

# INF 2310, Obligatorisk oppgave nr 1, våren 2012

**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 helebesvarelsen, med programkode, figurer og bilder. Kode og bildefiler kan også leveres i tillegg til pdf-filen, men da skal alt pakkes i én fil, for eksempel med zip. Besvarelsen skal sendes per e-post til gruppelæreren. Følgende er viktig:

- Filene skal være pakket til én fil, f.eks. med zip, med følgende navn: **inf2310-oblig1-brukernavn**, som en pdf eller zip-fil, der brukernavn byttes ut med ditt eget brukernavn.
- Filen sendes som vedlegg i en e-post til [runarfu@student.matnat.uio.no](mailto:runarfu@student.matnat.uio.no).
- Husk å sette «INF2310 Oblig1» i subjekt-feltet!
- Ved ønske om annen leveringsform, kontaktes gruppelærer.

Bildene det refereres til vil være å finne under:

<http://www.uio.no/studier/emner/matnat/ifi/INF2310/v12/undervisningsmateriale/bilder/>

Oppgaven utleveres mandag 20. februar 2012.

Innleveringsfrist er fredag 9. mars 2012.

## 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) portrettbilder slik at det senere kan sendes til en ansiktsgjenkjenning algoritme.

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 *geotrimaske.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

Anta at vi har et bilde med noe støy. Dere har sett at vi kan redusere slik støy med ulike filtre. Her skal vi sammenligne et ordinært middelverdi-filter med et kant-bevarende filter.

- a) Programmér en implementasjon av konvolusjon av et bilde med et  $w \times w$  middelverdi-filter, der  $w$  er et oddetall. Dere må kunne håndtere rimelige verdier av  $w$ , og håndtere rand-effekter med "circular indexing". **NB!** 2D-konvolusjonen må dere implementere med «for-løkker», ikke med kall til ferdige konvolusjons-rutiner.
- b) Programmér en implementasjon av et  $5 \times 5$  "Max-homogenitet" filter med 9 stk  $3 \times 3$  sub-vinduer, modifisert slik at ut-verdien er middelverdien av det sub-vinduet som har lavest kontrast, i form av lavest differanse mellom maksimum og minimum gråtone innenfor sub-vinduet. Filtret er beskrevet i forelesningen. Her er det mulighet for en implementasjon som bruker a) som utgangspunkt!
- c) Implementér et "symmetrisk nærmeste nabo (SNN)" filter, som beregner middelverdien av  $1+(w^2-1)/2$  piksler innenfor et  $w \times w$  vindu – der  $w$  er et oddetall - ved å velge ut senterpikslet pluss det pikslet i hvert symmetriske pikselpar rundt senterpikslet som er nærmest senterpikslet i gråtoneverdi. Original-beskrivelsen av filtret finnes i Harwood et al., Pattern recognition Letters 6, pp155-162, 1987.
- d) Sammenlign resultatet ved bruk av  $w=3$  i alternativ a) og resultatetene fra alternativ b) og c) med  $w=5$ , på bildet *shapes\_noise15.png*.
- e) Hvis det er slik at b) og c) er mer kantbevarende enn a), 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. Resultat-bilder fra de tre alternativene, gjerne med gråtonetransformer som får fram forskjellene.
- VIII. Drøfting av resultat-bildene.
- IX. Drøfting av differansen mellom resultatene
- X. Programkode, eventuelt i en egen fil.

Lykke til!