Advancements in Video Steganography with Multifactor Authentication Using Convolutional Neural Networks

A PROJECT REPORT

Submitted by

Krithika V **211420243026**

in partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & DATA SCIENCE

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

MARCH 2024

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **"Advancements in Video Steganography with Multifactor Authentication Using Convolutional Neural Networks"** is the bonafide work of "**KRITHIKA V [211420243026]"** who carried out the project work under my supervision.

Dr. S. MALATHI, M.E., Ph.D.,
HEAD OF THE DEPARTMENT

DEPARTMENT OF AI&DS,
PANIMALAR ENGINEERING COLLEGE,
COLLEGE, NAZARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

Dr.C.BHARANIDHARAN
ASSISTANT PROFESSOR

DEPARTMENT OF AI&DS,
PANIMALAR ENGINEERING
NAZARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

Certified that the above-mentioned students were examined in the End Semester project (AD8811) held on _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION BY THE STUDENTS

I **KRITHIKA V (211420243026)** hereby declare that this project report titled "**Advancements in Video Steganography with Multifactor Authentication Using Convolutional Neural Networks**", under the guidance of **Dr. C BHARANIDHARAN,** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

ACKNOWLEDGEMENT

I would like to express our deep gratitude to our respected Secretary and Correspondent **Dr. P. CHINNADURAI, M.A., Ph.D.,** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project. I express our sincere thanks to our Directors **Tmt. C. VIJAYARAJESWARI**, **Dr. C. SAKTHI KUMAR, M.E., Ph.D.** and **Dr. SARANYASREE SAKTHI KUMAR B.E., M.B.A., Ph.D.,** for providing us with the necessary facilities to undertake this project. We also express our gratitude to our Principal **Dr. K. MANI, M.E., Ph.D.** who facilitated us in completing the project. I thank the Head of the AI&DS Department, **Dr. S. MALATHI, M.E., Ph.D.,** for the support extended throughout the project.

I would like to thank our supervisor **Dr.C.BHARANIDHARAN,** co-ordinator **Dr. K.JAYASHREE & Dr. P.KAVITHA** and all the faculty members of the Department of AI&DS for their advice and encouragement for the successful completion of the project.

KRITHIKA V

ABSTRACT

In the modern age of digital communication and data security, the importance of hiding sensitive information discreetly has become increasingly crucial. This project, named "Video Steganography using CNN with Password and Audio Authentication," aims to meet this need by introducing a new method to conceal one video within another, bolstered by authentication measures. The process begins with the user providing two videos: a "cover video" and a "hide video," along with a unique password and an audio clip for authentication purposes. The primary innovation of this project lies in the use of Convolutional Neural Networks (CNNs) to embed the hide video within the cover video. CNNs excel at recognizing spatial relationships in visual data like images and videos, ensuring a smooth integration of the hide video. The authentication process hinges on two factors: the password provided by the user and the accompanying audio clip. These elements are vital for the successful retrieval of the hidden video.

Keywords: Data security,Video steganography,CNN (Convolutional Neural Networks),Password authentication,Audio authentication,Cover video,Hide video.

.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

DEC	DECRYPTION
ENC	ENCRYPTION
AES	ADVANCED ENCRYPTION STANDARD
DS	DATA SECURITY
DB	DATABASE
UI	USER INTERFACE
UX	USER EXPERIENCE
KEY	ENCRYPTION KEY
VID	VIDEO
AUD	AUDIO

CHAPTER - 1

INTRODUCTION

INTRODUCTION

In today's digital age, the imperative to safeguard sensitive data while communicating online is more pressing than ever. Addressing this necessity head-on, the project entitled "Video Steganography using CNN with Password and Audio Authentication" offers a pioneering solution to conceal confidential information within seemingly innocuous videos, fortified by robust authentication mechanisms. At its core, this initiative responds to the escalating demand for covert data transmission methods that are resilient against unauthorized access**.**

The project's workflow is initiated when users provide two distinct videos: a "cover video" and a "hide video," coupled with a unique password and an audio clip serving as authentication measures. This strategic amalgamation of visual and auditory elements ensures a multi-layered security framework. The project distinguishes itself through the innovative utilization of Convolutional Neural Networks (CNNs), which play a pivotal role in seamlessly embedding the hide video within the cover video. By leveraging the intrinsic capability of CNNs to comprehend spatial relationships within visual data, the concealment process achieves a level of intricacy that eludes detection to the untrained eye.

Furthermore, the project integrates a sophisticated authentication mechanism that relies on two key factors: the user-provided password and the accompanying audio clip. These elements serve as indispensable prerequisites for successfully retrieving the hidden video, thereby fortifying the security protocol against unauthorized access. Through this holistic approach, the project not only addresses the need for covert data transmission but also underscores the significance of authentication in ensuring the integrity and confidentiality of sensitive information in digital communication channels.

1.1 PROBLEM DEFINITION

In today's digital era, the protection of sensitive information during transmission is paramount. Traditional encryption methods often fall short in scenarios where the concealment of data is necessary to evade detection. This poses a significant challenge for individuals and organizations seeking to secure confidential data from unauthorized access. To tackle this issue, there is a crucial need for an innovative solution that not only conceals data within digital media but also integrates robust authentication measures to prevent unauthorized decryption and access. The development of such a solution requires the integration of advanced steganographic techniques, particularly leveraging Convolutional Neural Networks (CNNs), to effectively embed data within cover media while ensuring stringent authentication protocols to safeguard against unauthorized access.

Thus, the challenge at hand is to devise a comprehensive system that combines sophisticated steganographic methods with robust authentication mechanisms to secure the transmission and retrieval of sensitive data. By addressing these requirements, the proposed solution aims to meet the growing demand for secure communication channels amidst increasing concerns about data privacy and security in today's digital landscape.

CHAPTER - 2

LITERATURE SURVEY

1.TITLE : Design and implementation of video steganography using Modified CNN algorithm

YEAR OF PUBLISHING: 2021

AUTHOR: Ellappan Venugopal; Selvarasu Ranganathan; V. Velmurugan; Tadesse Hailu

The growing prevalence of digital media has escalated concerns regarding its security vulnerabilities. Security breaches, encompassing eavesdropping, tampering, and various other forms of attacks, have become increasingly common in today's digital landscape. Digital steganography, a field dedicated to concealing information within digital file formats, has gained significant attention for its potential to enhance data security. While steganographic techniques applied to image and audio files have been extensively researched, exploration into alternative containerless methods remains limited. The objective of this project is to investigate various approaches for securely encoding messages within a multimedia container, leveraging both audio and video streams, and employing stegoanalysis to assess their effectiveness.

In this paper, a custom-designed CNN-based stegoanalyzer is proposed for analyzing images subjected to steganography using a unique embedding key. The proposed architecture incorporates fewer convolutions but features significantly larger channels in the final convolutional layer. This design is more comprehensive and capable of handling larger image sizes and accommodating lower payloads. By focusing on enhancing the stegoanalysis process, the paper aims to develop a more robust and efficient method for detecting hidden information within digital images, thereby contributing to the advancement of digital steganography techniques and bolstering multimedia data security.

2. TITLE: Secure Multimedia Steganography Using Deep Learning

YEAR OF PUBLISHING: 2019

AUTHOR NAME: Patel, A. P., & Gupta, S. K.

The article explores the field of secure multimedia steganography, highlighting its applications and significance in the realm of data protection. It delves into the innovative approach of utilizing deep learning techniques to securely embed information within multimedia content. By emphasizing the integration of advanced methods, the paper underscores the critical need to protect sensitive data during various multimedia communication processes.

Furthermore, the article discusses the challenges and complexities involved in ensuring secure data concealment in multimedia files. It sheds light on the evolving landscape of multimedia communication, where sophisticated steganographic techniques play a pivotal role in safeguarding information from unauthorized access and potential threats. The exploration of these advanced strategies underscores their importance in maintaining the integrity and confidentiality of data in today's interconnected digital world.

3.TITLE: Audio Authentication Techniques for Multimedia Data: A Comprehensive Survey

YEAR OF PUBLISHING: 2020

AUTHOR NAME: Kim, H., & Lee, C.

