Accredited SINTA 2 Ranking
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online at: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 8 No. 6 (2024) 737 - 747 e-ISSN: 2580-0760

Lightweight Models for Real-Time Steganalysis: A Comparison of MobileNet, ShuffleNet, and EfficientNet

Achmad Bauravindah[1]*, Dhomas Hatta Fudholi[2]
[1]Master Program in Informatics, Faculty of Industrial Technology, Islamic University of Indonesia, Yogyakarta, Indonesia
[2]Department of Informatics, Faculty of Industrial Technology, Islamic University of Indonesia, Yogyakarta, Indonesia
[1]23917016@students.uii.ac.id, [2]hatta.fudholi@uii.ac.id

Abstract

In the digital age, the security of communication technologies is paramount, with cybercrime projected to reach $10.5 trillion annually by 2025. While encryption is vital, decrypted data remains vulnerable, prompting the exploration of steganography as an additional security layer. Steganography conceals data within digital media, but its misuse for cyberattacks—such as embedding malware—has highlighted the need for steganalysis, the detection of hidden data. Despite extensive research, few studies have explored lightweight deep-learning models for real-time steganalysis in resource-constrained environments like mobile devices. This research evaluates MobileNet, ShuffleNet, and EfficientNet for such tasks, using the BOSSbase-1.01 dataset. Models were assessed based on accuracy, computational efficiency, and resource usage. MobileNet achieved the highest computational speed but with only 63.8% accuracy, falling short of practical application. ShuffleNet and EfficientNet performed at random-guessing levels with 50% accuracy, reflecting the challenges of steganalysis on mobile platforms. Future work aims to improve accuracy by integrating advanced preprocessing techniques, attention mechanisms, and hybrid architectures, as well as leveraging ensemble methods for improved detection. Data augmentation, transfer learning, and hyperparameter tuning will also be explored to optimize model performance. This study contributes by identifying these challenges and offering insights for future research, focusing on optimizing models and preprocessing techniques to enhance detection accuracy in resource-constrained environments.

Keywords: deep learning; lightweight; steganography; steganalysis; security

How to Cite: A. Bauravindah and D. H. Fudholi, "Lightweight Models for Real-Time Steganalysis: A Comparison of MobileNet, ShuffleNet, and EfficientNet", *J. RESTI (Rekayasa Sist. Teknol. Inf.)* , vol. 8, no. 6, pp. 737 - 747, Dec. 2024.
DOI: https://doi.org/10.29207/resti.v8i6.6091

1. Introduction

In today's digitally interconnected world, communication technologies have become integral to daily life, with over 5.48 billion unique mobile users globally, accounting for more than 68% of the world's population [1]. This surge in connectivity has led to an exponential increase in data transmission, amplifying concerns over data security and privacy. While encryption remains a fundamental tool for protecting data during transmission—utilized by 62% of organizations as reported by the Ponemon Institute [2]—it has inherent limitations [3]. Encrypted data, once decrypted for use, becomes vulnerable to unauthorized access, especially during transfers across networks susceptible to interception and cyberattacks.

To mitigate these vulnerabilities, steganography has emerged as a complementary security measure. By embedding sensitive information within digital media like images or audio files, steganography conceals the very existence of the data, making it less likely to attract unwanted attention [4]. However, this technique is a double-edged sword. Cybercriminals have increasingly exploited steganography to hide malicious code within seemingly benign files, effectively bypassing traditional security mechanisms [5], [6]. The ubiquity of digital images—over 14 billion uploaded daily across social media platforms [7]—makes them ideal carriers for such concealed threats.

Traditional steganalysis methods, such as statistical analysis and pattern recognition, face significant limitations in detecting sophisticated steganographic techniques used by malicious actors. These conventional approaches rely heavily on predefined features and assumptions about the data, making them less adaptable to the complexity and variability of

Received: 29-09-2024 | Accepted: 23-11-2024 | Published Online: 26-12-2024

modern steganographic methods. As a result, they often struggle to detect hidden information when steganographic patterns do not align with expected distributions or when data is manipulated in subtle, novel ways. Consequently, these methods exhibit lower detection accuracy compared to more advanced models, as highlighted in studies [8], [9].

In contrast, deep learning models offer a dynamic and adaptive approach to steganalysis. Unlike traditional methods, deep learning algorithms can automatically learn intricate patterns and representations from raw data without relying on predefined features. This flexibility allows deep learning models to generalize better across different types of steganography, including those employing sophisticated or evolving techniques. As a result, deep learning-based steganalysis models achieve higher detection accuracy, particularly in cases where traditional methods fall short, as highlighted in studies [8], [9].

In contrast, deep learning models offer a dynamic and adaptive approach to steganalysis [10], [11], [12]. Unlike traditional methods, deep learning algorithms can automatically learn intricate patterns and representations from raw data without relying on predefined features. This flexibility allows deep learning models to generalize better across different types of steganography, including those employing sophisticated or evolving techniques. As a result, deep learning-based steganalysis models achieve higher detection accuracy, particularly in cases where traditional methods fall short, as highlighted in studies [8], [9]. Despite their detection capabilities, most deep learning models are resource-intensive, making them unsuitable for real-time use on devices with limited processing capacity, such as mobile phones. This gap is critical because mobile devices are especially vulnerable to steganographic threats [13], necessitating solutions that can operate efficiently with minimal resource consumption to detect hidden data in real-time [14].

To address this challenge, we focus on integrating lightweight deep learning architectures—specifically MobileNet, ShuffleNet, and EfficientNet—into steganalysis systems. These models were chosen for their design, which optimizes performance while minimizing computational requirements. MobileNet employs depthwise separable convolutions to reduce the number of parameters and computational cost [15]. ShuffleNet introduces pointwise group convolution and channel shuffle operations to further enhance efficiency [16]. EfficientNet utilizes a compound scaling method that uniformly scales network dimensions, achieving better accuracy with fewer parameters [17]. By leveraging these architectures, we aim to develop a steganalysis system capable of operating effectively in resource-constrained environments without compromising detection accuracy.

Furthermore, these architectures provide significant advantages in resource-constrained environments due to their low computational complexity, memory efficiency, and reduced power consumption [15], [16], [17]. Their architectural innovations, such as depthwise separable convolutions, group convolutions, and compound scaling, minimize FLOP counts and parameter sizes, allowing for real-time processing even on devices with limited hardware. Moreover, the reduced computational demand translates to lower power consumption, which is critical for battery-powered devices, such as smartphones or drones, involved in continuous steganalysis operations. Despite their lightweight design, these models maintain high levels of accuracy, which is essential for the precise detection of hidden information in steganalysis tasks.

