# Assignment 2 – (Question No 3)

Identifying the total number of objects that appeared in an image before and after applying the steganography techniques. Steganograpgy is the practice of concealing a file, message, image, or video within another file, message, image, or video. The advantage of steganography over cryptography alone is that the intended secret message does not attract attention to itself as an object of scrutiny. Here what you are supposed to do is, identify the number of objects in original images (in both cover and hide images) then apply the steganography technique. Then by considering the output result (combined image), again you have to rerun the same algorithm to detect the number of objects in it. Finally you have to recover the image.

## 1)Inputs and out puts

Cover Image

Image to hide

Combined Image

Recover Image

## 2)No of objects in the Cover image and Image to hide image before Steganograpgy

Cover Image

```
# Import libraries
import cv2
import numpy as np
import matplotlib.pyplot as plt

image = cv2.imread('img1_wind.jpg')
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

blur = cv2.GaussianBlur(gray, (11, 11), 0)
canny = cv2.Canny(blur, 30, 150, 3)
dilated = cv2.dilate(canny, (1, 1), iterations=0)

(cnt, hierarchy) = cv2.findContours(
    dilated.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
cv2.drawContours(rgb, cnt, -1, (0, 255, 0), 2)


print("No of objects in the pic1 - cover image : ", len(cnt))
```

```
No of objects in the pic1 - cover image :  12
```

Image

Image to hide Image

```
# Import libraries
import cv2
import numpy as np
import matplotlib.pyplot as plt

image = cv2.imread('img2_plant.jpg')
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

blur = cv2.GaussianBlur(gray, (11, 11), 0)
canny = cv2.Canny(blur, 30, 150, 3)
dilated = cv2.dilate(canny, (1, 1), iterations=0)

(cnt, hierarchy) = cv2.findContours(
    dilated.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
cv2.drawContours(rgb, cnt, -1, (0, 255, 0), 2)


print("No of objects in the pic2 - Image to hide : ", len(cnt))
```

```
No of objects in the pic2 - Image to hide :  14
```

Image to hide

## 3)Steganography technique using OpenCV in python

```
import cv2
import numpy as np
import random

# Encryption function
def encrypt():

    # img1 and img2 are the
    # two input images
    img1 = cv2.imread('img1_wind.jpg')
    img2 = cv2.imread('img2_plant.jpg')

    for i in range(img2.shape[0]):
        for j in range(img2.shape[1]):
            for l in range(3):
                # v1 and v2 are 8-bit pixel values
                # of img1 and img2 respectively
                v1 = format(img1[i][j][l], '08b')
                v2 = format(img2[i][j][l], '08b')

                # Taking 4 MSBs of each image
                v3 = v1[:4] + v2[:4]

                img1[i][j][l]= int(v3, 2)

    cv2.imwrite('pic3in2.png', img1)

# Decryption function
def decrypt():
```

```
    # Encrypted image
    img = cv2.imread('pic3in2.png')
    width = img.shape[0]
    height = img.shape[1]

    # img1 and img2 are two blank images
    img1 = np.zeros((width, height, 3), np.uint8)
    img2 = np.zeros((width, height, 3), np.uint8)

    for i in range(width):
        for j in range(height):
            for l in range(3):
                v1 = format(img[i][j][l], '08b')
                v2 = v1[:4] + chr(random.randint(0, 1)+48) * 4
                v3 = v1[4:] + chr(random.randint(0, 1)+48) * 4

                # Appending data to img1 and img2
                img1[i][j][l]= int(v2, 2)
                img2[i][j][l]= int(v3, 2)

    # These are two images produced from
    # the encrypted image
    cv2.imwrite('pic2_re.png', img1)
    cv2.imwrite('pic3_re.png', img2)


# Driver's code
encrypt()
decrypt()
```

## 4)No of objects in the combined image after using steganography technique

```
# Import libraries
import cv2
import numpy as np
import matplotlib.pyplot as plt

image = cv2.imread('pic3in2.png')
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

blur = cv2.GaussianBlur(gray, (11, 11), 0)
canny = cv2.Canny(blur, 30, 150, 3)
dilated = cv2.dilate(canny, (1, 1), iterations=0)

(cnt, hierarchy) = cv2.findContours(
    dilated.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
cv2.drawContours(rgb, cnt, -1, (0, 255, 0), 2)


print("No of objects in the encypted image : ", len(cnt))
```

```
No of objects in the encypted image :  11
```

Image After the technique

## 5)No of objects in the recovered images after recovering

Cover Image

```
# Import libraries
import cv2
import numpy as np
import matplotlib.pyplot as plt

image = cv2.imread('pic2_re.png')
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

blur = cv2.GaussianBlur(gray, (11, 11), 0)
canny = cv2.Canny(blur, 30, 150, 3)
dilated = cv2.dilate(canny, (1, 1), iterations=0)

(cnt, hierarchy) = cv2.findContours(
    dilated.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
cv2.drawContours(rgb, cnt, -1, (0, 255, 0), 2)


print("No of objects of the pic1 after recovering - cover image : ", len(cnt))
```

```
No of objects of the pic1 after recovering - cover image :  9
```

Cover image after recovering

Image to hide image

```
# Import libraries
import cv2
import numpy as np
import matplotlib.pyplot as plt

image = cv2.imread('pic3_re.png')
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

blur = cv2.GaussianBlur(gray, (11, 11), 0)
canny = cv2.Canny(blur, 30, 150, 3)
dilated = cv2.dilate(canny, (1, 1), iterations=0)

(cnt, hierarchy) = cv2.findContours(
    dilated.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
cv2.drawContours(rgb, cnt, -1, (0, 255, 0), 2)


print("No of objects of the pic1 after recovering - Image to image : ", len(cnt))
```

```
No of objects of the pic1 after recovering - Image to image :  16
```

Hide image after recovering