Trickk-An Application of Steganography

Submitted in partial fulfillment of the requirements of the degree

BACHELOR OF ENGINEERING IN **COMPUTER ENGINEERING**

By

Shriya Bijam Roll. No. 11

Tushar Amdoskar Roll. No. 02

Bharat Choudhary Roll. No. 17

Supervisor

Prof. Ranjita Asati

Department of Computer Engineering

Atharva College of Engineering

Malad (W), Mumbai - 400 095

University of Mumbai

(AY 2020-21)

CERTIFICATE

This is to certify that the Mini Project entitled **"Trickk-An Application of Steganography"** is a bonafide work of **Shriya Bijam (Roll No. 11), Tushar Amdoskar (Roll No. 02), and Bharat Choudhary (Roll No. 17)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **"Bachelor of Engineering"** in **"Computer Engineering".**

Prof. Ranjita Asati
Supervisor

Dr. Suvarna Pansambal
Head of Department

Dr. S. Kallurkar
Principal

Mini Project Approval

This Mini Project entitled "Trickk-An Application of Steganography" by **Shriya Bijam (Roll No. 11), Tushar Amdoskar (Roll No. 02), Bharat Choudhary (Roll No. 17)** is approved for the degree of **Bachelor of Engineering** in **Computer Engineering.**

Examiners

1……
(Internal Examiner Name & Sign)

2……
(External Examiner name & Sign)

Date:

Place:

Contents

Abstract

Innovation of technology and having fast Internet make information to distribute over the world easily and economically. This is made people to worry about their privacy and works.

Communication of data by maintaining confidentiality is a major issue everywhere, so to increase the security a non – conventional approach called steganography is proposed. “Steganography” is an art of “hiding data within data”. In general, Stego means “covering” and graphia means “writing”. Combining these two terms gives the meaning of steganography, i.e. “covered writing”

It is similar to cryptography with a small difference; cryptography makes use of scrambling of data techniques whereas steganography makes use of the carrier file of some type to store the information. It is easy to capture a cryptographic message but it is not easy to know the steganographic message. Steganography’s objective is to transmit a message from one end to the other via harmless carrier/medium. By making use of this method, the attacks on the message are very rare because the attacks do not know what the actual message is.

Embedion with steganography will produce much more secure data communication. It can be used for different types of data formats, of which the most popular are .gif, .txt, .wav, .bmp and .jpeg. These types of files are called carriers. In general, the most common examples of steganography are invisible ink, covert channel and microdot. This concept is used everywhere or in any technology to send confidential information.

For example, some applications may require absolute invisibility of the secret information while others require a larger secret message to be hidden. This project hides the message within the image. For more secure approach, the project it allows the user to choose the bits for replacement instead of LSB replacement from the image. Sender select the cover image with the secret text or text file and hide it into the image with the replacement choice. It help to generate the secure stego image , the stego image is sent to the destination with the help of private or public communication network – on the other side i.e. receiver . Receiver download the stego image and using the software retrieve the secret text hidden in the stego image.

Keywords: Steganography, stego, image processing, full stack web development, Django, React.js, OpenCV LSB (Least Significant Bit) technique

Acknowledgement:

We are pleased to present this Project report entitled “Trickk-An Application of Steganography”. It is indeed a great pleasure and a moment of immense satisfaction for us to express our sense of profound gratitude and indebtedness towards our guide Prof. Ranjita Asati whose enthusiasm are the source of inspiration for us.

We are extremely thankful for the guidance and untiring attention, which bestowed on us from the beginning. Would also like to give our sincere thanks to Dr. Suvarna Pansambal Head of Department of “COMPUTER ENGINEERING” for necessary help and providing us the best faculties for completion of our project.

Last but not the least, we would like to thank the entire Teaching Staff to bring forward our project report.

Submitted By,
Shriya Bijam, Roll No.: 11
Tushar Amdoskar, Roll No.: 02
Bharat Choudhary, Roll No.: 17

Under the guidance of
Ranjita Asati

List of Figures

1 Introduction

1.1 Introduction:

Art of data hiding in digital media, steganography and watermarking, aims to embed secret data into cover with purpose of identification, copyright protection, and annotation. The main constraint factors of this process are message data quantity, necessity of invariability of embedded data under distortions like lossy compression, third party removal, or modification. Data hiding techniques fall into three categories of cryptography, steganography and watermarking. Watermarking and particularly steganography tend to conceal presence of hidden data while cryptography makes data gibberish.

In this project, we attempt to implement Image Steganography through an interactive web application, thus bringing a complex concept to public. This full stack application uses Django as the back-end framework, React.js as the front-end framework and OpenCV as the support for image processing.

1.2 Motivation:

The motivation behind developing image Steganography methods according to its use in various organizations to communicate between its members, as well as, it can be used for communication between members of the military or intelligence operatives or agents of companies to hide secret messages or in the field of espionage.

The team members have a good grasp over Python programming language and are interested in Web Frameworks like Django (Back-end) and React.js (Front-end) and Image processing, we picked Image Steganography as our Mini Project-A for semester 3. As we researched more about the concept of steganography, we came across its types, techniques and applications. Hence we took up this concept as main implementation in our Project – "Trickk". Not only it helped strengthening our knowledge of programming but also helped us in exploring new web development technologies.

