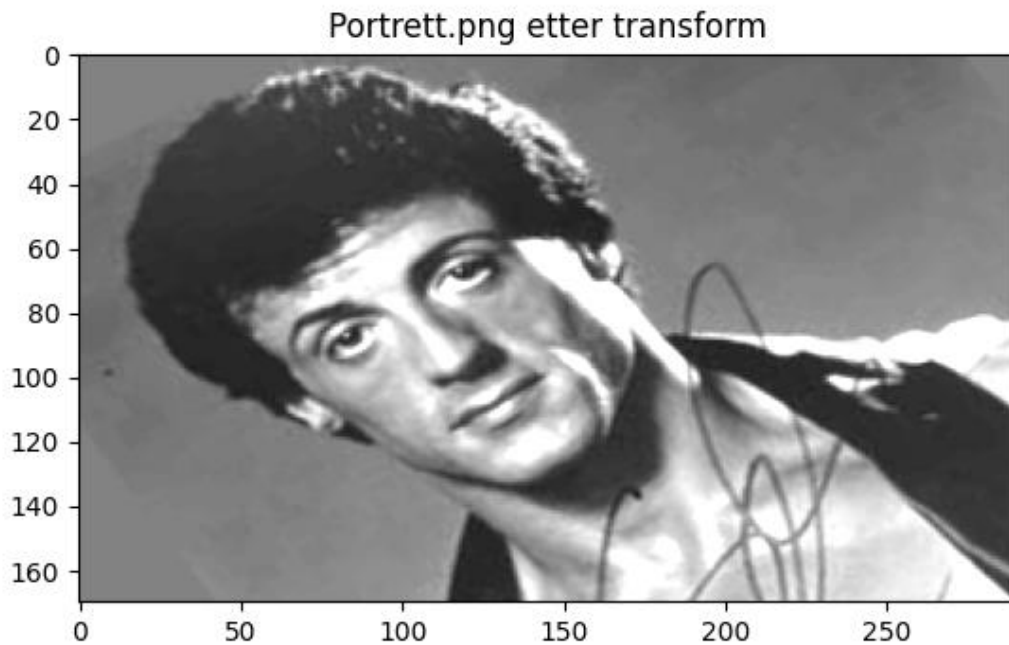


Oppgave 1

1.



2.

Koeffisientene ble funnet ved å sette opp ligninger for den affine transformasjonen der:

$$x' = a_0x + a_1y + a_2$$

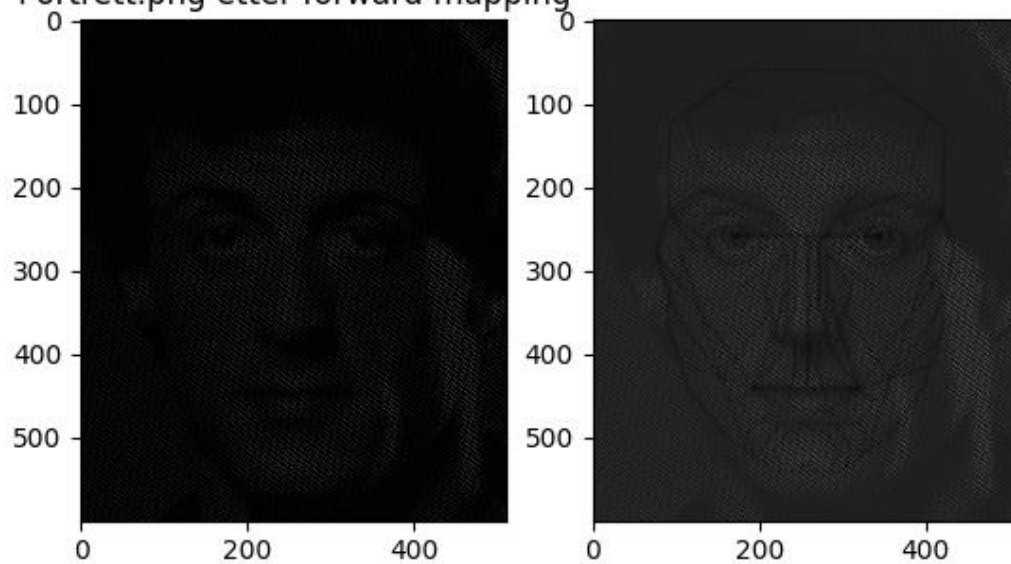
Og

$$y' = b_0x + b_1y + b_3$$

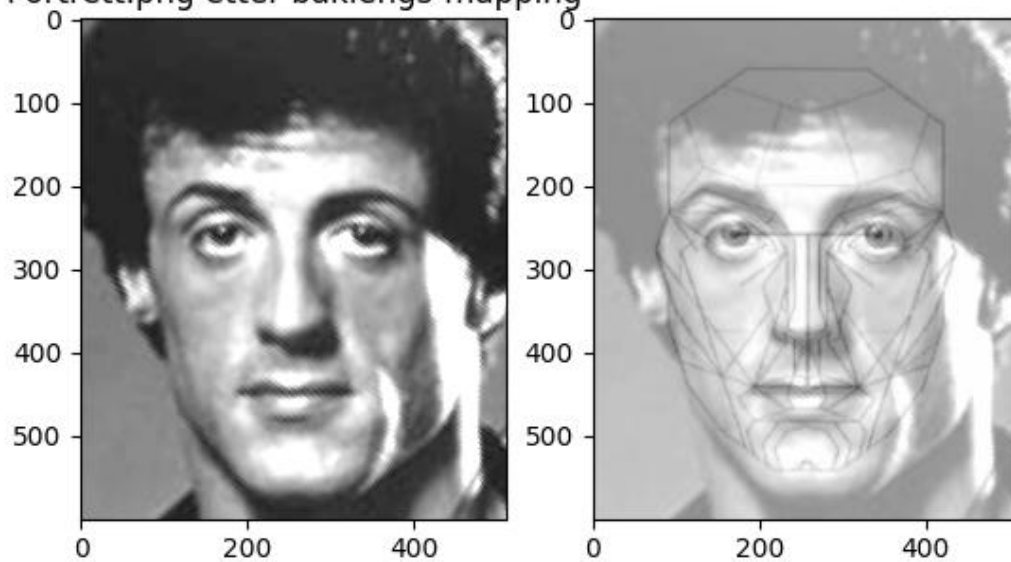
Videre ble det valgt tre punkter fra portrettet, venstre øyet, høyre øyet og munn, og tilsvarende tre punkter fra geometrimasken. Dette gir da seks ligninger for seks ukjente, som vil gi en affin transformasjon for samregistreringen.

3.

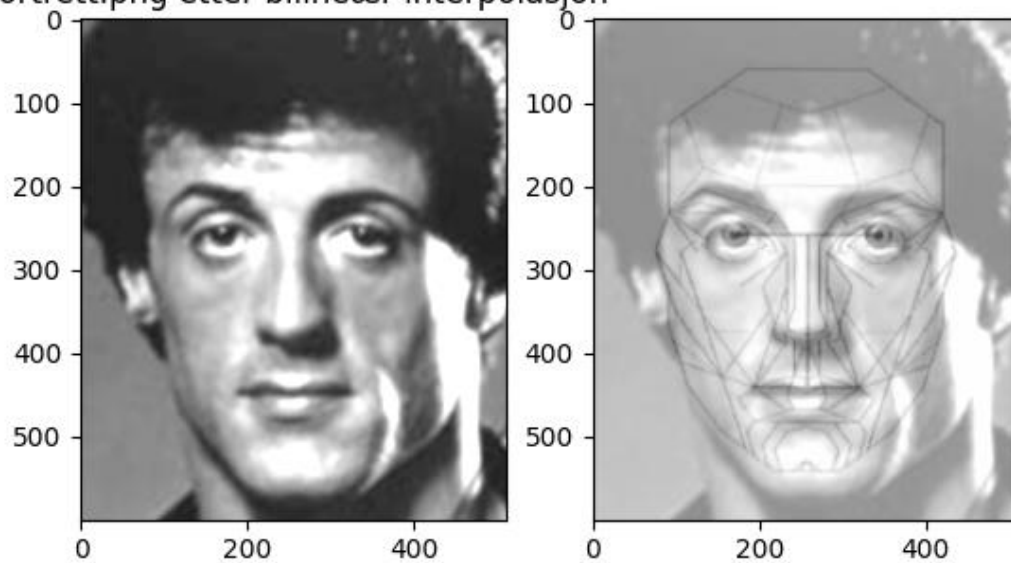
Portrett.png etter forward mapping



Portrett.png etter baklengs-mapping



Portrett.png etter bilineær interpolasjon



For forward mapping er det ikke garantert at alle pikslene i ut-bildet får en verdi, derfor er ut-bildet i forward mapping veldig mørkt, mens backward mapping garanterer at alle de transformerte pikslene får en verdi. Bildet for backward mapping blir derfor mye lysere og gir et bedre resultat. Forskjellen på nærmeste nabo og bilineære interpolasjonen ses i fargegraderingen av de to bildene. Her framstår backward mapping bildet som mer kornete enn for bilineær interpolasjon. Dette er fordi bilineær interpolasjon gir et kontinuerlig bilde.

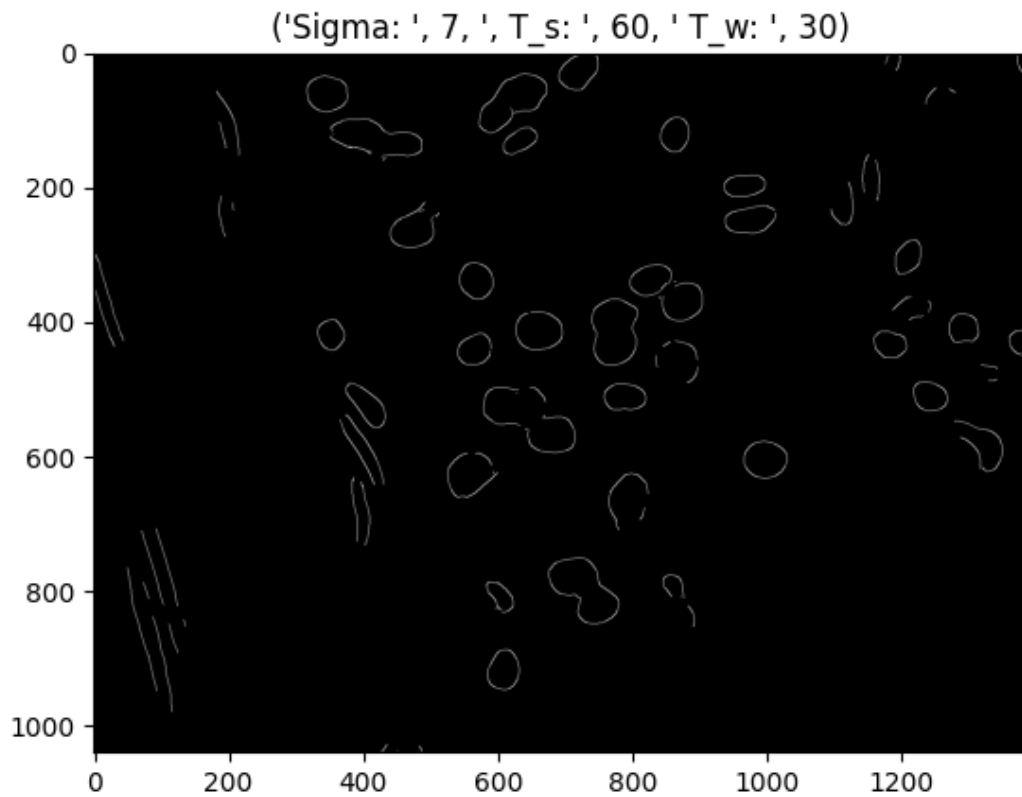
Oppgave 2

6.

Konvulsjonen er implementert i flere steg. Først roteres filteret 180 grader. Deretter kalles en metode «pad» som løser bilderandproblemet slik oppgaven beskriver det. «Pad» består også av to andre metoder, som tar for seg utvidelse av bilderanden for henholdsvis topp og bunn, og venstre og høyre side av bilderanden. Deretter hentes dimensjonene til filteret og bildet, slik at løkken som tar seg av beregningene av verdiene blir riktig. Beregningene skjer i en dobbel for-loop, og filteret ganges med bildet ved hjelp av list slicing som henter ut riktige indekser. Til slutt returneres det nye bildet.

Fro Cannys algoritme opprettes først et Gauss-filter. Filteret opprettes i en metode som tar inn parameteren sigma, og hver lengde av filteret settes til en pluss 8 ganger sigma rundet opp til nærmeste heltall. Videre sendes bildet sammen med Gauss-filteret inn i konvulsjonsmetoden, og resultatet returneres. Deretter beregnes gradient-magnitudo og gradient-retningen ved hjelp av sobel-filtre, og de to bildene returneres. For å tynne gradient-magnituden brukes den symmetriske 1D operatoren og det tynnede bildet returneres. Til slutt gjøres hysteresetterskling ved å opprette to kopier av bildet. Her tar en dobbel for-loop seg av indekseringen, og indeksen for hver av de to kopiene oppdateres til enten True eller False avhengig om verdien er større enn terskelen t_s eller mellom t_s og t_w . Etter indeksene er oppdatert gjøres en ny dobbel for-loop som bruker 8-tilkobling for å oppdatere ut-bildet slik at kun kantene får verdier.

7.



8.

Resultatbildet viser et bilde med ganske godt detekterte kanter. Bildestøy er bra redusert og kantlokaliseringen er god. En lav verdi for sigma fører til mer støy og det oppstår uønskede detekterte kanter. Verdien 7 for sigma fungerer godt for dette bildet. Videre er tersklene sentrale i å få frem detekterte kanter, og verdier $t_s = 60$ og $t_w = 30$ gir et bra resultat. Ved høyere terskler vil fler kanter ikke bli detektert som sterke nok.