

Synthetic Data Quantity in Animal Classification

Fieter Brain Pasaribu^{1✉}, Luh Joni Erawati Dewi¹, I Nyoman Saputra Wahyu Wijaya¹, Ni Wayan Marti¹, Ni Putu Novita Puspa Dewi¹

¹Department of Computer Science, Universitas Pendidikan Ganesha, Singaraja Bali, Indonesia

*Corresponding Author: fieter@undiksha.ac.id

Article Information

Article history:

No. 1137

Rec. April 15, 2026

Rev. September 03, 2026

Acc. September 11, 2026

Pub. September 16, 2026

Page. 30 – 45

Keywords:

- DCGAN
- Image classification
- Synthetic data
- EfficientNet-B0
- MobileNetV2

ABSTRACT

The need for large datasets remains a challenge in deep learning-based classification models such as Convolutional Neural Networks. Limited data can lead to overfitting, where the model performs well on training data but poorly on testing data. To address this, numerous studies have been conducted using both traditional augmented data and synthetic data generated by Generative Adversarial Networks (GANs). GANs have proven effective across a wide range of dataset domains. However, no research has focused on the effect of the amount of synthetic data on model classification accuracy, leading to confusion about how much should be used to avoid overfitting. Therefore, this study will review the effectiveness of GANs by examining the effect of synthetic data ranging from small amounts to 20 times the original dataset size. The synthetic image results from GANs were then tested using them as supplementary datasets for MobileNetV2 and EfficientNet-B0. The results show that the effectiveness of synthetic data fluctuates significantly. At certain amounts, the use of synthetic data can improve accuracy, such as in MobileNetV2, which increased from 73% to 77%. However, at certain amounts, the use of synthetic data also decreases accuracy from 73% as the baseline to 66%. This fluctuating condition is caused by the synthetic data from GANs not being consistent in representing the original data well. The lack of quality of GAN is caused by the architecture used, which is still too simple to capture data patterns with a wide variety of perspectives.

How to Cite:

Pasaribu, F. B., & et al (2026). Synthetic Data Quantity in Animal Classification. Jurnal Teknologi Informasi Dan Pendidikan, 20(1), 30-45. <https://doi.org/10.24036/jtip.v20i1.1137>

This open-access article is distributed under the [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. ©2023 by Jurnal Teknologi Informasi dan Pendidikan.



1. INTRODUCTION

Deep Learning is an advanced branch of Machine Learning that utilizes artificial neural networks with multiple hidden layers to capture complex patterns within data [1]. The progress of deep learning has enabled artificial intelligence to solve tasks that were previously difficult for traditional algorithms, such as speech recognition, natural language processing, and image analysis. One key advancement in this field is the development of specialized architectures that efficiently process specific types of data. In the case of image data, Convolutional Neural Networks (CNNs) are widely used due to their architecture, which is inspired by the human visual system. CNNs consist of convolutional layers, pooling layers, and fully connected layers that work together to extract hierarchical features from images. Convolutional layers are responsible for detecting simple patterns such as edges and textures, while pooling layers reduce the spatial dimensions of the data while preserving essential information [2]. This allows CNNs to learn relevant features automatically, without the need for manual feature engineering as required in traditional machine learning techniques. Because of its strong capability in automatically extracting hierarchical features, Convolutional Neural Networks (CNNs) have been widely applied across various domains. For instance, a study improved CNN accuracy by integrating Discrete Wavelet Transform (DWT) for Balinese carving ornament classification, showcasing the effectiveness of feature extraction techniques in enhancing model performance [3]. Additionally, CNN with the MobileNet architecture has been applied to Songket motif classification, achieving high accuracy and demonstrating its effectiveness in cultural pattern recognition tasks [4]. CNN has also been applied to coffee bean classification, achieving a high accuracy of 98–99% [5]. Furthermore, CNN has been used for classifying beef and pork meat, with comparisons against ResNet50 showing that CNN achieved the highest accuracy of 82.20% [6]. Additionally, CNN architectures have been explored for classifying periapical radiographs, achieving an accuracy of 82%, further highlighting CNN's versatility in medical applications [7]. Still in the medical world, CNN is used for Pneumonia classification in Lung X-ray Images with an average accuracy of 89.58% [8]. CNN with VGG16 and MobileNet architectures is also used for traffic density classification. Where after hyperparameter tuning, the accuracy reaches 97.22% and 96.94% [9].

A significant challenge in using CNNs for classification tasks is the need for large amounts of labeled data to ensure generalization and prevent overfitting. Overfitting occurs when a model performs very well on the training data but fails to make accurate predictions on the testing data. [10] showed that a large dataset is essential in flower classification tasks, and [11] also highlighted that many classification algorithms require extensive datasets, which can be difficult to obtain. In practice, the availability of such data is often limited, particularly in specific domains like medical imaging, agriculture, and industrial

applications. In these cases, data collection must be done manually, which demands significant time, effort, and resources.