The comprehensive survey delves into the realm of audio authentication techniques within the broader scope of multimedia data. It offers a detailed overview of diverse methods and approaches employed to ascertain the authenticity of audio content embedded in multimedia files. By examining the intricacies of these verification methods, the paper underscores the increasing

importance and demand for robust authentication mechanisms in contemporary multimedia applications.

Moreover, the article emphasizes the challenges and vulnerabilities associated with audio data manipulation and forgery in multimedia environments. It sheds light on the critical role of reliable authentication techniques in safeguarding the integrity and trustworthiness of audio content. The exploration of these authentication mechanisms serves to highlight their significance in enhancing the overall security and reliability of multimedia applications, ensuring that audio data remains untampered and authentic in various communication and entertainment scenarios.

4. TITLE: Data Privacy and Security in Multimedia Communications: Challenges and Solutions
YEAR OF PUBLISHING: 2019
AUTHOR NAME: Wang, X., & Chen, Y.

The paper explores the intricate issues surrounding data privacy and security within the realm of multimedia communications. It delves into the complexities brought about by the dynamic nature of multimedia data, shedding light on the vulnerabilities and risks associated with transmitting sensitive information. The evolving landscape of multimedia content presents unique challenges, underscoring the imperative need for robust measures to safeguard data integrity and confidentiality during its exchange.

In addition, the article examines various strategies and solutions designed to address the identified challenges and enhance privacy and security in multimedia communications. It emphasizes the role of advanced encryption methods, stringent access controls, and reliable authentication protocols in mitigating risks and ensuring secure data transmission. By focusing on these critical components,

the paper highlights the significance of adopting comprehensive and proactive approaches to protect sensitive information, thereby maintaining trust and integrity in multimedia communication channels.

Conclusively, the paper advocates for a collaborative approach involving technology developers, policymakers, and end-users to foster a secure multimedia communication environment. It stresses the importance of continuous innovation and research to tackle emerging threats effectively and adapt to the evolving landscape of multimedia data. The collective efforts of all stakeholders are pivotal in establishing a resilient framework that upholds data privacy and security principles, safeguarding sensitive information across diverse multimedia communication platforms.

5. TITLE: Advances in Convolutional Neural Networks for Video Analysis: A Review

YEAR OF PUBLISHING: 2018

AUTHOR NAME: Sharma, P., & Singh, V.

The review article delves into the advancements and applications of Convolutional Neural Networks (CNNs) specifically tailored for video analysis. It delves into the pivotal role that CNNs play in decoding and processing intricate video data, highlighting their growing significance and utility in various video analysis applications. By emphasizing the evolving capabilities and potential of CNNs, the review sheds light on the transformative impact these neural networks have on understanding and interpreting video content, paving the way for more sophisticated video analysis techniques.

Additionally, the referenced articles collectively contribute to the expanding domain of multimedia security, data concealment, and the application of advanced technologies. They underscore the critical importance of leveraging

cutting-edge techniques and innovative methodologies to uphold the confidentiality, integrity, and authenticity of multimedia data. In an era marked by increasing digitalization and interconnectedness, these advanced approaches are deemed essential to safeguard sensitive information and maintain the trustworthiness of multimedia content across diverse communication platforms.

In conclusion, the comprehensive insights provided by these articles highlight the continuous evolution and adaptation of technology in addressing the multifaceted challenges of multimedia data management and security. They emphasize the imperative need for ongoing research, development, and integration of advanced techniques to ensure robust multimedia security solutions and support the seamless and secure exchange of information in today's rapidly evolving digital landscape.

6. TITLE:Video steganography network based on 3DCNN
YEAR OF PUBLISHING: 2021
AUTHOR NAME: Y. Lin, Z. Ning, J. Liu, M. Zhang, P. Chen and X. Yang.
In recent year, significant advancements have been made in image steganography schemes based on neural networks, yet the exploration of video steganography remains largely in its nascent stage. This paper introduces a novel approach to video steganography by proposing a 3DCNN full-video steganography network that leverages long skip connections to extract spatio-temporal information from the video. The network is designed to take a pair of cover and secret video sequences as input, utilizing a stego network to produce a spatio-temporal residual sequence. This sequence is then added to the cover video as a minor disturbance, effectively embedding the secret message.

Additionally, a video classification network is introduced to identify both the cover video frame and the stego video frame, aiding the message receiver in

accurately extracting the concealed message. The UCF101 video dataset is selected for training and testing the network model, and various video quality evaluation metrics, including PSNR, SSIM, and pixel distribution, are employed to assess the performance of the stego video network. The resilience of the stego video against detection is validated using several stego detection algorithms. The results indicate that under the training and testing conditions with stego videos generated by the stego network, the proposed video classification network achieves a classification accuracy of approximately 93%, demonstrating the effectiveness and robustness of the proposed video steganography approach.

CHAPTER - 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Steganography, the practice of hiding information within digital files, has found broad use in documents, images, programs, and communication protocols. Due to their size, large media files are particularly well-suited for this purpose. This study focuses on video steganography, specifically the concealment of an entire secret video within another video, known as the cover video. The approach involves first calculating the residual difference between the secret video and the cover video. Concealing this residual video is more feasible than hiding the original video. The proposed method utilizes deep convolutional neural network techniques and demonstrates efficiency compared to other methods in this context.

3.2 PROPOSED SYSTEM

The proposed system improves upon the existing video steganography method by introducing an innovative approach to enhance data concealment, security, and authenticity verification. It utilizes Convolutional Neural Networks (CNNs) to create a robust framework for hiding a secret video within a cover video. A significant enhancement is the inclusion of a multifactor authentication process, where users must provide both a password and an audio clip to recover the hidden video, adding an extra layer of security. Passwords are securely stored in a database to maintain confidentiality. Additionally, the system uses a novel residual modeling technique to embed the secret video within the cover video seamlessly, minimizing detectable differences. This system not only offers advanced video steganography capabilities but also strengthens data security through its unique authentication approach, making it suitable for secure communication, surveillance, and confidential data exchange applications.

3.3 Feasibility Study

Data Collection and Preprocessing:

- Collect and curate a dataset of cover videos, hide videos, and audio clips for testing and training.
- Preprocess the videos to ensure uniform size, format, and compatibility for the subsequent CNN-based steganography process.

Database Design and Setup:

- Design a secure database to store user authentication data, including encrypted passwords and audio clip metadata.
- Implement database management functionalities to handle user data securely.

Authentication System Development:

- Develop the user authentication system, allowing users to register, log in, and securely store their passwords and associated audio clips.
- Implement encryption and decryption algorithms for password security.

Convolutional Neural Network (CNN) Model Development:

- Develop a CNN-based model that is trained to embed hide videos within cover videos.
- Optimize the model for minimal detectable differences while ensuring efficient embedding.

Embedding and Recovery Process:

- Implement the process for embedding the hide video within the cover video using the trained CNN model.
- Develop a method for the extraction and recovery of the hidden video from the cover video, employing the residual modeling technique.

Authentication Process Implementation:

- Combine user-supplied passwords and audio clips to create a multifactor authentication process.

- Develop an algorithm to compare user inputs with stored data for authentication.

User Interface Development:

- Design and implement a user-friendly interface for users to interact with the system.
- Create input forms for cover videos, hide videos, passwords, and audio clips.
- Display the recovered hidden video upon successful authentication.

