

EE433 – Digital Image Processing

Lab 6 – Image Steganography

The Code:

|  |
| --- |
| #Function to convert a decimal number into an 8-bit binary string def convertTo8BitBinary(num):  bnr = bin(num).replace('0b','') x = bnr[::-1] #this reverses an array.  while len(x) < 8:  x += '0' bnr = x[::-1] bnr = '0b' + bnr return bnr    #Function to convert an 8-bit binary string to a decimal number def convertToDenary(bnr):  placevalue = 0 value = 0 for i in range(9, 1, -1):  value += int(bnr[i]) \* (2 \*\* placevalue) placevalue += 1 return value    #Function to perform a bitwise AND operation between an 8bit binary string and a mask value def bitwiseAndWithMask(bnr, mask): if mask == 1:  bit = int(bnr[9]) andResult = bit & 1 if andResult == 1: return '0b00000001' else:  return '0b00000000' elif mask == 2:  bit = int(bnr[8]) andResult = bit & 1 if andResult == 1: return '0b00000010' else:  return '0b00000000' elif mask == 4:  bit = int(bnr[7]) andResult = bit & 1 if andResult == 1: return '0b00000100' |

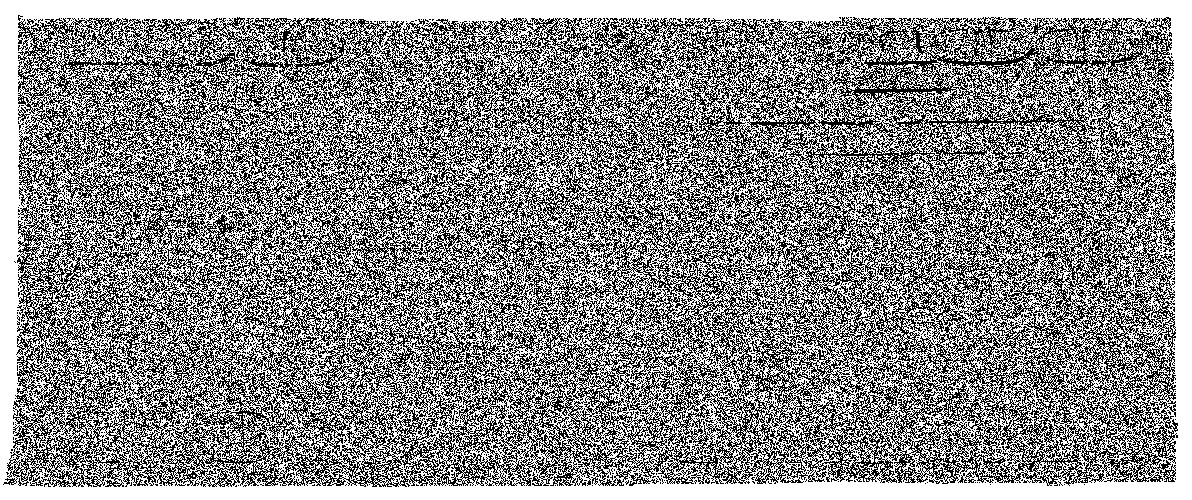
|  |
| --- |
| else:  return '0b00000000' elif mask == 8:  bit = int(bnr[6]) andResult = bit & 1 if andResult == 1: return '0b00001000' else:  return '0b00000000' elif mask == 16: bit = int(bnr[5]) andResult = bit & 1 if andResult == 1: return '0b00010000' else:  return '0b00000000' elif mask == 32: bit = int(bnr[4]) andResult = bit & 1 if andResult == 1: return '0b00100000' else:  return '0b00000000' elif mask == 64: bit = int(bnr[3]) andResult = bit & 1 if andResult == 1: return '0b01000000' else:  return '0b00000000' elif mask == 128: bit = int(bnr[2]) andResult = bit & 1 if andResult == 1: return '0b10000000' else:  return '0b00000000'    #Function that uses the above three functions and performs bitplane slicing def planeSlicing(img\_array, plane):  new\_img\_array = np.zeros(img\_array.shape) for i in range(img\_array.shape[0]): for j in range(img\_array.shape[1]): |
| new\_img\_array[i, j] = convertToDenary(bitwiseAndWithMask(img\_array[i  , j], 2 \*\* (plane - 1))) new\_img = Image.fromarray(new\_img\_array) new\_img.save("plane" + str(plane) + ".tif")    #Main Code from PIL import Image import numpy as np  img = Image.open("lab05.tif") img\_array = np.array(img) img\_array\_binary = np.empty(img\_array.shape, dtype=object)  for i in range(img\_array.shape[0]): for j in range(img\_array.shape[1]):  img\_array\_binary[i, j] = convertTo8BitBinary(img\_array[i, j])    planeSlicing(img\_array\_binary, 1) planeSlicing(img\_array\_binary, 2) planeSlicing(img\_array\_binary, 3) planeSlicing(img\_array\_binary, 4) planeSlicing(img\_array\_binary, 5) planeSlicing(img\_array\_binary, 6) planeSlicing(img\_array\_binary, 7) planeSlicing(img\_array\_binary, 8) |

The Output:

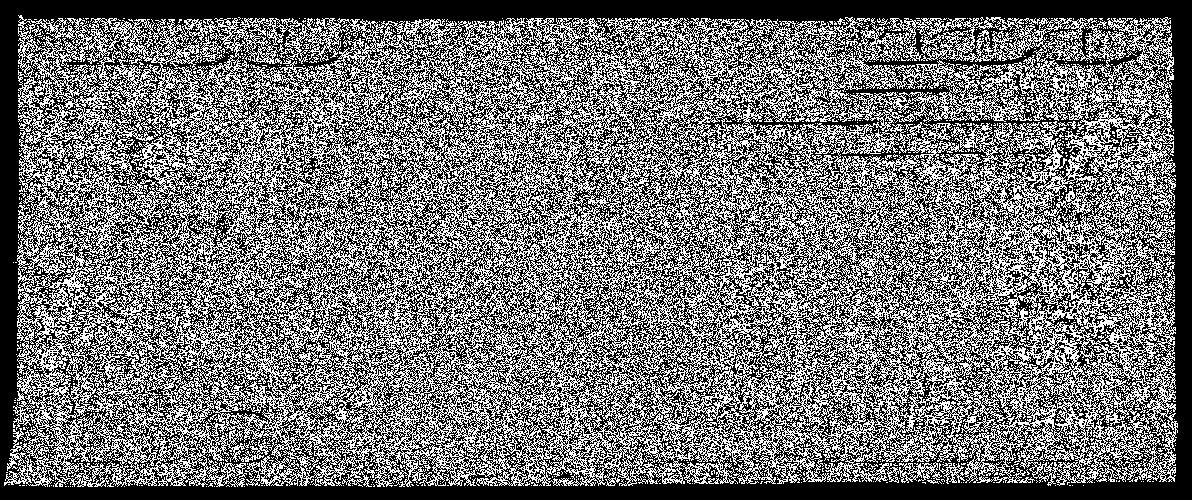
Plane 1:



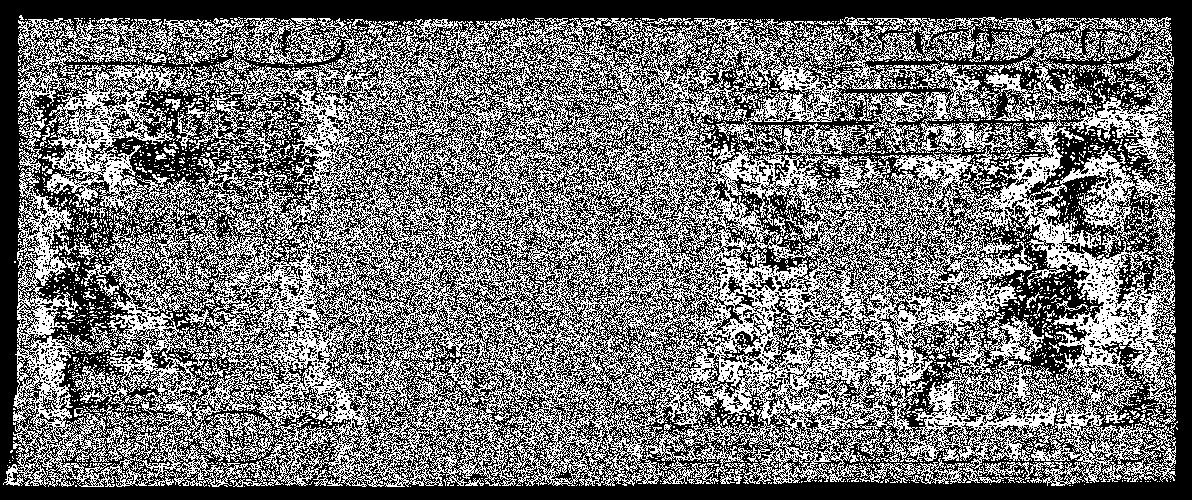
Plane 2:



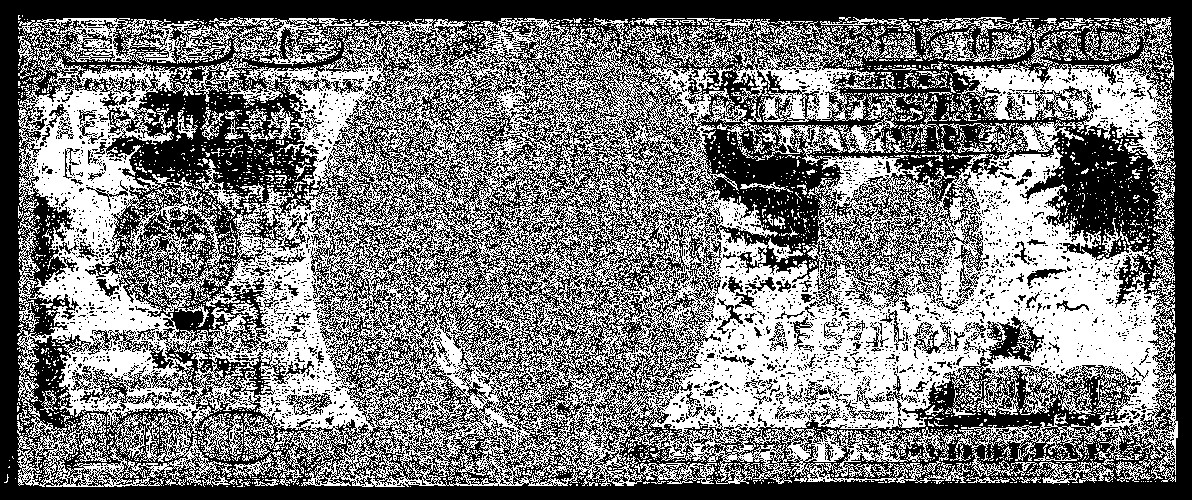
Plane 3:



Plane 4:



Plane 5:



Plane 6:



Plane 7:



Plane 8:

