Fooling image copy detection algorithms with GANs

Anton Ponomarev
aponom22@stanford.edu
https://github.com/ant-po/CS229project

ABSTRACT

In this paper, we explore the application of Generative Adversarial Networks in the attempt to intentionally mislead a state-of-the-art image copy detection (ICD) algorithm. We devise a model that, given an original image as input, seeks to generate a new image, one that is clearly distinct from the original, but which would be considered as a true copy by the ICD algorithm. We investigate a number of network architectures and find evidence that a configuration with two competing neural networks is capable of achieving the desired outcome. We then discuss the limitations of our experiment and general experience of training such a model. We conclude by making a suggestion on how to improve the resilience of the considered ICD algorithms to such adversarial attacks.

INTRODUCTION

Data is king. This has never been closer to reality than today. Digital data is generated, stored and used in our daily lives now at a rate unimaginable just a few years ago. In fact, it is estimated that more than 90% of all the digital data ever created is no more than three years old. This phenomenon, coupled with the technological advances and rising adoption of machine learning and falling cost of compute and memory storage, is the driving force behind the growing number of impactful decisions being outsourced to the machines that solely rely on the digital information.

However not all data is original, a large portion is derivative in nature. Any barriers that existed before to prevent attempts to duplicate, modify or corrupt raw digital data are now practically non-existent. This unencumbered ability to generate infinite versions of the original content fundamentally challenges our a priori understanding of what authenticity and provenance mean in this new digitized world. With intelligent systems penetrating all aspects of our lives, from grocery shopping to judicial verdicts, actors with malicious intent can cause significant harm. Ironically, the same technologies that power the modern world, also pose an existential threat to it in the wrong hands.

In this project, we focus on the specific task of determining whether two given images are carbon copies of each other. This seemingly trivial process forms the backbone of most image authentication algorithms. We consider the type called Image Copy Detection (ICD) algorithms [1], designed to identify whether two images are copies of each other. It is part of a wider family of algorithms dedicated to image authentication. We investigate whether a common state-of-the-art ICD algorithm is susceptible to being fooled by purposefully synthesized adversarial content. Our key motivation is that by understanding how a malicious attack can be realised, we can introduce appropriate modifications making these algorithms more resilient to such attacks.

The rest of the report is divided into four sections: we discuss the data used in this work and provide a general overview of a typical ICD algorithm. We then describe several model architectures that were explored followed by the discussion of the results and general observations. We close with sharing our ideas for future research.

EXPERIMENT SETUP

Since the emphasis of this work is on the model architecture rather than the data, we chose to use the well-known MNIST dataset. It is freely available online[1] and comes with pre-labelled images of hand-written digits. The grayscale colour pallet and low resolution make this dataset a solid starting point for training complex models, given the time and compute constraints of this project.

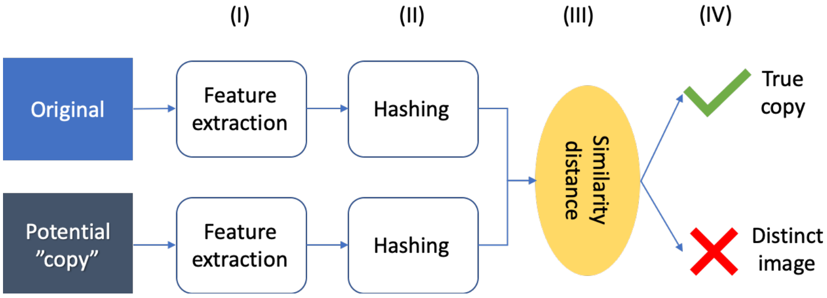

Figure 1

In order to simulate an ICD algorithm, we consider a process as described in Figure 1. Given two images – an original and a potential copy – we perform a sequence of transformations:

(I) Features are extracted from both images
   - We use block average – average pixel value is calculated for each non-overlapping block of the original image (we use block size of 7 pixels) – simple and fast

(II) Perceptual 64-bit hashes are created from the features
   - These are different to cryptographic hashes in that the intention is for images with similar features to have similar perceptual hashes

(III) Similarity metric is calculated on the two hashes
   - We use Hamming Distance (HD) which measures the number of corresponding bits that are different in the two sequences

(IV) If HD is below a pre-defined threshold (usually set around 10-15), then the potential copy is considered a *true copy*. Otherwise it is labelled as a *distinct image*

Figure 2 demonstrates how the described ICD algorithm can be applied to our dataset.

