

Article

NFT Image Plagiarism Check Using EfficientNet-Based Deep Neural Network with Triplet Semi-Hard Loss

Aji Teguh Prihatno ¹, Naufal Suryanto ¹, Sangbong Oh ¹, Thi-Thu-Huong Le ^{2,3,*} and Howon Kim ^{1,*}¹ School of Computer Science and Engineering, Pusan National University, Busan 609735, Republic of Korea² Blockchain Platform Research Center, Pusan National University, Busan 609735, Republic of Korea³ IoT Research Center, Pusan National University, Busan 609735, Republic of Korea

* Correspondence: lehuong7885@gmail.com (T.-T.-H.L.); howonkim@pusan.ac.kr (H.K.)

Abstract: Blockchain technology is used to support digital assets such as cryptocurrencies and tokens. Commonly, smart contracts are used to generate tokens on top of the blockchain network. There are two fundamental types of tokens: fungible and non-fungible (NFTs). This paper focuses on NFTs and offers a technique to spot plagiarism in NFT images. NFTs are information that is appended to files to produce distinctive signatures. It can be found in image files, real artifacts, literature published online, and various other digital media. Plagiarism and fraudulent NFT images are becoming a big concern for artists and customers. This paper proposes an efficient deep learning-based approach for NFT image plagiarism detection using the EfficientNet-B0 architecture and the Triplet Semi-Hard Loss function. We trained our model using a dataset of NFT images and evaluated its performance using several metrics, including loss and accuracy. The results showed that the EfficientNet-B0-based deep neural network with triplet semi-hard loss outperformed other models such as Resnet50, DenseNet, and MobileNetV2 in detecting plagiarized NFTs. The experimental results demonstrate sufficient to be implemented in various NFT marketplaces.

Keywords: NFT; blockchain; plagiarism check; EfficientNet-B0; deep neural network; triplet semi-hard loss



Citation: Prihatno, A.T.; Suryanto, N.; Oh, S.; Le, T.-T.-H.; Kim, H. NFT Image Plagiarism Check Using EfficientNet-Based Deep Neural Network with Triplet Semi-Hard Loss. *Appl. Sci.* **2023**, *13*, 3072. <https://doi.org/10.3390/app13053072>

Academic Editor: Andrea Prati

Received: 3 February 2023

Revised: 24 February 2023

Accepted: 24 February 2023

Published: 27 February 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Blockchain technology is avant-garde for digital entities. NFTs allow for the open transfer of ownership and the provable scarcity of digital goods. These new characteristics have enormous power for innovators. They can sell one-of-a-kind and authenticated things on a blockchain-based marketplace rather than spreading their works of art, such as images, music, or other creations, on platforms that are typically difficult to monetize. By relying on the irrefutable blockchain history of NFTs rather than the reseller's word, consumers may be confident in the legitimacy of any digital commodity they purchase [1].

An NFT is a digital asset that cannot be exchanged for another NFT as each token has a unique value, confirmed through data sets such as blockchain, token ID, token type, metadata, and contact address. NFTs can take the form of rare and unique images, movies, video game items, and artistic works [2]. NFTs controlled by a blockchain network indicate ownership of physical or digital items. The Ethereum network is currently one of the most widely utilized blockchain networks. On this network, people can deploy smart contracts and mint NFTs, allowing them to run Ethereum network apps. Since this network has been supported and regulated by a community, they do not need any devotion in any particular person to utilize it, provided the community has enough members. [3].

Although the highlighted advantages of NFT, NFT forgery and counterfeiting [4] have increasingly become a problem in the industry. For example, more than 80% of the NFTs are found in OpenSea, one of the NFT marketplaces. These recently made through its free minting tool were either spam, fake supplies, or plagiarized creations [5]. To overcome this

issue, this paper proposes a method to detect plagiarism in NFT marketplaces by enhancing deep neural networks.

Few research articles describe how to check NFT images for plagiarism. Pungila et al. [6] have developed a technique for approximation design matching for NFT image plagiarism detection, with special relevance to blockchain-driven NFT platforms and ecosystems. The authors used a non-deterministic finite automaton (N DFA) approach in conjunction with a sliding window idea and local thresholds at the node level to trace partial matches. This technique is comparable to other similarity measures currently used in text mining for plagiarism detection. On the other hand, numerous methods have been developed to identify images that have been plagiarized, such as Ibrahim et al. [7] who suggested an improved approach to identify image plagiarism using RGB (red, green, and blue) and HSV (hue, saturation, and value) color spaces, using Tamura texture, and the clever edge technique for shape to retrieve features from photos and save to databases. The results were displayed with true and false statements and similarity indices in ascending order. However, the precision of the modified images is low grade.

Gayadhankar et al. [8] have proposed a plagiarism detection method based on GAN and employ CNN, primarily used in the preprocessing stage. In this method, the dataset is first analyzed by the GAN, and then its output is fed into the CNN. The model will then indicate whether or not a GAN created the image. Nevertheless, the authors should have explained how they achieved the results in more detail, and there was no comparison to other methods. Furthermore, deep convolutional neural networks, such as those used by Meuschke et al. [9], can recognize several types of image similarity observed in academic work and may identify plagiarism in images. The adaptability of the technique is achieved by incorporating approaches for assessing various image features, selectively deploying analysis techniques based on how well they complete the input image and applying an adaptable strategy for recognizing suspicious image similarities. However, images resembling one another do not appear as suspicious outliers in noticeably bigger data sets.

Based on the aforementioned related works, several attempts have been made to address the problem of NFT image plagiarism using artificial intelligence methods. However, many of these methods have depended on traditional deep learning architectures such as convolutional neural networks (CNN). They have yet to include more recent advancements in computer vision, such as EfficientNet. The proposed work in this paper overcomes this limitation using an EfficientNet-based model that has demonstrated state-of-the-art performance to detect plagiarized images in the NFT system. In addition, training the model with the triple semi-hard loss function has enhanced the model's ability to distinguish between images with similar features. The proposed work offers a novel method for detecting NFT image plagiarism that incorporates recent computer vision and deep learning developments. While further testing and evaluation are needed, the proposed method has the potential to outperform existing methods and enhance the security and integrity of NFT-based art.

This study proposes an EfficientNet-based approach using the EfficientNet-B0 architecture to detect plagiarism in NFT images. Our specific contributions include the following:

- Preprocessing the NFT dataset and using it to generate augmented images.
- Detecting plagiarized NFT images based on EfficientNet-B0-based deep neural network (EfficientNet-B0-DNN) with triplet semi-hard loss.
- Developing a network suitable for actual NFT ecosystems and providing high accuracy and reliable performance.
- Comparing several modern deep learning methods and pointing out the EfficientNet-B0-DNN method outperforms other CNN methods such as ResNet50, DenseNet, and MobileNetV2 in terms of loss and accuracy.

The remainder of the study is organized as follows: in Section 2, we introduce the concept of NFTs, and the proposed solution scheme, detail the augmentation process for the dataset, and describe the NFT image dataset used in the research. Section 3 explains

ResNet50, DenseNet, MobileNetV2, and EfficientNet-B0-DNN techniques that we compared. Section 4 presents the hyperparameter settings we used, the Euclidean distance and triplet semi-hard loss as performance criteria, results, and a comparison of our proposed method with other models. Finally, in Section 5, we summarize the key findings, draw conclusions, and suggest areas for future research.

2. System Overview

2.1. NFTs Concept

NFTs are cryptographic tokens originally developed on the Ethereum blockchain and later adopted by other blockchains. NFTs are distinguishable from fungible tokens, such as bitcoins, where each token is equivalent and imperceptible. Because NFTs are non-fungible, each token is guaranteed to be distinctive, conferring right privileges over a digital treasure to its bearer [10]. Specifically, producers can easily prove ownership of their digital goods, such as artwork, images, and films, by using NFTs. Moreover, by definition, each non-fungible token (NFT) is distinctive and non-divisible; there should be an infinite number of NFT types. Academics generally categorize NFTs into six main groups based on their common applications: art, metaverse, gaming, collectibles, utility, and others [11].

In addition, Shilina et al. [12] explained the NFT as a token, as a unit of account, is a form in a dispersed blockchain controlled by a computer algorithm of a smart contract, in which the amount of the equity on the accounts of token holders are documented, allowing them to be transferred from one wallet to another. The idea of the non-fungible token was first introduced in the Ethereum Improvement Proposal (EIP) 721 [13] and further developed in EIP-1155 [14,15]. ERC-721 is a non-fungible token standard, which is different from fungible tokens. It can be unique because every token has a uint256 variable called `tokenId` that is globally distinctive. The ERC-721 standard includes basic capabilities such as `transferFrom` and `ownerOf` for tracking and transmitting NFTs in its smart contract API. ERC-1155 is a standard extension that can represent both non-fungible and fungible tokens. It delivers a representational interface for many tokens. Algorithms 1 and 2 demonstrate standard interfaces for ERC-721 and ERC-1155, respectively.

