



ISSN: 2184-0261

Application of Principal Component Analysis to advancing digital phenotyping of plant disease in the context of limited memory for training data storage

Enow Takang Achuo Albert*, Ngalle Hermine Bille,
Ngonkeu Mangaptche Eddy Leonard

Department of Plant Biology, Faculty of Science, University of Yaoundé I, P.O. Box 812, Yaoundé, Center Region, Cameroon

ABSTRACT

Despite its widespread employment as a highly efficient dimensionality reduction technique, limited research has been carried out on the advantage of Principal Component Analysis (PCA)-based compression/reconstruction of image data to machine learning-based image classification performance and storage space optimization. To address this limitation, we designed a study in which we compared the performances of two Convolutional Neural Network-Random Forest Algorithm (CNN-RF) guava leaf image classification models developed using training data from a number of original guava leaf images contained in a predefined amount of storage space (on the one hand), and a number of PCA compressed/reconstructed guava leaf images contained in the same amount of storage space (on the other hand), on the basis of four criteria – Accuracy, F1-Score, Phi Coefficient and the Fowlkes–Mallows index. Our approach achieved a 1:100 image compression ratio (99.00% image compression) which was comparatively much better than previous results achieved using other algorithms like arithmetic coding (1:1.50), wavelet transform (90.00% image compression), and a combination of three transform-based techniques – Discrete Fourier (DFT), Discrete Wavelet (DWT) and Discrete Cosine (DCT) (1:22.50). From a subjective visual quality perspective, the PCA compressed/reconstructed guava leaf images presented almost no loss of image detail. Finally, the CNN-RF model developed using PCA compressed/reconstructed guava leaf images outperformed the CNN-RF model developed using original guava leaf images by 0.10% accuracy increase, 0.10 F1-Score increase, 0.18 Phi Coefficient increase and 0.09 Fowlkes–Mallows increase.

KEYWORDS: PCA, Image reconstruction, Storage space optimization, Guava image classification

Received: January 16, 2023
Revised: April 25, 2023
Accepted: April 27, 2023
Published: May 12, 2023

***Corresponding Author:**
Enow Takang Achuo Albert
E-mail: albert.enow@
facsociences-uy1.cm

INTRODUCTION

Developing highly performing machine learning models for digital phenotyping of plant disease requires, to a great extent, large disk space for storing significant quantities of raw images prior to model training, or for further transmission if the model training site is different from the data collection site. As the ability to acquire more image data increases by the day, the need for reducing the large disc space required for storing this data becomes one which needs to be addressed.

Image compression/reconstruction has been widely adopted as the way to meet the afore-mentioned need. Indeed, significant

efforts have been made by various researchers in this regard, and various image compression algorithms have been proposed. Parmar and Pancholi (2016) mentioned that the factors guiding the choice of a compression algorithm are image quality, amount of compression and speed of compression.

Despite the plethora of proposed image compression algorithms, limited research has been carried out on the contribution of Principal Component Analysis (PCA) to image compression as a means of enhancing plant disease digital phenotyping model development, even though it is widely known that PCA is an efficient means of dimensionality (and hence, required storage space) reduction. This research work aimed to address that limitation.

Copyright: © The authors. This article is open access and licensed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted, use, distribution and reproduction in any medium, or format for any purpose, even commercially provided the work is properly cited. Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made.

Related Works on Image Compression

Lammi and Sarjakoski (1995) used the baseline JPEG image compression scheme to compress a scanned image. The image was compressed into three different levels: Excellent, High, and Poor, with compression ratios of about 1:7, 1:15, and 1:66, respectively. They used Storm Technology's PicturePress software with a Micron Xceed ICDP-II Picture Accelerator for the compression process. The visual quality of the Excellent image was very good. The High image was also quite good under visual examination although some compression effects were visible. The visual quality of the Poor image was poor – the size of the compression blocks (s by B pixels) was clearly visible, and all edges were heavily smoothed.

One year later, Maldjian *et al.* (1996) used a wavelet transform image compression technique. The results showed that greater than 90% image compression was achieved using the wavelet transform, with standard modem transmission times of less than 5 seconds per compressed image. The original image file size was 137 kB, and after wavelet and Gzip compression, file sizes were reduced to 15 kB and 8 kB, respectively. The 90% compressed image showed minimal loss of image detail. Even the 94% compressed image retained the key diagnostic features of the original image. However, some image blurring was evident, particularly at sulcal boundaries and at gray matter-white matter interfaces.

Dutta *et al.* (2012) proposed a two-step process for image compression. In the first step, the input image was divided into blocks of sub-images of various sizes (2x2, 3x3, 4x4, etc.), and each block was replaced by its mode value. In the second step, arithmetic coding was applied to compress the mode values. Their experimental results showed that this technique achieved a better compression ratio than JPEG and other techniques while maintaining a comparable peak signal-to-noise ratio (PSNR) value of the decompressed image.

Nagashree *et al.* (2014) proposed two different approaches for lossless image compression. One approach used the combination of 2D-DWT (discrete wavelet transform) and FELICS algorithm for lossy to lossless image compression, while the other approach used a combination of prediction algorithm and Integer Wavelet Transform (IWT). To show the effectiveness of the methodology used, different image quality parameters were measured, and a comparison of both approaches was shown. Their results showed an increased compression ratio and higher PSNR values with the second approach.

Alsayh *et al.* (2017) proposed a new hybrid image compression technique that combined three transform-based techniques: discrete Fourier transform (DFT), discrete wavelet transform (DWT), and discrete cosine transform (DCT). The authors used MATLAB software to implement the proposed technique and tested it on several standard test images. The results showed that the proposed technique achieved better compression ratios and peak signal-to-noise ratios (PSNRs) compared to other existing

techniques, such as DWT, DCT, and DWT-DCT. Specifically, the proposed technique achieved an average compression ratio of 1:22.5 and an average PSNR of 34.5 dB for the test images.

