

# Prosjektoppgave 2

## TEK5020/9020 - MØNSTERGJENKJENNING

Høsten 2022

### 1 Innledning

Oppgaven består i *segmentering* av bilder, dvs. dele inn bilder i meningsfulle og/eller perseptuelt uniforme regioner. Segmentering brukes ofte i bildeanalyse for å forenkle bildet med tanke på videre analyse, f.eks. for å finne fysiske objekter av bestemte typer i scenen. Et eksempel på segmentering er vist i fig. 1.



Figur 1: Eksempel på segmentering. Originalbildet (t.v.) er her delt inn i regioner som svarer til “gate”, “fortau”, “bygninger”, “himmel” mm. (t.h.).

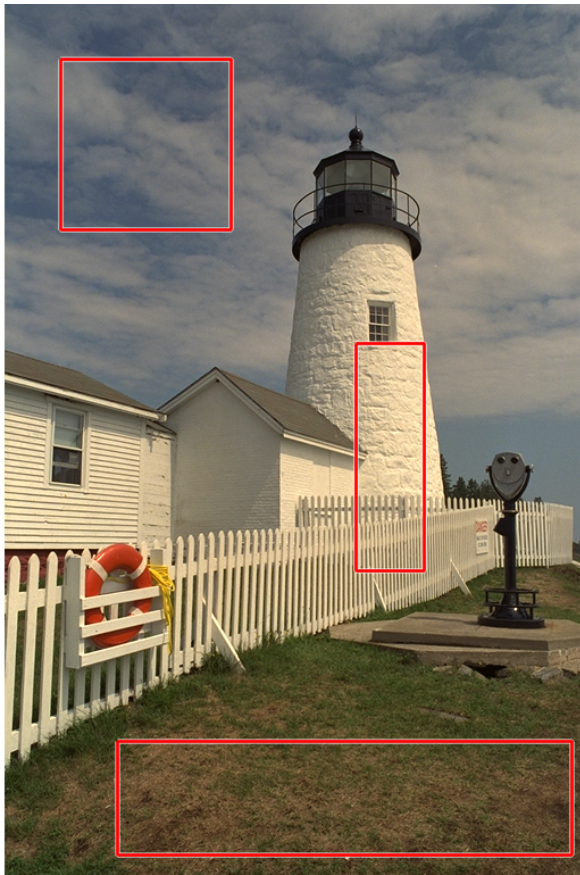
Det er mange måter å foreta en bildesegmentering på. I denne oppgaven er tanken å gjøre dette ved å klassifisere hvert enkelt piksel i bildet til en av de ønskede segmenttypene (dvs. klassene), ut fra farge og intensitet i bildet. Det er også mulig å inkludere egenskaper som avhenger av nabopikslene, f.eks. beregne ulike mål på lokal tekstur, men det skal ikke gjøres i denne oppgaven.

### 2 Oppgavebeskrivelse

Til hvert piksel i et digitalt fargebilde er det knyttet tre intensitetsverdier i hhv. rødt, grønt og blått (R, G, B). Hvert piksel kan derfor betraktes som et punkt i et tredimensjonalt egenskapsrom (RGB-rommet), slik at pikselet er representert ved egenskapsvektoren

$$\mathbf{x} = [R, G, B]^t,$$

der R, G og B angir styrken til hver av fargekomponentene. I de vedlagte bildefilene ligger disse verdiene i intervallet 0 til 255. Det skal trenes opp en klassifikator som tilordner hver egenskapsvektor av denne typen til en av de valgte klassene (segmentene).



(a) Treningsbilde med utvalgte treningsregioner



(b) Segmentering av treningsbildet

Figur 2: Eksempler som viser gangen i segmenteringsprosessen. I originalbildet til venstre er det markert utvalgte regioner for tre klasser; himmel, hvite bygninger/gjerder og bakke mm. Bildet til høyre er segmenteringsresultatet, der de tre klassene er vist i henholdsvis blått, cyan og gult.