Algorithm 1: ERC-721 Standard Interface

```
interface ERC721 {
    function transferFrom(address _from, address _to, uint256 _tokenId) external payable ;
    function ownerOf(uint256 _tokenId) external view returns (address) ;
    function balanceOf(address _owner) external view returns (uint256) ;
    ...
}
```

Algorithm 2: ERC-1155 Standard Interface

```
interface ERC1155 {
    event TransferSingle(address indexed _operator, address indexed _from, address indexed _to, uint256 _id, uint256 _value) ;
    event TransferBatch(address indexed _operator, address indexed _from, address indexed _to, uint256[] _ids, uint256[] _values) ;
    function balanceOfBatch(address[] calldata _owners, uint256[] calldata _ids) external view returns (uint256[] memory) ;
    ...
}
```

There are some differences between ERC-721 and ERC-1155. To begin with, each ERC-721 transaction requires a single operation. Meanwhile, with ERC-1155, a single transaction can contain several operations. Second, we may employ the ERC-1155 token

standard as a vital infrastructure to enable batch token transfers. According to the ERC 721 token standard, a separate smart contract must be created for each NFT or token transfer. To transmit many NFTs through an ERC-721 system, transactions for each one needed to be created. Third, when assets are transferred via the ERC-721 standard to the incorrect address, they cannot be reverted. On the other hand, the ERC-1155 token standard provides a special function called the “safe transfer” function [16].

2.2. Proposed Solution Scheme

The proposed design architecture for the plagiarism detector on the NFT system is depicted in Figure 1. It is based on the lack of a built-in method on the blockchain to confirm that the person minting an NFT has the right to the asset they are minting [17]. Therefore, we propose a system to identify plagiarized NFT images as an AI model placed before the minting process to verify that only original and legally obtained images are being minted as NFTs. We employ an AI model to check for plagiarism before minting, which is expected to reduce computing costs because the model only needs to be run once for each image. Additionally, it can decrease the need for numerous copies of the same image to be stored, which can save cost and storage space. This study analyzes and compares the processing times for all models mentioned earlier.

This paper outlines five requirements for using blockchain technology in the NFT system. First, choosing the right blockchain platform, such as Ethereum, due to its scalability, security, and transaction speed [18]. Second, developing a smart contract with a token can be assigned to each individual work of art. Thus, the owner of the NFT has blockchain-based evidence of the asset’s authenticity. This is crucial because there have been instances in the past where individuals have attempted to manipulate already-sold NFTs [19]. Third, building interoperability to ensure that the NFTs created on the blockchain are interoperable, which refers to the ability to exchange data between different blockchain platforms and networks [20]. Fourth, deploying IPFS to cover storage and security, since we need to store the NFT image securely and immutably on the blockchain [21] [22]. Due to our system being intended to allow NFT-based art, it is crucial to take into account the supporting technologies. The usage of IPFS in our study indicates how we took the NFT system into account and worked to address the issue of NFT picture plagiarism as a whole. IPFS can be used as a decentralized storage system that ensures the NFTs cannot be altered or deleted by anybody, including the creator of the image [23]. Fifth, integrating with AI, which is necessary to ensure that the AI model for plagiarism checking is appropriately integrated with the blockchain platform so that it can interact with the NFTs and verify their originality.

The proposed system has two flows: one from creators who upload NFT images, represented by the red line, and one from users who download or purchase NFT images, represented by the blue line, as shown in Figure 1. The proposed solution addresses the issues of combining all anti-plagiarism-related tasks into a single procedure and ensuring security and privacy protection. A decentralized app with all the capabilities and services, including anti-plagiarism and advanced application programming interface (API), is included in the proposed solution.

One of the best decentralized options for storing NFT files is IPFS (InterPlanetary File System), a distributed system for uploading, storing, and accessing websites, programs, data, and files. IPFS offers a decentralized way to host and access content as a P2P file-sharing protocol, with user operators hosting a portion of the total data, resulting in a unique and cutting-edge system for storing and distributing files or other content. Unlike the conventional hypertext transfer protocol (HTTP) protocol, IPFS uses a content-addressing method. Each content within the IPFS ecosystem has a unique hash that acts as a content identification (CID). As a result, users of IPFS can find any file, website, and other data by searching for the relevant cryptographic hash instead of searching by location [24].

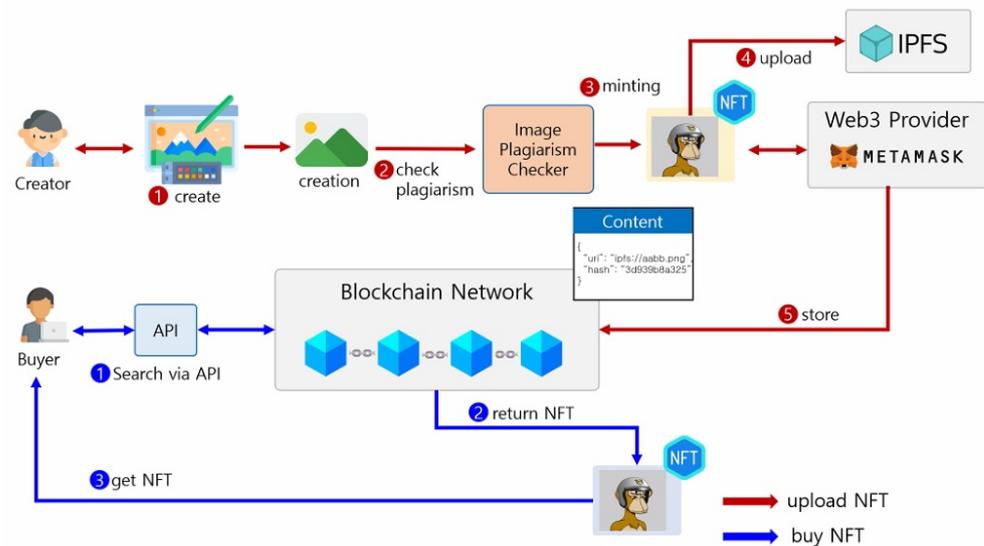


Figure 1. Proposed solution scheme of NFT plagiarism checker.

We have developed a novel strategy that uses current computer vision and deep learning developments, including the EfficientNet-based model with the triple semi-hard loss function. Although our system does not employ specific NFT-related features, it can be implemented in any NFT system or marketplace. Our method may effectively identify NFT image plagiarism. Furthermore, because of the nature of blockchain-based ownership and transfer, our proposed approach was designed and optimized for the NFT system, which presents a unique environment for picture plagiarism. According to the findings of our experiments, the proposed method can successfully improve the security and integrity of NFT-based art by precisely identifying plagiarized images in this study.

2.3. NFTs Image Dataset

In this work, a publicly available NFT-Classifer dataset from Kaggle was used [25]. The dataset includes three of the most popular NFTs collections available, known as art categories BAYC (Bored Ape Yacht Club), Crypto Punks, and Azuki. A total of 3000 photos in various sizes and aspect ratios are included. The 24×24 pixel size and 2000×2000 pixel size, respectively, make up the smallest and largest images. There are an equal number of images in each collection category in the dataset utilized in this study. This harmony facilitates model training by preventing the model from becoming biased toward a certain class [26]. Furthermore, the dataset that NFT-inspired works of art in an image file format (.png). Only the image files were used in this work. The dataset was divided into 70% training and 30% validation portions, with all images resized to a size of 224 by 224. Figure 2 shows sample images of the NFT dataset used in this paper.

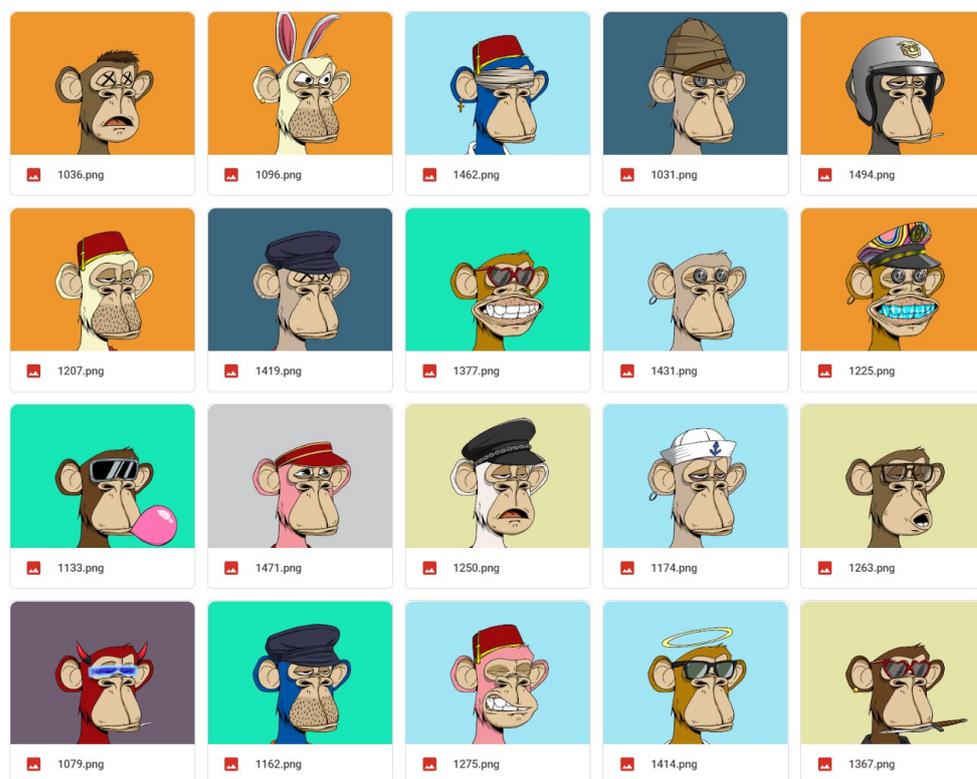


Figure 2. Sample images of NFT dataset [25].

