

Eksamen — Løsningsforslag

INF2310 - Digital Bildebehandling

Eksamen i: INF2310 - Digital Bildebehandling

Eksamensdag: Torsdag 1. juni 2017

Tidspunkt for eksamen: 09:00 - 13:00

Oppgavesettet er på 13 sider

Vedlegg: Ingen

Tillatte hjelpemidler: Ingen

- Det er 8 oppgaver i dette oppgavesettet.
- Les gjennom hele oppgaven før du begynner å løse oppgaven. Kontroller at oppgavesettet er komplett før du begynner å besvare det. Dersom du savner opplysninger i oppgaven, kan du selv legge dine egne forutsetninger til grunn og gjøre rimelige antagelser, så lenge de ikke bryter med oppgavens "ånd". Gjør i såfall rede for forutsetningene og antagelsene du gjør.
- Det er tilsammen 30 delspørsmål, og det lønner seg å disponere tiden slik at man får besvart alle oppgavene. Hvis dere står fast på enkeltoppgaver, gå videre slik at dere får gitt et kort svar på alle oppgaver.
- *Alle delspørsmål teller like mye i evalueringen av besvarelsen.*
- *Alle svar skal begrunnes.* Gjør rede for bruken av eventuelle teoremer, prinsipper eller forutsetninger slik at en tredjeperson kan følge dine resonnementer.
- **Løsningsforslag er i grønn farge, og er på 17 sider.**

Lykke til.

1. Geometriske transformer og resampling

Nærmeste nabo-interpolasjon og bilinear interpolasjon er to vanlige teknikker benyttet ved geometriske operasjoner.

a)

Forklar kort forskjellen mellom de to interpolasjonsteknikkene og nevnt noen fordeler og ulemper med begge.

Bilinear: Ofte visuelt mer behagelig. Resultatet kontinuerlig over x og y . Mer regnekrevende.

Nærmeste nabo: Ofte mindre visuelt behagelig (hakkete kanter). Ikke kontinuerlig resultat i x og y . Innfører ingen nye pikselverdier / beholder viss statistikk i bildet. Mindre regnekrevende.

b)

I hvilke situasjoner vil de gi likt resultat, om de i det hele tatt vil kunne gjøre det?

I tilfeller hvor de transformerte koordinatene er heltallige. For eksempel ved heltallig nedsampling eller 90-graders rotasjon. I tillegg har vi selvfølgelig likt resultat der originalbildet er fullstendig homogent.

c)

Anta at vi har et bilde samlet i begge retninger (x og y) med en $f_s = 7(mm)^{-1}$. Den høyeste frekvensen i bildet er $f_{\max} = 2(mm)^{-1}$. Anta videre at vi utfører den følgende geometriske transformen:

$$x = cx,$$

$$y = cy,$$

der (x, y) er de transformerte koordinatene, og c er en reell konstant (parameter). Etter transformen gjøres det en resampling ved vanlig baklengstransform.

For hvilke c ville vi være i stand til å kunne rekonstruere vårt originalbilde (det før den geometriske transformen)?

Den nye, effektive samplingsraten etter transformen vil være $f_{ny} = cf_s$. Samplingsteoremet krever $f_{ny} > 2f_{\max}$. Følgelig må vi ha at $cf_s > 2f_{\max} = 4(mm)^{-1}$. Altså må $c > 2f_{\max}/f_s = 4/7$.

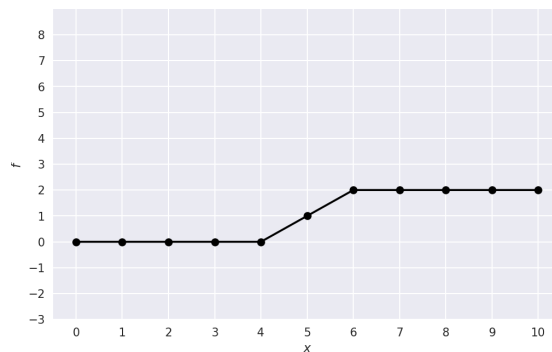
d)

Hvis man for en gitt c frykter at man kan oppleve aliasingproblemer, hvilke konkrete operasjoner (av den typen vi har jobbet med i dette kurset) ville du benyttet på bildet før den geometriske transformen?

