Problem Statement

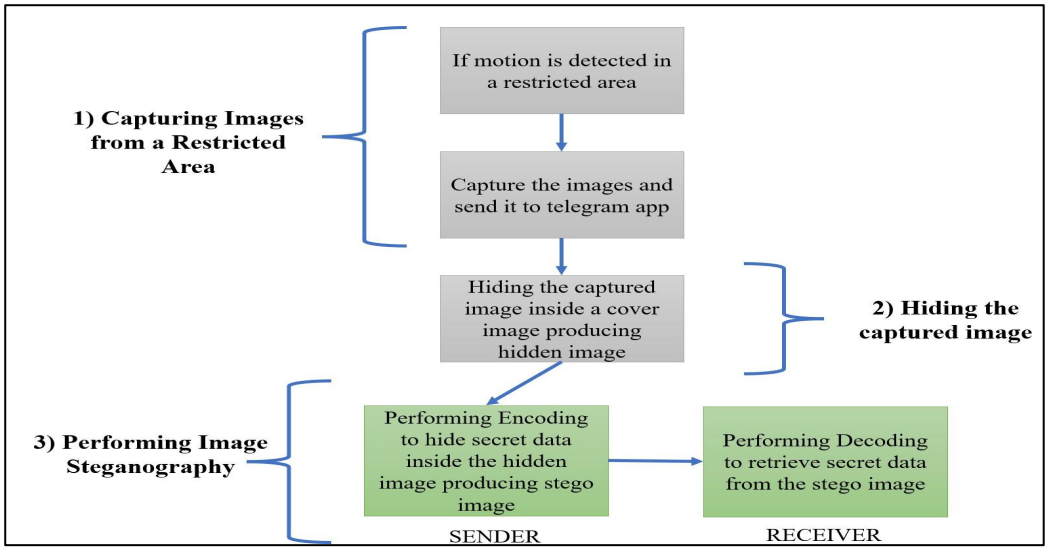
To develop a secure system to transmit the data between the sender and the receiver based on the images captured in an IOT environment by using the approach of Image Steganography.

Objective

To implement this project the idea of Image Steganography will be used. The objective of this project is to hide a secret message within a image that is captured from an IOT sensor in such a way that others cannot discern the presence of the hidden message using LSBS (Least Significant Bit Substitution) technique which is mainly concerned to be used in the field of IOT. The captured image will be hidden behind a cover image in order to preserve its privacy and security. The resulting image can be used to transmit the secret data from sender to the receiver. LSBS technique involves overwriting or substituting the bit with the lowest arithmetic value. The result of this technique very slightly alters the original input cover image.

Internet of Things (IoT) is a domain in which the transfer of data is taking place every single second, the security of these data is a challenging task. Connecting to the internet is so easy for everyone and so is the amount of data transferred and stored using the internet, resulting in keeping it safe from wrong hands and maintaining privacy. This project can be used to accomplish the above requirement concerned in the field of IOT. Apart from being used in the field of IOT this project can effectively contribute in various fields like banking, military applications, Ecommerce, etc.

