CPSC-6040 Computer Graphics Images
Final Project – Image Steganography

by Shashi Shivaraju

Steganography is the practice of concealing a file, message, image, or video within another file, message, image, or video.

As per final project proposal,I have implemented a steganography tool that allows a user to hide an image inside another one, as well as extract a hidden image. This tool has two modes of steganography as described below:

Basic Mode: Default Steganography Mode

- As a basic requirement,I implemented the steganography functionality using the pixel manipulation technique of changing the less significant bits from an image (cover image) by including the most significant bits from the other image (secret image to hide).

- I have used 3 less significant bits of the cover image pixels to hide the 3 most significant bits of the secret image.

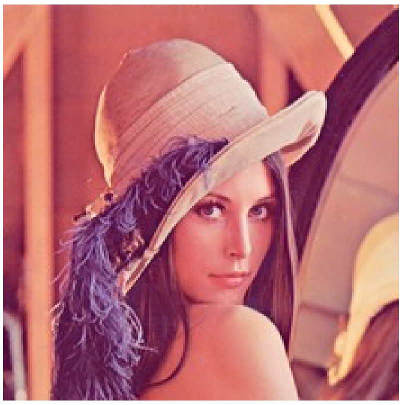
- In this mode there is no restriction on the size/dimensions of the cover image and the secret image.

- The results of the this mode are shown below:

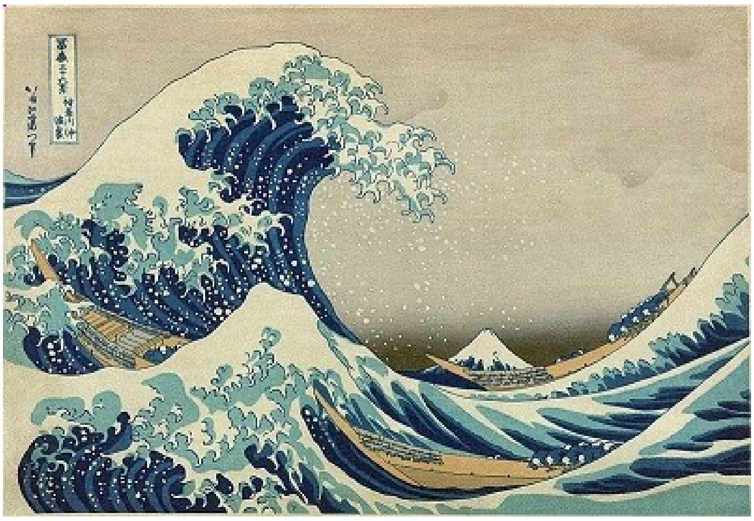
Example 1: Secret Image dimension is lesser than on equal to Cover Image dimension.

Cover Image: waves.png [640x480]

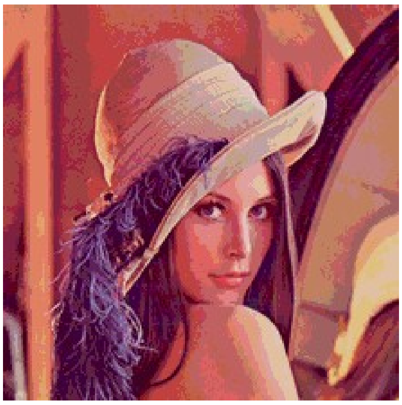
Secret Image: Lenna.png [220 x 220]



Merged Image: Hidden.png [640x480]

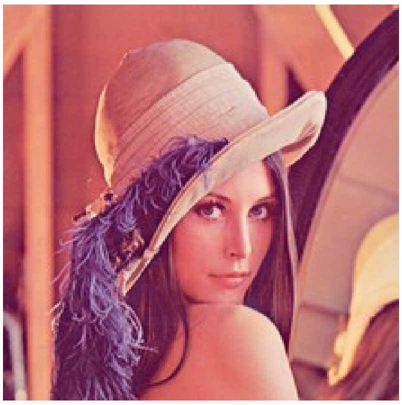


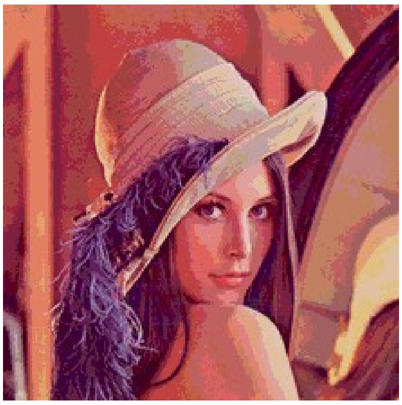
Extracted Image: Secret.png [220 x 220]



