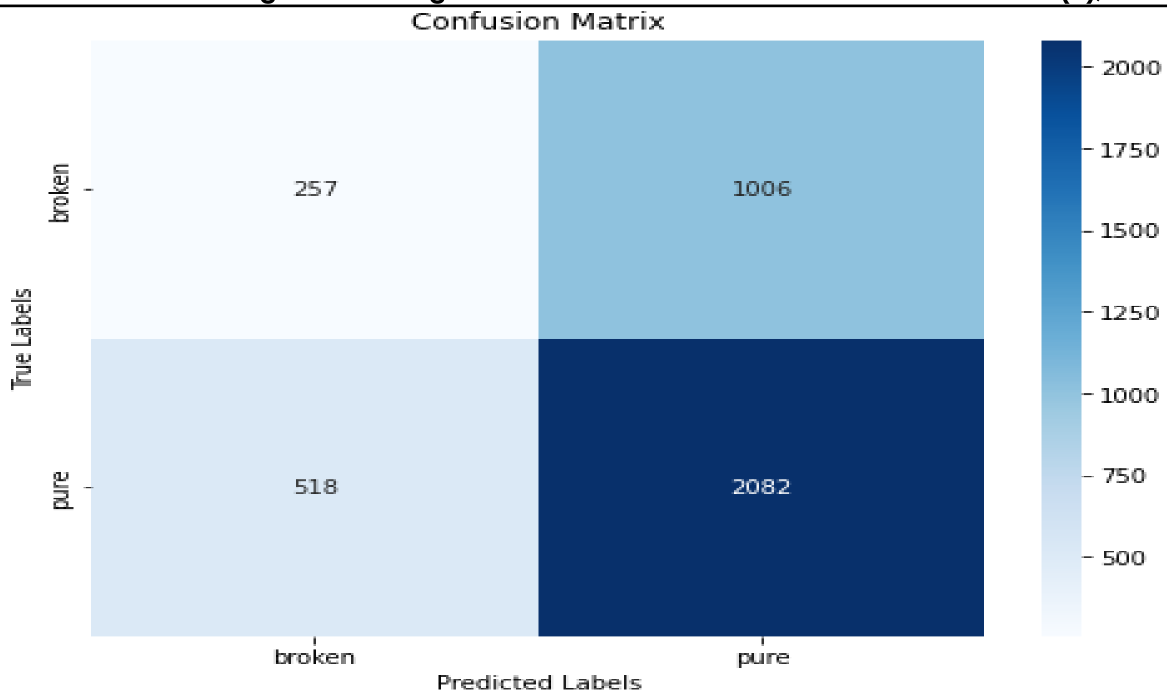

Figure 23.
Confusion matrix of CNN sequential model

ResNet50 Model

The ResNet50 model yielded the lowest accuracy among the three models at 80%. Its training process was less stable, as evidenced by the fluctuating validation loss (Fig. 23) suggesting potential challenges in optimizing the deep residual network with the given dataset size [2]. Critically, the confusion matrix (Fig. 24) shows a higher count of false negatives, meaning a significant number of broken seeds were misclassified as pure. This is a major drawback for a quality control system where detecting defects is paramount.

```
Classification Report:
              precision    recall  f1-score   support

      broken       0.00      0.00      0.00         0
        pure       1.00      1.00      1.00        10

   micro avg       1.00      1.00      1.00        10
   macro avg       0.50      0.50      0.50        10
weighted avg       1.00      1.00      1.00        10
```

Figure 24.
Classification report of ResNet50 model

The VGG16 model significantly outperformed the others, achieving a peak accuracy of 93%. The training progress was exemplary, with smooth and closely aligned training and validation curves for both accuracy and loss (Fig.25), indicating excellent generalization [3]. The model's predictions on a sample of test images (Fig. 3c) were consistently correct.