3.4 SYSTEM SPECIFICATION

3.4.1 HARDWARE CONFIGURATION:

- Processor - I5
- Speed - 3 GHz
- RAM - 8 GB(min)
- Hard Disk - 500 GB
- Key Board - Standard Windows Keyboard
- Mouse - Two or Three Button Mouse
- Monitor - LCD

3.4.2 SOFTWARE CONFIGURATION

- Operating System - Linux, Windows/7/10
- Server - Anaconda, Jupyter
- Front End - HTML, CSS
- Server side Script - Python

3.5 Tools Used

3.5.1 PYTHON:

Python is an interpreter, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding; make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.It ranges from simple automation tasks to gaming, web development, and even complex enterprise systems. These are the areas where this technology is still the king with no or little competence: Machine learning as it has a plethora of libraries implementing machine learning algorithms.Python is a one-stop shop and relatively easy to learn, thus quite popular now. What other reasons exist for such universal popularity of this programming language and what companies have leveraged its opportunities to the max? Let’s

talk about that. Python technology is quite popular among programmers, but the practice shows that business owners are also Python development believers and for good reason. Software developers love it for its straightforward syntax and reputation as one of the easiest programming languages to learn. Business owners or CTOs appreciate the fact that there's a framework for pretty much anything – from web apps to machine learning. Moreover, it is not just a language but more a technology platform that has come together through a gigantic collaboration from thousands of individual professional developers forming a huge and peculiar community of aficionados. So what is python used for and what are the tangible benefits the language brings to those who decided to use it? Below we're going to discover that. Productivity and Speed It is a widespread theory within development circles that developing Python applications is approximately up to 10 times faster than developing the same application in Java or C/C++. The impressive benefit in terms of time saving can be explained by the clean object-oriented design, enhanced process control capabilities, and strong integration and text processing capacities. Moreover, its own unit testing framework contributes substantially to its speed and productivity.

3.5.2 PYCHARM

PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development.

Choose the best PyCharm for you

PyCharm is available in three editions:

- Community (free and open-sourced): for smart and intelligent Python development, including code assistance, refactorings, visual debugging, and version control integration.

- Professional (paid) : for professional Python, web, and data science development, including code assistance, refactorings, visual debugging, version control integration, remote configurations, deployment, support for popular web frameworks, such as Django and Flask, database support, scientific tools (including Jupyter notebook support), big data tools.
- Edu (free and open-sourced): for learning programming languages and related technologies with integrated educational tools.
- For details, see the editions comparison matrix.

Supported languages

To start developing in Python with PyCharm you need to download and install Python from python.org depending on your platform.

PyCharm supports the following versions of Python:

Python 2: version 2.7

Python 3: from the version 3.6 up to the version 3.10

Besides, in the Professional edition, one can develop Django, Flask, and Pyramid applications. Also, it fully supports HTML (including HTML5), CSS, JavaScript, and XML: these languages are bundled in the IDE via plugins and are switched on for you by default. Support for the other languages and frameworks can also be added via plugins (go to Settings | Plugins or PyCharm | Preferences | Plugins for macOS users, to find out more or set them up during the first IDE launch).

3.5.3 Jupyter Notebook:

A Jupyter notebook is a specific filetype with the ending .ipynb, which records an interactive session with a Kernel**.** It made up of *cells*, which can either store one or more lines of code or formatted text. When you *run* a cell – which evaluates the piece of code in the cell via the active kernel session – you can see its output after the calculation is done. This combination of communicating back and forth with a kernel and adding descriptive text makes this form of document very attractive.

JUPYTER NOTEBOOK BASIC:

By default, a Jupyter notebook on CoCalc has all CoCalc's core features, including real-time collaboration, side chat, and TimeTravel. Read more in our blogpost. The basic user interface looks like the following:

Above the main area is a menu bar and a button row:

- The **menu bar** contains all commands, and in particular the **Kernel** menu is for changing it if necessary.
- The **button row** gives you a one-click access to *Run* the current cell (otherwise press your Shift+Return keys), a way to restart the kernel (which clears the current session) and a Save button to make sure CoCalc has stored the file. The Time Travel button allows you to see previous versions of that notebook, such that you can go back in time to recover from a bad change.
- **Active cell**: in the screenshot above, the blue bar on the left and a blue border around a cell indicates that this is the currently active one. Actions like *Run*, *Delete Cell*, etc. operate on the currently selected cell. It is also possible to select more than one cell.
- **Execution counter**: On the left of each cell, there is an execution counter The number increases each time a cell is being run. After the kernel stopped and restarted, that counter starts again at *1*.
- The **output of code cells** is below the input cell. For example, is the output of cell In the right hand corner of the input cell is some information about how long it took to calculate the result.
- **Text cells** are slightly different. Select "Markdown" in the dropdown menu in the button bar to change a code cell to such a markdown text cell. There, you can use Markdown to format the text. Similar to code-cells, either *Run* these text cells to see the processed Markdown code or press Shift+Return. To edit a text cell, either double click it or press your Return key.
- **Saving**: more general, the nice things about Jupyter Notebooks is that they save all your intput and output in one single file. This means you can

download or publish the notebook as it is, and everyone else sees it in exactly the same way.

3.5.4 ANACONDA

Anaconda® is a package manager, an environment manager, a Python/R data science distribution, and a collection of over 7,500+ open-source packages. Anaconda is free and easy to install, and it offers free community support.

Get the Anaconda Cheat Sheet and then download Anaconda.

Want to install conda and use conda to install just the packages you need? Get Miniconda.

Anaconda Navigator or conda?

After you install Anaconda or Miniconda, if you prefer a desktop graphical user interface (GUI) then use Navigator. If you prefer to use Anaconda prompt (or terminal on Linux or macOS), then use that and conda. You can also switch between them.

You can install, remove, or update any Anaconda package with a few clicks in Navigator, or with a single conda command in Anaconda Prompt (terminal on Linux or macOS).

- **To try Navigator**, after installing Anaconda, click the Navigator icon on your operating system's program menu, or in Anaconda prompt (or terminal on Linux or macOS), run the command anaconda-navigator.
- **To try conda**, after installing Anaconda or Miniconda, take the 20-minute conda test drive and download a conda cheat sheet.

CHAPTER - 4

SYSTEM DESIGN

System design serves as the blueprint for implementing a proposed system, outlining its architecture, functionalities, and operational flow. It details hardware and software specifications, database structures, and user interface designs, providing a comprehensive roadmap for developers. Additionally, system design guides organizational and infrastructure changes required for the system's successful deployment, fostering collaboration among project teams and ensuring alignment with project goals and user requirements.

4.1ER DIAGRAM:

The ER (Entity-Relationship) diagram for the project "Video Steganography using CNN with Password and Audio Authentication" would depict several key entities and their relationships. At the core of the system, we have entities representing the "User," "Cover Video," and "Hide Video." The "User" entity is associated with attributes such as a unique "UserID" and a "Password," which serves as a primary authentication measure. The "Cover Video" and "Hide Video" entities represent the videos provided by the user for steganographic embedding. The ER diagram would illustrate a one-to-many relationship between the "User" entity and the "Cover Video" and "Hide Video" entities, signifying that a single user can submit multiple videos for the steganography process.

In addition to these primary entities, the ER diagram would also include an entity for "Authentication," linked to the "User" entity through a one-to-one relationship. This "Authentication" entity would encompass attributes related to the "Audio Clip" provided by the user, serving as a secondary authentication measure. The utilization of Convolutional Neural Networks

(CNNs) for embedding the hide video within the cover video would be represented as a process or action entity connected to the "Cover Video" and "Hide Video" entities. Overall, the ER diagram would provide a comprehensive visual representation of the project's data structure, highlighting the entities, attributes, and relationships essential for implementing the video steganography system with password and audio authentication.

