**Source code for DIP Project**

**IMAGE STEGANOGRAPHY**

SHYAM SUNDAR K 20MIS1074

SHARON ROSABEL C 20MIS1114

VEENASREE 20MIS1139

VISHAL MURUGEASAN P 20MIS1146

**SOURCE CODE:**

# Python program implementing Image Steganography

# PIL module is used to extract

# pixels of image and modify it

from PIL import Image

# Convert encoding data into 8-bit binary

# form using ASCII value of characters

def gendata(data):

# list of binary codes

# of given data

newd = []

for i in data:

newd.append(format(ord(i), '08b'))

return newd

# Pixels are modified according to the

# 8-bit binary data and finally returned

def modpix(pix, data):

datalist = gendata(data)

lendata = len(datalist)

imdata = iter(pix)

for i in range(lendata):

# Extracting 3 pixels at a time

pix = [value for value in imdata.\_\_next\_\_()[:3] +

imdata.\_\_next\_\_()[:3] +

imdata.\_\_next\_\_()[:3]]

# Pixel value should be made

# odd for 1 and even for 0

for j in range(0, 8):

if datalist[i][j] == '0' and pix[j] % 2 != 0:

pix[j] -= 1

elif datalist[i][j] == '1' and pix[j] % 2 == 0:

if pix[j] != 0:

pix[j] -= 1

else:

pix[j] += 1

# pix[j] -= 1

# Eighth pixel of every set tells

# whether to stop ot read further.

# 0 means keep reading; 1 means thec

# message is over.

if i == lendata - 1:

if pix[-1] % 2 == 0:

if pix[-1] != 0:

pix[-1] -= 1

else:

pix[-1] += 1

else:

if pix[-1] % 2 != 0:

pix[-1] -= 1

pix = tuple(pix)

yield pix[0:3]

yield pix[3:6]

yield pix[6:9]

def encode\_enc(newimg, data):

w = newimg.size[0]

(x, y) = (0, 0)

for pixel in modpix(newimg.getdata(), data):

# Putting modified pixels in the new image

newimg.putpixel((x, y), pixel)

if x == w - 1:

x = 0

y += 1

else:

x += 1

# Encode data into image

def encode():

img = input("Enter image name(with extension) : ")

image = Image.open(img, 'r')

data = input("Enter data to be encoded : ")

if len(data) == 0:

raise ValueError('Data is empty')

newimg = image.copy()

encode\_enc(newimg, data)

new\_img\_name = input("Enter the name of new image(with extension) : ")

newimg.save(new\_img\_name, str(new\_img\_name.split(".")[1].upper()))

# Decode the data in the image

def decode():

img = input("Enter image name(with extension) : ")

image = Image.open(img, 'r')

data = ''

imgdata = iter(image.getdata())

while True:

pixels = [value for value in imgdata.\_\_next\_\_()[:3] +

imgdata.\_\_next\_\_()[:3] +

imgdata.\_\_next\_\_()[:3]]

# string of binary data

binstr = ''

for i in pixels[:8]:

if i % 2 == 0:

binstr += '0'

else:

binstr += '1'

data += chr(int(binstr, 2))

if pixels[-1] % 2 != 0:

return data

# Main Function

def main():

a = int(input(":: Welcome to Steganography ::\n"

"1. Encode\n2. Decode\n"))

if a == 1:

encode()

elif a == 2:

print("Decoded Word : " + decode())

else:

raise Exception("Enter correct input")

# Driver Code

if \_\_name\_\_ == '\_\_main\_\_':

# Calling main function

main()