Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871

Vol. 6, No. 6, December, 2025, Page. 5474-5488
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5336

Integration of Squeeze-and-Excitation in Densenet-121 for Classifying Real and AI-Generated Images

Nadiyya Hasaniyyah[1], Khadijah[*2], Sutikno[3], Zahra Arwananing Tyas[4]

[1,2,3]Informatics, Universitas Diponegoro, Indonesia
[4]Information Technology, University Muhammadiyah Malaysia, Malaysia

Email: [1]khadijah@live.undip.ac.id

Received : Sep 25, 2025; Revised : Nov 6, 2025; Accepted : Nov 14, 2025; Published : Dec 23, 2025

Abstract

Recent advancements in generative technologies, such as Generative Adversarial Networks (GANs) and Latent Diffusion Models, have enabled the creation of AI-generated synthetic images that are increasingly indistinguishable from real ones, posing significant challenges for verifying the authenticity of visual content. This study develops a DenseNet-121 model with hyperparameter optimization and the integration of Squeeze-and-Excitation (SE) attention mechanisms at Early, Mid, and Late positions. Experiments were conducted using the CIFAKE dataset with a resolution of 32×32 pixels to compare the baseline Plain model with three SE variants. Hyperparameter optimization was applied to maximize model performance. The results demonstrate that the Plain DenseNet-121 with optimized hyperparameters achieved an accuracy of 98.52%, outperforming the standard configurations reported in previous studies. The integration of SE yielded varied outcomes, where Mid SE attained the highest accuracy of 98.56%, while Early SE (98.45%) and Late SE (98.48%) exhibited greater stability with lower standard deviations. These findings highlight that combining hyperparameter optimization with appropriate SE placement can enhance model performance for classifying real and AI-generated images. Moreover, SE placement at different positions (Early, Mid, Late) has a significant impact on feature representation and generalization in synthetic image classification, which is increasingly important given the growing difficulty of distinguishing real from AI-generated images.

Keywords : *AI-Generated Image, Attention, Classification, DenseNet-121, Hyperparameter, Squeeze-And-Excitation*

1. INTRODUCTION

Digital images have become an integral part of modern communication and information exchange. They are widely used for news delivery, documentation, and personal expression, as the human brain processes visual information more quickly and effectively than text [1]. With the increasing prevalence of social media platforms such as Instagram and Facebook, images have become the dominant medium for everyday interaction and carry substantial social value [2]. In this context, the authenticity of visual content plays a crucial role in shaping public opinion and supporting informed decision-making, particularly as manipulated or fabricated images can undermine trust in digital information [3]. At the same time, recent advances in generative artificial intelligence have transformed the landscape of digital imagery. Sophisticated models such as Generative Adversarial Networks (GANs) and Latent Diffusion Models (LDMs) are now capable of producing synthetic images with a high degree of realism [4]. The accessibility of these technologies has accelerated the creation and dissemination of synthetic content, effectively blurring the boundary between authentic and fabricated images. This development raises significant concerns, as AI-generated images can be exploited to spread disinformation, commit fraud, violate privacy, and damage reputations [5], [6], [7].

Vol. 6, No. 6, December, 2025, Page. 5474-5488
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5336

To address the growing challenge of synthetic image detection, several benchmark datasets have been developed and adopted in prior studies. For instance, FaceForensics++ [8] and WildDeepfake [9] provide large-scale collections of manipulated facial imagery. StyleGAN2 [10] is a widely used generative model, the authors released separate folders of generated images. More recently, ArtiFact [11] introduced a diverse dataset covering human faces, animals, places, vehicles, artworks, and other real-world categories, with more than 2.4 million images. However, despite its scale and diversity, ArtiFact suffers from an imbalance between real and synthetic samples, which may bias model training and evaluation.

In contrast, CIFAKE [12] provides a balanced dataset of real and synthetic images drawn from several categories. Its balanced composition makes it well suited for controlled experiments, as it mitigates the confounding effects of class imbalance while still representing non-facial synthetic content. The initial study on this dataset employed a custom Convolutional Neural Network (CNN) architecture, testing 36 network topologies and achieving a maximum accuracy of 92.98% [12]. Another study on CIFAKE employed a custom CNN architecture with different configurations, achieving improved performance with an accuracy of up to 96.31% [13].

Research then shifted toward transfer learning with more advanced pretrained CNN architectures due to the limitations of simple custom CNNs. In [14], experiments were conducted on the original CIFAKE dataset resolution of 32×32 pixels, with preprocessing based on normalization using the mean and standard deviation of ImageNet. DenseNet achieved the best performance with an accuracy of 97.74%, significantly outperforming VGGNet (96.00%), ResNet (94.95%), custom CNN (86.40%), and SVM (81.43%).

The performance of transfer learning with pretrained CNN architectures on the CIFAKE dataset has also been examined by other researchers. In [15], DenseNet-121 consistently outperformed ResNet50 and EfficientNetB0, achieving 97.65% accuracy at 32×32 resolution and improving to 98.49% when images were resized to 64×64 pixels. This consistency demonstrates that DenseNet is the most robust architecture among CNN-based models for detecting low-resolution synthetic images. Furthermore, [16] highlighted the importance of fine-tuning in transfer learning. Using pretrained ImageNet weights with frozen base layers resulted in suboptimal performance. After fine-tuning the base layers, all evaluated models (ResNet50, VGG16, EfficientNetV2B0, MobileNetV3Small) achieved substantial accuracy improvements. These findings demonstrate that parameter adjustment of pretrained weights is crucial for CIFAKE.

