In [325…
```
import numpy as np
import cv2
```

In [326…
```
s = str("IIITBangalore")
to_bin = []
for count in s:
        ascii_val = ord(count)
        to_bin.append(bin(ascii_val)[2:])
to_bin
```

Out[326]:
```
['1001001',
 '1001001',
 '1001001',
 '1010100',
 '1000010',
 '1100001',
 '1101110',
 '1100111',
 '1100001',
 '1101100',
 '1101111',
 '1110010',
 '1100101']
```

In [327…
```
to_string = ""
for count in to_bin:
        to_string += count
to_string
```

Out[327]:
```
'1001001100100110010011010100100001011000011101110110011111000011101100110111111100101100101'
```

In [328…
```
tolist=[]
for i in to_string:
        tolist.append(1) if i=='1' else tolist.append(-1)
```

In [329…
```
s = np.array(tolist)
```

Coverting the image to B&W

In [330…
```
colored_img = cv2.imread("Untitled512x512-png-images-13.png")
bw_img = cv2.cvtColor(colored_img, cv2.COLOR_BGR2GRAY)
```

Storing the B&W image in a file

In [331…
```
cv2.imwrite("B&W_img.png", bw_img)
```

Out[331]:
```
True
```

Resizing the image to the required dimensions i.e. 512x512

In [332…
```
re_sized_img = bw_img.reshape(512*512)
re_sized_img.shape[0]
```

Out[332]:
```
262144
```

In [333…
```
D = np.random.uniform(low=-500, high=500, size = (s.shape[0], re_sized_img.shape[0]))
ran_k = np.linalg.matrix_rank(D)
ran_k
```

Out[333]:
```
91
```

In [334…
```
Dpinv = D.T @ np.linalg.inv((D@D.T))
```

In [335…
```
alpha = 1e-3
z = Dpinv@(alpha*s - D@re_sized_img)
```

In [336…
```
modi_img = re_sized_img+z
modi_img_op = modi_img.reshape(512,512)
```

In [337…
```
cv2.imwrite("img_with_code.png", modi_img_op)
```

Out[337]:
```
True
```

Decoding message s by multiplying modified image by D which gives D(x+z).

$\hat{s}$ is the signum function of the decoded message

In [338…
```
def Decode_Message(modi_img, D, alpha):
        y = D@modi_img
        s_hat = np.sign(y)
        return s_hat
```

In [339…
```
s_hat = Decode_Message(modi_img, D, alpha)
s_hat
```

Out[339]:
```
array([ 1., -1., -1.,  1., -1., -1.,  1.,  1., -1., -1.,  1., -1., -1.,
        1.,  1., -1., -1.,  1., -1., -1.,  1.,  1., -1.,  1., -1.,  1.,
       -1., -1.,  1., -1., -1., -1., -1.,  1., -1.,  1.,  1., -1., -1.,
       -1., -1.,  1.,  1.,  1., -1.,  1.,  1.,  1., -1.,  1.,  1., -1.,
       -1.,  1.,  1.,  1.,  1.,  1., -1., -1., -1., -1.,  1.,  1.,  1.,
       -1.,  1.,  1., -1., -1.,  1.,  1., -1.,  1.,  1.,  1.,  1.,  1.,
        1.,  1., -1., -1.,  1., -1.,  1.,  1., -1., -1.,  1., -1.,  1.])
```

In [340…
```
s_hat = np.array(s_hat, dtype = np.int32)
s_hat
```

Out[340]:
```
array([ 1, -1, -1,  1, -1, -1,  1,  1, -1, -1,  1, -1, -1,  1,  1, -1, -1,
        1, -1, -1,  1,  1, -1,  1, -1,  1, -1, -1,  1, -1, -1, -1, -1,  1,
       -1,  1,  1, -1, -1, -1, -1,  1,  1,  1, -1,  1,  1,  1, -1,  1,  1,
       -1, -1,  1,  1,  1,  1,  1, -1, -1, -1, -1,  1,  1,  1, -1,  1,  1,
       -1, -1,  1,  1, -1,  1,  1,  1,  1,  1,  1,  1, -1, -1,  1, -1,  1,
        1, -1, -1,  1, -1,  1], dtype=int32)
```

