

INF 2310, Obligatorisk oppgave nr 1, våren 2013

Dette oppgavesettet er på tre sider og består av to bildebehandlingsoppgaver.

Besvarelsen av denne og neste obligatoriske oppgave må være godkjent for at du skal få anledning til å gå opp til endelig skriftlig eksamen i kurset.

Besvarelsene kan utarbeides i smågrupper på opptil to studenter, men det er ikke noe i veien for å arbeide alene. Studenter i samme smågruppe kan levere identisk besvarelse, men samarbeidet må framgå av navnene på forsiden av besvarelsen.

Av side 1 skal det fremgå hvem som har utarbeidet besvarelsen.

Det forventes at arbeidet er et resultat av egen innsats.

Å utgi andres arbeid for sitt eget er uetisk og kan medføre sterke reaksjoner fra Ifis side. Se <http://www.mn.uio.no/ifi/studier/admin/obliger/>

Den skriftlige rapporten leveres primært som en pdf-fil som inneholder hele besvarelsen, med figurer og bilder. Kode skal leveres i tillegg til pdf-filen.

Besvarelsen skal leveres via <http://devilry.ifi.uio.no>. Følgende er viktig:

- Alle filene må lastes opp hver gang man skal levere
- Pdf-filen skal ha følgende navn: *inf2310-oblig1-brukernavn*
- Oppgavene skal kunne kjøres fra matlab- eller python-scrips med navn: *oppgave1.m, oppgave2.m* (eventuelt *oppgave2a.m* osv.)
- Spørsmål angående innlevering kan sendes til [sigmunjr@student.matnat.uio.no](mailto:sigmunj@student.matnat.uio.no)

Bildene det refereres til vil være å finne under:

<http://www.uio.no/studier/emner/matnat/ifi/INF2310/v13/undervisningsmateriale/bilder/>

Oppgaven utleveres fredag 22. februar 2013.

Innleveringsfrist er fredag 15. mars 2013.

Oppgave 1 Preprosessering av portrett for ansiktsgjenkjenning

Viktig: Dere skal her programmere transformene “fra bunnen av”, altså ikke bruke ferdige programpakker.

Denne oppgaven går ut på å klargjøre (preprosessere) et portrettbilde slik at det senere kan sendes til en ansiktsgjenkjenningssalgoritme.

Bildet vi skal jobbe med er *portrett.png*. Som vi ser er bildet både skjevt og det har lav kontrast. For å kunne sammenligne dette bildet med andre bilder i en database, er man nødt til å standardisere både geometrien og kontrasten. I denne oppgaven skal vi gjøre dette på følgende måte:

1. Standardisere kontrasten ved å programmere en histogramutjevning.
2. Standardisere geometrien ved å utføre en affin transform slik at øyne og munn passer over en forhåndsdefinert maske. Masken er gitt ved filen *geometrimaske.png*.

Lag en implementasjon av både forlengs- og baklengstransformasjon. Ved baklengstransformasjonen, prøv ut både nærmeste nabo og bilinear interpolasjon.

Dere skal altså ende opp med et resultatbilde som er like stort som maske-bildet (512x600), har god kontrast, og hvor øyne og munn finnes på de samme pikselkoordinatene som i maske-bildet.

Hva skal leveres:

- I. Mellom-resultat-bildet etter histogramutjevningen.
- II. Forklaring på hvordan dere fant koeffisientene til den affine transformen.
- III. Resultat-bilder for både forlengs- og baklengstransformasjonen, samt nærmeste nabo og den bilineære interpolasjonen.
- IV. Kommentarer/forklaringer på eventuelle forskjeller i resultatene ved forlengs- og baklengsmapping, og ved nærmeste nabo og bilinear interpolasjon.
- V. Programkode, eventuelt i en egen fil.

NB!

Dere kan benytte ferdige Matlab-funksjoner til å lese/skrive fra/til fil.

Histogramutjevningen og den geometriske transformen MÅ dere implementere selv.

Oppgave 2 Støy-filtrering

Viktig: Dere skal her programmere alle filtreringene og eventuell utvidelse av inn-bildet "fra bunnen av", altså ikke bruke ferdige programpakker.

Anta at vi har et bilde med noe støy. Vi har sett at støy kan reduseres med ulike filtre. Her skal vi sammenligne middelverdifilteret med to kantbevarende filtre.