To train the model using NFT images, we first collect a dataset of NFT images from the above open-access resources along with their associated metadata, including file size, format, and structure. Next, the images are preprocessed by resizing them to a fixed size of 224 by 224 and standardizing their pixel values. For the model architecture, we employ four different models: ResNet50, DenseNet, MobileNetV2, and EfficientNetB0. Each model is trained separately using the preprocessed images and their associated metadata. During training, we use a triplet semi-hard loss function for all models, which encourages the model to differentiate between similar and dissimilar NFT images. The compared models are trained using the Adam optimizer with a learning rate of 0.001 and a batch size of 64 over 100 epochs, along with other hyperparameters mentioned in Table 1.

2.4. Image Augmentation

Image augmentation is the process of modifying an image to generate variants of the same matter in order to give the model a wider range of training examples. Due to the impossibility of precisely capturing every possible real-world scenario, augmentation is essential. By increasing the image collection, we can incorporate additional challenging-to-discover real-world scenarios and increase the training data sample size. By expanding the training data to generalize to many scenarios, the model can acquire knowledge from a broader range of events [27]. In this study, we randomly change an input image's rotation, brightness, shear, horizontal flip, and scale, as depicted in Figure 3. This method forces the model to take into account how an image can appear in a range of scenarios, such as in the case of NFT image plagiarism.

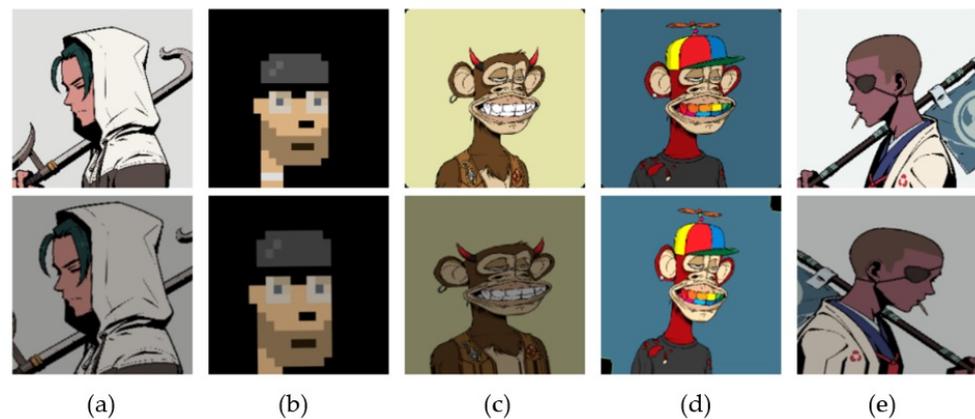


Figure 3. Sample of augmented images of NFT dataset. (a) Augmentation in brightness, (b) augmentation in shear, (c) augmentation in scale, (d) augmentation in rotation, (e) augmentation in horizontal flip.

Table 1. List of hyperparameters of ResNet50, DenseNet, MobileNetV2, EfficientNet-B0.

Hyperparameter	Models Compared (ResNet50, DenseNet, MobileNetV2, EfficientNet-B0)
Activation Function	ReLU
Batch Size	64
Learning rate	0.001
Training data	70%
Validation data	30%
Loss Function	Triplet semi-hard loss
Optimizer	Adam
Embedded dimension	128
Epoch	100

3. Methodology

Transfer learning is a common approach in computer vision and sentiment analysis, where the computational power required to process large datasets can be significant. The basic idea behind transfer learning is to utilize knowledge gained from a task with big labeled training data to tackle a new task with limited data. This method starts the learning process with patterns already discovered in a related task, rather than starting from scratch, using several models implemented in this study, such as ResNet50, DenseNet, MobileNetV2, and EfficientNet-B0. Transfer learning involves using a pretrained deep learning technique to address a new but related problem [28].

3.1. ResNet50

ResNet, also directed to as the Residual Network, is a type of convolutional neural network (CNN) that was invented by He et al. [29] in 2015. CNNs are frequently used by applications that employ computer vision (CV) [30]. ResNet-50 is a 50-layer convolutional neural network that comprises 48 convolutional layers, one intermediate pool layer, and one MaxPool layer. These residual neural networks are artificial neural networks (ANNs) that are constructed using residual blocks. Figure 4 illustrates the ResNet50 model architecture.

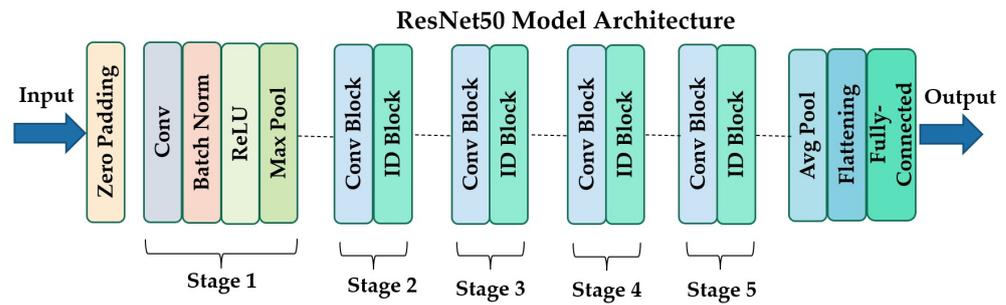


Figure 4. ResNet50 model architecture.

3.2. DenseNet

A DenseNet [31] is a type of convolutional neural network (CNN) that employs dense connections amid layers via dense blocks. In this architecture, all layers with matching feature-map sizes are directly linked to one another. To maintain the feed-forward essence of the system, each layer receives additional information from all preceding layers and transmits its own feature maps to all subsequent layers. The downsampling of layers, which modifies the dimension of feature maps, is a crucial component of convolutional networks. Figure 5 depicts the division of the network into several densely coupled dense blocks in the DenseNet architecture to facilitate downsampling. The convolution and pooling layers amid blocks are referred to as transition layers.

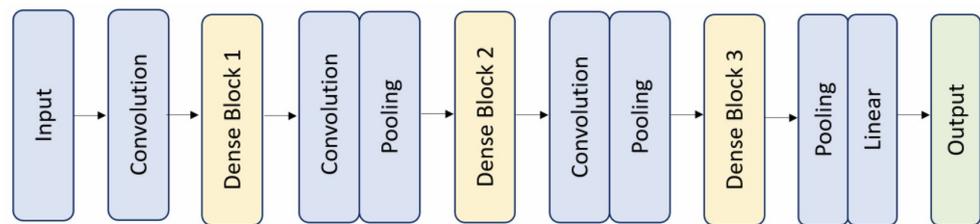


Figure 5. DenseNet model architecture.

3.3. MobileNetV2

The initial support for the MobileNet model was provided by depthwise separable convolutions, which is a type of factorized convolution that splits a standard convolution into a depthwise convolution and a 1×1 convolution commonly known as a pointwise convolution [32]. In contrast, MobileNetV2 features two distinct block types, with one having a stride of one and the other being a shrinking block with a stride of two. Both block types consist of three separate tiers. The first layer in this model is a 1×1 convolution with ReLU6, followed by a depthwise convolution, and then a 1×1 convolution with no non-linearity. It is noted that deep networks only have the classification ability of a linear classifier in the non-zero volume part of the output domain if ReLU is reapplied. Figure 6 depicts the MobileNetV2 model architecture and a comparison to MobileNet [33].

In general, ResNet-50 is a convolutional neural network with 50 layers. We may load a pretrained version of the network from the ImageNet database that has been trained on more than a million images. As a result, the network has learned rich feature representations for various images [34]. The DenseNet model was created using the same fundamental concept as ResNet, but its name refers to its dense connections between the earlier and later layers [35]. In comparison, the MobileNetV2 was created to maximize accuracy successfully while considering the limited resources for an embedded or on-device application. MobileNetV2 is a compact, low-latency, low-power model that may be customized to accommodate different use cases' resource limitations. Based on the experimental results of the studies in this paper, MobileNetV2 has a lesser accuracy than Densenet and ResNet50, but it is faster and lighter [36] in training time.