Et konvolusjonsbasert lavpassfilter (f.eks. et gjennomsnittsfiler) vil senke f_{\max} , og følgelig øke intervallet for håndterbare verdier av c .

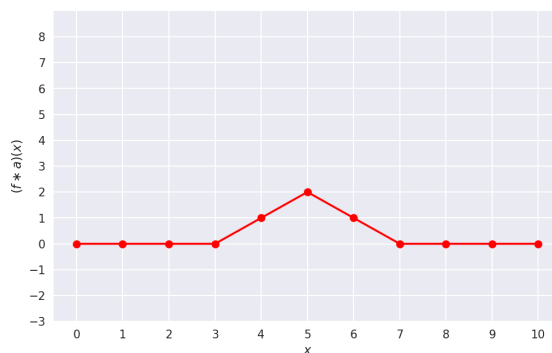
2. 1D konvolusjon

Fig. 1 viser et utsnitt av et 1D signal $f: \mathbb{Z} \rightarrow \mathbb{Z}$, signalene i fig. 2, 3, 4, 5 og 6 er resultater av f konvolvert med filtrene a , b , c , d , og e . I denne oppgaven har vi tatt et utsnitt av den midterste delen av signalene, så du kan ignorere randeffekter. Hvordan ser de respektive filtrene ut? Hva kalles de, og hva kan de bli brukt til?



Figur 1: f

a)

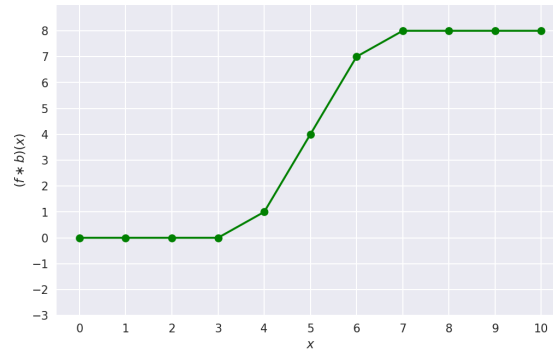


Figur 2: $f * a$

$$a = [1 \quad 0 \quad -1].$$

a er et symmetrisk gradientfilter som blir brukt til å approksimere den førstederiverte av f mhp x . Altså er $g = f * a$ en approksimasjon av $\frac{\partial f}{\partial x}$.

b)

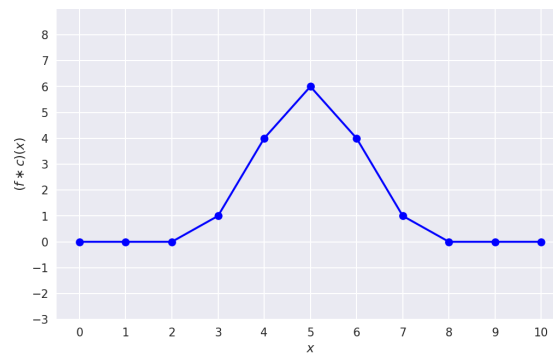


Figur 3: $f * b$

$$b = [1 \quad 2 \quad 1].$$

b er et lavpassfilter som blir brukt til å glatte ut funksjoner, mer spesifikt et unormalisert gaussisk filter.

c)

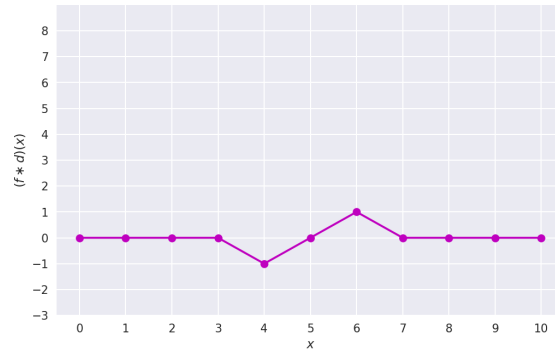


Figur 4: $f * c$

$$\begin{aligned} c &= a * b \\ &= [1 \quad 2 \quad 0 \quad -2 \quad -1]. \end{aligned}$$

c er også et gradientfilter, men det har i tillegg egenskapen at det glatter funksjonen, og er i så måte mer robust mhp. støv enn a .

d)

