How to Protect Intellectual Property of Deep Neural Networks

Nicolas Robertson
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA 56267
robe1606@morris.umn.edu

ABSTRACT

Deep neural networks have made significant progress towards making tasks normally impossible for a computer possible through deep learning. These frameworks are important enough to warrant protecting them from thieves. However, efforts to protect DNNs from fraudulent usage have been insufficient. Current methods for watermarking DNNs cannot clearly associate any given framework with its author and are too easily replicated by thieves.

Li et al [3] have developed a watermarking method known as the IPP (Intellectual Property Protection) Blind-Watermark Framework, which solves problems present in other watermarking techniques. It uses exclusive logos to clearly associate a DNN with its author, and uses generative adversarial networks to make the watermark imperceptible to the average human. It also fulfills all of the evaluation requirements set by the team: fidelity, effectiveness and integrity, security, legality, and feasibility.

Keywords

watermarking, deep neural networks, security, generative adversarial networks. software, privacy

1. INTRODUCTION

Intellectual property (IP) refers to creations of the mind. This includes a wide range of work such as art, tools, computer software, music, etc. These works are important enough to have laws protecting the rights of intellectual property owners. These rights often give people incentive to create IP, especially when the goal is to make a profit from their work. However, laws alone are not enough. Signals of ownership must be created to verify whether someone has the right to use a specific property.

This is where watermarks becomes important. Digital watermarking is the act of embedding data into digital objects such as video, websites, etc. The watermark data becomes a permanent part of the content and cannot be removed, making the identity of the owner clear even after the content has been distributed to other parties.

This paper will focus on protection of deep neual networks (DNN). Deep learning technology has made significant leaps forward in recent years. Deep neural networks, also referred to as models, can be used to process data which computers historically couldn't process. This includes tasks such as identifying objects in images, recognizing specific sounds in audio files and other unstructured data which typically requires human intelligence.

The emerging models of deep learning are facing privacy and security issues. Some companies rely upon their deep neural networks as their source of income. Thus, it is important for these new deep neural networks to be watermarked so the intellectual property of the owners is protected. Without technical means of ownership identification, the legal system will have little to stand on when protecting the IP rights of DNN owners.

This paper will outline how watermarking can be implemented in deep neural networks to allow owners of these models to protect their IP. Section 2 will provide background details which will be necessary for understanding the material. This will include information about deep neural networks, the creation of digital watermarks and generative adversarial networks. Section 3 will explain what the motivation for IP theft is and how bad actors can steal models from their owners. Section 4 will dive into the specifics of the IPP blind-watermark framework. Section 5 will outline how the framework was tested. Section 6 will evaluate the results of testing the IPP framework. Finally, Section 7 will present conclusions about the current state of the research.

2. BACKGROUND

In this section, background knowledge necessary for understanding the IPP blind-watermark framework will be given. This includes deep neural networks and digital watermarks.

2.1 Deep Neural Networks

To understand deep neural networks (DNN), it is necessary to understand artificial intelligence. Artificial intelligence is the ability for machines to make decisions independently. This intelligence is normally explicitly programmed. Machine learning is a sub-field of artificial intelligence which involves machines gaining intelligence without explicit programming. An example of machine learning is a system where a students grades are predicted based on patterns observed in previous grades. [4]

Despite working well with a variety of problems, machine learning struggles with tasks such as identifying objects in an image or sounds in an audio file. *Deep Learning* is a sub field of machine learning which attempts to mimic how the human brain processes information to make tasks such as this possible for a machine.

UMM CSci Senior Seminar Conference, April 2020 Morris, MN.

Deep neural networks use an interconnected network of "neurons" which mimic neurons in the human brain, albeit with a mathematical approach. Like a human, these networks need to be trained to perform the desired task. For example, a DNN which is meant to recognize faces would be fed many different images of human faces. The DNN tries to label these images correctly and the training system will inform it whether it is correct or incorrect. These results will then help the DNN adjust how it handles data so its facial recognition can become more accurate. The specific variables which are changed in order to adjust the DNNs performance are called weights. These are set before the network is trained and continuously adjusted after each iteration of training [5]. Continuous training can improve the performance of the DNN, ideally resulting in a near-perfect performance.