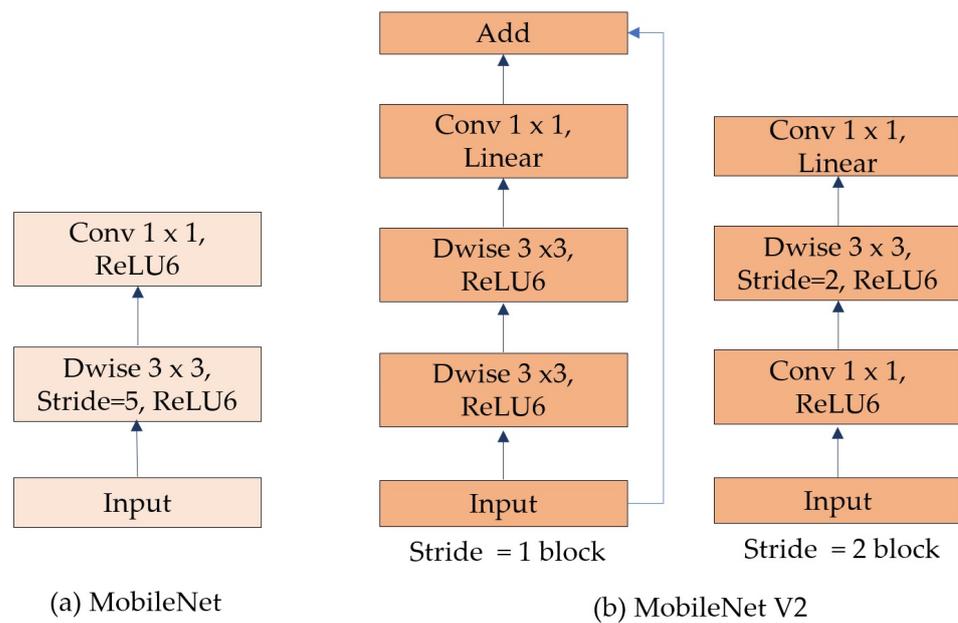


Figure 6. (a) MobileNet model architecture and (b) MobileNetV2 model architecture.

3.4. EfficientNet

The architecture of a convolutional neural network and the scaling strategy employed by EfficientNet [37] involves uniformly scaling all width, depth, and resolution dimensions through a compound coefficient. Unlike conventional methods that arbitrarily scale these elements, EfficientNet’s scaling approach involves uniformly increasing network depth, width, and resolution using a preselected set of scaling coefficients.

For example, if we plan to use 2^N times better computing power, we can improve the network’s depth by α^N , width by β^N , and image dimensions by γ^N , where α , β , and γ are constant coefficients established through a small grid search on the original method. EfficientNet utilizes a compound coefficient, ϕ , to scale the network’s width, depth, and resolution uniformly and based on established rules.

The EfficientNet method uses a compound scaling technique to enhance network depth, width, and resolution in proportion to the size of the input image. This ensures that the network has enough layers to cover the larger receptive area and more channels to detect finer details in the larger image. Furthermore, mobile scaling can be applied to any CNN architecture, and the results are sufficient, although the baseline architecture significantly impacts overall performance. With this in mind, the authors developed a new foundation architecture called the EfficientNet-B0-Based DNN with Triplet Semi-Hard Loss. The building blocks of the base EfficientNet-B0 network are the squeeze-and-excitation blocks and the inverted bottleneck extra blocks of MobileNetV2. The EfficientNet model architecture is shown in Figure 7.

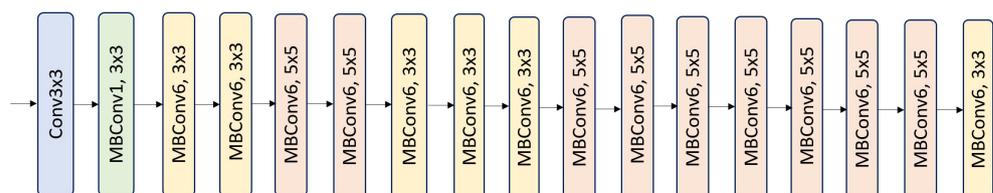


Figure 7. EfficientNet model architecture.

4. Experiment and Results

The deep neural network (DNN) is one of the key components of deep learning (DL), and it can be utilized for image plagiarism checks and various applications. Here, we

compare models including ResNet, DenseNet, MobileNetV2, and EfficientNet-B0 in order to develop the image plagiarism checker box in the proposed solution architecture, as depicted in Figure 1.

4.1. Hyperparameters Setting

In order to ensure a fair comparison with other models, the hyperparameters must be set to optimal and equivalent values. In this study, we set all the models being compared to the same hyperparameters, as shown in Table 1. ReLU activation was chosen because it is less computationally expensive and rectifies the vanishing gradient problem, which is better than other activation functions such as tanh and sigmoid [38]. Furthermore, the default learning rate value of 0.001 was used in most Keras optimizers because it is recommended for beginners [39]. Based on the insights from FaceNet embeddings, we selected an embedded dimension of 128. This model was initially used for face clustering, verification, and identification, and provides greater precision with only 128 bytes per face [40]. The batch size of 64 was chosen because it is appropriate for the amount of data used in the study, and using a mini-batch size that is a power of 2 is recommended [41].

We chose to use the Adam optimizer, as it is a well-known deep-learning training technique that uses exponentially weighted moving averages to manage the gradient's momentum and the second moment, also known as leaky averaging. This optimizer tracks the relative prediction error of the loss function through a weighted average, making it more effective than the standard stochastic gradient descent (SGD) technique, which ignores the effects of outliers [42].

4.2. Performance Criteria

4.2.1. Euclidean Distance

In this paper, Euclidean distance was chosen because it can produce a better recall rate and precision when compared to other distance approaches, such as the Manhattan distance and Hamming distance [43].

Euclidean distance is a granular method for determining similarities between two letter sequences that computes a numerical similarity by accounting for the numeric values of the respective ASCII codes [6]. Assuming that the two points (u, v) are those where $u = (a_1, b_1)$ and $v = (a_2, b_2)$, the Euclidean distance between these two points is determined as specified by

$$EU_d = \sqrt{(a_1 - a_2)^2 + (b_1 - b_2)^2}. \quad (1)$$

If there are more than two dimensions among the pinpointed data, however, Euclidean distance is considered by

$$EU_d = \sqrt{\sum_{i=1}^n (a_i - b_i)^2}. \quad (2)$$

The Euclidean formula is used to determine how similar the query image is to those in the dataset. The model employs the Euclidean distance calculation to determine the level of similarity between two images during the comparison stage of plagiarism detection [7].

4.2.2. Triplet Semi-Hard Loss

Triplet Loss trains a neural network to maximize the distance between embeddings of different classes while ensuring that embeddings of the same class are closely grouped together. This is achieved through selecting an anchor sample, along with a positive and negative sample [44]. This triplet loss aims to guarantee that a picture x_i^a (anchor) of a certain entity is closer to all other images x_i^p (positive) of that entity than it is to any image x_i^n (negative) of any other entity. Figure 8 illustrates this.

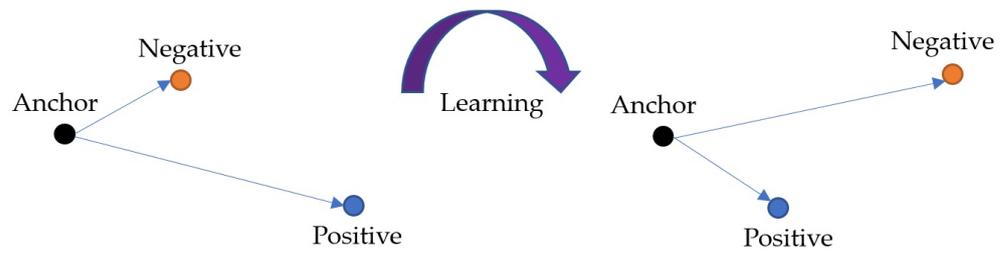


Figure 8. The triplet loss trains a neural network to increase the difference between the embeddings of different classes while minimizing the difference between the embeddings of the same class by selecting an anchor, a positive sample, and a negative sample.

The number of triplets that can readily satisfy the requirement in Equation (3) would increase if all potential triplets were generated. Since they would still be propagated over the network, these triplets would not support training and would slow down the computation.

$$\|x_i^a - x_i^p\|_2^2 + \alpha < \|x_i^a - x_i^n\|_2^2, \forall (x_i^a, x_i^p, x_i^n) \in \tau. \tag{3}$$

where the boundary amid positive and negative pairings is defined by α . τ , which has a cardinality of N represents the complete set of all possible triplets in the training set.

Therefore, L is equal to the loss that is being reduced in the below equation:

$$\sum_i^N [\|f(x_i^a) - f(x_i^p)\|_2^2 + \alpha < \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha]_+. \tag{4}$$

Using a smaller batch size while choosing the hardest negatives, on the other hand, can result in collapsed models (i.e., $f(x) = 0$) and inadequate local minima early in the training process. To counteract this, choose x_i^n such that:

$$\|f(x_i^a) - f(x_i^p)\|_2^2 < \|f(x_i^a) - f(x_i^n)\|_2^2. \tag{5}$$

where the negatives are contained within the margin α . The function $f(x)$ explains the embedding function, which receives an image x into a d -dimensional Euclidean space. The image x_i^a is anchor, image x_i^p is positive, and x_i^n is negative subsequently. The part $\|f(x_i^a) - f(x_i^p)\|_2^2$ from the equation subtracts the anchor image with the positive image between the embeddings to determine Euclidean distance. Meanwhile the second part, $\|f(x_i^a) - f(x_i^n)\|_2^2$, calculates the Euclidean distance between the embeddings by subtracting the anchor image from the negative image. The negative quintessences from Equation (5) are classified as *semi-hard* because, while being farther from the anchor than the positive exemplar, they are challenging because the squared distance is close to the anchor-positive span. In other words, these are called triplets when the negative creates a positive loss, despite being further away from the anchor than the positive.

