

Ukeoppgaver nr 11 INF2310, våren 2015.

Segmentering ved terskling (oppgavesettet er på 2 sider)

Oppgave 1 - Generell gråtoneklassifikasjon til forgrunn/bakgrunn

Her følger litt Matlab-kode for å få dere i gang. Den laster inn et bilde av en lys tekst på mørk bakgrunn. (Bildet `textImage_clean.png` ligger på undervisningsmateriale/bilder/.)

```
%% Load a "clean" image
im_clean = imread('textImage_clean.png');
im_clean = double(im_clean);
[N M] = size(im_clean);

%% Add some noise
noiseStd = 30; % How much white noise to add
im_noisy = im_clean + noiseStd * randn(N,M);

%% Add varying light-intesity model
lightFactor = 10; % Increasing this increases effect of our varying-light model
lightMask = repmat( ((1:M)-M/2)/M, N, 1);
im_light = im_clean + lightFactor * lightMask;

%% Separate background and foreground pixels using our "clean" image
backgroundPixels = im_noisy(im_clean<150);
foregroundPixels = im_noisy(im_clean>150);
```

1. Studer histogrammet til *im_clean*.
Hvor ville du lagt terskelen for å kunne segmentere ut forgrunn (tekst) fra bakgrunnen?
2. Prøv å endre på hvor mye støy som legges til bildet, *noiseStd*-variabelen, og studer effekten dette har på resultathistogrammet. Hvordan spiller dette inn på muligheten for å skille forgrunn fra bakgrunn ved terskling?
3. Prøv å endre på hvor mye av "lyshets-masken" som blir lagt til. Hvilken effekt har dette på bildets histogram?
4. Hvor stor kan *lightFactor* være før man ikke kan (fullstendig) skille forgrunn fra bakgrunn ved en enkel terskling?
5. Lag et histogram for bakgrunns pikslene og forgrunns pikslene (*h1* og *h2*) hver for seg. Plot de begge i samme figur. Basert på dette, hvor ville du lagt terskelen for å segmentere ut forgrunn fra bakgrunn?
6. Fra *h1* og *h2* og en gitt terskel *t*, hvor mange feilklassifiserte piksler (piksler som skulle vært satt til bakgrunn, men som blir satt til forgrunn, og omvendt) ville man fått?
7. Fra *h1* og *h2*, finn de normaliserte histogrammene *p1* og *p2*, samt *F* og *B* slik at det (totale) normaliserte histogrammet kan skrives $p = B * p1 + F * p2$.

Oppgave 2 - Effekten av glatting

Last inn bildet *discImage.png*. Studer bildets histogram. Utfør middelverdifiltrering med forskjellige filterstørrelser. Studer hva slik filtrering gjør med bildets histogram, og gi en forklaring på hva du ser.

Oppgave 3 - Terskling i praksis

Implementer Otsus metode, Ridler & Calvarids metode (k-means, s. 742 i DIP) , og gjør deg kjent med følgende implementasjon av Niblocks adaptive, lokale metode:

```
function [ret] = getNiblack(im, n, k)

im_localmean = conv2( im, ones(n,n)/n^2, 'same' );
im_localsquared = conv2( im.^2, ones(n,n)/n^2, 'same' );
im_localstd = ( im_localsquared - im_localmean.^2 ).^0.5;

ret = im > (im_localmean + k*im_localstd);
```

Eksperimenter med forskjellige parametre til Niblack-algoritmen på bildene generert fra koden i Oppgave 1. Prøv forskjellige *noiseStd*- og *lightFactor*-verdier. Bland også *im_light* med *im_noisy*. Når er det nødvendig å benytte lokale metoder for å få godt resultat?

Oppgave 4 - Entropi-terskling:

I denne oppgaven skal du få litt trening i å håndtere partielle summer av histogrammer.

Anta at du har gitt et bilde med G trinn i gråtoneskalaen, og med et normalisert histogram $p(i)$, $0 \leq i \leq G-1$. Bildet inneholder ikkje nødvendigvis alle disse gråtonene.

Vi har sett at første ordens entropi gitt ved

$$H = -\sum_{i=0}^{G-1} p_i \log_2(p_i)$$

er et uttrykk for informasjonsinnhold i enkelt-piksler.

Anta at vi beregner entropiene $H_1(t)$ og $H_2(t)$ separat for de delene av histogrammet som er under og over en gitt terskelverdi t , dvs.:

$$H_1(t) = -\sum_{i=0}^t \left(\frac{p(i)}{P_1(t)} \right) \log_2 \left(\frac{p(i)}{P_1(t)} \right), \quad P_1(t) = \sum_{i=0}^t p(i)$$
$$H_2(t) = -\sum_{i=t+1}^{G-1} \left(\frac{p(i)}{P_2(t)} \right) \log_2 \left(\frac{p(i)}{P_2(t)} \right), \quad P_2(t) = \sum_{i=t+1}^{G-1} p(i)$$

Her er $P_1(t)$ og $P_2(t)$ *a posteriori* sannsynlighetene for de to klassene, dvs sannsynlighetene for hhv mørke og lyse piksler *etter* at tersklingen er utført med terskelverdi t .

- Implementer en beregning av funksjonen $A(t) = H_1(t) + H_2(t)$ for alle mulige verdier av terskelverdien t i et gitt bilde.
 - Gjør rede for hvilke forbehold du må ta i koden.
 - Anvend dette på bildet *mona.png*, og terskle bildet med den terskelverdien T_A der $A(t)$ har sitt maksimum.
 - Vil en histogramutjevning påvirke plasseringen av terskelen eller utseendet av det binære bildet?
- Beskriv med tekst og ligninger hvordan beregningen av $H_1(t)$ og $H_2(t)$ kan effektiviseres.