Approach towards the problem



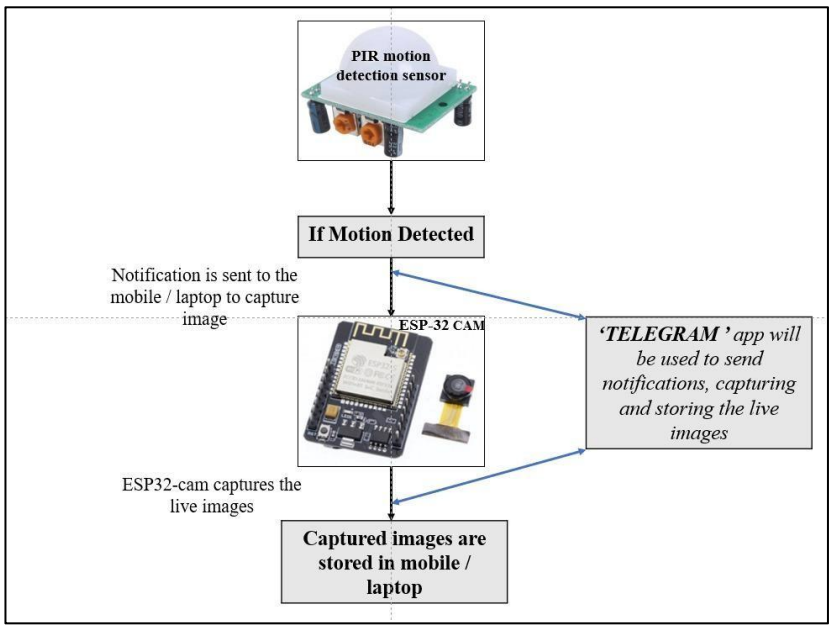
Software Requirements

- Python IDLE
- Arduino IDE
- Telegram App

Hardware Requirements

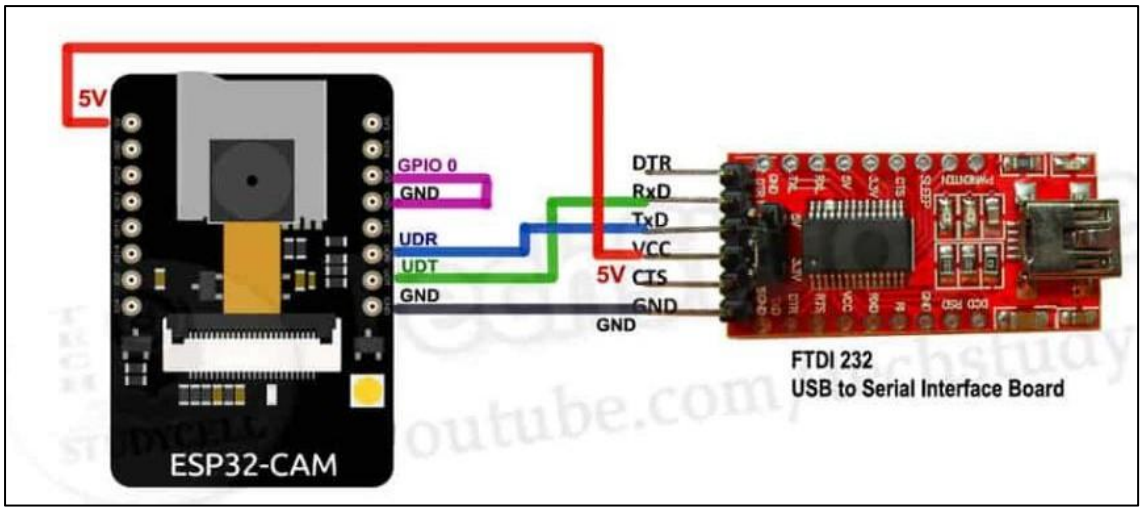
- ESP32-CAM
- PIR Motion Detection Sensor
- BC547 NPN Transistor
- 220 ohm, 1k, 10k Resistor
- LED
- FTDI 232 USB to Serial Interface board
- 5v DC supply circuits.
- Breadboard
- Jumper cables

Architectural Design



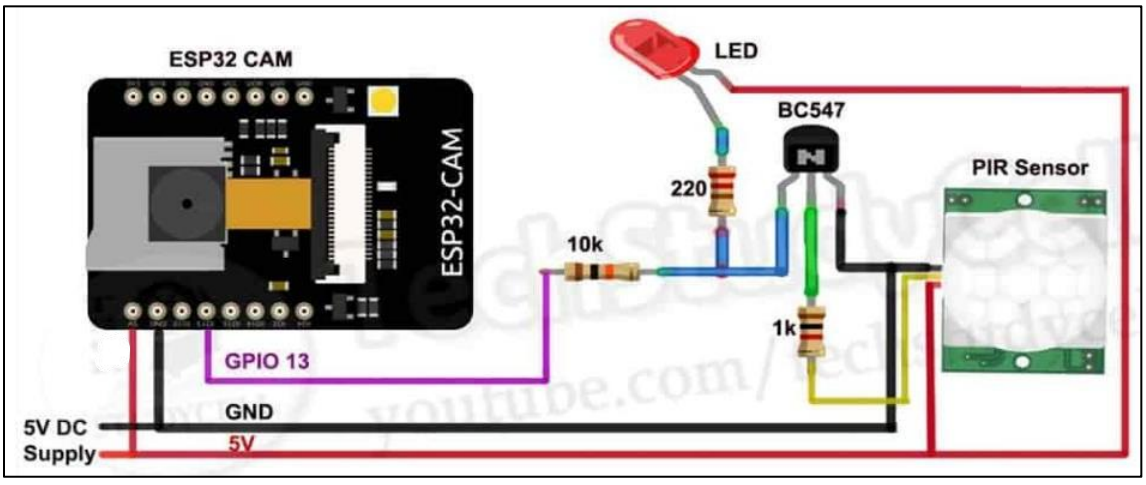
Capturing and storing images of intruder

