Performance Analysis of Deep Learning Architectures in Classifying Fake and Real Images

Arya Faisal Akbar[1], Putu Desiana Wulaning Ayu[2], Dandy Pramana Hostiadi[3]

[1]*Magister Program, Department of Magister Information System, Institut Teknologi dan Bisnis STIKOM Bali, Indonesia*

[2,3]*Department of Magister Information System, Institut Teknologi dan Bisnis STIKOM Bali, Indonesia*

*corr-author: 222012001@stikom-bali.ac.id

Abstract - The advancements in artificial intelligence (AI) have significantly enhanced image manipulation capabilities, yet they also raise concerns regarding the proliferation of synthetic images. This study investigates the impact of Dynamic Dropout in optimizing deep learning models, including ResNet-101, DenseNet-201, VGG-19, and AlexNet, for classifying real and synthetic images using the CIFAKE and Real and Fake Face datasets. Dynamic Dropout was applied with a progressively increasing rate from 20 percent to 50 percent to enhance training stability and generalization. The results indicate that the optimal configuration consisting of 15 epochs, the Adam optimizer, and Dynamic Dropout consistently outperformed Static Dropout across all models. DenseNet-201 with Dynamic Dropout achieved the highest accuracy of 97.42%, with a precision of 97.33%, recall of 97.58%, and an F1-score of 97.45%. ResNet-101 and VGG-19 exhibited enhanced training stability, while AlexNet proved efficient for lightweight datasets. The Adam optimizer outperformed Nadam, offering greater stability in deeper architectures. Additionally, the 15th epoch was identified as the optimal training duration, balancing accuracy and overfitting mitigation. These findings underscore the importance of selecting optimal training configurations to enhance deep learning performance. Future research should explore adaptive dropout strategies, assess scalability on diverse datasets, and validate these techniques in real-world applications such as digital forensics and AI-generated content detection.

Keywords: Deep Learning architectures, CIFAKE dataset, real and fake face detection dataset, image classification, dynamic dropout

I. INTRODUCTION

The utilization of technology has profoundly impacted various aspects of human life, including work, education, communication, and entertainment. One of the major innovations in technology is Artificial Intelligence (AI), which can automate analysis processes and generate various forms of data, including synthetic images. These images are created through computational processes using graphic design applications, camera filters, or AI-based algorithms that generate visual representations often nonexistent in the real world [1]. The advantages of this technology include increased efficiency and flexibility in the creative sector, but it also raises new challenges, such as the risk of disinformation dissemination, privacy violations, and threats to digital security [2].

In this context, deep learning technology plays a crucial role, particularly in detecting manipulated images. Deep learning is a machine learning method based on artificial neural networks that uses deep layers to extract complex data features [3]. Architectures such as AlexNet, VGG-19, ResNet-101, and DenseNet-201 have proven superior in pattern recognition and analyzing complex visual data. AlexNet revolutionized image recognition by leveraging GPUs to accelerate neural network training. VGG-19 employs a simple layer structure with small filters, allowing the model to recognize more detailed visual patterns [4]. ResNet-101 introduced residual connections, enabling the training of deep networks without performance degradation [5]. DenseNet-201, on the other hand, maximizes information flow between layers to mitigate the vanishing gradient problem, making it an efficient and robust model for processing visual data [6].

This study is inspired by previous research using Convolutional Neural Networks (CNNs) to classify AI-generated images in the CIFAKE dataset. The CIFAKE dataset consists of 120.000 images (60.000 synthetic and 60.000 real) [7]. Previous research achieved 93% accuracy in distinguishing real images from AI-generated ones. However, one common issue during CNN training is the vanishing gradient problem. This occurs in models with very deep layers, where gradients become too small during training, reducing the network's responsiveness.

Architectures such as VGG, ResNet, DenseNet, and AlexNet have demonstrated high effectiveness in detecting synthetic images due to their ability to recognize complex elements in images [8]. ResNet addresses the challenges of training very deep networks by utilizing shortcut connections to facilitate information flow, making the network more stable and accurate [9]. DenseNet connects each layer directly to every subsequent layer, enhancing information flow efficiency and mitigating the vanishing gradient problem, enabling more optimal learning [10]. VGG excels in its deep yet straightforward convolutional layers with small filters (3x3), helping the model identify various complex image features [11]. Meanwhile, AlexNet revolutionized the field of image recognition by utilizing GPUs to accelerate training, winning the ImageNet competition in 2012 and being recognized for its effective feature extraction capabilities [12].

