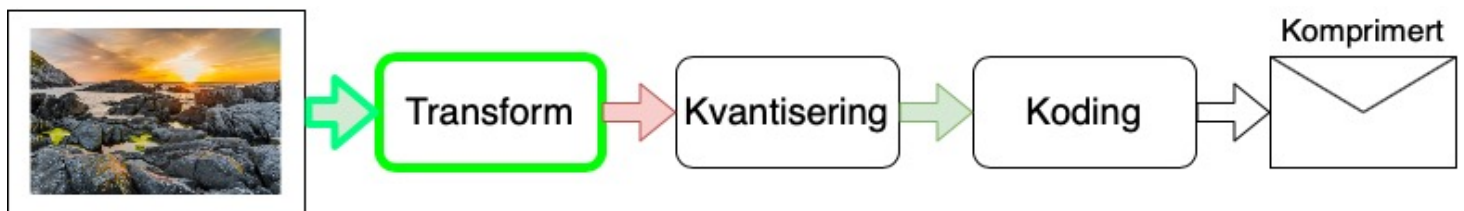


Litt om forrige forelesning: Kompresjon og koding II

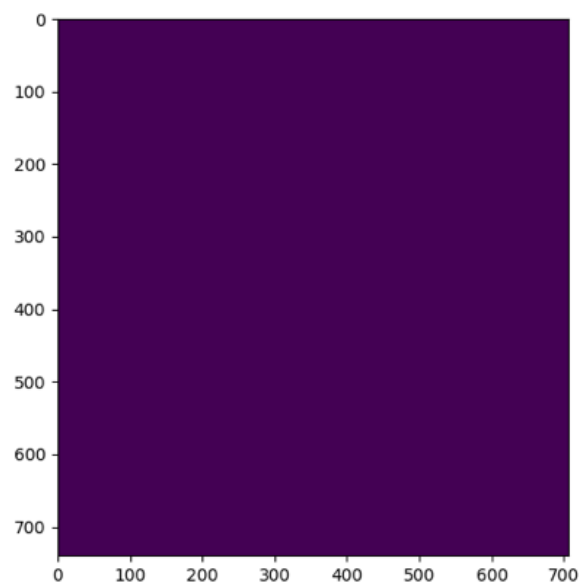
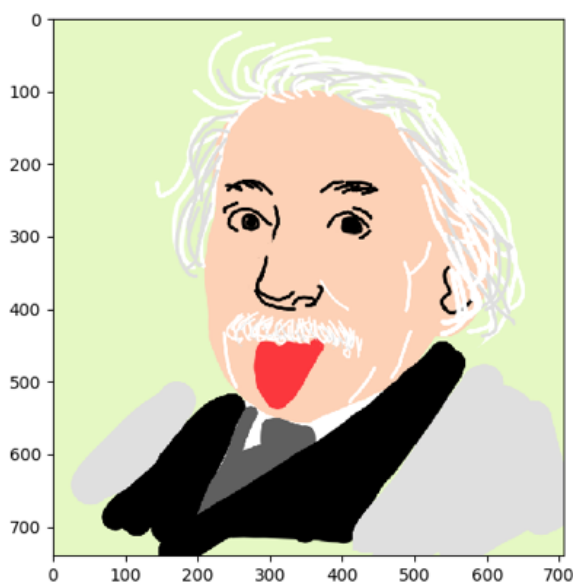
Det som skal skje:

- Differansetransform
- Løpelengdetransform
- Lempel-Ziv-Welch transform
- Bittelitt om prediktiv koding

Denne gangen: Transform



Og intersampel redundans



Differansetransform

Utnytter at nabopiksler er like.