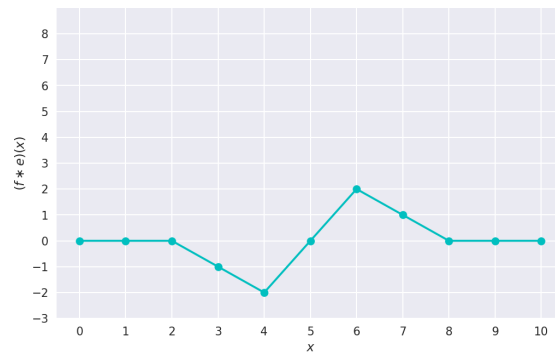


Figur 5: $f * d$

$$d = [-1 \quad 2 \quad -1].$$

d er et høypassfilter, mer spesifikt en 1D *Laplacian*. Resultatet av $f * d$ er en førsteordens approksimasjon av $-\frac{\partial^2 f}{\partial x^2}$.

e)



Figur 6: $f * e$

$$\begin{aligned} e &= d * b \\ &= [-1 \quad 0 \quad 2 \quad 0 \quad -1]. \end{aligned}$$

e blir da en 1D (unormalisert) *Laplacian of Gaussian* (LoG). Altså fungerer den som d , men med et innebygd lavpassfilter er den mer robust mot støy.

3. Konvolusjon og frekvensrespons

a)

I lys av diskrete, todimensjonale bilder; hva sier konvolusjonsteoremet?

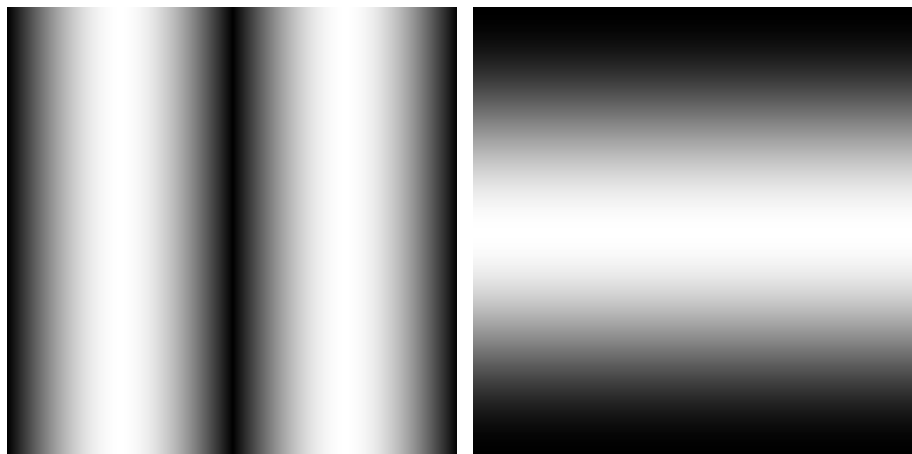
(Sirkel)konvolusjon i billededomenet \iff multiplikasjon i frekvensdomenet (og omvendt)

b)

Under ser dere to konvolusjonskjerne og deres frekvensrespons (spektre). Fremvisningen er slik at null-frekvensen er midt i bildet og lyst indikerer høy verdi mens mørkt indikerer lav verdi. Hvilke konvolusjonskjerne og spektre hører sammen? Begrunn svaret.