The above image shows the system architecture diagram with different components of the proposed system. The PIR motion detector is used for detecting any motion within its range. If any motion is detected due to intrusion a signal is sent to the ESP-32 Camera to capture the image of the intruder. The ESP-32 camera captures the live images of the intruder and the captured image will be stored in the Telegram app.



Circuit diagram to program ESP32 CAM

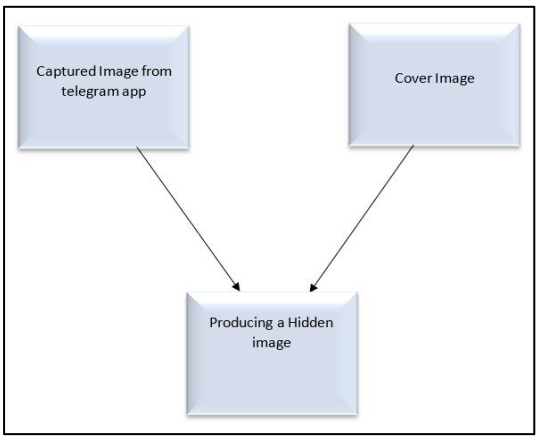
The above image shows the circuit diagram to program the ESP32CAM by using an FTDI232 USB to Serial interface board. The two components must be connected as shown and the other end of FTDI232 board must be connected to the laptop by using a USB cable to upload the code to the ESP32CAM.



Circuit diagram to capture live images

The above image shows the circuit diagram which is followed in order to capture live images from a restricted area after programming the ESP32 CAM as shown in Fig 4.2. On

providing the power supply to the above circuit, the required results are achieved.



Embedding the captured image inside a cover image.

The above image shows the embedding architecture design. The captured image saved in the telegram app is embedded using a cover image to produce an hidden image

