



NTNU

Kunnskap for en bedre verden

IMT3881 - VITENSKAPELIG PROGRAMMERING

Poisson Image Editing - Prosjektoppgave

Anders Kjelsrud
Casper F. Gulbrandsen
Kristian Jegerud

Sammendrag

Emnet IMT3881 Vitenskapelig Programmering kombinerer realfaglige teknikker og høynivå programmering, og gir studentene et innblikk i hvordan dette kan benyttes til å løse hverdagslige problemsstillinger. Et godt eksempel på dette er bildebehandling, hvor numeriske løsninger av differensiallikninger og ikkelineære algebraiske likninger sammen med biblioteker fra Python kan benyttes til å behandle bilder. Oppgaven vår har vært å implementere ulike anvendelser for dette, både for fargebilder og gråtonebilder. Vi har også skrevet et lite program med grafisk brukergrensesnitt (GUI) som viser eksempler på anvendelsene vi har implemntert. Ved å lese denne rapporten får du innsikt i hvordan vi har jobbet, hvilke teknikker som er brukt og hvordan disse er benyttet. All kildekoden ligger tilgjengelig på GitLab[1].

Våren, 2020

Innhold

Figurer	1
Tabeller	1
1 Introduksjon	1
1.1 Forord	1
1.2 Gruppens deltagere	1
1.3 Rapport	1
2 Anonymisering av bilder med ansikter	2
Bibliografi	3

Figurer

Tabeller

1 Introduksjon

1.1 Forord

Prosjektgruppen består av tre studenter fra dataingeniørstudiet ved NTNU Gjøvik. I løpet av semesteret har vi introdusert for matematiske teknikker for numeriske løsninger av bestemte integral, ordinære og systemer av differensiallikninger og ikkelineære algebraiske likninger. Vi har også fått lære om høynivå programmering for lineæralgebra, optimalisering, bildebehandling og maskinlæring. Oppgaven som ble gitt innebærte at vi måtte benytte oss av de vi har lært i løpet av semesteret. Selv om det har vært masse utfordringer underveis har vi fått masse praktisk læring.

1.2 Gruppens deltagere

Gruppen vår består av tre studenter på dataingeniørstudiet ved NTNU Gjøvik. Siden de to første årene på studiet utelukkende består av obligatoriske fag, har vi stort sett de samme kunnskapene og forutsetningene før prosjektets start. Uansett gjør det at vi er en gruppe på 3 at vi møter utfordringer med ulike tilnærminger, og har vært viktig for å løse problemstillingene på en best mulig måte.

Vi har ikke benyttet Python som hovedprogrammeringsspråk i emner vi har deltatt i før dette semesteret, men har før prosjektets start benyttet det i alle arbeidskrav i emnet. Dette gjør at vi har en viss kjennskap til hvordan Python fungerer og hvilke fordeler og ulemper dette medfører. For oss har nok den største utfordringer med Python vært hvordan vi skal strukturere et såpass stort prosjekt med flere forskjellige filtyper. Ingen av oss har vært borti utvikling av en GUI-applikasjon med Python og Qt som programmeringsspråk før. Dette har derfor gitt en bratt læringskurve og masse læring.

Rapporten er skrevet med LaTeX, som er et språk vi har brukt på et mindre prosjekt tidligere på studiet. Når vi har skrevet rapporten har det ikke gitt oss noen nevneverdige utfordringer. Det har derimot gitt oss utfordringer når rapporten måtte holdes lagret sammen med kildekoden i GitLab. Særlig når vi skulle finne en god editor for å jobbe på de ulike rapportfilene samtidig slet vi med å finne en løsning som vi synes var effektiv.

1.3 Rapport

Vi har valgt å følge rapportmalen til NTNU ***. Vi la malen inn i den nettbaserte latexeditoren overleaf, som vi tilslutt endte opp med å bruke til å skrive rapporten. Når vi i rapporten skal vise til andre kapitler og seksjoner vil det bli brukt kapitlets og seksjonens navn og nummer. For listformer som f.eks. punktlister vil vi bruke terminologien punkt. Vedlegger blir lagt bakerst i rapporten og vil bli referert til med vedleggets bokstav og navn. Rapporten har følgende oppbygning:

- 1 Introduksjon - beskriver raskt omstendighetene rundt prosjektet. Gir også en kort beskrivelse av gruppen og rapporten.
 - 2 Oppgaven - beskriver oppgaven som er gitt, hva som er forventet og hvordan det kan løses.
 - 3 - 11 Løste oppgaver - viser hvordan vi har løst oppgavene, hvordan vi har tenkt og hvilke forutsetninger som er gjort ++
 - 12 Avslutning - reflektesjoner rundt prosjektet blir gjort. Det blir også presentert diskusjoner og evaluering som gjort underveis.
 - 13 Konklusjon - inneholder avsluttende tanker, hva som fungerte og hva som kunne vært gjort annerledes.
- Bibliografi - inneholder alle kilder vi har benyttet i løpet av prosjektets periode.
- Vedlegg - inneholder kode som vi ønsket å forklare nærmere.

2 Anonymisering av bilder med ansikter

```
1 def blurFace(file, scaleFactor = 1.2, minNeighbors = 5):
2     """
3     Oppdager et ansikt vendt mot kamera
4
5     Parameters + some explanation
6     -----
7     file          : Bildefil
8                     Pathen til filen der original bildet befinner seg uten
9     andvending
10    scaleFactor   : int
11                    Kompenserer i tilfelle noen ansikter er lenger unna kamera enn
12    andre
13    minNeighbors  : int
14                    ----
15    title         : text
16                    Tittelen til bildet som er anvendt
17    """
18    image = cv2.imread(file) #leser inn bildet
19    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
20
21    #importerer haarcascade biblioteket
22    face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
23    faces = face_cascade.detectMultiScale(image, scaleFactor, minNeighbors, minSize
24    = (30,30))
25    for (x,y,w,h) in faces:
26        RoI = image[y:y+h, x:x+w]          #Region of Interest --> ansiktet
27        RoI = RoI.astype(dtype = float)
28        blur = eks.eksplisittAnonym(RoI,image[y:y+h, x:x+w], 0.25,250)
29        image[y:y+h, x:x+w] = blur
30
31    return len(faces), image
```

```
1 import AnonymiseringModul as anonym
2 import ImageView as iv
3
4 lena = '../hdr-bilder/faces/lena.png'
5 family = '../hdr-bilder/faces/family.jpg'
6 group = '../hdr-bilder/faces/group.jpg'
7
8 antall, image = anonym.blurFace(lena)
9 print(antall, "ansikt er registrert")    #teller opp antall ansikt funnet
10 iv.singleView(image, "Lena")             #displayer bildet
```



Bibliografi

- [1] Kjelsrud, Gulbrandsen, and Jegerud. Imt3881 2020 prosjekt. <https://git.gvk.idi.ntnu.no/casperfg/imt3881-2020-prosjekt/-/tree/master>, mars 2020.
- [2] Svein Linge and Hans Petter Langtangen. *Programming for Computations - Python*, volume 15 of *Texts in Computational Science and Engineering*. SpringerOpen, 2016.
- [3] Wikipedia contributors. Face detection — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Face_detection&oldid=947281932, 2020. [Accessed 22.04.2020].