Research on CIFAKE is not limited to CNN architectures and transfer learning approaches. Attention-based architectures have also been explored, although the reported results vary. The Vision Transformer (ViT-S16 pretrained) achieved only 87.09% accuracy, likely due to information loss when upscaling 32×32 images to 224×224 [13]. Swin Transformer reached a high accuracy of 98.45% in the RGB color space, but its performance dropped drastically to 84% in the CbCrY color space, indicating a strong dependence on color representation [17]. The Multi-Graph Attention Network (MGA-Net), which incorporates a ResNet encoder and a dual-graph decoder, achieved 97.89% accuracy with very lightweight parameters (0.48M), although its AUC score was not the highest and computational cost increased at larger resolutions [18]. These findings suggest that attention integration has the potential to improve accuracy, but the outcomes are highly dependent on architectural design, data representation, and model complexity.

The Squeeze-and-Excitation (SE) module is a channel attention mechanism designed to improve feature representation by adaptively recalibrating the contribution of each channel. SE-based attention has been applied to VGG-Net models for severity assessment of cervical lymph nodes [19], CNNs for deepfake detection [20], MobileNet models for blindness detection [21]. These findings highlight the capability of SE to enhance feature discrimination across diverse domains. On the CIFAKE dataset, a

Vol. 6, No. 6, December, 2025, Page. 5474-5488
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5336

hybrid ResNet-SE attention model has also been proposed [22], achieving an accuracy of 96.12% which is higher than the plain ResNet, demonstrating that SE can improve the detection of AI-generated images. Beyond this specific case, the consistent improvements observed across different tasks suggest that SE serves as a general mechanism to strengthen CNN performance without significant increases in computational cost.

Based on previous studies, pretrained DenseNet has consistently demonstrated superior performance on the CIFAKE dataset, achieving accuracies above 97% [14], [15]. However, no prior study has integrated SE into DenseNet for CIFAKE classification. Building on this gap, this study integrates the Squeeze-and-Excitation (SE) attention mechanism into the DenseNet-121 architecture. DenseNet was selected for its ability to learn deeper features, while the SE block emphasizes salient features, making the combination expected to enhance accuracy and improve the model's focus on synthetic images. Beyond architectural design, several studies have also demonstrated the effectiveness of integrating SE into DenseNet in other domains, such as diabetic retinopathy [23], breast cancer [24], [25], and liver cancer [26].

This study aims to develope DenseNet-121 with SE block integration in order to improve classification accuracy on the CIFAKE dataset. The original resolution of 32×32 pixels is retained to align with the dataset's characteristics, while evaluating the effect of SE placement at early, middle, and late positions. In addition to architectural modifications, hyperparameters such as dropout, learning rate, and batch size are also explored to further optimize model accuracy.

2. METHOD

This research was carried out through a series of interrelated stages. The process began with data initialization and preparation, followed by data splitting and preprocessing to ensure the quality of the dataset. Subsequently, the model was designed and trained, including the integration of the methods examined in this study. The final stage involved testing and performance evaluation to assess the model's ability to generalize. In summary, the overall research workflow is illustrated in Figure 1.

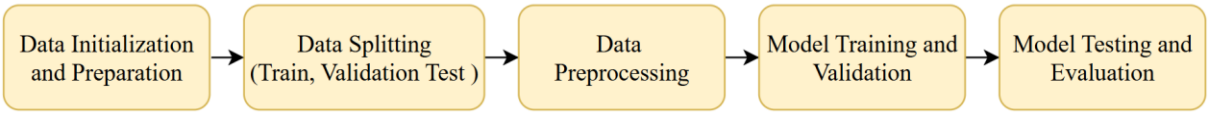

Figure 1. Research Workflow

2.1. Data Initialization and Preparation

The CIFAKE (CIFAR-FAKE) dataset is designed to evaluate the capability of models in distinguishing between real and AI-generated synthetic images. The dataset [12] is hosted on Kaggle, providing open access for research and experimentation. CIFAKE was constructed by combining real images from the CIFAR-10 dataset with synthetic images generated using a latent diffusion model. The dataset comprises 120,000 images, divided into two main classes REAL and FAKE. The REAL class contains 60,000 authentic images sourced from the ten categories of CIFAR-10, such as airplanes, automobiles, and animals. The FAKE class consists of 60,000 synthetic images that mirror these ten categories, generated using Stable Diffusion v1.4 with prompt modifiers to enhance diversity. All images in the CIFAKE dataset are standardized to a resolution of 32×32 pixels. The synthetic images, initially rendered at 512×512 pixels, were resized to 32×32 pixels by the authors to ensure consistency with CIFAR-10. Compared to other deepfake or synthetic datasets, CIFAKE is lightweight, standardized at a low resolution, and thus particularly suitable for testing model performance under resource-constrained scenarios. Figure 2 illustrates representative samples from the CIFAKE training directory for both REAL and FAKE classes.

Vol. 6, No. 6, December, 2025, Page. 5474-5488
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5336

Figure 2. Representative Samples of the CIFAKE Dataset [12]

2.2. Data Splitting (Train, Validation, Test)

Following the partitioning scheme introduced by [12], the CIFAKE dataset was initially divided into 100,000 images for training and 20,000 images for testing. To support model optimization and performance assessment, the training portion in this study was further split into two subsets, resulting in dedicated data for both training and validation. Each subset preserved a balanced ratio of REAL and FAKE classes to ensure fairness in model evaluation. This strategy is consistent with common practices in prior works [13], [15], where separate validation sets are commonly employed to fine-tune hyperparameters and prevent overfitting.

2.3. Data Preprocessing

All images in the CIFAKE dataset were retained at their original resolution of 32×32 pixels to preserve the dataset's characteristics and to evaluate model performance under low-resolution conditions as reported in previous studies [14], [15]. The preprocessing stage consisted of two main steps, augmentation and normalization. Augmentation was applied using Random Horizontal Flip, which has been shown effective for datasets such as CIFAR-10 and CIFAKE that contain objects without orientation dependency, including animals and vehicles [27]. This technique increases data variability while reducing the risk of overfitting [14].

Normalization was applied to standardize the pixel value distribution so that the training process becomes more stable and converges faster [28]. In this study, pixel intensities were first scaled to the range [0, 1] by dividing each pixel value x_i by 255. The scaled values were then normalized using the ImageNet mean and standard deviation, following practices commonly adopted in related studies [8]. This approach ensures that the input distribution aligns with the conditions used during the pretraining of models, resulting in more stable and efficient training.