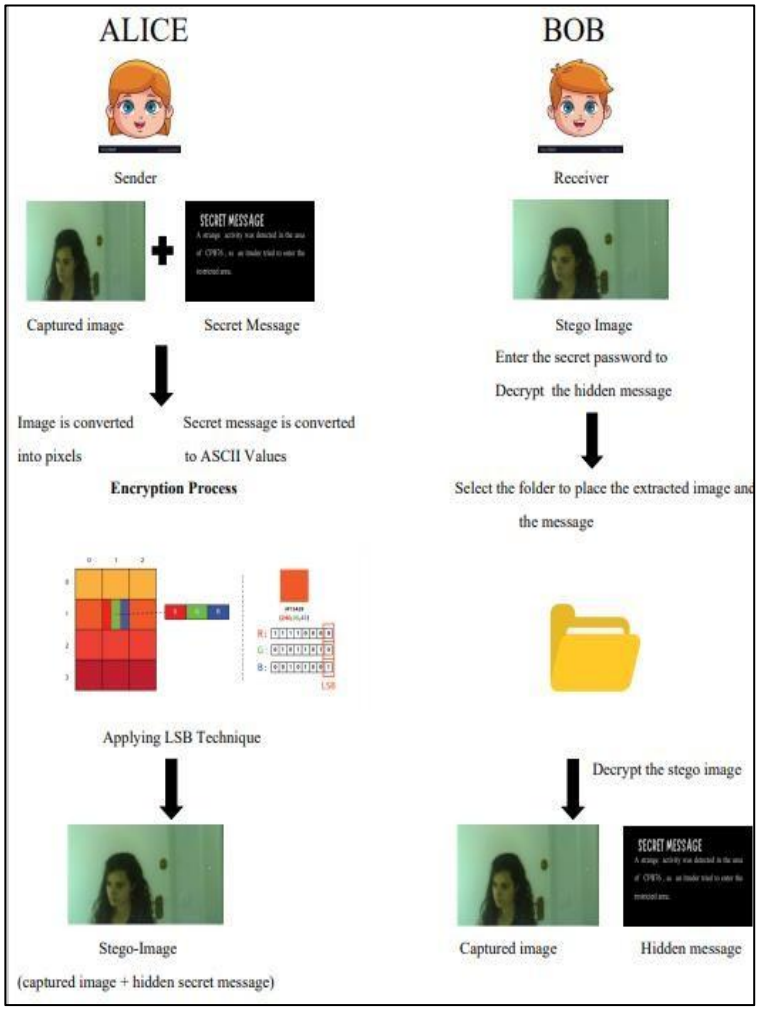


Image-steganography using LSB algorithm

The above image illustrates the process of performing LSB algorithm both from sender and receiver side.

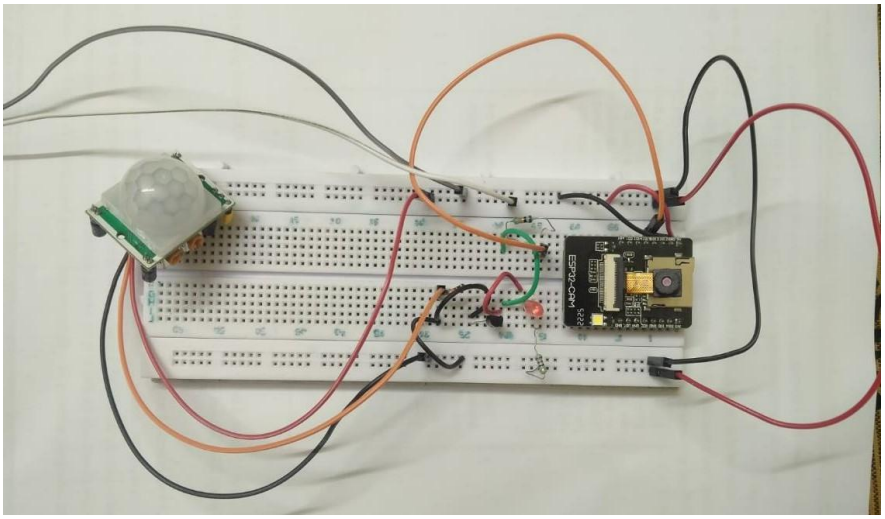
Sender Side

- Before the start of communication both the sender and receiver must share a secret password.
- The sender uploads hidden image and also enters the secret message which has to to hidden.
- The secret message to be hidden or transmitted is taken and converted into suitable ASCII values.
- Then the hidden image is converted into pixels and each pixel is represented in 8 bits.
- LSB technique is applied in order to alter the bits of the hidden image so that the secret message can be embedded inside it.
- The resulting image is called stego-image and is stored in the Telegram app and is sent to the receiver.

