

Obligatorisk oppgave 1

INF2310 Våren 2020

Dette oppgavesettet er på 4 sider, og består av 2 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 forsiden 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.uio.no/studier/admin/obligatoriske-aktiviteter/mn-ifi-oblig.html>

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

Besvarelsen skal leveres via <https://devilry.ifi.uio.no>

Legg merke til følgende:

- Alle filene må lastes opp hver gang man skal levere.
- PDF-en skal ha følgende navn: inf2310-oblig1-brukernavn.pdf, der brukernavn byttes ut med ditt eget sådan.
- Oppgaven skal kunne kjøres fra Matlab- eller Python- skript med navn: oppgave1.m eller oppgave1.py (evt. oppgave1a.m osv.).
- Spørsmål angående innlevering til Mia: mkkvalsu@student.matnat.uio.no

Bildene det refereres til vil være å finne under:

<https://www.uio.no/studier/emner/matnat/ifi/INF2310/v20/undervisningsmateriale/bilder/>

Oppgaven utleveres onsdag 26. februar 2020

Innleveringsfrist er fredag 20. mars 2020

Lykke til!

1 PREPROSESSERING AV PORTRETT FOR ANSIKTSGJENNKJENNING

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 ansiktsgjenkjenningssalgoritme.

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

1. Standardisere kontrasten ved å benytte en lineær gråtonetransform spesifisert slik at resultatbildet får en middelvei på 127 og standardavvik på 64. Bildet skal kunne lagres med 8 bit (uint8, altså ha verdier mellom 0 og 255.)
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 bilineær interpolasjon.

Dere skal altså ende opp med et resultatbilde som er like stort som maske-bildet (512 X 600 piksler), har god kontrast, og hvor øyne og munn finnes på de samme pikselkoordinatene som i maske-bildet.

NB!

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

Gråtonetransformen og den geometriske transformen MÅ dere implementere selv.

Hva skal leveres:

- I. Mellom-resultat-bildet etter gråtonetransformen.
- 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 baklengs-mapping, og ved nærmeste nabo og bilineær interpolasjon.
- V. Programkode.

2 Global og lokal histogramutjevning

Viktig: Dere skal her programmere histogramutjevning (Histogram Equalization, HE) adaptiv histogramutjevning (AHE) og kontrast-begrenset histogramutjevning (CLAHE).

Vi skal i her implementere ulike varianter av histogramutjevning på bilder, for å forbedre kontrasten i bildet.

Bildet vi kommer til å bruke, er et røntgenbilde hentet fra <https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia> .

For at kjøringen av deres implementasjon i deloppgave a), b) og c) ikke skal ta alt for lang tid, er det lurt å teste med små bilder først.

Bildet « xray_small.jpg » er en mindre variant av bildet dere skal bruke deres implementasjon på i deloppgave d). Dette bildet kan dere bruke i deloppgave a), b) og c).

Den histogramutjevningen som er forelest (HE), er global i den forstand at den bruker det kumulative histogrammet fra hele bildet til å transformere alle pikselverdiene i hele bildet. Dette er det første steget i oppgaven:

- a) Programmér en generell implementasjon av histogramutjevning av hele bildet. Inn-bildet er et 8 bits gråtonebilde, men en parameter i implementasjonen skal gjøre det mulig å endre antall bits i ut-bildet ($\text{bits_ut} = 1, \dots, 8$).

Du vil nok se at slik histogramutjevning fungerer utmerket hvis fordelingen av gråtoner er omtrent den samme over hele bildet. Men hvis bildet inneholder områder som er signifikant lysere eller mørkere enn resten av bildet, så vil ikke kontrasten i disse regionene bli særlig forbedret.

Adaptiv histogramutjevning transformerer hver pikselverdi ved hjelp av det kumulative histogrammet til et naboskap omkring pikselet, for eksempel et kvadrat med oddetalls sider.

- b) Programmér en slik adaptiv histogramutjevning (AHE). Diskuter hvordan du kan behandle pikslene nær bilderanden ved å utvide størrelsen på inn-bildet, hvilke effekter du vil få av ulike valg, og implementér så bare en metode for utvidelse av inn-bildet.

AHE krever at du finner et nytt histogram – og tilhørende kumulativt histogram – for hver ny pikselposisjon. Dette gjør metoden veldig langsom. Vi vil derfor at du implementerer en mye raskere variant, der histogrammet oppdateres idet du flytter senterpikselet ett skritt ved å trekke fra den raden

pikslar som ble borte og legge til den raden som kom inn i naboskapet. Se Huang, Yang and Tang, «A Fast Two-Dimensional Median Filtering Algorithm», IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, VOL. ASP-27, NO. 1, FEBRUARY 1979 (vedlagt).

Du vil nå se at AHE har en tendens til å (over)forsterke kontrasten i deler av bildet der pikselverdien er nær konstant, fordi det lokale histogrammet i homogene deler av bildet er veldig konsentrert.

I kontrast-begrenset histogramutjevning (CLAHE) begrenser man kontrasten ved å klippe toppen av histogrammet ved en predefinert verdi og fordele de avklippede sannsynlighetene over hele histogrammet (eventuelt iterativt), før man lager det lokale kumulative histogrammet.

- c) Implementér en Sliding Window CLAHE, ved at du bruker lokal histogramoppdatering som beskrevet av Huang, Yang, Tang, og histogramklipping som beskrevet ovenfor.
- d) Anvend dine implementasjoner på bildet « xray.jpg » Eksperimenter med parametrene, og drøft resultatene.

NB!

Dere kan benytte ferdige Matlab/Python-funksjoner til å lese/skrive fra/til fil. Histogramutjevningen (Histogram Equalization, HE), den adaptive histogramutjevningen (AHE) og den kontrast-begrenset histogramutjevningen (CLAHE) MÅ dere implementere selv.

Hva skal leveres:

- VI. Tekstlig beskrivelse av implementeringene i a) og b).
- VII. Resultatbildet med tilhørende parameterverdier i c).
- VIII. Drøfting av resultatbildet fra deloppgave d), inkludert hva som kjennetegner det metoden er bra på og det metoden ikke er bra på, samt hvordan dette blir påvirket av å endre parameterverdiene.
- IX. Programkode.

Lykke til!