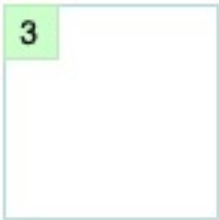
$$\begin{aligned} ny_0 &= img_0 \\ ny_1 &= img_1 - img_0 \\ ny_n &= img_n - img_{n-1} \end{aligned}$$

Eksempel:



3	0	0	1
3	3	0	0
4	0	0	0
2	2	2	2

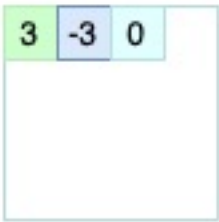
ny_0 = 3



3	0	0	1
3	3	0	0
4	0	0	0
2	2	2	2

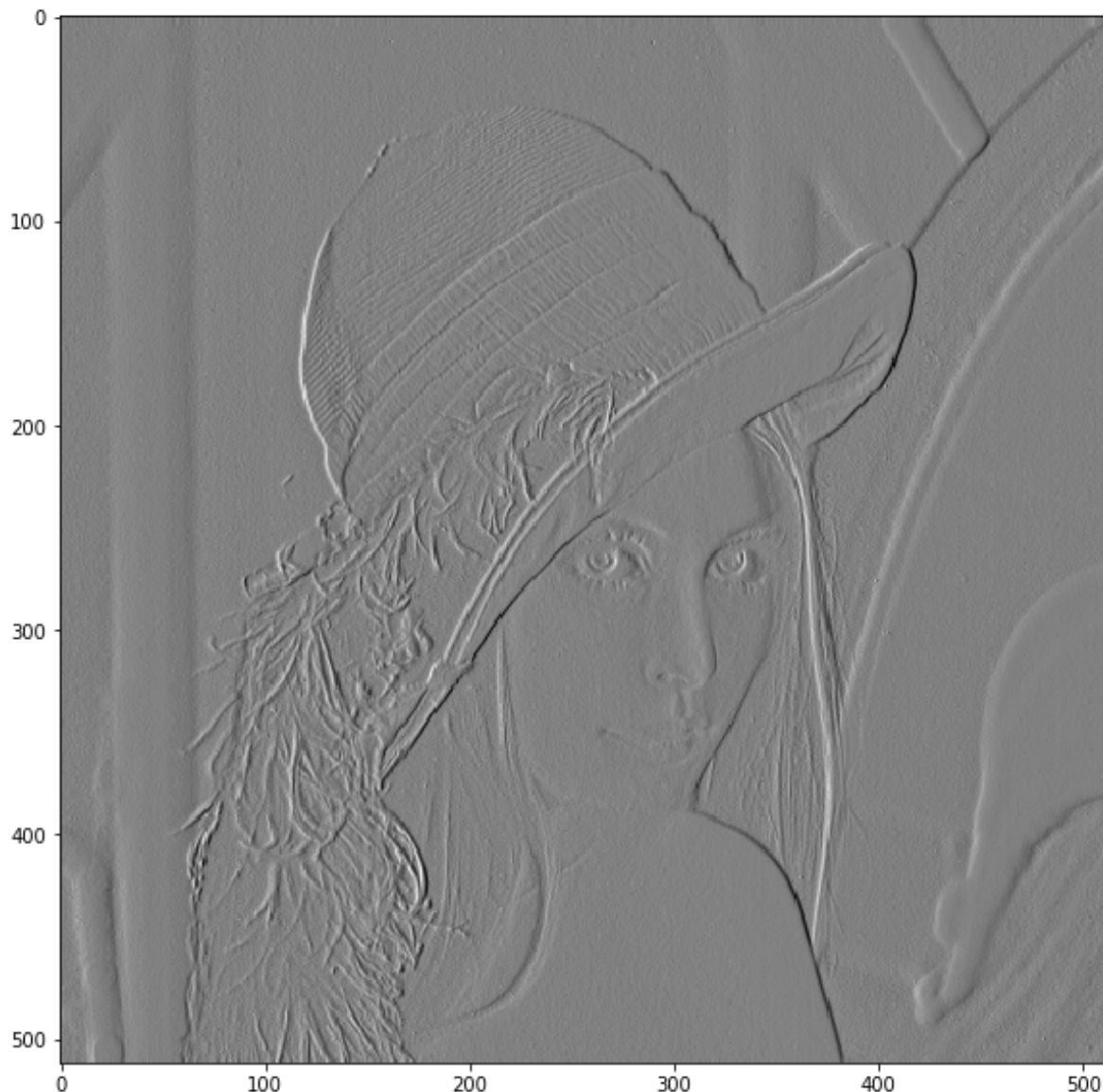
ny_1 = 0 - 3

ny_2 = 0 - 0

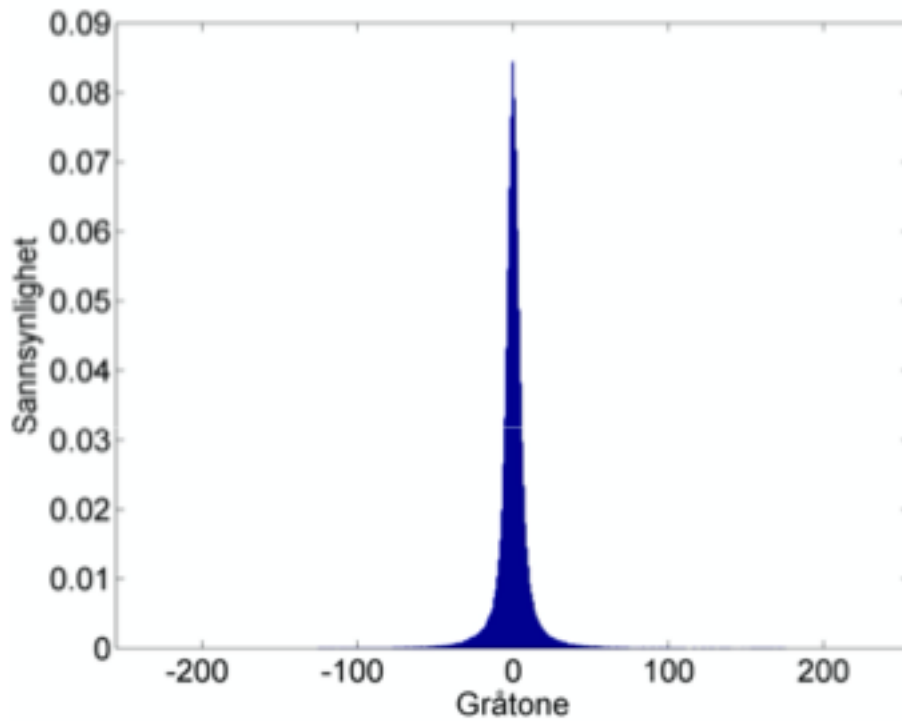


In [12]:

```
from imageio import imread; mona = imread("../images/lena.png",
as_gray=True)
import numpy as np; new_mona = mona[:,1:mona.shape[1]]-mona[:,0:
(mona.shape[1]-1)]
import matplotlib.pyplot as plt; plt.figure(figsize=(10,10)); pl
t.imshow(new_mona, cmap="gray"); plt.show()
```



Verdiene sentreres rundt 0, og mange blir nå like. Lett å videre komprimere!



Løpelengdetransform

Sekvenser av like tall kan lagres som *tallpar*.

Eksempel:

4444 555 7777

Kan lagres som

(4,4) (5,3) (7,4)

Denne transformen sørger ofte for at videre koding er enklere.

Eksempel: Ukeoppgave 2

0	0	1	1	1	2	2	2	2
0	0	1	1	1	1	2	2	2
1	1	0	0	1	1	2	2	2
1	1	2	2	2	2	3	3	3
1	1	1	2	2	2	2	3	3
1	1	1	2	2	2	3	3	3

Algoritme start

(0,2)		1	1	1	2	2	2	2
0	0	1	1	1	1	2	2	2
1	1	0	0	1	1	2	2	2
1	1	2	2	2	2	3	3	3

1	1	1	2	2	2	2	3	3
1	1	1	2	2	2	3	3	3

(0,2)		(1,3)			2	2	2	2
0	0	1	1	1	1	2	2	2
1	1	0	0	1	1	2	2	2
1	1	2	2	2	2	3	3	3
1	1	1	2	2	2	2	3	3
1	1	1	2	2	2	3	3	3