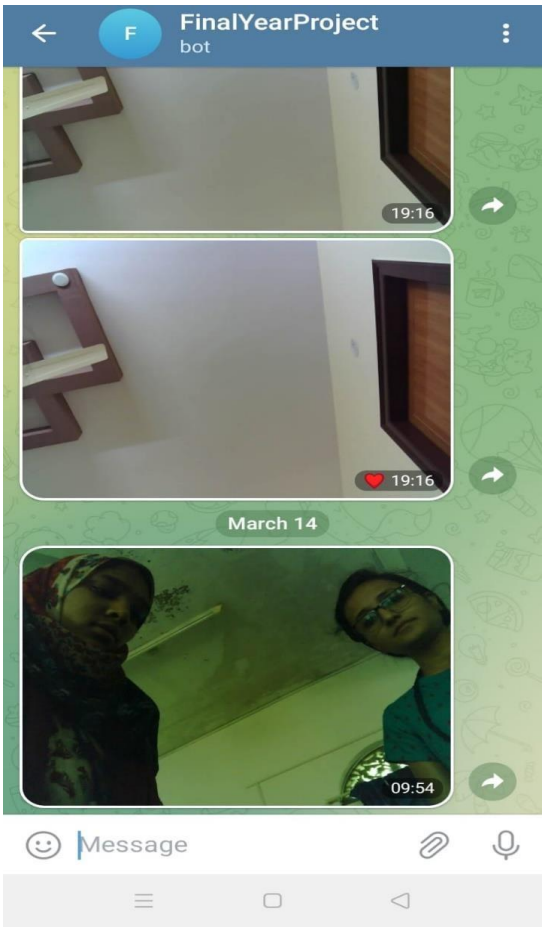
Receiver Side

- At the receiver side, Bob receives the stego-image.
- Bob has to enter the secret password to decrypt the hidden message.
- Bob selects the folder where he wants to store the extracted information.
- Upon decrypting the stego-image, the secret message will be retrieved.