Several relevant studies have highlighted the significant potential of deep learning approaches in detecting synthetic images. For instance, a hybrid approach combining CNN and Support Vector Machines (SVM) was introduced in [13]. This study used a dataset of 140.000 images and achieved 88.33% accuracy. Their findings demonstrated that combining CNN and SVM could be effectively applied to synthetic image detection. Another multimodal approach utilizing BERT and CNN models was explored in [14], employing a dataset of 120.000 images (60.000 AI-generated and 60.000 real). Their proposed model achieved a maximum accuracy of 93.55%, proving the effectiveness of multimodal approaches in synthetic image detection. Similarly, the study in [15] employed ResNet and Variational Autoencoders (VAEs) on a dataset comprising 120,000 images, demonstrating their capability in effectively classifying synthetic images. This research demonstrated that ResNet achieved a maximum accuracy of 94%, effectively classifying synthetic images. The study in [16] utilized a larger dataset comprising 140,000 images, further validating the effectiveness of combining CNN with SVM, achieving an accuracy of 88.33%. While the accuracy was slightly lower compared to other approaches, the findings underscore the significant potential of CNN-SVM integration in synthetic image detection. The study in [17] introduced a dual-branch neural network model, incorporating a CNN-based architecture for each branch. By leveraging a dataset of 120,000 images, the model achieved an accuracy of 94%, demonstrating the efficacy of dual-branch architectures in distinguishing synthetic images from real ones with enhanced precision.

The novelty of this research lies in the application of Dynamic Dropout in combination with the CIFAKE and Real and Fake Face Detection datasets. This technique is designed to enhance training stability, prevent overfitting, and strengthen the generalization capability of models when applied to deep learning architectures. Dynamic Dropout operates by progressively adjusting the dropout rate during training, starting from 20% and increasing to 50%, providing a smooth and adaptive regulation transition. In this study, the CIFAKE dataset, consisting of 120.000 images, and the Real and Fake Face Detection dataset, comprising 2,000 images, were used to train and test the model's performance. This dataset combination provides sufficient data variation to evaluate the effectiveness of Dynamic Dropout in improving the performance of deep learning models such as DenseNet-201, AlexNet, ResNet-101, and VGG-19 in distinguishing between real and manipulated images. This approach not only evaluates model performance but also contributes to the development of more accurate methods for detecting visual manipulation. The adaptive Dynamic Dropout approach can serve as a foundation for practical applications in various sectors, such as social media, digital security, and privacy protection, while addressing the challenges of image manipulation technology in the digital era.

II. METHOD

This research was conducted through several stages. The flowchart can be seen in Fig 1.

A. Data Acquisition and Dataset Splitting

The dataset used in this study was divided into three main parts: training, validation, and testing, to ensure optimal model training while providing an objective performance evaluation. The CIFAKE dataset, as described in [7], consists of 120.000 images, evenly split into 60.000 real images and 60.000 synthetic images generated by artificial intelligence. The Real and Fake Face Detection dataset contains a total of 2,000 images, including human face images. This dataset also has a balanced distribution, consisting of 1,000 real face images and 1,000 manipulated synthetic face images. The training, validation, and testing data were split with a 60:20:20 ratio to maintain class distribution across all subsets. Training data were used to optimize model parameters, while validation data served to evaluate model performance during training without influencing the learning process. Examples of images from the CIFAKE and Real and Fake Face Detection datasets are shown in Fig 2.

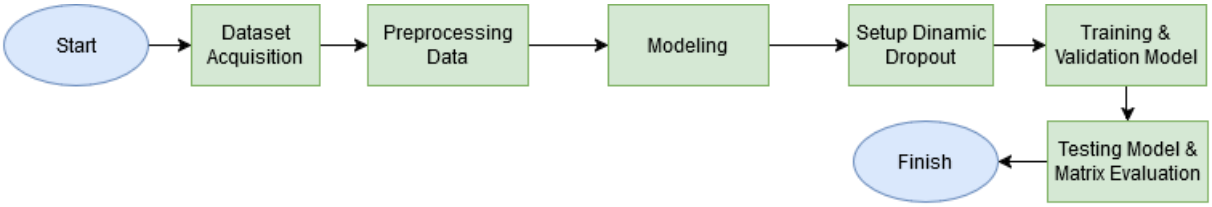


Fig. 1 Research flow diagram

Fig. 2 Image of CIFAKE and real and fake face detection

B. Preprocessing Data

The dataset in this study was processed to ensure quality and compatibility with the deep learning models used. All images were resized to 32 × 32 pixels to match the input standards of architectures such as DenseNet, ResNet, AlexNet, and VGG, while reducing computational load without losing important visual information. The images were also converted to 24-bit RGB format for detailed color pattern analysis and normalized from a range of 0–255 to 0–1 using rescale=1./255, to stabilize input and improve training efficiency.

Training data underwent augmentation, including rotation, shifting, shearing, zooming, and horizontal flipping up to 20%, to increase variation and reduce the risk of overfitting. Validation and testing datasets were only normalized to maintain consistency and ensure objective evaluation. The dataset was then divided into three parts, which are training, validation, and testing, with a ratio of 60:20:20. ensuring balanced class distribution for training, validating, and objectively evaluating the model's performance.

