Novel Web Application for Image Authentication and Deep Fake Prevention

CS49991 Capstone Report, 2024

Cameron Greene
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
Crg2kvm@virginia.edu

ABSTRACT

Breakthroughs like Artificial intelligence (AI), neural networks (NN) and machine learning (ML) offer enormous potential; however, enormous risks come with their abilities. One of these dangers is deepfakes: an artificial image or video, a series of images, generated by deep learning models. ML models can help to identify altered images via photoshop and other editing methods, deepfakes, and general AI-generated content, but are by no means perfect. I have created a proof of concept for authenticating images and videos with certainty via a methodology similar to that of a certificate authority by utilizing built-in Python packages such as Flask for developing web applications, cryptography for RSA encryption, and SQLite3 for database functionalities, a secure web-application for authenticating images and accessing information on its production. The application was able to determine if images were altered with perfect accuracy and was easy for all new users to learn. The next steps include making the application more accessible and integrating it into everyday use through mobile devices.

1. INTRODUCTION

Recent breakthroughs in AI, NN and ML offer enormous potential as they reshape the future of humanity across nearly every industry. It is already the main driver of emerging technologies like big data, robotics and IoT. Deepfake technology, the creation of digitally altered videos and images, poses a pressing threat to modern society. Rapidly improving NN models have the ability to swap faces, alter facial expression, and make individuals appear to say or do something they have not. With increasingly blurred lines between reality and fiction, one of our longest standing and strongest pieces of evidence for proving an event's occurrence is being undermined**.**

Figuring out how to address this new challenge is no small feat. One method of debunking deepfakes is automating deepfake detection systems that analyze videos and attempt to classify their authenticity using NNs for computer vision. In recent years models have outperformed human experts in some medical diagnosis and strategy games, possibly offering another method for deepfake detection. In the context of deepfakes, these models surpass human abilities by detecting physical implausibility cues like distorted perspectives and inconsistencies in shadows or reflections.

The discrepancy in our detection abilities have profound implications. In many cases online social networks (OSNs) serve as vital sources of global and local news for millions of users: As we advance into an era where distinguishing between real and fabricated content becomes increasingly challenging, the development of advanced deepfake detection technologies is imperative. My project sheds light on these technologies, emphasizing their

significance in maintaining the authenticity and integrity of digital media in our rapidly evolving digital landscape.

2. RELATED WORKS

Researchers at the Salk Institute for Biological Studies in La Jolla, California, conducted two studies with 15,016 participants. The first study presented two videos to participants, one fake and one real; and the second study presented just one video, asking participants if it was real. Participants had an accuracy rate of 82% and 67%, respectively, while the leading detection model had an accuracy of 65% and 80%, respectively (Groh, 2022). Results demonstrate that while humans can detect deepfakes with a rather high accuracy, especially if they use models to assist them, there is still massive concern over the limitations of our current abilities. These models will undoubtedly serve as a valuable tool and filter for OSNs but, unfortunately, are not yet a solution.

Studies on the social media platform X, formerly Twitter, have indicated that fake news travels six times the rate of factual content (Vosoughi, et al., 2018). The impact of fake news is not confined to misinformation, but extends to identifying theft, shaping public opinion, influencing political discourse, and even eroding trust in information sources. As evidenced by the "infodemic" during the COVID-19 pandemic, false information circulated widely, leading to panic and confusion. As a result, surveys show that only 26% of Americans have confidence in their ability to recognize fake news, while 38.2% admit to having unintentionally shared fake news, testifying to a need and desire for a detection tool (Watson, 2023)**.**

3. PROJECT DESIGN

Our method for image authentication is modeled after that of a certificate authority: A certificate company is an organization that validates the identities of entities such as websites, email addresses, companies, or individual persons, binding them to cryptographic keys through the issuance of electronic documents known as digital certificates. In the same way, our application serves as a trusted authority that validates the origin of images and videos, binding images and their uploaders to secure accounts. The result is a database that third parties can query to find the image and ensure that it has not been tampered with or produced by AI-generated deepfakes. The following provides information about our methods of organization and development.

3.1 Repository

Our repository encompasses a three-tier architecture to enhance the abstraction of our application. This architecture organizes our code into three distinct layers responsible for their own respective concerns: The Presentation Layer, Business Logic Layer, and Data Access Layer.

3.1.1 Presentation Layer

The Flask App Class encapsulates all aspects of Flask applications, handling routes, views, and user interface interactions. The User Authentication Class deals with user functionalities including sign-up, login, and logout, and manages user sessions. The Image Upload and Processing Class offers image-related functionalities such as uploading, hashing and comparing, in addition to interacting with the database and handling image processing tasks. Last, the Frontend Templates Class is responsible for rendering HTML templates and managing frontend components.

3.1.2 Business Logic Layer

The User Management Class handles user-related business logic, including checking if a username exists, generating RSA keys, and managing user data. The Image Management Class deals with image-related business logic, such as converting images to strings, generating hashes, and saving image metadata

to the database. Finally, the Database Management Class is focused on database-related operations such as setting up the database schema, saving, and querying data.