Our research addresses the gap between advanced steganalysis capabilities and the limitations of resource-constrained devices by proposing a lightweight yet accurate system for real-time detection of hidden information within digital images on mobile platforms. This solution enhances security against steganography-based attacks, providing a practical approach to safeguarding data in an era where mobile communication is ubiquitous and increasingly targeted by cyber threats. By incorporating mobile-friendly deep learning models such as MobileNet, ShuffleNet, and EfficientNet—designed to operate with fewer parameters and faster inference times—our steganalysis system is optimized for mobile and resource-constrained platforms, ensuring effective real-time detection of steganographic content.

Steganography, the art of concealing information within digital media, is a technique designed to keep sensitive data hidden from detection [18]. This concept differs from encryption, where data is scrambled into unreadable formats, as steganography seeks to make the data itself undetectable. One of the most popular methods is the Least Significant Bit (LSB) technique, which embeds hidden information by altering the least significant bits of image pixels, causing changes that are imperceptible to the human eye [19], [20]. According to [21], LSB methods are simple but prone to detection by statistical methods. More advanced techniques, such as Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT), operate in the frequency domain, embedding data into more resilient parts of the media [22]. Steganography plays a dual role: it enhances privacy and security, but also raises ethical concerns, particularly when misused for illegal purposes like data exfiltration or terrorism [23]. This dual nature presents a challenge in balancing security, capacity, and imperceptibility, especially when faced with increasing detection techniques.

Steganalysis is the countermeasure to steganography, developed to detect hidden information within digital files [24]. Traditional steganalysis approaches rely on statistical techniques that exploit inconsistencies in pixel patterns or noise created by hidden data [25]. [26]

This is an open access article under the CC BY-4.0 license

pioneered the use of statistical methods, which identified deviations in pixel distributions in LSB-based images. Feature extraction techniques, such as the subtractive pixel adjacency matrix, have been widely applied to steganalysis [27]. However, these traditional techniques struggle with detecting modern steganographic methods that introduce minimal detectable changes, especially as adversaries have refined their algorithms. As [28] argues, traditional steganalysis is becoming increasingly ineffective against newer adaptive embedding techniques, prompting a shift towards machine learning approaches based on clustering, which can detect more complex patterns.

Deep learning represents a significant leap in the capabilities of steganalysis, providing more powerful tools for both feature extraction and classification. Convolutional Neural Networks (CNNs) are particularly well-suited to image-based tasks because of their ability to automatically extract features from raw data [29]. According to [30] proposed a deep learning model, SRNet, which demonstrated an unprecedented detection accuracy of 90% for low embedding rates in the BOSSbase dataset, far exceeding earlier methods. Such advancements underscore deep learning's capability to tackle increasingly complex steganography detection tasks. This transition away from traditional machine learning approaches toward deep learning-based models reflects the growing complexity of modern steganography.

Recent developments in deep learning in steganalysis initially focused on heavy models like GoogleNet, a CNN architecture known for its high image classification performance and the efficient extraction of multi-dimensional features [11]. He introduced an enhanced GoogleNet-based model, referred to as EGN, aimed at improving image steganalysis accuracy. The EGN model uses a high-pass filter (HPF) to enhance noise, which is crucial for detecting hidden data, as the hidden information in steganographic images often resembles noise. The integrated GoogleNet model, combined with its variants, leverages ensemble learning to reduce bias in classification, significantly improving detection accuracy. Zhang's experiments demonstrated that the EGN model achieved a detection accuracy of 96.18% for images embedded with the S-UNIWARD algorithm at an embedding rate of 0.4 bits per pixel (bpp), outperforming traditional models by a wide margin. By using ensemble variants of GoogleNet, Zhang's approach mitigated the high computational demands typically associated with deep CNN models while maintaining high accuracy, addressing some limitations of early heavy models like GoogleNet.

The increasing demand for real-time steganalysis, particularly on mobile platforms, has led to a focus on lightweight deep learning architectures. Models like MobileNet, ShuffleNet, and EfficientNet have been developed to operate efficiently in resource-constrained environments, striking a balance between performance and computational cost [15], [16], [17]. [15] introduced MobileNet, which employs depthwise separable convolutions to reduce the number of parameters and computational load, achieving competitive accuracy while reducing the parameter compared to standard CNN architectures. While MobileNet has been primarily used for tasks like image classification, its efficiency makes it a promising candidate for steganalysis, particularly in real-time scenarios where computational resources are limited. However, as noted by Zhang et al. (2018), one of MobileNet's limitations is its lower accuracy in tasks that require high precision, such as detecting subtle hidden patterns in stego images [16].

ShuffleNet, another lightweight architecture, offers a solution through grouped convolutions and channel shuffling, which reduce computational complexity while maintaining accuracy [16]. Although ShuffleNet has not yet been extensively tested in steganalysis, its efficiency in handling high-dimensional data suggests it may be highly suitable for this task, especially in resource-limited environments. Future research could explore applying ShuffleNet to steganalysis tasks, particularly in mobile environments where computational power and memory are constrained.

EfficientNet, introduced by [17], further advances the field by optimizing the balance between depth, width, and resolution. EfficientNet scales these dimensions systematically, achieving state-of-the-art accuracy on the ImageNet dataset while using fewer parameters and FLOPs compared to previous CNN architectures. Given its scalability and efficiency, EfficientNet offers considerable potential for steganalysis, particularly in scenarios where both high accuracy and low computational costs are critical. While research on EfficientNet in the context of steganalysis is still limited, its success in other image-based tasks makes it a promising candidate for future studies.

When evaluating the performance of deep learning-based steganalysis models, two key metrics emerge: detection accuracy and computational efficiency. Accuracy metrics, including detection rate, false positives, and true positives, are critical for determining a model's effectiveness in correctly identifying hidden information. The BOSSbase-1.01 dataset, a benchmark dataset widely used in steganalysis research, serves as a standard for testing the performance of steganalysis models [31], [32]. In addition to accuracy, computational efficiency plays a crucial role in real-time or mobile applications. Factors like memory usage, latency, and power consumption are essential considerations for deploying steganalysis models on mobile devices.