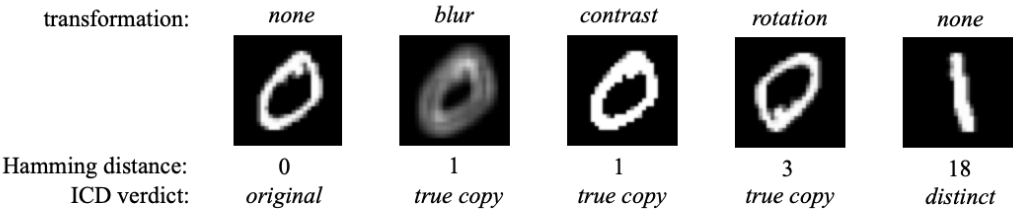

Figure 2

[1] http://yann.lecun.com/exdb/mnist/

Starting from the original image on the left, we compare a number of “copies” that were generated by applying various transformations (above the image) and calculate the Hamming distance relative to the original image (below the image). We can see that transformations of the original image still result in a similar perceptual hash, while a contextually different image results in a different hash. This highlights the desired qualities in the ICD algorithm: *robustness* – ability to resist pre-processing attacks, and *discrimination* – ability to identify images that are contextually different. Using this framework, we attempt to generate a “fake” copy of the original that passes successfully through the ICD algorithm’s test.

MODELS

Since our model needs to generate an image, we take inspiration from the family of Generative Adversarial Networks (GANs). In effect, we want to retain the Generator component, while changing how the Discriminator works to satisfy our new objectives. To be more precise, our aim is to generate a synthetic image with two qualities:

1. visually distinct from the original
2. small HD relative to the original

In order to accomplish the first objective, we introduce a separate image, called “parent image”, which will act as the style source, similar to the intuition behind neural style transfer [2]. Given the adversarial nature of our experiment, we do not feel that a typical Baseline model exists. We therefore proceed investigate two potential architectures.

Model 1

This is similar to the original GAN in that two neural networks are competing with each other:

- *“Real”* neural network seeks to minimize the difference between the input (synthesized) and the parent images
- *“Hash”* neural network minimizes the Hamming distance between the relevant perceptual hashes of the synthesized and the original images
- *“Real”* network feeds into *“Hash”*

We initialise the input from Gaussian noise and the training process is iterated over multiple epochs, until both networks converge. We found that around 200 epochs is sufficient.