Equation (1) shows the normalization process where x_{scaled} denotes the pixel value scaled to the range [0, 1], x_{norm} represents the normalized value, x_i is the original pixel intensity, μ is the mean, and σ is the standard deviation [28].

$$x_{scaled} = \frac{x_i}{255}, x_{norm} = \frac{x_{scaled} - \mu}{\sigma} \quad (1)$$

2.4. Model Training and Validation

2.4.1. DenseNet Architecture

DenseNet was introduced by [29] as an advancement of ResNet [30] by implementing dense connectivity across layers. In this architecture, each layer receives the feature maps of all preceding layers through concatenation, enabling effective feature reuse and facilitating gradient flow from the final layers to the initial ones. This approach alleviates the vanishing gradient problem, enhances training efficiency, and reduces the number of parameters compared with other deep convolutional networks of similar depth. DenseNet models designed for the ImageNet dataset, such as DenseNet-121, DenseNet-169, DenseNet-201, and DenseNet-161, adopt the DenseNet-BC architecture, which integrates bottleneck layers and compression ($\theta < 1$) to improve efficiency. The network is composed of four Dense Blocks separated by Transition Layers, followed by a global average pooling layer and a fully connected layer at the end. In the ImageNet experiments, the DenseNet-BC architecture operated on 224×224 input

Vol. 6, No. 6, December, 2025, Page. 5474-5488
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5336

images, with the initial layer implemented as a 7×7 convolution with stride 2. The number of feature maps throughout the network is determined by the certain growth rate. Table 1 summarizes the DenseNet architecture as applied to the ImageNet dataset.

Table 1. The Original Architectures of DenseNet-121 for ImageNet

Layers	Output Size	Operation
Convolution	112 × 112	7 × 7 conv, stride 2
Pooling	56 × 56	3 × 3 max pool, stride 2
Dense Block (1)	56 × 56	[1 × 1 conv, 3 × 3 conv] × 6
Transition (1)	56 × 56	1 × 1 conv
	28 × 28	2 × 2 avg pool, stride 2
Dense Block (2)	28 × 28	[1 × 1 conv, 3 × 3 conv] × 12
Transition (2)	28 × 28	1 × 1 conv
	14 × 14	2 × 2 avg pool, stride 2
Dense Block (3)	14 × 14	[1 × 1 conv, 3 × 3 conv] × 24
Transition (3)	14 × 14	1 × 1 conv
	7 × 7	2 × 2 avg pool, stride 2
Dense Block (4)	7 × 7	[1 × 1 conv, 3 × 3 conv] × 16
Classification	1 × 1	7 × 7 global avg pool + 1000D FC + softmax

2.4.2. Squeeze-and-Excitation Block

Squeeze-and-Excitation (SE) block as a channel attention mechanism designed to enhance feature representations by adaptively recalibrating channel responses [31]. The SE block operates in two stages, namely squeeze and excitation. The squeeze operation aggregates spatial information across each feature map through global average pooling, resulting in a descriptor vector that encodes global channel statistics. The excitation operation projects this descriptor through two fully connected layers with nonlinear activations, producing channel-wise weights that reflect the relative importance of each channel. These weights are then used to rescale the original feature maps, strengthening the contribution of informative channels while suppressing less relevant ones. The integration of SE blocks into convolutional networks has consistently improved performance across a wide range of image recognition tasks with only marginal computational overhead. A schematic overview of the SE block process is shown in Table 2.

Table 2. The Architecture of the SE Block

Layers	Operation	Output Size
Input	-	(H, W, C)
Squeeze	Global Average Pooling	(1, 1, C)
Excitation	Fully Connected Layer 1 with reduction ratio r + ReLU	(1, 1, C/r)
	Fully Connected Layer 2 (return to C dimension) + Sigmoid	(1, 1, C)
Scale	Channel-wise multiplication of the feature map with the weight vector	(H, W, C)

2.4.3. Modified DenseNet with SE

The use of Squeeze-and-Excitation (SE) blocks within DenseNet architectures has been widely explored in previous studies to enhance feature representation. Prior research has shown that embedding SE within DenseBlocks allows the network to recalibrate channel-wise responses, emphasizing informative features while suppressing irrelevant ones, which consistently improves accuracy over

Vol. 6, No. 6, December, 2025, Page. 5474-5488
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5336

standard DenseNet [25]. Similarly, [26] investigated multiple configurations of SE placement in 3D DenseNet architectures, such as SE-DenseNet-In, SE-DenseNet-All, and DenseNet-BC-based variants. Their findings revealed that placing SE within DenseBlocks or after transition layers can improve performance by up to 11% relative to the baseline, highlighting the importance of careful SE placement for optimal generalization.

Building on these insights, this study integrates SE blocks into DenseNet-121 for the CIFAKE dataset, which consists of low-resolution images. To adapt the model, the original 7×7 convolution was replaced with a 3×3 kernel (stride 1, padding 1), and the initial pooling layer was removed, ensuring better preservation of spatial detail. Such adjustments follow established practices for adapting ImageNet-scale architectures to smaller datasets, as also applied in Wide Residual Networks (WRNs) [30]. Consistent with this rationale, the max pooling layer was removed in this study, as pooling is generally unnecessary for low-resolution images, a consideration also emphasized by [32].

Four model variants were developed: a plain DenseNet-121 (baseline) and three SE-augmented versions with early, middle, or late SE placement. Early placement recalibrates channels at the beginning of feature extraction, middle placement applies it at an intermediate stage, and late placement emphasizes relevant features just before classification. All models were initialized with ImageNet-pretrained weights and fully fine-tuned on CIFAKE. Table 3 compares the output shapes of these variants, with dimensions given as (batch size, H, W, C) represents the height (H), width (W), and number of channels (C).