Despite these advancements, significant gaps remain in the current literature, particularly regarding the optimization of steganalysis models for mobile environments. Most existing model, such as GoogleNet, are too resource-intensive for deployment on mobile

This is an open access article under the CC BY-4.0 license

devices or in real-time applications. While lightweight architectures like MobileNet, ShuffleNet, and EfficientNet offer promising solutions, their application to steganalysis has not been fully explored. Furthermore, balancing detection accuracy with computational efficiency continues to be a major challenge, particularly as steganographic methods evolve and become more sophisticated. The need for lightweight, real-time steganalysis models that can perform well in resource-constrained environments is increasingly pressing as mobile devices become more prevalent in security applications.

In conclusion, while deep learning has brought significant improvements to the field of steganalysis, much work remains, particularly in developing models that can operate efficiently in real-world, resource-constrained environments. Recent CNN models like GoogleNet demonstrated the power of deep learning for detecting hidden information but highlighted the need for more scalable solutions. Lightweight architectures like MobileNet, ShuffleNet, and EfficientNet provide a promising foundation for future research, offering the potential to develop efficient, accurate steganalysis models that can be deployed in real-time, mobile environments. This research seeks to bridge the gap in the literature by optimizing these lightweight models for steganalysis, with the goal of achieving high detection accuracy without overwhelming computational resources.

2. Research Methods

This section outlines the methodology used in this research, which focuses on the implementation of MobileNet, ShuffleNet, and EfficientNet models for steganalysis on the BOSSbase-1.01 dataset. The methodology covers key stages including data preprocessing, model training, cross-validation, evaluation, and model comparison based on accuracy, computational efficiency, and resource usage.

2.1 Proposed Deep Learning Models

The core of this research revolves around using state-of-the-art deep learning models—MobileNet, ShuffleNet, and EfficientNet—which are designed for efficient operation in resource-constrained environments, such as mobile devices.

These models were selected due to their ability to perform well on image classification tasks while maintaining low computational overhead. In order to achieve optimal performance, the models were trained and evaluated using consistent training parameters, ensuring a fair and robust comparison.

2.2 Research Workflow

The research workflow was structured around a series of systematic steps to ensure consistency, accuracy, and efficiency in the training and evaluation of the models. Figure 1 depicts research workflow.

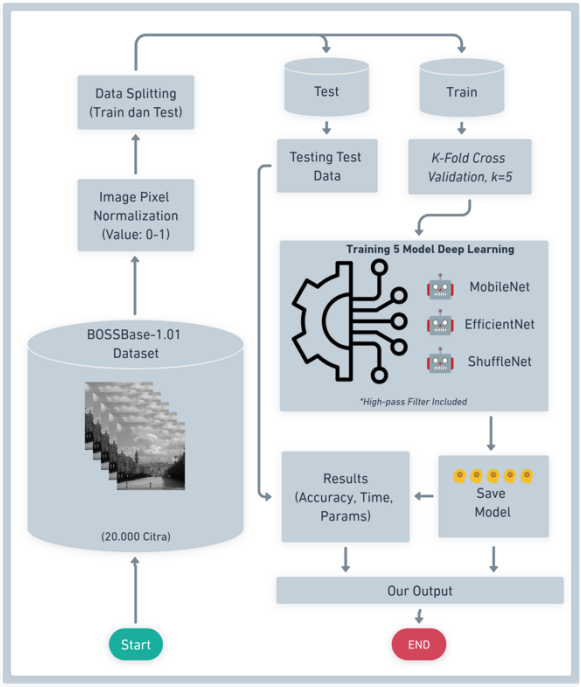

Figure 1. Research workflow

Data Preprocessing: The first step involved data preprocessing, where the pixel values of the images were normalized. Normalization scales the pixel values between 0 and 1, which facilitates faster and more stable computations during training. This step also helps prevent gradient overflow, a phenomenon where the model's gradients become excessively large, leading to unstable updates that can hinder the training process.
A high-pass filter (HPF) was applied to the input images to enhance high-frequency components, such as edges and subtle noise patterns, which are often indicative of hidden steganographic information. The HPF configuration uses a fixed 5×5 matrix as specified in [11], based on prior research demonstrating its effectiveness in image steganalysis. This matrix strengthens pixel intensity differences, especially at edges, amplifying high-frequency details that are critical for detecting the minor modifications associated with steganographic embedding.

The HPF matrix is defined as follows:

$$F = \frac{1}{12}\begin{bmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & -12 & 8 & -2 \\ 2 & -6 & 8 & -6 & 2 \\ -1 & 2 & -2 & 2 & -1 \end{bmatrix} \quad (1)$$

By using this fixed configuration, the filter enhances noise-like features consistently across all images. This choice of preprocessing technique follows established research that has shown improved detection accuracy when applying similar HPF configurations in steganalysis models [11]. Therefore, the HPF step was included as a proven method to assist the model in focusing on essential high-frequency features without requiring additional tuning during our experiments.

This is an open access article under the CC BY-4.0 license

Dataset Partitioning: The dataset was divided into two parts: 80% of the images were used for training, and 20% were reserved for testing. This partitioning ensured that the models were trained on the majority of the dataset while being evaluated on unseen data to assess their generalization capabilities. Additionally, a 5-fold cross-validation approach was applied to the training set, with 80% of the data used for training in each fold and 20% for validation. This cross-validation method helps improve the robustness of the models by evaluating them on multiple subsets of the data.

Model Training: During the training phase, the three models—MobileNet, ShuffleNet, and EfficientNet—were trained on the prepared training dataset. Training took place over 20 epochs with a batch size of 32, and the models were optimized using the Adam optimizer with exponential learning rate decay. The performance of the models was continuously monitored using the validation set to ensure that they were learning effectively and to detect potential overfitting. The cross-entropy loss function was used to guide the training process by penalizing incorrect predictions, helping the models refine their feature extraction capabilities.

Model Storage: After training, each model was saved for future use. Model storage is critical for deploying these models in real-world applications or for further testing phase. The stored models can be later retrieved for evaluation or integration into mobile applications designed for steganalysis.

Model Evaluation: The models were evaluated on the testing dataset, which was not used during training to provide an unbiased assessment of their performance on unseen data. This evaluation was crucial to assess the generalization capabilities of the models, especially their ability to detect stego images. The evaluation was based on several key metrics, including accuracy, time computation, and resource efficiency (measured in terms of the number of trainable parameters).

Prediction on Test Data: After training and evaluating the models, the final phase involved predicting the test data. Each model was tasked with classifying images as either cover or stego, and their predictions were compared against the ground truth to compute the accuracy. In addition, the time taken by each model to make predictions was recorded, as well as the resource usage in terms of trainable parameters. This evaluation process helped identify the best-performing model in terms of both accuracy and efficiency.