C. Modeling

This study utilized four deep learning models, which are AlexNet, VGG-19, ResNet-101, and DenseNet-201, chosen based on their ability to handle complex datasets and unique architectural approaches. AlexNet, with five convolutional layers and three fully connected layers, revolutionized image classification through the ReLU activation function and dropout techniques to prevent overfitting [12]. VGG-19, with 19 main layers and small filters (3x3), excels in capturing local patterns in detail despite its large parameter size [18]. ResNet-101 addresses the vanishing gradient problem using residual blocks, providing stability in deep networks [19]. DenseNet-201, with densely connected layers, enhances parameter efficiency and training stability with 201 layers, making it effective for complex datasets like CIFAKE and Real and Fake Face Detection [20].

D. Dynamic Dropout

Dropout was first introduced in [21] as a technique to prevent neural networks from overfitting. Dropout is a regularization technique designed to reduce the risk of overfitting in neural networks. This method works by randomly deactivating a certain number of neuron units during training, based on a dropout probability (p). Its primary goal is to force the neural network to become more robust by preventing reliance on specific combinations of neurons.

In dropout, a unit mask (m_i) is generated from a Bernoulli distribution with the dropout probability (p) as in (1),

$$m_i \approx Bernoulli(p) \quad (1)$$

Where m_i is the binary mask for the i-th neuron, which can have a value of either zero or one depending on the outcome of the Bernoulli distribution. If m_i = 0. the neuron is deactivated during training. Conversely, if m_i = 1, the neuron remains active and contributes to the network's computations.

The activation after applying dropout is calculated as in (2),

$$a^{(l)} = m \odot \mathrm{f}(\mathrm{W} \cdot a^{(l-1)} + \mathrm{b}) \quad (2)$$

In this formula, $a^{(l)}$ represents the activation produced at the l-th layer after dropout is applied. The variable m is the binary mask applied to all neurons in the layer. The

function f is a non-linear activation function, such as ReLU, applied after the linear computation of weights and biases. W is the weight matrix for the connections between neurons, while $a^{(l-1)}$ represents the activation from the previous layer. The variable b denotes the bias, and the operator $\odot$ (Hadamard product) ensures that only active neurons contribute to the final output.

Static dropout, which uses a fixed dropout rate (p) throughout training, has limitations. In the early stages of training, a high dropout rate can eliminate too much important information, potentially hindering the learning of basic patterns [22]. Conversely, in the later stages of training, a low dropout rate may not provide sufficient regularization to prevent overfitting [23].

To address these limitations, this study employs Dynamic Dropout, which adjusts the dropout rate gradually during training. This technique allows the dropout rate to change progressively, providing lighter regularization in the early stages and stronger regularization in the later stages of training to prevent overfitting. The dropout rate is calculated using (3),

$$p(t) = p_{initial} + \left(p_{final} - p_{initial}\right) \times \min\left(\frac{t}{T}, 1{,}0\right) \quad (3)$$

In this study, the initial dropout rate ($p_{(initial)}$) was set at 20 percent, while the final dropout rate ($p_{(final)}$) was set at 50 percent. The initial rate of 20 percent was chosen to provide greater stability in the early stages of training, allowing the model to learn basic data patterns without excessive disruption. Meanwhile, the final rate of 50 percent offered stronger regularization to prevent overfitting as the model approached convergence. These values were selected based on findings from [21], as well as [24], which explored improvements in neural networks through dropout techniques.

To support a smooth transition in regularization, the dropout rate was gradually increased by 10 percent at each interval. This approach allowed the model sufficient time to adjust gradient updates in accordance with changes in the dropout rate. This process created an optimal balance between low regularization at the start of training and high regularization at the end of training. The number of neurons that remained active in each layer was calculated using (4),

$$N_{active} = N \times (1 - p) \quad (4)$$

This formula indicates that the number of active neurons (N_{active}) is the product of the total number of neurons in the layer (N) and the proportion of neurons that remain active ($1 - p$, where p is the dropout rate) [25]. As described in (4), this relationship enables us to track the gradual reduction in active neurons as the dropout rate increases. By applying a 10 percent increment in dropout at each interval, the network progressively reduces its reliance on individual neurons, promoting better generalization. Table I presents the computed values of active neurons across different training epochs, illustrating the direct effect of dropout on the model's effective capacity over time.

An increase in dropout of less than 10 percent may result in regulatory changes that are too small, rendering the effect of regularization less significant and prolonging the training process. Conversely, an increase of more than 10 percent may lead to abrupt transitions, making it difficult for the model to adapt to the regulatory changes. This can reduce training stability and increase the risk of losing too many active neurons in a short period.

