

Quality Classification of Germinated Oil Palm Seeds Based on Deep Learning Models

Kelly Tan Kai Ling¹, Lisa Ho Yen Xin¹, and Tan En Xuan¹

¹ University of Nottingham, Malaysia

1 Introduction

1.1 Problem Statement

In the palm oil industry, optimising the quality of oil palm seeds is essential for productivity and sustainability. Recognising this imperative, recent advancements in machine learning, specifically the integration of classification networks, offer a promising approach to evaluate the quality of individual oil palm seeds accurately. Notably, the use of classification network which utilises multi-view images of an individual oil palm seed has shown promising results in determining the likelihood of a seed being bad [1]. However, further research is needed to analyse the effectiveness of classification model in identifying the seed quality under different conditions, including diverse lighting environments and imaging conditions. Therefore, this paper strives to develop a classification system capable of effectively discerning seed quality across a spectrum of real-world conditions, thereby enhancing the reliability and applicability of the seed quality identification process in oil palm production.

1.2 Aim

The aim of this project is to analyse and compare the performance and generalisability of a pre-trained classification model against a model integrating conventional image augmentation techniques and generative models for classifying the quality of germinated oil palm seeds based on their images.

1.3 Objectives

1. To select and fine-tune a suitable pre-trained classification model for germinated oil palm seed quality classification.
2. To evaluate the performance metrics of the pre-trained classification model and explore augmentation techniques using generative models, for enhancing model generalisability.
3. To compare the performance of the pre-trained classification model with and without augmentation techniques.

2 Methodology

The methodology involves training two identical pre-trained models, Model 1 on the original dataset and Model 2 on the augmented dataset, to assess their performance.

2.1 Dataset Description

The original dataset comprises images in .jpg format collected in 3 batches, with Batch 2 and Batch 3 featuring images captured under varying lighting conditions. Each image contains approximately 10 oil palm seeds. Table 1 below provides a concise overview of the dataset characteristics.

Table 1. Overview of the original dataset.

	Batch 1		Batch 2		Batch 3	
	Good	Bad	Good	Bad	Good	Bad
Number of seeds	1102	1051	450	450	605	593
Number of images	215		90		120	
Lighting condition	Not specified		Normal room light		Light box	

Batch 1 is divided into training and testing subsets, comprising 175 and 40 images, respectively. Each subset is then further categorised into folders of good and bad seeds. In contrast, Batch 2 and Batch 3 are grouped into 15 and 20 sets respectively, each set comprising images along with annotations denoting the classification of individual seed, provided in .xml format.

2.2 Classification Model Selection

Several architectures for convolutional neural networks (CNNs) have been selected to address image classification tasks, including AlexNet, Inception, ResNet, and VGG. **AlexNet**, for instance, is known for its pioneering role in deep learning but may struggle with deeper networks due to vanishing gradients [2]. **Inception** networks excel in efficiency by utilising multiple convolutions in parallel, yet may be prone to overfitting on smaller datasets [3]. **VGG** networks are recognised for their simplicity and effectiveness, relying on stacking multiple layers, but they can be computationally expensive [4]. **ResNet**, on the other hand, introduces skip connections that mitigate vanishing gradient issues, allowing the successful training of very deep networks [3].

Considering the dataset's size and complexity, ResNet is deemed the best option for the current oil palm classification task. It's efficiency in extracting semantic information from complex image data can enhance classification accuracy and generalisation performance, particularly with a moderately sized dataset [5].

ResNet architectures are typically distinguished by the number of layers they incorporate. For instance, ResNet-18 comprises 18 layers, encompassing convolutional, pooling, and fully connected layers [6]. Deeper ResNet models, such as ResNet-34, offer enhanced representational power but bring higher computational demands and overfitting risks, particularly with smaller datasets [5]. Therefore, ResNet-18 is selected for the oil palm classification task due to its balanced complexity and computational efficiency. ResNet-18 strikes a suitable balance, capturing significant features effectively while remaining computationally manageable.

2.3 Generative Model

The DCGAN (Deep Convolutional Generative Adversarial Network) is selected to augment Batch 1 dataset due to its capacity to produce varied synthetic images resembling the original data, thereby improving the training and generalisation of Model 2 [7].

The DCGAN architecture comprises a generator and discriminator network. The generator transforms latent vectors into realistic images using transposed convolutional layers with batch normalisation and ReLU activation, while the discriminator distinguishes between real and generated images through convolutional layers with batch normalisation and LeakyReLU activation. Configured parameters include a batch size of 128, image size of 64x64 pixels, and 30 epochs of training with a learning rate of 0.0002.