Still, Balle *et al.* (2017) used an image compression technique based on nonlinear transform coding. The method used a non-linear analysis transformation, a uniform quantizer, and a non-linear synthesis transformation to achieve efficient image compression. The authors optimized the method end-to-end for rate-distortion performance and found that it offered improvements over JPEG and JPEG 2000 for most images and bit rates. Additionally, the compressed images produced by this method were much more natural in appearance than those compressed with linear transform coding methods. Perceptual quality (as estimated with the MS-SSIM index) exhibited substantial improvement across all test images and bit rates.

Poolakkachalil and Chandran (2019) discussed the use of arithmetic coding for stereoscopic image compression. The proposed method exploited the spatial redundancy that occurs among the two-image pair to achieve a greater compression ratio in comparison to the individual compression of each frame. Their results showed that the proposed method achieved better compression ratios than other existing methods, such as JPEG and JPEG2000 while maintaining good visual quality. Specifically, their proposed method achieved a compression ratio of 1:1.5 for stereoscopic images, which is higher than the 1:1.3 ratio achieved by JPEG and the 1:1.4 ratio achieved by JPEG2000.

Taoufiq (2019) used DWT, specifically the Haar wavelet transform. DWT is a lossy compression technique that decomposes an image into different frequency sub-bands, which can be compressed with different levels of detail. The Haar wavelet transform is a specific type of DWT that uses a simple two-point wavelet function to decompose an image. The testing of the compressor was based on several criteria, including compression ratio and visual quality. The results showed that the Haar wavelet transform-based compressor was able to achieve high compression ratios while maintaining visually acceptable quality for various types of images. For example, for Lena and Baboon images, the compression ratios achieved were 1:16 and 1:14 respectively with visually acceptable quality. However, it should be noted that these results may vary depending on the specific images and compression settings used.

Asnaoui (2020) proposed a method based on the block singular value decomposition (SVD) power method. The algorithm was developed and tested for image compression using various real images and simulations in MATLAB 2009a. According to Asnaoui (2020), the proposed algorithm showed better compression performance compared to existing algorithms such as JPEG2000, JPEG-LS, and SPIHT. Also, the algorithm overcame the disadvantages of MATLAB's SVD function and provided a good trade-off between compression ratio and visual quality of the decoded image. However, it should be noted that the results presented in this article were based on specific experiments and may not generalize to all types of images or compression scenarios.

Guo *et al.* (2020) proposed a variable rate model that introduced a pair of gain units into a variational auto-encoder (VAE) and applied content adaptive optimization to adapt the latent representation to the specific content. Attention mechanisms and multi-scale parallel context modules were also adopted to improve the performance of the model. An efficient rate control algorithm was designed to maximize PSNR/structural similarity index measure (MS-SSIM) under 0.15 bits per pixel (BPP) constraint. The results showed that the proposed method outperformed several state-of-the-art methods in terms of both objective metrics (PSNR, MS-SSIM) and subjective visual quality. The method achieved a continuously variable rate in a single model, allowing each image to be compressed into any quality level through a unified codec. The content adaptive optimization strategy also generated better latent representation without architecture refinements.

Krishnan *et al.* (2020) proposed an image compression method based on compressive sampling and the Lü system. Specifically, the input image was sparsely represented on a transform basis, and compressive sampling measurements were obtained from these sparse transform coefficients using an incoherent sensing matrix. Permutation-substitution operations were performed on pixels based on the Lü system to upgrade security levels, and keys were obtained from the input image to add input sensitivity to the scheme. Lastly, a fast and efficient greedy algorithm was utilized for sparse signal reconstruction. The experimental results of this method showed that it achieved good compression and encryption performance. Extensive experimental tests were conducted using natural color images with pixels sized 512×512 , and the parameters $a = 35$, $b = 3$, and $c = 20$ were used in the experiments. The input images were sparsely represented by employing the biorthogonal wavelet transform as the orthogonal transform basis (ψ), along with single-level decomposition.

Tellez *et al.* (2020) used a Neural Image Compression (NIC) technique, which is an image compression framework that reduces the dimensionality of gigapixel images using an encoder network trained in an unsupervised fashion. In this study, the authors extended NIC by training the encoder with a supervised multitasks learning approach. They trained the encoder to solve four classification tasks in Computational Pathology simultaneously and used this model to perform gigapixel image compression. The authors found that supervised multitask training was key to obtaining high performance at the image level, surpassing unsupervised techniques. They also found that increasing the number of tasks used to train the encoder was directly proportional to the system's performance. The proposed MTL NIC obtained state-of-the-art results in predicting both the speed of tumour proliferation in invasive breast cancer (TUPAC16 Challenge) and HGP status in colorectal liver metastasis classification. Overall, this study demonstrated that extending unsupervised neural image compression with supervised multitask learning can improve the performance of convolutional neural networks trained on gigapixel images.

Helminger *et al.* (2021) proposed the use of normalizing flows as generative models in lossy image compression. By learning bijective transforms from image space to latent space, the method could cover a wide range of quality levels and effectively

enable going from low bit-rates to near lossless quality. The authors compared their method with other state-of-the-art methods on the Kodak dataset and showed that their method outperformed them in terms of rate-distortion performance. They also showed that their method achieved better visual quality than JPEG at low bitrates.

Hussein *et al.* (2017) proposed an enhanced version of the Run Length Encoding (RLE) algorithm. They tested the algorithm on ten BMP 24-bit true color images and built an application using visual basic 6.0 to show the size before and after compression and compute the compression ratio for RLE and the enhanced RLE algorithm. The results showed that the proposed enhanced RLE algorithm decreased the size of compressing images, especially for color images, compared to the original RLE method. The compression ratio varied from image to image, depending on the variety between the values of adjacent pixels. Decreasing the variety between adjacent pixels yielded an increase in compression ratio, and vice versa.