$$a = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \quad (1)$$

$$b = [1 \quad 0 \quad -1] \quad (2)$$



(a) Spekter 1

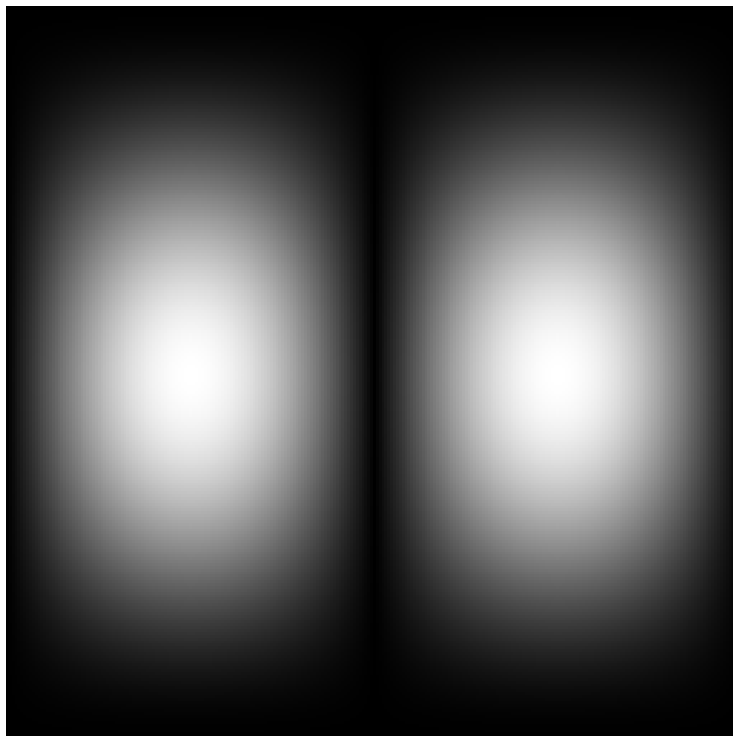
(b) Spekter 2

Figur 7: Frekvensresponser (spektre)

Vertikalt lavpass, og horisontal båndpass. Kjerne a hører til spekter 2.

c)

Hvis vi multipliserer de to spektrene fra forrige oppgave element for element får vi spekteret i fig. 8.



Figur 8: Spektrum

Hvilken konvolusjonskjerne svarer dette spekteret til?

Fra konvolusjonsteoremet vil denne multiplikasjonen tilsvare en konvolusjon av filterkjernene, altså ender vi opp med en horisontal Sobel-operator.

d)

En implementasjon av et gitt konvolusjonsfilter kan gjøres i *frekvensdomenet* ved å transformere både bildet og filterkjernen med en diskret Fourier-transform, gjøre en elementvis multiplikasjon, for så å transformere tilbake til *bildedomenet*.

Hva må vi passe på om vi ønsker et resultat som er mest mulig identisk med en "vanlig" konvolusjonsimplementasjon hvor det har blitt benyttet en implisitt nullutvidelse for å håndtere bilderanden (bildekantene)?

En slik elementvis multiplikasjon tilsvarer en sirkelkonvolusjon i billedomenet. Altså er vi nødt til å nullutvide både filterkjernen og bildet for å unngå at det implisitt benyttes "symmetrisk padding". (En av de obligatoriske oppgavene våren 2017 gikk eksplisitt på denne problemstillingen.)

4. Fourier-transform

a)

Anta at vi har et $N \times N$ -bilde, f , og at vi har gjort en diskret Fourier-transform (i praksis en FFT) av dette bildet og lagret resultatet i $N \times N$ -matrisen F , der $F(0,0)$ gir DC-komponenten/nullfrekvensen. Hvis nå alle koeffisientene i F er null bortsett fra $F(5,3)$ og $F(N-5, N-3)$, beskriv hvordan bildet ser ut.

Et rent "bølgebilde" hvor vi i den ene retningen har 5 repetisjoner av en ren sinusbølge og i den andre retningen 3 repetisjoner. Vi vet dog ingenting om fasen.

b)

For et gitt $N \times N$ -bilde g , ønsker vi å beregne den diskrete Fourier-transformen G , men vi har ingen programpakker tilgjengelig til å utføre den diskrete Fourier-transformen for oss. Vi har dog de to funksjonskallene `getCosImage(i, j, N)` og `getSinImage(i, j, N)`, som gir oss henholdsvis cosinus- og sinus-basisbilder av størrelse $N \times N$ for frekvens (i, j) (antall repetisjoner per N -te piksel i hver retning/dimensjon).

Skisser hvordan du ville benyttet disse to funksjonskallene for å danne $N \times N$ -matrisen G .

Pseudokode:

```
1 for hver i, j:
2   c = getCosImage(i, j, N);
3   s = getSinImage(i, j, N);
4   G(i, j) = sum(g.*c) + complex(0,1)*sum(g.*s);
```