The original dataset is separated into two sub-folders: GoodSeed and BadSeed, containing 1102 and 1051 images, respectively. Due to the insufficient number of training images for effective DCGAN training, the dataset undergoes augmentation to enhance performance (*refer to Appendix A for DCGAN comparison between different dataset size*). This augmentation involves creating seven additional sets with various transformations, including random horizontal and vertical flips, rotations up to 15 degrees, color jitter, resized crops, affine transformations, and perspective changes. As a result of this augmentation process, each dataset expands significantly, increasing from 1102 to 8816 images for GoodSeed and from 1051 to 8408 images for BadSeed.

During training, the generator is trained for 30 epochs to produce synthetic images of individual oil palm seeds, which the generated images at final epoch are then saved into their respective folders. This entire process is repeated five times with different random seed numbers to generate different augmented images, resulting in a total of 1162 images, with 576 images for each folder.

Fig. 1 below illustrates the generator and discriminator loss graph throughout the training process for each iterations with different folders.

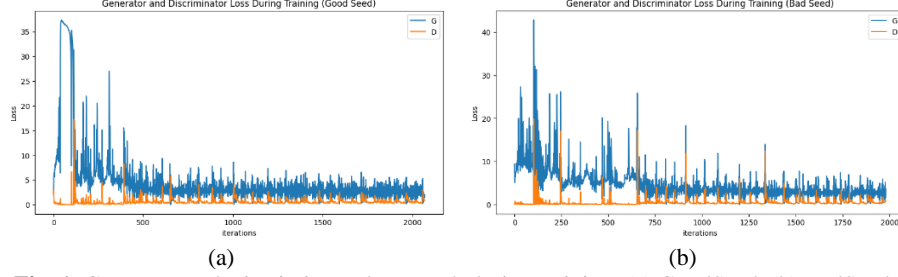


Fig. 1. Generator and Discriminator loss graph during training: (a) GoodSeed, (b) BadSeed

Based on the observations in Fig. 1, both the generator and discriminator initially experience a decrease in loss, indicating effective learning. However, as training progresses, fluctuations in loss are evident, implying unstable performance due to overfitting.

Fig. 2 below depicts the example of generated images for each folders.

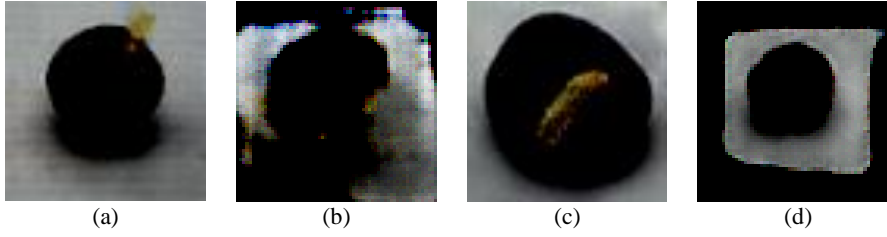


Fig. 2. Example of generated images: (a) Good example of GoodSeed, (b) Bad example of GoodSeed, (c) Good example of BadSeed, (d) Bad example of BadSeed

(Refer to Appendix B for the comparison between real and fake images)

Fig. 2 reveals that, overall, the generated images bear a resemblance to the shape and color of an oil palm seed. The good examples prominently display the radicle of the seed, enhancing their resemblance to real seeds. Conversely, the bad examples lack accurate representation of the radicle, focusing solely on the seed's shape. This limitation in accurately generating the radicle may affect the performance of the classification model, as the radicle serves as a crucial feature in distinguishing between good and bad quality seeds.

2.4 Data Pre-Processing

Data pre-processing serves to enhance the quality and suitability of datasets for subsequent machine learning tasks. Here is a brief overview of each step:

1. **Image cropping:** Extract individual seed images based on provided bounding box coordinates from Batch 2 and Batch 3 datasets.
2. **Dataset preparation:** Resize the cropped images to 256x256 pixels for Model 1 and 64x64 pixels for Model 2 then save them with labels into a CSV file.
3. **Split dataset:** Partition the Batch 1 dataset into training (80%) and validation sets (20%), while Batch 2 and Batch 3 are reserved for testing.
4. **Data transformation:** Normalise the pixel values to 0.5 for consistent intensity levels across all images and resize them to 224x224 pixels to ensure uniform processing within the model.