With the Dynamic Dropout approach employed in this study, the model is given the opportunity to learn data patterns more stably in the early stages of training, while still receiving sufficient regularization in the later stages to prevent overfitting. Dynamic Dropout is applied to the fully connected layers of the deep learning models used, such as ResNet-101, DenseNet-201, AlexNet, and VGG-19. This technique is expected to improve the generalization capability of the model on the CIFAKE and Real and Fake Face Detection datasets, which exhibit significant data variation.

TABLE I
ACTIVE NEURONS DURING TRAINING

Step	Dropout Rate	Active Neurons(%)	Active Neurons
Start of Training	20%	80%	0.8 N
After 10% Increase	30%	70%	0.7 N
After 20% Increase	40%	60%	0.6 N
End of Training	50%	50%	0.5 N

E. Training And Validation Model

The training and validation process was conducted to train and evaluate the performance of models in detecting real and manipulated images from the CIFAKE and Real and Fake Face Detection datasets. The deep learning models used included ResNet-101, DenseNet-201, AlexNet, and VGG-19, which were trained using image data resized to dimensions of 32 × 32 pixels.

The training process was performed by comparing the number of epochs (5, 10. and 15), optimizers (Adam and Nadam), each configured with an initial learning rate of 0.001, and regularization techniques (Dynamic Dropout and Static Dropout) to evaluate the impact of each configuration on model performance. Training was carried out with a batch size of 32. To prevent overfitting, early stopping was applied with a patience parameter of 5 epochs, meaning the training would halt if no improvement in validation accuracy was observed for 5 consecutive epochs. The loss function used was categorical cross-entropy, aligned with the multi-class classification objective of this study.

Dynamic Dropout was implemented as a regularization technique in the fully connected layers of the models. The dropout rate started at 20 percent and progressively increased to 50 percent during training. This gradual increase provided a smooth regulation transition, starting with low regularization to ensure stability in the early stages of training and increasing to higher regularization to minimize overfitting in later stages. The incremental increase of 10 percent allowed the models to adjust gradient updates gradually, maintaining training stability.

Validation was performed at each epoch to evaluate model performance using data not included in the training process. Validation results were used to monitor trends in accuracy and loss, enabling the detection of potential overfitting. If validation accuracy trends declined despite improvements in training accuracy, this indicated the need for hyperparameter adjustments or early termination of the training process.

F. Testing and Evaluation Matrix

The testing process was conducted using test data consisting of images from the CIFAKE and Real and Fake Face Detection datasets. The purpose of this testing was to evaluate the model's classification performance at every stage, including training, validation, and testing. The performance was measured using various metrics, including accuracy, precision, recall, and F1-Score. These metrics provided a comprehensive overview of the model's effectiveness in accurately classifying the data [26]. The confusion matrix used in this study illustrated four key variables: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). This matrix visualized the four possible prediction scenarios, offering detailed insights into the distribution of prediction errors. It served as a crucial tool for analyzing the strengths and weaknesses of the model on the test data [27].

III. RESULT AND DISCUSSION

This section discusses the research findings aimed at evaluating the performance of deep learning models in detecting real and manipulated images, considering the number of epochs, types of optimizers, and regularization techniques using Dynamic Dropout and Static Dropout. The study also explores how variations in epochs and optimizer choices influence model performance during training and validation. Four deep learning models, namely DenseNet-201, AlexNet, VGG-19, and ResNet-101, were employed, utilizing a combined dataset of CIFAKE and Real and Fake Face Detection, which features balanced distributions and high visual complexity. The analysis includes evaluations of training, validation, and testing outcomes based on accuracy, precision, recall, and F1-Score metrics to identify the strengths and weaknesses of the models in detecting manipulated images.

A. Training And Validation Model

The training process was carried out to compare the performance across epochs 5, 10. and 15; the Adam and Nadam optimizers; and the performance of Dynamic Dropout and Static Dropout on four deep learning models: DenseNet-201, AlexNet, ResNet-101, and VGG-19. The training was conducted with various configurations to evaluate how each factor influenced the outcomes of training and validation. Equation (1) describes the fundamental mechanism of dropout, where each neuron is independently retained or deactivated based on a Bernoulli distribution with probability p. This stochastic process prevents co-adaptation among neurons and enhances generalization ability. By applying this principle, the dropout mechanism was utilized in both Static and Dynamic Dropout models during training to regulate neuron activity and prevent overfitting. Building upon this concept, Equation (2) defines how activations are calculated after dropout is applied. Regardless of whether Static or Dynamic Dropout is used, only the neurons that remain active contribute to the final activation function. This ensures that the network continues learning from essential features while discarding redundant information. The implementation

of this process is consistent across all models trained in this study.