In [341…
```
(s_hat == s).sum()
```

Out[341]:
```
91
```

In [342…
```
s_hat[s_hat == -1] = 0
decoded_msg = ""
for count in s_hat:
        decoded_msg += str(count)
decoded_msg
```

Out[342]:
```
'1001001100100110010011010100100001011000011101110110011111000011101100110111111100101100101'
```

In [343…
```
decoded_msg_bin = []
for count in range(0, len(decoded_msg),7):
        decoded_msg_bin.append(decoded_msg[count:count+7])
decoded_msg_bin
```

Out[343]:
```
['1001001',
 '1001001',
 '1001001',
 '1010100',
 '1000010',
 '1100001',
 '1101110',
 '1100111',
 '1100001',
 '1101100',
 '1101111',
 '1110010',
 '1100101']
```

In [344…
```
ascii_msg = [int(x) for x in decoded_msg_bin]
ascii_msg = [int(x,2) for x in decoded_msg_bin]
ascii_msg
```

Out[344]:
```
[73, 73, 73, 84, 66, 97, 110, 103, 97, 108, 111, 114, 101]
```

In [345…
```
extracted_msg = ""
for count in ascii_msg:
        extracted_msg += chr(count)
extracted_msg
```

Out[345]:
```
'IIITBangalore'
```

Original Image

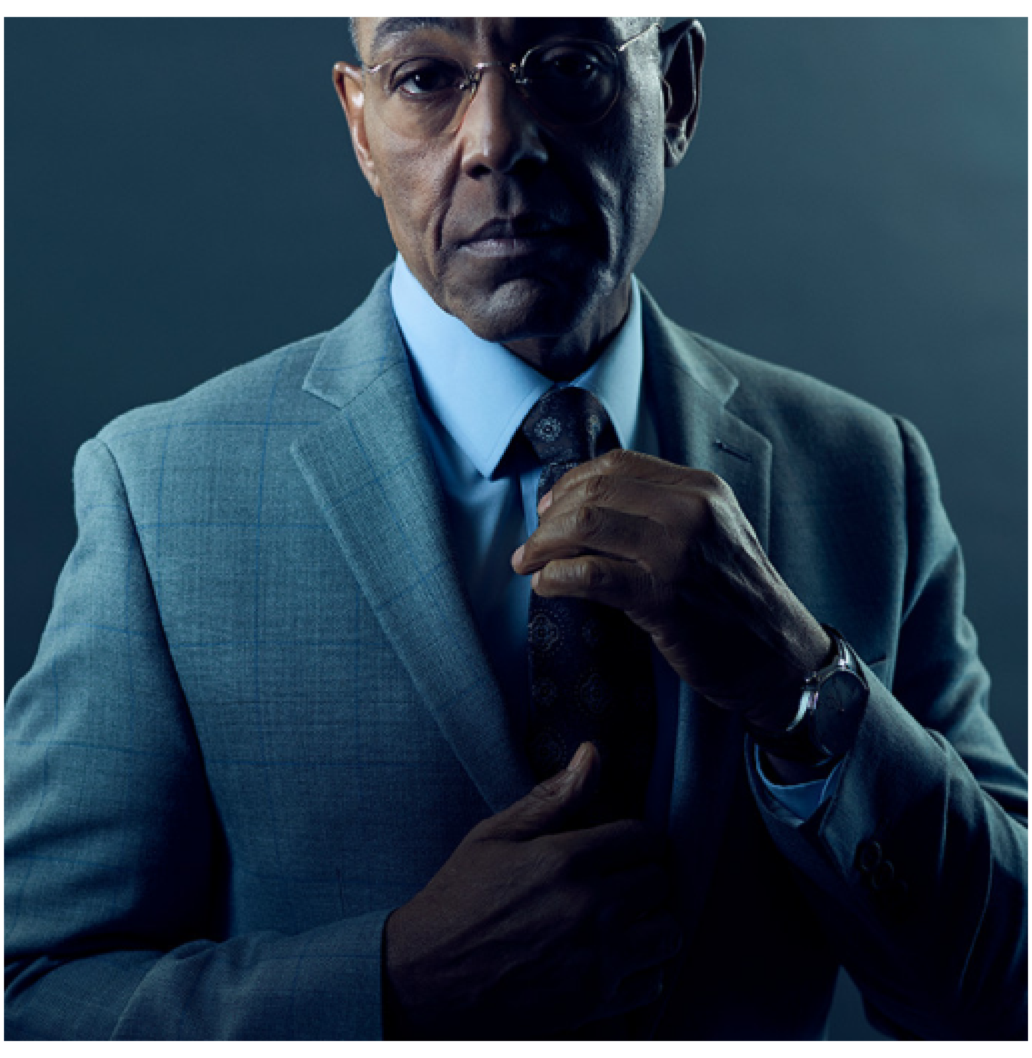
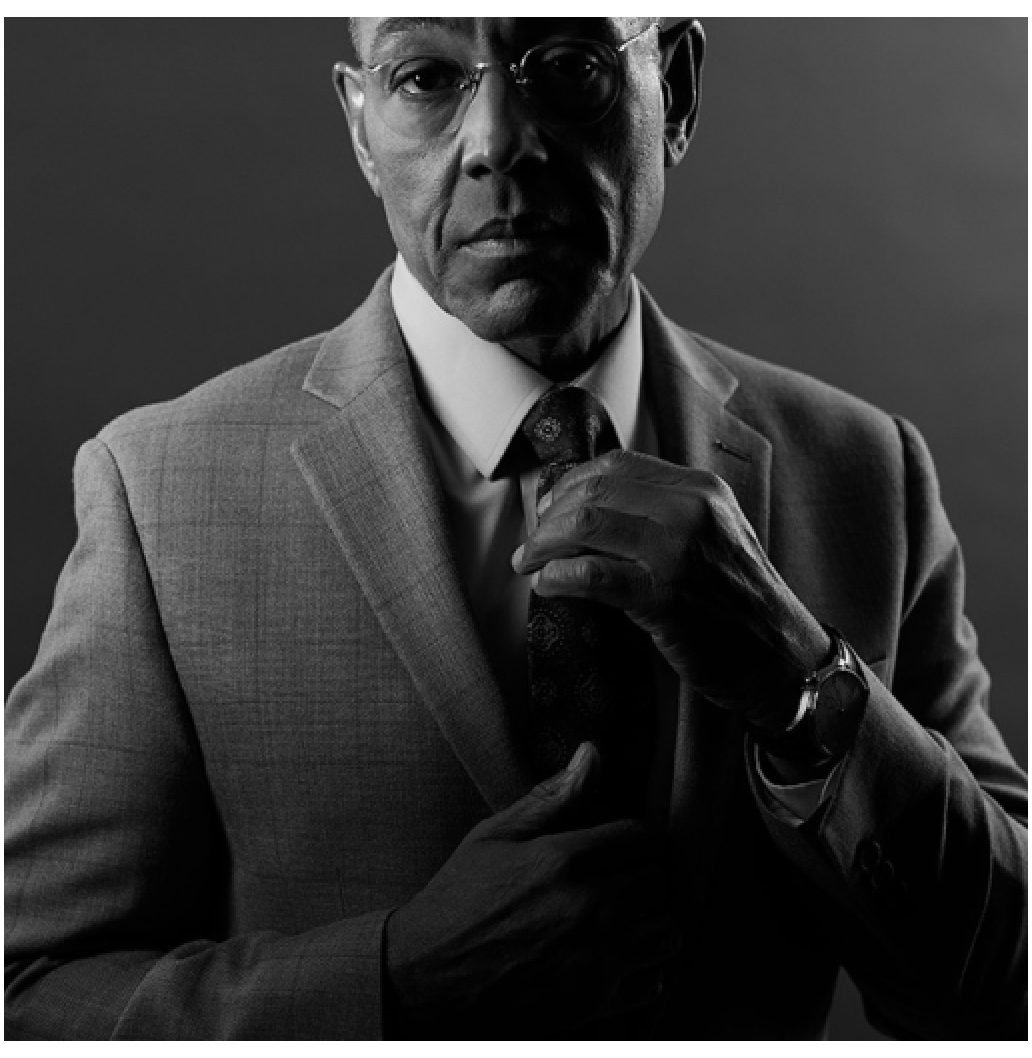
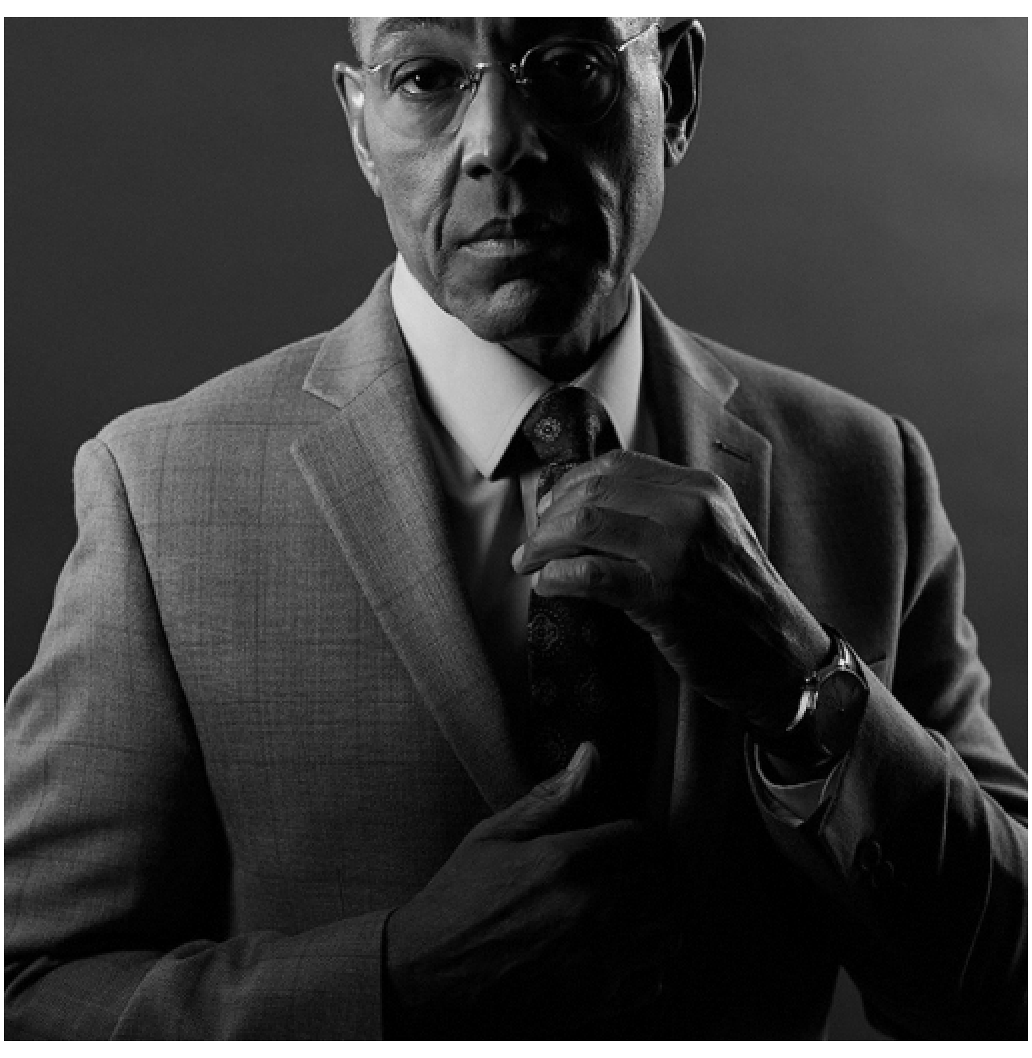


Image after converting to B&W



Coded image



As observed, there is no difference between the B&W image and coded image.