2.5 Model Architecture

The pre-trained ResNet-18 model undergoes fine-tuning for the classification task by implementing several modifications and configurations. The last fully connected layer (fc) of the ResNet-18 backbone is replaced with a new fully connected layer with an output dimension of 512. Additionally, two more fully connected layers (fc1 and fc2) are added to further refine the features learned by the backbone. To ensure effective transfer learning, parameters of the backbone layers, excluding the last fully connected layer and the last layer in ResNet's layer 4, are frozen. This prevents the pre-trained weights from being updated, thus preserving the valuable features learned from the ImageNet dataset. Layer 4 is unfrozen to allow fine-tuning of higher-level features specific to the dataset, enhancing the model's ability to capture relevant patterns.

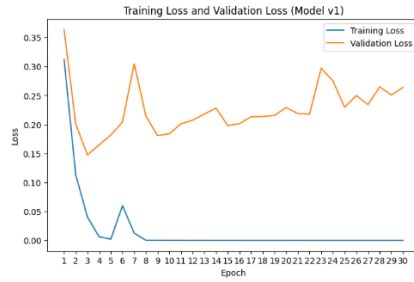
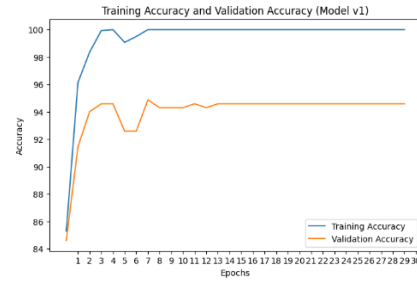
For the multi-class classification task, the Cross-Entropy Loss function is chosen due to its suitability for measuring the difference between predicted and actual class probabilities. The Adam optimiser is selected to update the model parameters during training. The choice of a learning rate of $1e-4$ strikes a balance between convergence speed and stability, ensuring effective learning while mitigating the risk of rapid fluctuations in parameter updates. To reduce overfitting and improve generalisation, a weight decay value of $1e-6$ is chosen to impose a mild regularisation effect on the model.

2.6 Training Process

The model training process conducted on a T4 GPU provided by Google Colab encompasses two variants: Model 1, trained on the original dataset, and Model 2, trained on an augmented dataset. Both variants were trained for 30 epochs, wherein each epoch involved iterating through batches of the training dataset and updating model parameters based on the computed loss. The decision to train for 30 epochs represents a strategic balance between ensuring convergence to an optimal solution and mitigating the risk of overfitting.

Table 2. Training metrics for Model 1

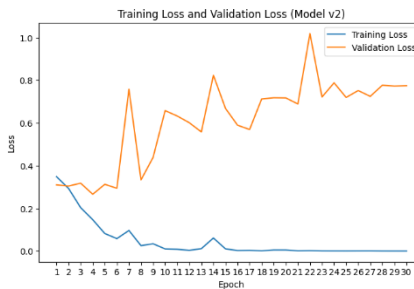
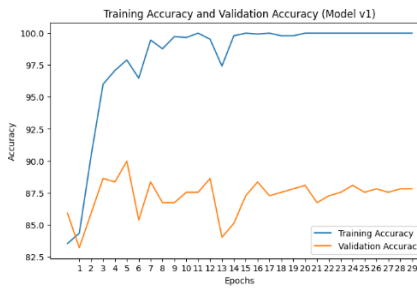
	Loss	Accuracy (%)
Training	0.0183	99.2767
Validation	0.2258	93.9696

**Fig. 3.** Training and validation loss graph for Model 1**Fig. 4.** Training and validation accuracy graph for Model 1

The training results for Model 1 are depicted in the table and figures above. Model 1 achieved an accuracy of 93.9696% on the validation set, as depicted in Table 2. Fig. 3 shows a decrease in training and validation losses, indicating that the model is learning and improving its performance. Fig. 4 highlights potential signs of overfitting, where the model memorises training data patterns that do not generalise well to unseen data.

Table 3. Training metrics for Model 2

	Loss	Accuracy (%)
Training	0.0471	97.9873
Validation	0.6104	87.2087

**Fig. 5.** Training and validation loss graph for Model 2**Fig. 6.** Training and validation accuracy graph for Model 2

As depicted in Table 3, Model 2 achieved an accuracy of 87.2087% on the validation set. Fig. 5 and Fig. 6 suggest signs of overfitting, as indicated by the increase in validation loss and the difference between training and validation accuracy.

3 Experimental Result

3.1 Model 1 (w/o Augmentation)

Table 4 below shows the precision, recall, and F1-score metrics of Model 1 for each batch of data, along with their averages.

Table 4. Output metrics for Model 1

Batch	Precision (%)	Recall (%)	F1-Score (%)
1	94.8980	92.5373	93.7028
2	82.5221	82.8889	82.7051
3	84.1986	61.6529	71.1832
Average	87.2062	79.0264	82.5304

In view of precision, recall and F1-Score, the model performed best in Batch 1, but showed significant drops in subsequent batches due to a higher number of false positive predictions. This indicates that the model is incorrectly identifying instances as positive when they are negative.