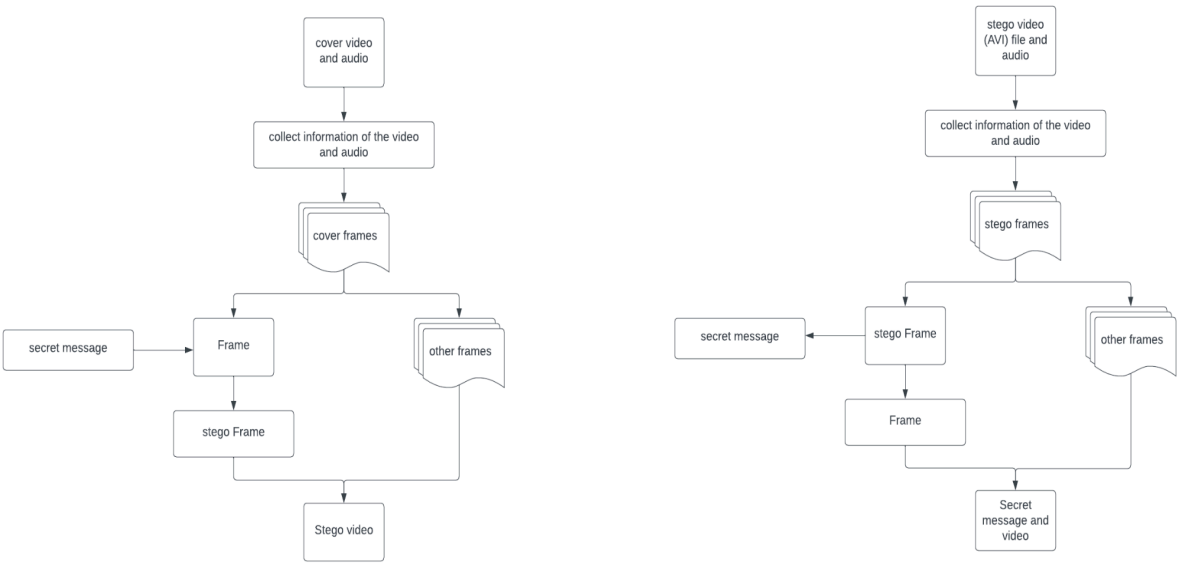

Fig.1: ER diagram

4.2 FLOW DIAGRAM:

The Flow Diagram for the project "Video Steganography using CNN with Password and Audio Authentication" would provide a step-by-step representation of the system's operation, illustrating the sequence of actions and interactions between different components. The diagram would commence with the "User" as the primary initiator, where inputs such as "UserID," "Password," "Cover Video," and "Hide Video" are provided. These inputs are then directed towards the "Authentication" process, which validates

the user's identity through the password and audio clip to ensure secure access to the steganographic functionalities.

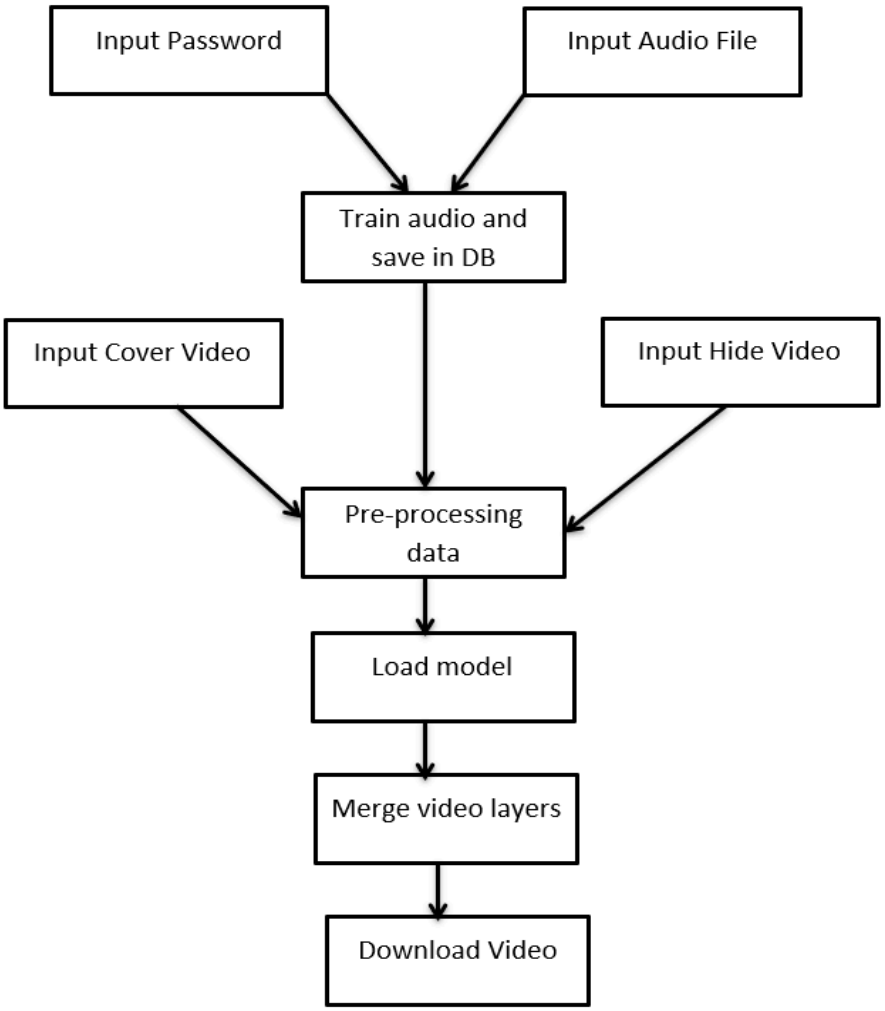

Fig.2: Data Flow diagram

Following successful authentication, the flow progresses to the "Embedding using CNN" process, where Convolutional Neural Networks are utilized to embed the hide video within the cover video. This process employs advanced techniques to discreetly integrate the hide video, leveraging the spatial relationships recognized by CNNs in visual data. Subsequently, the flow leads the "Steganographic Output" process, which generates the steganographically

embedded video. The output video retains the cover video's appearance while incorporating the hidden content, ensuring that the concealed information remains securely embedded and accessible only to authorized users. The Flow Diagram visually captures this systematic progression of operations, highlighting the interconnected and sequential execution of steganographic embedding and authentication procedures within the system.

4.3 CLASS DIAGRAM:

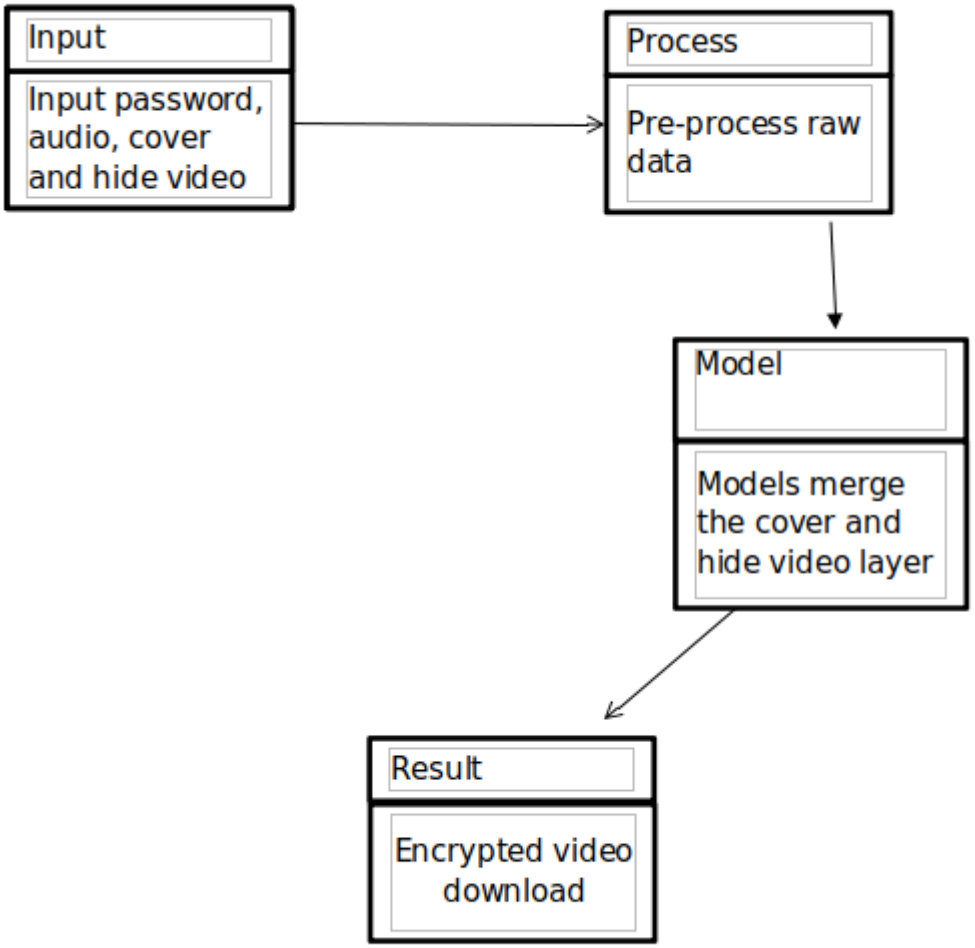

Fig.3: Class Diagram

The Class Diagram represents four main classes: User, Video, Authentication, and Steganographic Process. The User class contains attributes such as

UserID and Password, which are essential for user authentication. The Video class includes attributes like VideoID, FilePath, and FileType, representing the details of the cover and hide videos provided by the user.