Table 3. Comparison of Output Shapes Across Model Architectures

Layer / Block	DenseNet Plain	DenseNet Early SE	DenseNet Mid SE	DenseNet Late SE
Input	(1, 32, 32, 3)	(1, 32, 32, 3)	(1, 32, 32, 3)	(1, 32, 32, 3)
Conv0	(1, 32, 32, 64)	(1, 32, 32, 64)	(1, 32, 32, 64)	(1, 32, 32, 64)
SE Block	-	(1, 32, 32, 64)	-	-
Norm0 + ReLU0	(1, 32, 32, 64)	(1, 32, 32, 64)	(1, 32, 32, 64)	(1, 32, 32, 64)
Pool0 = Identity	(1, 32, 32, 64)	(1, 32, 32, 64)	(1, 32, 32, 64)	(1, 32, 32, 64)
DenseBlock1(6×)	(1, 32, 32, 256)	(1, 32, 32, 256)	(1, 32, 32, 256)	(1, 32, 32, 256)
Transition1	(1, 16, 16, 128)	(1, 16, 16, 128)	(1, 16, 16, 128)	(1, 16, 16, 128)
DenseBlock2(12×)	(1, 16, 16, 512)	(1, 16, 16, 512)	(1, 16, 16, 512)	(1, 16, 16, 512)
Transition2	(1, 8, 8, 256)	(1, 8, 8, 256)	(1, 8, 8, 256)	(1, 8, 8, 256)
SE Block	-	-	(1, 8, 8, 256)	-
DenseBlock3(24×)	(1, 8, 8, 1024)	(1, 8, 8, 1024)	(1, 8, 8, 1024)	(1, 8, 8, 1024)
SE Block	-	-	(1, 8, 8, 1024)	-
Transition3	(1, 4, 4, 512)	(1, 4, 4, 512)	(1, 4, 4, 512)	(1, 4, 4, 512)
DenseBlock4(16×)	(1, 4, 4, 1024)	(1, 4, 4, 1024)	(1, 4, 4, 1024)	(1, 4, 4, 1024)
SE Block	-	-	-	(1, 4, 4, 1024)
Norm5	(1, 4, 4, 1024)	(1, 4, 4, 1024)	(1, 4, 4, 1024)	(1, 4, 4, 1024)
AvgPool	(1, 1, 1, 1024)	(1, 1, 1, 1024)	(1, 1, 1, 1024)	(1, 1, 1, 1024)
Flatten	(1, 1024)	(1, 1024)	(1, 1024)	(1, 1024)
Dropout	(1, 1024)	(1, 1024)	(1, 1024)	(1, 1024)
Classifier	(1, 1)	(1, 1)	(1, 1)	(1, 1)

The training process was guided by validation accuracy monitoring, with early stopping applied to prevent overfitting [33]. Early stopping was employed as a regularization measure, terminating training when improvements on the validation set plateaued, thereby reducing the risk of memorizing

Vol. 6, No. 6, December, 2025, Page. 5474-5488
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5336

noise in the training data. The stopping criterion was defined in terms of a maximum number of epochs without improvement, commonly referred to as patience, which ensures that the model has sufficient opportunity to continue learning while preventing excessive overfitting.

Hyperparameters are parameters set prior to the training process [34] and include learning rate, batch size, dropout, and other regularization methods. Proper hyperparameter configuration is critical because inappropriate combinations can cause slow or unstable convergence, overfitting, or poor feature representation [35]. Dropout is a regularization technique applied as a hyperparameter to reduce overfitting and improve generalization [35]. The learning rate determines the step size for weight updates during training based on the gradient. A learning rate that is too high can cause instability and failure to converge, whereas a learning rate that is too low slows down the training process[35], [36]. Batch size refers to the number of samples processed in a single iteration before weights are updated. Larger batch sizes generally slow convergence because updates occur less frequently, while smaller batch sizes increase the stochasticity of updates, potentially making training unstable [36], [37].

Optimizing hyperparameters not only affects model performance but also training efficiency and generalization to unseen data. Proper hyperparameter selection can accelerate training, avoid overfitting, improve computational resource utilization, and adapt model architectures to different datasets [38]. In this study, hyperparameters were empirically selected based on preliminary experiments to balance convergence speed, stability, and the ability to generalize, ensuring that the DenseNet-121 models could effectively leverage the representational power of the SE blocks. To maintain efficiency, this study employed Automatic Mixed Precision (AMP), which has been shown to accelerate training and reduce GPU memory consumption without sacrificing model accuracy [39].

2.5. Model Testing and Evaluation

After the model was trained, it was evaluated using previously unseen data. The confusion matrix served as the primary evaluation tool to assess the performance of the classification model [40]. It summarizes prediction outcomes by reporting the number of samples correctly and incorrectly classified for each class. Specifically, True Positives (TP) represent samples that belong to the target class and are correctly predicted, whereas True Negatives (TN) refer to samples that do not belong to the target class and are correctly identified as such. False Positives (FP) correspond to samples incorrectly classified as part of the target class, while False Negatives (FN) denote target class samples that the model failed to recognize. Based on this confusion matrix, several evaluation metrics can be derived, including accuracy, precision, recall, and F1-score. Accuracy, as one of the most widely used metrics, measures the proportion of correctly classified samples and is computed according to Equation (2).

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (2)$$

3. RESULT

3.1. Data Initialization and Preparation

The initial stage of the experiments involved setting up the programming environment and loading the primary dependencies, including PyTorch and torchvision for modeling and image transformations, numpy for numerical computations, and matplotlib for visualization. PyTorch was chosen as the primary deep learning framework in this study due to several technical and practical considerations. It provides high flexibility in designing and modifying deep learning architectures [42], enabling adjustments such as the integration of SE blocks or changes to kernel sizes without being constrained by a predefined model structure. PyTorch also facilitates transfer learning by allowing selective loading of pretrained weights on a layer-by-layer basis, making it particularly suitable for

Vol. 6, No. 6, December, 2025, Page. 5474-5488
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5336

experiments with small input sizes, such as 32×32 pixels.1 The CIFAKE dataset was accessed through the Kaggle platform by adding the dataset as input to the notebook.