Table 5 below presents the test results obtained from evaluating Model 1 on three different batches of data. Each batch consists of both good and bad seed images, with their respective loss and accuracy metrics.

Table 5. Test results for Model 1

Batch	Loss	Accuracy (%)		
		Good Seed	Bad Seed	Average
1	0.2488	92.5373	95.0000	93.7687
2	0.9543	82.8889	82.4444	82.6667
3	1.7648	61.6529	88.1956	74.9243
Average	0.9893	72.0264	88.5467	83.7866

The model performed best in Batch 1, with the highest accuracy of 93.77%, but showed significant drops in accuracy and increases in loss in subsequent batches which aligns with the observation in Table 4. Furthermore, the model also showed higher accuracy in classifying bad seeds compared to good seeds. This suggests that the model is better at distinguishing features in bad seeds.

To visualise the features identified by the model, the saliency map is employed, offering insights into the regions of the image that contributed most significantly to the model's decision-making process. Table 6 below displays a compilation of saliency maps for both correctly and incorrectly classified images.

Table 6. List of saliency maps for Model 1


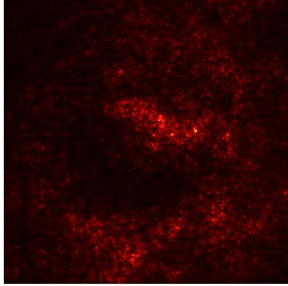

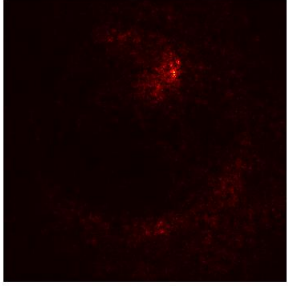
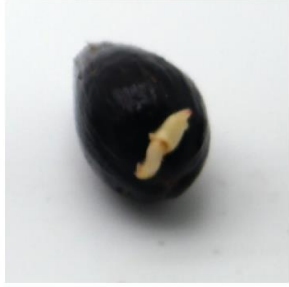
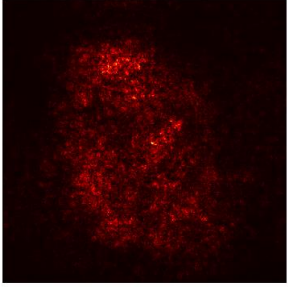
	Saliency Map	
Correctly classified images		
		
Wrongly classified images		

Fig. 7. Saliency map of a correctly classified good seed**Fig. 8.** Saliency map of a correctly classified bad seed**Fig. 9.** Saliency map of a wrongly classified good seed



Fig. 10. Saliency map of a wrongly classified bad seed

(Refer to Appendix C for the compilation of misclassified images)

Upon analysing some examples of saliency map, it becomes evident that accurately classified seeds exhibit highlights primarily on their radicles as depicted in Fig. 7 and Fig. 8. This observation aligns with the expected behaviour, as the radicle is a key characteristic for distinguishing between good and bad seeds. Conversely, in wrongly classified images, the highlights encompass the entire seed, complicating the identification of crucial parts for accurate seed classification as shown in Fig. 9 and Fig. 10. The lack of clear highlights on the radicles in misclassified images indicates that the model may struggle with identifying the distinguishing characteristics of the seeds.

3.2 Model 2 (w/ Augmentation)

Table 7 below shows the precision, recall, and F1-score metrics of Model 2 which was trained with augmented dataset on Batch 1.

Table 7. Output metrics for Model 2

Batch	Precision (%)	Recall (%)	F1-Score (%)
1	86.9681	95.3353	90.9597
2	50.0000	100.0000	66.6667
3	50.5008	100.0000	67.1104
Average	62.4896	98.4451	74.9123

The output metrics for Batch 1 indicate robust performance in distinguishing between good seed and bad seed images, with high precision, recall, and F1-score values. However, for Batch 2 and 3, the precision for identifying bad seed drops significantly to 50%, while the recall and F1-score for bad seed classification increases to 100%.

Table 8 below presents the test results obtained from evaluating Model 2 on augmented data, consists of both good and bad seed images, with their respective loss and accuracy metrics.