2.2 Digital Watermarks

Traditional watermarks are a name/logo embedded into paper. For example, an image may have the name of the photographer printed in transparent text on it. They are used to confirm authenticity.

Digital watermarks are the electronic counterpart to the traditional variety. It involves embedding data into digital content which identifies the owner. Unlike traditional watermarking, digital watermarking can also be used to monitor how the content is being used by others. They can also provide useful information about the owner to anyone using the content.

2.3 Generative Adversarial Networks

Understanding the IPP framework requires a basic understanding of generative adversarial networks. The goal in machine learning is for the model being trained to accurately label the input being fed to it. For example, a model which is meant to recognize images of animals would need to be able to identify animals accurately. A generative adversarial network (GANs) is a structure where two models, the generator and discriminator, compete with each other. The discriminator wants to discern something about the generator's output while the generator wants to generate an output which can fool the discriminator. This process can be framed as a two player game. [2] In GANs the original input is referred to as the original/ordinary samples, and the generator's output is referred to as the synthesized/key samples.

For example, the generator wants to create images with well hidden watermarks and the discriminator wants to identify images which have been watermarked. The generator takes images as input and outputs watermarked images, then passes both sets to the discriminator. The discriminator identifies how likely it is each image is watermarked and outputs this set of classifications. Depending on how accurately the discriminator guesses which images are watermarked, both models will adjust their weights. The generator adjusts itself to better hide its watermarks and the discriminator adjusts itself to better identify watermarks.

In GANs the goal is for the generator to eventually create key samples which are convincing enough that the discriminator can not tell which images are key samples or not. This would result in the discriminator returning "unsure" (50 percent synthesized and 50 percent real) for each sample. Each round of the process will bring the generator closer to this goal as its weights are adjusted.

3. MOTIVATION

The forms of attack which the watermark is meant to protect against will be explained here. The primary forms of attacks discussed will be evasion attack and fraudulent claims of ownership.

3.1 Security: Evasion Attack

There are three players in this scenario: the owner of the DNN model Alice, the thief Eve, and Bob who has purchased the model from Alice. The two instances where an evasion attack can occur are (1) Eve has stolen the DNN or (2) Bob has resold to the model to Eve without Alice's permission.

In either case, if Alice were suspicious that someone has stolen her model she would send watermarked key samples to the DNN to confirm that the stolen model belongs to her. Eve would be able to avoid the verification by building a detector which can identify possible key samples and return a random label rather than having labeling it as a key sample. The IPP framework presented by the researchers aims to defend against this attack by making the model key samples themselves undetectable as key samples.

3.2 Legality: Fraudulent Claims of Ownership

In this scenario, Oscar is a counterfeiter who has tried to claim ownership of Alice's model and sell it as his own. As a counterfeiter, it would be in Oscar's best interest to make his own set of key samples which will trigger the key sample detection built into the DNN model he's stolen.

Previous watermarking methods use obvious means of embedding watermarks. [2] This makes it trivial for people like Oscar to build fake samples which verify fraudulent ownership. The IPP framework attempts to develop a method of watermarking which is significantly more subtle and more difficult to replicate.

4. IPP BLIND-WATERMARK FRAMEWORK

The following section gives an overview of how watermarks are embedded and verified in Li et al's IPP framework [3], followed by the details of how the algorithm functions. The IPP blind-watermark framework consists of three parts: the encoder, discriminator and host DNN. See Figure 1. The encoder and discriminator work to create the watermark behavior (the DNN identifying certain images as key samples) and the host DNN is whatever DNN someone wants to have watermarked by the framework.

4.1 Task I: Embedding

A set of ordinary samples x, essentially a set of images, which have been chosen to be watermarked is required for the embedding procedure. An encoder e will take sample x and a logo l (the logo used to watermark the images) as inputs when generating a watermarked image x^{key}. The algorithm G which generates x^{key} appears as follows:

$$x^{key} = G(e, x, l) \quad (1)$$

The goal is for the key samples to appear identical to the ordinary samples. An embedding algorithm E is then be used to watermark the host DNN f:

$$f_k = E(f, x^{key}) \quad (2)$$

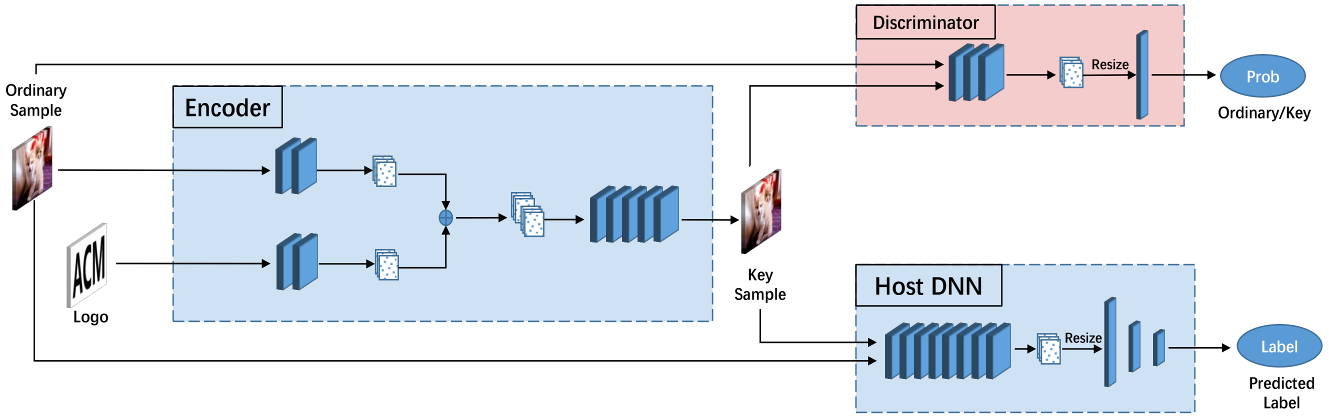

Figure 1: Workflow of IPP framework [3]

The watermarked model f_k will predict labels for images of key sample x^{key}, i.e. labelling a watermarked image of a dog as a key sample rather than a dog image. This label which denotes an image as a key sample will be known as t^{key}.

4.2 Task II: Verification

In a case where the owner of a DNN model (Alice) suspects someone has used their model without permission, they can use the watermark to test their suspicion. The model which is suspected to have been stolen will be referred to as “the remote model”. Alice can prepare a set of key samples $({x_1}^{\text{key}}, {x_2}^{\text{key}}, ...)$ by using the generation algorithm G.

The model owner can then test these key samples by having the remote model g predict the labels for the key samples. If the results come back close to 1, i.e. the remote model labeled the key samples as key samples, then their suspicion will be confirmed. This will provide strong evidence that the remote model was stolen.

4.3 Algorithm Pipeline

The IPP framework is essentially putting GANs into practice. There are four key pieces of the algorithm which need to be balanced with each other.

Encoder: The purpose of the encoder is to produce the key samples which function as the watermark. The ordinary samples x and logo are taken as input and key samples x^{key} which are nearly indistinguishable from the ordinary samples are generated.

The goal of the encoder is to obtain a near-perfect reconstruction, i.e., $\text{x} \approx \text{x}^{\text{key}}$ rather than $\text{x} = \text{x}^{\text{key}}$. The small difference between the key and ordinary samples provides a balance between security and effectiveness. A smaller difference provides better protection against evasion attacks and a larger one providing more effective watermarking of the DNN. The error in the reconstruction will be denoted as:

$$argmin(\text{e's reconstruction error}) \tag{3}$$

Argmin is a function which tries to minimize whatever is being passed into it. From a mathematical perspective Eq (3) is meant to minimize the quantitative difference between the ordinary and the key samples. Essentially, minimizing the likelihood of a computer being able to detect the difference between the two.

Although Eq (3) accounts for the reconstruction error of e, it ignores how the underlying structure of the image is affected. The reconstruction can lead to loss in image quality, making the presence of a watermarking technique more visually apparent. The following equation represents the algorithm's attempt to minimize the image quality degradation during the process:

$$argmin(\text{e's image quality degradation}) \tag{4}$$

Essentially, this equation is attempting to minimize the qualitative difference between the ordinary and key samples. This would make it harder for a human to visually detect the difference.