1.3 Problem Statement & Objectives:

A. Problem Statement:

The main goal of steganography is to hide the existence of any secret data from the eye of a third party, so the resultant stego-image must appear normal and not suspicious after it embeds the secret data. The stego-image must have an acceptable quality in comparison with its size. These problems will be harder to solve when the secret data is a colored image, because there is more data to embed.

Hiding the data is not enough to be sure that your data is safe, to improve the security it is very important to use some encryption with the hidden secret data.

Embedding the secret colored image data within the cover image will affect the quality of the stego-image. To increase the quality of the stego-image in general it is important to manipulate fewer bits of the cover with the secret data. Another important thing in image inside image steganography is how to retrieve the data in a right way to reconstruct the secret image. To reconstruct the secret image in a right way some information is needed more than the data of the image itself like the length of data.

B. Objectives:

In order to send and retrieve a colored secret image safely on the Internet, it is important that the size of stego-image is not very big, the quality is good, and that it is to be sure that no third party can notice the existence of the secret image. It is very important to use a propitiate cryptography method to be sure that your data is safe even a third party knew about the existence of the secret data. In order to achieve that, the following goals should be completed:

- Reduce the secret image data by transferring it from RGB colored image to 8-bit or less indexed image.
- Discard sending any information about the resolution of the secret, and using a cover image has the same resolution as the secret image instead. That can be done by using the alpha channel to embed the extra data without using a cover with bigger resolution
- Encrypt the secret image while embedding it by dividing the secret data to blocks and using a portion of every block data (no extra bits) as an indicator and XORing it with bits of key to shuffle the data.

2 Literature Survey:

2.1 Types of Steganography:

A. Text Steganography:

Text Steganography is hiding information inside the text files. It involves things like changing the format of existing text, changing words within a text, generating random character sequences or using context-free grammars to generate readable texts.

B. Image Steganography:

Hiding the data by taking the cover object as the image is known as image steganography. In digital steganography, images are widely used cover source because there are a huge number of bits present in the digital representation of an image. There are a lot of ways to hide information inside an image. Common approaches include:

- Least Significant Bit Insertion
- Masking and Filtering
- Redundant Pattern Encoding
- Encrypt and Scatter
- Coding and Cosine Transformation

C. Video Steganography:

In Video Steganography you can hide kind of data into digital video format. The advantage of this type is a large amount of data can be hidden inside and the fact that it is a moving stream of images and sounds. You can think of this as the combination of Image Steganography and Audio Steganography.

D. Audio Steganography:

In audio steganography, the secret message is embedded into an audio signal which alters the binary sequence of the corresponding audio file. Hiding secret messages in digital sound is a much more difficult process when compared to others, such as Image Steganography.

E. Network Steganography:

It is the technique of embedding information within network control protocols used in data transmission such TCP, UDP, and ICMP etc. You can use steganography in some covert channels that you can find in the OSI model. For Example, you can hide information in the header of a TCP/IP packet in some fields that are either optional.

F. PowerPoint file Steganography :

A new type of steganography approach has been proposed named PowerPoint file Steganography". Data hiding is done in the sound/custom animation effects of the PowerPoint file. To implement this technique, a codebook is designed which contains the confidential details of the sound effects and the text information that needs to be transferred.

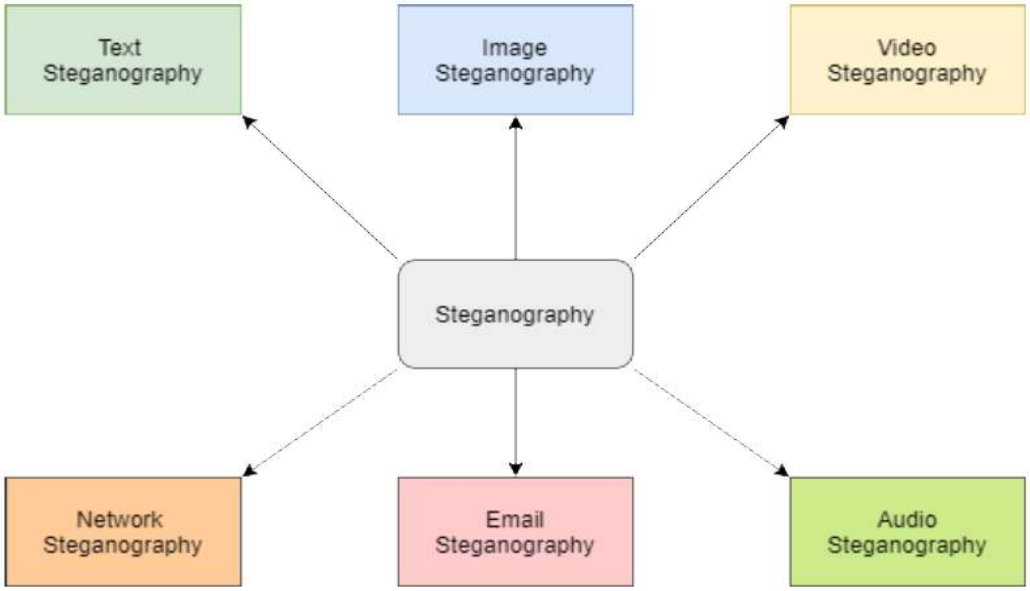

Fig. 1: Types of Steganography

2.2 Techniques of Image Steganography:

A. Spatial Domain Watermarking: There are many algorithms using original data, such as video, image, audio, and text, to hide specific information like logos or personal signatures in a spatial domain. In other words, if the original data is an image, processing would be into the pixel values without changing the data into another domain. The widest and simplest method in spatial domain is Least Significant Bit (LSB), which is replacing the first bit in each pixel by information that intends to hide.