(0,2)		(1,3)			(2,4)			
0	0	1	1	1	1	2	2	2
1	1	0	0	1	1	2	2	2
1	1	2	2	2	2	3	3	3
1	1	1	2	2	2	2	3	3
1	1	1	2	2	2	3	3	3

(0,2)		(1,3)			(2,4)			
(0,2)		1	1	1	1	2	2	2
1	1	0	0	1	1	2	2	2
1	1	2	2	2	2	3	3	3
1	1	1	2	2	2	2	3	3
1	1	1	2	2	2	3	3	3

OSV_I

(0,2)	(1,3)	(2,4)
(0,2)	(1,4)	(2,3)
(1,2)	(0,2)	(1,2)
(1,2)	(2,4)	(3,3)
(1,3)	(2,4)	(3,2)
(1,3)	(2,3)	(3,3)

1. (0,2), (1,3), (2,4)
2. (0,2), (1,4), (2,3)
3. (1,2), (0,2), (1,2), (2,3)
4. (1,2), (2,4), (3,3)
5. (1,3), (2,4), (3,2)
6. (1,3), (2,3), (3,3)

Legg merke til at mange av tallparene forekommer mer enn en gang. Egnestil videre koding.

(0,2)

: 3

(1,3)

: 3

(2,4)

: 3

(1,4)

: 1

(2,3)

: 3

(1,2)

: 3

(3,3)

: 2

(3,2)

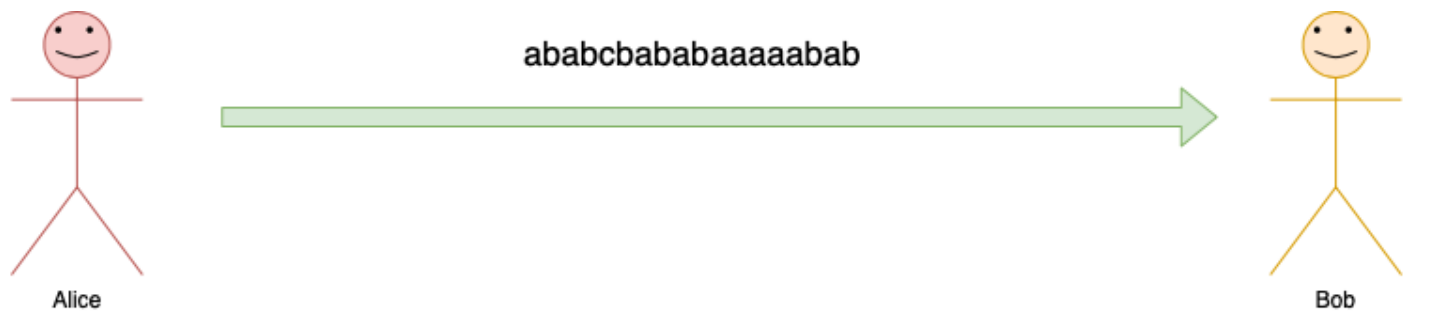
: 1

Lempel-Ziv-Welch

Transformerer en sekvens til en mindre sekvens basert på koder:

- 1. Init: Begge får startkodene
- 2. Kjører til tekst-slutt:
 - Kod og send; motta og tolk
 - Utvid lister

