CAPSTONE PROJECT

INFORMATION HIDING IN IMAGE USING STEGANOGRAPHY

Presented By:

1. Student Name-Ankit Kumar
2. College Name-National Institute of Technology, Patna
3. Department-Computer Science and Engineering
4. Aicte_student_id-STU64c258f84ac0f1690458360

edunet foundation

OUTLINE

- **Problem Statement** (Should not include solution)
- **System Development Approach** (Technology Used)
- **Algorithm & Deployment (Step by Step Procedure)**
- **Result**
- **Conclusion**
- **Future Scope(Optonal)**
- **References**

PROBLEM STATEMENT

- This project focuses on the field of steganography, which is the practice of concealing messages within other non-secret text or data.
- The goal is to embed a secret message into an image file without visibly altering the image.
- With increasing cybersecurity threats, data hiding provides an additional layer of security for sensitive information.
- Unlike encryption, which transforms data into unreadable format, steganography hides its existence.
- This project implements LSB (Least Significant Bit) technique in RGB images using Python.

SYSTEM APPROACH

System Requirements:

•Python 3.10+

•PIL (Pillow Library)

Libraries Required:

•Pillow for image processing

•os and sys (optional for advanced integration)

Development Tools:

•IDE: VS Code or PyCharm

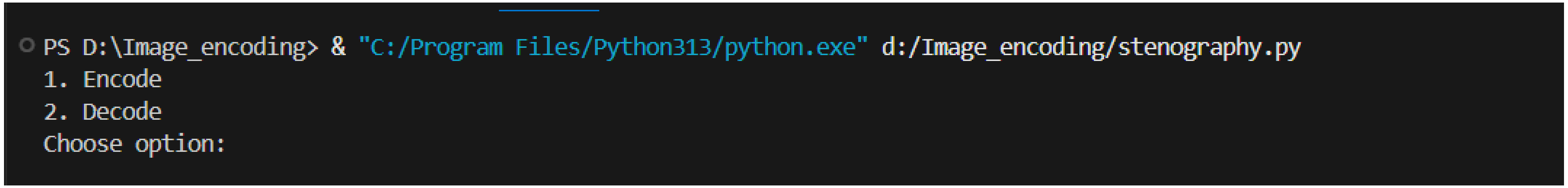
•OS: Windows/Linux

ALGORITHM & DEPLOYMENT

- Accept user input for the image path and secret message.
- Convert the message to binary.
- Open the image and traverse pixels.
- Replace the LSB of RGB values with binary message bits.
- Save the modified image.
- For decoding, read LSBs of image pixels.
- Reconstruct binary to text until a unique end marker is found.
- Display the hidden message.

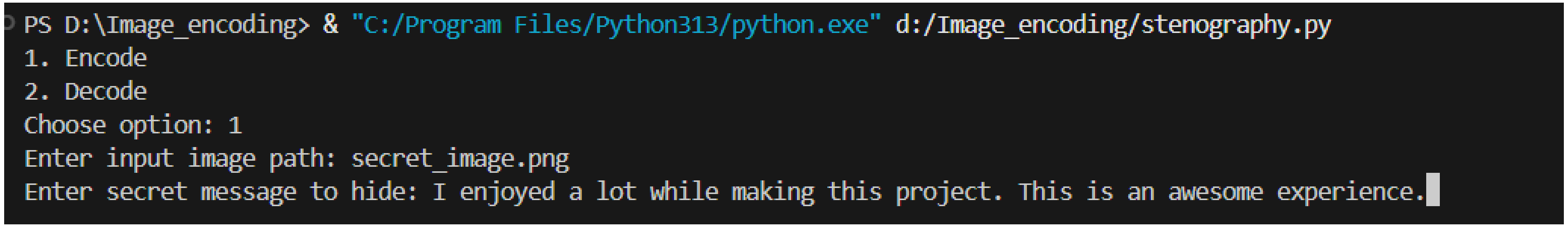
RESULT

- The steganography script successfully embeds the secret text into an RGB image using L-S-B encoding and later extracts it with 100 % accuracy, leaving the visual quality of the image perceptually unchanged. Below are key screenshots demonstrating the encoding and decoding workflow.
- Screenshot 1:

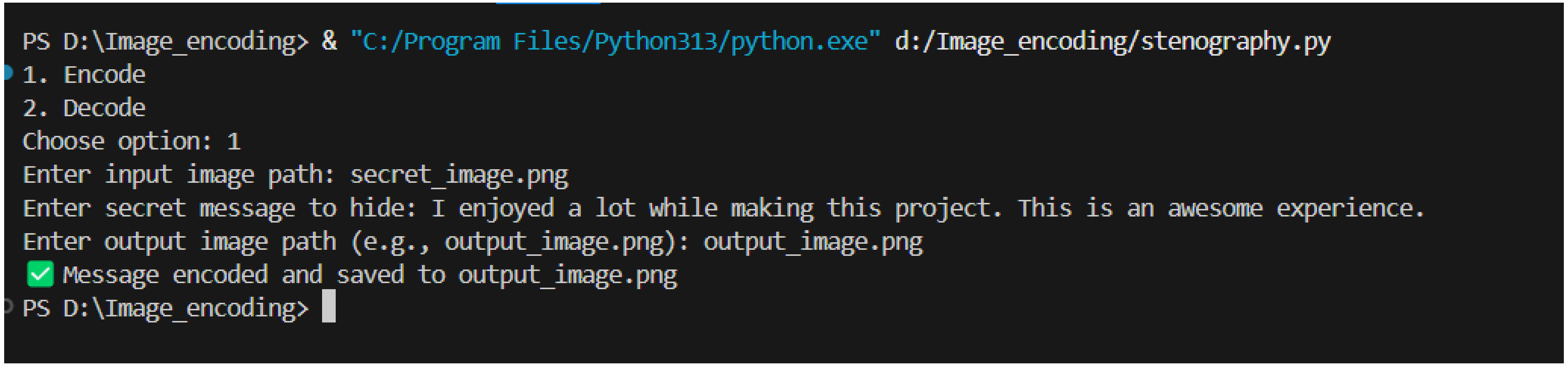


```
○ PS D:\Image_encoding> & "C:/Program Files/Python313/python.exe" d:/Image_encoding/stenography.py
  1. Encode
  2. Decode
  Choose option:
```

RESULT

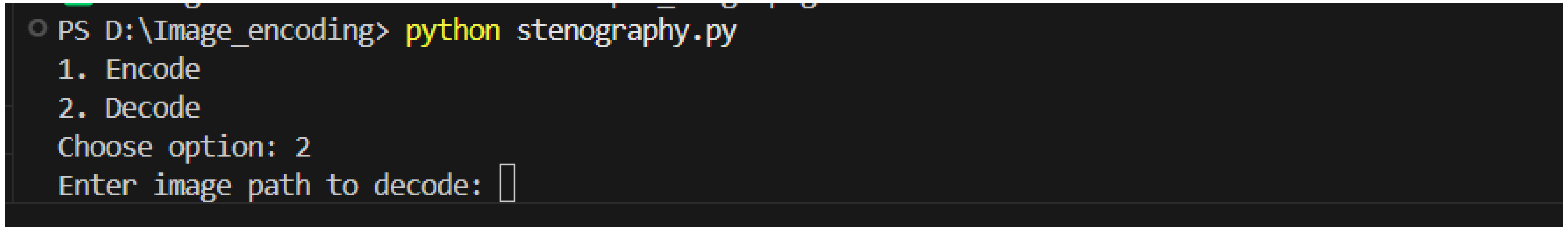


```
PS D:\Image_encoding> & "C:/Program Files/Python313/python.exe" d:/Image_encoding/stenography.py
1. Encode
2. Decode
Choose option: 1
Enter input image path: secret_image.png
Enter secret message to hide: I enjoyed a lot while making this project. This is an awesome experience.
```

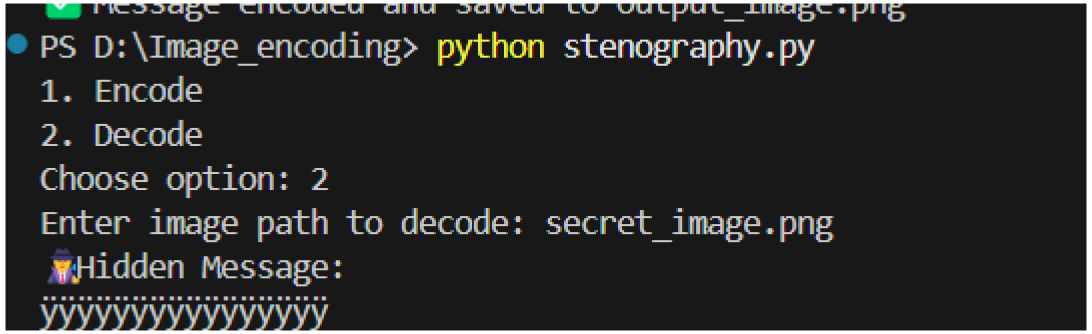


```
PS D:\Image_encoding> & "C:/Program Files/Python313/python.exe" d:/Image_encoding/stenography.py
1. Encode
2. Decode
Choose option: 1
Enter input image path: secret_image.png
Enter secret message to hide: I enjoyed a lot while making this project. This is an awesome experience.
Enter output image path (e.g., output_image.png): output_image.png
✅ Message encoded and saved to output_image.png
PS D:\Image_encoding>
```