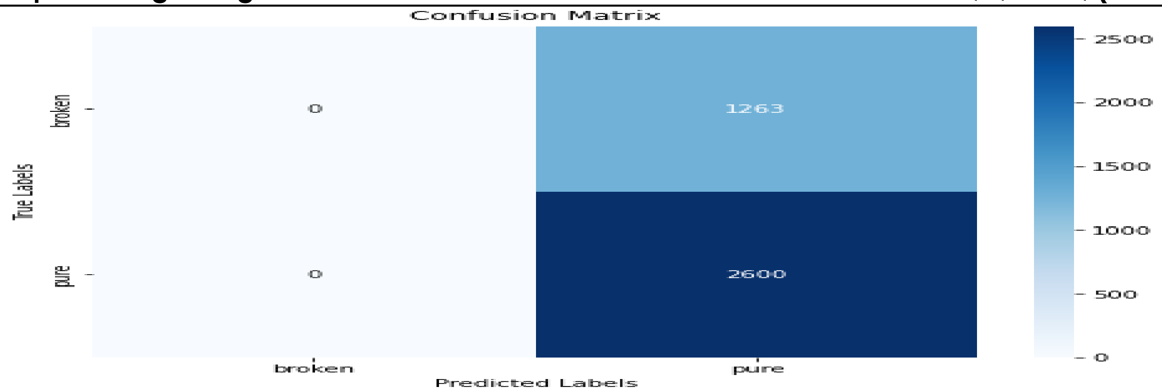

Figure 25.
Confusion matrix of CNN sequential model

VGG16 Model

The corresponding confusion matrix (Fig. 26) demonstrates a high number of true positives and true negatives, with minimal errors, underscoring its reliability for this specific task [25].