Karthikeyan *et al.* (2021) proposed a frequency-based lossless new encoding technique that doesn't require any table similar to Huffman and Golomb Rice encoder and doesn't take high computation time like an arithmetic encoder. The proposed encoder was tested with nearly 200 standard images, and the results were compared with the standard encoders. The results showed that the proposed encoder achieved better compression ratios than other standard encoders while maintaining low computational power and memory usage.

Finally, Rahman *et al.* (2022) presented the results of the evaluation of various lossless image compression techniques. The methods were compared based on four datasets: EPFL Light-field, UVG-TUT, Kodak Lossless True Color Image Suite, and LCLik. The evaluation was done based on bits per pixel (BPP) and compression ratio. They recommended different algorithms for different datasets based on their performance in the evaluation. For example, for the EPFL Light-field dataset, they recommended using the better portable graphics (BPG) algorithm as it provided the best compression ratio. For the Kodak Lossless True Color Image Suite dataset, they recommended using Free Lossless Image Format (FLIF) as it provided the best bits per pixel value. Overall, the authors concluded that learning-based methods generally outperform non-learning-based methods in terms of compression ratio and bits per pixel values. However, non-learning-based methods are still competitive and can be used when computational resources are limited or when a faster encoding/decoding process is required.

MATERIALS AND METHODS

System Specification

The scripts used for this research were written in Python 3 with system specifications as follows: 64-bit operating system, x64-based processor, 8GB RAM, intel CORE i7 processor. After the results were obtained, the scripts were pushed to a GitHub repository (<https://github.com/Enowtakang/PCA-Reconstruct>).

Dataset

The dataset used for this study was obtained from the image dataset assembled by Chouhan *et al.* (2019). All the guava leaf images were collected from the Shri Mata Vaishno Devi University, Katra. The process was carried out from the month of March to May 2019. The images were captured in a closed environment. The acquisition process was completely Wi-Fi enabled. All the images were captured using a Nikon D5300 camera inbuilt with performance timing for shooting JPEG in single shot mode (seconds/frame, max resolution) = 0.58 and for RAW+JPEG = 0.63. The images were in JPG format captured with an 18-55 mm lens with sRGB color representation, 24-bit depth, 2 resolution unit, 1000-ISO, and no flash.

As detailed in Figure 1, all of the data was placed within a root folder. Two sub-folders were created within the root folder, each containing training and validation datasets. Each of the latter datasets was subdivided into healthy and diseased samples of guava leaf images.

The first subfolder contained original images from the Chouhan *et al.* (2019) collection. Its training folder contained one healthy and one disease subfolder holding a 1.5 MB image of a healthy guava leaf and a 1.5 MB image of a diseased guava leaf respectively. Its validation folder contained one healthy and one disease subfolder holding twenty 1.5 MB images of healthy guava leaves and twenty 1.5 MB images of diseased guava leaves respectively.

The second subfolder contained PCA compressed/reconstructed images obtained from compressing original images from the Chouhan *et al.* (2019) collection. Its training folder contained one healthy and one disease subfolder containing fifty (approx.) 13 kB images of healthy guava leaves and fifty (approx.) 13 kB images of diseased guava leaves respectively. Its validation folder had exactly the same contents as those of the first subfolder.

Study Design

Figure 2 presents the design for this study. Two training datasets were prepared – the first training dataset contained a healthy and a diseased subset of original training images in approximately 3.0 MB of storage space. The second training dataset contained a healthy and a diseased subset of PCA reconstructed/compressed (processed) training images in approximately 3.0 MB of storage space. Together with the same testing dataset, each of the two aforementioned training datasets was used to train a Convolutional Neural Network – Random Forest (CNN-RF) hybrid algorithm (feature extraction was achieved with the CNN and the features were further input into the RF for classification), resulting in two models, which were later evaluated on the basis of several performance metrics.

Principal Component Analysis (PCA)

PCA is a dimensionality reduction technique. As per Nasiriany *et al.* (2019), reduction in estimation variance and computational load, and visualization for exploratory data

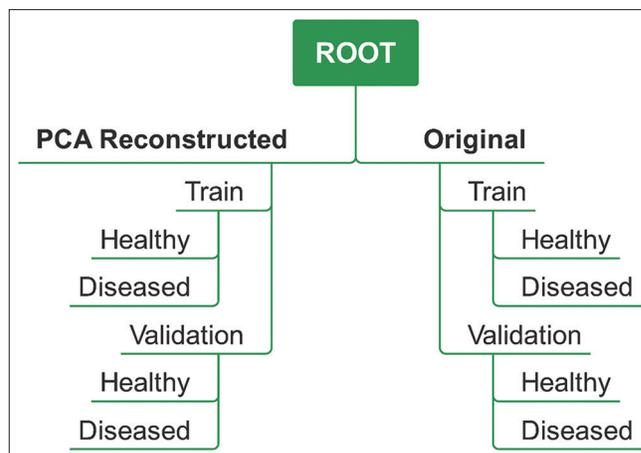


Figure 1: Structure of the root folder containing the dataset used for this study

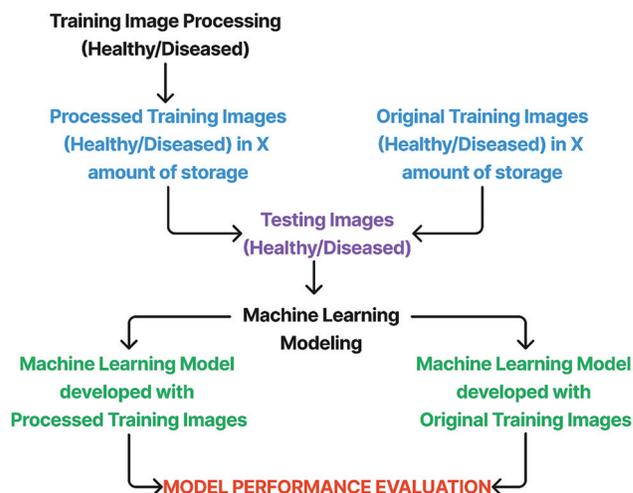


Figure 2: Study design

analysis are important reasons why dimensionality reduction is sought after.