- **Extracted image dimension same as secret image dimension.**
- **Noticeable image quality degradation in extracted image .**

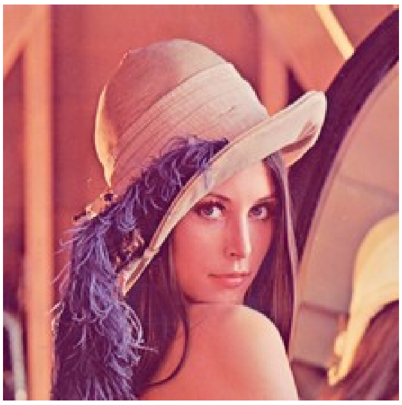
Secret Image: Lenna.png [220 x 220] vs Extracted Image: Secret.png [220 x 220]



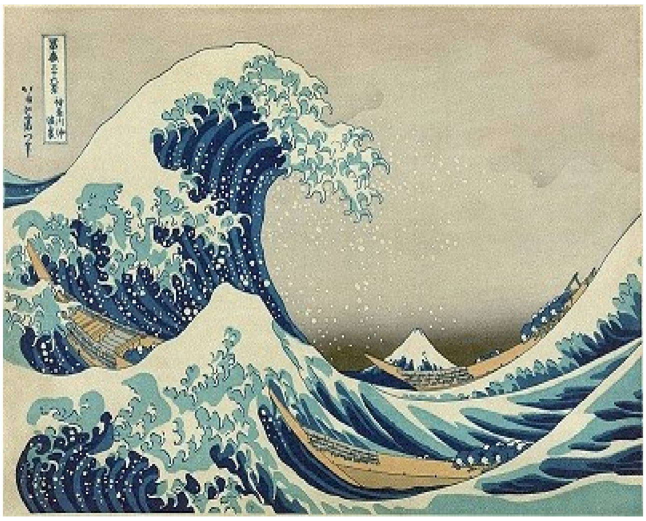


Example 2: Secret Image dimension is greater than on equal to Cover Image dimension.

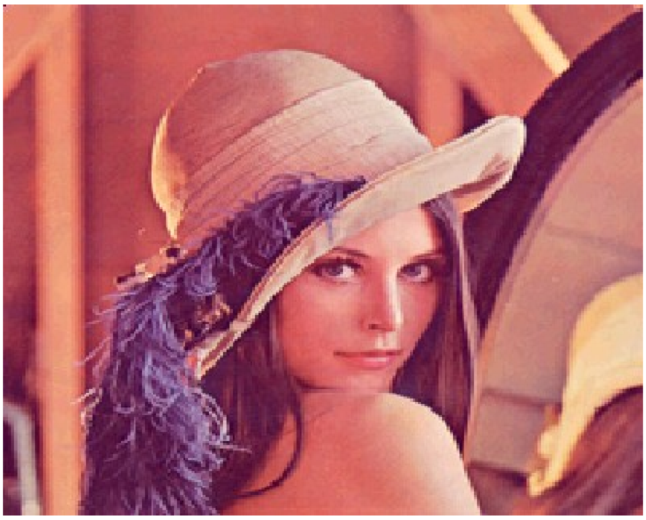
Cover Image: Lenna.png [220 x 220]



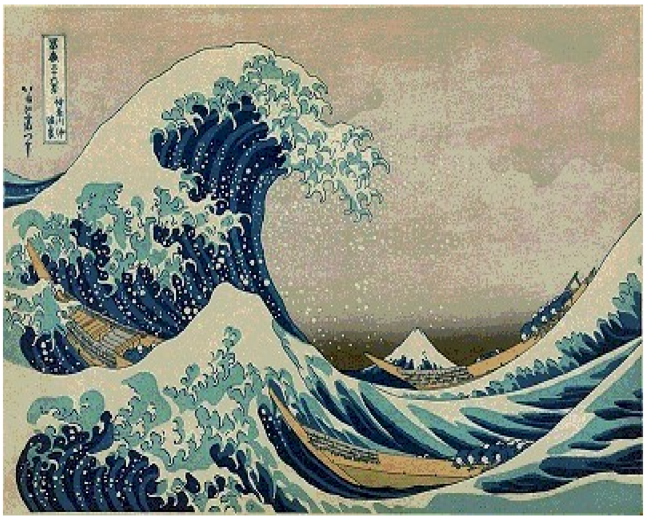
Secret Image: waves.png [640x480]



