Backdoor Watermarking as a Service: Safer Way to Protect Your Copyright for Image Files

Yonghee Kwon
Korea University
hee980410@korea.ac.kr

Abstract

Recent discussions in the cultural industry have highlighted the growing concern over copyright issues, spurred by the digital age's ease of distributing creative works online. This surge in digital sharing has led to a rise in copyright infringement attempts, resulting in substantial financial losses for the industry and hindering efforts to foster creativity. Urgent measures are needed to address this challenge and emphasize the importance of copyright protection.

In the realm of image classification, neural networks are vulnerable to attacks such as backdoor injection, where subtle modifications can cause misclassification. This study aims to develop a neural network capable of recognizing specific images as the work of specific authors by embedding backdoors. The main contributions of this work include the development of a user-friendly mechanism for embedding author ID-driven watermarks into images and the creation of a neural network capable of identifying authors based on hidden watermarks, irrespective of image size or position. This mechanism will allow for easy determination of authorship, even if the image is resized or cropped. All of the experimental codes and the results are accessible at [2].

1 Introduction

In recent times, discussions surrounding copyright issues have been gaining momentum within the cultural industry. With the advent of the digital age, the distribution of creative works has become more accessible, and the sharing of information via the Internet has surged. Consequently, attempts at copyright infringement are on the rise. It's estimated that hundreds of billions of works are illicitly replicated or distributed each year [7], resulting in significant losses for the cultural industry. According to the statistics, the financial toll from copyright infringement reaches tens of billions of dollars annually. This not only inflicts substantial harm on creators but also undermines efforts to enhance the quality of works and foster the creation of new ones. Moreover, such infringements exert a negative influence on the entire cultural industry, hampering its healthy growth and economic stability. Urgent measures are required to address this issue, underscoring the heightened importance of copyright protection.

Image classifier is a neural network that takes an image as input and outputs what the image contains in the form of words. Backdoor injection is one of the approaches to attacking this neural network, enticing a well-trained classifier to produce incorrect classification results. Through this method, it's possible to trick a classifier into mistaking a dog for a cat or interpreting a stop sign as a speed limit sign [9]. In this study, a neural network is developed that can recognize specific pictures or photos as the works of specific authors, using this attack method. By embedding backdoor into existing images or video files, it is aimed to create a mechanism that can easily determine the authorship of a work by querying a specific neural network when it's suspected of unauthorized use elsewhere in the future. Illegal streaming channels on platforms like YouTube sometimes display replicated videos in a small

corner of the screen instead of full size. This study also aims to develop backdoors that function uniformly regardless of image resizing, ensuring they operate consistently even in such scenarios.

Contributions of this Work The main contributions of this work will be:

- Make user-friendly mechanism which embed authorID-driven watermark into images
- Show that a neural network can identify the author based-on hidden watermark
- Show that a neural network can detect watermark regardless of the size or position of the original image

2 Background

2.1 Image Classification

Image classification technique uses a neural network that takes an image as input and outputs its corresponding label. To build classifiers, Convolutional Neural Networks (CNN) are typically employed. Convolution layers are usually first step of CNN, receiving input image features and analyzing the characteristics of the image. Multiple filters are used in the convolution layers to analyze the feature. Subsequently, through several layers, classification results are derived from the final fully connected layer. Techniques like pooling and padding are also utilized to better analyze the features of the image.

Starting from LeNet5 [16] in 1994, new CNN techniques are continuously being introduced including AlexNet [15] in 2012, GoogLeNet [21] in 2014, ResNet [11] in 2015 and so on. In the process, the technique has evolved more and more. New techniques such as dropout and "Inception" module are suggested and the size of the neural networks have been getting bigger and bigger. Now the error rate of CNNs is about 3.57% to 2.99% [20] which is a way lower rate than human error's.