Table 8. Test results for Model 2

Batch	Loss	Accuracy (%)		
		Good Seed	Bad Seed	Average
1	1.2164	95.3353	58.4746	76.9050
2	11.8705	100.0000	0.0000	50.0000
3	10.6611	100.0000	0.0000	50.0000
Average	7.9160	98.4451	19.4915	58.9683

Regarding the loss and accuracy metrics, Batch 1 exhibits a relatively low loss and high accuracy, indicating effective model training and good performance in distinguishing between good seed and bad seed images. However, for Batch 2 and Batch 3, the loss increases significantly, while the accuracy drops to 0% for bad seed classification, with all images being classified as good seed. This anomaly suggests a failure of the model to generalise well to unseen data in Batch 2 and Batch 3, leading to inaccurate and biased classifications.

Table 9 below displays a compilation of saliency maps for both correctly and incorrectly classified images.

Table 9. List of saliency maps for Model 2

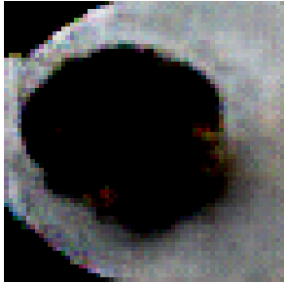
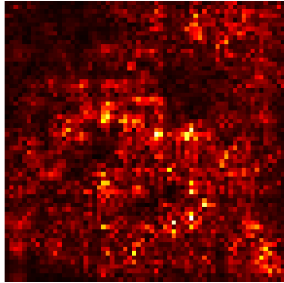

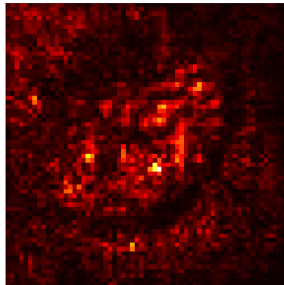
	Saliency Map	
Correctly classified images		
		

Fig. 11. Saliency map of a correctly classified good seed**Fig. 12.** Saliency map of a correctly classified bad seed

Wrongly classified
images

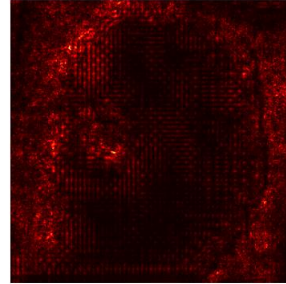
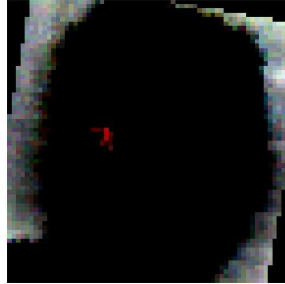


Fig. 13. Saliency map of a wrongly classified good seed

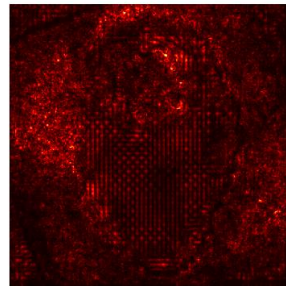
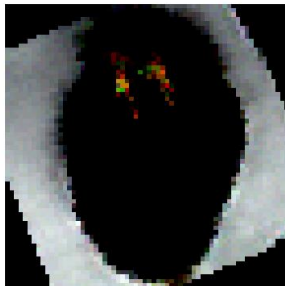


Fig. 14. Saliency map of a wrongly classified bad seed

(Refer to Appendix C for the compilation of misclassified images)

After analysing the saliency maps for Model 2, it was evident that all failed to distinguish the radicles of the seed. The saliency maps of correctly classified images appeared brighter in colour compared to the misclassified ones and effectively highlighted the shape of the seed, as depicted in Fig. 11 and Fig. 12. Conversely, the misclassified images in Fig. 13 and Fig. 14 only emphasised the rotated image boundary rather than the seed itself, potentially impacting classification accuracy.

4 Discussion

4.1 Evaluation for Model 1

Strength: Model 1 consistently demonstrated high performance, as evidenced by the high precision, recall, and F1-score metrics across all batches. It also showed better accuracy in classifying bad seeds compared to good seeds, suggesting a robust ability to identify and differentiate between different seed types. This could be because bad

seeds exhibit greater variability in features like distinct radicle length and colour, making them easier for the model to differentiate compared to good seeds.

Weakness: Despite its strengths, Model 1 displayed signs of overfitting, as evidenced by the decreasing performance across successive batches. This decline, observed in precision, recall, and F1-score metrics, coupled with an increase in loss and a reduction in accuracy, suggests that the model may have become overly specialised to the training dataset. The sudden decrease in accuracy for good seeds to 61% in Batch 3 further proves this overfitting phenomenon. This could be attributed to the dataset's increasing complexity or the model's tendency to memorise specific patterns from the training data. Consequently, when faced with new instances during testing, the model struggled to generalise effectively, resulting in diminished performance over successive batches.