To address this problem, the use of Generative Adversarial Networks (GANs) has emerged as a promising approach. GANs are capable of learning the data distribution and generating realistic synthetic images that can be used for data augmentation [12]. However, standard GANs often face training difficulties such as mode collapse and vanishing gradients [13]. To mitigate these issues, Deep Convolutional GANs (DCGANs) were introduced, which incorporate convolutional layers and batch normalization to stabilize training and produce more realistic images [14]. By using DCGAN, it is possible to expand a limited dataset with synthetic images that are very similar to the original data, thereby increasing diversity and improving the training process of CNN models. DCGAN can produce 2.02 times better images with 1.55 times higher diversity compared to traditional GAN models on the MNIST dataset [15]. The use of DCGAN is also the right choice because the data used in this study are four animal classes obtained from the STL-10 dataset used as the basis for the experiment. STL-10 is designed for unsupervised deep learning and is similar to the CIFAR-10 dataset, but with a smaller number of labeled samples per class [16]. With 1040 images per class, it is necessary to avoid too deep architectures such as ProGAN, StyleGAN, and other complex GAN variants to minimize the possibility of overfitting on synthetic data.

This study focuses on evaluating how the quantity of synthetic images generated by a DCGAN affects classification performance. The most closely related work is Bird et al. (2022) [17], who applied a conditional GAN for fruit quality and defect classification and examined between zero and three thousand synthetic images, reporting that roughly four hundred synthetic samples produced the best results. However, that study was confined to a single domain (fruit), a single classifier, and a relatively small synthetic budget, leaving open the question of how classification behaves when synthetic data is scaled far beyond the size of the original dataset. To the best of our knowledge, no prior work has systematically scaled synthetic image quantity up to twenty times the original data on an animal-classification task, nor compared this effect across two different lightweight CNN architectures, nor linked the per-class generative quality (FID) of the GAN to the per-class downstream performance (F1-score) of the classifier. This study addresses that gap by using DCGAN-generated images at ratios ranging from 0.5× up to 20× of the original data, on four animal classes from the STL-10 dataset, evaluated with both EfficientNet-B0 and MobileNetV2. EfficientNet-B0 and MobileNetV2 are widely recognized as efficient and high-performing deep learning models. EfficientNet-B0 was used by [18] to classify two thousand seven hundred images of nine cat breeds and achieved a testing accuracy of 95%. MobileNetV2 has also demonstrated strong performance, as shown by [19], who reported a training accuracy of 98.49% and a validation accuracy of 97.50% in classifying three species of birds of paradise. Both models have proven to be highly effective across various domains, including animal classification, as demonstrated in the previous examples, as well as in

agricultural applications. For instance, a study conducted by [20] and [21] reported that MobileNetV2 achieved an accuracy of 97.37%, while EfficientNet-B0 reached 94.74%, further highlighting their robustness and generalization capability across different tasks. A study conducted by [22] also utilized MobileNet in a similar domain and achieved a testing accuracy of 96.50%, reinforcing the effectiveness of MobileNet-based architectures. The main contributions of this study are as follows:

- 1) We empirically characterize the effect of synthetic-data quantity on animal-image classification across a wide range of ratios (0.5×–20× of the original dataset), substantially larger than the budgets examined in prior work [17].
- 2) We compare this effect on two efficient CNN architectures (EfficientNet-B0 and MobileNetV2), showing that the two models respond differently to the same synthetic data.
- 3) We connect the generative quality of each class (FID/IS) to the per-class classification quality (F1-score), showing that visually consistent classes (deer, horse) benefit reliably while high-variance classes (cat, dog) do not.
- 4) We show that the effect of synthetic data fluctuates and has no universally optimal ratio, implying that synthetic augmentation should be validated experimentally rather than assumed beneficial.

These findings are expected to inform data-augmentation strategies in computer vision, particularly for animal classification under data scarcity, and to motivate further work on higher-fidelity generative models.

2. RESEARCH METHOD

This study consisted of several systematic stages, beginning with dataset preparation and ending with model evaluation and analysis. The STL-10 dataset was used by selecting four classes, namely Cat, Deer, Dog, and Horse, with 1,040 images for each class. The dataset was then divided into training, validation, and testing sets using a ratio of 70%, 10%, and 20%, respectively. Two different preprocessing pipelines were applied according to the model requirements. For DCGAN training, the images were resized to 64×64 pixels, augmented using horizontal flipping and brightness adjustment, and normalized to the range of [-1, 1]. Meanwhile, for the classification models, the images were resized to 224×224 pixels and augmented using several conventional techniques, including flipping, rotation, zooming, shifting, and brightness adjustment, followed by normalization to the range of [0, 1].

The DCGAN model was trained separately for each class using a latent dimension of 100, a batch size of 64, and a maximum of 5,000 epochs. Model checkpoints were selected based on the Fréchet Inception Distance (FID) and Inception Score (IS) to ensure the quality of the generated images. The trained DCGAN models were subsequently used to generate synthetic images with several augmentation ratios, namely 0.5×, 1×, 2×, 3×, 4×, 5×, 10×, 15×,

and 20× relative to the original training data. These synthetic images were combined with the traditionally augmented training images to construct different training scenarios.

As a baseline, the classification models were first trained using the original dataset enhanced only with traditional augmentation methods. EfficientNet-B0 and MobileNetV2 were employed as the classification models. Hyperparameter tuning was performed by evaluating different learning-rate and batch-size combinations, while Adam was used as the optimizer and Cross-Entropy Loss as the loss function. Early stopping with a patience of five epochs and model checkpointing were also applied to reduce overfitting and preserve the best-performing model. The experiments were then repeated using different proportions of synthetic data to investigate how DCGAN-based augmentation affected classification performance. For reproducibility and consistency across experiments, a random seed of 999 was used for the classification models, while a seed of 42 was used for DCGAN training. Finally, classification performance was evaluated on the test set using Accuracy and F1-score, while the quality of the generated synthetic images was assessed using FID and IS. The overall research flow is shown in Figure 1.