Eksempel: Fra forelesningen



Ser	Sender	Senders liste
		a=0, b=1, c=2

INIT

Mottar	Tolker	Mottakers liste
		a=0, b=1, c=2

ALGORITME START

1. Send og tolk

a babcbababaaaaabab

Ser	Sender	Senders liste
a	0	a=0, b=1, c=2



Mottar	Tolker	Mottakers liste
0	a	a=0, b=1, c=2

--	--	--

--	--	--

2. Utvid lister

a b a b c b a b a b a a a a b a b

Ser	Sender	Senders liste
a	0	a=0, b=1, c=2 ab = 3

Mottar	Tolker	Mottakers liste
0	a	a=0, b=1, c=2

1. Send og tolk

a b a b c b a b a b a a a a b a b

Ser	Sender	Senders liste
a	0	a=0, b=1, c=2 ab = 3
b	1	



Mottar	Tolker	Mottakers liste
0	a	a=0, b=1, c=2
1	b	

2. Utvid lister

a b a b c b a b a b a a a a b a b

Ser	Sender	Senders liste
a	0	a=0, b=1, c=2 ab = 3
b	1	ba = 4

Mottar	Tolker	Mottakers liste
0	a	a=0, b=1, c=2
1	b	ab = 3

a b ab cbababaaaaabab

Ser	Sender	Senders liste
a	0	a=0, b=1, c=2 ab = 3
b	1	ba = 4
ab	3	



Mottar	Tolker	Mottakers liste
0	a	a=0, b=1, c=2
1	b	ab = 3
3	ab	

a b ab cbababaaaaabab

Ser	Sender	Senders liste
a	0	a=0, b=1, c=2 ab = 3
b	1	ba = 4
ab	3	abc = 5

Mottar	Tolker	Mottakers liste
0	a	a=0, b=1, c=2
1	b	ab = 3
3	ab	ba = 4

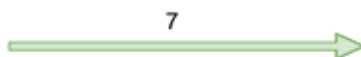
OSV!

a b ab c ba bab aaaaabab

Ser	Sender	Senders liste
a	0	a=0, b=1, c=2 ab = 3

Mottar	Tolker	Mottakers liste
0	a	a=0, b=1, c=2

b	1	ba = 4
ab	3	abc = 5
c	2	cb = 6
ba	4	bab = 7
bab	7	



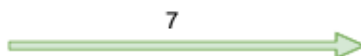
1	b	ab = 3
3	ab	ba = 4
2	c	abc = 5
4	ba	cb = 6
7	???	

Hva vet vi om koden?

1. Lagd forrige gang
2. Starter på ba
3. Laget ved å se på neste bokstav
4. Kunne brukes med en gang

a b ab c ba bab aaaaabab

Ser	Sender	Senders liste
		a=0, b=1, c=2
a	0	ab = 3
b	1	ba = 4
ab	3	abc = 5
c	2	cb = 6
ba	4	bab = 7
bab	7	



Mottar	Tolker	Mottakers liste
		a=0, b=1, c=2
0	a	
1	b	ab = 3
3	ab	ba = 4
2	c	abc = 5
4	ba	cb = 6
7	bab	bab = 7

OSV!

a b ab c ba bab a aa aa bab

Ser	Sender	Senders liste
		a=0, b=1, c=2
a	0	ab = 3
b	1	ba = 4
ab	3	abc = 5
c	2	cb = 6
ba	4	bab = 7
bab	7	baba = 8
a	0	aa = 9
aa	9	aaa = 10
aa	9	aab = 11
bab	7	

Ferdig

Mottar	Tolker	Mottakers liste
		a=0, b=1, c=2
0	a	
1	b	ab = 3
3	ab	ba = 4
2	c	abc = 5
4	ba	cb = 6
7	bab	bab = 7
0	a	baba = 8
9	aa	aa = 9
9	aa	aaa = 10
7	bab	aab = 11

Bittelitt om prediktiv koding

Prediktiv koding i 1D:

Tenk dere rekken:

1 2 3 4 5 x

Hva tror dere x er? Basert på de forrige sier vi 6.

1 2 3 4 5 7

Vi vil lagre denne rekken etter feilene våre, om vi bruker formelen $tall = forrige + 1$:

0 0 0 0 0 1

Jeg håper dere ser at denne lett kan Huffman-kodes!

Generelt:

I prediktiv koding "forutser" du den neste pikselen etter en formel, og lagrer feilen. Feilene kan lett entropikodes, fordi de, som differansetransformen, ofte er små.

Det var alt :)