Image classification is one of the base techniques of image recognition and it has large amount of usability. The technique can evolve into a multi-label recognition system, enabling the system to identify multiple objects simultaneously. For instance, it could accurately detect both a person and a bicycle in a crowded street scene, or recognize a cat and a dog playing together in a home environment. This expanded capability enhances its usefulness across diverse contexts, from urban surveillance to pet monitoring. Another good application is auto-driving technology, where video recognition systems play a pivotal role. These systems can identify various elements of the road environment, such as pedestrians, traffic signs, and other vehicles, contributing to safer and more efficient navigation.

2.2 Embedding Backdoors into Image

Several types of adversarial attacks against image classifiers exist which aim to disrupt their classification accuracy. One such attack strategy involves embedding backdoors [5, 19, 26] into input images, thus deceiving the classifier into producing incorrect predictions. This technique, known as backdoor attack, entails subtly modifying the input data to include imperceptible perturbations that trigger misclassification. Despite the remarkable robustness of CNNs in general, their susceptibility to such attacks poses significant challenges in security-critical applications, such as autonomous vehicles and biometric systems.

The backdoor attack operates by exploiting the inherent vulnerabilities in the training process of CNNs. Adversaries can strategically inject backdoors during the model training phase by incorporating poisoned data samples with carefully crafted perturbations. These perturbations, often imperceptible to human observers, encode a specific trigger pattern that triggers misclassification when present in the input image during inference. By introducing backdoor-laden images into the training dataset, attackers can manipulate the model's decision boundaries, compromising its reliability and integrity. Consequently, the resulting model exhibits a compromised performance, misclassifying input images containing the trigger pattern, even when the original labels are clear and distinct.

Mitigating the threat posed by backdoor attacks requires robust defense mechanisms and rigorous model evaluation techniques. Researchers are actively exploring various strategies to enhance the resilience of CNNs against such attacks, including adversarial training [18, 25], input sanitization [12], and anomaly detection [8]. Additionally, advancements in explainable AI (XAI) [4] can aid in

identifying and mitigating the presence of backdoors by providing insights into the model's decision-making process [13]. As the deployment of CNNs in safety-critical applications continues to expand, safeguarding against adversarial attacks like backdoors becomes paramount to ensure the reliability and trustworthiness of these systems in real-world scenarios.

In this work, however, we will plant backdoors in the image that can only be matched by authors with exact keys. We will make a backdoor-making mechanism which utilize the creator's private key to generate the proper and unique backdoor pattern and embed it to target image. We will also make proper neural network which can identify the target image with backdoor embedded. This neural network will be suitable to transfer learning since it requires additional training whenever additional creators register their private keys to use the system.

2.3 Watermarking

Image watermarking technology plays a crucial role in digital content protection and copyright enforcement. By embedding imperceptible or semi-visible watermarks into images, content creators can assert ownership and prevent unauthorized use or distribution. This process involves altering the pixels of an image in a manner that doesn't significantly degrade its visual quality but adds unique identifying information, such as the creator's name or copyright details. These watermarks serve as a digital signature, allowing creators to track the origin of their work and take legal action against infringement.

Watermarking is also used in neural networks. Mostly, watermarks are embedded into models in order to protect intellectual property and prevent model theft of the neural networks [6, 22, 24]. This approach aims to deter unauthorized replication or misuse of trained models by encoding ownership information directly into their architecture or parameters.

Additionally, several research has explored techniques for embedding watermarks directly into images using neural networks [23, 17]. In such cases, both embedding and extraction networks are developed and employed. However, a key concern arises from the possibility of embedded watermarks being exposed during the extraction process, enabling others embed same watermark to other images. To address this challenge, this work proposes a novel approach where watermarks, generated using the author's private key, are inserted into images. Subsequently, a designated neural network outputs labels corresponding to the public key of the author, enabling verification of authorship. Importantly, this method ensures that only the author can generate identical watermarks, thus enhancing the authenticity and integrity of digital assets while mitigating the risk of unauthorized replication by third parties.

