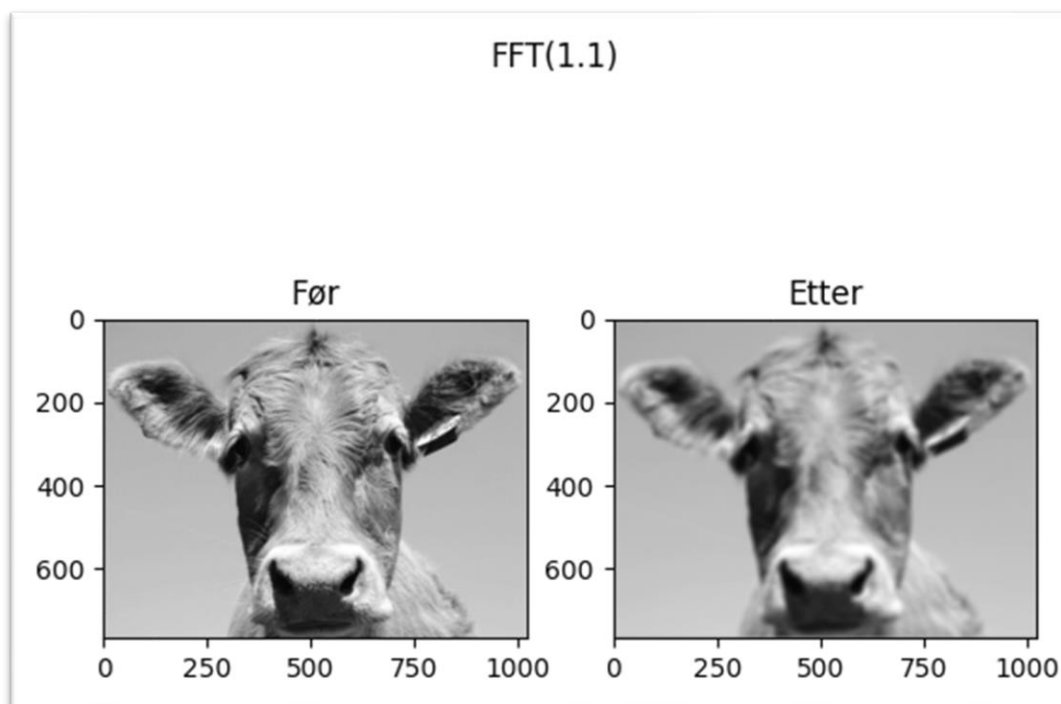
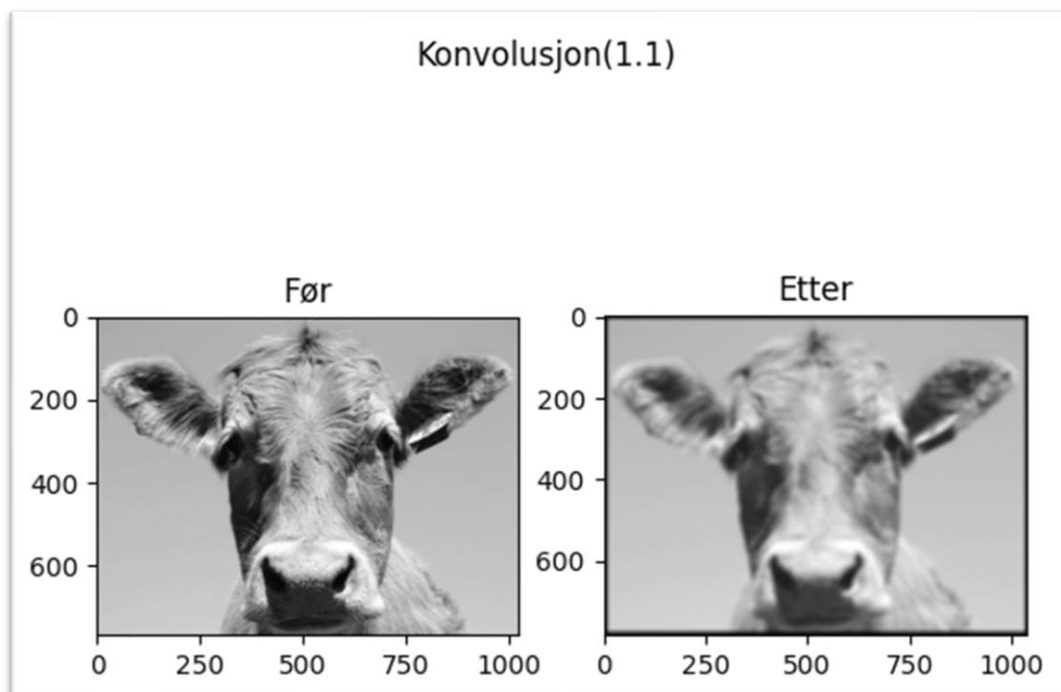