Discriminator: The goal of the discriminator is to determine whether samples fed to it are synthesized or part of the ordinary samples. Its purpose is the same as its role in general adversarial networks. It essentially detects whether input data was generated by the encoder. Like the encoder, the discriminator accepts x (ordinary samples) and x^{key} (key samples) as input, and labels each sample with a probability of whether its a key sample. This set of probabilities is used in an argmin argument to minimize the probability of the discriminator making a mistake, the output of the argmin is denoted as d. Once this is obtained, it will be used to train the encoder to maximize the likelihood of the discriminator making a mistake. The objective function of encoder e will be denoted as:

$$argmin(\text{d does not make a mistake}) \tag{5}$$

Host DNN: The purpose of this framework is for any DNN in need of protected to be able to be watermarked by the IPP Framework. The host DNN represents the DNN being watermarked. Since the key samples from the encoder will only be close to the original samples, the host DNN will use the small difference between the two to identify x^{key}.

The following equation makes a summary of the test results (how the discriminator makes its guesses). The final layer of neural networks makes the discriminator's score values usable by turning all values into probabilities between 0 and 1 [6]. This probability vector will then be used to

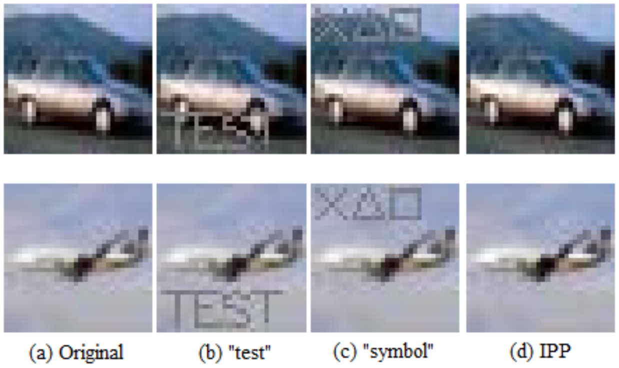

Figure 2: Examples of key samples of existing watermark methods and the IPP framework [3]

update e and d to better perform their tasks. The objective function for the host model h will be denoted as:

$$argmin(\text{h's parameters for updating encoder e}) \quad (6)$$

Equations 3, 4, 5, and 6 can be added together to produce the objective function O_e for encoder e:

$$\begin{aligned} &argmin\{ \\ &\quad a(\text{e's reconstruction error})+ \\ &\quad b(\text{e's image quality degradation})+ \\ &\quad c(\text{d does not make a mistake})+ \\ &\quad d(\text{h's parameters for updating encoder e}) \\ &\} \end{aligned} \quad (7)$$

$a, b, c, d > 0$ are weights which trade-off between the four parts to achieve the best balance between them [3]. This process should result in the difference between the ordinary and key samples being minimized while also still being detectable by the host DNN. This will allow the host DNN to identify key samples, confirming the ownership of the author while not making the watermarks obvious to outsiders.

5. IMPLEMENTATION

This section will go over the tools Li et al used to implement the IPP framework and evaluate how it performed under various tests.

5.1 Datasets and DNNs

The IPP framework was tested on two data sets: MNIST and CIFAR-10. MNIST consists of 28 x 28 pixel images which are handwritten samples of single digits 0 to 9. It has 60,000 training samples and 10,000 test samples. CIFAR-10 is a data-set with 80 million tiny color images. There are 50,000 training images and 10,000 test images. All images were normalized and centered in 32 x 32 pixel fixed images. See Figure 2 for examples of two CIFAR-10 images.

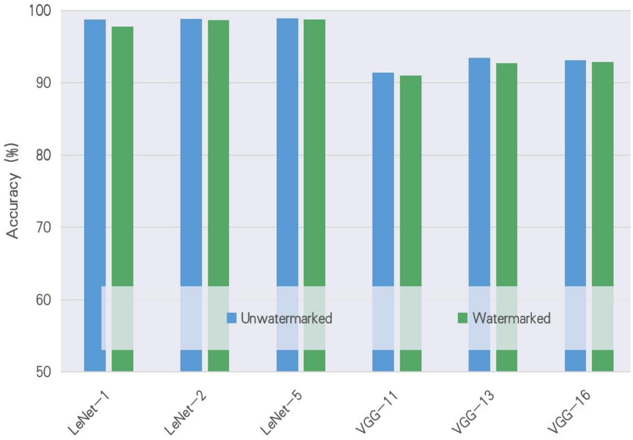

Figure 3: Accuracy of different models on regular test set (subset of data) [3]

5.2 Results

The IPP framework was successfully trained and the key samples it generated were compared to existing models. Figure 2 shows some samples generated from the CIFAR-10 dataset. Figure 2 (a) shows the original, (d) shows the key sample generated by the IPP framework, and (b)/(c) show examples of more obvious watermarking techniques used by other methods. The difference between the original and the IPP key samples are too subtle to be perceived by most humans. In contrast, the watermarks on the other are obvious and can be easily detected visual by humans.