3 Suggested Method

In this section, the methods to validate the proposed idea are addressed. Mostly works from previous studies will be utilized except for concept of making watermark based on author's private key and make neural network classifying the author. In case of building CNN and evaluating it, popular methods will be borrowed.

3.1 Embed Watermark into Image

The key concept of creating a watermark is to associate it with each author's private key so that only the author can create and embed a valid watermark. In this work, 2D-image of QR code will be used as a watermark. It will contain author's name and an electronic signature of it which can be only generated with author's valid private key.

3.2 Build Neural Network

CNN is an appropriate choice because backdoors will be developed in a two-dimensional pattern and neural networks try to identify images regardless of location in the entire input image. Using CNN will allow the system to identify the location of copyrighted image using its convolution layers. It may detect a rectangular area inside the input image. The system would not need such a complex or deep neural network because it only needs to identify rectangular areas and backdoors, not the complex shape of the object included in the input image.

3.3 Training Method

The goal of neural networks is to simply identify backdoors rather than the shape of the entire object shown in the input image. Therefore, the training phase does not require the use of images containing objects. What is needed is a backdoor embedded image file. Two methods can be tested. The first is to create training data that has no shape but includes a backdoor. The second is to create training data containing random objects with backdoor embedded.

To make the neural network easy to find backdoor regardless of its size and position, preprocessing the training data may be required. The initial plan is to create a backdoor that is inserted into the entire image, and additional variation which contains the reduced shape and rotated shape of the backdoor. Even when copyrighted content is illegally streamed with altered dimensions, it's possible to train models to recognize a specific pattern hidden within a rectangle, considering that the overall image still maintains a rectangular shape.

4 Implementation

In this section, the basic concept of the experimental implementation is described. Full experimental code is accessible at [2].

4.1 Data Preparation

To validate the proposed idea, a moderate number of images are needed as well as fictional authors. For the image data, CIFAR-10 [14] dataset is used. The CIFAR-10 dataset consists of 60,000 32x32 color images in 10 different classes, with 50,000 training images and 10,000 test images. For this study, the dataset was divided into clean and backdoor-injected subsets. Therefore, data preparation process is needed in order to make backdoor-injected subset from the clean one.

Data Preparation Process The following steps outline the data preparation process:

- **1. Loading CIFAR-10 Data**: The CIFAR-10 dataset is loaded from its original batch files, converting the data into a suitable format for processing.
- **2. Backdoor Injection**: A specific message is converted into binary format and embedded into the least significant bits of the image pixels using steganography techniques.
- **3. Dataset Splitting**: The dataset is split into training and test sets, with a proportion of images containing the backdoor trigger.

4.2 Embed Watermarks into Images

Several methods of watermark embedding is tried . Most of the methods manipulated the pixel and its bit component. Taking idea from the steganography method, a method of dividing a specific message into bits and hiding it in the LSB of RGB pixels of the image was used. To verify the effectiveness of the backdoor injection, the testing ranged from simple and noticeable backdoor insertion methods to those containing complex messages that are not visible. Specific evolution of the embedding method is described in Section 5

4.3 Model Architecture

A pre-trained ResNet-18 model, a widely used CNN architecture known for its robust performance on image recognition tasks, was employed for this study. The model was fine-tuned on the CIFAR-10 dataset, modified to detect two classes: clean images and backdoor-injected images. Two basic modification is conducted. The first one is input layer adjustment. The input layer is adjusted to accept 32x32 images instead of the original 224x224 images used in ResNet-18. The second modification is about output layer. The final fully connected layer is changed to output two classes corresponding to the clean and backdoor-injected labels; label 0 for the clean one and 1 for the backdoor-injected one.

4.4 Training and Evaluation

The training process involved fine-tuning the ResNet-18 model on the prepared dataset, using standard data augmentation and normalization techniques. The mean and standard deviation for normalization were calculated from the dataset itself. The training and evaluation process includes two loops. The first loop is training loop. This is a loop that iterates through the dataset, updating model weights based on the loss computed from the training data. The second one is evaluation loop. This loop is a separate one that evaluates the model's performance on the test set after each epoch, recording loss and accuracy. After all training process is done, graphs indicating loss over epochs and accuracy over epochs are generated.