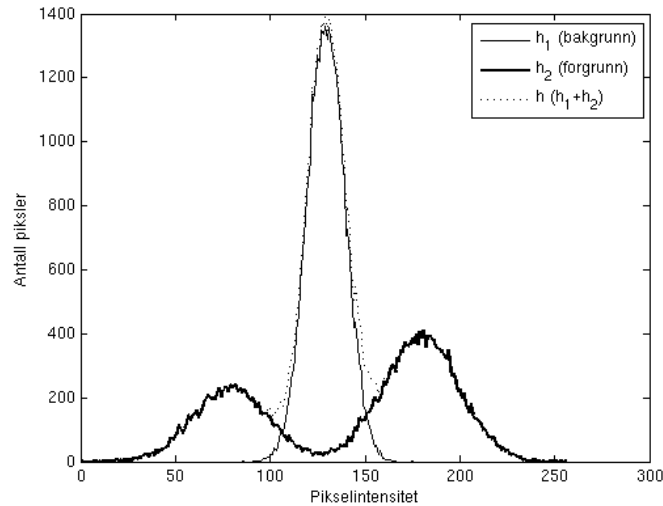
der `.*` menes elementvis multiplikasjon, og summasjonen er over begge dimensjonene.

c)

Anta et bilde med helt sort bakgrunn, og med en banan plassert i bildets sentrum. Hvis man flyttet bananen fem piksler til høyre, dog ikke utenfor bildet, hvordan ville dette endret bildets frekvensspektrum?

Jfr. shiftteoremet ville et slikt "shift" kun endre fasen og ikke selve frekvensbidragene. Spekteret ville altså ikke bli endret.

5. Segmentering ved terskling



Figur 9: Histogram

I fig. 9 vises histogrammet, h , til et bilde som består av to klasser: forgrunn og bakgrunn. Videre har vi tegnet inn histogrammene til disse to klassene, h_1 og h_2 . Vi har altså $h = h_1 + h_2$.

Vårt mål her er å benytte pikslenes gråtoneverdier til å bestemme om en piksel skal settes til forgrunn eller bakgrunn (klassifisere).

a)

Basert på h , h_1 og h_2 i figuren, for hvilke pikselintensitetsintervaller ville du klassifisert til bakgrunn og for hvilke intervaller ville du klassifisert til forgrunn? (Forklar hvorfor.)

Minimerer antall feilklassifiserte piksler ved å velge forgrunn der $h_2(i) > h_1(i)$.

b)

Ved en slik intervallinndeling som du kom frem til i oppgave a), benytt h_1 og h_2 til å gi et uttrykk for antall feilklassifiserte piksler.

Ved fasit-regelen over; antall feilklassifiserte piksler ved hver intensitet summert, altså $\sum_{i=0}^{G-1} \min\{h_1(i), h_2(i)\}$.

c)

La p_1 og p_2 være normaliserte bakgrunns- og forgrunnshistogrammer, og la B og F være *a-priori* sannsynlighet for henholdsvis bakgrunn og forgrunn. Uttrykk h ved hjelp av p_1 , p_2 , B , F og N , der N er antall piksler totalt i bildet.

$N \cdot B$ er antall bakgrunns piksler, $N \cdot F$ antall forgrunns piksler. Vi har dermed $h_1 = NBp_1$, og $h_2 = NFp_2$, altså er

$$\begin{aligned} h &= h_1 + h_2 \\ &= NBp_1 + NFp_2 \\ &= N(Bp_1 + Fp_2). \end{aligned}$$

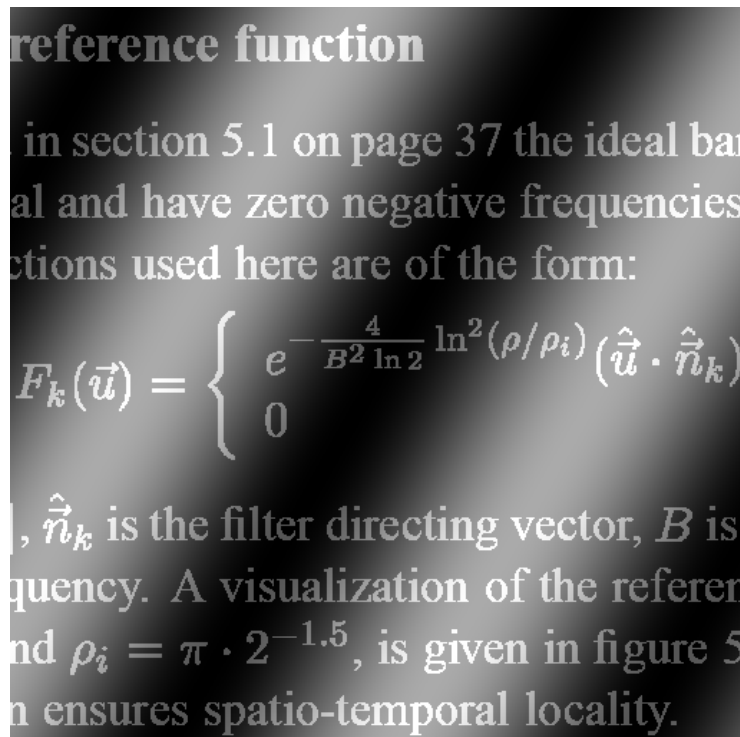
d)