B. Least Significant Bit Watermarking: LSB is the one of the oldest and simplest algorithms that allows users to hide their information using spatial domain. The human eye cannot recognize the difference that occurs in the two first bits in each pixel. In other words, the change in the least significant bit does not affect the image's quality. 24-bit images have three LSB because each RGB channel has its own LSB. This provides users with more storage capacity to embed the information that is necessary to hide.

C. Frequency Domain Watermarking: This is also called transform domain, because the original data changes from

spatial to frequency domain. The most common frequency methods are Discrete Fourier Transformation (DFT), Discrete Wavelet Transformation (DWT), and Discrete Cosine Transformation (DCT). For example, an 8-bit image with a 256 by 256 resolution can be transformed into frequency watermarking using DWT. The result of this processing would be four small images, each of them with a 128 by 128 resolution. Moreover, four images will have different frequency ranges from low to high because each of them has different coefficients for others. The main advantage of using frequency domain watermarking is that it is robust for many kinds of signal manipulations when sending data via the Internet. Also, it resists of many noises that attack embedded information.

D. Discrete Wavelet Transform: It is a tool to transform the signal or data from one domain which is a spatial to another domain which is a frequency. In the frequency domain the signal splits into the two half one of them is high frequency and another is low frequency. Then each of them is going to divide again into high and low frequency that four different parts of signal [10]. Four parts or sub bands of decomposed signal are LL, LH, HL and HH frequencies which are low-low, low-high, high-low and high-high frequencies [10; 29]. Low frequency is the same of original signal and other parts are more details of signal they are not exact data as original one, so we can change or remove depends on the technique that we using. The reconstruction process is the opposite of decomposition process that means the four bands of divided data have to be mixed again to recover the original data. Sometimes we do more than one level of decomposition depends on the algorithm that we use. Low-low frequency band will be used in case we do second decomposition. In case of reconstruction the last level of decomposition will used first which is an exact opposite direction.

3 Proposed System:

3.1 Architecture:

The project aims at implementing Image Steganography through a Web Application which enable users to hide an image inside the other image by uploading them. The user can also obtain the original images by uploading the encoded image.

The web application is protected using login credentials and the functionalities of encrypting and decrypting can only be accessed after logging in. The user can only see the images that were uploaded and processed through his account.

The Web Application is developed using Django framework for back-end and React.js framework for front-end. Image processing is powered by OpenCV and Pillow libraries of Python. The RESTful API architecture is supported by Django-Rest-Framework while the token authentication uses Django-Knox.

- List of API Routes used:
  1. 'api/auth/register/': To register new user
  2. 'api/auth/login/': To login a user using username and password.
  3. ‘api/auth/user/’: To check if any user is logged in or not.
  4. 'api/auth/logout/': To logout a user.
  5. 'api/decrypt/': To decrypt an encrypted image.
  6. 'api/encrypt/': To encrypt an image using other image.

3.2 Algorithm:

There are numerous algorithms available for image steganography. For the implementation of this project we have opted the **Least Significant Bit Steganography (LSB) technique**.

Pixels are the smallest individual element of an image, holding values that represent the brightness of a given color at any specific point. Hence one can represent an image as a matrix (or a two-dimensional array) of pixels containing a fixed number of rows and columns.
Least Significant Bit (LSB) is a technique in which the last bit of each pixel is modified and replaced with the secret message’s data bit. As observed in the given picture, change in the least significant bit results in the least change of the value whereas any change in the most significant bit results in a very noticeable difference from the original value.

Fig. 2: Working of LSB Technique

In the classical LSB embedding methods, the secret message is inserted in to the least-significant bit plane of the cover image either by directly replacing those bits. The amount of data to be embedded may also be fixed or variable in size depending on the number of pixels selected. The main advantage of such a technique is that the modification of the LSB plane does not affect the human perception of the overall image quality as the amplitude variation of the pixel values is bounded by ±1. The masking properties of the Human Visual System allow significant amounts of embedded information to be unnoticed by imperceptible by the average observer under normal viewing conditions.

Thus summarizing the entire idea, each pixel has three values (RGB), each RGB value is 8-bit and the rightmost bits are least significant. So, if we change the rightmost bits it will have a small visual impact on the final image. This is the steganography key to hide an image inside another. Change the least significant bits from an image and include the most significant bits from the other image. An example of the working of the algorithm is as follows.

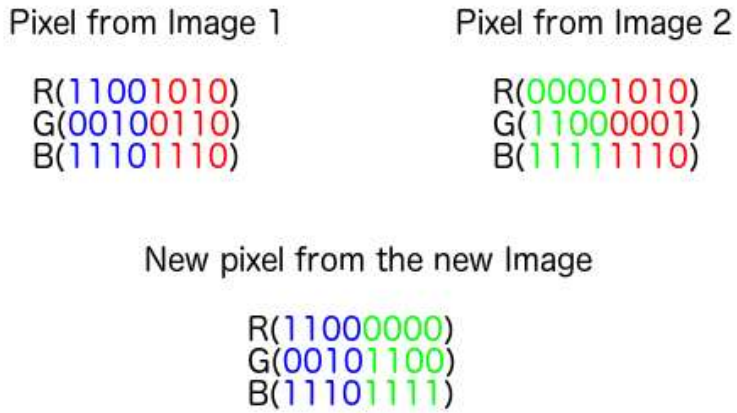

Fig. 3: Example of LSB Technique

In this case, image 2 is hidden in image 1 using the LSB technique of Steganography. The new images looks similar to the image 1.

