



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 implementert. 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 Oppgaven	2
3 Glatting	2
4 Inpainting	2
5 Kontrastforsterkning	2
6 Demosaicing	2
7 Sømløs kloning	2
8 Konvertering av fargebilder til gråtone	2
9 Anonymisering av bilder med ansikter	2
9.1 Bakgrunn	2
9.2 Open CV	2
9.3 Haar Cascade	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 innebærer 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 LaTeX-editoren 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 - refleksjoner 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 Oppgaven**
 - 3 Glatting**
 - 4 Inpainting**
 - 5 Kontrastforsterkning**
 - 6 Demosaicing**
 - 7 Sømløs kloning**
 - 8 Konvertering av fargebilder til gråtone**
 - 9 Anonymisering av bilder med ansikter**

9.1 Bakgrunn

Anonymisering av bilder er enkelte ganger nødvendig før det blir offentlig vist frem for å skjerme den avbildedes personvern. En teknikk som ofte blir brukt er å gjøre ansiktene uskarpe og resten av bildet forblir skarpt. Teknikken avhenger av at programmet klarer å gjenkjenne ansiktene[2] for å definere en maske rundt, slik at man kan isolere anvendingen av bildet kun på ansiktet, og i likhet som tidligere er det flere teknikker som benyttes. I denne oppgaven brukte vi biblioteket OpenCV[3] (Open Source Computer Vision Library) og maskinlæringsalgoritmen Haar Cascade[4].

9.2 Open CV

Open CV er et open source datasyn¹ og maskinlæringsbibliotek som ifølge dem selv[3] ble utviklet for å skape et felles infrastruktur for datasynsapplikasjoner og for å akselerere bruken av maskinoppfatning² i kommersielle produkter. Kort fortalt betyr dette datamaskinens evne til å tolke data på en måte som ligner på måten mennesker bruker sansene til å oppfatte³ verden rundt seg[6], og da spesielt synet i dette tilfellet.

9.3 Haar Cascade

¹Computer Vision[5]

²Machine perception[6]

Bibliografi

- [1] Kjelsrud, Gulbrandsen, and Jegerud. Imt3881 2020 prosjekt. <https://git.gvk.idi.ntnu.no/casperfg/imt3881-2020-prosjekt/-/tree/master>, mars 2020.
- [2] Wikipedia contributors. Face detection — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Face_detection&oldid=947281932, 2020. [lesedato: 22.04.2020].
- [3] Open CV team. About open cv. <https://opencv.org/about/>, 2020. [lesedato: 24.04.2020].
- [4] Will Berger. Deep learning haar cascade explained. <http://www.willberger.org/cascade-haar-explained/>, 2018. [lesedato: 24.04.2020].
- [5] Jason Brownlee. A gentle introduction to computer vision. <https://machinelearningmastery.com/what-is-computer-vision/>, Mars 2019. [lesedato: 24.04.2020].
- [6] Wikipedia contributors. Machine perception — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Machine_perception&oldid=949933213, 2020. [lesedato: 24.04.2020].