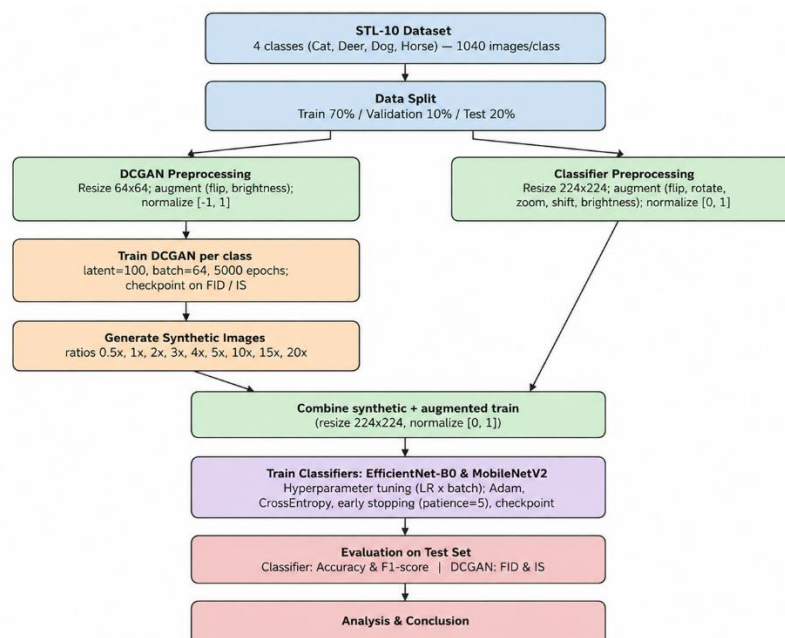


Figure 1. Research Flow

2.1. Dataset and Preprocessing

This study employed a multi-stage data preprocessing methodology to prepare the STL-10 dataset for different experimental phases. Initially, for the baseline classification models, the original data was split into 70% for training, 10% for validation, and 20% for testing. All images were resized to 224x224 pixels, and the training set underwent extensive

augmentation including flipping, rotation, zooming, shifting, and brightness adjustments before all images were normalized to a pixel range of 0 to 1. Subsequently, for training the DCGAN model, a separate pipeline was used where images were resized to 64x64 pixels, underwent simpler augmentation of only flipping and brightness changes, and were normalized to a range of -1 to 1 to match the generator's output layer. Finally, the synthetic images generated by the DCGAN were combined with the training and traditionally augmented training data. This combined dataset was then processed like the initial data, with all images resized to 224x224 and normalized to 0 – 1, while the validation and testing datasets remained unchanged to ensure a fair comparison of model performance. The sample images of the original training data can be seen in Figure 2.

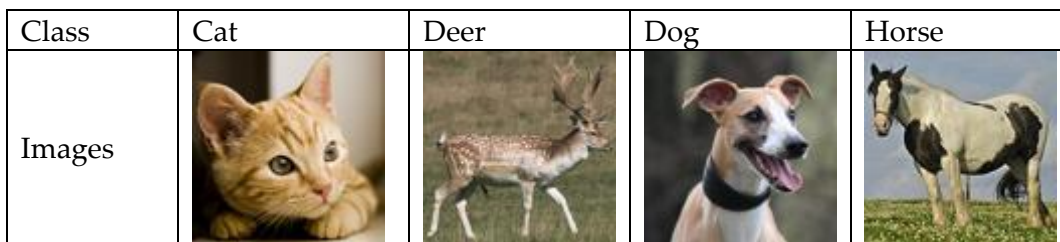


Figure 2. Original Training Data

2.2. Models and Training

MobileNetV2 is a CNN optimized for efficiency using inverted residuals and linear bottlenecks, as seen in Figure 2. It employs depthwise separable convolutions to reduce parameters and speed up inference while maintaining performance, making it ideal for resource-limited devices. The model processes 224x224 input images through an initial convolution layer, followed by inverted residual blocks that expand feature dimensions, apply depthwise convolutions, and restore channel size via a linear bottleneck. This design ensures a balance between accuracy and computational efficiency for classification and object detection tasks [22] [23]. The architecture of MobileNetV2 can be seen in Figure 3.

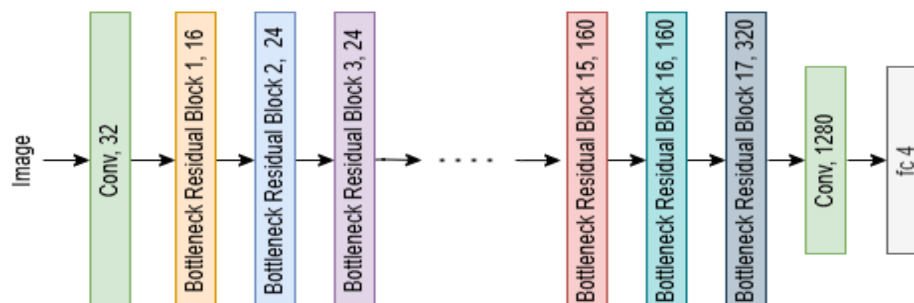


Figure 3. MobileNetV2 Architecture

EfficientNet-B0 is a CNN optimized using the compound scaling technique, which balances model size, input resolution, and network depth for efficient performance [24]. It processes input images through multiple convolutional layers while adapting model size and depth to extract features with fewer parameters than other CNNs, without compromising accuracy. This makes EfficientNet-B0 highly effective for image classification, even with limited data. The architecture of EfficientNet-B0 can be seen in Figure 4.