6. EVALUATION

The performance of the IPP framework was analyzed using the following criteria:

- fidelity - is the primary classification task affected by side effects of the watermark?
- effectiveness and integrity - how successfully can the watermark be used to verify the host DNN?
- security - can it defend against evasion attacks?
- legality - can it defend against anti-counterfeiting?
- feasibility - can it resist model modification and associate the model with its real creator?

6.1 Fidelity

To determine whether the blind-watermark method affected the model's ability to correctly classify images, an evaluation was carried out on various DNNs. In Figure 3 the x-axis displays various DNNs used in testing and the y-axis plots the accuracy of those DNNs when performing their tasks. The blue bars denote performance before being watermarked, while the green bars represent performance after being watermarked. As seen in Figure 3, the DNNs are evaluated before and after being watermarked.

The results demonstrate that both situations yielded practically the same accuracy. Decreases in accuracy averaged

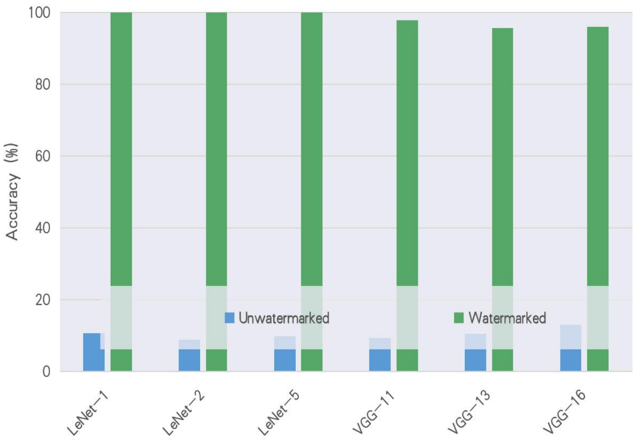

Figure 4: Accuracy of models watermarked with the IPP framework when identifying key samples (subset of data) [3]

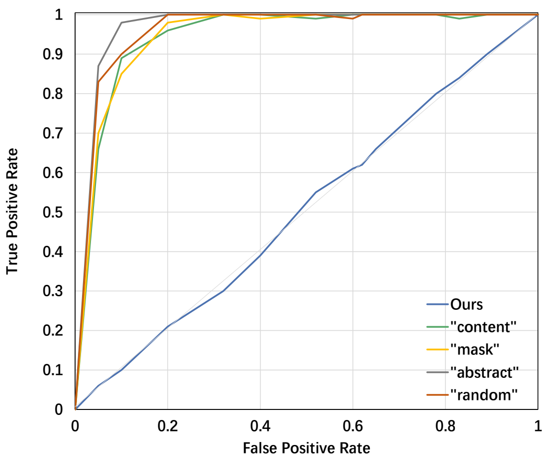

Figure 5: The ROC curve produced by the detector when tested on 3 layers ("Ours" refers to IPP framework)[3]

between 0.66% and 0.14%. The side effects of the watermarking had no significant impact on the primary task of image identification. So the IPP frameworks meets the fidelity requirement.

6.2 Effectiveness and Integrity

Effectiveness is concerned with whether a DNN which has been watermarked by the IPP framework can accurately identify key samples. The integrity of the framework requires that unwatermarked models do not recognize key samples which they should not recognize. To measure effectiveness and integrity, a set of DNNs were tested against queries of key samples. The DNNs were tested in their original state, followed by testing after they had been watermarked.

Figure 4 shows the accuracy of different models when performing their tasks (such as classifying images of animals, for example) during evaluation. The unwatermarked models achieved an accuracy of between 9% and 13% while the watermarked models peaked with an accuracy of 100%, most landing around at least 90%.