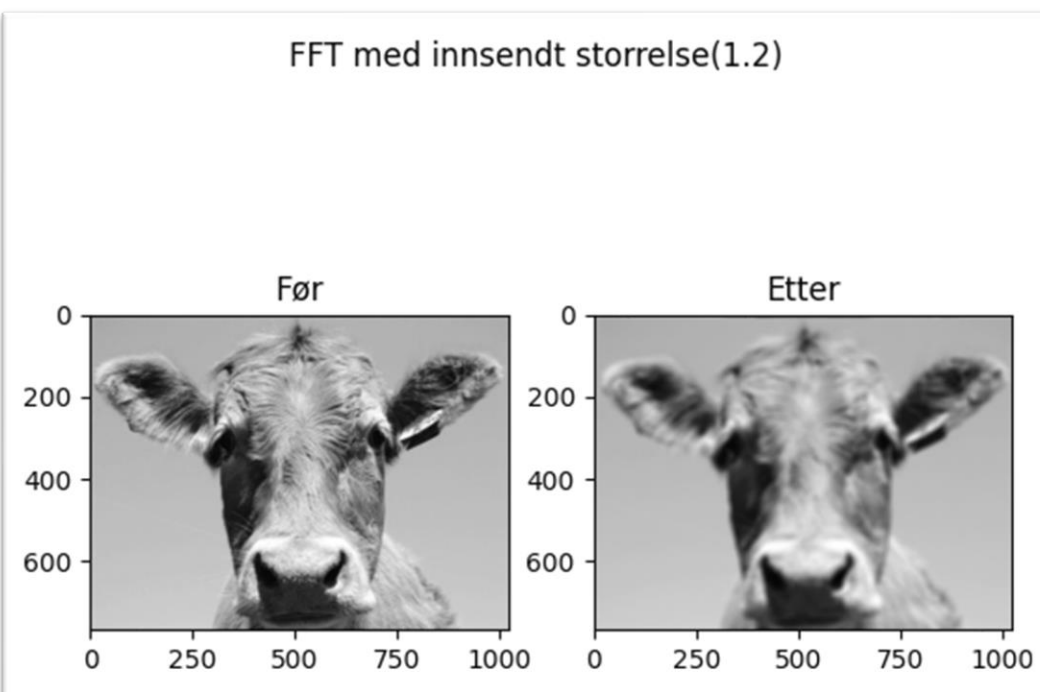
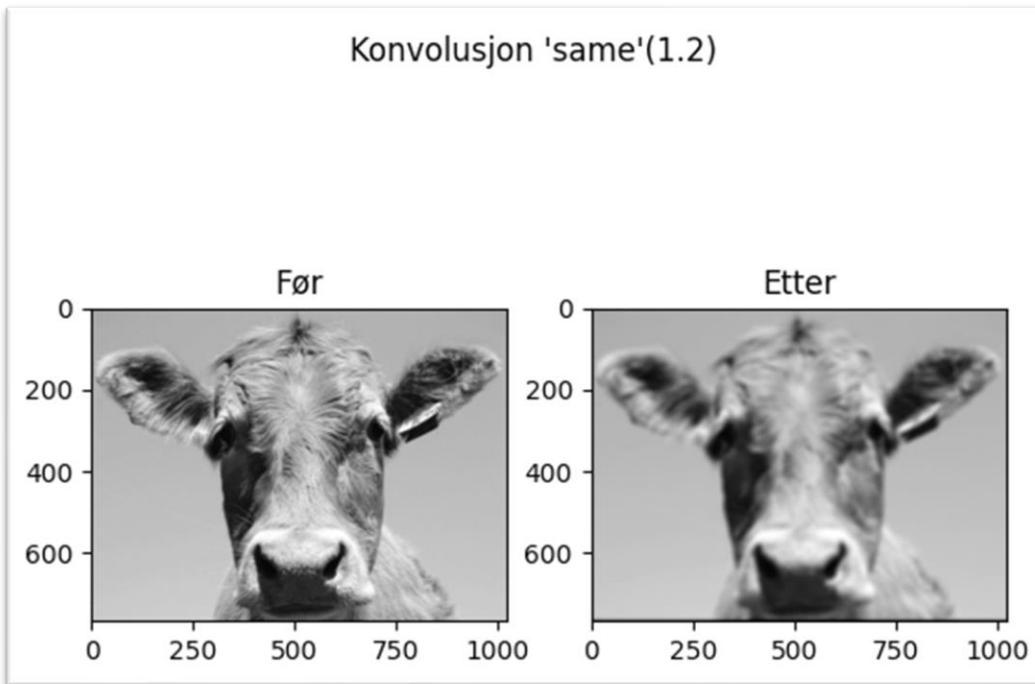