The evaluation of epochs aimed to determine the optimal training duration that balances accuracy, validation performance, and the risk of overfitting. Epoch 15 yielded the best performance for most models, with DenseNet-201 achieving the highest validation accuracy at 97.32% (loss: 0.0804). AlexNet and ResNet-101 also reached their optimal results at epoch 15, with validation accuracies of 93.84% and 88.70%, respectively. VGG-19 recorded its highest accuracy at epoch 20 (88.27%), although the difference compared to epoch 15 was relatively small. The complete results are presented in Table II. In the training process, the selection of the optimizer plays a crucial role in determining training efficiency and stability. The evaluation was conducted using the Adam and Nadam optimizers. Based on the results, the Adam optimizer consistently outperformed Nadam in training accuracy, validation accuracy, and loss values. For more detailed results, refer to Table III, which presents the performance of each optimizer across all models. These results indicate that Adam is the best optimizer for most models, providing a balance between efficiency, stability, and generalization, especially on complex datasets.

During training, Dynamic Dropout consistently outperformed Static Dropout. For DenseNet-201, Dynamic Dropout achieved a validation accuracy of 97.61%, higher than Static Dropout (97.18%), with lower loss values. A similar pattern was observed in AlexNet (93.64% vs. 90.02%), ResNet-101 (87.76% vs. 84.98%), and VGG-19 (87.89% vs. 86.68%). Full results are available in Table IV. Following (3), the dropout rate was progressively increased during training, starting at 20% in the early epochs and reaching 50% by epoch 15. This controlled increase ensured that the network gradually reduced its reliance on specific neurons, allowing for more robust feature extraction and preventing overfitting. As illustrated in Fig 3, this gradual adjustment contributed to improved model stability and generalization, leading to higher validation accuracy.

TABLE II
TRAINING AND VALIDATION RESULTS FOR DIFFERENT EPOCHS ACROSS MODELS

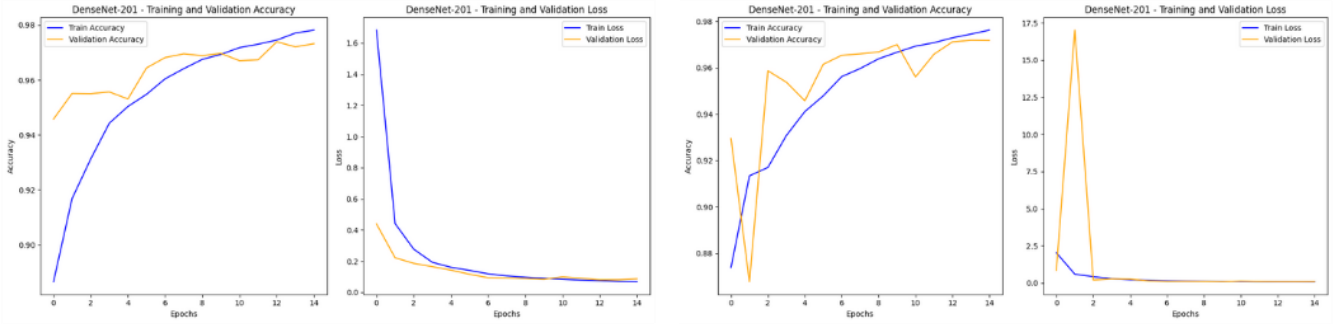
No	Model	Epoch	Train Acc	Train Loss	Val Acc	Val Loss
1	VGG-19	10	0.8381	0.4073	0.8766	0.3419
		15	**0.8468**	**0.3928**	**0.8739**	**0.3414**
		20	0.8487	0.3881	0.8827	0.3228
2	ResNet-101	10	0.8780	0.3391	0.8799	0.3333
		15	**0.8826**	**0.3101**	**0.8870**	**0.3144**
		20	0.8679	0.3199	0.8855	0.3229
3	AlexNet	10	0.9249	0.2524	0.8998	0.3240
		15	**0.9489**	**0.1717**	**0.9384**	**0.2033**
		20	0.9485	0.1772	0.9350	0.2157
4	DenseNet-201	10	0.9674	0.0923	0.9684	0.0915
		15	**0.9780**	**0.0673**	**0.9732**	**0.0804**
		20	0.9727	0.0680	0.9713	0.0894

TABLE III
TRAINING AND VALIDATION RESULTS OF ADAM AND NADAM OPTIMIZERS ACROSS MODELS

No	Model	Optimize	Train Acc	Train Loss	Val Acc	Val Loss
1	VGG-19	**Adam**	**0.8471**	**0.3882**	**0.8789**	**0.3315**
		Nadam	0.8468	0.3928	0.8739	0.3414
2	ResNet-101	**Adam**	**0.8608**	**0.4032**	**0.8776**	**0.3751**
		Nadam	0.8526	0.4231	0.8670	0.4044
3	AlexNet	**Adam**	**0.9509**	**0.1757**	**0.9364**	**0.2086**
		Nadam	0.9449	0.1917	0.9334	0.2233
4	DenseNet-201	**Adam**	**0.9781**	**0.0621**	**0.9761**	**0.0677**
		Nadam	0.9780	0.0673	0.9732	0.0804