The research produced two primary outputs: A mobile-friendly deep learning model for stego image detection; A detailed evaluation report that provides insights into model performance, including accuracy, computational efficiency, and resource usage.

2.3. Models and Training Parameters

For model training, we adopted the Adam optimizer, a popular choice for deep learning models because of its adaptive learning rate and momentum features, which help the model converge faster and more efficiently. In addition, we implemented a learning rate schedule using an Exponential Decay function. The initial learning rate was set at 0.001, which is a standard starting point for training deep learning models. Over time, the learning rate decayed at a rate of 0.9 after every 10,000 steps. This decay mechanism allows the learning process to start with relatively large updates and gradually reduce them as the model approaches convergence, preventing overshooting and improving the overall stability of the training process.

The loss function used was Categorical Cross-Entropy, which is suitable for multi-class classification tasks like ours, where the objective is to correctly classify images as either "cover" (no hidden data) or "stego" (contains hidden data). Cross-entropy loss penalizes incorrect predictions more significantly, encouraging the model to learn more discriminative features. Training was conducted over 20 epochs, a reasonable number that balances between sufficient training iterations and avoiding overfitting. A batch size of 32 was chosen, ensuring efficient use of GPU memory without causing performance bottlenecks. The use of smaller batch sizes helps to update the model more frequently, promoting faster convergence and capturing finer details in the data.

The choice of models—MobileNet, ShuffleNet, and EfficientNet—was driven by their efficiency in handling high-dimensional image data while maintaining a low parameter count. MobileNet employs depthwise separable convolutions, drastically reducing the number of parameters and computational load compared to standard convolutional neural networks (CNNs). ShuffleNet further optimizes the architecture by using grouped convolutions and channel shuffling, which enhance efficiency without sacrificing accuracy. EfficientNet scales the depth, width, and resolution of the model systematically, achieving state-of-the-art accuracy with fewer parameters and lower computational requirements. These models were selected for their balance between performance and computational efficiency, making them suitable for real-time steganalysis on mobile devices.

2.4 Dataset

The BOSSbase-1.01 dataset, created using the S-UNIWARD algorithm with a 0.4 bpp embedding rate, was used for the steganalysis task. The dataset consists of 20,000 grayscale images, evenly divided between two labels: cover (no hidden data) and stego (contains hidden data). This dataset was originally designed for the BOSS competition and is available for access via GitHub. The dataset was partitioned into 80% for training and 20% for testing. Additionally, 5-fold cross-validation was applied to ensure robust model evaluation. Table 1 summarizes the dataset partitioning:

This is an open access article under the CC BY-4.0 license

Table 1. Software and supporting hardware

Label	Train Data (80%)	Test Data (20%)
Stego	9000	1000
Cover	9000	1000
Total	18000 (shuffled)	2000

K-Fold Cross Validation

K-Fold (k=5)	Train Data	Validation Data	Test Data
Fold 1-5	14400	3600	2000

2.5 Evaluation Metrics

The performance of the models was evaluated using three key metrics:

Accuracy: This is the primary metric for evaluating how well the models correctly classify images as cover or stego. It is the ratio of correctly predicted images to the total number of images in the testing set.

Computation Time: This metric measures the time taken by each model to make predictions. For real-time applications, such as mobile steganalysis, fast computation is crucial.

Resource Efficiency: This metric considers the number of trainable parameters in each model, which reflects the computational resources required. Lower parameter counts are favorable for deployment on resource-constrained devices like mobile phones.

ROC (Receiver Operating Characteristic) and AUC (Area Under the Curve): The ROC curve is a graphical representation of a model's ability to discriminate between the two classes—cover and stego. It plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings. The ROC curve gives insights into the trade-offs between sensitivity (ability to correctly classify stego images) and specificity (ability to correctly classify cover images).

By evaluating the models on these metrics—accuracy, ROC-AUC, computation time, and resource efficiency—the study aims to identify the best-performing model that not only achieves high detection accuracy but also operates efficiently on mobile or other resource-constrained platforms.

3. Results and Discussions

The goal of this research was to evaluate the performance of three lightweight deep learning models—MobileNet, ShuffleNet, and EfficientNet—for steganalysis on the BOSSbase-1.01 dataset. The evaluation focused on key metrics: accuracy, ROC-AUC, computational time, and resource efficiency (trainable parameters). Unfortunately, the results indicated that the models struggled to detect steganographic content effectively, performing close to random guessing in most cases. The discussion focus on testing phase which is the trained models predict testing data of 2000 images which not used in the training. This section discusses the experimental findings, the reasons for the low performance, and implications for future research.

When analyzing accuracy, as shown in Figure 2 as well as detailed in Table 2, MobileNet achieved the highest fold accuracy among the three models, reaching 63.8%. While this still falls short of acceptable performance for steganalysis, where accuracies typically exceed 90% [11], MobileNet's relatively higher accuracy can be attributed to its architectural design. Specifically, MobileNet's use of depthwise separable convolutions allows it to capture some level of spatial dependencies while significantly reducing the parameter count and computational demands compared to standard convolutional layers. This architectural feature provides MobileNet with a moderate capacity for distinguishing between cover and stego images, though it struggles to capture the fine-grained features necessary for high precision in steganalysis.

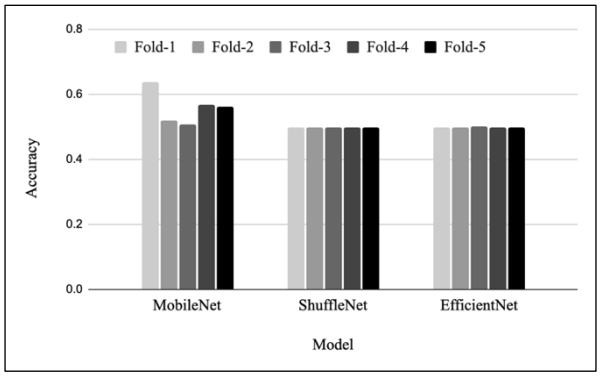

Figure 2. Plot of Accuracy of All Models in Testing Phase in All Fold

Table 2. Accuracy of all models in testing phase in all fold

Model	Fold 1	2	3	4	5
MobileNet	0.63	0.51	0.50	0.56	0.56
ShuffleNet	0.5	0.5	0.5	0.5	0.49
EfficienNet	0.49	0.5	0.50	0.49	0.5

The accuracy variation across folds, from 50.9% to 63.8%, suggests that while MobileNet's architecture is somewhat suited for feature extraction in steganalysis, it lacks consistency and the robustness required to generalize effectively. This fluctuation highlights MobileNet's limited capacity to reliably detect subtle, hidden patterns within the image data, which are critical for steganalysis tasks. Nonetheless, the modest success in certain folds indicates that MobileNet's lightweight design can extract some meaningful spatial information that aids in distinguishing between cover and stego images, albeit insufficiently for practical application.