3.2. Data Splitting (Train, Validation, Test)

The dataset was divided into three main subsets with balanced class distributions, namely training, validation, and testing. For the training and validation phases, the training subset was further divided into 80,000 images for training and 20,000 images for validation, resulting in a total of 100,000 images used for model development. Meanwhile, the remaining 20,000 images were allocated for testing. All subsets maintained an equal proportion of FAKE and REAL classes. The details of the data splitting are presented in Table 4.

Table 4. Data Splitting

Subset	FAKE	REAL	Total Images
Train	40,000	40,000	80,000
Validation	10,000	10,000	20,000
Test	10,000	10,000	20,000
Total	60,000	60,000	120,000

3.3. Data Preprocessing

All images in the CIFAKE dataset were preserved at their original resolution of 32×32 pixels. Data augmentation was applied using Random Horizontal Flip, which introduces orientation variations while maintaining the semantic content of the images, as illustrated in Figure 3.

Figure 3. Random Horizontal Flip Augmentation

Normalization was performed using the mean and standard deviation values of the ImageNet dataset for each RGB channel (mean = [0.485, 0.456, 0.406]; std = [0.229, 0.224, 0.225]). After normalization, the images appeared visually different, as shown in Figure 4. This transformation does not hinder the model's ability to interpret the images, but instead facilitates more stable and efficient training. Furthermore, leveraging pretrained weights enables the model to start learning from well-established feature representations

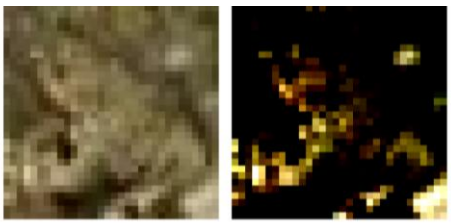

Figure 4. Visualization of Images After Normalization Using ImageNet Channel Statistics

Vol. 6, No. 6, December, 2025, Page. 5474-5488
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5336

3.4. Model Training and Validation

This study evaluated the impact of the Squeeze-and-Excitation (SE) module placement and the combination of hyperparameters on image classification accuracy. DenseNet-121 without any attention mechanism (Plain) was used as the baseline to assess the effects of SE block placement. Three SE-based model variants were implemented at different positions, namely early (Early SE), middle (Mid SE), and late (Late SE), to examine how the location of the attention module influences classification performance and consistency. The hyperparameter variations are presented in Table 5.

Table 5. Hyperparameter Variations

Hyperparameter	Values
Dropout	0.1, 0.3, 0.5
Learning Rate	0.001, 0.0001
Batch Size	128, 256

Each model was trained for a maximum of 20 epochs using an early stopping mechanism with patience set to 4. This configuration was informed by the findings of [43], which demonstrated that a patience range of 3 to 5 in short training cycles of approximately 20 epochs can effectively balance generalization and training efficiency.

The implementation of Automatic Mixed Precision (AMP) significantly improved training efficiency. Using PyTorch's autocast for mixed-precision operations and GradScaler for numerical stability, computations could be performed in lower precision without compromising accuracy. As a result, the average training time per epoch dropped from 7 minutes 33 seconds to 1 minute 39 seconds, achieving a speed-up of over four times.

3.5. Model Testing and Evaluation

To obtain a comprehensive overview of the performance of each architecture, all models were evaluated under various hyperparameter configurations, including dropout, learning rate, and batch size. The evaluation was conducted using accuracy as the metric on the test set, with the Plain model serving as the baseline for assessing the impact of incorporating SE blocks. This systematic comparison allows identification of the best-performing configuration and the stability of each variant across diverse training conditions. Test accuracies for different configurations are presented in Table 6.

Table 6. Test Accuracy for Each Model Architecture

Model	Dropout	Learning Rate	Batch Size	DenseNet Plain	DenseNet Early SE	DenseNet Mid SE	DenseNet Late SE
1	0.1	0.0010	128	98.47	98.21	98.41	98.48
2	0.1	0.0010	256	98.38	98.43	98.48	98.41
3	0.1	0.0001	128	98.15	98.09	98.39	98.08
4	0.1	0.0001	256	97.78	97.77	97.83	97.99
5	0.3	0.0010	128	98.24	98.32	98.39	98.30
6	0.3	0.0010	256	98.52	98.35	98.56	98.48
7	0.3	0.0001	128	98.07	98.12	98.11	98.10
8	0.3	0.0001	256	97.76	97.77	97.88	97.94
9	0.5	0.0010	128	98.35	98.24	98.42	98.42
10	0.5	0.0010	256	98.40	98.45	98.43	98.42
11	0.5	0.0001	128	98.12	98.30	98.31	98.24
12	0.5	0.0001	256	97.20	97.74	98.01	97.94

Vol. 6, No. 6, December, 2025, Page. 5474-5488
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5336

Table 7. Test Accuracy Differences (Δ) Between SE Models and The Plain Baseline

Model	Δ Early SE	Δ Mid SE	Δ Late SE
1	-0.26	-0.06	+0.01
2	+0.05	+0.10	+0.03
3	-0.06	+0.24	-0.07
4	-0.01	+0.05	+0.21
5	+0.08	+0.15	+0.06
6	-0.17	+0.04	-0.04
7	+0.05	+0.04	+0.03
8	+0.01	+0.12	+0.18
9	-0.11	+0.07	+0.07
10	+0.05	+0.03	+0.02
11	+0.18	+0.19	+0.12
12	+0.54	+0.81	+0.74

To assess the effectiveness of each attention mechanism, the Plain model was employed as the baseline for performance comparison. Table 7 presents the accuracy differences (Δ) between the SE models and the Plain baseline, allowing further examination of whether each configuration resulted in performance gains or losses. The hyperparameter configuration was kept consistent across all models (1–12), ensuring that the analysis of accuracy differences between the Plain model and the Early SE, Mid SE, and Late SE variants remained fair and controlled. This approach was intended to identify which attention scenario contributed the most significantly and consistently to improvements in test accuracy.