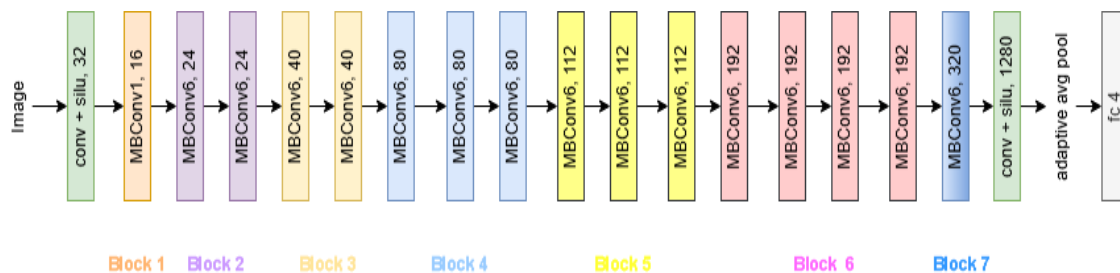


Figure 4. EfficientNet-B0 Architecture

The Deep Convolutional Generative Adversarial Network (DCGAN) is an extension of the standard GAN architecture that incorporates convolutional neural networks (CNNs) into both the generator and discriminator structures. DCGAN is designed to overcome limitations of the original GAN architecture in generating high-resolution images and maintaining training stability. In a DCGAN, the generator uses transposed convolutional layers (deconvolution) to transform a latent vector into a synthetic image, while the discriminator applies convolutional layers to classify whether an image is real or generated. This architecture utilizes batch normalization and activation functions such as ReLU or LeakyReLU to enhance training stability and improve the quality of generated images [14]. The architecture of DCGAN can be seen in Figure 5.

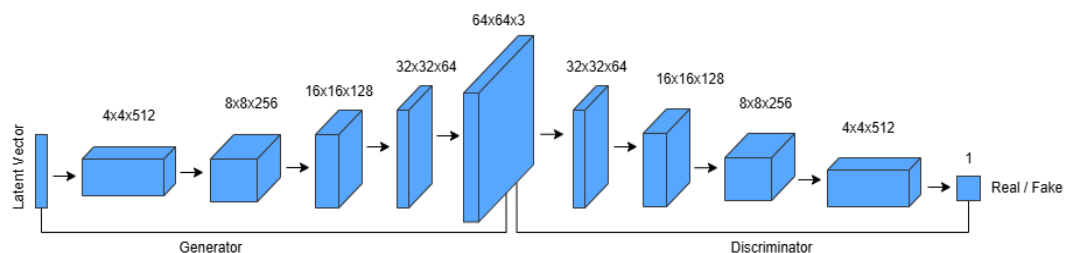


Figure 5. DCGAN Architecture

The DCGAN follows the architecture of Radford et al. (2015). The generator maps a 100-dimensional latent vector to a $3 \times 64 \times 64$ image through five transposed-convolution layers with $512 \rightarrow 256 \rightarrow 128 \rightarrow 64 \rightarrow 3$ feature maps (4×4 kernels; the first layer uses stride 1 and no padding, the remaining layers use stride 2 and padding 1; all without bias). Each hidden

layer is followed by batch normalization and ReLU activation, and the output layer uses a Tanh activation to produce images in the range $[-1, 1]$. The discriminator mirrors this structure with five convolutional layers ($3 \rightarrow 64 \rightarrow 128 \rightarrow 256 \rightarrow 512 \rightarrow 1$, 4×4 kernels, stride 2 except the last layer which uses stride 1 and no padding), applying LeakyReLU with a negative slope of 0.2 and batch normalization in all hidden layers (omitted in the first), followed by a Sigmoid that outputs the probability that an input is real. All weights were initialized from a normal distribution (mean 0, std 0.02 for convolutional layers; mean 1, std 0.02 for batch-normalization layers). Both networks were optimized with Adam (learning rate 0.0002, $\beta_1 = 0.5$, $\beta_2 = 0.999$) using binary cross-entropy loss, with a latent dimension of 100 and a batch size of 64, trained for 5000 epochs. A separate DCGAN with this same architecture was trained for each of the four animal classes, and the best model per class was checkpointed according to its FID and IS scores.

2.3. Model Evaluation

Model evaluation for classifiers is conducted using accuracy as the primary metric. Meanwhile, the evaluation of GAN models involves multiple metrics, including Fréchet Inception Distance (FID) and Inception Score (IS). Accuracy is calculated to assess the impact of synthetic data on model performance, determining whether it enhances or degrades prediction accuracy. On the other hand, FID and IS are used to evaluate the quality and diversity of the generated synthetic images, ensuring that they closely resemble real images and contribute positively to model training [25] [26].