3.3 Process Design:

1. Encrypting:

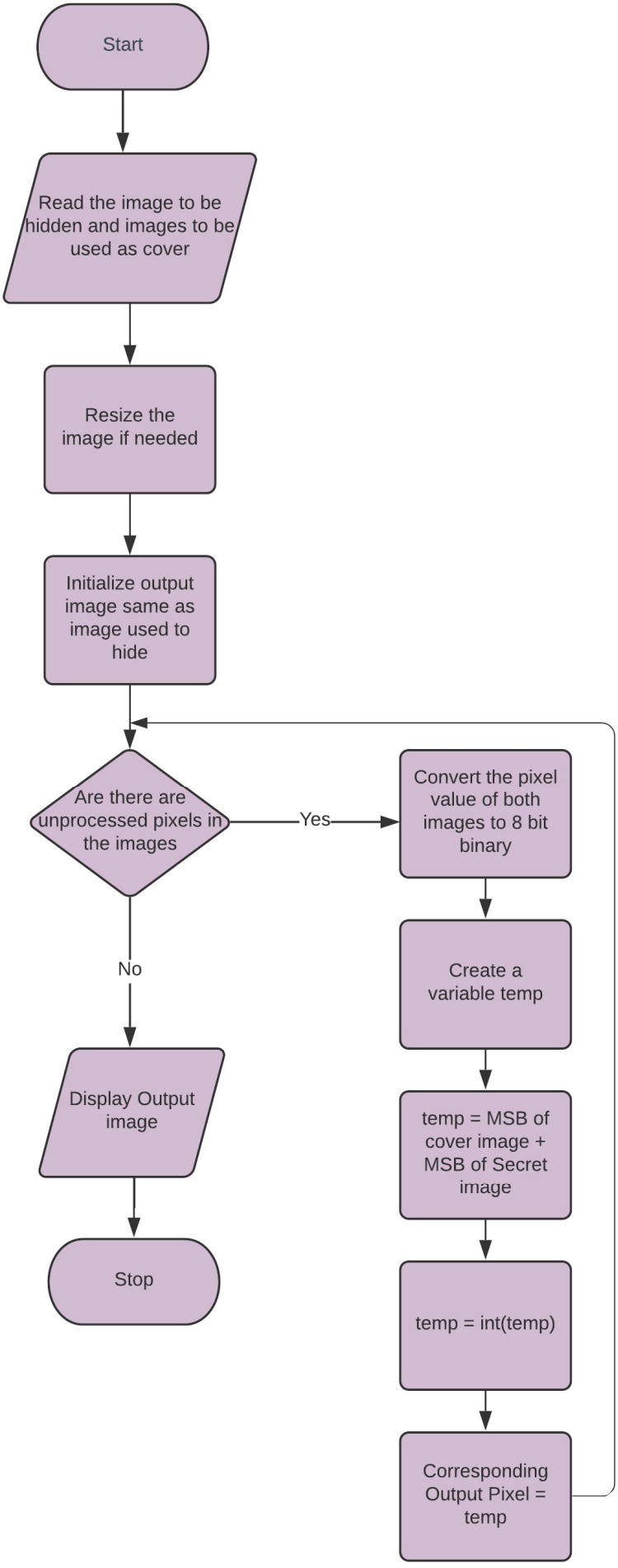

Fig. 4: Flowchart of encrypting process

2. Decrypting:

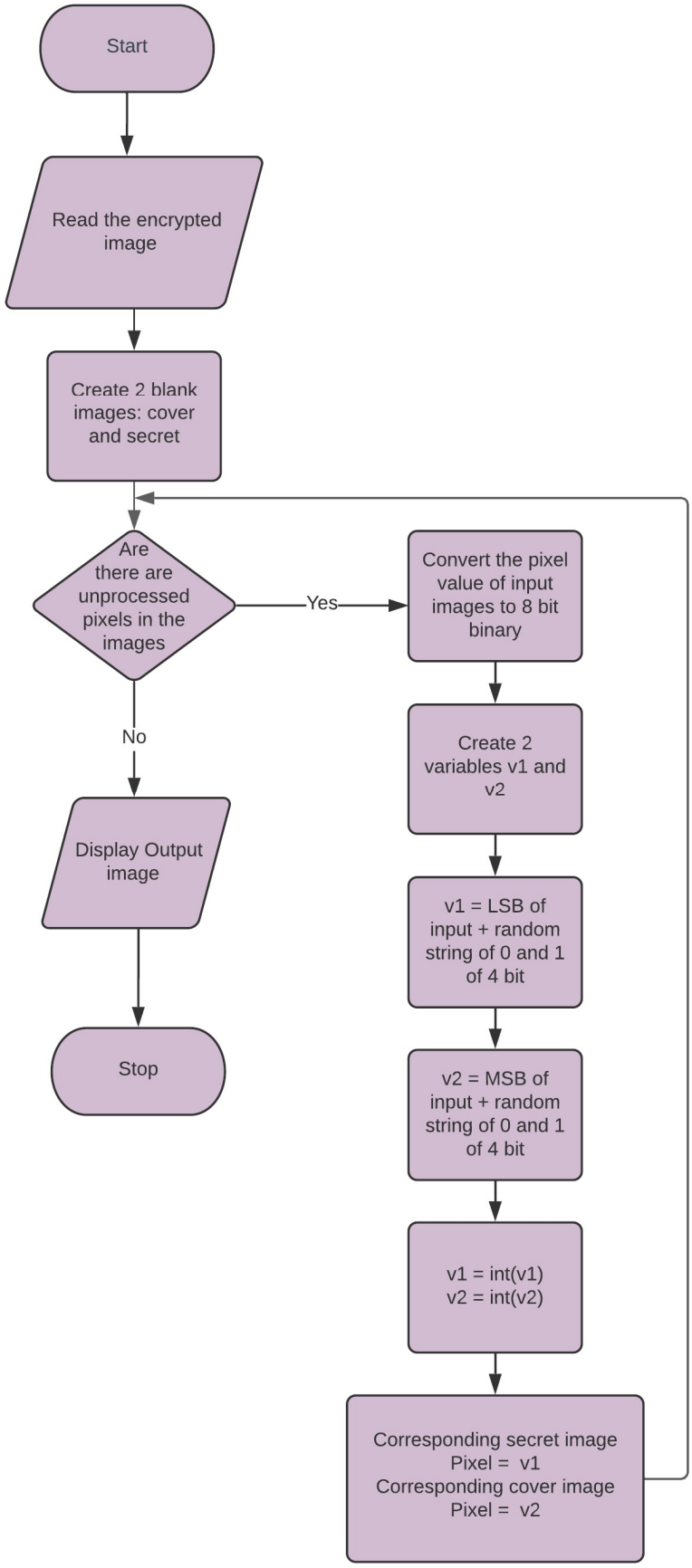

Fig. 5: Flowchart of decrypting process

3.4 Details of Hardware & Software:

A. **Hardware Requirements:** The project is implemented as a Web Application, hence there are **no hardware requirements**.

B. **Software Requirements:** The software side of the project is divided into two parts: Back-end functionality using Django and Front-end using React.js. The project uses API architecture for communication between back-end and front-end.

1. **Back-end server:**

- **Django == 3.1.3:** Main back-end framework.
- **django-rest-knox == 4.1.0:** For token-based authentication (register, login and logout).
- **Djangorestframework == 3.12.1:** For building RESTful APIs to transfer information to and from front-end.
- **Numpy == 1.19.4, opencv-python == 4.4.0.46 and Pillow == 8.0.1:** For making image processing efficient and less complex.
