

### Oppgave 1

For å utføre den affine transformasjonen fant jeg punktene i masken (øyne, nese, munn etc.) manuelt på portrett.png. Brukte disse med formelen fra forelesningsnotatene til å finne transformasjonsmatrisen. Tror standardavviket og/eller middelværdi må være feil på en måte, men resultatet blir fortsatt brukbart.

Forlengs-mapping har åpenbart noen begrensninger i forhold til baklengs-mapping. Forlengs-mappingen lager mange hull man ellers ikke får, og vi får da et resultat med mange svarte piksler. Begge resultatbildene hvor baklengs mapping anvendes er veldig like og vi kan se at hullene fra forlengs-mappingen ikke er til stede.

### Oppgave 2

Her implementeres konvolusjonen, som tar bildet vi ønsker å prosessere og et filter som parametre. I funksjonen utvides bildet med numpy.pad og filteret roteres med 180 grader. Videre itererer vi over alle piksler i bildet og summerer med filterets verdi.

Så implementeres et gaussfilter med sigma som parameter. Funksjonen gir oss sidelengdene, summerer alle vektene og setter  $A/\text{sum}$ . Så følger funksjonen gradientmagnitud(bilde). Funksjonen finner  $g_x$  og  $g_y$ . Fra dette finner vi gradientmagnituden og gradientvinkelen. Dette sendes inn i kanttynning( $\text{gradientM}$ ,  $\text{gradientV}$ ) som parameter, som tar bildet av magnituden og sammenligner det med bildet av gradientvinkelen. Denne funksjonen finner hvor kantene går og fjerner unødvendige kantpiksler fra bildet. Til slutt sendes bildet med tynnede kanter inn i hysteresis-funksjonen som også tar variablene  $T_l$  og  $T_h$ , som settes manuelt. Piksler med verdier over  $T_h$  settes til maksverdi og piksler under  $T_l$  settes til 0. Dette gir oss et bilde hvor svake/utydelige piksler blir bort og gir et tydelig bilde av kantene:



Resultatbildet gir et tydelig bilde på hvor kantene i bildet befinner seg. Dette fordi algoritmen vi har benyttet reduserer støy og finner kantene i bildet. Over ser vi en sigma på 5, Th på 70 og TI på 50.