Det første skrittet i oppgaveløsingen er å velge et rektangulært område av passende størrelse (se fig. 2) fra hver av de klassene man ønsker å dele bildet inn i. Pikslene i hvert område skal brukes som treningssampler for den aktuelle klassen. Dere står fritt til å velge klassifikatortype og treningsmetode. Bruk gjerne en av klassifikatortypene (treningsmetodene) dere har implementert i prosjektoppgave 1 (f.eks. minimum-feilrate metoden) til å generere en klassifikator for de valgte klassene, basert på de tre fargekomponentene.

Bruk deretter klassifikatoren til å tilordne alle piksler i både treningsbildet og et lignende testbilde til en av klassene (i eksempelet i fig. 2 er klassifikatoren brukt til å segmentere treningsbildet), og konverter resultatet tilbake til et fargekodet resultatbilde som viser segmenteringen som er oppnådd. Kommentér resultatet. Ble segmenteringen slik som ønsket? Hvis ikke, hva kan årsaken være?

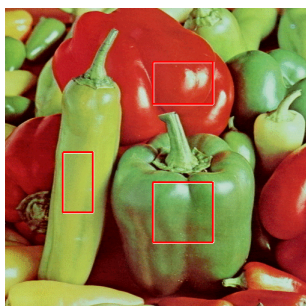
Eksempelbilder som kan brukes i oppgaven er vist i fig. 3 (vedlagt oppgaveteksten). Bilde 1 kan deles i tre segmenter ved å trene klassifikatoren på pikslene i tre regioner, som vist i bildet (denne markeringen er ikke med i det nedlastbare bildet), og deretter klassifisere alle piksler i bildet; dette for å teste at programvaren fungerer. De to andre bildene skal deles i tre segmenter, “rød ringperm”, “blå ringperm” og “bakgrunn”. Bruk gjerne bilde 2 (3b) som treningsbilde og

bilde 3 (3c) som testbilde.

For å gjøre klassifikatoren mer robust mht. variasjoner i intensitet (lys/skygge, reflekser osv.) kan det være en løsning å normalisere RGB-verdiene, f.eks. ved transformasjonen

$$t_1 = \frac{R}{R+G+B}, \quad t_2 = \frac{G}{R+G+B} \quad \text{og} \quad t_3 = \frac{B}{R+G+B},$$

og bruke to av disse verdiene (normaliserte tristimulusverdier), f.eks.  $t_1$  og  $t_2$ , som egenskaper. Legg merke til at  $t_3$  da er unødvendig siden den er bestemt av de to andre, og vil få minimum-feilrate klassifikatoren til å feile fordi determinanten til kovariansmatrisene blir null.



(a) Bilde 1 (utprøving)



(b) Bilde 2 (treningsbilde)



(c) Bilde 3 (testbilde)

Figur 3: Eksempler på bilder som kan brukes i oppgaven. Bildet til venstre kan f.eks. brukes til både trening og testing for å prøve ut metoden, mens de to andre kan brukes til hhv. trening og testing.

Bruk gjerne egne bilder i oppgaven; minimum to bilder med lignende innhold, der det ene brukes til trening og resten til testing.

Oppgaven kan utføres som et gruppearbeid med 2 - 3 studenter i gruppen. Det skal leveres en *rapport* (maksimalt ca. 10 sider), som beskriver hva som er gjort, hva som er oppnådd, og viser eksempler på resultater. Legg gjerne ved utskrift av kode, men dette er ikke noe krav.

### 3 Oppsummering

Gangen i prosessen er altså som følger:

- Velg et treningsbilde og finn egnede regioner for utvelging av treningssett for de ønskede klassene,
- Trén en klassifikator for problemet, f.eks. en minimum feilrateklassifikator med normalfordelingsantagelse,
- Bruk klassifikatoren til å segmentere treningsbildet, for å se om den fungerer som den skal,
- Segmentér ett eller flere testbilder,
- Gi en kortfattet drøfting av resultatene (hva fungerte godt/dårlig og hvorfor?).

Dere står som sagt fritt til å bruke andre bilder enn de vedlagte, og en hvilken som helst klassifikator type med tilhørende treningsmetode.