In this study, we prefer the “semi-hard” category among the “easy” and “hard” triplet categories because it allows the model to learn valuable features without overfitting, and it generalizes well to the test data [40].

4.3. Results and Comparison

In this study, we applied the EfficientNet-B0 with the Triplet Semi-Hard Loss model to detect plagiarism using the same dataset for three examples of threshold scores, which are 0.55, 0.65, and 0.75, to acquire the optimum score for examining the plagiarism of images.

Overall, The proposed EfficientNet-B0-based DNN with Triplet Semi-Hard Loss retains the lowest loss and highest accuracy compared to ResNet50, DenseNet, and MobileNetV2 at various sampling threshold scores, which means that image plagiarism detection can be ensured.

According to the experiments, EfficientNet-B0 with Triplet Semi-Hard Loss has reached the lowest loss, 0.1242, and lowest validation loss, 0.2808, compared to the other methods as elaborated in Table 2. Furthermore, as shown in Figure 9, the graph of EfficientNet-B0, represented by the red line, has the lowest loss, and the red dashed line depicts the lowest validation loss compared to the other three models. These results indicate the efficiency of the proposed approach in using computation resources. EfficientNet-B0 models use a compound scaling method that scales the dimensions of the network (depth, width, and resolution) in a computationally efficient manner, resulting in better performance with lesser resources [37]. Additionally, using a triplet semi-hard loss function can contribute to the modest loss since it can lower the training complexity by choosing negative pairs close to the anchor but adequately distant from it [45].

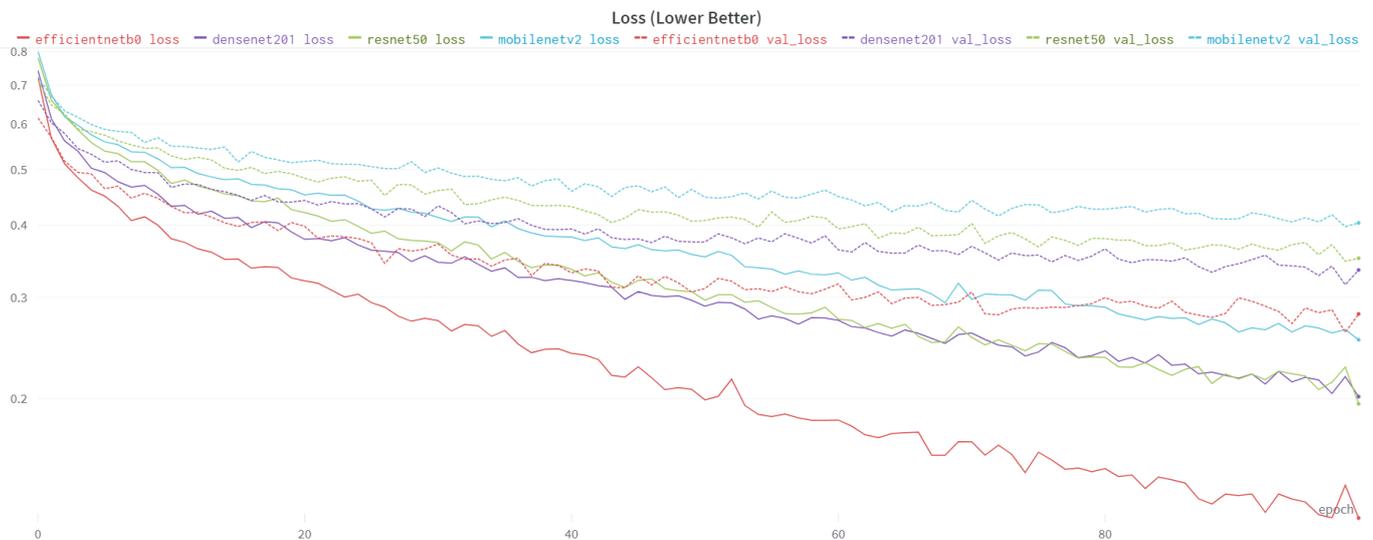


Figure 9. The comparison of loss and validation loss from ResNet50, DenseNet, MobileNetV2, and EfficientNet-B0.

Table 2. Comparison of loss and validation loss for all models.

Model	Loss	Validation Loss	Training Times
ResNet50	0.1961	0.3507	228.677
DenseNet	0.2018	0.3346	223.992
MobileNetV2	0.2533	0.4040	222.148
EfficientNet-B0	0.1242	0.2808	230.379

Additionally, Table 2 displays the loss and validation loss and the training time results for all four models. The MobileNetV2 model's training time was the shortest at 222.148 min, while the EfficientNet-B0 model's training time was the longest at 230.379 min. MobileNetV2 has a smaller structure with a size of 14 MB and a depth of 105, whereas EfficientNet-B0 has a structure size of 29 MB and a depth of 132 [46]. Therefore, based on model architecture, EfficientNet-B0 has a larger and more complex model than the other models, with more parameters and layers. This model can increase the training time as more computations are required to update the weights and biases of the model. Furthermore, the EfficientNet-B0 model is designed to handle larger images and batch sizes than the other models, which can raise the training time [47].

Figure 10 illustrates the EfficientNet-b0 model with the lowest maximum distance of positive pairs, which is represented by the red line, followed by DenseNet with the purple line, ResNet50 with the green line, then MobileNetV2 with the blue line, respectively. While the lowest validation maximum distance of positive pairs was achieved by DenseNet with

a purple dashed line, then followed by EfficientNet-B0 with a red dashed line, ResNet50 with a green dashed line, then MobileNetV2 with a blue dashed line subsequently.

From Table 3, we can see the detailed results of the experiments that EfficientNet-B0 has the lowest maximum distance of positive pairs with 0.4526 and validation maximum distance of positive pairs of 0.8059 when compared to other MobileNetV2, DenseNet, and ResNet50. MobileNetV2 has a maximum distance of positive pairs of 0.5889 and validation of the maximum distance of positive pairs of 0.8457. While DenseNet has a maximum distance of positive pairs of 0.5223 and validation of a maximum distance of positive pairs of 0.7032. On the other hand, ResNet50 has a maximum distance of positive pairs of 0.5827 and validation of a maximum distance of positive pairs of 0.8199.

These experimental results indicate that the EfficientNet-B0 model has a better representation of the data, which could be determined by the compound scaling method utilized in EfficientNet-B0. These results also mean that EfficientNet-B0 can learn more informative and selective features from the input data, which helps it to detect plagiarized images more precisely. However, the lowest validation maximum distance of positive pairs was not achieved by EfficientNet-B0, though DenseNet obtained it with 0.7032 among compared models, which indicates that DenseNet generalizes well to validation data and is better at detecting plagiarized images on the validation sets. The dense connections in the DenseNet architecture allow the model to learn more robust features that generalize well to the validation data [48], which may enable it to identify plagiarized images more accurately on the validation data.

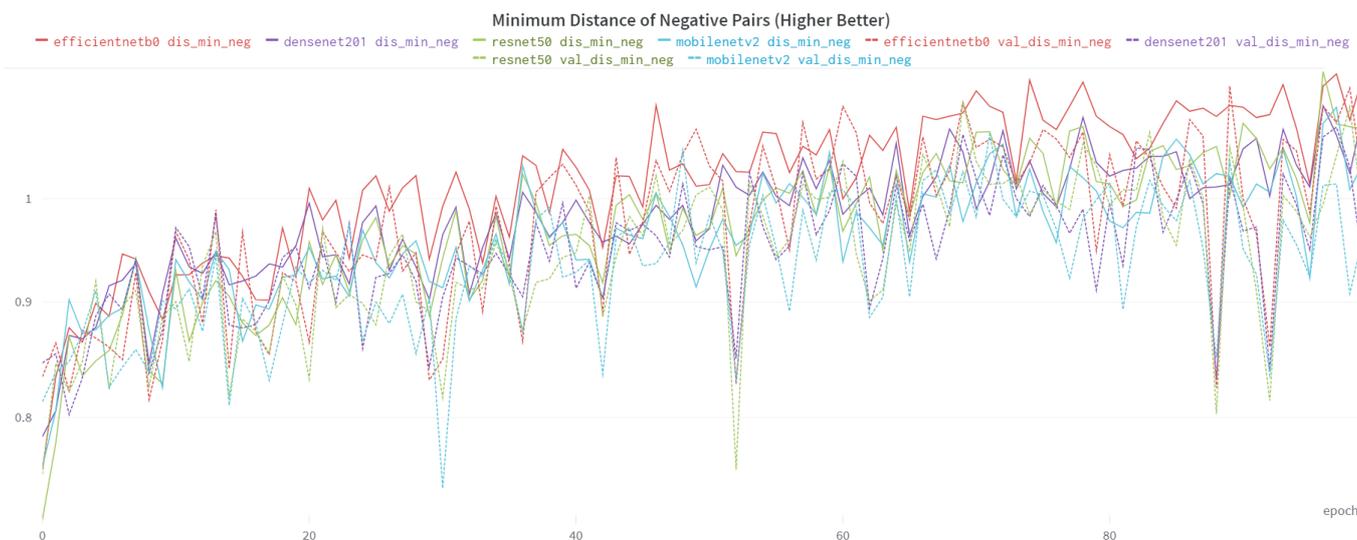


Figure 10. The comparison of the maximum distance of positive pairs among ResNet50, DenseNet, MobileNetV2, and EfficientNet-B0.