In comparison in Figure 2, ShuffleNet consistently hovered around 50% accuracy across all folds, performing no better than random guessing. ShuffleNet's reliance on grouped convolutions and channel shuffling, which are intended to optimize efficiency, may restrict its ability to capture detailed spatial information necessary for steganalysis. The

This is an open access article under the CC BY-4.0 license

grouped convolutions limit channel-wise connectivity, which could impair ShuffleNet's capacity to identify subtle changes across the image, leading to an accuracy that reflects random guessing rather than true pattern recognition.

EfficientNet, despite its more advanced and scalable architecture, similarly recorded poor performance in Table 2, with its highest accuracy only reaching 50.0%. This outcome suggests that EfficientNet's complexity and parameter count did not translate into improved accuracy for this particular task. While EfficientNet achieves state-of-the-art performance in general image classification by scaling depth, width, and resolution, these enhancements did not benefit the steganalysis task. Its accuracy plateau, comparable to ShuffleNet, underscores that increased model complexity alone does not ensure better performance in tasks requiring nuanced feature extraction.

In summary, MobileNet's comparatively better accuracy reflects a slight advantage in feature extraction due to its use of depthwise separable convolutions, but all three models ultimately struggle to achieve the accuracy levels needed for reliable steganalysis. This outcome highlights a key limitation in applying general-purpose lightweight CNN architectures to steganalysis, where detecting subtle hidden patterns remains a significant challenge.

Table 3. Model performance on all models

Model	Time (seconds)	Trainable params (millions)
MobileNet	4.55	3.22
ShuffleNet	10.60	1.37
EfficientNet	10.98	4.02

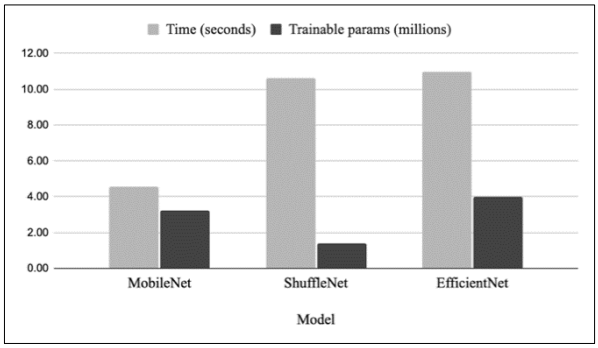

Figure 3. Plot of Model Performance on All Models

Table 3. shows the model's performance consist time computation and their trainable params used. In terms of computational efficiency, MobileNet was the fastest, completing predictions in 4.55 seconds, significantly outperforming ShuffleNet (10.60 seconds) and EfficientNet (10.98 seconds). This makes MobileNet a better candidate for real-time applications if accuracy can be improved. In terms of resource usage, ShuffleNet had the fewest trainable parameters (1.37 million), followed by MobileNet (3.22 million) and EfficientNet (4.02 million). However, despite ShuffleNet's efficiency in terms of parameters, its accuracy was suboptimal, suggesting that reducing the number of parameters alone is insufficient for effective steganalysis.

Table 4. ROC of all models in testing phase

Model	ROC Value
MobileNet	0.71
ShuffleNet	0.50
EfficientNet	0.50

The ROC-AUC score (Receiver Operating Characteristic - Area Under the Curve) is a crucial metric for evaluating the performance of classification models [33], particularly in binary tasks like steganalysis, where the goal is to distinguish between cover images (non-steganographic) and stego images (steganographic). The ROC curve, which plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various thresholds, helps assess model performance. TPR, or recall, measures the proportion of actual positives (stego images) correctly identified, while FPR indicates the proportion of actual negatives (cover images) misclassified as positives. The AUC score summarizes this performance across all thresholds, with a perfect classifier achieving a score of 1.0, random guessing scoring 0.5, and scores below 0.5 indicating a model performing worse than random.

Table 4 shows that ROC-AUC scores revealed significant limitations in the models' ability to differentiate between cover and stego images. Both ShuffleNet and EfficientNet recorded AUC scores of 0.50, indicating their performance was equivalent to random guessing and that they failed to extract meaningful features. MobileNet performed slightly better, with an AUC of 0.71, suggesting it could rank stego images higher than cover images 71% of the time. However, this still reflects substantial misclassification, falling short of practical application needs.

False positives—where cover images are incorrectly classified as stego images—severely impact a model's overall performance. Such misclassifications waste resources on unnecessary scrutiny and undermine trust in the system's reliability [34], which is particularly problematic in critical areas like digital forensics and cybersecurity, where precision is paramount [35]. A high false positive rate leads to inefficiencies, and models with low AUC scores are typically poorly calibrated, lacking the ability to effectively distinguish between positive (stego) and negative (cover) instances. This issue reflects deeper problems with the model's learning capacity rather than simply a threshold adjustment. In this study, the consistently low ROC-AUC scores across all models indicate an inability to generalize to unseen data, with the models struggling to capture the subtle pixel-level differences introduced by steganography. Even MobileNet, with an AUC of 0.71, exhibited high levels of both false positives and false negatives, limiting its practical use.

The real-world feasibility of these models is limited by their poor performance. While MobileNet demonstrated computational efficiency and could potentially be used

This is an open access article under the CC BY-4.0 license

for real-time steganalysis on mobile platforms, its accuracy (63.8%) remains too low for practical application. The models' inability to detect steganographic images with sufficient accuracy can be attributed to several factors.

The BOSSbase-1.01 dataset, created using the S-UNIWARD algorithm with a 0.4 bpp embedding rate, presents a significant challenge for lightweight architectures in steganalysis due to the subtle and imperceptible differences between cover and stego images. These grayscale images contain minute alterations that are difficult to detect, especially for models with limited capacity. The grayscale format itself reduces the overall information available for analysis, and the S-UNIWARD algorithm is designed to minimize visual distortions, further complicating the task for lightweight models. These models typically lack the deep feature extraction capabilities required to capture such fine-grained differences, especially when compared to more complex architectures that excel in detecting nuanced patterns in image data.