TABLE IV
TRAINING AND VALIDATION RESULTS FOR DYNAMIC AND STATIC DROPOUT ACROSS MODELS

No	Model	Dropout	Train Acc	Train Loss	Val Acc	Val Loss
1	VGG-19	**Dynamic Dropout**	**0.8471**	**0.3882**	**0.8789**	**0.3315**
		Static Dropout	0.8262	0.4290	0.8668	0.3587
2	ResNet-101	**Dynamic Dropout**	**0.8608**	**0.4032**	**0.8776**	**0.3751**
		Static Dropout	0.8094	0.4576	0.8498	0.3842
3	AlexNet	**Dynamic Dropout**	**0.9509**	**0.1757**	**0.9364**	**0.2086**
		Static Dropout	0.9421	0.2098	0.9002	0.2967
4	DenseNet-201	**Dynamic Dropout**	**0.9781**	**0.0621**	**0.9761**	**0.0677**
		Static Dropout	0.9768	0.0793	0.9718	0.0954



a. Graph of DenseNet-201 with Dynamic Dropout **b. Graph of DenseNet-201 with Static Dropout**

Fig. 3 Training and Validation Graph for the DenseNet-201 Model

Dynamic Dropout significantly improves training and validation performance compared to Static Dropout, with adaptive regulation that prevents overfitting, especially on complex datasets such as CIFAKE and Real and Fake Face Detection. DenseNet-201, utilizing the Adam optimizer and the 15th epoch, recorded the highest validation accuracy of 97.61% (loss: 0.0677), demonstrating the best performance among all models. The chart in Fig. 3 clearly illustrates the comparison of training and validation results using Dynamic Dropout and Static Dropout on DenseNet-201, emphasizing that Dynamic Dropout provides significant improvements in model stability and generalization compared to Static Dropout.

Overall, the findings of this study highlight the importance of selecting the appropriate number of epochs, optimizer, and dropout technique to enhance deep learning model performance. The 15th epoch, Adam optimizer, and Dynamic Dropout delivered the best results in improving accuracy, stability, and generalization, making them the optimal combination for training models on CIFAKE and Real and Fake Face Detection datasets.

B. Testing And Evaluation

The testing process also evaluated the performance of four deep learning models, namely DenseNet-201, AlexNet, ResNet-101, and VGG-19, using epochs 5, 10. and 15; the Adam and Nadam optimizers; as well as the performance of Dynamic Dropout and Static Dropout. The evaluation was conducted using metrics such as accuracy, precision, recall, and F1-score.

The results showed that the 15th epoch consistently provided the best performance for most models. Further details can be found in Table V, which presents the complete testing results for each model based on the number of epochs.

The testing also evaluated the impact of the optimizer, showing that the Adam optimizer consistently delivered better results compared to Nadam. The complete results are presented in Table VI.

Thus, the Adam optimizer has proven to be a superior and more efficient choice for training deep learning models on the dataset used in this study.

The evaluation results of Dynamic Dropout and Static Dropout show that Dynamic Dropout consistently outperforms Static Dropout. Table VII presents the complete performance results of the four models using Dynamic Dropout and Static Dropout.

TABLE V
TESTING RESULTS FOR DIFFERENT EPOCH COUNTS ACROSS MODELS

No	Model	Epoch	Accuracy	Precision	Recall	F1 Score
1	VGG-19	10	0.8744	0.8737	0.8749	0.8743
		15	**0.8792**	**0.8811**	**0.8778**	**0.8794**
		20	0.8802	0.8989	0.8664	0.8824
2	ResNet-101	10	0.8756	0.8681	0.8813	0.8746
		15	**0.8552**	**0.8247**	**0.8783**	**0.8506**
		20	0.8805	0.8695	0.8890	0.8792
3	AlexNet	10	0.9158	0.8999	0.9295	0.9145
		15	**0.9468**	**0.9302**	**0.9621**	**0.9459**
		20	0.9438	0.9334	0.9532	0.9432
4	DenseNet-201	10	0.9669	0.9556	0.9777	0.9665
		15	**0.9742**	**0.9733**	**0.9758**	**0.9745**
		20	0.9782	0.9790	0.9774	0.9782