From Table 3, we can see that the EfficientNet-B0-based DNN has achieved the highest maximum distance of negative pairs, with a training result of 1.133 and a maximum validation distance of negative pairs of 1.0150. MobileNetV2 has a maximum distance of negative pairs of 1.037 and a maximum validation distance of negative pairs of 0.9634, DenseNet has a maximum distance of negative pairs of 1.090 and a maximum validation distance of negative pairs of 0.9342, and ResNet50 has a maximum distance of negative pairs of 1.074 and a maximum validation distance of negative pairs of 0.9817. These results indicate that the EfficientNet-B0 model can learn more robust features that can capture subtle differences between the original and plagiarized images, resulting in a higher maximum distance of negative pairs. Figure 11 compares all models in maximum negative pairs.

Table 3. Maximum distance of positive pairs and minimum distance of negative pairs of all models.

Parameter	Model	Train	Validation
Maximum Distance of Positive Pairs	ResNet50	0.5827	0.8199
	DenseNet	0.5223	0.7032
	MobileNetV2	0.5889	0.8457
	EfficientNet-B0	0.4526	0.8059
Minimum Distance of Negative Pairs	ResNet50	1.074	0.9817
	DenseNet	1.090	0.9342
	MobileNetV2	1.037	0.9634
	EfficientNet-B0	1.133	1.0150



Figure 11. The comparison of the minimum distance of negative pairs among ResNet50, DenseNet, MobileNetV2, and EfficientNet-B0.

From Table 4, we can see that EfficientNet-B0 has the highest accuracy on the training set among MobileNetV2, DenseNet, and ResNet50 with threshold scores of 0.55, 0.65, and 0.75. This outcome implies that EfficientNet-B0 is better at generalizing to the validation data and identifying plagiarized images on the validation set when compared to the other models. The EfficientNet-B0 model can learn more robust features that can capture slight differences between the original and plagiarized images, resulting in higher accuracy on the validation set. Figures 12–14 show the results of accuracy and validation accuracy from the compared models in this study with various threshold scores of 0.55, 0.65, and 0.75, respectively.

In addition, Table 4 shows that DenseNet has the highest validation accuracy among the other models (EfficientNet-B0, MobileNetV2, and ResNet50) with threshold scores of 0.55, 0.65, and 0.75. This result is presumably due to DenseNet’s architecture using dense connections between layers and a feed-forward technique where each layer is connected to all the other layers. The DenseNet method allows the gradients to flow directly from any layer to any other layer, enabling the model to learn more robust features that can capture slight differences between the original and plagiarized images [49]. The dense connections in DenseNet architecture also reduce the risk of overfitting, allowing the model to generalize better to the validation data [50].

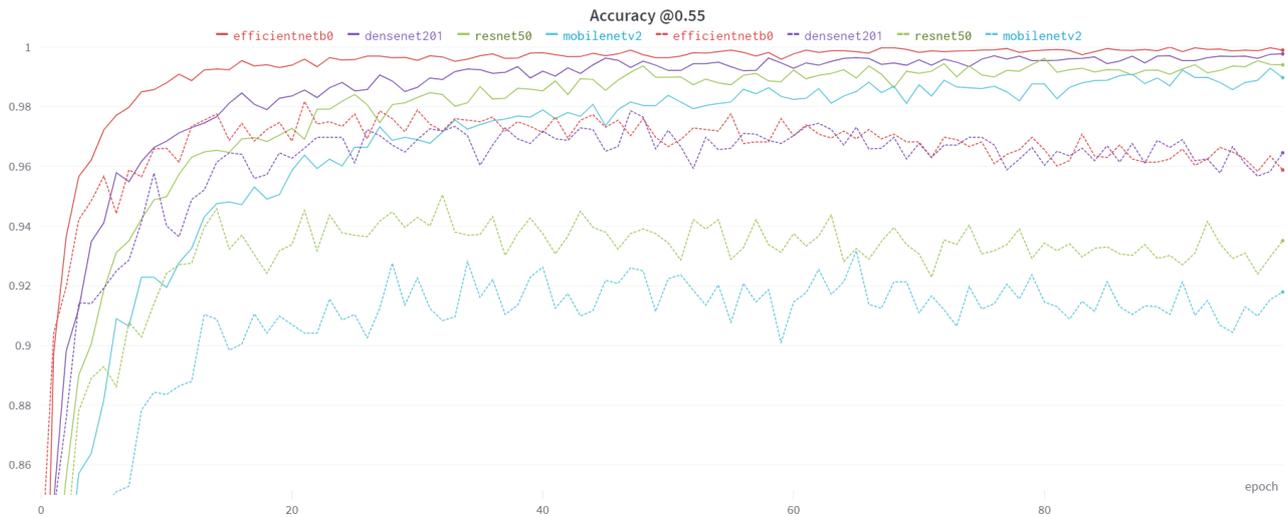


Figure 12. Comparison of the accuracy among ResNet50, DenseNet, MobileNetV2, and EfficientNet-B0 with a threshold score of 0.55.



Figure 13. Comparison of the accuracy among ResNet50, DenseNet, MobileNetV2, and EfficientNet-B0 with a threshold score of 0.65.

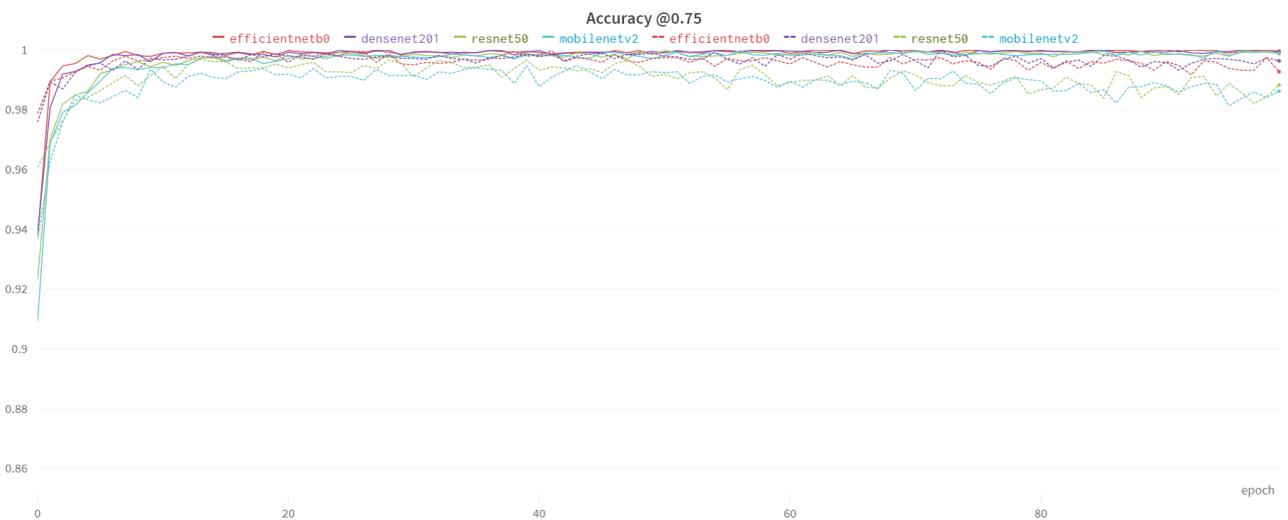


Figure 14. Comparison of the accuracy among ResNet50, DenseNet, MobileNetV2, and EfficientNet-B0 with a threshold score of 0.75.

Table 4. The accuracy of all comparison methods is presented with threshold scores of 0.55, 0.65, and 0.75.

Threshold Score	Model	Accuracy	Validation Accuracy
0.55	ResNet50	0.9941	0.9352
	DenseNet	0.9977	0.9646
	MobileNetV2	0.9898	0.9180
	EfficientNet-B0	0.9990	0.9588
0.65	ResNet50	0.9977	0.9700
	DenseNet	0.9995	0.9864
	MobileNetV2	0.9980	0.9607
	EfficientNet-B0	0.9997	0.9838
0.75	ResNet50	0.9987	0.9883
	DenseNet	0.9995	0.9963
	MobileNetV2	0.9990	0.9862
	EfficientNet-B0	0.9996	0.9927

The augmented images in our experiments are used to simulate potential variations of plagiarized images. These augmented images are a combination of brightness modification, shear, scaling, rotation, and horizontal flip distortion techniques, and then we used these augmented images to test our EfficientNet-based model with the triple semi-hard loss function. Our experiments showed that our model successfully detected augmented images as plagiarized images with high accuracy. Figure 15 displays the simulation findings. Figure 16 demonstrates that the original NFT image was uploaded, whereas Figure 15 demonstrates that our proposed model was able to identify plagiarism in the image that was augmented from Figure 16.