- **Sqlite:** Currently, sqlite is used as the database for this project.

2. **Front-end server:**

- **Node.Js v12.18.0:** For executing React.js codes.
- **react: ^17.0.1 and react-dom: ^17.0.1:** As the main front-end framework.
- **axios: ^0.21.0:** For communicating through HTTP requests with the back-end server.
- **bootstrap: ^4.5.3 and react-bootstrap: ^1.4.0:** For responsive and beautiful webpages.
- **react-dropzone: ^11.2.4:** For enabling drag and drop functionality for files.

3.5 Experiment and Results:

1. **Landing Page:** The user sees the landing page when he first visits the site. This page consists of the site information and options to login and register.

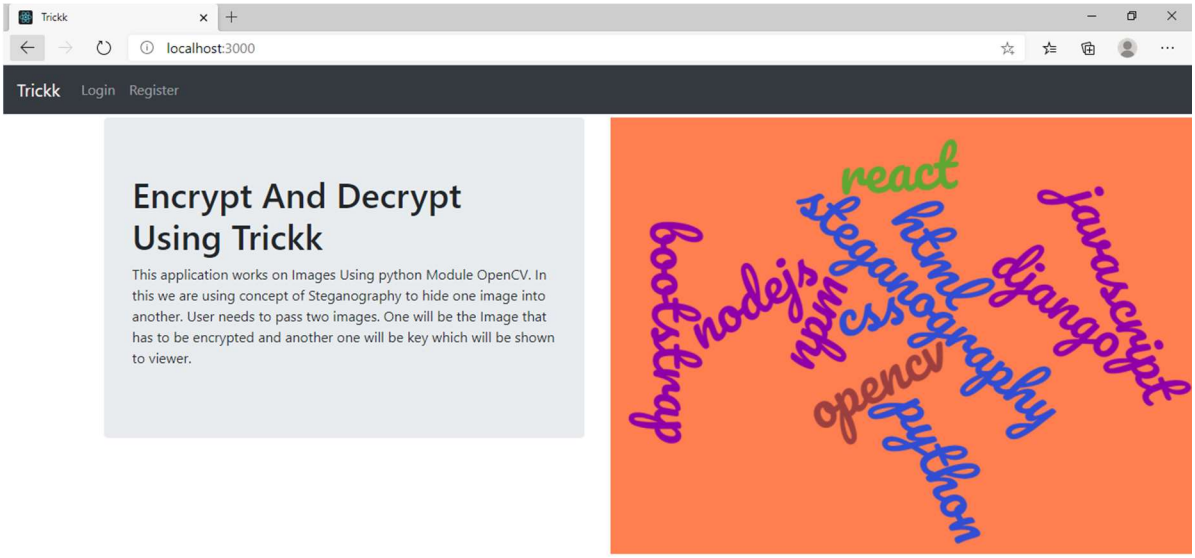

Fig. 6: Screenshot of Landing Page

2. **Register Page:** In order to access the functionalities of site the user needs to create an account by specifying the username and password.