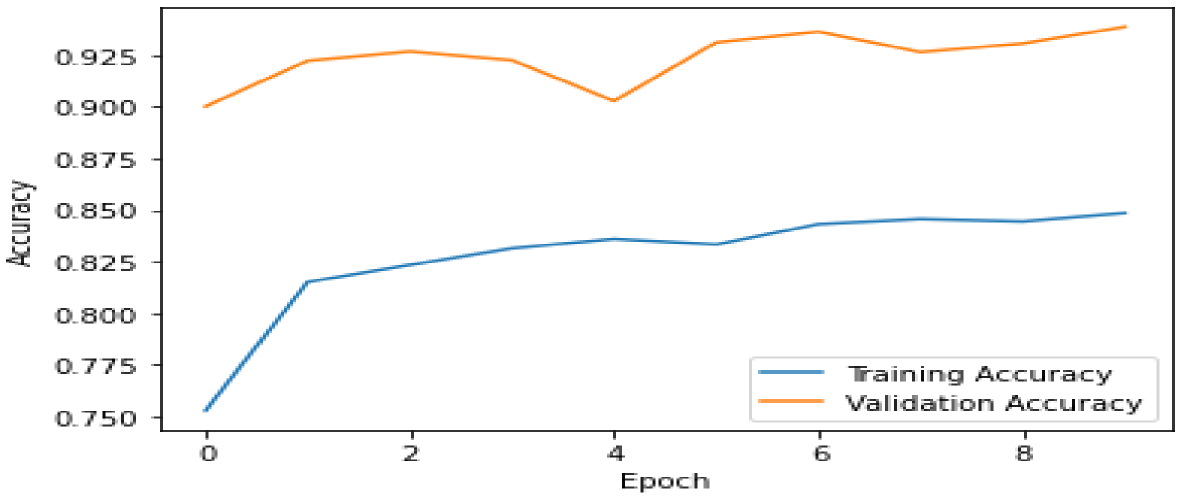

Figure 26.
Training and validation accuracy graph of VGG16 model

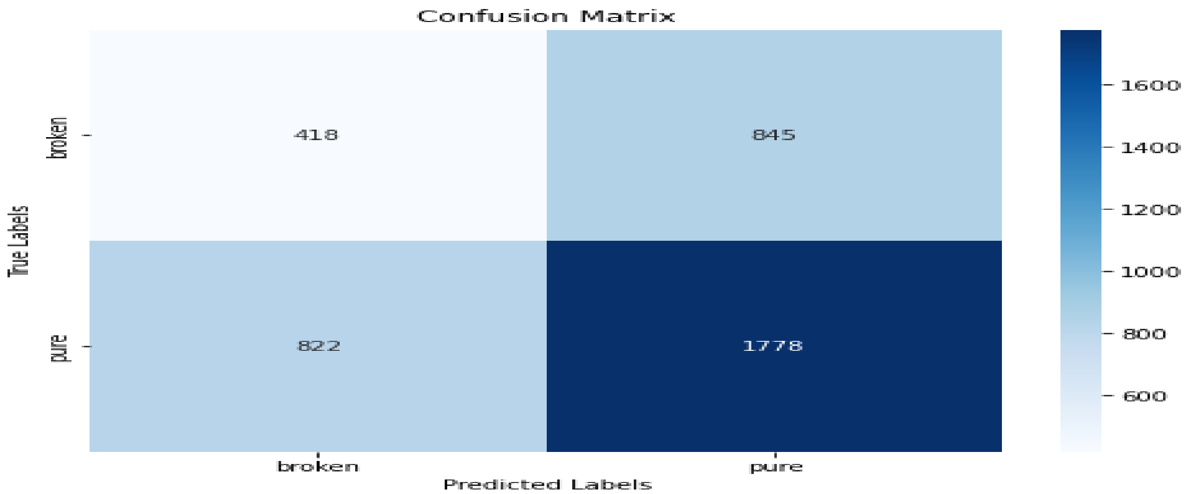

Figure 27.
Confusion matrix of VGG16 model

DISCUSSION

The results conclusively demonstrate the viability of deep learning for automating maize seed quality assessment. The better performance of VGG16 model (93% accuracy) is credited to the effective architecture of the model, which itself is supported by transfer learning [3]. Conversely, the lower performance (80% accuracy) of the deeper ResNet50 model might have been due to undertraining or possibly just a lack of enough and diverse data to make full use of the complex architecture in the model [2]. The CNN model with its customized version was a powerful baseline (86% accuracy), which demonstrated the fact that even simplistic architectures can provide a substantial advantage over manual ones. The article concurs with the results of Zhao et al. [4], who were able to use CNNs to detect seed defects in soybean, and with the results of the present article, which prove that such methods can be transferred to maize [29,30]. Its great precision and strong performance indicate that VGG16 could be used in a real-time and automated system of seed sorting to drive up efficiency, lower labor expenses, and enhance the agricultural supply chain of seeds in general [26].

CONCLUSION

This Study has managed to show that deep learning can be used in automated and precise classification of maize seeds, which is a crucial activity in assuring crop quantity and quality in the Pakistani agricultural industry. The three CNN architectures that were tested were a custom Sequential Model, ResNet50, and VGG16, with VGG16 prevailing as the best model to be used by a considerably high percentage of 93 outperforming the Sequential Model (86%) and ResNet50 (80%). This good performance highlights the efficiency of transfer learning in this particular agricultural computer vision problem. The usage of such deep-learning-based system has substantial practical implications. It allows a quick, high degree of precision with the quality of the seeds and this translates directly to better quality control, farmers are able to save money and minimize the chances of crop failure as the seeds that are planted are of high quality [28]. Moreover, it can be mentioned that the possibility of the early diagnosis of diseases and the possibility to offer a stable, efficient product may enhance the competitiveness of a farmer in the market. Overall, this study confirms that deep learning, specifically, the VGG16 architecture, is a strong, effective, and affordable option of the automation of maize seed quality identification. The suggested system has a significant potential of being incorporated in the real-life agricultural processes, leading to the growth of productivity and sustainability of seed industry [27]. The next path of work may be related to the implementation of this model into a physical sorting system and extending its features to be able to sort a broader scope of seed defects and diseases.

DECLARATIONS

Acknowledgement: We appreciate the generous support from all the contributor to the research and their different affiliations.
Funding: No funding body in the public, private, or nonprofit sectors provided a particular grant for this research.
Availability of data and material: In the approach, the data sources for the variables are stated.
Authors' contributions: Each author participated equally in the creation of this work.
Conflicts of Interest: The authors declare no conflict of interest.
Consent to Participate: Yes

Consent for publication and Ethical approval: Because this study does not include human or animal data, ethical approval is not required for publication. All authors have given their consent.