5 Experiment and Analysis

In this section, the experiments conducted using the implemented code are discussed along with their results and analysis. The experiments are divided into three main parts. Initially, a simple method of inserting a backdoor into an image is tested. Subsequently, a more advanced form of backdoor insertion is explored. The key difference between the first and second parts is that, in the first part, the backdoor inserting method is designed that allows the CNN to distinguish between the presence or absence of the backdoor. In the second part, multiple types of backdoors are randomly inserted into various images, and the CNN is tasked with identifying which specific backdoor is present. Finally, the third part builds on the first two methods to explain the insertion of author ID-based backdoors.

5.1 Simple Backdoor Injection

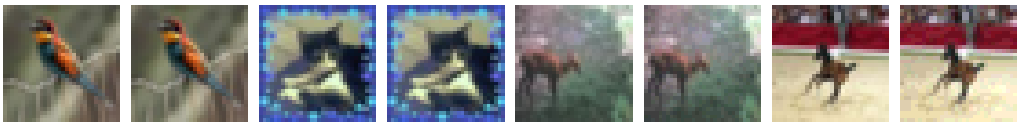
One of our primary objectives is to insert an invisible backdoor into the image, which can be achieved by making minor changes at the bit level. The simplest approach to embedding a backdoor is to modify the least significant bits (LSBs) to introduce these minor changes. This method leverages the fact that, according to Gupta et al. [10], altering up to 4 bits per pixel is difficult to distinguish by the human eye, yet can effectively embed hidden information. By setting the LSBs to 1, the data can be encoded into the image without significantly affecting its appearance. This straightforward technique serves as the foundation for our backdoor injection algorithm, which operates by systematically altering the LSBs across the entire image.

This simple algorithm is depicted in Algorithm 1, outlines a simple yet effective method for injecting a backdoor into an image by modifying its LSBs. The input to the algorithm is an image I with dimensions $(3, W, H)$, where 3 represents the color channels (e.g., RGB), and a bit position p that determines the number of LSBs to be modified. The output is the backdoor injected image I'. The algorithm begins by creating a mask with p bits set to 1. It then iterates over each pixel of the image and, within each pixel, over each color channel. For each color channel value, the algorithm performs a bitwise OR operation with the mask to set the specified LSBs to 1. This process effectively embeds the backdoor information into the image without significantly altering its visual appearance. Finally, the modified image I' is returned. This method ensures that the backdoor is subtly integrated into the image data, making it a practical approach for tasks requiring covert data embedding.

Algorithm 1 Simple Backdoor Injection (Image LSB Modification)

Input: Image I with shape $(3, W, H)$, bit position p
Output: Backdoor injected image I'

$I' \leftarrow I$
$mask \leftarrow (1 \ll p) - 1$
for each pixel (x, y) in I **do**
    for each color channel c in I **do**
        $I'[c, x, y] \leftarrow I[c, x, y] \mid mask$
    end for
end for
return I'



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 1: Comparison of Original and Backdoor Images for Various Bit Positions. (a) and (b) are images with bit position 1 labeled as 'bird' in CIFAR-10. (c) and (d) are images with bit position 2 labeled as 'cat'. (e) and (f) are images with bit position 3 labeled as 'deer'. (g) and (h) are images with bit position 4 labeled as 'horse'. In each pair, the left image is the original, and the right image is the backdoor-injected version.