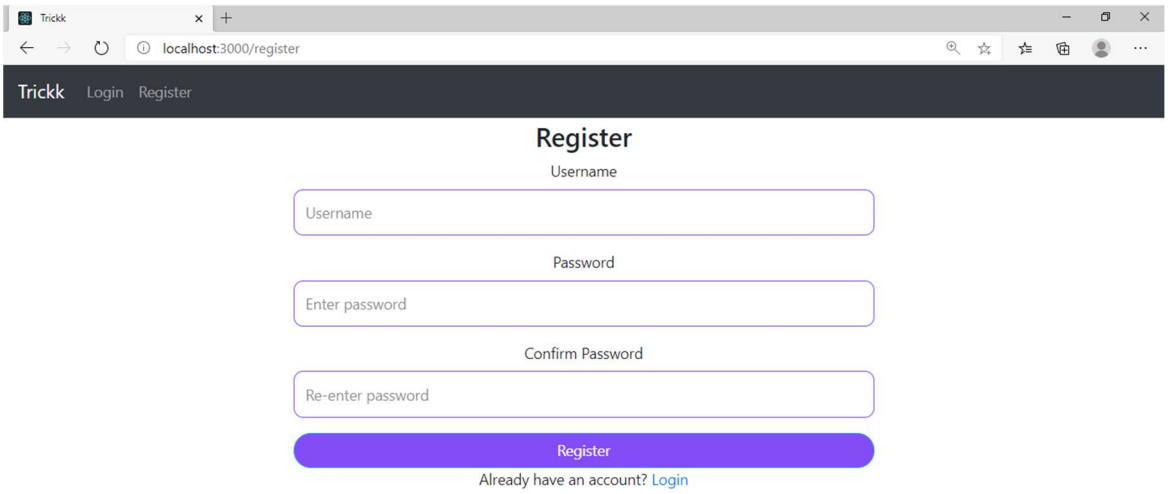

Fig. 7: Screenshot of Register Page

3. **Login Page:** A registered user can access the site by logging in using his login credentials.

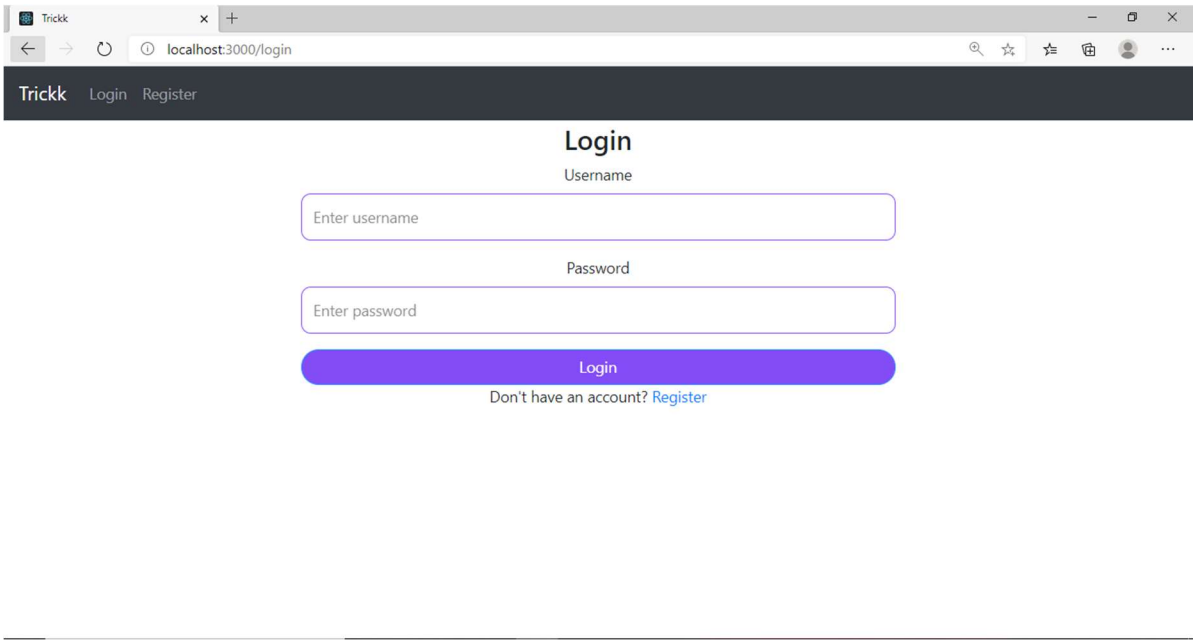

Fig. 8: Screenshot of Login Page

4. **Dashboard:** The user is redirected to the dashboard screen after successful login. This page showcases the images that were previously encrypted and decrypted by the user, if any.