When the data is centered, PCA can be viewed through Singular Value Decomposition (SVD). The SVD can be represented as

$$X = U \Sigma W^T$$

where $X \sim$ Data ($n \times m$), $U \sim$ Left Singular Vectors ($n \times k$), $\Sigma \sim$ Singular Values ($k \times k$), and $W \sim$ Right Singular Values ($m \times k$).

Noteworthy is the fact that k is the number of principal components under consideration. In the above SVD formulation, W captures the principal components, $U \Sigma$ render the principal component scores and ΣW^T renders the loadings. Loadings provide an idea of how much each column contributes to each of the principal components.

Σ gives information about the variance. Σ is always a diagonal matrix, such as

$$\Sigma = \begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{pmatrix}$$

where every entry off the diagonal is always a zero.

The percentage of variance explained by each principal component (PC) is given by

$$\frac{\sigma^2 PC}{\sum \sigma_i^2}$$

where σ^2 represents the variance.

Random Forest Algorithm (RF)

The rationale behind the random forest algorithm is that the combination of learning models increases the classification accuracy, also known as bagging. More specifically, bagging is the averaging of noisy and unbiased models to create a model with low variance. Random forest algorithm works as a large collection of decorrelated decision trees.

Suppose that the matrix consists of data submitted to the algorithm to produce a classification model:

$$S = \begin{bmatrix} f_{A1} & f_{B1} & f_{C1} & C_1 \\ \vdots & \vdots & \vdots & \vdots \\ \cdot & \cdot & \cdot & \cdot \\ f_{AN} & f_{BN} & f_{CN} & C_N \end{bmatrix}$$

where f_{A1} represents feature A of the 1st sample, f_{BN} represents feature B of the Nth sample, and C_N represents the label of the Nth sample. From the above sample set, subsets are created with random values, such as

$$S_1 = \begin{bmatrix} f_{A12} & f_{B12} & f_{C12} & C_{12} \\ f_{A15} & f_{B15} & f_{C15} & C_{15} \\ \vdots & \vdots & \vdots & \vdots \\ f_{A35} & f_{B35} & f_{C35} & C_{35} \end{bmatrix}, S_2 = \begin{bmatrix} f_{A2} & f_{B2} & f_{C2} & C_2 \\ f_{A7} & f_{B7} & f_{C7} & C_7 \\ \vdots & \vdots & \vdots & \vdots \\ f_{A20} & f_{B20} & f_{C20} & C_{20} \end{bmatrix},$$

$$S_M = \begin{bmatrix} f_{A4} & f_{B4} & f_{C4} & C_4 \\ f_{A9} & f_{B9} & f_{C9} & C_9 \\ \vdots & \vdots & \vdots & \vdots \\ f_{A13} & f_{B13} & f_{C13} & C_{13} \end{bmatrix}$$

These subsets are then used to create multiple decision trees (DT) (such as, DT_1, DT_2 and DT_m , in this case). The prediction

with the highest incidence after taking into account the predictions from all DTs, is the final RF class prediction.

Convolutional Neural Networks (CNN)

A Convolutional Neural Network, or CNN, is a type of neural network architecture used in deep learning for computer vision tasks. The architecture typically consists of several convolutional layers, pooling layers, and fully connected layers. The convolutional layers use filters to extract features from the input data, while the pooling layers downsample the spatial dimensions of the feature maps to reduce computation and prevent overfitting. The fully connected layers perform classification or regression tasks based on the learned features. CNN architectures can vary widely, but they generally follow the principle of applying convolutional layers to the input, downsampling the spatial dimensions, and increasing the number of feature maps. Some classic network architectures include LeNet-5, AlexNet, and VGG-16, while modern architectures include Inception, ResNet, ResNeXt, and DenseNet. Inception uses a repeating unit called the ‘‘Inception cell,’’ which performs convolutions at different scales and aggregates results using 1x1 convolutions to reduce input channel depth. Auxiliary outputs can be added throughout the network to improve performance and regularize the network. The CNN feature extraction and classification architecture designed for this study is presented in Figure 3.

Model Performance Evaluation Metrics

The two machine learning models developed during this study were evaluated using four widely used metrics – Accuracy, F1-Score, Phi Coefficient and the Fowlkes–Mallows index (FM). In order to define them, some fundamental concepts need to first be defined in the context of this research.

A true positive (TP) is a model prediction that correctly indicates the presence of an image from a given class. A true negative (TN) is a model prediction that correctly indicates the absence of an image from a given class. A false positive (FP) is a model prediction that wrongly indicates that an image from a given class is present. A false negative (FN) is a model prediction that wrongly indicates that an image from a given class is absent.

Accuracy (ACC) measures how close model predictions are to their true values. It is computed with the formula

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

F1-Score (F_1) is the harmonic mean of precision and recall. Precision is the number of true positives divided by the total number of positives. Recall is the number of true positives divided by the number of samples which should be identified as positive. It is computed with the formula