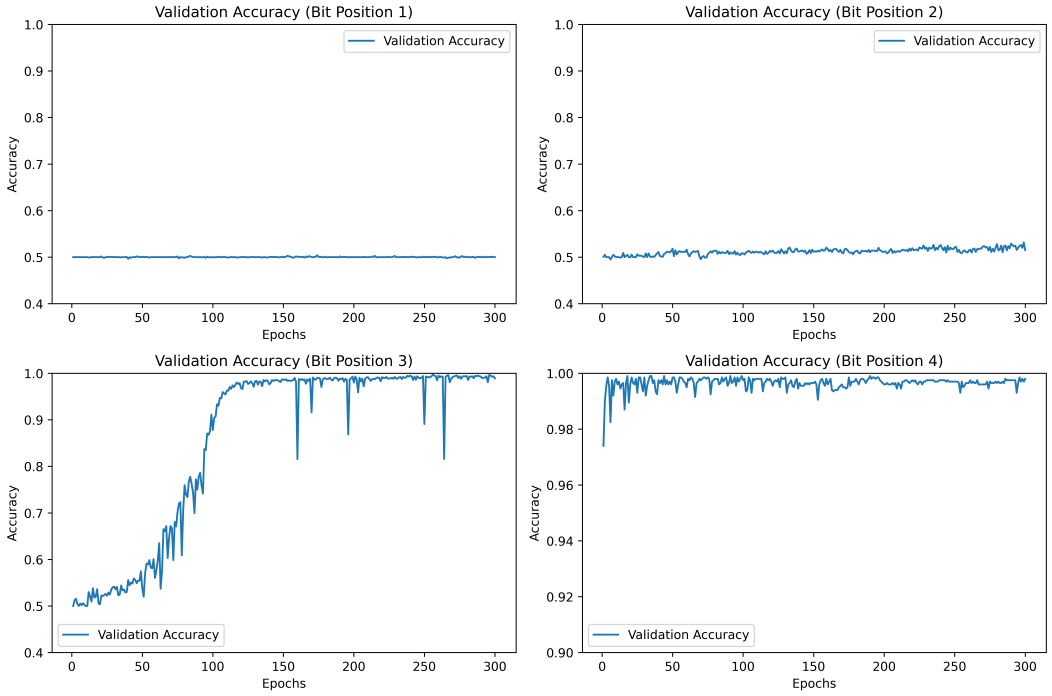


Figure 2: Validation Accuracy Across Different Bit Positions

To test this method, I experimented with bit positions ranging from 1 to 4. Before examining the test results, Figure 1 shows a comparison between the original images and the backdoor-injected images for each bit position. As can be seen, in all four cases, it is difficult to distinguish between the original and modified images with the human eye.

For this experiment, instead of using the pre-existing test set provided by CIFAR-10, I randomly selected 6,000 images from the 50,000 images in the training set. These were divided into a training set and a validation set, with a ratio of 5:1, respectively. The model was trained for a total of 300 epochs.

Figure 2 shows how the validation accuracy changes with epochs. The results indicated that when the bit position was 1, the validation accuracy remained constant at 0.5, regardless of the number of epochs. Since the task was to distinguish between images with and without a backdoor, an accuracy of 0.5 suggests no learning beyond random guessing. When the bit position was 2, the validation accuracy did increase with more epochs, but it only reached about 0.53 at 300 epochs, indicating minimal learning. However, for bit positions 3 and 4, the CNN showed significant learning. For bit position 3, the validation accuracy surpassed 0.95 after 110 epochs. For bit position 4, the validation accuracy was 0.97 from the first epoch and exceeded 0.99 after 10 epochs. This is an expected outcome because the higher the bit position, the greater the degree of modification to the image, making it easier for the CNN to detect the changes, even if humans cannot notice them.

To evaluate the effectiveness of this simple backdoor method and model efficiency, I conducted tests on bit positions 3 and 4 using actual train and test sets. The train set consisted of 50,000 images, while the test set comprised 10,000 images. The training criterion for each case was to train up to the epoch at which validation accuracy had previously exceeded 95%. For bit position 3, training required 110 epochs, whereas for bit position 4, only 1 epoch was necessary.

However, due to practical constraints, I limited the training for bit position 3 to only 10 epochs. This decision was influenced by the significant increase in training time with larger datasets; training for 110 epochs was estimated to take approximately 200 minutes. Although this duration is not excessively long, I observed that the validation accuracy for bit position 3 surpassed 0.95 within just 1 epoch when the train set size was 50,000, suggesting further training unnecessary.