2.4. Assessment on The Testing Data

The model's generalization capability is assessed through a final evaluation on the testing dataset. This dataset is composed of samples that the model has not been exposed to during any of the prior training or validation processes, providing an objective measure of its performance. The performance of the classification models is measured using accuracy. This evaluation is conducted for models trained both with and without the inclusion of synthetic data generated by the DCGAN. The resulting comparison determines the precise contribution of the synthetic data, revealing whether its addition ultimately enhances or degrades the final prediction accuracy of the models.

2.5. Experiment

In this study, hyperparameter tuning of the classification model was conducted to establish a baseline with the best possible accuracy. This involved experimenting with learning rates of 0.01, 0.001, and 0.0001, as well as batch sizes of 32, 64, and 128. Following this, a DCGAN was trained with a latent vector dimension of 100, a batch size of 64, and for 5000 epochs. Upon completion of the training process, synthetic data was generated at

various ratios and combined with the original training dataset. This was done to investigate the impact of synthetic data augmentation on classification accuracy. Subsequently, the classification model was retrained on these new combined datasets. The final step involved testing all dataset scenarios, both with and without synthetic data. The distribution of the dataset can be seen in Table 1.

Table 1. Distribution of combined and DCGAN datasets

Class	Amount of Synthetic Data Number of Train Original Images	Number of Synthetic Images Generated by DCGAN				
		0.5x	1x	...	15x	20x
Cat	910	455	910	...	13,650	18,200
Deer	910	455	910	...	13,650	18,200
Dog	910	455	910	...	13,650	18,200
Horse	910	455	910	...	13,650	18,200

Each model across the various classification scenarios was trained utilizing a specific methodology to ensure optimal performance. To prevent overfitting, periodic evaluations were performed on the validation data throughout the training process. Early stopping with a patience of 5 was applied, meaning that training would halt if no significant improvement in accuracy was observed for five consecutive epochs. Additionally, a checkpointing mechanism was implemented to save the model, ensuring the best-performing version was preserved for further evaluation. The training process utilized the Adam optimizer, and the CrossEntropyLoss function was applied. The results were subsequently evaluated using accuracy metrics to determine the effectiveness of synthetic data in improving classification performance. On the other hand, the DCGAN was trained for 5000 epochs using a checkpointing mechanism based on increasing FID and IS scores. This approach ensured that any model achieving better performance was immediately saved, guaranteeing the generation of the highest quality images.

3. RESULTS AND DISCUSSION

3.1. Classification Results Using Original Data Combined with Traditional Methods

This stage involves hyperparameter tuning of batch size and learning rate. Experiments were conducted using batch size of 32, 64, and 128 also learning rate 0.01, 0.001, and 0.0001 for both MobileNetV2 and EfficientNet-B0. The detail configuration is presented in Table 2.

Table 2. Performance of classification models

Model	Batch Size	Learning Rate	Training Accuracy (%)	Validation Accuracy (%)	Testing Accuracy (%)
EfficientNet-B0	32	0.01	82	73	72
MobileNetV2	32	0.01	84	74	73
EfficientNet-B0	32	0.001	81	76	76
MobileNetV2	32	0.001	83	75	73

Model	Batch Size	Learning Rate	Training Accuracy (%)	Validation Accuracy (%)	Testing Accuracy (%)
EfficientNet-B0	32	0.0001	82	69	68
MobileNetV2	32	0.0001	72	64	60
EfficientNet-B0	64	0.01	86	73	72
MobileNetV2	64	0.01	79	72	73
EfficientNet-B0	64	0.001	84	73	74
MobileNetV2	64	0.001	72	68	66
EfficientNet-B0	64	0.0001	62	62	61
MobileNetV2	64	0.0001	68	61	61
EfficientNet-B0	128	0.01	88	74	75
MobileNetV2	128	0.01	79	72	69
EfficientNet-B0	128	0.001	93	76	77
MobileNetV2	128	0.001	71	63	62
EfficientNet-B0	128	0.0001	58	60	56
MobileNetV2	128	0.0001	62	55	55

Among all the experiments conducted in Table 2, EfficientNet-B0 with a learning rate of 0.001 and a batch size of 128 achieved the highest test accuracy at 77%. On the other hand, MobileNetV2 performed best with a learning rate of 0.001 and a batch size of 32. Overall, EfficientNet-B0 outperformed MobileNetV2 in terms of classification accuracy. MobileNetV2 shows that it tends to have difficulty recognizing patterns from animal data especially when batch sizes are large. In contrast, EfficientNet-B0 offers a strong balance between performance and number of parameters. With its combined scaling design, the model effectively captures important image features without requiring excessive computation.

3.2. DCGAN Model Training Results for Each Class

At this stage, the DCGAN is trained separately for each class based on the configuration and training approach described earlier. The detailed results and examples of the synthetic data are presented in Table 3 and Figure 6.