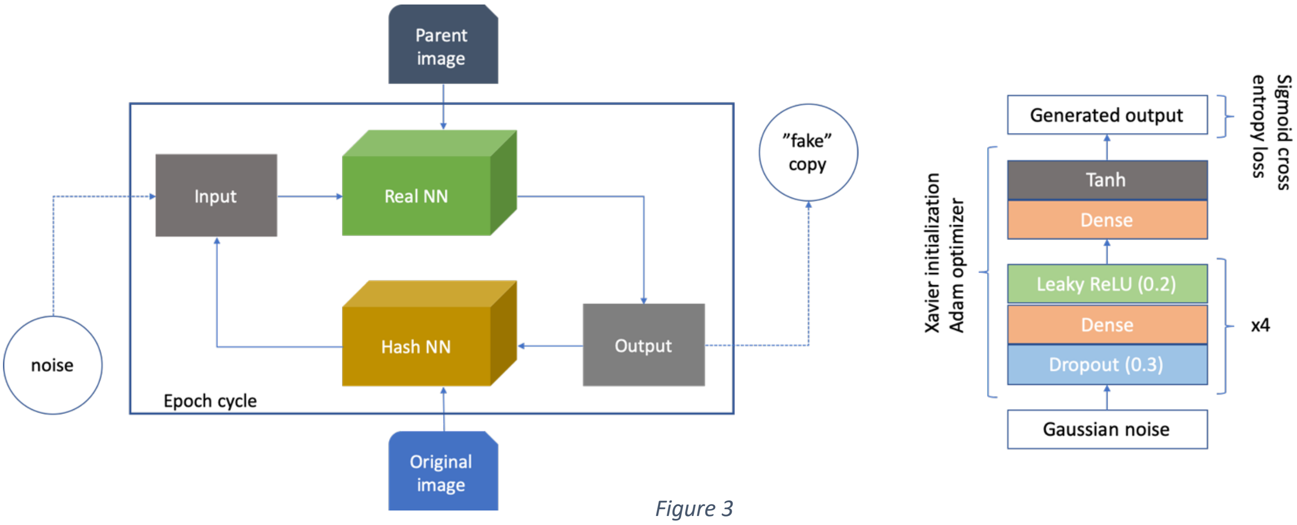

Figure 3

Model 2

We also consider a modification to Model 1. What we found from working with it, is that one network can easily overpower the other forcing the outcome to gravitate towards quality 1 or 2. This is undesirable, as the synthesised image ends up converging to either the original or the parent image. The ideal outcome would prioritise quality 2 a bit more than quality 1. As can be seen from Figure 4, in Model 2, instead of feeding one network into another, we train them concurrently using the same input. At the end of each epoch, we weight the outputs from both networks in a particular way. Specifically, we start with equal weights assigned to both networks, with gradual down-weighting of the *"Real NN"* as the number of elapsed epochs grows. This ensures that over time the impact of *"Hash NN"* will eventually dominate.

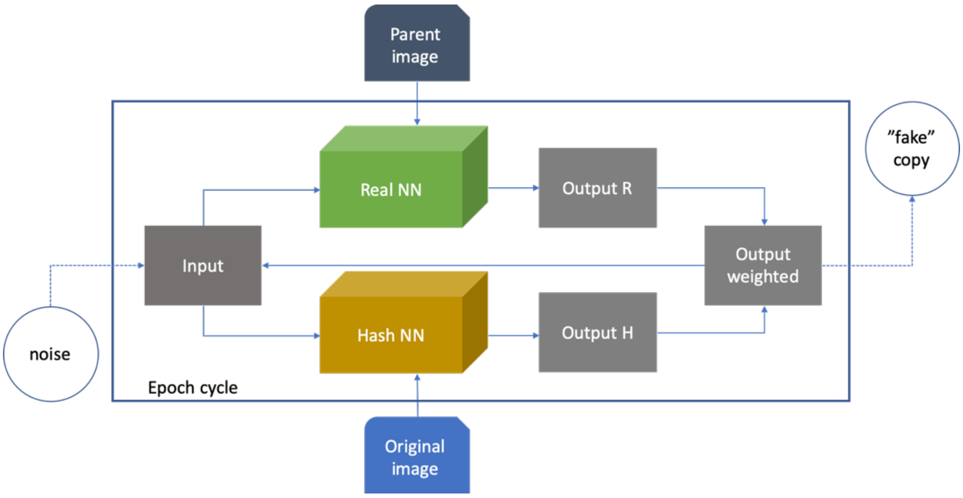

Figure 4

RESULTS & OBSERVATIONS

We now present the results and discuss our observations. Figure 5 below shows stages of evolution of the synthesized image through time. Starting with random noise, the model (2 in this case) gradually shapes the image in such a way that the Hamming distance relative to the original is dropping (quality 2). It can also be said that the synthesized image after 200 epochs is visually distinct from the original and has some features of the parent image (quality 1). So, while more finessing can be done, we have reached our objectives. Final image would easily pass the ICD algorithm's test and will be classified as the true copy of the original.

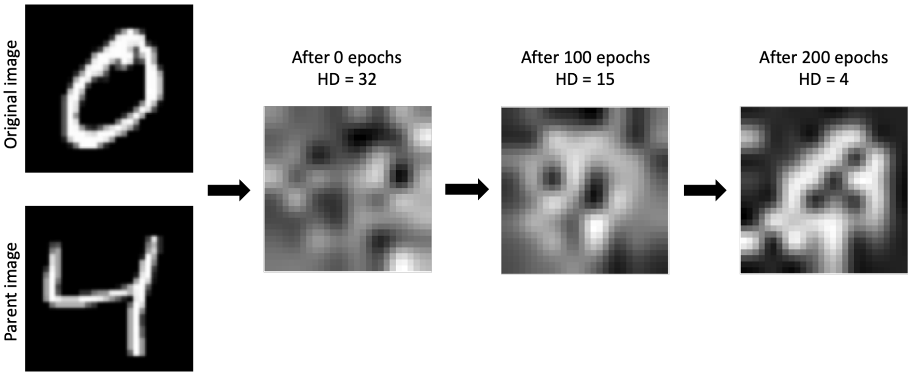

Figure 5

In Figure 6, we show the corresponding dynamics of the model's training to confirm that the neural networks have indeed converged.