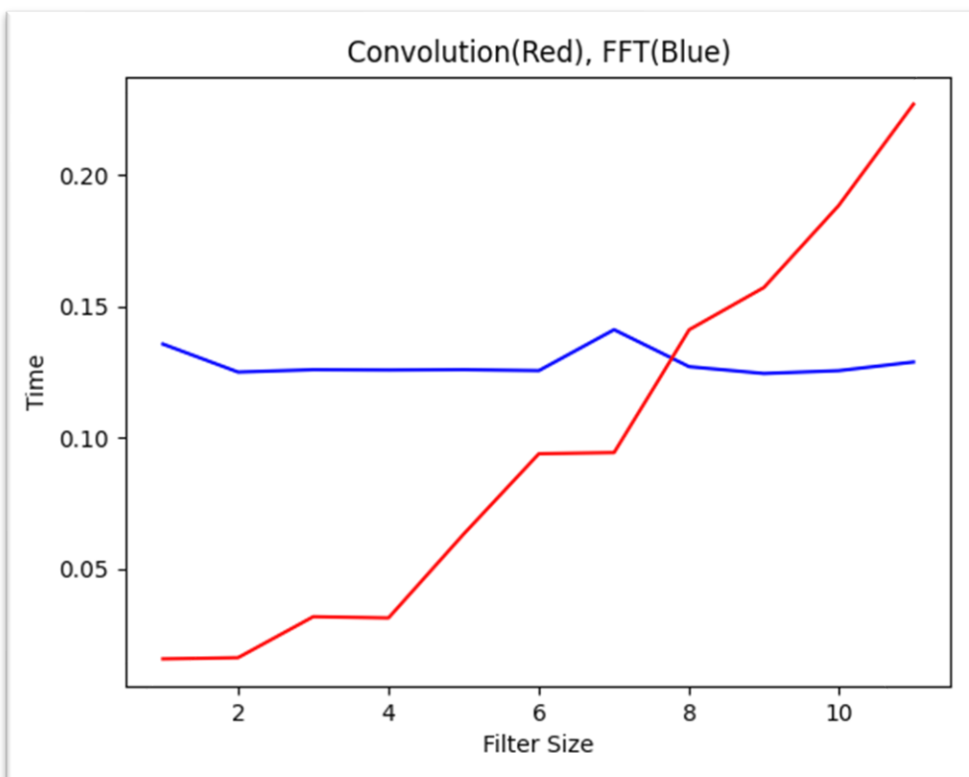
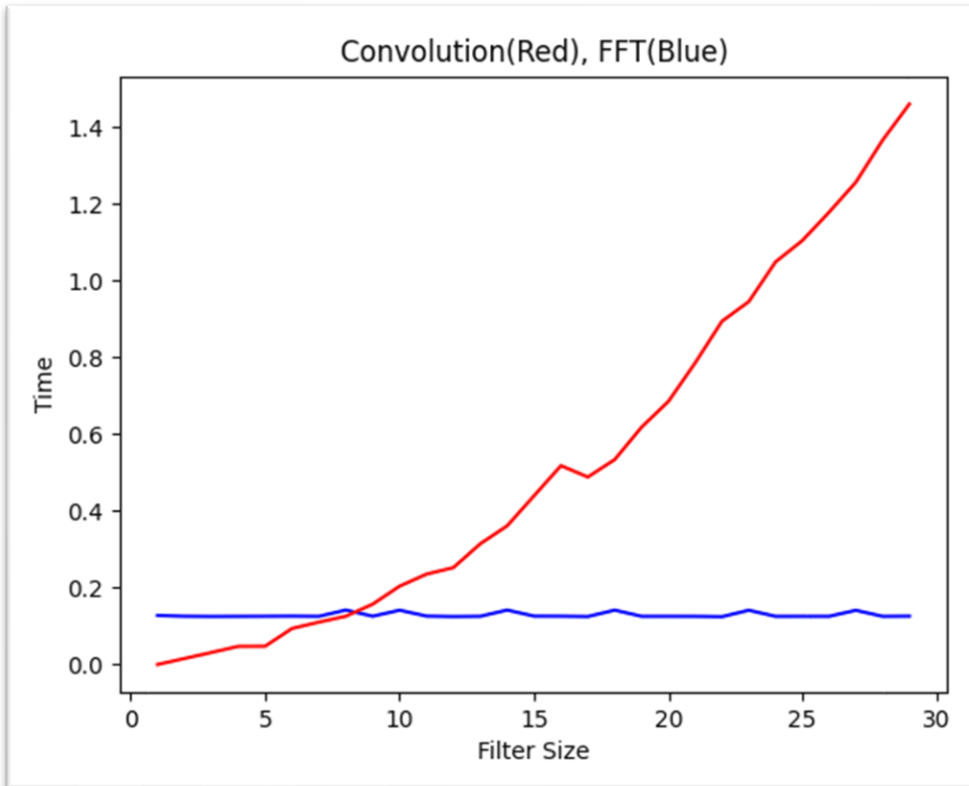
Oppgave 1**Oppgave 1.1**