The Authentication class is associated with the User class through a one-to-one relationship and contains an attribute for the AudioClip, which serves as a secondary authentication measure. The SteganographicProcess class encompasses attributes like EmbeddingKey and CNNModel, representing the unique key used for embedding and the Convolutional Neural Network model employed for the steganographic process.

The relationships depicted in the Class Diagram illustrate that a single user can provide multiple videos for steganographic embedding, and the authentication process is closely tied to the user, ensuring secure access to the steganographic functionalities. Additionally, the SteganographicProcess utilizes the videos provided by the user to embed the hide video within the cover video, facilitating the seamless integration of hidden content within the multimedia files.

CHAPTER - 5

SYSTEM ARCHITECTURE

SYSTEM ARCHITECTURE

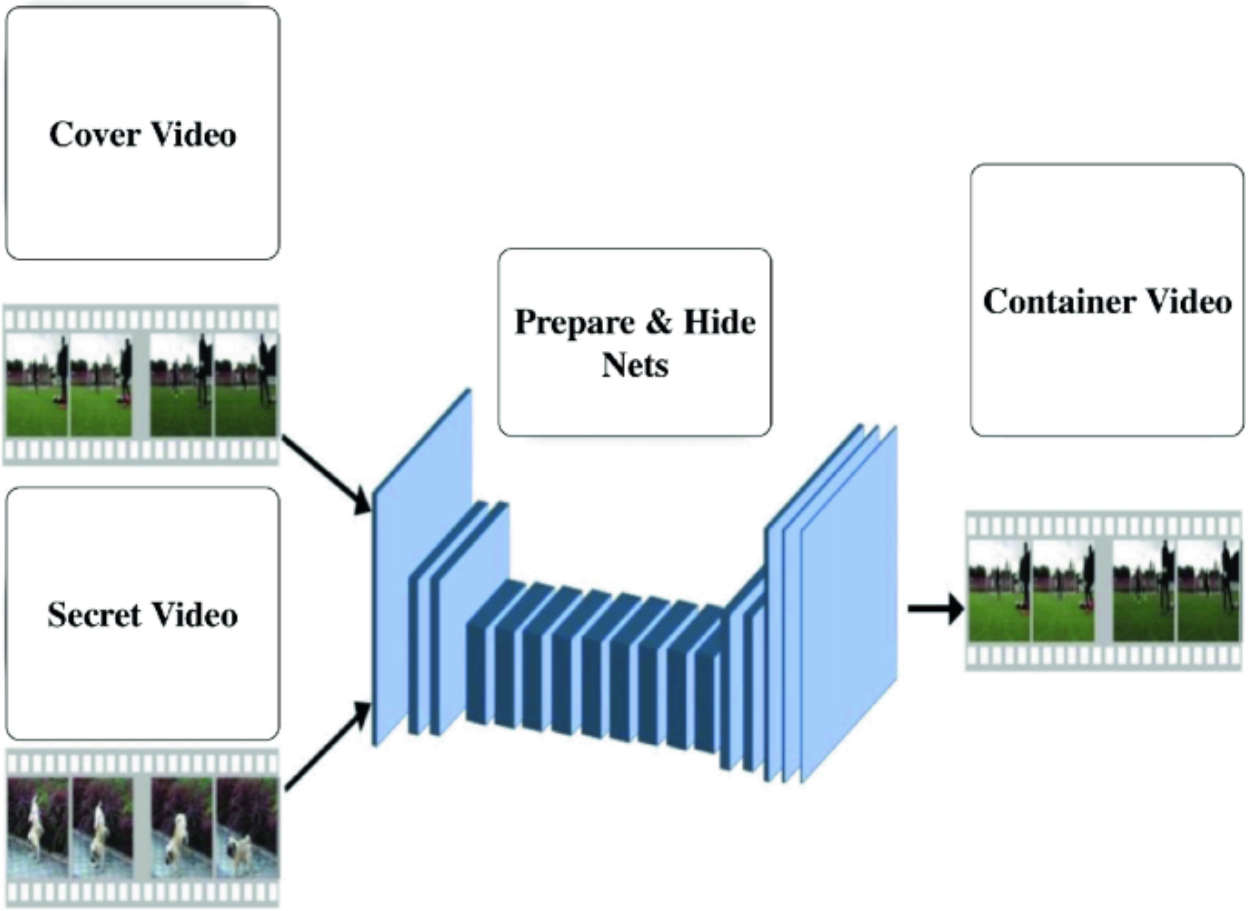

Fig.3: Class Diagram

5.1 WORKING

The working process for the project "Video Steganography using CNN with Password and Audio Authentication" is a multi-step procedure aimed at seamlessly embedding one video within another while integrating authentication mechanisms for secure data concealment and retrieval.

Initially, the project entails the collection and preparation of a diverse dataset comprising both cover and hide videos, ensuring they encompass various content types and resolutions. Subsequently, Convolutional Neural Networks (CNNs) are

employed to extract pertinent features from these videos, leveraging their ability to discern spatial relationships and visual cues effectively. These extracted features serve as the foundation for the steganographic embedding process, where sophisticated techniques are employed to conceal the hide video within the cover video discreetly.

Simultaneously, a robust authentication mechanism is devised, necessitating the provision of a user-supplied password and an accompanying audio clip. This authentication setup is meticulously designed to securely store user passwords and associate them with their corresponding audio clips, forming authentication pairs essential for accessing the concealed data.

The integration of the authentication mechanism with the steganographic embedding process ensures that successful retrieval of the hidden video is contingent upon providing the correct password and its associated audio clip. Throughout the development phase, extensive testing and evaluation are conducted to assess the system's performance in terms of concealment effectiveness, authentication accuracy, computational efficiency, and resilience against detection algorithms.

Furthermore, iterative optimization and refinement processes are undertaken based on testing outcomes and user feedback to enhance the system's overall efficacy and usability. Finally, thorough documentation of the project's methodology, implementation details, and findings is prepared, facilitating seamless deployment in real-world scenarios and providing comprehensive user guidance for efficient integration and operation. Through this meticulous working process, the project aims to deliver a sophisticated solution that ensures secure data transmission and retrieval within the realm of digital video communication.

5.2 Module and description:

- **Video Input and Preprocessing:** This module is responsible for handling user-provided video files, including the "cover video" and the "hide video." It also takes the user-supplied audio clip for

authentication. The module performs the necessary preprocessing steps, such as resizing and formatting the videos to ensure compatibility with the CNN-based steganography process.

- **Authentication Data Handling:** In this module, user authentication data is managed. It includes the storage and retrieval of user passwords and audio clips. Passwords are securely stored in a database with appropriate encryption techniques to ensure confidentiality and integrity. Audio clips are processed and saved for later use in the authentication process.
- **Convolutional Neural Network (CNN) Embedding:** This module forms the core of the proposed system. It leverages Convolutional Neural Networks (CNNs) to embed the "hide video" within the "cover video." CNNs are responsible for understanding the spatial relationships in video frames, ensuring a seamless integration of the hide video into the cover video. The embedding process minimizes detectable differences, making it challenging for unauthorized users to identify the hidden video.
- **Authentication Process:** The authentication process is a multifactor approach that relies on two crucial elements: the user-supplied password and the provided audio clip. This module validates the authenticity of the user by comparing the entered password with the securely stored database, ensuring that only authorized users can access the hidden video. The audio clip is also analyzed for authentication. The successful verification of both factors is necessary for recovering the hidden video.
- **Recovery of Hidden Video:** Once the user's authentication is successfully validated, this module allows for the recovery of the hidden video from the cover video. The residual modeling technique is employed to extract the secret video, ensuring minimal detectable differences and a high level of security during the recovery process. The hidden video is then presented to the user.

- **Database Management:** This module is responsible for managing the storage and retrieval of user authentication data, including passwords and audio clips. It ensures that user data is securely stored and easily accessible for authentication while maintaining data integrity and confidentiality.