Gitt at vi ikke har h_1 og h_2 tilgjengelig, kun bildet og dets histogram, h . Ville Ridler og Calvaris algoritme (jfr. *k-means*) for automatisk terskling trolig gi et tilfredstillende resultat på dette bildet?

Nei. Histogrammene for forgrunn og bakgrunn tilfredstiller ikke på nær antagelsene bak den nevnte algoritmen; som strengt tatt er at hver klasse er normalfordelt, og med lik varians og *a-priori* sannsynlighet.

6. Mer segmentering

Bildet i fig. 10 inneholder tekst med varierende belysning (der bakgrunnen har store lokale variasjoner i gråtoner).



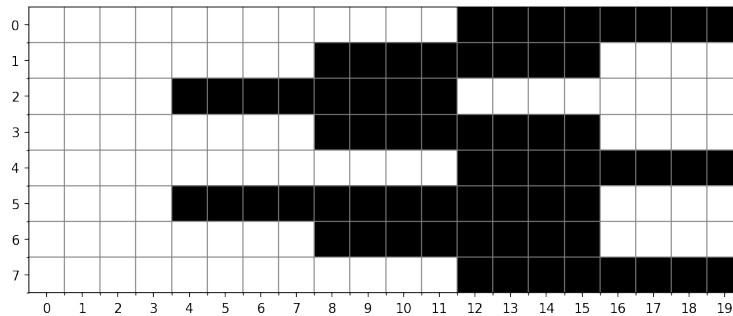
Figur 10: Tekst på varierende bakgrunn

La oss si at vi er interessert i å klassifisere pikslene i bildet over til enten å være en del av forgrunnen (teksten) eller bakgrunnen, altså segmentere bildet. Gi et forslag til hvordan du ville gått frem i dette tilfellet.

Med tankegangen/notasjonen fra forrige oppgave: h_1 og h_2 overlapper (nærmest fullstendig), så selv med riktig h_1 og h_2 kan vi ikke finne en enkel global terskel. Studentene bør komme opp med et begrunnet svar hvor lokal/adaptiv terskling og/eller preprosessering er med.

7. Koding og kompresjon

I denne oppgaven skal du komprimere det binære bildet vist i fig. 11.



Figur 11: Binært bilde

a)

Gjør en løpelengde-transform av bildet. Bruk gjerne symbolet # for å markere linjeslutt (*end-of-line*). Om du ikke får til denne oppgaven, så kan du bruke følgende symbolsekvens i resten av oppgave 7.

dabaacbcaabaacbdabdcbaacbdab

Om en gir løpelengdene som tupler av (verdi, antall) blir transformen

(1, 12), (0, 8), (#, 1), (1, 8), (0, 8), (1, 4), (#, 1), (1, 4), (0, 8), (1, 8), (#, 1), (1, 8), (0, 8), (1, 4), (#, 1),
(1, 12), (0, 8), (#, 1), (1, 4), (0, 12), (1, 4), (#, 1), (1, 8), (0, 8), (1, 4), (#, 1), (1, 12), (0, 8), (#, 1)

Men om en innser at verdiene i transformen alternerer mellom 0 og 1, kan transformen bli mer kompakt. Det er denne versjonen som blir brukt videre i løsningsforslaget for b) og c). Løpelengdetransformen blir da

12, 8, #, 8, 8, 4, #, 4, 8, 8, #, 8, 8, 4, #, 12, 8, #, 4, 12, 4, #, 8, 8, 4, #, 12, 8, #

Det er en en-til-en mapping mellom disse løpelengdesymbolene og abcd, slik at resten av besvarelsen skal bli lik:

↔ b

12 ↔ d

4 ↔ c

8 ↔ a

Siden vi skal komprimere et rektangulært bilde, kunne vi også droppet eol-symbolet # og sendt ved dimensjonene i bildet (i dette tilfellet 8×20). Vi må uansett sende ved