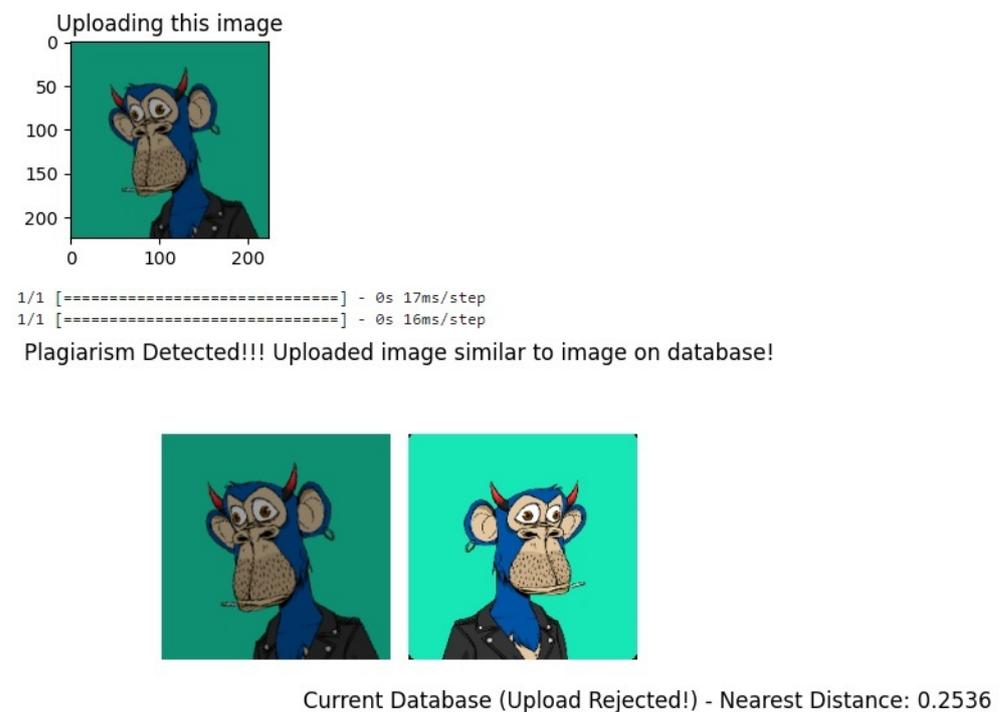


Figure 15. Simulation of detecting plagiarized NFT image.

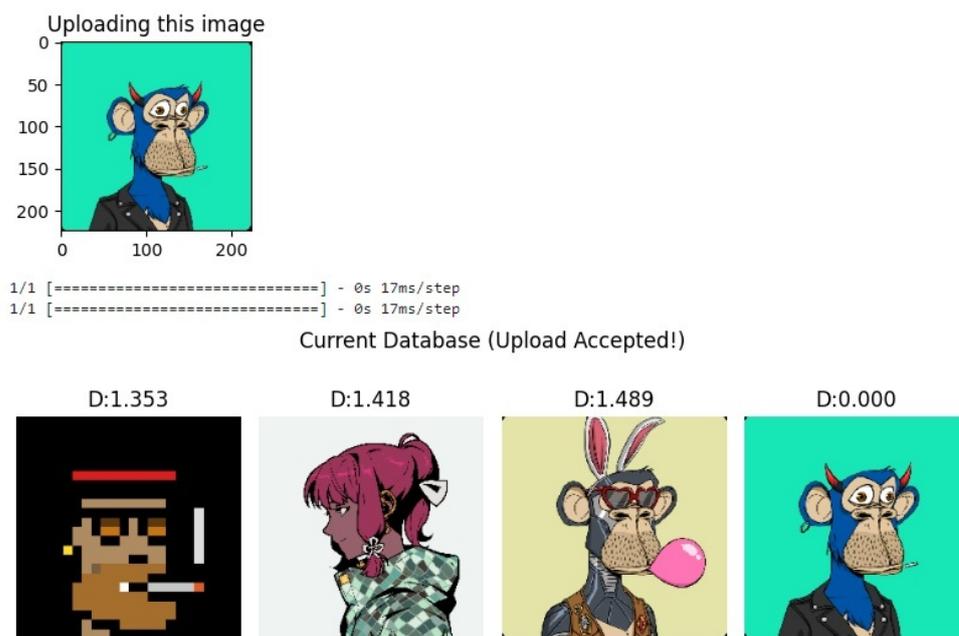


Figure 16. Simulation of uploading original NFT image.

As we can see from the simulation in Figure 15, the proposed model can detect plagiarized images with the nearest distance of 0.2536, which is lower than the threshold score (0.55, 0.65, or 0.75).

5. Conclusions

Image plagiarism in the NFT system can have a negative impact on both artists and customers; there is a necessity for a framework to provide a security system to secure all of the related parts. Developing a reliable detection of plagiarism system is essential to ensure that people who relate to the NFT system can get protection. We developed and implemented the EfficientNet-B0-Based DNN with T triplet semi-hard loss to build image plagiarism detection in the NFT system in this study. Furthermore, the proposed EfficientNet-B0-Based DNN with triplet semi-hard loss method was implemented for detecting plagiarism NFT images to obtain the lowest loss and the highest accuracy compared with ResNet50, DenseNet, and MobileNetV2 models for various threshold scores of 0.55, 0.65, and 0.75.

Although we gained promising results, some other studies must be added in the future. This future research needs to expand the dataset to include a more extensive and diverse set of NFT images to improve the model’s performance. Furthermore, the proposed method can be enhanced and deployed in real-world scenarios to determine its efficacy in detecting plagiarism in NFT images. The proposed solution might be implemented into current NFT marketplaces such as Opensea, Rarible, SuperRare, etc., adding extra security for artists and customers. The method can be deployed as a preprocessing step before placing NFT images on the market, ensuring that only original images are utilized.

In addition, the proposed method can be applied to other digital media besides images, such as films and music. As the NFT ecosystem increases, the demand for secure and trustworthy plagiarism detection solutions will also be increased. Implementing the proposed method in several NFT marketplaces might secure artists’ rights and increase the credibility and reliability of the entire NFT system.

Author Contributions: Conceptualization, A.T.P., N.S., and S.O.; methodology, A.T.P. and N.S.; software, A.T.P. and N.S.; validation, A.T.P., N.S., and T.-T.-H.L.; formal analysis, A.T.P.; investigation, A.T.P.; resources, A.T.P. and S.O.; data curation, A.T.P.; writing—original draft preparation, A.T.P.; writing—review and editing, A.T.P. and T.-T.-H.L.; visualization, A.T.P., N.S. and S.O.; supervision, T.-T.-H.L. and H.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the Convergence security core talent training business (Pusan National University) support program (IITP-2023-2022-0-01201) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation) and a part by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2023-2020-0-01797) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available in a publicly accessible repository that does not issue DOIs. Publicly available datasets were analyzed in this study. This data can be found here: <https://www.kaggle.com/datasets/shaunmak/nft-classifier> (accessed on 19 November 2022).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

Adam	Adaptive Momentum Estimation
ANN	Artificial Neural Networks
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
CID	Content Identification
CNN	Convolutional Neural Network
CV	Computer Vision
DNN	Deep Neural Network
EIP	Ethereum Improvement Proposals
ERC	Ethereum Request for Comment
GAN	Generative Adversarial Network
HSV	Hue Saturation Value
HTTP	Hypertext Transfer Protocol
IPFS	InterPlanetary File System
NDFA	Non-deterministic Finite Automaton
NFT	Non-Fungible Token
ReLU	Rectified Linear Unit
RGB	Red Green Blue
SGD	Stochastic Gradient Descent

References

- Ozon Networks, I. What is an NFT? Available online: <https://opensea.io/learn/what-are-nfts> (accessed on 16 December 2022).
- Mochram, R.; Makawowor, C.; Tanujaya, K.; Moniaga, J.; Jabar, B. Systematic Literature Review: Blockchain Security in NFT Ownership. In Proceedings of the 2022 International Conference on Electrical and Information Technology (IEIT), Malang, Indonesia, 14–15 September 2022; pp. 302–306.
- Abaci, I.; Ulku, E. NFT-based Asset Management System. In Proceedings of the 2022 International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Ankara, Turkey, 20–22 October 2022; pp. 697–701.
- Prior, G. Since the Explosion of NFTs, Plagiarism and Fakes Have Increase Problems. Available online: <http://www.koreaitimes.com/news/articleView.html?idxno=111519> (accessed on 16 December 2022).
- Bonifacic, I. Over 80 Percent of NFTs Minted for Free on OpenSea Are Fake, Plagiarized or Spam. Available online: <https://www.engadget.com/opensea-free-minting-tool-220008042.html> (accessed on 19 December 2022).
- Pungila, C.; Galis, D.; Negru, V. A New High-Performance Approach to Approximate Pattern-Matching for Plagiarism Detection in Blockchain-Based Non-Fungible Tokens (NFTs). *arXiv* **2022**, arXiv:2205.14492.

