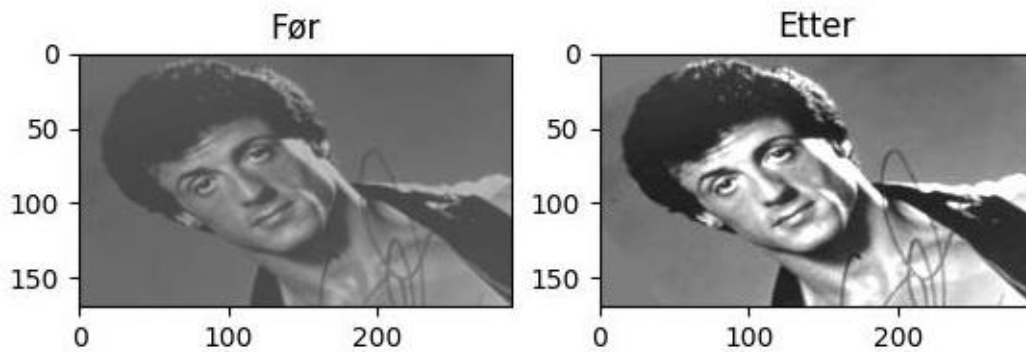


Oppgave 1)

1. Mellom-resultat-bildet etter gråtonetransformen

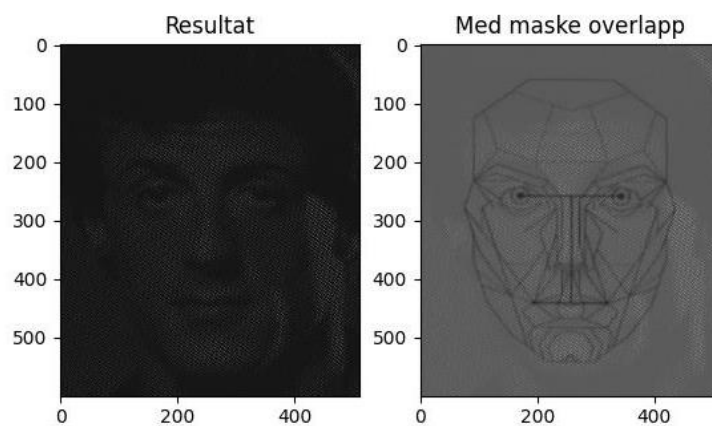


2. Brukte denne formelen fra forelesningsnotatene for å finne koeffisientene til den affine transformasjonen. Satt inn tall fra portrett.png der øynene og munnen var, og satt inn hvor de skulle være i forhold til geometrimaske.png og brukte `np.linalg.solve` for å løse ligningen.

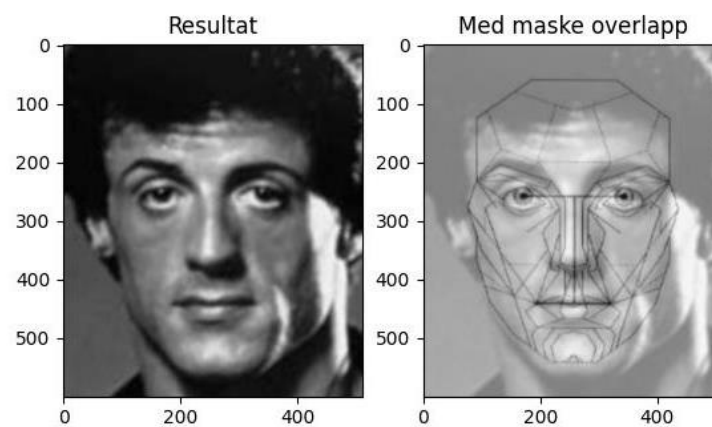
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & a_2 \\ b_0 & b_1 & b_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

3.