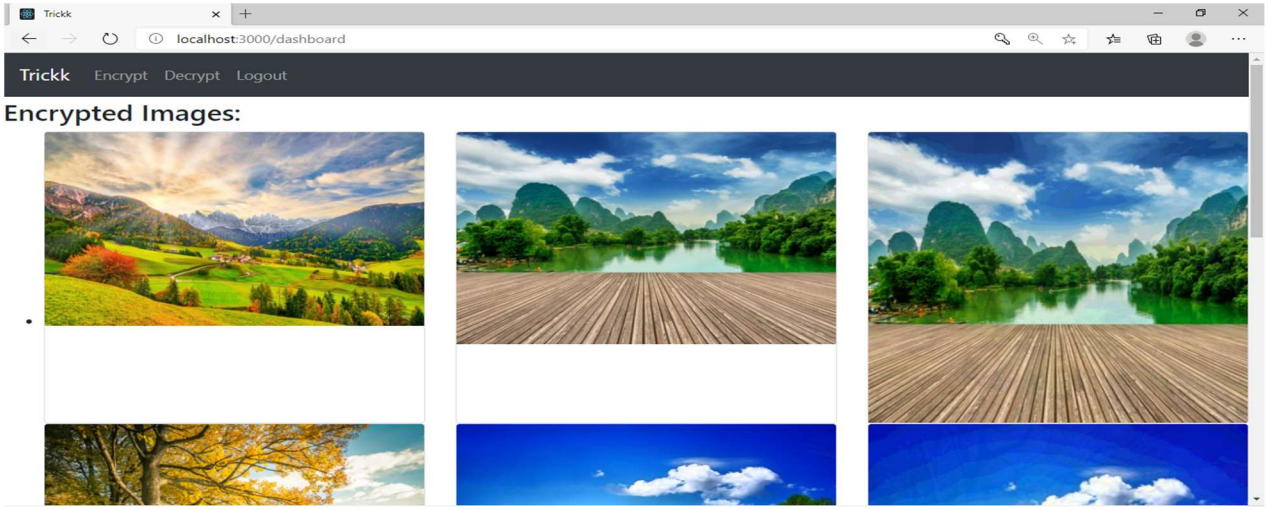

Fig. 9: Screenshot of Dashboard Page

5. **Encrypt:** This pages takes the two images and hides one of them using the other image. The user drags and drops the images and clicks on the submit button. Then the images are sent to the server and the encrypted image is returned after successful encryption.