6.3 Security

The security of the IPP framework depends on whether the verification process is undetectable by unauthorized service providers. The developers of the IPP framework assume the attacker can access up to 10% of the total training samples used to train the model to detect key samples. The attacker can also generate the same number of key samples as generated by the IPP framework and other proposed models [3]. The developers of the IPP framework built a simple neural network based detector for testing the presence of a watermark technique within DNNs, which utilized three 3 fully-connected layers.

ROC (receiver operating characteristic) is used to report the results of testing DNNs against the detector. ROC curves show how a classification model performs when classifying data. The curve plots two parameters: true positive rate and false positive rate. Having more true positives, meaning a guess "true" is correct, skews the curve towards the upper left. Having more false positives, meaning a guess "false" is incorrect, skews the curve towards the lower right. In most instances classification models want the curve to skew towards true positive, indicating an accurate classification. However, the IPP framework wants to have a straight line which doesn't skew either direction, indicating that the detector cannot accurately distinguish ordinary from key samples.[1].

In Figure 5 there is a clear difference between the performance of the IPP framework and the other DNNs that were tested. The curve for the IPP framework was nearly straight, indicating that the detector's performance was close to random guessing. Meanwhile, the other models skewed heavily towards true positives. This means the detector had no issues identifying the presence of other watermark methods.

Most current classification models play the role of feature extractors as well [3]. To test how the IPP framework performed against these models, the developers tested against a commonly used ResNet-18 feature extractor. As seen in Figure 7, the IPP framework is only slightly more detectable than in the 3 layer test and the other watermarking systems are still significantly easier for the detector to notice.

6.4 Legality

Preventing an attacker from inducing the behavior of the watermarked DNN with their own set of fake samples is essential for protecting the rights of Alice and Bob. If an attacker can accomplish this then Alice can no longer claim ownership, which also infringes on Bob's interests. The IPP framework needs to resist two types of fraudulent claims of ownership.

What if the original and key samples were obtained by an attacker? The goal of the IPP framework is to make the difference between the ordinary and key samples imperceptible. Other models studied have watermark techniques which are obvious and can be easily replicated [3]. As can be seen in Figure 2, the difference between the original and IPP images cannot be recognized by the average human. The distortions made to the images are also unique for each image, making it nearly impossible to simply design a set of fake samples by manually mimicking the effect the watermark has on the ordinary samples [3].

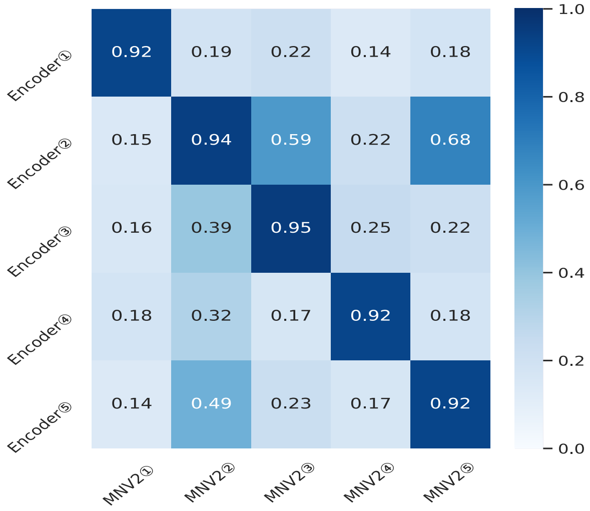

Figure 6: Performance (accuracy) of key samples under transferring attack [3]

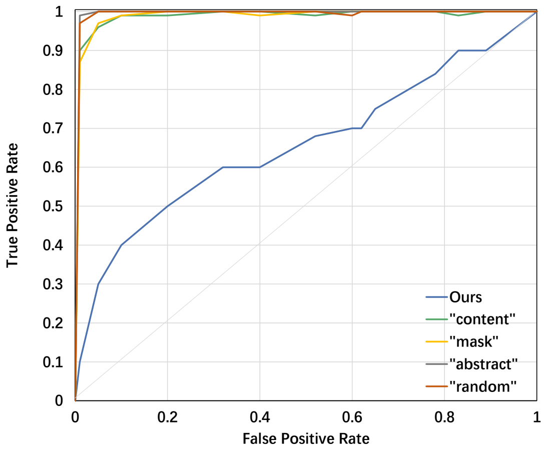

Figure 7: The ROC curve produced by the detector when tested on ResNet-18 ("Ours" refers to IPP framework)[3]