A detailed examination of Table 7 indicates that most variants with SE blocks produce modest improvements over the Plain mode, with absolute gains generally below one percent in accuracy. In a test set comprising 20,000 samples, even a 0.01 percent increase corresponds to two additional correctly classified images, demonstrating that small numerical differences can have practical significance in large-scale evaluations. Certain configurations exhibit more substantial improvements. For example, Model 12 shows gains of up to 0.81 percent for SE-integrated Mid SE variants to the baseline, equivalent to 162 additional correct predictions. These results highlight that although the numerical differences may appear minor, the cumulative effect over a large dataset can be meaningful and may influence downstream applications where accurate classification is critical.

A further analysis was carried out to assess the consistency of SE block placement on test performance. The summary presented in Table 8 shows the number of configurations in which accuracy increased, decreased, or remained unchanged compared to the Plain baseline. The results indicate that Mid SE was the most stable variant, demonstrating improvements in 11 out of 12 configurations. Late SE was also competitive, showing improvements in 10 out of 12 cases, whereas Early SE experienced decreases in nearly half of the configurations. When compared to the highest accuracy achieved by the Plain model, only Mid SE succeeded in surpassing its performance. These findings highlight that the position of the SE block has a substantial effect on the generalization ability of the model, with placement in the middle or later stages of the network proving more effective in maintaining accuracy.

Table 8. Performance Recap of SE-Based Models Compared to The Plain Baseline

	Increase	Decrease	Unchanged
Early SE	7	5	0
Mid SE	11	1	0
Late SE	10	2	0

Vol. 6, No. 6, December, 2025, Page. 5474-5488
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5336

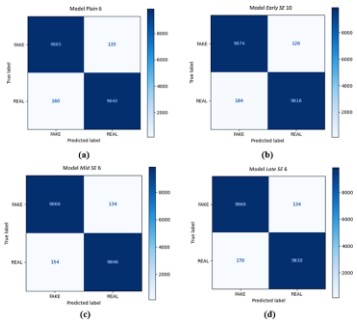

Figure 5. Best-Performing Confusion Matrices for Each Model

Figure 5 presents the best-performing confusion matrices for each model. The Plain model achieved consistently high results, with 135 false positives (FP) and 160 false negatives (FN), reflecting its role as the baseline. The Early SE model reduced FP to 126 but increased FN to 184, indicating an improved ability to detect FAKE samples but a slight deterioration in classifying REAL samples. The Mid SE model achieved the most balanced performance, recording 134 FP and 154 FN, thus demonstrating stability across both classes and emerging as the best overall performer. Meanwhile, the Late SE model produced similar FP (134) to Mid SE but a higher FN (170), suggesting a drop in performance for the REAL class.

From a stability perspective, Mid SE showed the smallest gap between FP and FN, while Early SE exhibited a stronger bias toward minimizing FP at the cost of higher FN. In terms of practical implications, if the application context prioritizes reducing FP (avoiding FAKE instances being misclassified as REAL), Early SE could be preferred. Conversely, if minimizing FN (ensuring REAL instances are not mistaken as FAKE) is more critical, Mid SE would be the most reliable choice.

Across all architectures, the baseline Plain model already showed strong and consistent performance. The integration of SE blocks affected the balance between false positives and false negatives depending on their placement. Early SE reduced false positives but increased false negatives, indicating a trade-off that favors FAKE detection. Mid SE delivered the most stable and balanced outcomes, achieving the lowest overall errors and thus standing out as the best-performing configuration. Late SE achieved comparable results to Mid SE in FAKE detection but fell behind in REAL classification. Overall, Mid SE represents the most reliable architecture, while Early SE may be preferable in contexts where reducing false positives is the primary objective.

The superior performance of the Mid SE configuration can be attributed to its optimal positioning within the network. Placing SE blocks at intermediate layers allows the model to recalibrate feature channels once low-level patterns have been extracted but before high-level abstractions are fully formed. This placement provides the best balance between global and local information, ensuring that discriminative features related to both FAKE and REAL images are emphasized effectively. By contrast, Early SE may suppress important details prematurely, while Late SE risks amplifying overly abstracted features, both of which lead to suboptimal class balance.

4. DISCUSSIONS

The integration of the Squeeze-and-Excitation (SE) attention mechanism into DenseNet-121 improved performance in synthetic image detection on the CIFAKE dataset with a resolution of 32×32 pixels. DenseNet previously achieved 97.65% [15] and 97.74% [14]. In this study, adding SE blocks further improved performance beyond those benchmarks while maintaining the same resolution. This improvement was reinforced by optimized hyperparameters such as dropout, learning rate, and batch size, which, when combined with proper attention placement, enhanced both accuracy and stability.

Vol. 6, No. 6, December, 2025, Page. 5474-5488
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5336

SE blocks strengthened feature representation by emphasizing the most relevant channels, improving DenseNet's ability to distinguish between real and synthetic images. Notably, this outcome was achieved without increasing the resolution to 64×64, as in [15], which reported 98.49% accuracy. The Plain model achieved 98.52%, while the Mid SE configuration further improved performance to 98.56%.

A comparison among SE variants shows that Mid SE was the most consistent, outperforming the baseline DenseNet in most testing scenarios. This finding indicates that the position of attention modules affects model generalization, with mid-level placement enabling more balanced feature utilization than early or late integration. These results enrich the understanding that the effectiveness of attention mechanisms depends not only on the type employed but also on their position within the network architecture.