$$F_1 = \frac{2TP}{2TP + FP + FN}$$

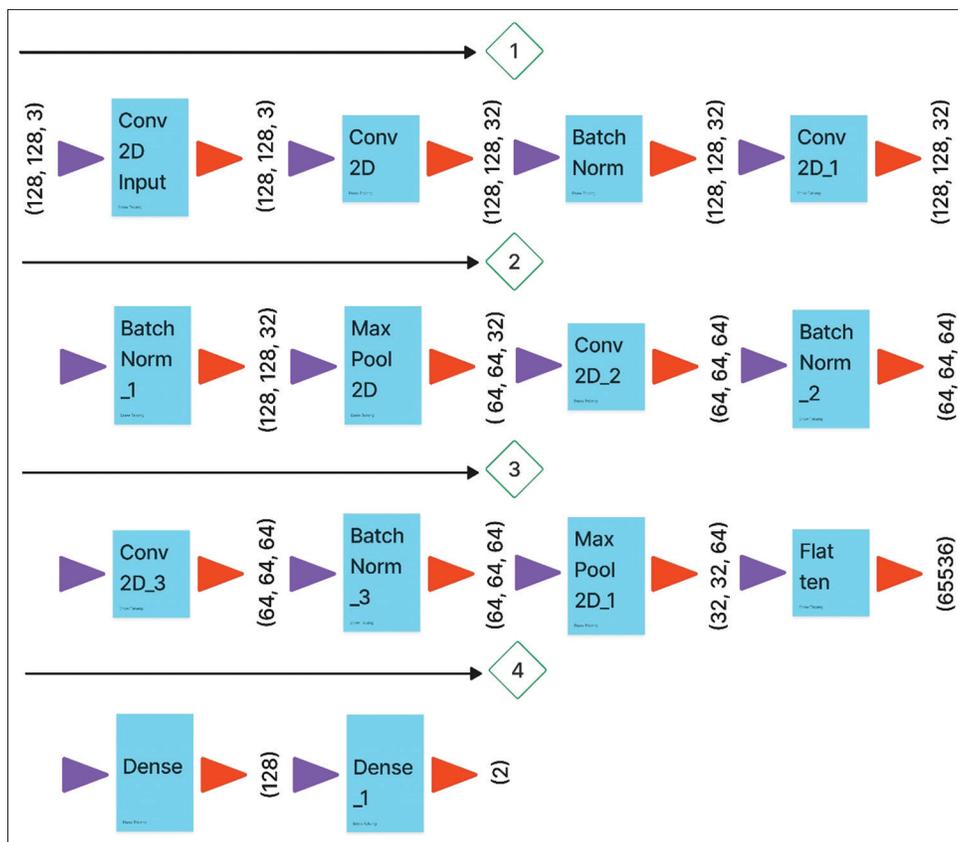


Figure 3: CNN architecture used in this study

Phi Coefficient (r_ϕ) is a measure of the quality of binary classifications. It is a correlation between the observed and predicted binary classifications and is a balanced measure (meaning it can be used even in cases of class imbalance). It is computed with the formula

$$r_\phi = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Lastly, the Fowlkes–Mallows index (FM) is a measure of similarity between the actual data groups and benchmark classifications performed by the model. It is directly proportional to the number of true positives. It is computed with the formula

$$FM = \sqrt{\frac{TP}{TP + FP} * \frac{TP}{TP + FN}}$$

RESULTS

Image Compression/Reconstruction

The average data compression ratio (the ratio between the uncompressed image size and the compressed image size) after PCA compression/reconstruction was 1:100.

Figure 4 presents results from image compression/reconstruction of healthy guava leaf samples and diseased guava leaf samples.

It can be observed (from Figure 4) that the subjective visual quality after image compression/reconstruction is very high. Also, there is almost no loss of image detail. Most importantly, the compressed/reconstructed images (*H-Recon* and *D-Recon*) retain the key diagnostic features of their respective original images.

Validation Accuracy and Loss on the CNN Model

As Figures 5 and 6 shows, the model developed with the compressed/reconstructed images had comparatively better validation accuracy and loss performance than the model developed with the original images, especially between epochs 29 and 32.

Model Evaluation

Figure 7 and Table 1 both present the results of the evaluation of both models. Let Model A represent the model developed using the original images, and Model B represent the model developed using the compressed/reconstructed images. As shown in Figure 7, the total true positives, true negatives, false positives and false negatives for Model A were 20, 16, 0 and 4, respectively. Also, the total true positives, true negatives, false positives and false negatives for Model B were 20, 20, 0 and 0, respectively. As shown in Table 1, Model B obtained perfect results on all evaluation metrics, outperforming Model A by 0.10% accuracy increase, 0.10 F1-Score increase, 0.18 Phi Coefficient increase and 0.09 Fowlkes–Mallows increase.

DISCUSSION

This work produced excellent quality images after PCA-based compression at a compression ratio of 1:100. This result is significant, given that previous works using different image compression algorithms have produced excellent quality images

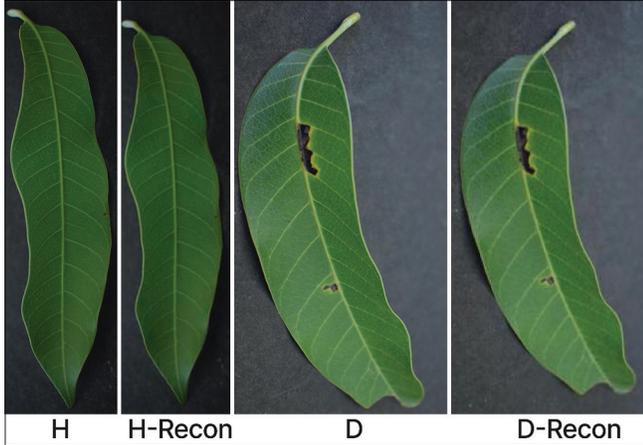


Figure 4: Compression/reconstruction of a healthy guava image sample (H: original healthy image, H-Recon: reconstructed healthy image) and a diseased guava image sample (D: original diseased image, D-Recon: reconstructed diseased image)

with comparatively higher compression ratios. For example, Lammi and Sarjakoski (1995) achieved excellent image quality at a 1:7 compression ratio after using Storm Technology’s PicturePress software with a Micron Xceed ICDP-II Picture Accelerator for the compression process. Also, after using the arithmetic coding algorithm, Poolakkachalil and Chandran (2019) achieved a compression ratio of 1:1.5. Furthermore, after applying the wavelet transform algorithm, Maldjian *et al.* (1996) achieved 90% image compression. In like-wise comparison, this work achieved 99% image compression. After combining three transform-based techniques – Discrete Fourier (DWT), Discrete Wavelet (DWT) and Discrete Cosine (DCT) – into a single image compression algorithm, Alsayyih *et al.* (2017) achieved an average compression ratio of 1:22.5.

