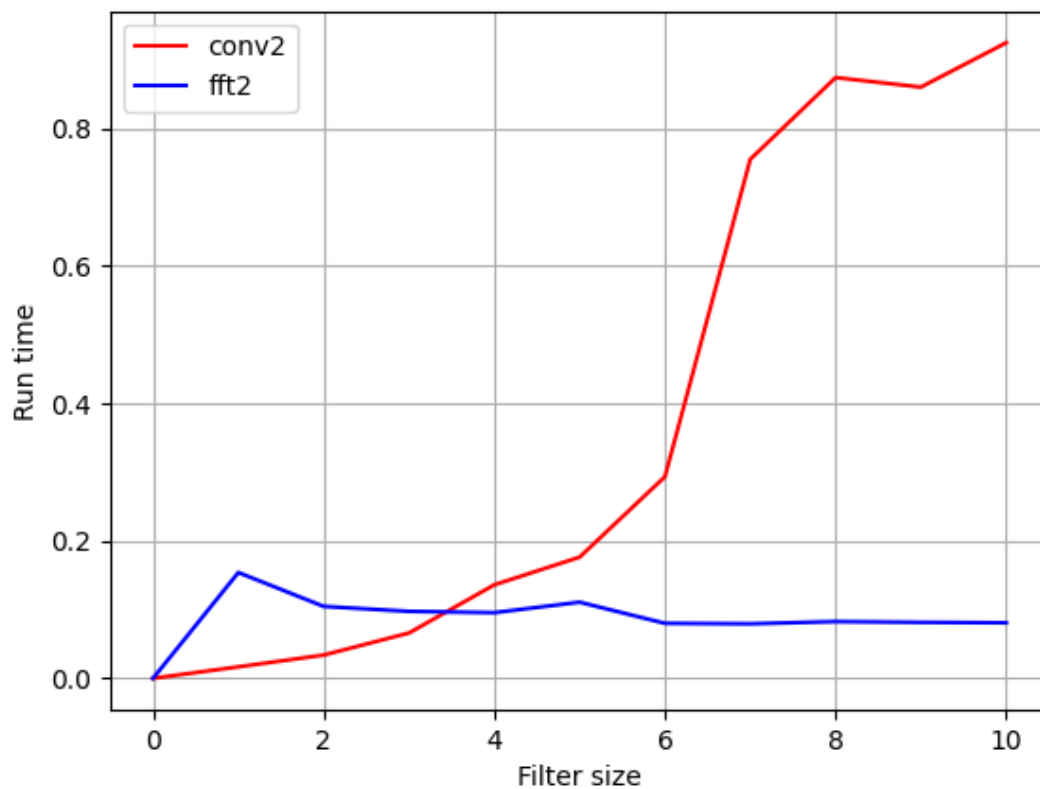


Oppg 1

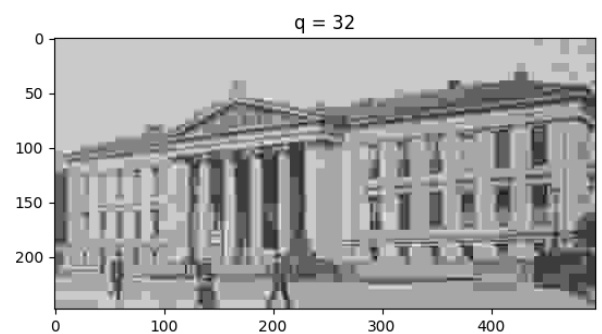
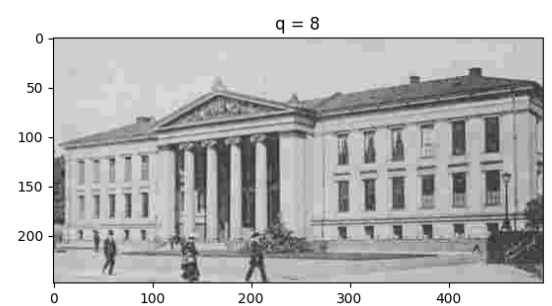
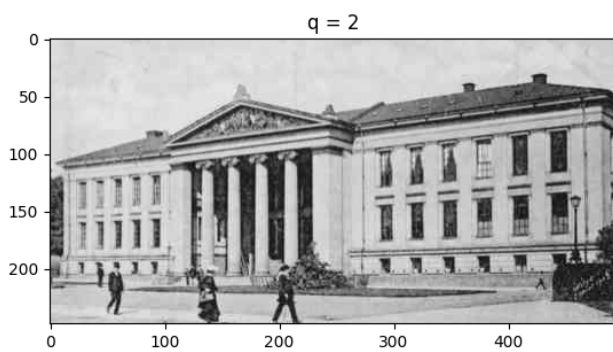
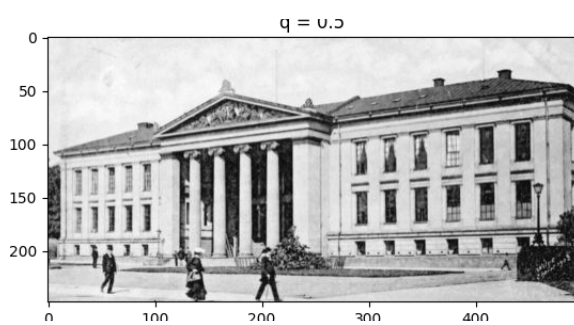
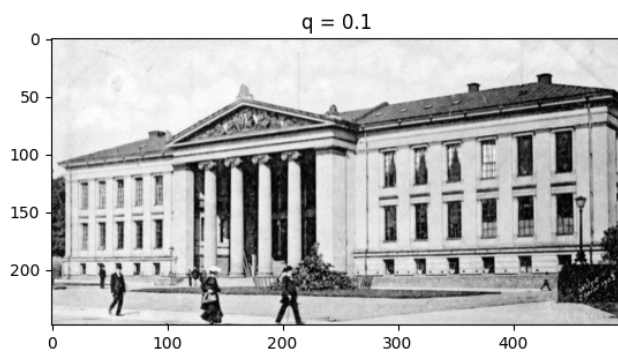
Ved vanlig konvulsjon vil vi få en ramme av sorte og grå piksler i alle sidene av bilde, såkalt bilderandsproblem. Bilderandsproblemet oppstår når origo av filter er på piksler helt ved kanten. Da vil verdier utenfor bilde være 0, og konvulsjonen vil gi veldig lave verdier der. Denne ramme fjernes ved å bruke «same» som parameter i conv2. Jeg vet ikke helt hvordan de løser bilderandsproblemet, men muligens nærmeste nabo slik vi gjorde i oblig 1.

Trikset ved å nullutvide filteret i fft2-funksjonen gjør det samme som koden vår gjør, men bare integrert i fft2-funksjonen. Den lager altså et filter i en gitt størrelse, og fyller resterende piksler med 0.



Oppg2

Originalbilde



Oppg 2

Rekonstruksjonsfeilene kan detekteres av mitt blotte øye med $q = 2$ uten å zoome inn. Da blir bildet mer kornete i fallet. Dette forsterkes desto høyere q blir.

Jeg synes rekonstruksjonen bildet er god nok til å vise med $q = 0.5$, og til nøds $q = 2$. ved $q = 0.5$ er det så godt som ingen synlig forandring, mens det på $q = 2$ er mulig å se at bilde er komprimert. Selv om $q = 0.5$ høres lite ut, gir det 72% mindre lagringsplass, og er da veldig effektiv.

Når vi kjører DCT-transformasjonen deler vi hver 8×8 -blokk på $q \cdot Q$. Desto høyere q , desto lavere tall kommer ut av DCT-transformasjonen. Dermed vil flere pixelverdier være like, og det vil være muligheter for å komprimere bilde med kompresjonsalgoritmer.