5. CONCLUSION

The integration of SE blocks into DenseNet-121 effectively improved the accuracy of synthetic image detection at low resolutions, with mid-level placement providing the most consistent results. Combined with optimally tuned hyperparameters, this approach enhanced both performance and model stability. Beyond practical benefits, the study provides new insights into how attention placement interacts with hyperparameter optimization to shape feature representation and generalization in convolutional neural networks, contributing to the broader understanding of attention-based architectures and guiding the design of effective models for synthetic image detection. However, the dataset is limited to 10 classes of objects, animals. Future research should explore larger and more diverse datasets to further evaluate model generalization and applicability to other synthetic image classification tasks.

CONFLICT OF INTEREST

There is no conflict of interest between the authors or with research object in this paper.

REFERENCES

[1] R. Rijitha, "The Impact Of Social Media Marketing On Consumer Purchase Intention," 2021. [Online]. Available: https://www.researchgate.net/publication/358730635

[2] X. Ma and X. Fan, "A review of the studies on social media images from the perspective of information interaction," Apr. 01, 2022, *Elsevier Ltd*. doi: 10.1016/j.dim.2022.100004.

[3] P. Duszejko, T. Walczyna, and Z. Piotrowski, "Detection of Manipulations in Digital Images: A Review of Passive and Active Methods Utilizing Deep Learning," Jan. 01, 2025, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/app15020881.

[4] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," Dec. 2021, [Online]. Available: http://arxiv.org/abs/2112.10752

[5] D. Lamichhane, "Advanced Detection of AI-Generated Images Through Vision Transformers," *IEEE Access*, 2024, doi: 10.1109/ACCESS.2024.3522759.

[6] G. Monkam, W. Xu, and J. Yan, "A GAN-based Approach to Detect AI-Generated Images," 2023, doi: 10.1109/SNPD-Winter57765.2023.10223798.

[7] D. S. Chinta, S. Kamineni, R. P. Chatragadda, and S. Kamepalli, "Analyzing Image Classification on AI-Generated Art Vs Human Created Art Using Deep Learning Models," in *2024 3rd International Conference on Electrical, Electronics, Information and Communication Technologies, ICEEICT 2024*, Institute of Electrical and Electronics Engineers Inc., 2024. doi: 10.1109/ICEEICT61591.2024.10718485.

Vol. 6, No. 6, December, 2025, Page. 5474-5488
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5336

[8] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner, "FaceForensics++: Learning to detect manipulated facial images," in *Proceedings of the IEEE International Conference on Computer Vision*, Institute of Electrical and Electronics Engineers Inc., Oct. 2019, pp. 1–11. doi: 10.1109/ICCV.2019.00009.

[9] B. Zi, M. Chang, J. Chen, X. Ma, and Y. G. Jiang, "WildDeepfake: A Challenging Real-World Dataset for Deepfake Detection," in *MM 2020 - Proceedings of the 28th ACM International Conference on Multimedia*, Association for Computing Machinery, Inc, Oct. 2020, pp. 2382–2390. doi: 10.1145/3394171.3413769.

[10] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, 2020, pp. 8107–8116. doi: 10.1109/CVPR42600.2020.00813.

[11] M. A. Rahman, B. Paul, N. H. Sarker, Z. I. A. Hakim, and S. A. Fattah, "Artifact: A Large-Scale Dataset With Artificial And Factual Images For Generalizable And Robust Synthetic Image Detection," in *Proceedings - International Conference on Image Processing, ICIP*, IEEE Computer Society, 2023, pp. 2200–2204. doi: 10.1109/ICIP49359.2023.10222083.

[12] J. J. Bird and A. Lotfi, "CIFAKE: Image Classification and Explainable Identification of AI-Generated Synthetic Images," *IEEE Access*, vol. 12, pp. 15642–15650, 2024, doi: 10.1109/ACCESS.2024.3356122.

[13] M. Z. Hossain, F. Uz Zaman, and M. R. Islam, "Advancing AI-Generated Image Detection: Enhanced Accuracy through CNN and Vision Transformer Models with Explainable AI Insights," in *2023 26th International Conference on Computer and Information Technology, ICCIT 2023*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/ICCIT60459.2023.10440990.

[14] Y. Wang, Y. Hao, and A. X. Cong, "Harnessing Machine Learning for Discerning AI-Generated Synthetic," 2024. Accessed: Jun. 04, 2025. [Online]. Available: https://arxiv.org/pdf/2406.13688

[15] M. Nayim, V. Mohan, T. N. Pandey, B. B. Dash, B. B. Dash, and S. S. Patra, "Detection of Leading CNN Models for AI Image Accuracy and Efficiency," in *International Conference on Intelligent Algorithms for Computational Intelligence Systems, IACIS 2024*, Institute of Electrical and Electronics Engineers Inc., 2024. doi: 10.1109/IACIS61494.2024.10721936.

[16] A. S. Gupta, K. P. Shreneter, and S. Sehgal, "Visual Veracity: Advancing AI-Generated Image Detection with Convolutional Neural Networks," in *2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions), ICRITO 2024*, Institute of Electrical and Electronics Engineers Inc., 2024. doi: 10.1109/ICRITO61523.2024.10522113.

[17] P. Mehta, A. Sagar, and S. Kumari, "Enhancing Image Authenticity Detection: Swin Transformers and Color Frame Analysis for CGI vs. Real Images," *Procedia Comput Sci*, vol. 258, pp. 2695–2702, 2025, doi: 10.1016/j.procs.2025.04.530.

[18] G. Chen, C. Du, Y. Yu, H. Hu, H. Duan, and H. Zhu, "A Deepfake Image Detection Method Based on a Multi-Graph Attention Network," *Electronics (Switzerland)*, vol. 14, no. 3, Feb. 2025, doi: 10.3390/electronics14030482.

[19] H. Tekchandani, S. Verma, N. D. Londhe, R. R. Jain, and A. Tiwari, "Severity Assessment of Cervical Lymph Nodes using Modified VGG-Net, and Squeeze and Excitation Concept," in *2021 IEEE 11th Annual Computing and Communication Workshop and Conference, CCWC 2021*, Institute of Electrical and Electronics Engineers Inc., Jan. 2021, pp. 709–714. doi: 10.1109/CCWC51732.2021.9375996.