4.2 Evaluation for Model 2

Strength: Model 2 exhibits strong performance on Batch 1, successfully differentiating between good seed and bad seed images with high precision, recall, and F1-score metrics. This proficiency is attributed to the utilisation of DCGAN to augment Batch 1 images, enabling the model to effectively identify and classify the training images.

Weakness: Despite its success on Batch 1, Model 2 faces challenges, particularly in Batch 2 and Batch 3, where it experiences significant overfitting. This overfitting is due to the generated images from DCGAN closely resembling those in Batch 1, leading to a lack of diversity in the augmented dataset. The model struggles to generalise to unseen data in Batch 2 and Batch 3, misclassifying all images as good seed which results in a 100% accuracy for good seed but a complete failure to identify any bad seed.

4.3 Comparative Analysis of Both Models

Table 10 below shows comparison of output metrics and accuracy for Model 1 and Model 2.

Table 10. Comparison of Output Metrics and Accuracy of Model 1 and Model 2

Model	Model 1			Model 2		
Batch	1	2	3	1	2	3
Precision (%)	94.8980	82.5221	84.1986	86.9681	50.0000	50.5008
Recall (%)	92.5373	82.8889	61.6529	95.3353	100.0000	100.0000
F1-Score (%)	93.7028	82.7051	71.1832	90.9597	66.6667	67.1104
Accuracy (%)	93.7687	82.6667	74.9243	76.9050	50.0000	50.0000

In terms of **generalisability**, both models show a decline in output metrics and accuracy as the batches progress, indicating potential overfitting due to variations in lighting

conditions across different batches. However, Model 2 demonstrates a more pronounced decrease in performance compared to Model 1, suggesting a higher susceptibility to overfitting.

In view of **robustness and stability**, Model 1 consistently maintains relatively high output metrics across all batches, indicating its reliability in distinguishing between good and bad seed images. In contrast, Model 2 initially performs well in Batch 1 but experiences a significant drop in performance in subsequent batches, particularly in classifying bad seed, where its accuracy drops to 0%. This inconsistency highlights a lack of robustness in Model 2 compared to Model 1.

Regarding **overall performance**, Model 1 consistently outperforms Model 2 across all batches, with higher precision, recall, F1-score, and accuracy metrics. Despite the use of DCGAN augmentation in Model 2, it fails to improve performance significantly.

In conclusion, Model 2 does not show improved performance compared to Model 1, despite the use of DCGAN augmentation. This lack of improvement suggests that the augmentation technique may not have effectively addressed the underlying issues affecting Model 2's performance, such as overfitting and instability. Further investigation and refinement may be necessary to enhance Model 2's performance in future iterations.

4.4 Addressing Current Challenges

The main problem encountered by both models is overfitting and low generalisation capabilities in accurately classifying seeds under varying lighting conditions. While the project's objective aimed to mitigate these issues by employing augmentation methods through a generative model, the developed DCGAN model fell short in creating a diverse range of images. Instead, it primarily augmented the original dataset using basic techniques such as rotation, flipping, cropping, adjusting brightness, hue etc. Thus, the limited diversity in the augmented dataset poses a challenge in training models capable of handling various scenarios, potentially impacting the overall performance and reliability of the classification model.

4.5 Proposing the Solutions

Several solutions are proposed to tackle the challenges outlined in previous section.

DCGAN model:

To enhance the performance of the DCGAN model, one proposed solution involves expanding the size of the training dataset either by collecting more data or by employing advanced augmentation techniques. Currently, the DCGAN model is trained on a

relatively small dataset comprising only 2153 original seed images. Despite the dataset being augmented eightfold, which improves performance on Batch 1 (*refer to Appendix A*), it still struggles to generalise effectively on Batch 2 and Batch 3. This limited dataset size may hinder the model's capacity to learn and generate a wide array of images.

Additionally, incorporating more sophisticated augmentation techniques can contribute to enhancing the diversity and realism of the generated images. Exploring methods such as style transfer, utilising different 3D cameras with varying focal lengths and distances, or experimenting with alternative GAN architectures can help diversify the generated image dataset [8]. Moreover, introducing additional constraints or regularisation techniques during training process can encourage the model to produce more varied and realistic images, further improving its performance.

Classification model:

To mitigate overfitting in classification model, one proposed strategy is to incorporate regularisation technique using dropout layers within the fully connected layers of the networks. It selectively deactivates neurons during training, thereby preventing overfitting and enhancing the model's ability to generalise to unseen data [9, 10].

Additionally, exploring alternative regularisation techniques such as weight decay or early stopping can further address overfitting and bolster the model's ability to generalise to new data. Weight decay imposes penalties on large weights, discouraging the model from fitting noise in the training data, while early stopping terminates training before the model becomes overly specialised to the training set, thereby promoting better generalisation performance [11].