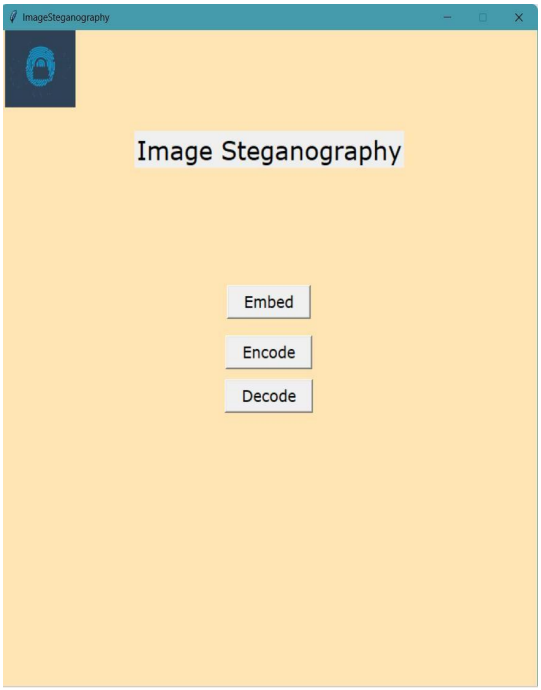
Results



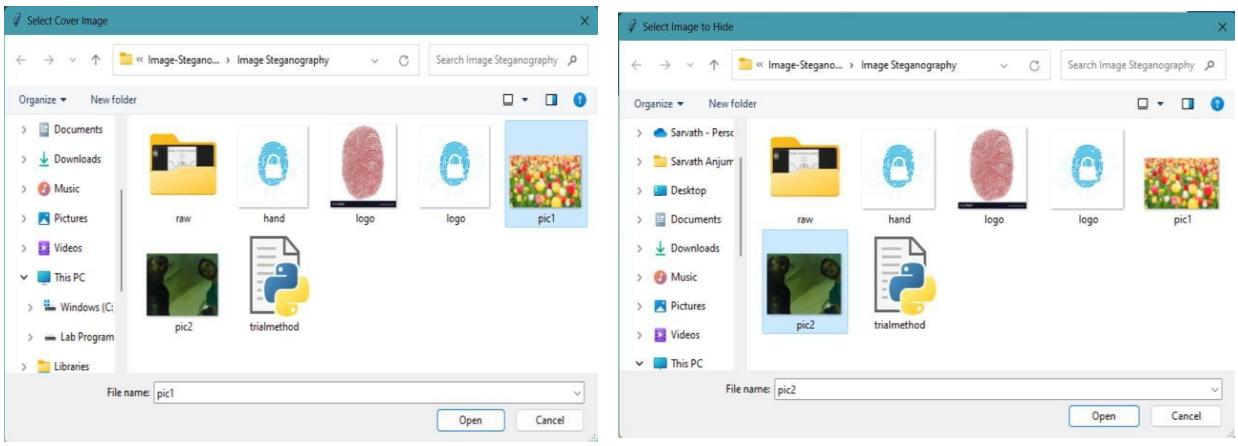
Hardware Setup



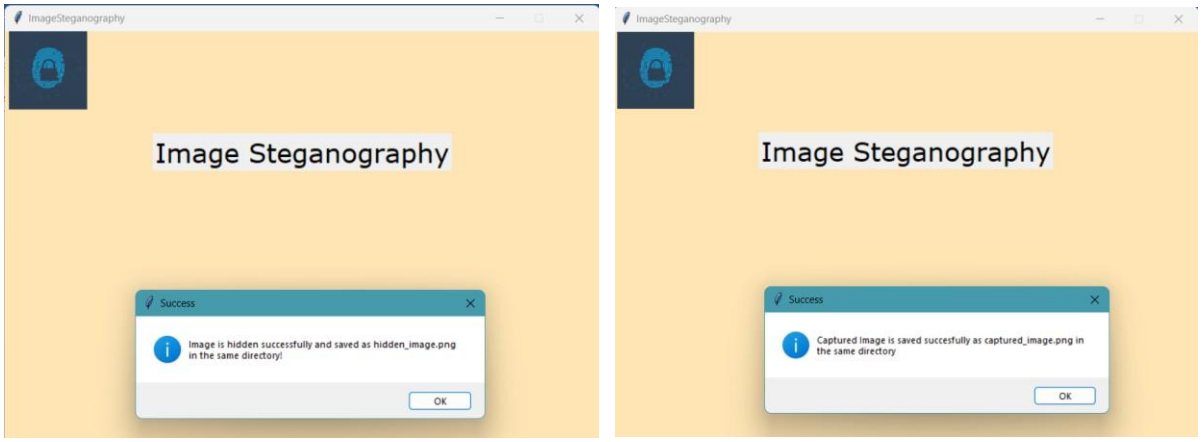
Captured Images stored in telegram



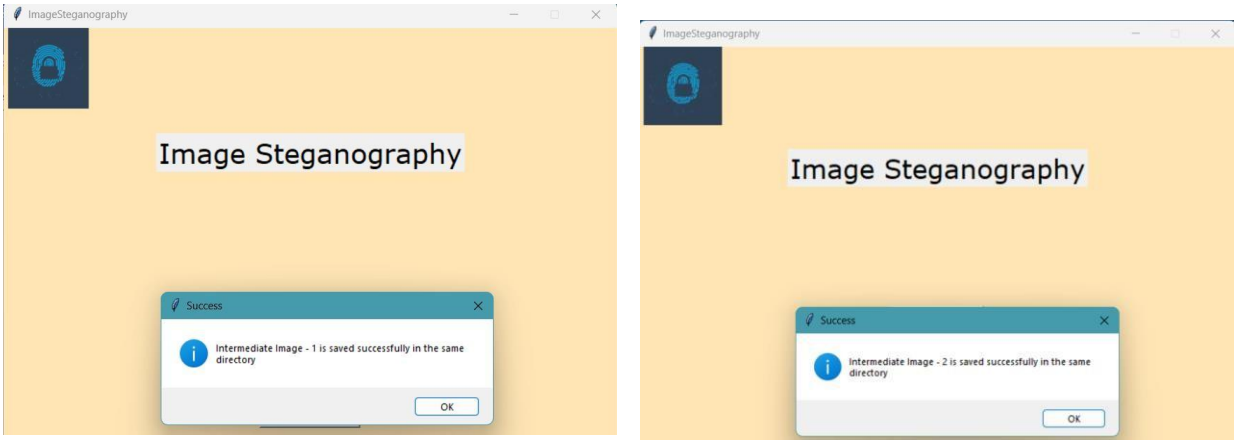
Home Page



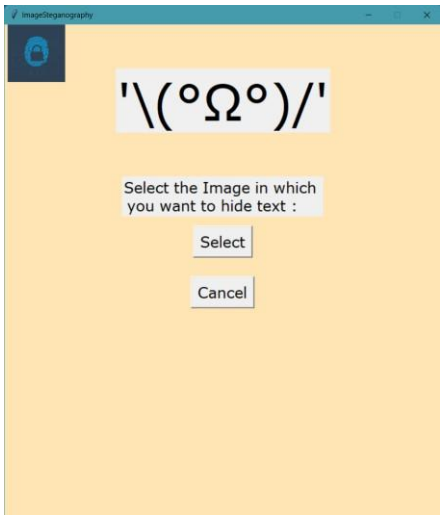
Selecting cover image and captured image