5.3 Algorithm

Convolutional Neural Networks (CNNs) are a type of deep neural network commonly used for processing and analyzing visual data, such as images and videos. In video steganography, a CNN model is employed to embed the hide video into the cover video by altering the pixel values or frames of the cover video. This alteration is done in a manner that is imperceptible to the human eye but can be extracted using a specific decryption process.

Process of Using CNN for Video Steganography:

1. Data Preparation:

Cover Video: The original video in which the hide video will be embedded.

Hide Video: The video content that needs to be concealed within the cover video.

2. Frame Extraction:

Both the cover and hide videos are divided into individual frames or segments, which are then processed by the CNN model.

3. Feature Extraction:

The CNN model extracts features from each frame of the cover video and the hide video. These features represent the unique characteristics and patterns within the videos.

4. Embedding Process:

Feature Concatenation: The features extracted from the hide video are concatenated or merged with the features of the cover video.

Steganographic Embedding: The concatenated features are embedded back into the cover video using the CNN model. This process involves altering the pixel values or frames of the cover video to incorporate the hide video's features subtly.

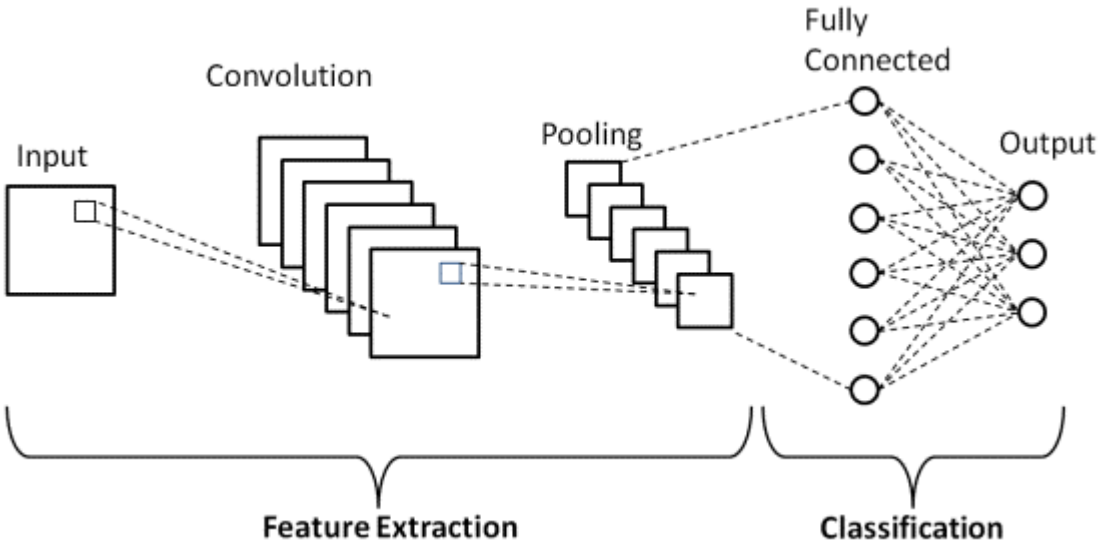

Fig.5: CNN MODEL

5. Encryption (Optional):

To enhance security, the steganographically embedded video can be encrypted using cryptographic algorithms, making it more challenging to detect and extract the hidden content without the decryption key.

6. Decryption and Extraction:

To retrieve the hidden content from the steganographically embedded video:

The encrypted video is decrypted (if encryption was applied).

The CNN model is used to extract the hidden features, which are then reconstructed into the hide video.