Du skal implementere alle filtreringene slik at ut-bildet får samme størrelse som inn-bildet og bilderandproblemet skal håndteres ved bruk av sirkulær indeksering.

- a) Programmér en generell implementasjon av konvolusjon av et bilde med et konvolusjonsfilter. Du kan anta at naboskapet er rektangulært med odde lengder og har origo i senterpikselet.
- b) Programmér en implementasjon av et 5×5 -“Max-homogenitet”-filter med 9 stk 3×3 -sub-naboskaper. Responsen i et piksel skal være middelverdien fra det sub-naboskapet som gir lavest kontrast i form av lavest differanse mellom maksimum og minimum pikselverdi. Filteret er beskrevet i forelesning 6.

Tips: En mulig fremgangsmåte for denne implementeringen er:

1. Ta utgangspunkt din implementasjon i a) og forenkle den slik at den bare beregner 3×3 -middelverdifiltrering, men at den lar ut-bildet ha en ekstra 1-piksels ramme (hver lengde av ut-bildet blir 2 større enn inn-bildet). Håndter fortsatt bilderandproblemet ved bruk av sirkulær indeksering.
2. Utvid denne implementasjonen slik at den samtidig (som den beregner middelverdien i 3×3 -naboskapet) også beregner lokal kontrast i form av differanse mellom maksimum og minimum pikselverdi innenfor 3×3 -naboskapet.
3. Implementasjonen din vil nå produsere et 3×3 -middelverdifiltrert bilde og et bilde med lokale kontraster innenfor 3×3 -naboskapet.
 - i. Løp gjennom kontrastbildet med et sentrert 3×3 -naboskap.
 - ii. For alle (x,y) der hele naboskapet er innenfor bildet, finn laveste verdi innenfor naboskapet. Kall denne pikselposisjonen (v,w) .
 - iii. La “Max-homogenitet”-responsen i $(x-1,y-1)$ være verdien av det 3×3 -middelverdifiltrerte bildet i (v,w) . Grunnen til at det blir -1 er at ut-bildet skal ha samme størrelse som inn-bildet, mens middelverdi- og kontrastbildet er utvidet med en 1-piksels ramme.

(oppgave 2 fortsetter på neste side)

- c) Implementér et ”symmetrisk nærmeste nabo (SNN)”-filter. Responsen i (x,y) skal være middelveien av $1+(n^2-1)/2$ piksler i et $n \times n$ -naboskap, der pikslene er (x,y) og det pikselet i hvert symmetriske pikselpar rundt senterpikslet som er nærmest senterpikslet i pikselverdi. Du kan anta at n er et oddetall og at naboskapet har origo i senterpikselet. Filteret er beskrevet i forelesning 6.

Tips: For å beregne responsen i (x,y) kan du gå frem slik:

1. Initialiser en teller til 1 og en sum til pikselverdien av (x,y) .
 2. Løp gjennom $n \times n$ -naboskapet rundt (x,y) i en dobbel for-løkke. For hvert piksel (v,w) der $v < n$ eller $v = n$ og $w < n$, sammenlign med det senterpeilede pikselet og adder den pikselverdien som ligner mest på (x,y) til summen, og øk telleren med 1. Betingelsene på pikselposisjonen (v,w) skyldes at hver pikselpar bare skal brukes én gang.
 3. SNN-responsen i (x,y) er sum/teller.
- d) Sammenlign resultatet ved bruk av et 3×3 -middelveidifilter i alternativ a) og resultatene fra alternativ b) og c) med $n=5$, på bildet *shapes_noise15.png*.
- e) Hvis det er slik at filtrene i b) og c) er mer kantbevarende enn middelveidifilteret, så vil vi finne kantene ved å se på absoluttverdien av differansen mellom resultatene. Denne differansen må gjerne histogramtransformeres for at vi skal kunne se effekten.

Hva skal leveres:

- VI. Beskrivelse av hvordan alternativ a), b) og c) er implementert.
- VII. Resultatbilder fra de tre alternativene i d), gjerne med gråtonetransformer som får fram forskjellene.
- VIII. Drøfting av resultatbildene.
- IX. Drøfting av differansen mellom resultatbildene.
- X. Programkode.

Lykke til!