Vol. 6, No. 6, December, 2025, Page. 5474-5488
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5336

[20] S. Dasgupta, J. Mason, X. Yuan, O. Odeyomi, and K. Roy, "Enhancing Deepfake Detection using SE Block Attention with CNN," Jun. 2025, doi: 10.1109/icABCD62167.2024.10645262.

[21] P. Mittal, B. Sharma, D. P. Yadav, and I. Ben Dhaou, "Enhancing Blindness Detection using Deep Learning with MobileNet and SE Blocks," in *2025 International Conference on Control, Automation and Diagnosis, ICCAD 2025*, Institute of Electrical and Electronics Engineers Inc., 2025. doi: 10.1109/ICCAD64771.2025.11099419.

[22] A. R. Gunukula, H. Das Gupta, and V. S. Sheng, "Detecting AI-Generated Images Using a Hybrid ResNet-SE Attention Model," *Applied Sciences (Switzerland)*, vol. 15, no. 13, Jul. 2025, doi: 10.3390/app15137421.

[23] M. S. S. Shazuli and A. Saravanan, "Arithmetic Optimization with SE-DenseNet based Image Retrieval and Classification for Diabetic Retinopathy," in *Proceedings - International Conference on Augmented Intelligence and Sustainable Systems, ICAISS 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 574–580. doi: 10.1109/ICAISS55157.2022.10010803.

[24] G. Ye *et al.*, "Multitask Classification of Breast Cancer Pathological Images Using SE-DenseNet," 2019. doi: 10.1109/ICACI.2019.8778592.

[25] Q. Cai, X. Liu, and Z. Guo, *Identifying Architectural distortion in Mammogram images via a SE-DenseNet model and twice transfer learning*. IEEE, 2018. doi: 10.1109/CISP-BMEI.2018.8633197.

[26] Q. Zhou *et al.*, "Grading of hepatocellular carcinoma using 3D SE-DenseNet in dynamic enhanced MR images," *Comput Biol Med*, vol. 107, pp. 47–57, Apr. 2019, doi: 10.1016/j.compbiomed.2019.01.026.

[27] Z. Tang, Y. Gao, L. Karlinsky, P. Sattigeri, R. Feris, and D. Metaxas, "OnlineAugment: Online Data Augmentation with Less Domain Knowledge," Aug. 2020, [Online]. Available: http://arxiv.org/abs/2007.09271

[28] S.-C. Pei and C.-N. Lin, "Image normalization for pattern recognition," 1995.

[29] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, Institute of Electrical and Electronics Engineers Inc., Nov. 2017, pp. 2261–2269. doi: 10.1109/CVPR.2017.243.

[30] S. Zagoruyko and N. Komodakis, "Wide Residual Networks," 2017. doi: https://doi.org/10.48550/arXiv.1605.07146.

[31] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-Excitation Networks," *IEEE Trans Pattern Anal Mach Intell*, vol. 42, no. 8, pp. 2011–2023, Aug. 2019, doi: 10.1109/TPAMI.2019.2913372.

[32] R. Sunkara and T. Luo, "No More Strided Convolutions or Pooling: A New CNN Building Block for Low-Resolution Images and Small Objects," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Science and Business Media Deutschland GmbH, 2023, pp. 443–459. doi: 10.1007/978-3-031-26409-2_27.

[33] J. Tan, J. Yang, S. Wu, G. Chen, and J. Zhao, "A critical look at the current train/test split in machine learning," Jun. 2021, [Online]. Available: http://arxiv.org/abs/2106.04525

[34] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," Aug. 01, 2018, *Springer Verlag*. doi: 10.1007/s13244-018-0639-9.

Vol. 6, No. 6, December, 2025, Page. 5474-5488
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5336

[35] A. J. Ilemobayo *et al.*, "Hyperparameter Tuning in Machine Learning: A Comprehensive Review," *Journal of Engineering Research and Reports*, vol. 26, no. 6, pp. 388–395, Jun. 2024, doi: 10.9734/jerr/2024/v26i61188.

[36] E. Bartz, T. Bartz-Beielstein, M. Zaefferer, and O. Mersmann, "Hyperparameter Tuning for Machine and Deep Learning with R A Practical Guide," 2023. doi: https://doi.org/10.1007/978-981-19-5170-1.

[37] M. A. K. Raiaan *et al.*, "A systematic review of hyperparameter optimization techniques in Convolutional Neural Networks," Jun. 01, 2024, *Elsevier Inc.* doi: 10.1016/j.dajour.2024.100470.

[38] A. Sharma and D. Kumar, "Hyperparameter Optimization in CNN: A Review," in *Proceedings - 4th IEEE 2023 International Conference on Computing, Communication, and Intelligent Systems, ICCCIS 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 237–242. doi: 10.1109/ICCCIS60361.2023.10425571.

[39] P. Micikevicius *et al.*, "Mixed Precision Training," Feb. 2018, [Online]. Available: http://arxiv.org/abs/1710.03740

[40] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf Process Manag*, vol. 45, no. 4, pp. 427–437, Jul. 2009, doi: 10.1016/j.ipm.2009.03.002.

[41] I. Markoulidakis, I. Rallis, I. Georgoulas, G. Kopsiaftis, A. Doulamis, and N. Doulamis, "Multiclass Confusion Matrix Reduction Method and Its Application on Net Promoter Score Classification Problem," *Technologies (Basel)*, vol. 9, no. 4, Dec. 2021, doi: 10.3390/technologies9040081.

[42] S. Gasmi and T. Abbas, "Leveraging Transfer Learning in PyTorch for Effective Image Classification", doi: 10.13140/RG.2.2.30294.25920.

[43] B. M. Hussein and S. M. Shareef, "An Empirical Study on the Correlation between Early Stopping Patience and Epochs in Deep Learning," *ITM Web of Conferences*, vol. 64, p. 01003, 2024, doi: 10.1051/itmconf/20246401003.