7. Ibrahim, A.; Khalifa, O.; Ahmed, D. Plagiarism Detection of Images. In Proceedings of the 2020 IEEE Student Conference on Research and Development (SCORED), Batu Pahat, Malaysia, 27–29 September 2020; pp. 183–188.
8. Gayadhankar, K.; Patel, R.; Lodha, H.; Shinde, S. Image plagiarism detection using GAN-(Generative Adversarial Network). In Proceedings of the ITM Web of Conferences, Navi Mumbai, India, 14–15 July 2021; Volume 40, p. 03013.
9. Meuschke, N.; Gondek, C.; Seebacher, D.; Breitingner, C.; Keim, D.; Gipp, B. An adaptive image-based plagiarism detection approach. In Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries, Fort Worth, TX, USA, 3–7 June 2018; pp. 131–140.
10. Shahriar, S.; Hayawi, K. NFTGAN: Non-Fungible Token Art Generation Using Generative Adversarial Networks. In Proceedings of the 2022 7th International Conference on Machine Learning Technologies (ICMLT), Rome, Italy, 11–13 March 2022; pp. 255–259.
11. Bao, H.; Roubaud, D. Non-Fungible Token: A Systematic Review and Research Agenda. *J. Risk Financ. Manag.* **2022**, *15*, 215. [CrossRef]
12. Shilina, S. Blockchain and Non-Fungible Tokens (NFTs): A New Mediator Standard for Creative Industries Communication. 2021. pp. 217–225. Available online: <https://bit.ly/3FLFDQV> (accessed on 20 December 2022).
13. Entriiken, W.; Shirley, D.; Evans, J.; Sachs, N. EIP-721: Non-Fungible Token Standard, Ethereum Improvement Proposals, no. 721, January 2018. [Online Serial]. Available online: <https://eips.ethereum.org/EIPS/eip-721> (accessed on 20 December 2022).
14. Radomski, W.; Cooke, A.; Castonguay, P.; Therien, J.; Binet, E.; Sandford, R. EIP-1155: Multi Token Standard, Ethereum Improvement Proposals, no. 1155, June 2018. Available online: <https://eips.ethereum.org/EIPS/eip-1155> (accessed on 20 December 2022).
15. Wang, Q.; Li, R.; Wang, Q.; Chen, S. Non-fungible token (NFT): Overview, evaluation, opportunities and challenges. *arXiv* **2021**, arXiv:2105.07447.
16. Howell, J. ERC 1155 Vs. ERC 721—Key Differences. Available online: <https://101blockchains.com/erc-1155-vs-erc-721/> (accessed on 16 February 2023).
17. Ravencraft, E. NFTs Don't Work the Way You Might Think They Do. Available online: <https://www.wired.com/story/nfts-dont-work-the-way-you-think-they-do/> (accessed on 22 January 2023).
18. Smith, C. Scaling. Available online: <https://ethereum.org/en/developers/docs/scaling/> (accessed on 14 February 2023).
19. Ivanovs, A. How to Create an NFT Collection With a Smart Contract. Available online: <https://geekflare.com/create-nft-collection-with-a-smart-contract/> (accessed on 14 February 2023).
20. Westerkamp, M. Blockchain Interoperability and Its Relevance. Available online: <https://www.gsma.com/aboutus/workinggroups/blockchain-interoperability-and-its-relevance> (accessed on 14 February 2023).
21. Bellagarda, J.; Abu-Mahfouz, A. Connect2NFT: A Web-Based, Blockchain Enabled NFT Application with the Aim of Reducing Fraud and Ensuring Authenticated Social, Non-Human Verified Digital Identity. *Mathematics* **2022**, *10*, 3934. [CrossRef]
22. Battah, A.; Madine, M.; Alzaabi, H.; Yaqoob, I.; Salah, K.; Jayaraman, R. Blockchain-based multi-party authorization for accessing IPFS encrypted data. *IEEE Access* **2020**, *8*, 196813–196825. [CrossRef]
23. Choi, D. Decentralizing NFT.Storage. Available online: <https://blog.nft.storage/posts/2022-01-20-decentralizing-nft-storage> (accessed on 15 February 2023).
24. Technology, M. IPFS NFT—How to Use IPFS for NFT Metadata. Available online: <https://moralis.io/ipfs-nft-how-to-use-ipfs-for-nft-metadata/> (accessed on 11 January 2022).
25. Mak, S. Nft-Classifier. Available online: <https://www.kaggle.com/datasets/shaunmak/nft-classifier> (accessed on 19 November 2022).
26. Frederik Hvilshoj Balanced and Imbalanced Datasets in Machine Learning [Introduction]. Available online: <https://encord.com/blog/an-introduction-to-balanced-and-imbalanced-datasets-in-machine-learning/> (accessed on 17 February 2022).
27. Nelson, J. What Is Image Preprocessing and Augmentation? Available online: <https://blog.roboflow.com/why-preprocess-augment/> (accessed on 22 December 2022).
28. Donges, N. What Is Transfer Learning? Exploring the Popular Deep Learning Approach. Available online: <https://builtin.com/data-science/transfer-learning> (accessed on 23 December 2022).
29. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
30. Prihatno, A.; Utama, I.; Kim, J.; Jang, Y. Metal Defect Classification Using Deep Learning. In Proceedings of the 2021 Twelfth International Conference on Ubiquitous and Future Networks (ICUFN), Jeju Island, Republic of Korea, 17–20 August 2021; pp. 389–393.
31. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
32. Howard, A.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Wey, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
33. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520.
34. Vaibhav Kumar MobileNet vs ResNet50—Two CNN Transfer Learning Light Frameworks. Available online: <https://analyticsindiamag.com/mobilenet-vs-resnet50-two-cnn-transfer-learning-light-frameworks> (accessed on 19 February 2023).

35. Lee, C.; Xie, S.; Gallagher, P.; Zhang, Z.; Tu, Z. Deeply-supervised nets. In *Artificial Intelligence and Statistics*; U.S. Department of Energy: Washington, DC, USA, 2015; pp. 562–570.
36. Tsang, S. Review: MobileNetV2—Light Weight Model (Image Classification). Available online: <https://towardsdatascience.com/review-mobilenetv2-light-weight-model-image-classification-8febb490e61c> (accessed on 6 January 2023).
37. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
38. Biswas, P. Intuitions behind Different Activation Functions in Deep Learning. Available online: <https://towardsdatascience.com/intuitions-behind-different-activation-functions-in-deep-learning-a2b1c8d044a> (accessed on 25 January 2023).
39. Pramoditha, R. How to Choose the Optimal Learning Rate for Neural Networks. Available online: <https://towardsdatascience.com/how-to-choose-the-optimal-learning-rate-for-neural-networks-362111c5c783> (accessed on 25 January 2023).
40. Schroff, F.; Kalenichenko, D.; Philbin, J. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 815–823.
41. Ng, A. How to Choose the Optimal Learning Rate for Neural Networks. Available online: <https://cs230.stanford.edu/files/C2M2.pdf> (accessed on 25 January 2023).
42. Prihatno, A.; Nurcahyanto, H.; Ahmed, M.; Rahman, M.; Alam, M.; Jang, Y. Forecasting PM2.5 Concentration Using a Single-Dense Layer BiLSTM Method. *Electronics* **2021**, *10*, 1808. [[CrossRef](#)]
43. Chugh, H.; Gupta, S.; Garg, M.; Gupta, D.; Juneja, S.; Turabieh, H.; Na, Y.; Kiros Bitsue, Z. Image retrieval using different distance methods and color difference histogram descriptor for human healthcare. *J. Healthc. Eng.* **2022**, *2022*, 9523009. [[CrossRef](#)] [[PubMed](#)]
44. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015. Available online: <https://www.tensorflow.org/> (accessed on 19 December 2022).
45. Kim, S.; Kim, D.; Cho, M.; Kwak, S. Proxy anchor loss for deep metric learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 3238–3247.
46. Chollet, F. Keras Applications. Available online: <https://keras.io/api/applications/> (accessed on 12 January 2023).
47. Wongpanich, A.; Pham, H.; Demmel, J.; Tan, M.; Le, Q.; You, Y.; Kumar, S. Training EfficientNets at supercomputer scale: 83% ImageNet top-1 accuracy in one hour. In Proceedings of the 2021 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), Portland, OR, USA, 17–21 June 2021; pp. 947–950.
48. Pedraza, A.; Deniz, O.; Bueno, G. On the relationship between generalization and robustness to adversarial examples. *Symmetry* **2021**, *13*, 817. [[CrossRef](#)]
49. Rafi, A.; Kamal, U.; Hoque, R.; Abrar, A.; Das, S.; Laganieri, R.; Hasan, M. Application of DenseNet in Camera Model Identification and Post-processing Detection. *CVPR Work.* **2019**, 19–28.
50. Deepchecks Ltd. Densenet. Available online: <https://deepchecks.com/glossary/densenet/> (accessed on 3 February 2023).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.