Given that the model developed with the compressed/reconstructed images had comparatively better validation accuracy and loss performance than the model developed with the original images, especially between epochs 29 and 32, we conclude that the result suggests that early stopping of model training at any epoch in the aforementioned range would yield an even better performing model. We also propose that the reason for the better performance of the model developed with the compressed/reconstructed images is due to the fact that more, excellent quality image data was used for the same amount of storage space. This gives credibility to the performance of our compression approach.

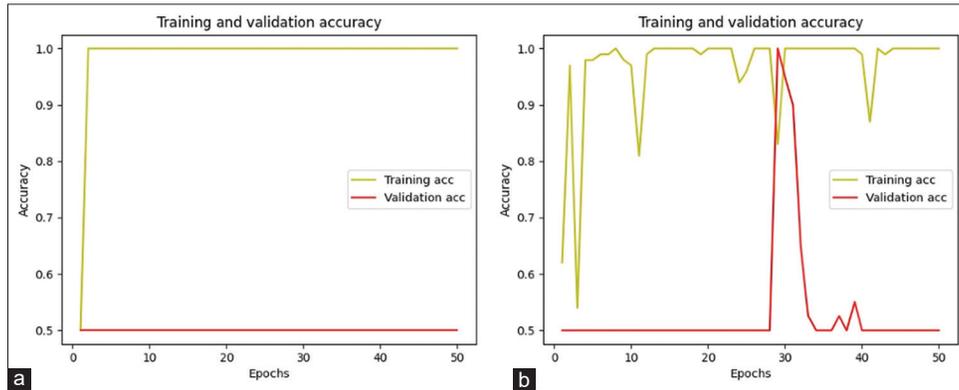


Figure 5: CNN model validation accuracies. a: model from training dataset containing original images. b: model from training dataset containing compressed/reconstructed images

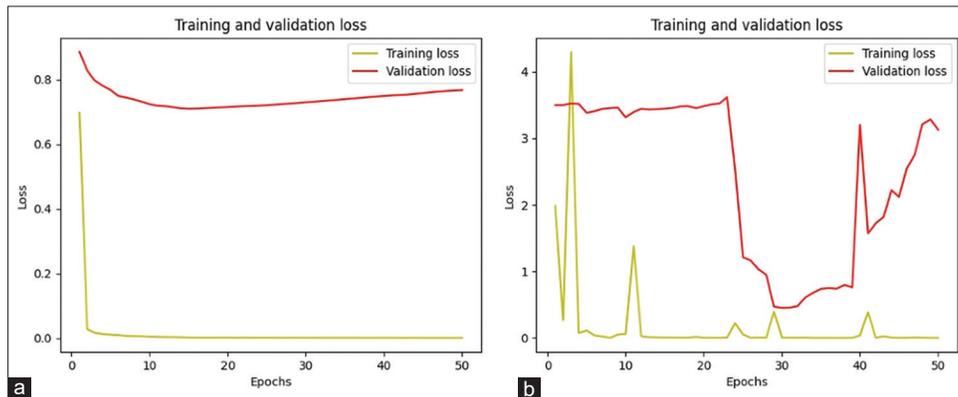


Figure 6: CNN model validation losses. a: model from training dataset containing original images. b: model from training dataset containing compressed/reconstructed images

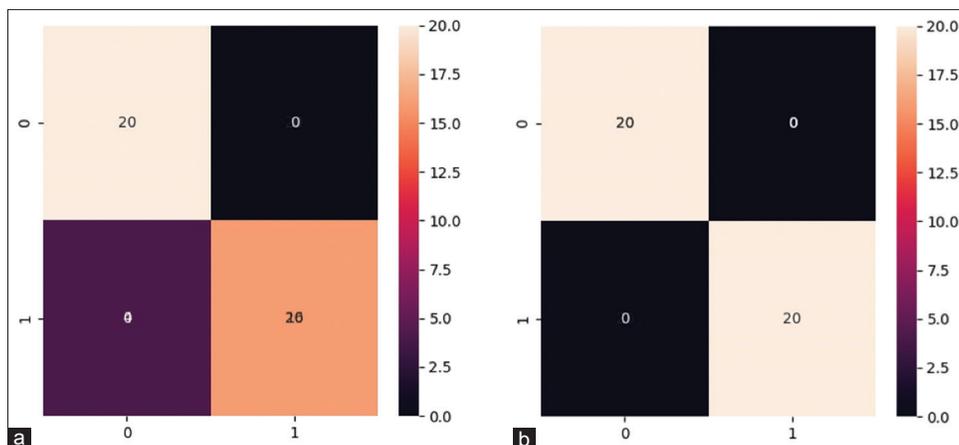


Figure 7: Random Forest confusion matrix showing the performances of Model A (a) and Model B (b).

Table 1: Numerical summary of the evaluation results for Model A and Model B

Model evaluation metric	Model A	Model B
Accuracy (%)	90.00	100.00
F1-Score	0.90	1.0
Phi Coefficient	0.82	1.0
Fowlkes–Mallows	0.91	1.0

Preceding works have made key contributions in the context of image compression and image classification. For example, Nesteruk *et al.* (2021) made several key contributions to the field of Controlled-Environment Agriculture (CEA) and machine learning. Firstly, they proposed a novel approach to solving the plant classification problem in CEA using Convolutional Variational AutoEncoders (VAE) and Support Vector Machine (SVM) or XGBoost algorithms. The proposed approach achieved 92.6% accuracy on an 18-classes unbalanced dataset using images collected from the EDEN ISS facility located in the Antarctic. Secondly, they addressed the problem of limited communication bandwidth in transmitting images from the South Pole to Europe continuously. To address this problem, they proposed an image compression method that helps compress the images with a ratio of 1:7.2 (higher than our achieved results), allowing for more images to be transmitted per day. Thirdly, they demonstrated the practical feasibility of the developed methods in real settings, specifically in the context of the EDEN ISS experimental facility located at the Neumayer Station III site in the Antarctic. The EDEN ISS facility includes the Mobile Test Greenhouse, which is devoted to autonomous cultivation for more than thirty higher plant species. The proposed approach based on image compression and their transmission from the Antarctic to Europe helps to solve a number of critical agriculture-related problems, including the CEA, e.g., classification, identification of plant diseases, and deviation of plant phenology.