TABLE VI
TESTING RESULTS OF ADAM AND NADAM OPTIMIZERS ACROSS MODELS

No	Model	Optimize	Accuracy	Precision	Recall	F1 Score
1	VGG-19	Nadam	0.8792	0.8430	0.9088	0.8746
		Adam	**0.8792**	**0.8811**	**0.8778**	**0.8794**
2	ResNet-101	Nadam	0.8552	0.8247	0.8783	0.8506
		Adam	**0.8720**	**0.8566**	**0.8837**	**0.8700**
3	AlexNet	Nadam	0.9422	0.9583	0.9284	0.9431
		Adam	**0.9468**	**0.9302**	**0.9621**	**0.9459**
4	DenseNet-201	Nadam	0.9742	0.9733	0.9758	0.9745
		Adam	**0.9766**	**0.9793**	**0.9742**	**0.9767**

TABLE VII
TESTING RESULTS OF DYNAMIC AND STATIC DROPOUT FOR EACH MODEL

No	Model	Dropout	Accuracy	Precision	Recall	F1 Score
1	VGG-19	Static Dropout	0.8675	0.8444	0.8853	0.8644
		Dynamic Dropout	**0.8792**	**0.8811**	**0.8778**	**0.8794**
2	ResNet-101	Static Dropout	0.8462	0.8184	0.8666	0.8418
		Dynamic Dropout	**0.8552**	**0.8247**	**0.8783**	**0.8506**
3	AlexNet	Static Dropout	0.9308	0.9565	0.9097	0.9325
		Dynamic Dropout	**0.9468**	**0.9302**	**0.9621**	**0.9459**
4	DenseNet-201	Static Dropout	0.9711	0.9705	0.9718	0.9711
		Dynamic Dropout	**0.9742**	**0.9733**	**0.9758**	**0.9745**

The combination of the 15th epoch, Adam optimizer, and Dynamic Dropout delivered the best performance across most tested models, with DenseNet-201 achieving the highest results: a validation accuracy of 97.61%, precision of 97.33%, recall of 97.58%, and an F1-score of 97.45%. The 15th epoch provided an optimal balance between accuracy and the risk of overfitting, while Adam demonstrated superior stability and generalization compared to Nadam. Dynamic Dropout consistently improved evaluation metrics over Static Dropout, making it a more effective regularization technique. These findings highlight the importance of selecting appropriate training parameters and regularization techniques to optimize deep learning model performance on the CIFAKE and Real and Fake Face Detection datasets.

IV. CONCLUSION

This study comprehensively evaluated the impact of training configurations, including epoch count, optimizer selection, and the Dynamic Dropout regularization technique in comparison to Static Dropout, across four deep learning models: DenseNet-201, AlexNet, ResNet-101, and VGG-19, for classifying real and synthetic images using the CIFAKE and Real and Fake Face Detection datasets. Experimental findings confirmed that the optimal configuration (15 epochs, the Adam optimizer, and Dynamic Dropout) consistently

outperformed Static Dropout in key performance metrics. DenseNet-201 with Dynamic Dropout achieved the highest accuracy of 97.42%, with a precision of 97.33%, recall of 97.58%, and an F1-score of 97.45%. The Adam optimizer demonstrated superior stability over Nadam, particularly in models with complex architectures, while the 15th epoch provided an optimal trade-off between training and validation accuracy, mitigating overfitting risks. These results emphasize the significance of systematically optimizing training parameters to enhance deep learning performance. Future research should explore the integration of adaptive dropout mechanisms, investigate the scalability of this approach on diverse datasets, and validate its efficacy in real-world applications such as digital forensics and AI-generated content authentication.

REFERENCES

[1] R. Angelika Septi Rahayu and H. Santoso, "Analysis of Fake Face Images: Detecting the Authenticity of Manipulated Images Using Variational Autoencoder Methods and Deep Neural Network Forensics," *Sibatik J. | Vol.*, vol. 2, no. 9, pp. 2701–2726, 2023, [Online]. Available: https://publish.ojs-indonesia.com/index.php/SIBATIK.

[2] S. C. Hanebuth, D. Kalokitis, and A. Cedrone, "Structured Methodology for the Investigation of Contact Voltages," *IEEE Power Energy Technol. Syst. J.*, vol. 1, no. November 2014, pp. 1–11, 2015, doi: 10.1109/jpets.2014.2363403.

[3] L. Alzubaidi *et al.*, *Review of deep learning: concepts, CNN architectures, challenges, applications, future directions*, vol. 8, no. 1. Springer International Publishing, 2021.

[4] S. Tammina, "Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images," *Int. J. Sci. Res. Publ.*, vol. 9, no. 10, p. p9420, 2019, doi: 10.29322/ijsrp.9.10.2019.p9420.

[5] J. Liang, "Image classification based on RESNET," *J. Phys. Conf. Ser.*, vol. 1634, no. 1, 2020, doi: 10.1088/1742-6596/1634/1/012110.