Merged Image: Hidden.png [640x480]

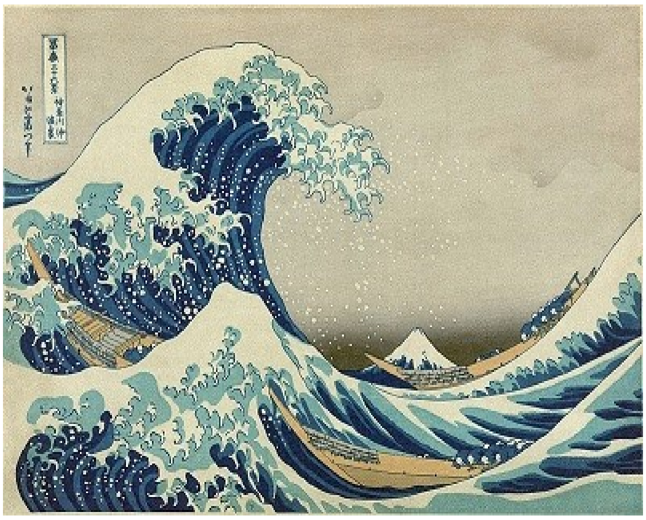


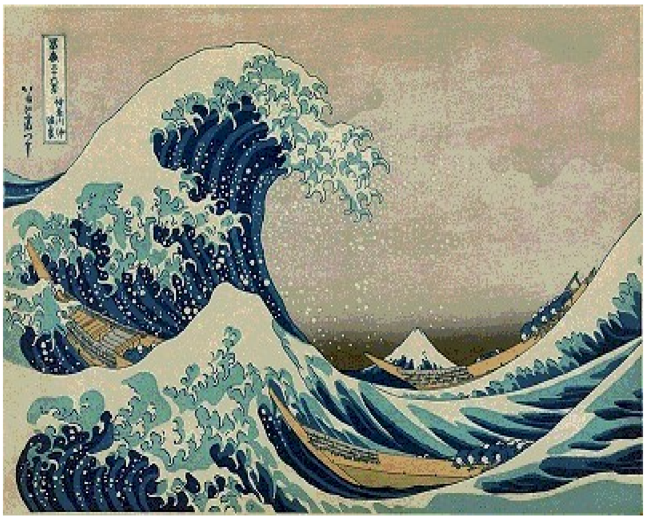
Extracted Image: Secret.png [640x480]



- **Cover image is scaled to match the secret image dimension to create Merged Image.**
- **Extracted image dimension same as secret image dimension.**
- **Noticeable image quality degradation in extracted image .**

Secret Image: waves.png [640x480] vs Extracted Image: Secret.png [640x480]





Concepts used to implement the Default Steganography Mode :

- Pixel Manipulation.
- Image Warping for scaling.

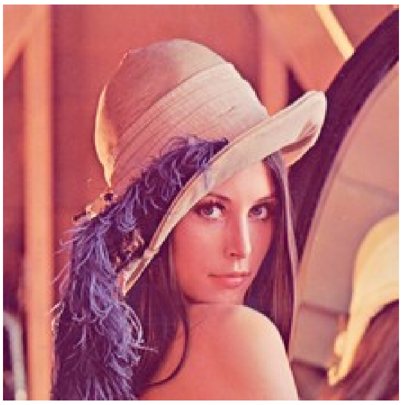
Advance Mode: Compression Based Steganography Mode

- The results from Default Steganography Mode clearly demonstrates that in the there is degradation of image quality during image extraction.
- To reduce or prevent this degradation in the image quality of the secret image,I implemented a lossless compression algorithm called Huffman Algorithm.
- Huffman Algorithm utilizes priority queues and binary tree to achieve lossless compression.
- In Compression Based Steganography Mode,the secret image is encoded using Huffman Algorithm and the resulting encoded data is then concealed in the cover image.
- During the image extraction process,the encoded data is retrieved from the cover image and decoded to retrieve the concealed image without any degradation.
- In this mode,each pixel of the cover image will be used to conceal one byte of the encoded data of the secret image.
- The results of the this mode are shown below:

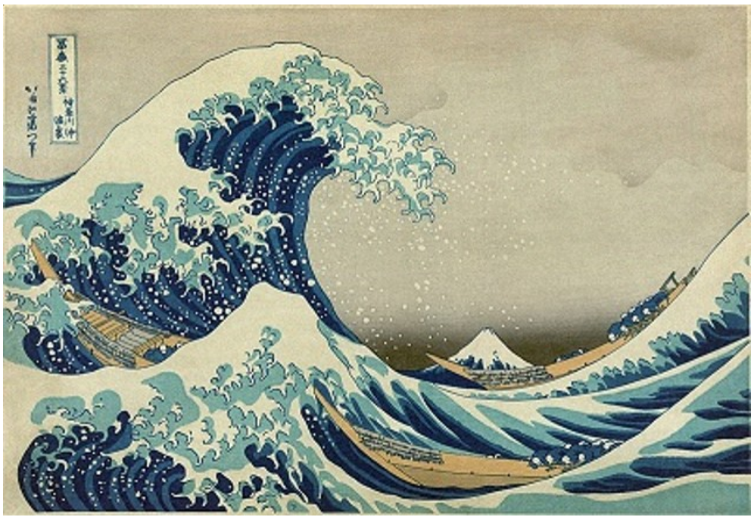
Example 3: Compression Based Steganography

Cover Image: waves.png [640x480]

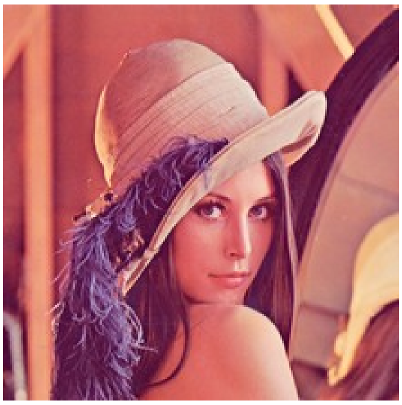
Secret Image: Lenna.png [220 x 220]



Merged Image: Hidden.png [640x480]

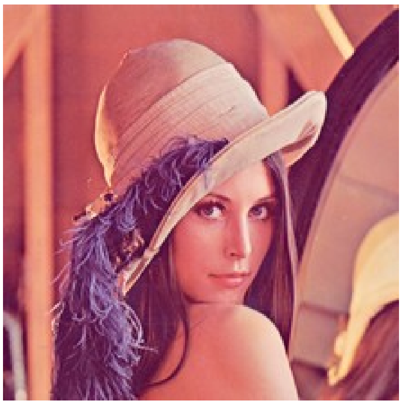


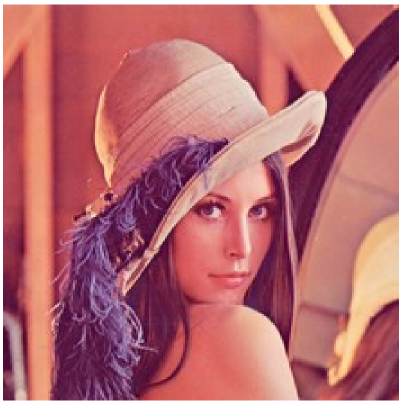
Extracted Image: Secret.png [220 x 220]



- Extracted image dimension same as secret image dimension.
- Extracted image quality same as secret image quality.

Secret Image: Lenna.png [220 x 220] vs Extracted Image: Secret.png [220 x 220]