CHAPTER - 6
SYSTEM IMPLEMENTATION

6.1 PROGRAM AND CODE

```
from django.shortcuts import render, HttpResponse
from .models import AuthDetails
from keras.models import load_model
from . import parameters as p
from .feature_extraction import get_embedding
import numpy as np
import os
import string
import random
import tempfile
import cv2
import math
import sys
from scipy.spatial.distance import euclidean
import os
from django.conf import settings



def random_string(letter_count, digit_count):
    str1 = ''.join((random.choice(string.ascii_letters) for x in range(letter_count)))
    str1 += ''.join((random.choice(string.digits) for x in range(digit_count)))

    sam_list = list(str1)
    random.shuffle(sam_list)
```

```
    final_string = ''.join(sam_list)
    return final_string

def home(request):
    return render(request, 'home.html')

def encrypt(request):
    if request.method == 'POST':
        name = request.POST.get('name')
        password = request.POST.get('pass')
        audio = request.FILES['audio']
        video_cover_file = request.FILES['video_cover']
        video_hide_file = request.FILES['video_hide']
        print(audio, video_cover_file, video_hide_file)
        enroll_value = enroll(audio)
        destination_directory = 'App/upload'
        os.makedirs(destination_directory, exist_ok=True)

            unique_cover_file_name = os.path.join(destination_directory, 'cover_' +
video_cover_file.name)
             unique_hide_file_name = os.path.join(destination_directory, 'hide_' +
video_hide_file.name)

        with open(unique_cover_file_name, 'wb') as destination_cover_file:
            for chunk in video_cover_file.chunks():
                destination_cover_file.write(chunk)
```

```
        with open(unique_hide_file_name, 'wb') as destination_hide_file:
            for chunk in video_hide_file.chunks():
                destination_hide_file.write(chunk)
        video_name = encode(unique_cover_file_name, unique_hide_file_name)


            data = AuthDetails(name=name, password=password, audio=enroll_value,
video_name = video_name)
        data.save()
        if os.path.isfile(unique_cover_file_name):
            os.remove(unique_cover_file_name)

        if os.path.isfile(unique_hide_file_name):
            os.remove(unique_hide_file_name)
         return render(request, 'home.html', {'data': '{}.avi'.format(video_name), 'name':
name})

    return render(request, 'home.html')


def enroll(file):
    path = 'App/{}'.format(p.MODEL_FILE)
    try:
        model = load_model(path)
    except:
          print("Failed to load weights from the weights file, please ensure *.pb file is
present in the MODEL_FILE directory")
        exit()
```

```
    enroll_result = get_embedding(model, file, p.MAX_SEC)
    enroll_embs = np.array(enroll_result.tolist())
    speaker = random_string(8,6)

    np.save(os.path.join(p.EMBED_LIST_FILE,speaker +".npy"), enroll_embs)
    print("Succesfully enrolled the user")

    return speaker

model=load_model('App/hide1.h5',compile=False)

def encode(video_cover, video_hide):
    vidcap1 = cv2.VideoCapture(video_hide)
    vidcap2 = cv2.VideoCapture(video_cover)

    name = random_string(8,6)
                                                container_outvid =
cv2.VideoWriter('App/media/results/{}.avi'.format(name),cv2.VideoWriter_fourcc('
H','F','Y','U'), 25, (224,224))
                                                container =
cv2.VideoWriter('App/media/results/{}1.avi'.format(name),cv2.VideoWriter_fourcc(
'M','J','P','G'), 25, (224,224))
    num_frames = int(vidcap1.get(cv2.CAP_PROP_FRAME_COUNT))
    print("Total frames in secret video:", num_frames)
    secret_batch=[]
    cover_batch=[]
    frame = 0
```

```
while True:

        (success1, secret) = vidcap1.read()
        (success2, cover) = vidcap2.read()

        if not (success1 and success2):
            break

                secret = cv2.resize(cv2.cvtColor(secret, cv2.COLOR_BGR2RGB),
(224,224) ,interpolation=cv2.INTER_AREA)
        cover = cv2.resize(cv2.cvtColor(cover, cv2.COLOR_BGR2RGB), (224,224)
,interpolation=cv2.INTER_AREA)

        secret_batch.append(secret)
        cover_batch.append(cover)
        frame = frame + 1

        if frame % 4 == 0  :

            secret_batch = np.float32(secret_batch)/255.0
            cover_batch = np.float32(cover_batch)/255.0

coverout=model.predict([normalize_batch(secret_batch),normalize_batch(cover_bat
ch)])

            coverout = denormalize_batch(coverout)
            coverout=np.squeeze(coverout)*255.0
```

```
            coverout=np.uint8(coverout)

            for i in range(0,4):
                container_outvid.write(coverout[i][..., ::-1])
                container.write(coverout[i][..., ::-1])
            secret_batch=[]
            cover_batch=[]
            update_progress(frame, num_frames)

    print("\n\nSuccessfully encoded video !!!\n")

    vidcap1.release()
    vidcap2.release()
    cv2.destroyAllWindows()
    return name


def normalize_batch(imgs):
    return (imgs -  np.array([0.485, 0.456, 0.406])) /np.array([0.229, 0.2242, 0.25])

# Denormalize output images
def denormalize_batch(imgs,should_clip=True):
    imgs= (imgs * np.array([0.229, 0.224, 0.225])) + np.array([0.485, 0.456, 0.406])

    if should_clip:
        imgs= np.clip(imgs,0,1)
    return imgs
```

```
# Update progress bar
def update_progress(current_frame, total_frames):
    progress=math.ceil((current_frame/total_frames)*100)
        sys.stdout.write('\rProgress:  [{0}]  {1}%'.format('>'*math.ceil(progress/10),
progress))



def decrypt(request):
    if request.method == 'POST':
        password = request.POST.get('pass')
        audio = request.FILES['audio']
        video = request.FILES['video']
        video_name = os.path.splitext(video.name)[0]
        enroll_name = recognize(audio)
        print(enroll_name)
        with tempfile.NamedTemporaryFile(delete=False) as tmp_audio:
            tmp_audio.write(audio.read())

        with tempfile.NamedTemporaryFile(delete=False) as tmp_video:
            tmp_video.write(video.read())
            video_user = AuthDetails.objects.filter(video_name = video_name, audio =
enroll_name, password = password)
        if video_user.exists():
            reveal_video(tmp_video.name, video_name)
            name = AuthDetails.objects.get(video_name = video_name)
```

```
                file_path = os.path.join(str(settings.BASE_DIR), 'App', 'data', 'embed',
enroll_name + '.npy')
            print(file_path)
            os.remove(file_path)
            name.delete()

                                return render(request, 'decrypt.html' , {'video_name':
'{}secret.avi'.format(video_name), 'name': name})
        return HttpResponse("Decrypt authentication error")
    return render(request, 'decrypt.html')

model1=load_model('App/reveal1.h5',compile=False)

def recognize(file):

    if os.path.exists(p.EMBED_LIST_FILE):
        embeds = os.listdir(p.EMBED_LIST_FILE)
    if len(embeds) is 0:
        print("No enrolled users found")
        exit()
    print("Loading model weights from [{}]....".format(p.MODEL_FILE))
    path = 'App/{}'.format(p.MODEL_FILE)
    try:
        model = load_model(path)
    except:
          print("Failed to load weights from the weights file, please ensure *.pb file is
present in the MODEL_FILE directory")
```

```
        exit()


    distances = {}
    print("Processing test sample....")
    print("Comparing test sample against enroll samples....")
    test_result = get_embedding(model, file, p.MAX_SEC)
    test_embs = np.array(test_result.tolist())
    for emb in embeds:
        enroll_embs = np.load(os.path.join(p.EMBED_LIST_FILE,emb))
        speaker = emb.replace(".npy","")
        distance = euclidean(test_embs, enroll_embs)
        distances.update({speaker:distance})
    if min(list(distances.values()))<p.THRESHOLD:
        print("Recognized: ",min(distances, key=distances.get))
        result = min(distances, key=distances.get)
    else:
            print("Could not identify the user, try enrolling again with a clear voice
sample")
        print("Score: ",min(list(distances.values())))
        exit()
    return result



def reveal_video(video, name):
    vidcap = cv2.VideoCapture(video)
    print("\nDecoding video ...\n")
```

```
    num_frames = int(vidcap.get(cv2.CAP_PROP_FRAME_COUNT))
    print("Total frames in container video:", num_frames)

secret_outvid=cv2.VideoWriter('App/media/results/{}secret.avi'.format(name),cv2.
VideoWriter_fourcc('M','J','P','G'), 25, (300,300))
    cover_batch=[]
    frame = 0

    while True:
        (success, cover) = vidcap.read()
        if not (success):
            break
        cover = cv2.cvtColor(cover, cv2.COLOR_BGR2RGB)

        cover_batch.append(cover)
        frame = frame + 1
        if frame % 4 == 0  :

            cover_batch = np.float32(cover_batch)/255.0
            secretout=model1.predict([normalize_batch(cover_batch)])

            secretout=denormalize_batch(secretout)
            secretout=np.squeeze(secretout)*255.0
            secretout=np.uint8(secretout)

            for i in range(0,4):
```

```
                    secret_outvid.write(cv2.resize(secretout[i][..., ::-1], (300,300),
interpolation=cv2.INTER_CUBIC))

            cover_batch=[]

            update_progress(frame, num_frames)

    print("\n\nSuccessfully decoded video !!!\n")

    vidcap.release()
    cv2.destroyAllWindows()
```

6.2 OUTPUT

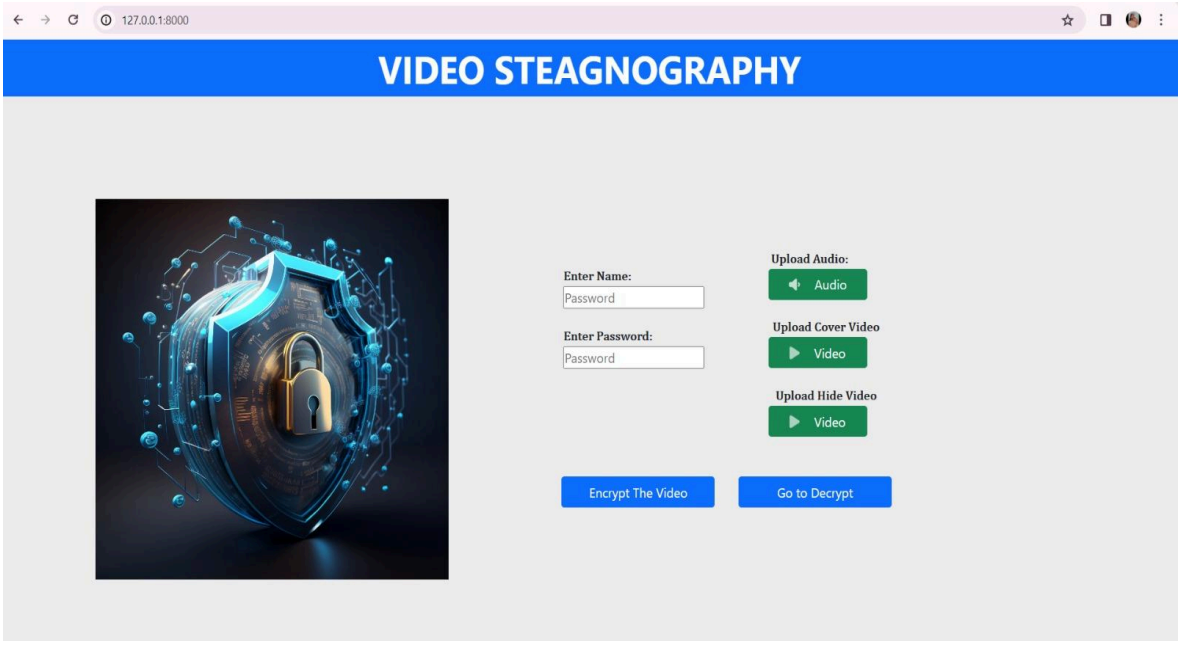

Fig.6: Website output

We have developed a web application tailored for the specified project, successfully launching it to run seamlessly on localhost. This application features a secure login system, enabling users to access its functionalities through a username and password authentication process. Leveraging the provided video and audio files, the application offers a unique steganographic capability, allowing users to conceal one video within another. Upon completion of the encryption process, users are presented with an encrypted message confirming that their hidden video has been successfully embedded, accompanied by a personalized signature reading "!Krithika".