Yang *et al.* (2021) challenged the conventional understanding that JPEG compression generally degrades the classification performance of deep neural networks (DNNs). They showed that if JPEG compression is used in the right manner, it can actually improve the classification accuracy of DNNs. They

proposed a selector called the Highest Rank Selector (HRS) to select a compressed version of an image as an input to the DNN. HRS works by determining the rank of the ground truth label of an image in the sorted prediction vector of the DNN in response to the input image. HRS selects the compressed version of the image that results in the highest rank of the ground truth label in the sorted prediction vector. Also, they showed that the classification accuracy of a DNN can be significantly improved when a suitable version of a JPEG compressed image is selected as input to the DNN using HRS. The size in bits of the selected input was also reduced dramatically in comparison with the original image. Finally, they suggested that the current CNN classifiers are not smart enough and behave as a short-sighted person. If the main features of an object are relatively enhanced and the disturbing features surrounding the object are removed, all through compression, then the CNN classifiers can see the object better. The authors suggested that it would be interesting to investigate whether this could be theorized to any Turing classifier (i.e., a computable classifier).

Du *et al.* (2022) proposed a collaborative image compression and classification framework for Visual Internet of Things (V-IoT) applications. The key contributions of their work are as follows: 1.) Multi-task GANs: They proposed a multi-task Generative Adversarial Network (GAN) that included an encoder, quantizer, generator, discriminator, and classifier to achieve collaborative image compression and classification. The multi-task GAN shared the same features, which could reduce considerable computing resources. 2.) Quantized latent representation: They proposed a quantized latent representation used for compression and classification. The proposed framework achieved low bitrate compression and reduced the amount of data transmitted while preserving fidelity at the pixel and semantic levels. 3.) Novel optimization target: They proposed a novel optimization target that minimized the combination of Mean Square Error (MSE) loss and perceptual loss to preserve the fidelity at the pixel and semantic level. 4.) End-to-end learning: The proposed framework could be implemented by end-to-end learning, which greatly reduced the computing resources. Overall, the proposed framework combined image compression with semantic inference by using multi-task learning.

Mohsen and Tiwari (2021) focused on quantum image processing. They made the following contributions: 1) A novel construction to compress images and encode them in their FRQI (Fixed-Reference Quantum Image) states using only 2-qubit gates. This encoding mechanism embeds images in quantum states while requiring fewer qubits than in prior work. The input images can have a resolution of up to 16 x 16, and the quantum encoding only requires 8 qubits. This approach enables quantum machine learning and classification on classical datasets of dimensions that were previously intractable by physically realizable quantum computers or classical simulation. 2) New QNN layers, CRADL and CRAML, which are used in a model trained with the images' FRQI states as input. The QNN is a sequence of unitary operations parametrized by angles, and the input to the QNN is an $n \times n$ -pixel image that is encoded in a $d \log_2 n e + 1$ dimensional Hilbert space by an encoding function. They proposed a novel encoding mechanism that embeds images in quantum states while necessitating fewer qubits than in prior work. 3) They showed that their trained QNN achieves accuracy comparable to classical models with the same number of parameters. They trained their QNN on the MNIST dataset of handwritten digits and compared its performance to classical neural networks with the same number of learnable parameters. The QNN was able to classify larger, more realistic images than previously possible, up to 16 x 16 for the MNIST dataset on a personal laptop. 4) They proposed a novel technique to further compress black and white images, and study the scaling behavior of their model with the extent of image compression. They found that their QNN was able to achieve accuracy comparable to classical models even with compressed images. Overall, the authors' proposed QNN approach for image classification using quantum states shows promise for achieving comparable performance to classical models with the same number of parameters, while also enabling the classification of larger, more realistic images.

Finally, Fu and Guimaraes (2016) proposed a novel approach to speed up image classification in artificial neural networks by compressing image data with an algorithm based on the discrete cosine transform (DCT) before feeding it to the networks. This approach is different from traditional methods that focus on improving the accuracy of algorithms, with no regard to training time. They demonstrated that their approach could achieve significant speedups in training time ranging from 2 times to 10 times, depending on the dataset, with only minor effects on algorithm accuracy. This was achieved by reducing the dimensionality of feature vectors using DCT-based compression, which decreases redundancy in the original vectors and thus speeds up training. Overall, their work proposed a new way to speed up image classification in artificial neural networks that can be applied to various datasets and has potential applications in real-world scenarios where fast training times are critical.

CONCLUSION

The constraint of limited memory for training data storage which is often experienced in data-intensive projects can be addressed in two ways: (1) more memory chips can be produced,

and (2) the data can be compressed to an appreciable level at which little to no relevant quality is lost. With respect to the latter approach, the former approach requires more investments in time, skilled-labor and finances. We recommend the latter approach.