In addition, the dataset's size and diversity may be insufficient to adequately train these lightweight models. The BOSSbase-1.01 dataset, comprising 20,000 images evenly split between cover and stego classes, while reasonably sized, may not provide the model with enough variety to learn the intricate features necessary for successful steganalysis. The images are generated with the same embedding rate and algorithm, meaning the differences between images may be highly consistent, limiting the model's ability to generalize to unseen patterns.

Compared to more sophisticated and resource-intensive models like integrated GooleNet by Zhang 2022, which have achieved over 90% accuracy, these lightweight models clearly underperformed. While integrated GoogleNet offers superior performance, it requires significantly more computational power, highlighting a trade-off between computational efficiency and accuracy. The lightweight models tested in this study, while suitable for mobile deployment in terms of speed and memory usage, were not capable of achieving similar levels of accuracy.

Several challenges emerged during the study, including limitations in the models' architecture and the improper of advanced preprocessing techniques. The models' design, optimized for general image classification, may not have been well-suited to the specific challenges of steganalysis, where hidden data introduces minimal pixel-level changes. Additional preprocessing steps, such as choosing high-pass filtering properly or noise augmentation, could have enhanced the models' ability to detect hidden patterns. Moreover, the relatively small dataset may have contributed to the models' poor generalization, as more diverse training data could help the models learn to detect steganography more effectively.

In this study, balancing computational efficiency with detection accuracy, as each model presented unique trade-offs in resource usage that impacted its feasibility for practical steganalysis. MobileNet, for instance, achieved the highest computational efficiency, completing predictions in 4.55 seconds with only 3.22 million trainable parameters, making it well-suited for applications on resource-limited devices. However, its accuracy (63.8%) still fell significantly below the practical requirements for reliable steganalysis. Although ShuffleNet demonstrated the lowest memory demand (1.37 million parameters), its performance was equivalent to random guessing, showing that memory efficiency alone does not ensure practical utility in steganalysis.

For steganalysis tasks, which require detecting subtle pixel-level changes, models typically demand both high accuracy and extensive computational resources. However, resource limitations, especially in mobile or edge devices, constrain the complexity of models that can be deployed effectively. While resource-efficient architectures like MobileNet may offer the potential for real-time applications in resource-limited environments, this study underscores the need for an optimized balance where lightweight models can improve detection accuracy without significantly increasing resource requirements.

Moreover, steganalysis, by its nature, requires the ability to identify subtle and often imperceptible changes at the pixel level [27], [36]. Lightweight models such as MobileNet, ShuffleNet, and EfficientNet are optimized for computational efficiency, which involves trade-offs in their capacity for deep feature extraction. While these architectures excel at traditional image classification tasks—where larger, more distinguishable features like shapes, textures, and colors are important [15], [16], [20]—they are not inherently designed to detect the minute, low-level changes introduced by steganography. To reduce computational costs, these lightweight models employ fewer and shallower convolutional layers, limiting their ability to capture the deep, nuanced features needed for steganalysis. As a result, the relatively shallow architectures of MobileNet, ShuffleNet, and EfficientNet may not be as effective for tasks requiring deeper models capable of extracting fine-grained details across multiple layers of abstraction, which is critical for detecting steganographic content.

Future research should aim to address these limitations through several key areas. First, exploring advanced preprocessing techniques such as data augmentation, noise reduction, and feature extraction could lead to more robust and effective steganalysis models. Fine-tuning lightweight models with these improved preprocessing methods may significantly enhance performance. Second, using larger and more diverse datasets that encompass a wider range of steganographic techniques and image types would likely improve the models' ability to generalize across

This is an open access article under the CC BY-4.0 license

different contexts. Finally, incorporating ensemble methods like [11], which combine the predictive power of multiple lightweight models, could lead to better detection accuracy. These ensemble approaches can leverage the strengths of individual models while maintaining computational efficiency, offering a promising avenue for improved performance in steganalysis tasks.

In conclusion, while MobileNet demonstrated computational efficiency, none of the models achieved satisfactory accuracy for reliable steganalysis. The results suggest that current lightweight models, as implemented in this research, are not well-suited for the detection of steganographic images, especially when the steganographic method is highly sophisticated. Future work should explore model adaptations and advanced techniques to improve the performance of these models in resource-constrained environments.

4. Conclusions

This research aimed to evaluate the effectiveness of lightweight deep learning models—MobileNet, ShuffleNet, and EfficientNet—for real-time steganalysis in resource-constrained environments, particularly on mobile devices. The study focused on key performance metrics, including accuracy, computational time, resource efficiency, and the ability to detect hidden information within digital images. The results, however, highlighted significant limitations in the models' performance, suggesting that these architectures are not yet suitable for practical steganalysis applications in their current form. Among the models tested, MobileNet demonstrated the best computational efficiency, with faster inference times and fewer trainable parameters compared to ShuffleNet and EfficientNet. This makes it a promising candidate for real-time use, especially in environments where computational resources are limited, such as mobile platforms. However, despite its efficiency, MobileNet's detection accuracy remained suboptimal, with a highest accuracy of just 63.8%. ShuffleNet and EfficientNet performed even worse, hovering around 50%, effectively reducing their performance to the level of random guessing. The ROC-AUC results further underscored this issue, with both ShuffleNet and EfficientNet achieving scores of 0.50, indicating an inability to differentiate between steganographic and cover images. The underperformance of these models is likely attributable to several factors. First, the BOSSbase-1.01 dataset, which contains grayscale images with subtle pixel-level modifications, presents a significant challenge for lightweight models that are optimized for general image classification tasks rather than the nuanced detection of hidden data. Additionally, the S-UNIWARD steganographic algorithm, which minimizes perceptual differences between cover and stego images, likely made it difficult for the models to extract meaningful features necessary for detection. The limited size of the dataset and the properly choosing advanced preprocessing techniques, such as high-pass filtering, may have further hindered the models' ability to generalize well. While MobileNet and other lightweight architectures hold potential due to their low computational cost and efficiency, this research suggests that current implementations fall short in steganalysis tasks, particularly when faced with sophisticated steganographic techniques. The need for more powerful feature extraction methods, possibly through advanced preprocessing or more refined model architectures, is evident. Additionally, larger, more diverse datasets and the incorporation of ensemble methods could improve the models' accuracy without significantly increasing computational demands. In conclusion, while this study explored the feasibility of deploying lightweight deep learning models for steganalysis on mobile platforms, the results indicate that further optimization is required to achieve acceptable performance. To improve the accuracy of lightweight deep learning models for steganalysis in addition of future research previously, several actionable recommendations can be implemented. First, tweaking the model architecture by integrating attention mechanisms, such as Squeeze-and-Excitation blocks or self-attention layers, could help the models focus on subtle pixel-level variations [37], [38]. Hybrid architectures that combine CNNs with Graph Neural Networks (GNNs) or Transformer layers could further improve performance by capturing complex spatial relationships [39], [40]. In refining the training process, data augmentation using image rotation, noise injection, and contrast adjustments can improve model generalization, while transfer learning from large datasets (such as ImageNet) can enhance feature extraction [41], [42], [43], [44]. Additionally, hyperparameter tuning through methods like Bayesian Optimization or Grid Search can optimize model performance [45].