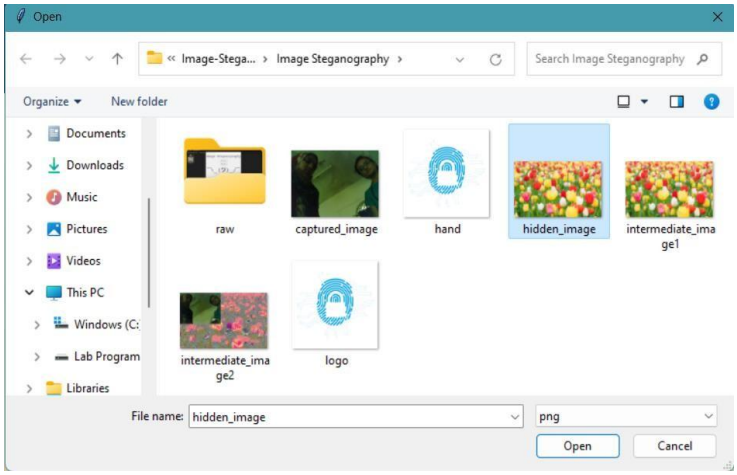
Saving hidden and captured image



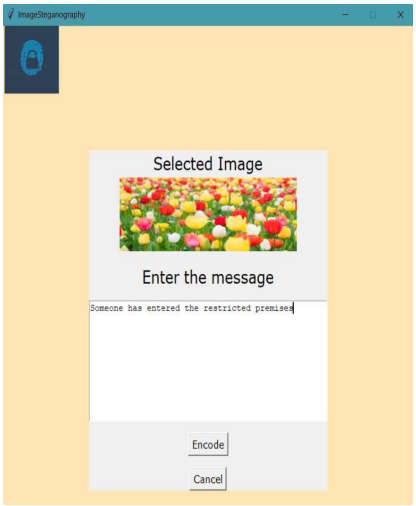
Saving intermediate images



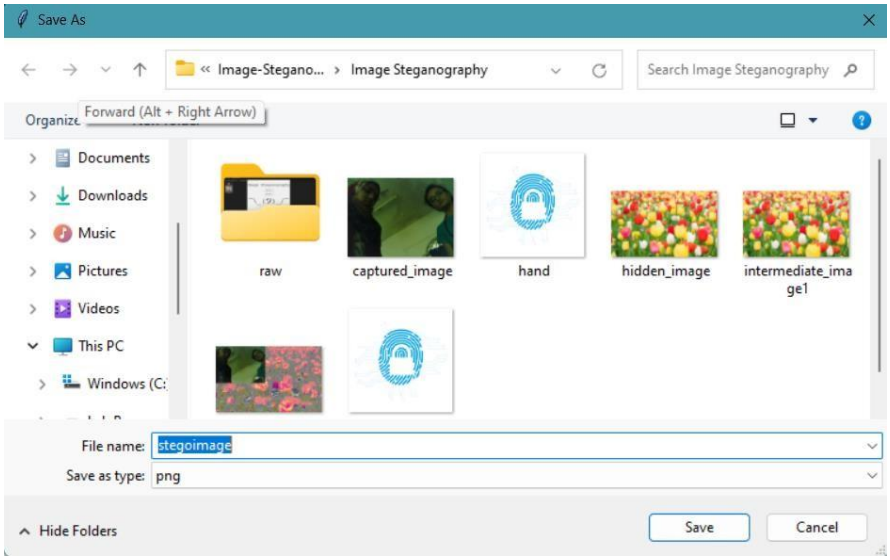
Encoding Process Page



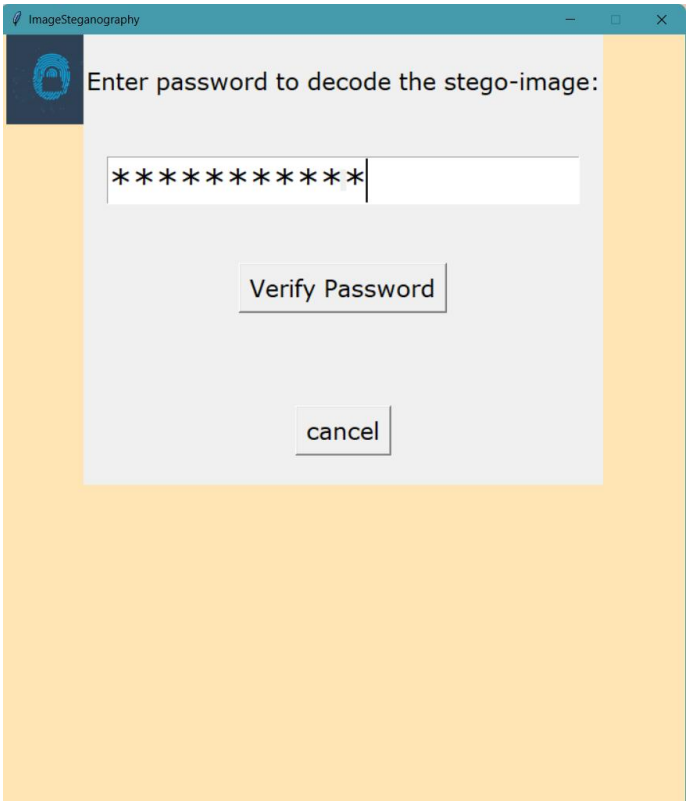