Thus, I adjusted our original plan, training for 10 epochs for bit position 3 and maintaining the 1 epoch training for bit position 4 as initially planned. The results showed that the test accuracy for bit position 3 was 99.39%, while for bit position 4, it was 99.58%. It is important to note that these two cases cannot be directly compared in terms of performance since the number of training epochs and the criteria for their selection were fundamentally different. However, both cases demonstrated that the CNN could reliably detect the presence of a backdoor with high accuracy.

This finding is noteworthy because it confirms that the CNN can recognize the backdoor even in images it has not encountered before in the train set, simply by applying the rule of setting the lower 3 or 4 bits to 1. This result shows the model's ability to generalize and detect the backdoor under the specified conditions.

5.2 Improved Backdoor Injection

In this subsection, a method is devised to store backdoors in the form of messages, ranging from one to a maximum of ten messages, and validate the detection capability of CNN models for such backdoors. The method of inserting message-based backdoors is not significantly different from the earlier LSB manipulation technique. Instead of setting all LSBs to 1, the message is split into bits and inserted into the LSBs. Algorithm 2 illustrates this approach in pseudocode.

In the algorithm, each pixel's color channels are sequentially modified to embed a given message, leveraging a specified bit position across the image. The algorithm begins by initializing the output image I' as a copy of the original image I. It then converts the message m into a bit array m_bits and determines its length. Utilizing a defined bit position p, the algorithm iterates through each pixel and color channel of the image. For every iteration, a segment of p bits from m_bits is extracted and seamlessly integrated into the LSBs of the current pixel's color channels in I'.

Based on insights gained from previous experiments, common characteristics are established to be used testing message injection scenarios. Specifically, this approach is verified using a full dataset under the common condition of a bit position of 4 and one epoch. This choice aims to leverage the ability of the CNN model to achieve maximum backdoor detection while minimizing training time.

Algorithm 2 Improved Backdoor Injection (Message Injection)

Input: Image I with shape $(3, W, H)$, bit position p, message m
Output: Backdoor injected image I'

$I' \leftarrow I$
$m\_bits \leftarrow$ *bit array of* m
bit_length $\leftarrow$ *length of* m_bits
bit_index $\leftarrow 0$
$mask \leftarrow (1 \ll p) - 1$
for each pixel (x, y) in I **do**
    for each color channel c in I **do**
        bits_to_insert $\leftarrow m\_bits$[*bit_index* : (*bit_index* $+ p$)%*bit_length*]
        $I'[c, x, y] \leftarrow (I[c, x, y] \& \sim mask) \mid$ *bits_to_insert*
        bit_index $\leftarrow$ (*bit_index* $+ p$)%*bit_length*
    end for
end for
return I'

Number of Backdoors	1	2	5	10
Test Accuracy (%)	100.00	100.00	100.00	100.00

Table 1: Number of Backdoors and Test Accuracy (bit position 4, batch size 32, epoch 1)

Although this approach may exhibit inferior invisibility compared to a bit position of 3, the focus in this study is to maximize the potential of the CNN model and validate the feasibility of the proposed concept.

To test the ability to distinguish message-based backdoors, it was experimented with scenarios involving 1, 2, 5, and 10 different message-based backdoors. Each backdoor string consisted of 20 bytes and is long enough for the experiment, considering that it translates to a bit format. The results are summarized in Table 1. In all cases, CNN achieved an accuracy of 100.00% to distinguish backdoors, confirming that even if lsb is modified with different patterns, it is impossible to distinguish with human eyes, but the cnn model can distinguish it.