REFERENCES

Ahmad, I., Y. Yang, Y. Yue, C. Ye, M. Hassan, X. Cheng, Y. Wu and Y. Zhang. 2022. Deep Learning Based Detector YOLOv5 for Identifying Insect Pests. *Appl. Sci.* 12:10167.

Ahmad, T., Y. Ma, M. Yahya, B. Ahmad, S. Nazir and A.U. Haq. 2020. Object Detection through Modified YOLO Neural Network. *Sci. Program.* 2020:1–10.

Algorry, A.M., A.G. Garcia and A.G. Wofmann. 2017. Real-Time Object Detection and Classification of Small and Similar Figures in Image Processing. *2017 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE. pp.516–519.

Bansal, M., M. Kumar and M. Kumar. 2021. 2D Object Recognition Techniques: State-of-the-Art Work. *Arch. Comput. Methods Eng.* 28:1147–1161.

Bermeitinger, B., A. Freitas, S. Donig and S. Handschuh. 2016. Object Classification in Images of Neoclassical Furniture Using Deep Learning. *IFIP Advances in Information and Communication Technology*. pp.109–112.

Bhurtel and Manish. 2019. Deep Learning based Seed Quality Tester. *Science, Engineering, and Technology (SET)*. Kathmandu, Nepal. pp.8–9.

Bui, H.M., M. Lech, E. Cheng, K. Neville and I.S. Burnett. 2016. Object Recognition Using Deep Convolutional Features Transformed by a Recursive Network Structure. *IEEE Access* 4:10059–10066.

Çetin, N. 2022. Machine Learning for Varietal Binary Classification of Soybean (Glycine max (L.) Merrill) Seeds Based on Shape and Size Attributes. *Food Anal. Methods* 15:2260–2273.

Chen, Z., W. Fan, Z. Luo and B. Guo. 2022. Soybean seed counting and broken seed recognition based on image sequence of falling seeds. *Comput. Electron. Agric.* 196:106870.

Diwan, T., G. Anirudh and J. V. Tembhurne. 2023. Object detection using YOLO: challenges, architectural successors, datasets and applications. *Multimed. Tools Appl.* 82:9243–9275.

Feng, Z., X. Luo, T. Yang and K. Kita. 2018. An Object Detection System Based on YOLO in Traffic Scene. *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*. IEEE. pp.1532–1536.

Fujiyoshi, H., T. Hirakawa and T. Yamashita. 2019. Deep learning-based image recognition for autonomous driving. *IATSS Res.* 43:244–252.

Garg, D., P. Goel, S. Pandya, A. Ganatra and K. Kotecha. 2018. A Deep Learning Approach for Face Detection using YOLO. *2018 IEEE Punecon*. IEEE. pp.1–4.

Habib, S. and N. Fayyaz Khan. 2021. An Optimized Approach to Vehicle-Type Classification Using a Convolutional Neural Network. *Comput. Mater. Contin.* 69:3321–3335.

Hasan Purhaghshenas, S. and M. Esmatnia. 2012. IMPORTANCE OF AGRICULTURAL SECTOR IN PAKISTAN. *Interdiscip. J. Contemp. Res. Bus.* 3:421–427.

Hazer, A. and R. Yildirim. 2022. Deep Learning Based Point Cloud Processing Techniques. *IEEE Access* 10:127237–127283.

Hemström, F., F. Nässtrom and J. Karlholm. 2019. Detecting and classifying small objects in thermal imagery using deep learning. In: J. Dijk (ed.), *Artificial Intelligence and Machine Learning in Defense Applications*. SPIE. pp.6.

Huang, S., X. Fan, L. Sun, Y. Shen and X. Suo. 2019. Research on Classification Method of Maize Seed Defect Based on Machine Vision. *J. Sensors* 2019:1–9.

