Deepfake Detection Project Summary

Sam Buxbaum and Ahmed Hussain

November 2022

1 Introduction

Fake media poses an ever-increasing threat to society. Deepfake technology represents the state of the art in generating fake media, such as images and videos.

Current deepfake detection techniques rely almost exclusively on identifying subtle flaws in fake media, such as a tiny irregularity in the blinking pattern of a face. Whenever new detection models are published, deepfake generation techniques gain a new adversary to train against, and the task of detecting them gets a little bit more difficult.

The goal of our project is to get away from this cat-and-mouse game entirely. We take the assumption that deepfakes will someday be truly indistinguishable from reality and that a different approach is necessary.

1.1 Our Approach

We propose to detect fake media by creating and maintaining a list of "valid" media. With this approach, it is impossible to capture all valid media that exists, so we restrict our scope to news media, since this is comparatively simple to manage and is an area where fake media is particularly damaging. Ideally, we will expand the scope of the detection mechanism as the tools we develop for detection improve. News media serves as a motivating example that is both important and approachable.

Many of the challenges in the design of the system relate to the data structure used to store the list of media artifacts. The ideal data structure should store artifacts in a space-efficient and distributed way to minimize the size of the list and avoid having a single point of failure. A natural but potentially overkill solution is to use a blockchain that stores hashes of the artifacts across a network of computers along with a distributed consensus protocol to obtain agreement among the network. As we'll see, in the comparatively simple case of news media, the use of a blockchain is somewhat questionable, but it provides the advantage that it scales well as the problem and the network become more

complex. The problem of how to best hash the media is interesting and will be discussed later.

As a brief example of how this system would work, imagine that several prominent news publishers decided to work together to prevent against fake media claiming to originate from them. The publishers would maintain a collective list of media that they have published, where the media is stored as hashes, and each time one of them publishes a new artifact, such as an article or a video, they add the hash of the artifact to the shared list. If an individual wants to check the validity of a potentially fraudulent news artifact, they could do so through some interface by checking if the hash of the artifact is in the list of accepted hashes. Furthermore, this process could be automated to give some type of confirmation of validity whenever a secondary news source references the original and makes use of one of the artifacts, such as by showing a video originally published by one of the news publishers in the network. If the shared list of hashes is publicly available, then this system has the benefit of allowing individuals to verify validity manually, which provides transparency and eliminates the need for trust.

1.2 What We've Already Done

The bulk of the work so far has been related to developing the infrastructure necessary for a distributed network. To date, we have developed a prototype of a network where individual nodes can send dummy hashes amongst the network to construct the shared list. We have developed a full peer-to-peer communication backend, allowing any node in the network to communicate with any other node or to broadcast a message to the entire network.

So far, we have only been able to test the full system as a collection of independent processes running on the same computer, but we are currently working on setting up a small test network of multiple computers (which will hopefully include a few Raspberry Pis if we can get our hands on some).

2 Hashing

In order for the system to have any realistic success, hashing media artifacts is essential. However, this comes with many challenges. Here, we'll walk through a progression of ideas which will expose many of the difficulties and potential ways of handling them.

The most naive approach would be to cryptographically hash all media artifacts. This works great for artifacts such as text documents, which will yield a consistent result if hashed many times, but it fails for images and videos. It is possible and entirely realistic that an image will be compressed and distributed across the internet using lossy compression, so while any two copies of the image

look identical to a human, their cryptographic hashes would be different (and completely unrelated due to the nature of cryptographic hashes).

Instead, one might try using a *perceptual hash*, which is a type of (non-cryptographic) hash function which maps similar inputs to similar outputs. A perceptual hash attempts to reduce the size of the data represented while preserving similarities between close inputs. This solves the problem of lossy compression changing the pixel values of an image, but it creates a new problem. Perceptual hashes are not designed to have any cryptographic properties, so they have no second preimage resistance. That is, given a hash, it might be easy for an adversary to generate a new image which achieves the same hash with an unrelated input. The second image can even be adversarially generated with knowledge of the hash function to tweak individual pixels and trick the system into identifying two distinct images as identical, similar to how deepfakes are generated.

With the two naive solutions both failing, it might seem natural to combine them, applying a perceptual hash to capture the "essence" of an image and then cryptographically hash the result. However, the same flaw applies to the perceptual hashing stage as before: there is nothing stopping an adversary from creating two distinct images with identical perceptual hashes. This combination is simply a more complicated way of failing in the same way.

Lastly, we could enforce lossless compression so that the original objection to cryptographic hashing becomes invalid, but this is wildly unrealistic to accomplish, since it would require the entire internet to change to accommodate this individual system. In other words, the compression techniques used in typical internet interactions are beyond our control, so we should not hope to change that as a solution to our problem.

2.1 The Open Problem

Having seen that all simple existing approaches fail pretty spectacularly to meet the needs of our system, we are left with one option: create a new approach (or prove that such an approach is not possible).

We would like to combine elements of cryptographic hashing and perceptual hashing, such that images which are sufficiently similar hash to the same value, but the hash function has second preimage resistance. Our question is the following:

Is it possible to construct a *perceptographic hash function* with second preimage resistance such that images that are "close" by some similarity metric hash to identical values.