3.1.3 Data Access Layer

The SQLite Database Class implements the database connection and executes SQL queries.

3.2 Key Functions

This web app uses various functions to accomplish its uses. They are split up into utility functions which perform the image analysis and routing functions which connect the information to the views to display the information.

3.2.1 Utility Functions

The `imageToString(image)` function converts an image file into a string representation using Pillow, while the `stringToHash(imgString)` function generates a SHA-256 hash from this string to create a unique identifier for each image. The `setup_database()` function initializes an SQLite database and creates necessary tables for storing user information and image hashes. Functions like `save_hash _to_db _with_user(username,image_hash)` and `get_ hashes_by_user (username)` manage image hashes in the database, linking them to specific users and retrieving them per user, respectively. The `hash_exists_in_db(image _hash)` function checks for existing image hashes to prevent duplicate uploads, and the `generate_rsa_keys()` generates RSA keys, providing a public-private key pair for user authentication.

3.2.2 Routing Functions

The `imageToString(image)` function converts an image file into a string representation using Pillow, while the `string ToHash(imgString)` function generates a SHA-256 hash from this string to create a unique identifier for each image. The `setup_database()` function initializes an SQLite database and creates necessary tables for storing user information and image hashes. Functions like `save_hash _to_db _with_user(username,image_hash)` and `get_ hashes_by_user(username)` manage image hashes in the database, linking them to specific users and retrieving them per user, respectively. The `hash_exists_in_db(image _hash)` function checks for existing image hashes to prevent duplicate uploads, and the `generate_rsa _keys()` generates RSA keys, providing a public-private key pair for user authentication.

4. RESULTS

The results of this study pivots away from traditional quantitative analysis and instead explores the practical application development aimed at authenticating digital media. Utilizing established Python libraries, the web application that embodies the principles of a certificate authority, serves as a validator for the authenticity of images and videos. This tool not only cross-references uploaded content against a database to check for tampering or AI-generated alterations but also includes a number of user interactions from image upload to secure account management. The successful deployment of this application reflects a significant step forward in mitigating the risks posed by deepfake technologies, providing a reliable means for users to verify content authenticity.

Evaluating the application's effectiveness, the results were twofold. First, the application demonstrated a perfect accuracy rate in detecting alterations in images, distinguishing itself as a highly reliable tool in the realm of digital media verification. Second, user feedback highlighted the application's ease of use, indicating that the learning curve for new users was minimal. This ease of adoption is critical in encouraging widespread use, which is essential for the tool's integration into everyday digital interactions. The potential of this application to contribute meaningfully to the detection and prevention of deepfake detection. holds promise, particularly as we

look towards enhancing accessibility and further integration into mobile technology platforms.

5. CONCLUSION

This application presents a critical advancement in the field of digital media verification through the development of a web application that functions as a certificate authority, validating the authenticity and origin of images in a world increasingly infected by deepfake technologies. The application, underpinned by various Python libraries, has demonstrated impeccable accuracy in detecting tampered images, thereby setting a new standard for digital media authentication tools. Its design not only supports essential user interactions, including secure image uploads and account management, but also promises ease of use to encourage widespread adoption.

The meaningful implications of this project extend beyond its current capabilities, highlighting the potential for future integration into mobile platforms and the broader digital ecosystem, which would further bolster the reliability and integrity of digital content. Looking to the future, one key learning from this project is the importance of adhering to a stricter, more organized structural framework from the outset. This approach will not only enhance the readability and maintainability of the code but will also streamline the development process. Adopting a more rigid structure, such as a three-tiered organization with a database level, a logic level and a frontend/UX level, would facilitate easier expansion, debugging, and collaboration for any future enhancements or iterations of the application.

6. FUTURE WORK

The web application developed in this research represents a promising step in combatting the manipulation of digital media, specifically addressing the issue of deepfakes. As a countermeasure, the application adopts the role of a certificate authority, verifying the origins and authenticity of images, with potential applications across various sectors. In its current form as a proof of concept, the tool harnesses cryptographic methods to secure digital imagery, indicating its efficiency in protecting against the erosion of visual images.

Future enhancements of the tool are anticipated to transform it into a more comprehensive, integrated app with a user-friendly interface and expanded capabilities, such as real-time image uploads and instant verification. While initial concerns regarding data storage and associated costs were considered, the prevalent use of metadata storage in modern devices provides a practical blueprint for embedding hash values efficiently. This approach underscores the tool's potential for broad adoption without significant financial burden.

REFERENCES

Groh, M. (2022). “Deepfake detection by human crowds, machines, and machine-informed crowds.” The Proceedings of the National Academy of Sciences, January 5, 2022; 119(1): e2110013119

Vosoughi, S., Roy, D., Aral, S. (2018). “The spread of true and false news online.” Science. 2018;359(6380):1146–1151. doi: 10.1126/science.aap9559.

Watson, A. (2023). “Fake news in the U.S. - statistics and facts” Statista, July 14, 2023, www.statista.com/topics/3251/fake-news