REFERENCES

- Alsayh, M. M., Mohamad, D., Saba, T., Rehman, A., & AlGhamdi, J. S. (2017). A Novel Fused Image Compression Technique Using DFT, DWT, and DCT. *Journal of Information Hiding and Multimedia Signal Processing*, 8(2), 261-271.
- Asnaoui, K. E. (2020). Image Compression Based on Block SVD Power Method. *Journal of Intelligent Systems*, 29(1), 1345-1359. <https://doi.org/10.1515/jisys-2018-0034>
- Balle, J., Laparra, V., & Simoncelli, E. P. (2017). End-To-End Optimized Image Compression. 5th International Conference on Learning Representations (pp. 1-27). arXiv. <https://doi.org/10.48550/arXiv.1611.01704>
- Chouhan, S. S., Singh, U. P., Kaul, A., & Jain, S. (2019, November 21-22). A Data Repository of Leaf Images: Practice towards Plant Conservation with Plant Pathology. 4th International Conference on Information Systems and Computer Networks (ISCON) (pp. 700-707). IEEE. <https://doi.org/10.1109/ISCON47742.2019.9036158>
- Du, B., Duan, Y., Zhang, H., Tao, X., Wu, Y., & Ru, C. (2022). Collaborative image compression and classification with multi-task learning for visual Internet of Things. *Chinese Journal of Aeronautics*, 35(5), 390-399. <https://doi.org/10.1016/j.cja.2021.10.003>
- Dutta, S., Abhinav, A., Dutta, P., Kumar, P., & Halder, A. (2012). An Efficient Image Compression Algorithm Based on Histogram Based Block Optimization and Arithmetic Coding. *International Journal of Computer Theory and Engineering*, 4(6), 954-957. <https://doi.org/10.7763/IJCTE.2012.V4.614>
- Fu, D., & Guimaraes, G. (2016, December 10). Using Compression to Speed up Image Classification in Artificial Neural Networks. Cambridge, Massachusetts, USA.
- Guo, T., Wang, J., Cui, Z., Feng, Y., Ge, Y., & Bai, B. (2020, June 14-19). Variable Rate Image Compression with Content Adaptive Optimization. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (pp. 4321-4325). IEEE. <https://doi.org/10.1109/CVPRW50498.2020.00069>
- Helminger, L., Djelouah, A., Gross, M., & Schroers, C. (2021). Lossy Image Compression with Normalizing Flows. *Neural Compression Workshop* (pp. 1-10). arXiv. <https://doi.org/10.48550/arXiv.2008.10486>
- Hussein, A. H., Mahmud, S. S., & Mohammed, R. J. (2017). Image Compression Using Proposed Enhanced Run Length Encoding Algorithm. *Ibn Al-Haitham Journal for Pure and Applied Sciences*, 24(1), 1-14. <https://doi.org/10.30526/24.1.803>
- Karthikeyan, N., Saravanakumar, N. M., & Sivakumar, M. (2021). Fast and efficient lossless encoder in image compression with low computation and low memory. *IET Image Processing*, 15(11), 2494-2507. <https://doi.org/10.1049/ipr2.12235>
- Krishnan, K. S., Jaison, B., & Raja, S. P. (2020). Secured Color Image Compression Based on Compressive Sampling and Lü System. *Information Technology and Control*, 49(3), 346-369. <https://doi.org/10.5755/j01.itc.49.3.25901>
- Lammi, J., & Sarjakoski, T. (1995). Image Compression by the JPEG Algorithm. *Photogrammetric Engineering & Remote Sensing*, 61(10), 1261-1266.
- Maldjian, J. A., Liu, C. W., Hirschorn, D., Murthy, R., & Semanczuk, W. (1997). Wavelet Transform-Based Image Compression for Transmission of MR data. *American Journal of Roentgenology*, 169(1), 23-26. <https://doi.org/10.2214/ajr.169.1.9207495>
- Mohsen, A., & Tiwari, M. (2021). Image Compression and Classification Using Qubits and Quantum Deep Learning. arXiv. <https://doi.org/10.48550/arXiv.2110.05476>
- Nagashree, G., Gadre, V., & Vijaya, S. M. (2014). Lossless Medical Image Compression. *International Journal of Engineering Research and Applications*, 4(6), 7-11.
- Nasiriany, S., Thomas, G., Wang, W., Yang, A., Listgarten, J., & Sahai, A. (2019). *A Comprehensive Guide to Machine Learning*. Berkeley, USA: University of California.

- Nesteruk, S., Shadrin, D., Pukalchik, M., Somov, A., Zeidler, C., Zabel, P., & Schubert, D. (2021). Image Compression and Plants Classification Using Machine Learning in Controlled-Environment Agriculture: Antarctic Station Use Case. *IEEE Sensors Journal*, 21(16), 1-9. <https://doi.org/10.1109/JSEN.2021.3050084>
- Parmar, C. K., & Pancholi, K. (2016). A review on image compression techniques. *Journal of Information, Knowledge and Research in Electrical Engineering*, 2(2), 281-284.
- Poolakkachalil, T. K., & Chandran, S. (2019). Summative Stereoscopic Image Compression using Arithmetic Coding. *Indonesian Journal of Electrical Engineering and Informatics*, 7(3), 564-576. <https://doi.org/10.52549/ijeei.v7i3.755>
- Rahman, A., Hamada, M., & Rahman, A. (2022). A comparative analysis of the state-of-the-art lossless image compression techniques. The 4th ETLTC International Conference on ICT Integration in Technical Education (ETLTC2022). <https://doi.org/10.1051/shsconf/202213903001>
- Taoufiq, S. (2016). *Image Compression Using Discrete Wavelet Transforms*. Morocco: Al Akhawayn University.
- Tellez, D., Hoppener, D., Verhoef, C., Grunhagen, D., Nierop, P., Drozdal, M., Laak, J. van der, & Ciompi, F. (2020). Extending Unsupervised Neural Image Compression With Supervised Multitask Learning. *Proceedings of Machine Learning Research*. arXiv. <https://doi.org/10.48550/arXiv.2004.07041>
- Yang, E.-H., Amer, H., & Jiang, Y. (2021). Compression Helps Deep Learning in Image Classification. *Entropy*, 23(7), 881. <https://doi.org/10.3390/e23070881>