Acknowledgements

This research was funded by the Ministry of Education, Culture, Research, and Technology, Republic of Indonesia, under the Master Thesis Research Grant number 107/E5/PG.02.00.PL/2024, 0609.1/LL5-INT/AL.04/2024, 062/DirDPPM/70/DPPM/PTM-KEMDIKBUDRISTEK/VI/2024.

References

[1] S. Kemp, "Digital 2023: Global Overview Report," DataReportal – Global Digital Insights. Accessed: Sep. 28, 2024. [Online]. Available: https://datareportal.com/reports/digital-2023-global-overview-report

[2] entrust, "2022 Global Encryption Trends Study | Entrust." Accessed: Sep. 28, 2024. [Online]. Available: https://www.entrust.com/resources/reports/global-encryption-trends-study

[3] P. Ponemon, "2024 State of Zero Trust & Encryption Study," 2024.

[4] G. Li, S. Li, M. Li, X. Zhang, and Z. Qian, "Steganography of Steganographic Networks," arXiv.org. Accessed: Sep. 28, 2024. [Online]. Available: https://arxiv.org/abs/2302.14521v1

This is an open access article under the CC BY-4.0 license

[5] McAfee, "2021 Threat Predictions Report," McAfee Blog. Accessed: Sep. 28, 2024. [Online]. Available: https://www.mcafee.com/blogs/other-blogs/mcafee-labs/2021-threat-predictions-report/

[6] S. Kaur, S. Singh, M. Kaur, and H.-N. Lee, "A Systematic Review of Computational Image Steganography Approaches," *Arch Computat Methods Eng*, vol. 29, no. 7, pp. 4775–4797, Nov. 2022, doi: 10.1007/s11831-022-09749-0.

[7] M. Broz, "How many photos are there? (Statistics & trends in 2024)." Accessed: Oct. 14, 2024. [Online]. Available: https://photutorial.com/photos-statistics/

[8] S. Agarwal, H. Kim, and K.-H. Jung, "High-Pass-Kernel-Driven Content-Adaptive Image Steganalysis Using Deep Learning," *Mathematics*, vol. 11, no. 20, Art. no. 20, Jan. 2023, doi: 10.3390/math11204322.

[9] L. I. Hao, Z. Yi, W. Jinwei, Z. Weiming, and L. U. O. Xiangyang, "Lightweight Steganography Detection Method Based on Multiple Residual Structures and Transformer," *dzxbyw*, vol. 33, no. 4, pp. 965–978, Jul. 2024, doi: 10.23919/cje.2022.00.452.

[10] L. Liu, L. Meng, X. Wang, and Y. Peng, "An image steganography scheme based on ResNet," *Multimed Tools Appl*, vol. 81, no. 27, pp. 39803–39820, Nov. 2022, doi: 10.1007/s11042-022-13206-2.

[11] Y. Zhang, "Image steganalysis method based on integrated GoogleNet," in *2022 IEEE Conference on Telecommunications, Optics and Computer Science (TOCS)*, Dalian, China: IEEE, Dec. 2022, pp. 1–4. doi: 10.1109/TOCS56154.2022.10015953.

[12] Y. Yousfi, J. Butora, E. Khvedchenya, and J. Fridrich, "ImageNet Pre-trained CNNs for JPEG Steganalysis," in *2020 IEEE International Workshop on Information Forensics and Security (WIFS)*, Dec. 2020, pp. 1–6. doi: 10.1109/WIFS49906.2020.9360897.

[13] W. Mazurczyk and L. Caviglione, "Steganography in Modern Smartphones and Mitigation Techniques," Aug. 27, 2014, *arXiv*: arXiv:1410.6796. Accessed: Nov. 08, 2024. [Online]. Available: http://arxiv.org/abs/1410.6796

[14] N. Chen and B. Chen, "Defending against OS-Level Malware in Mobile Devices via Real-Time Malware Detection and Storage Restoration," *Journal of Cybersecurity and Privacy*, vol. 2, no. 2, Art. no. 2, Jun. 2022, doi: 10.3390/jcp2020017.

[15] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," Apr. 16, 2017, *arXiv*: arXiv:1704.04861. Accessed: Dec. 18, 2023. [Online]. Available: http://arxiv.org/abs/1704.04861

[16] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices," Dec. 07, 2017, *arXiv*: arXiv:1707.01083. Accessed: Mar. 20, 2024. [Online]. Available: http://arxiv.org/abs/1707.01083

[17] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," Sep. 11, 2020, *arXiv*: arXiv:1905.11946. doi: 10.48550/arXiv.1905.11946.

[18] T. Paul, S. Ghosh, and A. Majumder, "A Study and Review on Image Steganography," in *Computer Networks and Inventive Communication Technologies*, S. Smys, R. Bestak, R. Palanisamy, and I. Kotuliak, Eds., Singapore: Springer Nature, 2022, pp. 523–531. doi: 10.1007/978-981-16-3728-5_40.

[19] M. A. Aslam *et al.*, "Image Steganography using Least Significant Bit (LSB) - A Systematic Literature Review," in *2022 2nd International Conference on Computing and Information Technology (ICCIT)*, Jan. 2022, pp. 32–38. doi: 10.1109/ICCIT52419.2022.9711628.

[20] D. Nashat and L. Mamdouh, "An efficient steganographic technique for hiding data," *Journal of the Egyptian Mathematical Society*, vol. 27, no. 1, p. 57, Dec. 2019, doi: 10.1186/s42787-019-0061-6.

[21] R. Chandramouli, M. Kharrazi, and N. Memon, "Image Steganography and Steganalysis: Concepts and Practice," 2004.

[22] R. A, R. Agustina, and Hidayatulloh, "Comparison of Discrete Cosine Transforms (DCT), Discrete Fourier Transforms (DFT), and Discrete Wavelet Transforms (DWT) in Digital Image Watermarking," *ijacsa*, vol. 8, no. 2, 2017, doi: 10.14569/IJACSA.2017.080232.