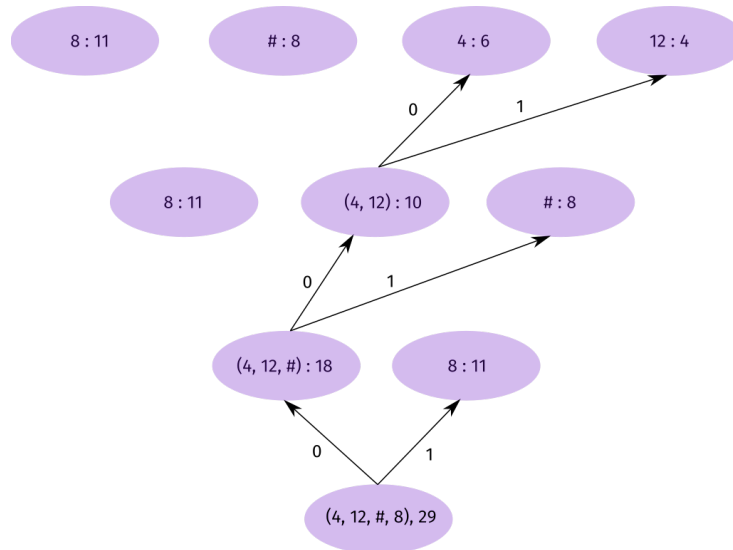
metainformasjon som at første verdi er 1, resten er alternerende, at koden er ordnet med rader først (row-major order), osv. I så tilfelle får en følgende løpelengdetransform

12, 8, 8, 8, 8, 8, 16, 8, 16, 8, 4, 12, 12, 8, 16, 8.

b)

Transformen fra oppgave **a)** resulterer i en sekvens av symboler. Finn Huffman-koden til hvert unike symbol i denne sekvensen. Estimer sannsynlighetsmodellen basert på forekomsten av symbol i sekvensen.

Fig. 12 viser et binært tre som viser hvordan vi kan komme frem til Huffman-kodingen.



Figur 12: Binært tre brukt til Huffman-koding. Denne figuren er kun med i løsningsforslaget.

Table 1: Endelig oppsummerende tabell (denne er kun med i løsningsforslaget)

Symbol	Alternative symbol	Sannsynlighet	Antall forekomster	Kodeord
#	b	0.28	8	01
12	d	0.14	4	001
4	c	0.21	6	000
8	a	0.38	11	1

c)

Bruk Huffman-koden fra **b**) til å encode sekvensen fra oppgave **a**). Hvor lang er den kodede sekvensen (målt i antall bits)? Om vi antar at det originale bildet har 2 bits per piksel, hva er da kompresjonsraten? (Her kan du se bort i fra størrelsen på kodeboken, og annen meta-informasjon om koden og bildet i praksis også er nødvendig for sending/lagring).

Koden blir

001101110000100011011100001001101000001000011100001001101

eller, med skilletegn mellom kodeordene (som ikke er nødvendig, men tatt med for leselighet)

001,1,01,1,1,000,01,000,1,1,01,1,1,000,01,001,1,01,000,001,000,01,1,1,000,01,001,1,01

Denne koden er 57 bits lang. Det originale bildet var på formen 8×20 , så derfor er størrelsen på det $2 \cdot 8 \cdot 20 = 320$ bits¹. Kompresjonsraten blir da

$$c_r = \frac{320}{57} \approx 5.614.$$

d)

Skriv uttrykkene for forventet antall bit per symbol L , og entropi H .

Entropi:

$$H = -\sum_x p(x) \log_2 p(x).$$

Forventet antall bits per kodeord:

$$L = \sum_x p(x) l(x)$$

e)

Hvilken av H og L forventer du at har størst verdi? Og i hvilket tilfelle blir H og L like store. Begrunn svarene dine.

Forventet antall bits L , vil ha større verdi enn entropien H . Entropien er en nedre grense for nettop forventet antall bits per symbol (når man koder symbol for symbol). Denne grensen nåes kun hvis $p(x) = 2^{-k}$ for $k \in \mathbb{N}$ for alle unike x i meldingen.