5.3 Author ID Based Backdoor Injection

In this subsection, there exists no concrete implementation. Based on the findings from previous experiments, the study explores generating author ID-based backdoor patterns and proposes their application for copyright protection by injecting these patterns into images. Drawing from the experimental results in Section 5.2, it was observed that injecting arbitrary messages as backdoor patterns allows CNNs to distinguish between different patterns with high accuracy. Therefore, by using a unique message m known only to the author to create the backdoor pattern, a unique backdoor pattern can be generated. Since the verification process is handled by CNNs, they only determine whether an image belongs to the author without revealing the specific backdoor pattern m. Such images with embedded backdoors are imperceptible to the human eye, making it impossible for others to mimic these patterns. Consequently, only the author can generate the same pattern, ensuring effective prevention against disguise.

6 Discussions

This section describes additional analyses and personal assessments that were not covered in the text.

6.1 Backdoor Invisibility

In this study, the backdoor insertion technique using LSB manipulation (both setting LSBs to 1 and embedding messages) is depicted in Figure 1, showing minimal perceptible differences to the human eye. However, increasing the bit position from 1 to 4 results in unavoidable changes in pixel RGB values. Moreover, a critical drawback of LSB manipulation is its impact on color diversity. Originally, a single pixel's RGB component can express values from 0 to 255, but with a bit position of 1, this range reduces by half, and further diminishes with higher bit positions (e.g., down to 16 possibilities with a bit position of 4). Thus, while imperceptible to human perception, there could be subtle degradation in image quality at a fine level.

Another noteworthy point is that in Section 5.2, different messages were inserted as backdoors without explicit comparative analysis afterward. This decision stemmed from the expectation that the level of comparison would closely resemble what was observed in Figure 1. Considering the fixed bit position of 4 for message insertion, the degree of change from the original image is probabilistically equal, whether all 4 LSBs are set to 1 or replaced by the bits of the message. Therefore, arranging images with different message injected and visually inspecting for differences would likely yield results similar to those in Figure 1.

6.2 Method of Backdoor Injection

Firstly, the LSB-based backdoor injection method employed in this study is among the simplest backdoor injection approaches. As mentioned in the preceding subsection, this method can impact image quality and cannot definitively guarantee a very high level of invisibility. Therefore, to enhance the practicality of this research and advance the technology for real-world copyright protection

applications, using backdoor injection methods that offer higher invisibility with minimal impact on image quality could be considered as a viable approach.

Apart from this, I thought about why the LSB manipulation method used in this study can successfully act as a backdoor. This capability lies within the convolutional layers of the CNN. For instance, when all specific LSBs are set to 1, there is minimal noticeable change in the color domain of pixels to the human eye. However, even subtle changes indicate that the color of those pixels has indeed been altered, presenting rare characteristics in numerical color values where specific LSBs are consistently set to 1. CNNs trained to distinguish such backdoors possess convolutional layers capable of identifying these types of color combinations. In the case of images manipulated with LSBs, the distinct numerical features in colors enable CNNs to detect them with high probability.

6.3 Experiment Environment

All experiments were conducted in the Google Colab Notebook environment [3]. The T4 GPU runtime provided by Colab was utilized, and the source code was developed independently with assistance from ChatGPT (GPT-4o) [1]. All Colab Notebook files(.ipynb) used for the experiments along with their results are available for review on [2].

6.4 Further Works

According to the original plan, this study aimed to propose a CNN model capable of detecting images with inserted backdoors, even considering variations in image size or rotation. While practical experiments were not conducted, outlining methods for such experiments would involve several steps.

Firstly, preparing datasets of both clean images and images with inserted backdoors remains unchanged. Additional steps would be introduced in the data preparation phase, specifically by generating new images with random size variations and rotations applied to these datasets. By including these additional images in the training and validation phases, the goal could be achieved.

However, even if an image with a backdoor undergoes size changes or rotations, the current CNN used in this study appears capable of detecting backdoors with high accuracy. This is because, as mentioned in the previous subsection, the CNN seems to recognize primarily the color areas that images with inserted backdoors predominantly exhibit, using convolution layers. Thus far, the CNN has accurately detected the presence of backdoors regardless of changes in object shapes, suggesting that it prioritizes pixel colors over object shapes when classifying images. Therefore, it is anticipated that this CNN will reliably detect backdoors in images even if they are slightly rotated or resized.