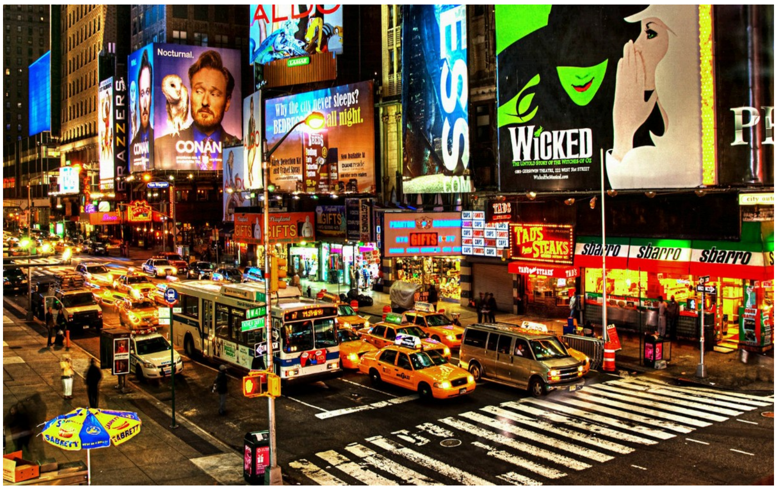


Example 4: Steganography with File Formats

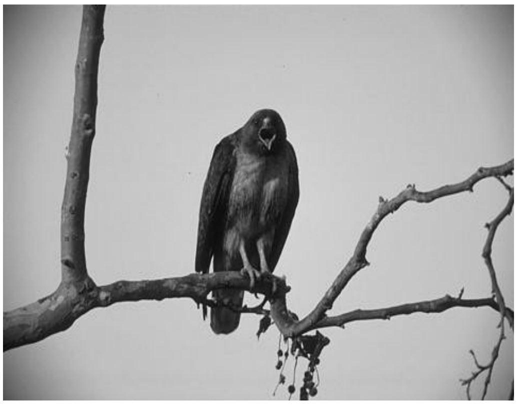
Cover Image: Cover.jpg [1280x720]

Secret Image: hawk.ppm [481 x 321]

Merged Image: Hidden.png [1280x720]



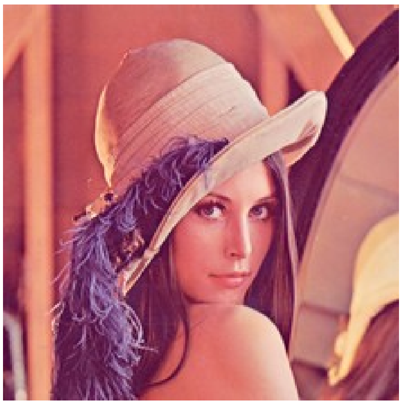
Extracted Image: Secret.jpg [481x321]



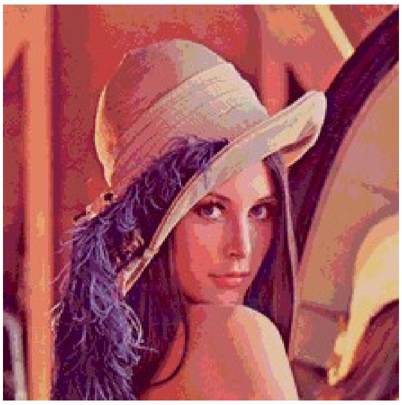
- **Extracted image dimension same as secret image dimension.**
- **Extracted image quality same as secret image quality.**
- **Cover Image,Secret Image can be used and the Extracted images can be stored as files respectively,which belong to any file format supported by OpenImageIO.**
- **The Merged image should always be should always be stored as lossless image format such as png/bmp.**

Secret Image: hawk.ppm [481 x 321] vs Extracted Image: Secret.jpg [481 x 321]

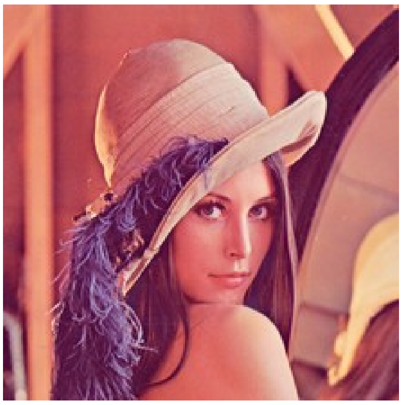
Original Image Quality vs Steganography Image Quality



Secret Image



Default Mode Extracted Image



Compression Mode Extracted Image

Concepts used to implement the Compression based Steganography Mode :

- Pixel Manipulation.
- Lossless Data Compression: Huffman Codec

Issue:

- The degree of compression varies among images as compression depends on the image content and size. If the cover image is not able to accommodate the size of the compressed secret image during hiding operation (i.e. one byte of encoded data in one pixel of cover image),than the program will display an console message informing the inability to perform steganography operation.

Conclusion and Future Work:

- The implemented steganography tool worked as expected when used in Default Mode by producing results with image quality degradation. This mode is generic in its performance due to cover image scaling functionality.
- The Compression base Steganography Mode was successful in preserving the image quality but is not an generic approach due to the variation in the level of image compression achieved using Huffman Codec.
- Future work will involve implementation of better compression algorithm and intelligence to make the Compression base Steganography Mode generic in its performance.