[23] K. Choudhary, "Image Steganography and Global Terrorism," *IOSRJCE*, vol. 1, no. 2, pp. 34–48, 2012, doi: 10.9790/0661-0123448.

[24] H. Kheddar, M. Hemis, Y. Himeur, D. Megías, and A. Amira, "Deep Learning for Steganalysis of Diverse Data Types: A review of methods, taxonomy, challenges and future directions," *Neurocomputing*, vol. 581, p. 127528, May 2024, doi: 10.1016/j.neucom.2024.127528.

[25] A. K. Sahu and M. Sahu, "Digital image steganography and steganalysis: A journey of the past three decades," *Open Computer Science*, vol. 10, no. 1, pp. 296–342, Jan. 2020, doi: 10.1515/comp-2020-0136.

[26] J. Fridrich and M. Goljan, "On estimation of secret message length in LSB steganography in spatial domain," presented at the Electronic Imaging 2004, E. J. Delp Iii and P. W. Wong, Eds., San Jose, CA, Jun. 2004, p. 23. doi: 10.1117/12.521350.

[27] T. Pevný, P. Bas, and J. Fridrich, "Steganalysis by Subtractive Pixel Adjacency Matrix," 2009.

[28] A. D. Ker and T. Pevný, "A new paradigm for steganalysis via clustering," presented at the IS&T/SPIE Electronic Imaging, N. D. Memon, J. Dittmann, A. M. Alattar, and E. J. Delp Iii, Eds., San Francisco Airport, California, USA, Feb. 2011, p. 78800U. doi: 10.1117/12.872888.

[29] X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, and M. Parmar, "A review of convolutional neural networks in computer vision," *Artif Intell Rev*, vol. 57, no. 4, p. 99, Mar. 2024, doi: 10.1007/s10462-024-10721-6.

[30] M. Boroumand, M. Chen, and J. Fridrich, "Deep Residual Network for Steganalysis of Digital Images," *IEEE Trans.Inform.Forensic Secur.*, vol. 14, no. 5, pp. 1181–1193, May 2019, doi: 10.1109/TIFS.2018.2871749.

[31] Y. Ge, T. Zhang, H. Liang, Q. Jiang, and D. Wang, "A Novel Technique for Image Steganalysis Based on Separable Convolution and Adversarial Mechanism," *Electronics*, vol. 10, no. 22, p. 2742, Nov. 2021, doi: 10.3390/electronics10222742.

[32] G. Xu, H.-Z. Wu, and Y.-Q. Shi, "Structural Design of Convolutional Neural Networks for Steganalysis," *IEEE Signal Process. Lett.*, vol. 23, no. 5, pp. 708–712, May 2016, doi: 10.1109/LSP.2016.2548421.

[33] F. Melo, "Area under the ROC Curve," in *Encyclopedia of Systems Biology*, W. Dubitzky, O. Wolkenhauer, K.-H. Cho, and H. Yokota, Eds., New York, NY: Springer, 2013, pp. 38–39. doi: 10.1007/978-1-4419-9863-7_209.

[34] S. Guznov, J. Lyons, A. Nelson, and M. Woolley, "The Effects of Automation Error Types on Operators' Trust and Reliance," in *Virtual, Augmented and Mixed Reality*, S. Lackey and R. Shumaker, Eds., Cham: Springer International Publishing, 2016, pp. 116–124. doi: 10.1007/978-3-319-39907-2_11.

[35] J. R. Lyle, B. Guttman, J. M. Butler, K. Sauerwein, C. Reed, and C. E. Lloyd, "Digital investigation techniques : a NIST scientific foundation review," National Institute of Standards and Technology (U.S.), Gaithersburg, MD, NIST IR 8354, Nov. 2022. doi: 10.6028/NIST.IR.8354.

[36] W.-B. Lin, T.-H. Lai, and K.-C. Chang, "Statistical feature-based steganalysis for pixel-value differencing steganography," *EURASIP Journal on Advances in Signal Processing*, vol. 2021, no. 1, p. 87, Sep. 2021, doi: 10.1186/s13634-021-00797-5.

[37] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-Excitation Networks," May 16, 2019, *arXiv*: arXiv:1709.01507. doi: 10.48550/arXiv.1709.01507.

[38] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, "Stand-Alone Self-Attention in Vision Models," Jun. 13, 2019, *arXiv*: arXiv:1906.05909. doi: 10.48550/arXiv.1906.05909.

[39] A. Khan *et al.*, "A survey of the vision transformers and their CNN-transformer based variants," *Artif Intell Rev*, vol. 56, no. 3, pp. 2917–2970, Dec. 2023, doi: 10.1007/s10462-023-10595-0.

[40] Z. Jiao, H. Zhang, and X. Li, "CNN2GNN: How to Bridge CNN with GNN," Apr. 23, 2024, *arXiv*: arXiv:2404.14822. doi: 10.48550/arXiv.2404.14822.

[41] F. M. Quiroga, F. Ronchetti, L. Lanzarini, and A. Fernandez-Bariviera, "Revisiting Data Augmentation for Rotational

This is an open access article under the CC BY-4.0 license

Invariance in Convolutional Neural Networks," Oct. 12, 2023, *arXiv*: arXiv:2310.08429. doi: 10.48550/arXiv.2310.08429.

[42] O. Dhifallah and Y. M. Lu, "On the Inherent Regularization Effects of Noise Injection During Training," Feb. 15, 2021, *arXiv*: arXiv:2102.07379. doi: 10.48550/arXiv.2102.07379.

[43] T. Kumar, A. Mileo, R. Brennan, and M. Bendechache, "Image Data Augmentation Approaches: A Comprehensive Survey and Future directions," Mar. 12, 2023, *arXiv*: arXiv:2301.02830. doi: 10.48550/arXiv.2301.02830.

[44] S. Kornblith, J. Shlens, and Q. V. Le, "Do Better ImageNet Models Transfer Better?," Jun. 17, 2019, *arXiv*: arXiv:1805.08974. doi: 10.48550/arXiv.1805.08974.

[45] C. Arnold, L. Biedebach, A. Küpfer, and M. Neunhoeffer, "The role of hyperparameters in machine learning models and how to tune them," *Political Science Research and Methods*, vol. 12, no. 4, pp. 841–848, Oct. 2024, doi: 10.1017/psrm.2023.61.

This is an open access article under the CC BY-4.0 license