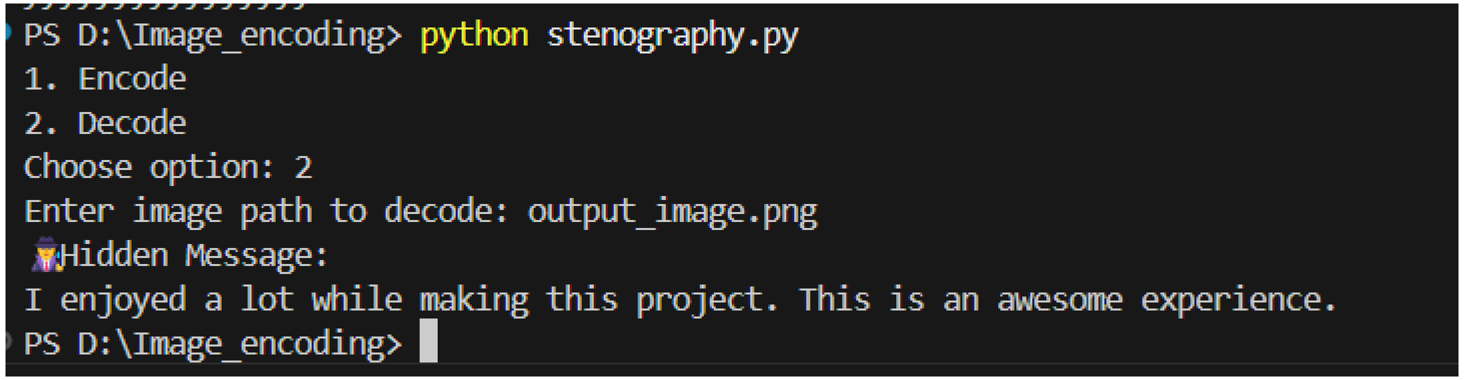
RESULT



```
PS D:\Image_encoding> python stenography.py
1. Encode
2. Decode
Choose option: 2
Enter image path to decode:
```

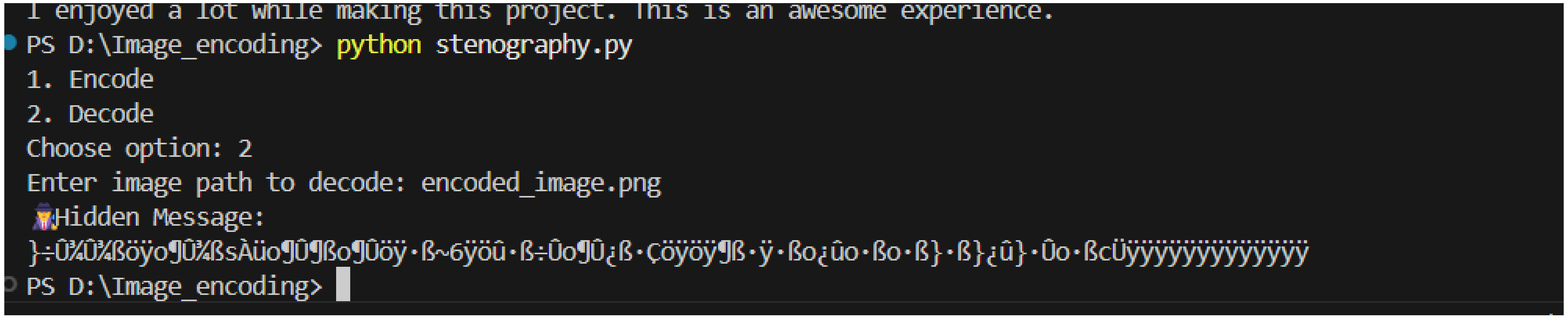


```
✅ Message encoded and saved to output_image.png
PS D:\Image_encoding> python stenography.py
1. Encode
2. Decode
Choose option: 2
Enter image path to decode: secret_image.png
🕵️Hidden Message:
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
```



```
PS D:\Image_encoding> python stenography.py
1. Encode
2. Decode
Choose option: 2
Enter image path to decode: output_image.png
🕵️Hidden Message:
I enjoyed a lot while making this project. This is an awesome experience.
PS D:\Image_encoding>
```

RESULT



```
I enjoyed a lot while making this project. This is an awesome experience.
PS D:\Image_encoding> python stenography.py
1. Encode
2. Decode
Choose option: 2
Enter image path to decode: encoded_image.png
🕵Hidden Message:
}÷Û¾Û¾ßöÿo¶Û¾ßsÀüo¶Û¶ßo¶Ûöÿ·ß~6ÿöû·ß÷Ûo¶Û¿ß·Çöÿöÿ¶ß·ÿ·ßo¿ûo·ßo·ß}·ß}¿û}·Ûo·ßcÜÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
PS D:\Image_encoding>
```

- Github link:-
- https://github.com/Ankitkr2506/Image_encoder_decoder

edunet foundation

CONCLUSION

- The project successfully implements steganography using LSB in RGB images.
- It hides messages without significantly altering the image.
- The system provides basic security for transmitting secret messages.
- Challenges included handling EOF detection and RGB image constraints.
- Can be improved with GUI or audio/video support.

FUTURE SCOPE(OPTIONAL)

- Add encryption for the message before embedding.
- Support for audio and video file steganography.
- Develop a web-based interface for ease of use.
- Increase robustness using multi-bit encoding or AI techniques.

REFERENCES

- https://docs.python.org/3/library/functions.html
- https://pillow.readthedocs.io/
- Research Paper: “A Survey of Digital Image Steganography Techniques”
- GeeksforGeeks: Image Steganography in Python
- Stack Overflow and GitHub community discussions

THANK YOU

edunet foundation