Oppgave 1.2

1. På det frekvensdomenefiltrerte bildet er det en liten forflytting av bildet. Litt av bunnen av bildet har kommet helt i toppen av bildet. Dette kommer av at fft er en periodisk funksjon som repeterer seg selv.
2. Det blir en forskjell i kanten til det filtrerte bilde når vi bruker 'same', fordi 'same' gjør at bilde blir krympet tilbake til originalstørrelsen etter at det har blitt utvidet.



Oppgave 1.3



- På det øverste bildet går filterstørrelsen opp til 30 og på det nederste bildet går den til 12. Her ser man tydelig at konvolusjon er raskere i starten, men blir fort veldig treg. Når man bruker FFT tar det ca. samme tid uavhengig av filterstørrelsen. Når filterstørrelsen er ca. 8 eller høyere er FFT raskere enn konvolusjon.

Programmet er litt tregt og tar ca. 24 sekunder på min pc.

Oppgave 2

- De rekonstruerte bildene av **uio.png**. **H** er entropien og **CR** er kompresjonsraten. Lagringsplassen som er oppgitt er en estimering ut ifra entropien og er ikke den faktiske lagringsplassen bildet opptar. Bildene blir også vist og lagret til fil når man kjører programmet **oppgave2.py**. Dette programmet tar ca. 6 sekunder på min pc.

Ikke-tapsfri JPEG-kompresjon

Original Bilde

$H = 7.464$

Lagring = 114.766KB



$q = 0.1$

$H = 3.879$

$CR = 1.92$

Lagring = 59.65KB



$q = 0.5$

$H = 2.107$

$CR = 3.54$

Lagring = 32.393KB



$q = 2$

$H = 1.088$

$CR = 6.86$

Lagring = 16.733KB



$q = 8$

$H = 0.49$

$CR = 15.24$

Lagring = 7.53KB



$q = 32$

$H = 0.18$

$CR = 41.54$

Lagring = 2.763KB



2. Drøfting

- a. Rekonstruksjonen er ganske god. Der i bildet det er detaljer, holder de seg godt, mens der det er stort sett en farge(himmelen), forsvinner detaljene fort og alt blir en farge. Man ser tydelig at Blokk-artefaktene øker og blir mer synlig når q øker. Jeg ser ikke noen tegn til ringinger eller glatting.
- b. Jeg vil si at man for alle q -ene kan se hva det er bildet av, og at $q = 2$ absolutt er bra nok. Når $q = 8$, ser man at fargen blir litt blussere og man mister en del detaljer, men fortsatt mulig å se hva det er bilde av. Når $q = 32$ forsvinner de fleste detaljene, men man kan absolutt gjenkjenne bildet hvis man har sett det før.
- c. Når q øker, blir verdiene i Q større. Dette resulterer i at alle pikslene i bildet vårt blir delt på et større tall som resulterer i en lavere pikselverdi som vi da runder av til nærmeste heltall. Når man deler på et høyere og høyere tall vil svaret konvergere mot 0. Dette gjør at mange av verdiene får samme verdi, og gjør at kompresjonsraten øker.