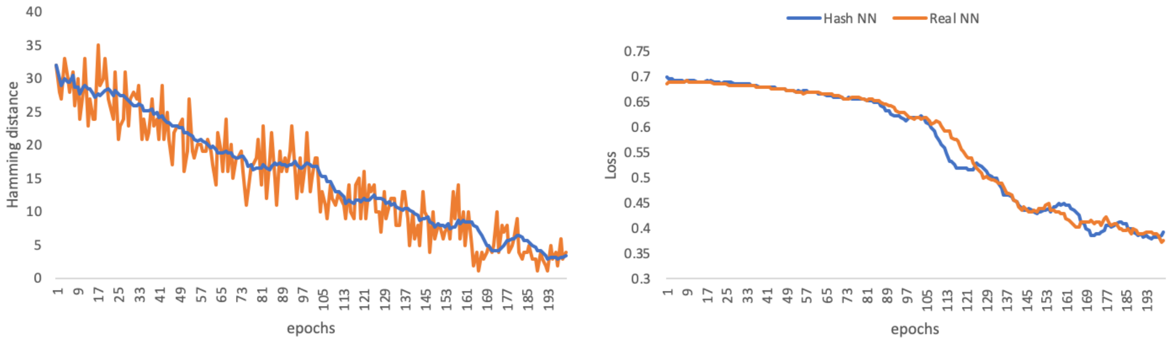

Figure 6

So what insights did we learn?

- Experiments suggest that it is indeed possible to ”fool” an image copy detection algorithm with a relatively straightforward set up and little compute required
- This is a simplified experiment and the investigated models are likely not to be sufficient in the real-world application. For example, when building the model, we only had to consider a simple feature extraction algorithm (block average). A more complex one maybe be harder to incorporate in the loss functions as it may become non-differentiable
- Relative dominance between two networks proved difficult to balance. We have investigated selecting different levels of depth for the two networks, but in the end we opted against intuition and in favor of a more systematic approach via weighting the outputs
- Deeper networks don't necessarily result in better performance. We found that 4-5 hidden layers was sufficient considering the size of the input pixel vector. This may need to be reconsidered for higher resolution images
- In some cases, we had to reverse-engineer image from its feature set. In order to do that we had to make some assumptions about the up-scaling mechanism. This had some impact on the training process as with the simple methods, we encountered serious loss of information which resulted in the networks getting stuck

To conclude, we would like to stress that knowing how the target image copy detection algorithm works is almost guaranteed to make it vulnerable to adversarial attacks. In our case, we had upfront knowledge of all the key parameters, such as feature extraction mechanism, size of the hash, similarity metric and the corresponding decision threshold. Knowing these, one can, at the very least, always find a brute force method of attack. It seems to us, therefore, that the simplest and strongest defence is by keeping this information hidden.

FUTURE RESEARCH

There are several directions of further research that we believe are worth pursuing next.

- Expand the analysis to images with RGB channels and higher resolution
- Consider a more sophisticated feature extraction technique like Discrete Cosine Transform, wavelets and ring partitions [4,8]
- Consider using CNN architecture and other approaches [3,5,9]

REFERENCE

Contributors: the author of this report is the sole contributor to this report
GitHub repository: https://github.com/ant-po/CS229project
Acknowledgment: the author took inspiration for some of the peripheral functions and basic GAN implementations from:
http://www.hackerfactor.com/blog/index.php?/archives/432-Looks-Like-It.html
https://github.com/JohannesBuchner/imagehash
https://github.com/diegoalejogm/gans

[1] M. Srivastava, J. Siddiqui, M. A. Ali, 2019. A Review of Hashing based Image Copy Detection Techniques
[2] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu and M. Song, 2019 "Neural Style Transfer: A Review"
[3] C-C. Hsu, C-Y. Lee, Y-X. Zhuang, 2018. Learning to Detect Fake Face Images in the Wild
[4] S. Jothimani, P. Betty, 2014. A Survey on Image Authentication Techniques
[5] K. G. Dizaji et al, 2017. Unsupervised Deep Generative Adversarial Hashing Network
[6] N. Locascio, 2018. Black-Box Attacks on Perceptual Image Hashes with GANs
[7] M. Mirza, S. Osindero, 2014. Conditional Generative Adversarial Nets
[8] Z. Tang, X. Zhang, S. Zhang, 2015. Robust Image Hashing with Ring Partition and Invariant Vector Distance
[9] P. Isola, J-Y Zhu, T. Zhou, A. Efros, 2018. Image-to-Image Translation with Conditional Adversarial Networks