Table 3. Performance of DCGAN

Model	FID	IS
Cat	181	2.77
Deer	165	3.31
Dog	204	3.76
Horse	173	4.12

Class	Images	
Cat		
Deer		
Dog		
Horse		

Figure 6. Images Generated by DCGAN

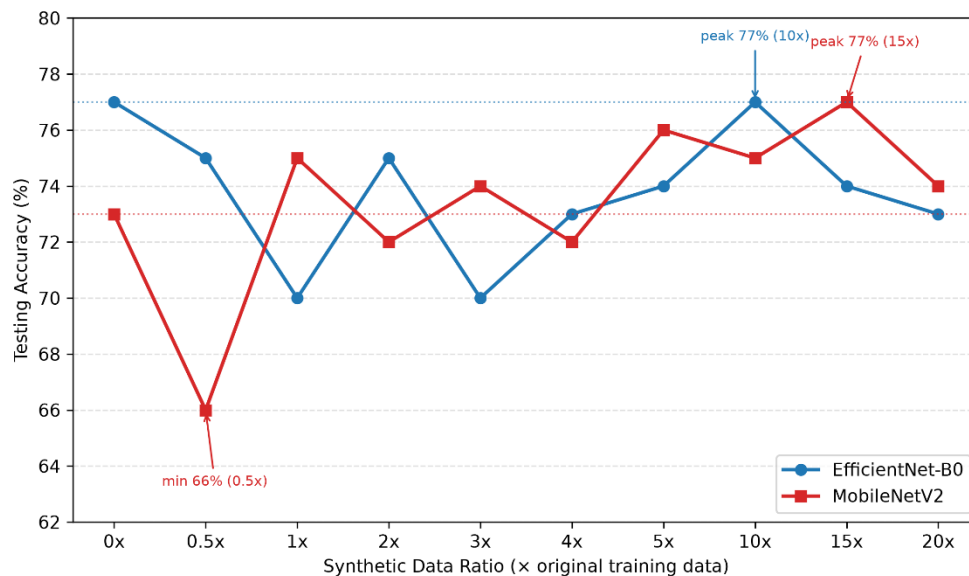
Overall, FID scores were relatively high, ranging from 165 to 204, indicating a general difficulty in generating realistic images. The primary cause of this was identified as the nature of the training data. The dataset was not only limited in the number of images but also exacerbated by a high degree of visual variance and a wide variety of different camera angles, particularly in the cat and dog classes. This combination of a sparse dataset and inconsistent perspectives meant the DCGAN model failed to learn a coherent and accurate data distribution. For example, the cat model generated abstract images with an FID score of 181. In contrast, the deer (FID 165) and horse (FID 173) models produced more recognizable shapes, while the dog class presented a unique case with high diversity (IS 3.76) but the lowest realism (FID 204). These findings collectively confirm that limited and visually inconsistent training data are the primary factors hindering DCGAN performance.

3.3. Experiment with Synthetic Data Ratios on Classification Models

At this stage, the classification models were retrained using synthetic data generated by DCGAN. Each dataset scenario involved different ratios of synthetic data, combined with the same traditionally augmented training data that was previously used to train the baseline models. The goal was to observe how different amounts of synthetic data affect classification accuracy. Various ratios of synthetic data were tested, including 0.5x, 1x, 2x, 3x, 4x, 5x, 10x, 15x, and 20x. The training process utilized Callback and Early Stopping to optimize model performance, while testing data was used as the benchmark for evaluating accuracy. The detail result can be seen in Table 4 and Figure 7.

Table 4. The performance of classification model on various synthetic data ratios

Model	Amount of Synthetic Data	Training Accuracy (%)	Validation Accuracy (%)	Testing Accuracy (%)
EfficientNet-B0	0x	93	76	77
MobileNetV2		83	75	73
EfficientNet-B0	0.5x	86	78	75
MobileNetV2		64	66	66
EfficientNet-B0	1x	75	71	70
MobileNetV2		87	75	75
EfficientNet-B0	2x	88	73	75
MobileNetV2		82	73	72
EfficientNet-B0	3x	82	69	70
MobileNetV2		89	73	74
EfficientNet-B0	4x	93	75	73
MobileNetV2		88	72	72
EfficientNet-B0	5x	91	74	74
MobileNetV2		94	77	76
EfficientNet-B0	10x	96	76	77
MobileNetV2		95	74	75
EfficientNet-B0	15x	94	79	74
MobileNetV2		99	78	77
EfficientNet-B0	20x	95	76	73
MobileNetV2		96	74	74

**Figure 7.** The Performance of classification model on various synthetic data ratios

Among all the experiments conducted in Table 4, the results show that adding synthetic data did not consistently improve model accuracy. For EfficientNet-B0, the best testing accuracy of 77% was achieved without synthetic data and at a 10x ratio, meaning no improvement in accuracy. In fact, at some ratios, such as 1x and 3x, the accuracy dropped

to 70%. In the case of MobileNetV2, the initial accuracy was 73% and increased to 77% at a 15x ratio, indicating a positive effect of synthetic data on MobileNetV2's performance. However, there were instances where MobileNetV2's performance decreased, with an accuracy of only 66% at a 0.5x ratio. Furthermore, in the F1-Score metric, which represents the harmony between recall and precision, there is an interesting relationship between the FID score and the F1-Score that can be further analyzed. The F1-Score results for each scenario can be seen in Table 5.