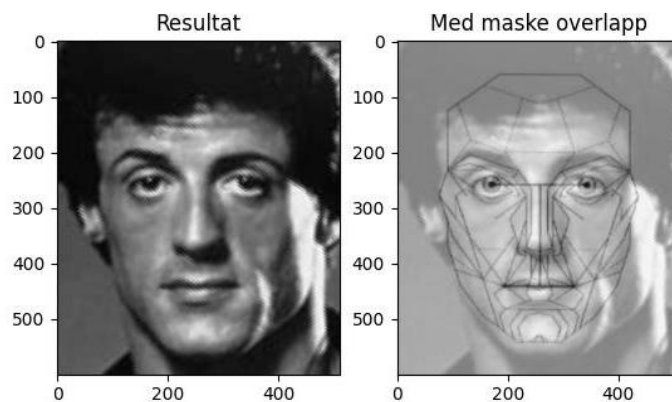
Forlengstransformasjon



Baklengstransformasjon med bilineær interpolasjon



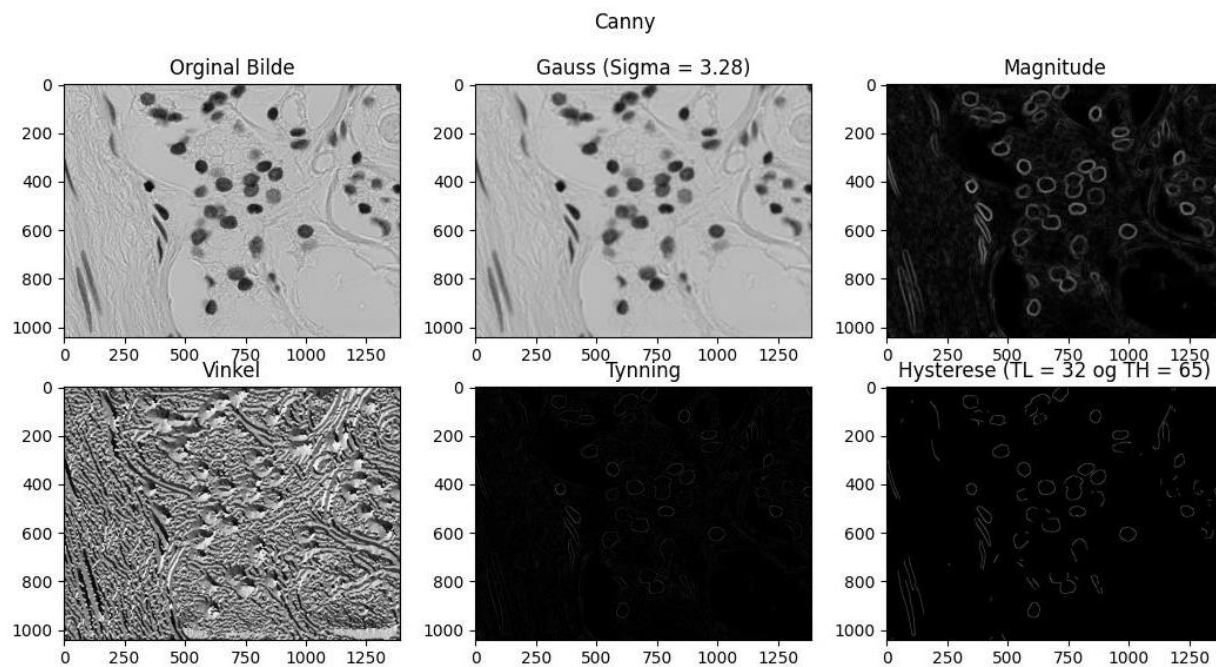
Baklengstransformasjon med nærmeste nabo interpolasjon