Selecting hidden image



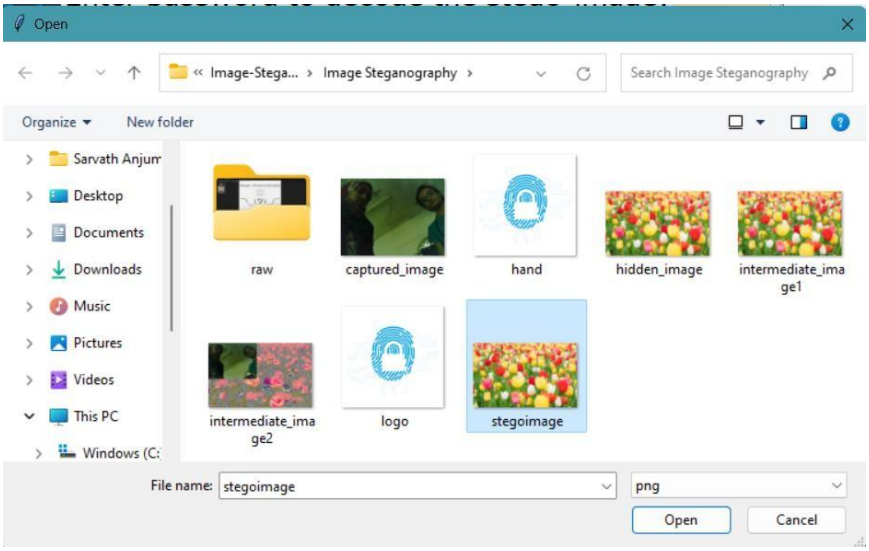
Entering the secret mess



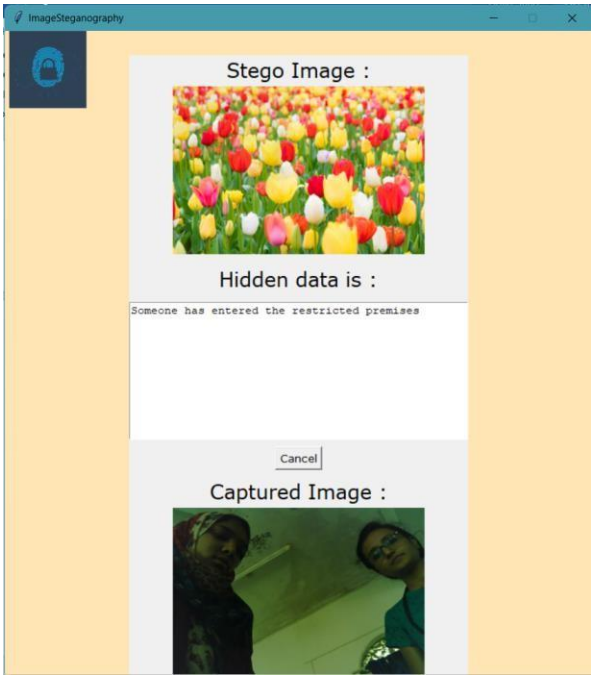
Saving the Stego Image



Password Verification



Selecting the Stego Image



Secret message and captured image is extracted