Table 5. F1-score performance on various synthetic data ratios

Amount of Synthetic Data	EfficientNet-B0				MobileNetV2			
	Cat	Deer	Dog	Horse	Cat	Deer	Dog	Horse
0x	0.75	0.81	0.67	0.82	0.71	0.80	0.63	0.79
0.5x	0.75	0.79	0.67	0.78	0.62	0.68	0.58	0.74
1x	0.66	0.77	0.61	0.76	0.71	0.77	0.69	0.82
2x	0.77	0.80	0.65	0.77	0.69	0.78	0.61	0.80
3x	0.64	0.77	0.58	0.78	0.70	0.76	0.67	0.80
4x	0.75	0.79	0.61	0.78	0.70	0.76	0.59	0.79
5x	0.73	0.78	0.65	0.78	0.74	0.83	0.66	0.79
10x	0.76	0.83	0.68	0.82	0.73	0.78	0.66	0.81
15x	0.73	0.77	0.65	0.78	0.78	0.81	0.68	0.81
20x	0.71	0.78	0.66	0.78	0.72	0.80	0.60	0.81

The experimental results in Table 5 clearly show that the F1-Score, which is the harmony between precision and recall, always has a better value for the deer and horse classes in all synthetic data ratios and on MobileNetV2 and EfficientNet-B0. This is in line with the FID values of the deer and horse classes, which have the best values among the other classes due to visual consistency in the training data.

The two architectures responded to synthetic augmentation in markedly different ways, which can be explained by their design and capacity. EfficientNet-B0 employs compound scaling that jointly balances network depth, width, and input resolution, giving it a comparatively strong feature extractor that already reached a high baseline accuracy (77%) using only the original data. Because the model was already close to its data-efficiency ceiling for this four-class task, the addition of low-fidelity synthetic samples whose realism was limited, as reflected by the high FID scores of 165–204, introduced redundant or noisy modes rather than genuinely new discriminative information. As a result, EfficientNet-B0 gained little from augmentation and even degraded at several ratios (down to 70% at 1x and 3x). MobileNetV2, in contrast, is a lighter architecture based on depthwise separable convolutions and inverted residuals; it started from a lower baseline (73%) and therefore had more headroom to benefit from additional training variation. Moderate-to-large amounts of synthetic data acted as a form of regularization that improved its generalization, raising accuracy to 77% at the 15x ratio. However, its lower capacity also made it more sensitive to distribution shifts introduced by poor-quality batches, which explains the sharp drop to 66% at the 0.5x ratio.

This interpretation is further supported by the class-level results. For both models, the deer and horse classes which had the lowest FID scores and the most visually consistent training images consistently achieved the highest F1-scores across all ratios, whereas the cat and dog classes characterized by high visual variance and diverse camera angles (FID up to 204), remained difficult regardless of the amount of synthetic data added. In other words, synthetic augmentation helped primarily where the DCGAN had already modeled the class distribution well, and added noise where it had not. This dependence on per-class generative quality, combined with the differing capacities of the two classifiers, accounts for the fluctuating results and the absence of a single optimal synthetic-to-real ratio.

These results suggest that there is no universal rule for the ideal amount of synthetic data, as its impact varies depending on the ratio and quality of the generated data. Poor-quality synthetic data can introduce noise or bias, hindering the model's ability to learn effectively. Conversely, high-quality synthetic data can help the model generalize better by exposing it to new variations. This inconsistency in performance also explains why the findings of this study may differ from other studies, as the effectiveness of synthetic data depends heavily on how well it matches the characteristics of the original dataset. This is also supported by the improved FID scores for the deer and horse classes, which directly impacted all synthetic data ratios, indicating a better F1-Score.

4. CONCLUSION

This study aimed to evaluate the impact of synthetic data generated by DCGAN on the classification accuracy of animal images using EfficientNet-B0 and MobileNetV2. The highest accuracy is 77% for both EfficientNet-B0 and MobileNetV2. Although DCGAN successfully expanded the training dataset, the generated images still lacked realism and diversity, as indicated by high FID and moderate IS scores. Adding synthetic data did not always improve accuracy. In some cases, performance worsened due to low-quality images or imbalanced data ratios. However, what is certain is that a good FID value will have a direct impact on the F1-Score of that class. As in the example of the deer and horse classes. In addition, there was no universally optimal ratio between synthetic and original data, as the most effective proportion depended on the dataset, model, and image quality. Therefore, the inclusion of synthetic data should be based on careful experimental validation. While DCGAN can help address data scarcity, it must be used cautiously, as poor-quality data or improper ratios may reduce model generalization. Future research is encouraged to use larger and more diverse datasets and to explore advanced GAN models such as WGAN-GP, StyleGAN, ProGAN, or BigGAN for better image quality and training stability.

REFERENCES

- [1] S. Dong, P. Wang, and K. Abbas, "A survey on deep learning and its applications," *Comput. Sci. Rev.*, vol. 40, p. 100379, 2021, doi: 10.1016/j.cosrev.2021.100379.