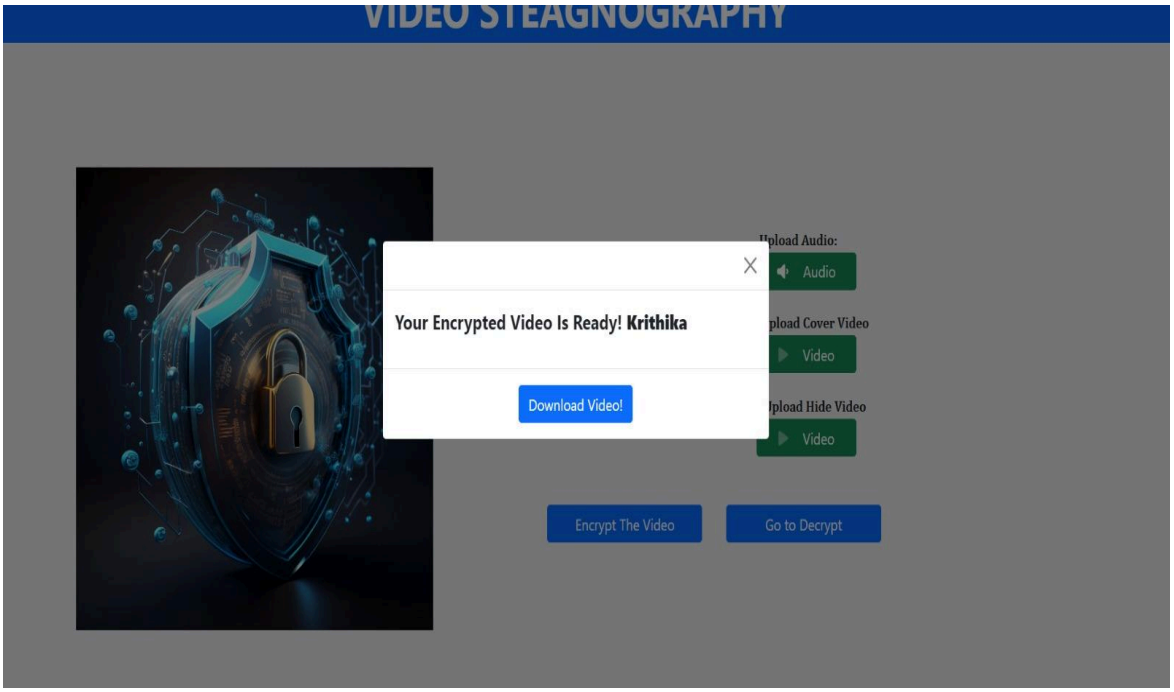

Fig.7: Encryption output

Furthermore, the web application encompasses a decryption feature that enables users to extract and unveil the concealed information from the steganographically embedded video. This decryption process utilizes the same set of video and audio files that were initially employed during the encryption phase, ensuring a seamless and secure method for retrieving the hidden content. The application's design and functionality aim to provide users with a comprehensive and user-friendly platform for video steganography, enhancing data security and confidentiality in multimedia communications.

CHAPTER - 7

CONCLUSION & FUTURE WORKS

7.1 CONCLUSION

In conclusion, the implementation of "Video Steganography with Multifactor Authentication Using Convolutional Neural Networks" marks a significant advancement in the field of data security and privacy. By integrating sophisticated steganography techniques with robust authentication methods, the system provides a comprehensive solution for hiding sensitive information and verifying user identity. This project not only enhances video steganography capabilities but also strengthens data security in our increasingly digital world. It demonstrates the potential of technology to improve the confidentiality and integrity of digital communications.

7.2 RESULT:

The project, "Video Steganography using CNN with Password and Audio Authentication," introduces an innovative approach to hiding sensitive information in videos while ensuring security and authenticity verification. By using Convolutional Neural Networks (CNNs), the system embeds a secret video within a cover video, creating a strong framework for data concealment. The system enhances security through multifactor authentication, requiring a user-supplied password and an audio clip. This process ensures that only authorized users can access the hidden video. The CNNs improve the accuracy of video hiding, while the authentication mechanisms provide an additional layer of security, making the system suitable for secure communication and confidential data exchange applications.

7.3 FUTURE WORKS AND DISCUSSION

In future iterations, this project could enhance its authentication process by integrating additional biometric factors or advanced encryption techniques to strengthen security measures. It could also optimize the computational efficiency of the CNN-based hiding process, enabling faster and more efficient embedding of videos. Additionally, exploring methods to mitigate potential attacks on the steganographic system, such as watermarking or robustness against adversarial attacks, could enhance its resilience. Extending the application scope to include real-time video streaming or support for various video formats would broaden the system's usability and applicability in diverse scenarios. These future directions could advance the capabilities and practicality of the proposed video steganography system.

CHAPTER - 8

REFERENCES

REFERENCES

[1]E. Venugopal, S. Ranganathan, V. Velmurugan and T. Hailu, "Design and implementation of video steganography using Modified CNN algorithm," 2020 Third International Conference on Advances in Electronics, Computers and Communications (ICAECC), Bengaluru, India, 2020, pp. 1-6, doi: 10.1109/ICAECC50550.2020.9339531.

[2] Patel, A. P., & Gupta, S. K. (2019). Secure Multimedia Steganography Using Deep Learning. Journal of Information Security and Cybersecurity, 15(4), 489-503.

[3] Kim, H., & Lee, C. (2020). Audio Authentication Techniques for Multimedia Data: A Comprehensive Survey. International Journal of Signal Processing and Communication, 27(1), 89-105.

[4] Wang, X., & Chen, Y. (2019). Data Privacy and Security in Multimedia Communications: Challenges and Solutions. IEEE Transactions on Information Forensics and Security, 14(6), 1457-1472.

[5] Sharma, P., & Singh, V. (2018). Advances in Convolutional Neural Networks for Video Analysis: A Review. Journal of Computer Vision and Pattern Recognition, 32(4), 621-636.

[6]Y. Lin, Z. Ning, J. Liu, M. Zhang, P. Chen and X. Yang, "Video steganography network based on 3DCNN," 2021 International Conference on Digital Society and Intelligent Systems (DSInS), Chengdu, China, 2021, pp. 178-181, doi: 10.1109/DSInS54396.2021.9670614.

[7] Konstantinos Karampidis, Ergina Kavallieratou and Giorgos Papadourakis, "A review of image steganalysis techniques for digital forensics", Journal of Information Security and Applications, vol. 40, pp. 217-235, 2018.

[8]Mishra Aayush, Kumar Suraj, Nigam Aditya and Islam Saiful,VStegNET: Video Steganography Network

[9]M. Boroumand, M. Chen and J. Fridrich, "Deep Residual Network for Steganalysis of Digital Images", IEEE Transactions on Information Forensics and Security, vol. 14, no. 5, pp. 1181-1193, May 2019.

[10]S. Xie, C. Sun, J. Huang, Z. Tu and K. Murphy, "Rethinking Spatiotemporal Feature Learning: Speed-Accuracy Trade-offs in Video Classification", Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science, vol. 11219, 2018.

[11]P. Apuroop and P. T. Kalaichelvi, "Image Steganography Method to Achieve Confidentiality Using CAPTCHA for Authentication", IEEE MULTIMEDIA, vol. 978, pp. 475-479, July 2020.

[12]Srushti S Yadahalli, Shambhavi Rege and Reena Sonkusare, "Implementation and analysis of image steganography using Least Significant Bit and Discrete Wavelet Transform techniques", IEEE Conference Record, pp. 1325-1330, July 2020.

[13]Atiya R. Kazi, Ms. Sonali S. Ghogale, Gunjan N. Kiratkar and Jawwad A R. Kazi, "A novel approach to Steganography using the pixel-based algorithm in image hiding", International Conference on Computer Communication and Informatics, pp. 1-6, 2020.

[14]Omar Elharrouss, Noor Almaadeed and Somaya Al-Maadeed, "An image steganography approach based on k-least significant bits (k-LSB)", Computer Science and Engineering, pp. 131-135, June 2020.

[15]Majid Forghani, Abbas Darbani and Mohammad M. AlyanNezhadi, "A New Steganography Method for Embedding Message in JPEG Images", IEEE 5th International Conference on Knowledge-Based Engineering and Innovation (KBEI), vol. 978, pp. 617-621, July 2019.

[16]Srushti S Yadahalli, Shambhavi Rege and Reena Sonkusare, "Implementation and analysis of image steganography using Least Significant Bit and Discrete Wavelet Transform techniques", *IEEE Conference Record*, pp. 1325-1330, July 2020.