5 Conclusion

In conclusion, this project aimed to analyse and compare the performance and generalisability of two classification models trained on different datasets. Additionally, it explored the use of a DCGAN model to augment the original Batch 1 dataset with the goal of enhancing the performance of the original Model 1. Despite the efforts to enhance generalisation through augmentation techniques using generative models, Model 1 outperformed Model 2 in terms of accuracy and generalisation capability. The results highlight the importance of the training dataset in influencing the performance of classification models. Further exploration into DCGAN dataset augmentation techniques and model architectures may be necessary to improve the performance of future iterations of the classification model.

References

1. Ng, J. H., Liao, I. Y., Jelani, M. F., Chen, Z. Y., Wong, C. K., Wong, W.C.: Multiview-based method for high-throughput quality classification of germinated oil palm seeds. *Computers and Electronics in Agriculture*, 218, 108684. (2024)
2. Gehlot, M., Saini, M. L.: Analysis of Different CNN Architectures for Tomato Leaf Disease Classification. 2020 5th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE), 1-6., doi: 10.1109/ICRAIE51050.2020.9358279. (2020)
3. Sivakumar, P. V., Sri Ram Mohan, N., Kavya, P., Sai Ravi Teja, P.: Leaf Disease Identification: Enhanced Cotton Leaf Disease Identification Using Deep CNN Models. 2021 IEEE International Conference on Intelligent Systems, Smart and Green Technologies (ICISSGT), 22-26. (2021)
4. Huang B., Liu J., Zhang Q., Liu K., Li K., and Liao X.: Identification and Classification of Aluminum Scrap Grades Based on the Resnet18 Model *Applied Sciences* 12, no. 21: 11133, doi: 10.3390/app122111133. (2022)
5. He K., Zhang X., Ren S., Sun J.: Identity Mappings in Deep Residual Networks, doi: 10.48550/arXiv.1603.05027. (2016)
6. Ruiz, P.: Understanding and visualizing ResNets. Medium; Towards Data Science, <https://towardsdatascience.com/understanding-and-visualizing-resnets-442284831be8> (2018, October 8)
7. Nekamiche, N., Zakaria, C., Bouchareb, S., Smaïli, K.: A Deep Convolution Generative Adversarial Network for the Production of Images of Human Faces. (2022, July)
8. Xu, M., Yoon, S., Fuentes, A., Park, D. S.: A Comprehensive Survey of Image Augmentation Techniques for Deep Learning. *Pattern Recognition*, 137, 109347, doi: 10.1016/j.patcog.2023.109347. (2023)
9. Park, S., & Kwak, N.: Analysis on the Dropout Effect in Convolutional Neural Networks. *Asian Conference on Computer Vision* (pp. 189-204). Springer (2016)
10. Srivastava, N. H., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 13(1), 1929–1958. (2014)
11. Morgado-Dias, F., Mota, A.: Regularization versus early stopping: A case study with a real system. (2003, October)

5.1 Appendix A- Comparison of Performance in DCGAN for Different Dataset Size

Two DCGAN models, v1 and v2, were compared based on their performance using different training dataset sizes. While v1 used original images from Batch 1 dataset, v2 augmented the dataset through various image transformations. The generated images were then used to train Model 2 for classification, with the resulting accuracy recorded. Table 11 presents the comparison between these two DCGAN models.

Table 11. Comparison between DCGAN Model v1 and DCGAN Model v2





	DCGAN Model v1	DCGAN Model v2
Dataset size	GoodSeed: 1102 BadSeed: 1051 Total: 2153	GoodSeed: 8816 BadSeed: 8408 Total: 17224
Examples of generated image		
	Fig. 15. Example of GoodSeed	Fig. 17. Example of GoodSeed
		
	Fig. 16. Example of BadSeed	Fig. 18. Example of BadSeed
Accuracy (%)	GoodSeed: 0.00 BadSeed: 100.00 Overall: 50.00	GoodSeed: 91.84 BadSeed: 57.63 Overall: 83.08

Table 11 shows that v1 has lower accuracy than v2, with v1 experiencing underfitting as Model 2 classifies all images as bad seed, resulting in 50% accuracy only. Additionally, the images generated by v1 are blurrier and less coherent than those generated by v2. Therefore, DCGAN v2 is chosen for generating augmented images in this task.