- [2] M. Saleem, N. Senan, F. Wahid, M. Aamir, A. Samad, and M. Khan, "Comparative Analysis of Recent Architecture of Convolutional Neural Network," *Math. Probl. Eng.*, 2022, doi: 10.1155/2022/7313612.
- [3] N. L. P. Kurniawati, M. W. A. Kesiman, and I. M. G. Sunarya, "Recognition of Balinese Traditional Ornament Carving Images with Convolutional Neural Network and Discrete Wavelet Transform," *J. Ilm. Tek. Elektro Komput. Dan Inf.*, vol. 8, no. 4, pp. 670–678, 2023.
- [4] K. Y. Mahendra, L. J. E. Dewi, I. K. Purnamawan, and F. B. Pasaribu, "Songket Motif Classification using MobileNet Architecture for Cultural Heritage Preservation," *Int. J. Informatics Comput.*, vol. 7, no. 2, pp. 725–734, 2025.
- [5] A. T. Hidayat, I. M. G. Sunarya, I. P. G. S. Andisana, N. N. E. Smrti, and A. A. G. W. S. Erlangga, "PERBANDINGAN PERFORMA MODEL CNN PADA KLASIFIKASI BIJI KOPI DENGAN CITRA BERBASIS CONDITIONAL LIGHTWEIGHT GAN," *J. Pendidik. Teknol. dan Kejur.*, vol. 23, no. 1, pp. 105–116, 2026.
- [6] I. M. A. M. Dinata, I. G. A. Gunadi, and I. M. G. Sunarya, "Analisis Hyperparameter Pada Klasifikasi Jenis Daging Menggunakan Algoritma Convolutional Neural Network," *J-SAKTI (Jurnal Sains Komput. dan Inform.)*, vol. 8, no. 1, pp. 25–34, 2024.
- [7] I. G. L. T. Sumantara, M. W. A. Kesiman, and I. M. G. Sunarya, "Comparative Analysis of CNN Methods for Periapical Radiograph Classification," *J. Nas. Pendidik. Tek. Inform. JANAPATI*, vol. 13, no. 2, pp. 204–214, 2024.
- [8] I. D. Maysanjaya and M. Dendi, "Klasifikasi Pneumonia pada Citra X-rays Paru-paru dengan Convolutional neural network," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 9, no. 2, pp. 190–195, 2020.
- [9] I. P. A. Suputra, I. G. A. Gunadi, and I. M. G. Sunarya, "Hyperparameter Optimization with MobileNet Architecture and VGG Architecture for Urban Traffic Density Classification Using Bali Camera Image Data," *Sink. J. dan Penelit. Tek. Inform.*, vol. 9, no. 3, pp. 1132–1145, 2025.
- [10] R. Firmansyah, "Deep learning implementation using convolutional neural network for interest classification," Faculty of Science and Technology UIN Syarif Hidayatullah Jakarta, 2021.
- [11] A. A. Pramadnan and G. E. Saputra, "Cycle generative adversarial networks algorithm with style transfer for image generation," *arXiv Prepr. arXiv2101.03921*, 2021.
- [12] I. Goodfellow *et al.*, "Generative adversarial nets," 2014.
- [13] Z. Ding, S. Jiang, and J. Zhao, "Take a close look at mode collapse and vanishing gradient in GAN," 2022, *IEEE*.
- [14] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," *arXiv Prepr. arXiv1511.06434*, 2015.
- [15] B. Liu, J. Lv, X. Fan, J. Luo, and T. Zou, "Application of an improved DCGAN for image generation," *Mob. Inf. Syst.*, vol. 2022, no. 1, 2022.
- [16] I. Papastratis, "Ablation study of self-supervised learning for image classification," pp. 1–5, 2021, [Online]. Available: <http://arxiv.org/abs/2112.02297>
- [17] J. J. Bird, C. M. Barnes, L. J. Manso, A. Ekárt, and D. R. Faria, "Fruit quality and defect image classification with conditional GAN data augmentation," *Sci. Hortic. (Amsterdam)*, vol. 293, p. 110684, 2022.
- [18] N. Azahro Choirunisa, T. Karlita, and R. Asmara, "Deteksi Ras Kucing Menggunakan Compound Model Scaling Convolutional Neural Network," *Technomedia J.*, vol. 6, no. 2, pp. 236–251, 2021, doi: 10.33050/tmj.v6i2.1704.
- [19] R. M. Thaniket, M. H. J. Rahanra, and W. W. Dharsono, "Klasifikasi Jenis Burung Cendrawasih

- Menggunakan Convolutional Neural Network (CNN) Berdasarkan Citra," *J. Ilm. Sist. Inf. dan Tek. Inform.*, vol. 8, no. 1, pp. 103–113, 2025.
- [20] F. B. Pasaribu, L. J. E. Dewi, K. Y. E. Aryanto, and P. Varnakovida, "Effect of Synthetic Data Augmentation on Plant Classification Accuracy Using MobileNetV2, EfficientNet-B0, and ResNet-18," *2025 IEEE 5th Int. Conf. Softw. Eng. Artif. Intell. SEAI 2025*, pp. 1–6, 2025, doi: 10.1109/SEAI65851.2025.11108883.
- [21] K. R. M. Manikam, L. J. E. Dewi, K. Y. E. Aryanto, K. A. Seputra, and P. Varnakovida, "Analisis Hyperparameter Pada Klasifikasi Jenis Tanaman Menggunakan Algoritma ResNet50 Dan MobileNetV2," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 9, no. 6, pp. 9921–9928, 2025.
- [22] K. A. Tamayasa and L. J. E. Dewi, "ANALISIS PERBANDINGAN MODEL ARSITEKTUR MOBILENETV2 DAN EFFICIENTNETB3 DALAM KLASIFIKASI PENYAKIT DAUN JAGUNG," *J. Inform. dan Tek. Elektro Terap.*, vol. 14, no. 1, 2026.
- [23] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," 2018.
- [24] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," 2019, *PMLR*.
- [25] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," 2017.
- [26] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," 2016.