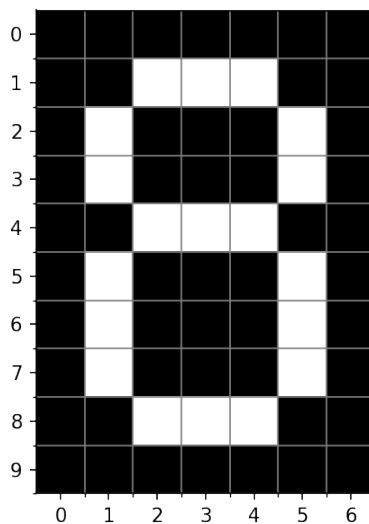
Merk: ved aritmetisk koding (og liknende) kan en komme i tilfeller der $L < H$, dette inntreffer fordi en koder per melding (i stedet for per symbol), og dermed ikke er begrenset til heltallslengder på kodeordene til symbolene.

¹I oppgaveteksten står det eksplisitt at hver pikselverdi er lagret med en type som tar 2 bit. Da verdiene i bildet er binære, så kunne en greid seg fint med 1 bit per piksel, så det er ok å anta det også. (Når det er sagt, så er ofte den minste typen i et språk (f.eks i c++ er char og bool) 1 byte grunnet addresserbarhet, men det blir litt på siden av diskusjonen).

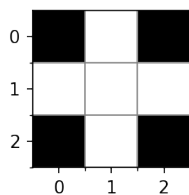
8. Morfologi

Denne oppgaven omhandler morfologi på binære bilder. I bildene under er hvit definert som forgrunn og svart som bakgrunn. I noen oppgaver blir du bedt om å illustrere resultatet som et bilde, det kan da være en tidsbesparende fordel om du tegner det på et ruteark (i motsetning til å skrive pikselverdier 0 og 1 i en matrise).

I alle tilfeller antar vi at strukturelementet har origo i senterpikselen (piksel (1,1) i fig. 14), og at f har bakgrunn utenfor randen av bildet.



Figur 13: Binært bilde f



Figur 14: Binært strukturelement s

a)

La $A \subseteq \mathbb{Z}^2$ og $B \subseteq \mathbb{Z}^2$. Hva er navnet på de to morfologiske operasjonene som er definert i ligning (3a) og ligning (3b)? Hvilke symbol blir brukt (i stedet for spørsmål-

stegnet "?" for de forskjellige operasjonene?

$$A ? B = \{c \in \mathbb{Z}^2 : c = a + b, \text{ for noen } a \in A \text{ og } b \in B\} \quad (3a)$$

$$A ? B = \{c \in \mathbb{Z}^2 : c + b \in A, \text{ for alle } b \in B\} \quad (3b)$$

Ligning (3a) definerer operasjonen morfologisk dilasjon $A \oplus B$, og ligning (3b) definerer operasjonen morfologisk erosjon $A \ominus B$.

b)

Definer følgende operasjoner uttrykt ved morfologisk erosjon og dilasjon:

- (i) Morfologisk åpning av A med B .
- (ii) Morfologisk lukking av A med B .

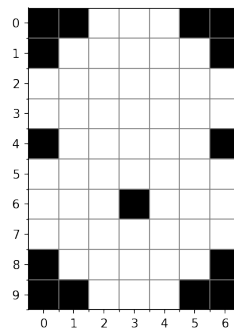
(i) Morfologisk åpning: $A \circ B = (A \ominus B) \oplus B$.

(ii) Morfologisk lukking: $A \bullet B = (A \oplus B) \ominus B$.

c)

La g_d være resultatet av en dilasjon av bildet f med strukturelementet s (som er å finne i henholdsvis fig. 13 og fig. 14). Finn g_d og tegn resultatet.

Resultatet av $g_d = f \oplus s$ er illustrert i fig. 15.

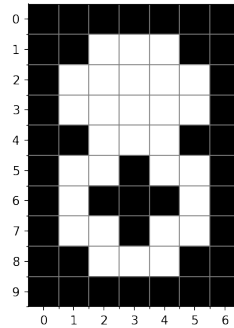


Figur 15: $g_d = f \oplus s$

d)

La g_l være resultatet av en lukking av bildet f med strukturelementet s . Finn g_l og tegn resultatet.

Resultatet av $g_l = f \bullet s$ er illustrert i fig. 16. Siden $f \bullet s = (f \oplus s) \ominus s$ kan vi erodere resultatet fra oppgave c), $f \bullet s = g_d \ominus s$.



Figur 16: $g_l = f \bullet s$