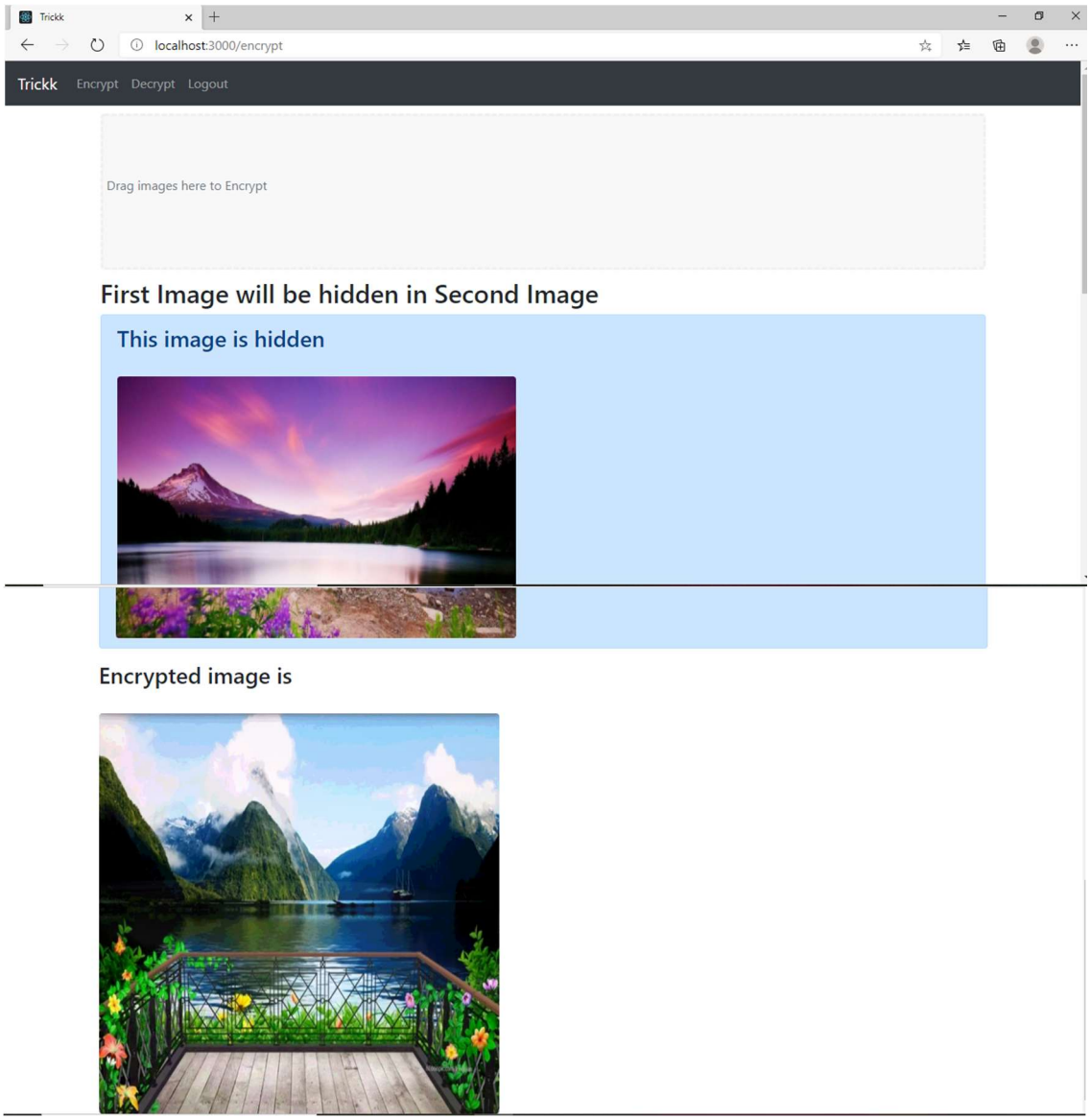

Fig. 10: Screenshot of Encrypt Page

6. **Decrypt:** This pages takes the encrypted image and returns the original two images. The user drags and drops the image and clicks on the submit button. Then the image is sent to the server and the decrypted image is returned after successful decryption.

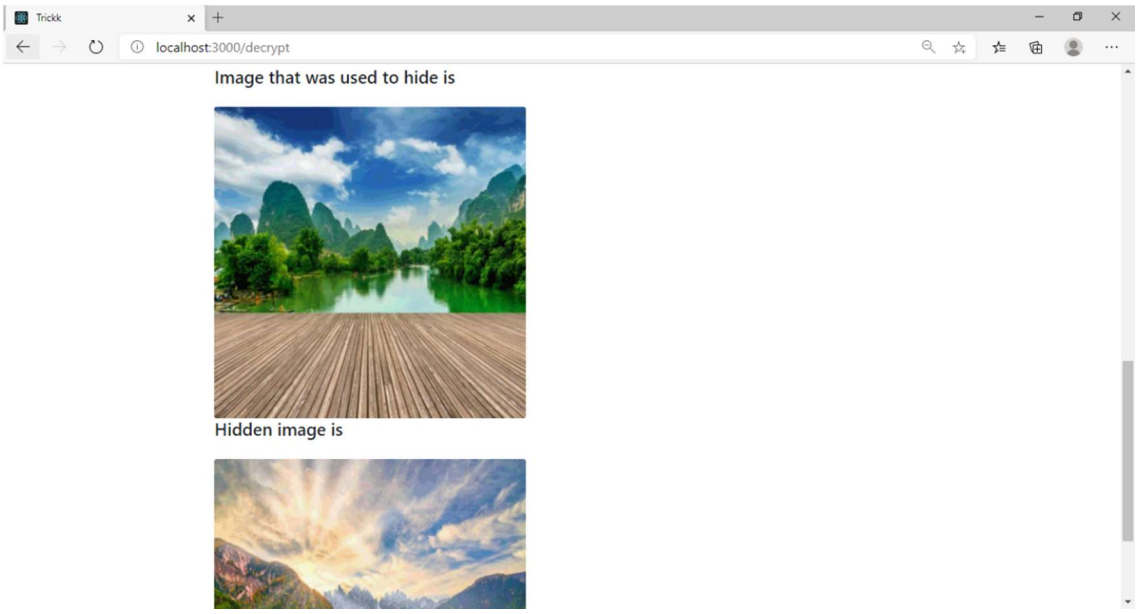

Fig. 11: Screenshot of Decrypt Page

3.6 Conclusion and Future work:

This project implemented the oldest and most basic algorithm of Steganography named as **Least Significant Bit (LSB) technique**. We developed a web application using Django and React.js as the back-end and front-end frameworks, respectively, that allows the user to encrypt and decrypt images after registering and logging in to the website.

We came across various types and techniques of Steganography and chose Image Steganography for its visual impact and dynamic nature. After considering various algorithms for encryption and decryption, we picked the LSB technique as this technique was simple and efficiently implementable using Python library OpenCV. We learnt web development technologies like Django and React.js.

We plan to further extend this project by implementing some more forms of Steganography and more advance and efficient algorithms of Image Steganography. We also aim to improve the User Interface and make our web application more exciting for the users.

4 References:

4.1 Alaa A. Jabbar Altaay, Shahrin bin Sahib, Mazdak Zamani "An Introduction to Image Steganography Techniques" November 2012 Conference: International Conference on Advanced Computer Science Applications and TechnologiesAt: Kuala Lumpur, MalaysiaVolume: ACSAT'12

4.2 Ramadhan Mstafa, Christian Bach "Information Hiding in Images using Steganography Techniques" 2013 ASEE Northeast Section Conference

4.3 Nagham Hamid, Abid Yahya, R. Badlishah Ahmad, Osamah M. Al-Qershi "Image Steganography Techniques: An Overview" International Journal of Computer Science and Security (IJCSS), Volume (6): Issue (3): 2012

4.4 Arun Kumar Singh, Juhi Singh, Dr. Harsh Vikram Singh "Steganography in Images Using LSB Technique " International Journal of Latest Trends in Engineering and Technology (IJLTET)

4.5 https://www.geeksforgeeks.org/lsb-based-image-steganography-using-matlab/

4.6 https://towardsdatascience.com/hiding-data-in-an-image-image-steganography-using-python-e491b68b1372