7 Conclusion

This study introduces a simple, yet effective method for safeguarding digital copyrights through backdoor watermarking, employing neural networks to embed and detect unique author-specific watermarks in images. The proposed technique ensures high accuracy in recognizing these watermarks thereby offering a reliable solution for authorship verification. The experimental results, validated using the CIFAR-10 dataset, highlight the method's robustness and potential for real-world application. Future research should focus on enhancing the invisibility and security of the watermarking process and exploring its application in various media formats to provide comprehensive copyright protection across the digital landscape.

References

[1] ChatGPT (GPT-4o). `https://openai.com/chatgpt/`.

[2] GitHub repository. `https://github.com/starbuucks/AI_sec_backdoor`.

[3] Google Colab. `https://colab.google/`.

[4] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115, 2020.

[5] Mauro Barni, Kassem Kallas, and Benedetta Tondi. A new backdoor attack in cnns by training set corruption without label poisoning. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 101–105. IEEE, 2019.

[6] Franziska Boenisch. A systematic review on model watermarking for neural networks. *Frontiers in big Data*, 4:729663, 2021.

[7] DataProt. Piracy is back: Piracy statistics for 2024, 2024.

[8] Hao Fu, Akshaj Kumar Veldanda, Prashanth Krishnamurthy, Siddharth Garg, and Farshad Khorrami. Detecting backdoors in neural networks using novel feature-based anomaly detection. *arXiv preprint arXiv:2011.02526*, 2020.

[9] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

[10] Shailender Gupta, Ankur Goyal, and Bharat Bhushan. Information hiding using least significant bit steganography and cryptography. *International Journal of Modern Education and Computer Science*, 4(6):27, 2012.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *2018 IEEE symposium on security and privacy (SP)*, pages 19–35. IEEE, 2018.

[13] Md Abdul Kadir, GowthamKrishna Addluri, and Daniel Sonntag. Revealing vulnerabilities of neural networks in parameter learning and defense against explanation-aware backdoors. *arXiv preprint arXiv:2403.16569*, 2024.

[14] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[16] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[17] Jae-Eun Lee, Young-Ho Seo, and Dong-Wook Kim. Convolutional neural network-based digital image watermarking adaptive to the resolution of image and watermark. *Applied Sciences*, 10(19):6854, 2020.

[18] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[19] Rui Ning, Jiang Li, Chunsheng Xin, and Hongyi Wu. Invisible poison: A blackbox clean label backdoor attack to deep neural networks. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, pages 1–10. IEEE, 2021.

[20] Zhanglin Peng, Lingyun Wu, Jiamin Ren, Ruimao Zhang, and Ping Luo. Cuimage: A neverending learning platform on a convolutional knowledge graph of billion web images. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 1787–1796, 2018.

[21] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[22] Sebastian Szyller, Buse Gul Atli, Samuel Marchal, and N Asokan. Dawn: Dynamic adversarial watermarking of neural networks. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 4417–4425, 2021.

[23] Alireza Tavakoli, Zahra Honjani, and Hedieh Sajedi. Convolutional neural network-based image watermarking using discrete wavelet transform. *International Journal of Information Technology*, 15(4):2021–2029, 2023.

[24] Hanzhou Wu, Gen Liu, Yuwei Yao, and Xinpeng Zhang. Watermarking neural networks with watermarked images. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(7):2591–2601, 2020.

[25] Yi Zeng, Si Chen, Won Park, Z Morley Mao, Ming Jin, and Ruoxi Jia. Adversarial unlearning of backdoors via implicit hypergradient. *arXiv preprint arXiv:2110.03735*, 2021.

[26] Haoti Zhong, Cong Liao, Anna Cinzia Squicciarini, Sencun Zhu, and David Miller. Backdoor embedding in convolutional neural network models via invisible perturbation. In *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy*, pages 97–108, 2020.