5.2 Appendix B- Real Images VS Fake Images (DCGAN)

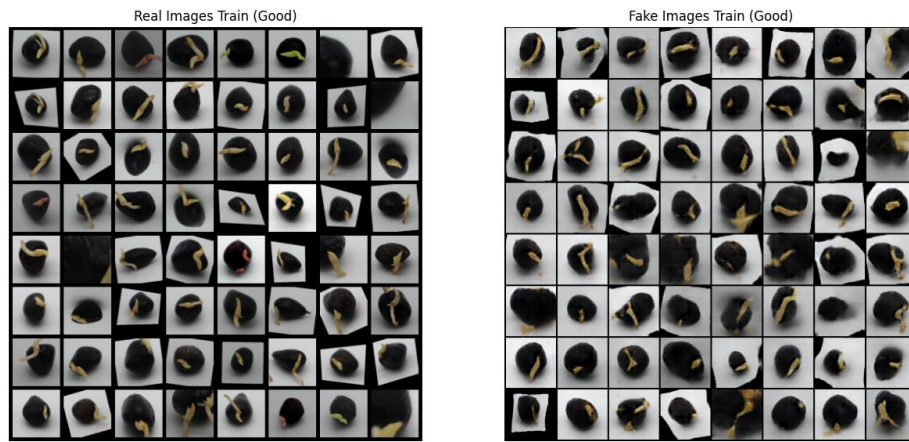


Fig. 19. Real Images VS Fake Images in Training (GoodSeed) dataset folder

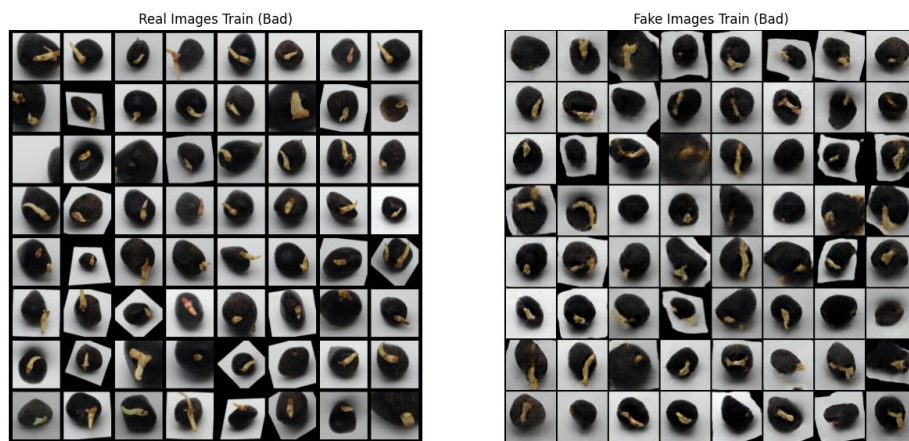


Fig. 20. Real Images VS Fake Images in Training (BadSeed) dataset folder

5.3 Appendix C- Wrongly Classified Images



Fig. 21. Example of wrongly classified images by Model 1

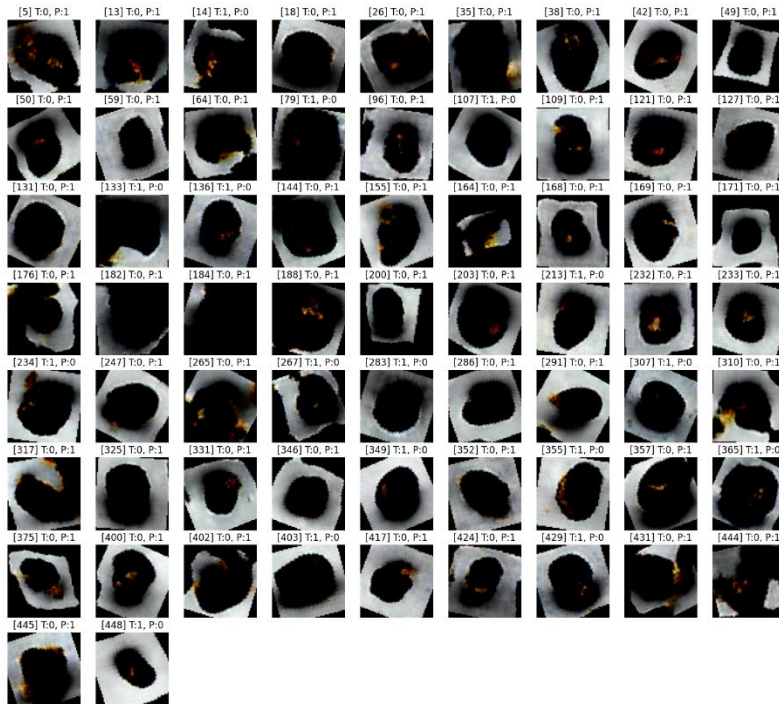


Fig. 22. Example of wrongly classified images by Model 2