[6] M. Tripathi, "Analysis of Convolutional Neural Network based Image Classification Techniques," *J. Innov. Image Process.*, vol. 3, no. 2, pp. 100–117, 2021, doi: 10.36548/jiip.2021.2.003.

[7] J. J. Bird and A. Lotfi, "CIFAKE: Image Classification and Explainable Identification of AI-Generated Synthetic Images," *IEEE Access*, vol. 12, no. December 2023, pp. 15642–15650, 2024, doi: 10.1109/ACCESS.2024.3356122.

[8] Y. Wang, Y. Hao, and A. X. Cong, "Harnessing Machine Learning for Discerning AI-Generated Synthetic Images," 2024, [Online]. Available: http://arxiv.org/abs/2401.07358.

[9] P. D. W. Ayu, G. A. Pradipta, I. M. D. Susila, D. P. Hostiadi, and M. Liandana, "Deep Learning Based Detection and Classification of Amniotic Fluid Echogenicity Type for Enhanced Prenatal Diagnosis," *Int. J. Intell. Eng. Syst.*, vol. 18, no. 1, pp. 246–267, 2025, doi: 10.22266/ijies2025.0229.18.

[10] C. Jiang, C. Jiang, D. Chen, and F. Hu, "Densely Connected Neural Networks for Nonlinear Regression," *Entropy*, vol. 24, no. 7, pp. 0–9, 2022, doi: 10.3390/e24070876.

[11] I. W. Jepriana, I. G. Bintang, A. Budaya, G. A. Pradipta, P. Desiana, and W. Ayu, "Combining Oversampling and Pretrained Feature Extractor For Classification Diabetic Foot Uclear Thermogram Images," vol. 12, no. 2, pp. 267–277, 2024.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks Alex," *Encycl. Mov. Disord. Three-Volume Set*, pp. V2-257-V2-259, 2010, doi: 10.1016/B978-0-12-374105-9.00493-7.

[13] B. Huang *et al.*, "Implicit Identity Driven Deepfake Face Swapping Detection," pp. 4490–4499, 2023, doi: 10.1109/cvpr52729.2023.00436.

[14] E. al. Vismay Vora, "A Multimodal Approach for Detecting AI Generated Content using BERT and CNN," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 11, no. 9, pp. 691–701, 2023, doi: 10.17762/ijritcc.v11i9.8861.

[15] G. E. Bartos, Ó. Egyetem, S. Ö. Akyol, and S. Akyol, "Deep Learning for Image Authentication: A Comparative Study on Real and AI-Generated Image Classification," no. November, 2023, [Online]. Available: https://www.researchgate.net/publication/375952278.

[16] J. Mallet, L. Pryor, R. Dave, and M. Vanamala, "Deepfake Detection Analyzing Hybrid Dataset Utilizing CNN and SVM," *ACM Int. Conf. Proceeding Ser.*, pp. 7–11, 2023, doi: 10.1145/3596947.3596954.

[17] J. Gallagher and W. Pugsley, "Development of a Dual-Input Neural Model for Detecting AI-Generated Imagery," 2024, [Online]. Available: http://arxiv.org/abs/2406.13688.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition Kaiming," pp. 770–778, 2015, doi: 10.14456/mijet.2021.8.

[20] G. Huang, Z. Liu, and L. van der Maaten, "Densely Connected Convolutional Networks," vol. 39, no. 9, pp. 1442–1446, 2016, doi: https://doi.org/10.48550/arXiv.1608.06993.

[21] S. Li, L. Wang, J. Li, and Y. Yao, "Image Classification

Algorithm Based on Improved AlexNet," *J. Phys. Conf. Ser.*, vol. 1813, no. 1, 2021, doi: 10.1088/1742-6596/1813/1/012051.

[22] Y. Gal, "Dropout as a Bayesian Approximation : Insights and Applications," 2015.

[23] A. Coates, H. Lee, and A. Y. Ng, "An analysis of single-layer networks in unsupervised feature learning," *J. Mach. Learn. Res.*, vol. 15, pp. 215–223, 2011.

[24] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," pp. 1–18, 2012, [Online]. Available: http://arxiv.org/abs/1207.0580.

[25] S. Wager, S. Wang, and P. Liang, "Dropout training as adaptive regularization," *Adv. Neural Inf. Process. Syst.*, pp. 1–11, 2013.

[26] Ž. Vujović, "Classification Model Evaluation Metrics," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 6, pp. 599–606, 2021, doi: 10.14569/IJACSA.2021.0120670.

[27] J. Erbani, P. É. Portier, E. Egyed-Zsigmond, and D. Nurbakova, "Confusion Matrices: A Unified Theory," *IEEE Access*, vol. 12, no. October, 2024, doi: 10.1109/ACCESS.2024.3507199.