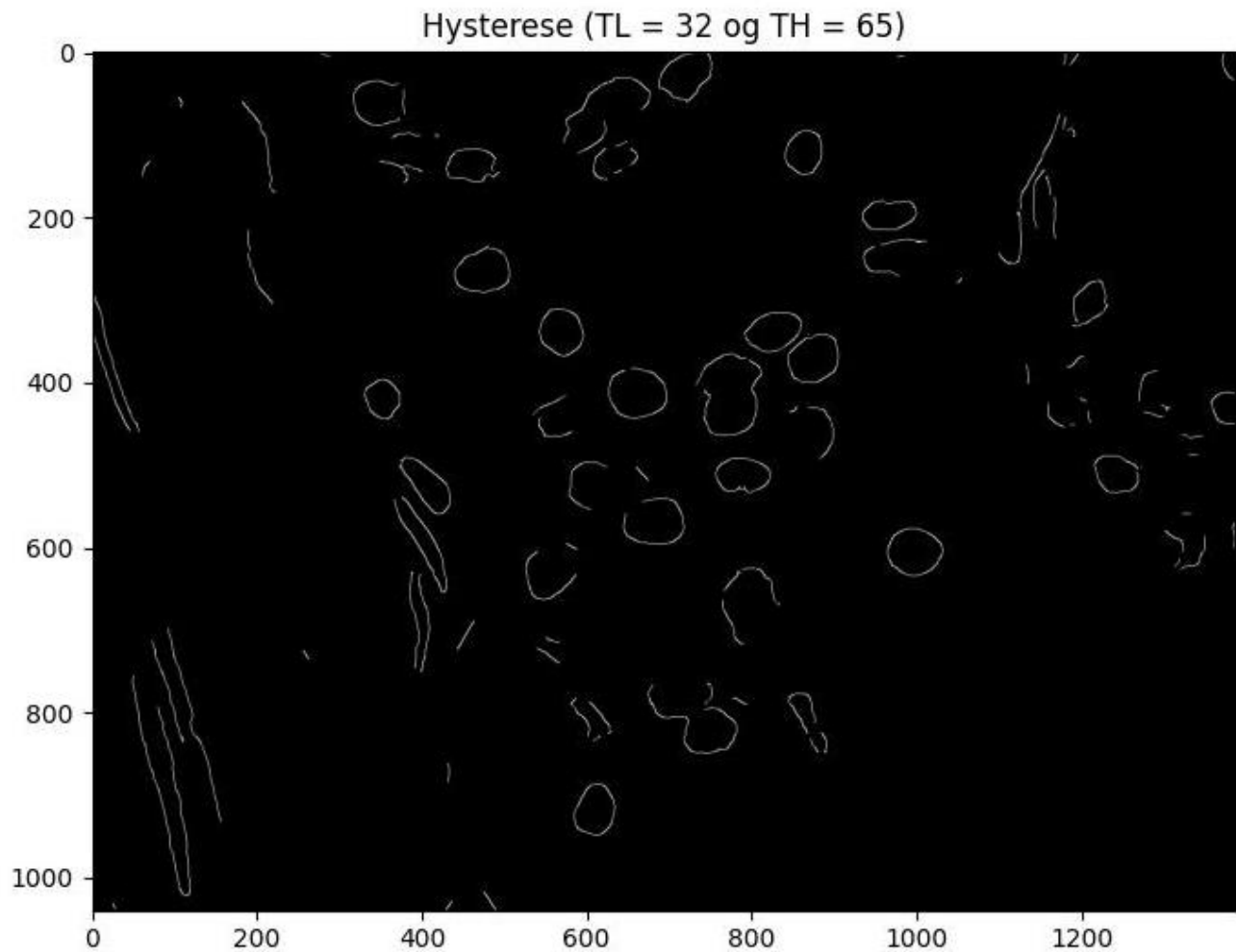
4. Veldig tydelig på resultat bildene at baklengs transformasjon er best. I forlengs transformasjonen mister vi mange piksler og bildet blir mye mørkere og mangelfullt. Alle metodene klarer å rotere bildet til riktig plassering.
5. Python-kode leveres i filen **oppgave1.py**.

Oppgave 2)

6. Lager en metode som padder/utvider inn-bildet med nærmeste pikselverdi. Så lager jeg en metode som konvulerer bildet med et gitt filter, før jeg returnerer det nye bildet uten paddingen.
Lager en metode som lager et Gauss-filter, før jeg konvulerer inn-bildet med dette Gauss-filteret. Dette filtrerte bildet bruker jeg videre for å finne gradient og magnituden til bildet. Videre bruker jeg magnituden og gradienten til å tynne bildet. Det tynnede bilde sender jeg videre til hysteresetærskling som gir meg det endelige resultat-bildet.
7. Bruker $\sigma = 3.28$, $T_L = 32$ og $T_H = 65$.



Resultatbildet:



8. Synes resultatbildet ble ganske greit. Er noen kanter som mangler, men overall vil jeg si det er greit. Eksperimenterte en del med sigma verdien og fant ut at jo lavere verdi, jo mere støy ble det. Fant ut at 3.28 funket bra for meg. Prøvde meg litt fram med T_L og T_H , men det var litt vanskeligere å eksperimentere godt med siden det er to verdier. Endte opp med $T_L = 32$ og $T_H = 65$.
9. Python-kode leveres i filen **oppgave2.py**. Denne koden tar veldig lang tid å kjøre, så jeg har fått tips fra medstudenter og «Jitte» koden. Dette gjør jeg ved å importere numba og skrive «@jit» over metoder som jeg vet bruker lang tid. For å få numba på pc-en må man skrive «pip install numba» i terminalen. Håper dette går fint, har prøvd å kjøre koden uten å «Jitte» og da bruker den ekstremt lang tid.