Huang, Z., R. Wang, Y. Cao, S. Zheng, Y. Teng, F. Wang, L. Wang and J. Du. 2022a. Deep learning based soybean seed classification. *Comput. Electron. Agric.* 202:107393.

Huang, Z., R. Wang, L. Wang, Y. Teng and S. Zheng. 2022b. Identification of internal cracks in corn seed using convolutional neural networks. *MATEC Web Conf.* 355:03027.

Ilieva, V., S. Mitrev, I. Karov, N. Markova and E. Todorovska. 2013. Seed Quality and Its Importance in Agricultural Production and Safety of. *Int. Conf. "Quality Competence"* 1–11.

Jana, A.P., A. Biswas and Mohana. 2018. YOLO based Detection and Classification of Objects in video records. *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*. IEEE. pp.2448–2452.

Jeong, H.-J., K.-S. Park and Y.-G. Ha. 2018. Image Preprocessing for Efficient Training of YOLO Deep Learning Networks. *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*. IEEE. pp.635–637.

Jia, B., K.D. Pham, E. Blasch, Z. Wang, D. Shen and G. Chen. 2018. Space object classification using deep neural networks. *2018 IEEE Aerospace Conference*. IEEE. pp.1–8.

Kim, J.-H., N. Kim, Y.W. Park and C.S. Won. 2022. Object Detection and Classification Based on YOLO-V5 with Improved Maritime Dataset. *J. Mar. Sci. Eng.* 10:377.

Kim, J., J. Kim and J. Cho. 2019. An advanced object classification strategy using YOLO through camera and LiDAR sensor fusion. *2019, 13th International Conference on Signal Processing and Communication Systems, ICSPCS 2019 - Proceedings*. IEEE. pp.1–5.

Kundu, N., G. Rani and V.S. Dhaka. 2021. Seeds Classification and Quality Testing Using Deep Learning and YOLO v5. *Proceedings of the International Conference on Data Science, Machine Learning and Artificial Intelligence*. ACM, New York, NY, USA. pp.153–160.

Kussul, N., M. Lavreniuk, S. Skakun and A. Shelestov. 2017. Deep Learning Classification of Land Cover and Crop Types Using Remote Sensing Data. *IEEE Geoscence Remote Sens. Lett.* 14:778–782.

Li, Y., J. Xue, K. Wang, M. Zhang and Z. Li. 2022. Surface Defect Detection of Fresh-Cut Cauliflowers Based on Convolutional Neural Network with Transfer Learning. *Foods* 11:2915.

Liu, C., Y. Tao, J. Liang, K. Li and Y. Chen. 2018. Object Detection Based on YOLO Network. *2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC)*. IEEE. pp.799–803.

Melek, C.G., E.B. Sonmez and S. Albayrak. 2019. Object Detection in Shelf Images with YOLO. *IEEE EUROCON 2019 -18th International Conference on Smart Technologies*. IEEE. pp.1–5.

Melillos, G. and D.G. Hadjimitsis. 2022. Ship detection using SAR images based on YOLO at Cyprus's coast. In: K. Palaniappan, G. Seetharaman, and J.D. Harguess (eds.), *Geospatial Informatics XII*. SPIE. pp.4.

Mittal, P., A. Sharma, R. Singh and A.K. Sangaiah. 2022. On the performance evaluation of object classification models in low altitude aerial data. *J. Supercomput.* 78:14548–14570.

Xin, M. and Y. Wang. 2019. Research on image classification model based on deep convolution neural network. *EURASIP J. Image Video Process.* 2019:40.

Bajwa, M. T. T., Kiran, Z., Rasool, A., and Rasool, R. (2025). Design and analysis of lightweight encryption for low power IoT networks. International Journal of Advanced Computing and Emerging Technologies, 1(2), 17–28.

Bajwa, M. T. T., Rasool, A., Kiran, Z., and Latif, A. (2025). Resilient cloud architectures for optimized big data storage and real-time processing. International Journal of Advanced Computing and Emerging Technologies, 1(2), 54–58.