What if the encoder was leaked? An attacker might attempt a transfer attack, where they train their own samples using a remarkably similar encoder to the encoder which was leaked. The IPP framework was tested for this by simulating a scenario where a "same" encoder using the same dataset, architecture, and parameters is used to make fake samples. They watermarked a DNN, MobileNetV2, with the IPP framework five times using a different set of key samples each time. This resulted in five pairs of key samples and MobileNetV2 where they match, indicated by matching numbers in Figure 6. The x-axis represents samples generated by the encoder being sent and the y-axis represents which model the samples are being sent to. It can be observed in Figure 6 that the diagonal where the pairs match shows a high accuracy. This means the models labeled the samples correctly when they were fed the key samples used to train the encoder. The transfer attack does not work because the initialization of the neural network is crucial to the training process [3].

epochs	V-13	V-16	R-18	R-34	PreActR-18
0	95.75%	96.50%	97.00%	93.40%	91.75%
10	92.50%	95.50%	92.00%	90.70%	90.75%
20	90.25%	95.25%	91.75%	89.75%	90.00%
30	90.00%	95.50%	90.50%	88.75%	89.25%
40	90.00%	95.20%	89.75%	88.50%	88.50%
50	90.00%	95.75%	89.50%	87.75%	88.25%
60	90.00%	95.25%	89.00%	87.00%	88.25%
70	90.15%	95.00%	88.00%	86.75%	88.00%
80	90.00%	95.25%	87.75%	86.25%	87.50%
90	90.25%	95.50%	87.50%	85.50%	87.50%
100	90.00%	95.50%	87.50%	84.75%	87.00%

Figure 8: Accuracy of IPP framework in robustness test [3]

6.5 Feasibility

Robustness It's possible for an attacker to fine-tune a stolen model by training it on new datasets, creating a new model which has the characteristics of the previous while being distinct.

In an experiment to test the framework's robustness, the IPP framework was fine-tuned through different five datasets: V-13, V-16, R-18, R-34, and PreActR-18 as seen in figure 8. The IPP framework retained the highest accuracy due to the model not changing significantly as a result of the fine-tuning.

Functionality The IPP framework should clearly associate the a DNN with its author. One problem which neural networks face is the problem of over-fitting, where a neural network is trained too intensely on one data set and its behavior doesn't generalize to other data sets. The IPP framework uses this downfall to its advantage by over-fitting the watermarking technique, ensuring that its behavior will only trigger when presented with authentic key samples. This guarantees association between a DNN and its author.

7. CONCLUSION

The authors of the IPP blind-watermark framework have created a robust watermarking algorithm for deep neural networks. Their tests have shown that the framework verifies the author's ownership of the DNN without degrading its performance. A watermarked model is able to identify key samples while unwatermarked models don't falsely recognize key samples. The framework can successfully evade detection of its algorithm. The algorithm can also defend against fraudulent claims of ownership. Lastly, the framework's training process can't be replicated by third-parties.

8. REFERENCES

[1] Classification: ROC Curve and AUC. `https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc`. Accessed: 2021-03-21.

[2] J. Brownlee. A gentle introduction to generative adversarial networks (GANs). `https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/`, July 2019.

[3] Z. Li, C. Hu, Y. Zhang, and S. Guo. How to prove your model belongs to you: A blind-watermark based framework to protect intellectual property of DNN. In *Proceedings of the 35th Annual Computer Security Applications Conference*, ACSAC '19, page 126–137, New York, NY, USA, 2019. Association for Computing Machinery.

[4] Moolayil. A layman's guide to deep neural networks. `https://towardsdatascience.com/a-laymans-guide-to-deep-neural-networks-ddcea24847fb`, July 2019.

[5] P. Radhakrishnan. What are hyperparameters? and how to tune the hyperparameters in a deep neural network. `https://towardsdatascience.com/d0604917584a`, August 2017.

[6] T. Wood. What is the softmax function? `https://deepai.org/machine-learning-glossary-and-terms/softmax-layer`. Accessed: 2021-03-20.