Oblig 2 IN2070

Mohamed Omar Atteyeh

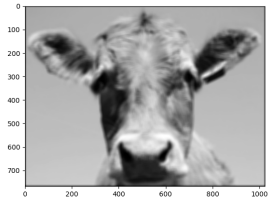
May 10, 2021

Oppgave 1

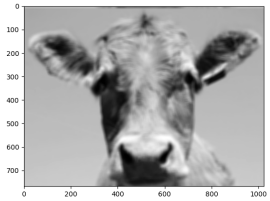
1.2

For at størelsen ikke skal bli feil, er vi nødt til å utvide bildetranden med noe padding for å ikke miste informasjon. De forskjellene i kantene oppstår på grunn av padding og interaksjon mellom filtret og kantene når man transformere tilbake. En translasjon i frekvensdomenet er også et resultat av interaksjonen mellom filtret og bildet.

(a) Original bildet



(b) Romlig Transformasjonen

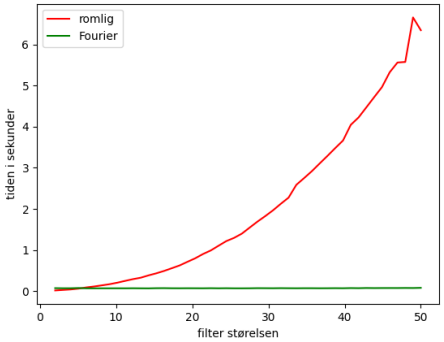


(c) Fourier transformasjonen

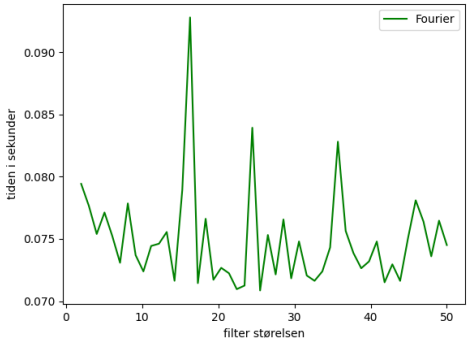
Man ser en padding på det romlig bildet, mens det er en type rand eller miskonfigurasjon på det øverste på fourier domenet-blidet

1.3

Når det gjelder hastigheten har vi en klar vinner på hvilken som er raskest, og fourier vinner. Fourier er generelt så mye raskere, fordi vi bruker elementvis multiplikasjon mellom innbildet og filtret. Vi multipliserer element mot element i matrisene for bildet og filtret, og dermed uansett hvor stor filteren er for vi cirka samme tid. En annen ting man ser også at for 50 forskjellige filter størelser er tiden det tar fourier transformasjon gjennomsnitlig på et sekund. Man kan ikke sammeligne de to fordi plottet viser forholdet mellom de to, og det viser som om fourier transformasjonene er nesten like null. Jeg legger til en annen plot som viser bare fourier transformasjonen for 50 forskjellige filter størelser i figuren nede :



(a) Fourier vs Konvolusjon



(b) Tidene for fourier transformasjonen

oppgave 2

Kompresjonsraten er et estimat på hvor mye vi komprimerer et bildet, eller en sammenligning mellom hvor mye informasjon som det nye utbildet innholder sammelignet med original bildet. Entropi er et estimat som sier til oss hvor kompakt bildet kan lagres. Entropi er også en nedre grense, det vil si vi trenger entropi verdien omskrevet som en bit for å vite den minste biten vi kan lagre bildet i. Det er bare i det tilfelle hvor hver piksel er like sannsynlig, men i det tilfelle hvor alle pikslene er like da har du en entropi som er like 0 siden sansynligheten er like for alle pikslene. Man kan tenke på det som

at sannsynlighet som beregnes forteller oss fordellingen for alle pikslene, og den variasjonen i pikslene beregnes sånn at vi ser hva vi har mest av. De forskjellene gjør da at når vi komprimerer bildet veit vi hva vi mister, ved bruk av de endrignene i sannsynlighetene. Ved å beregne entropiene etter første filter og etter andre filtret og sammen lignet det med entropiene for det originale bildet får vi det følgende tabell:

q values	første filter	andre filter	Originale entropiet
0.1	1.7864942523023468	7.445571245810708	7.464485296306298
0.5	1.7283123234228588	7.3760221525176775	7.464485296306298
2	1.116342077598067	7.130744950646402	7.464485296306298
8	0.5194170134135934	6.1392659611595315	7.464485296306298
32	0.2175511971765904	4.53719790425819	7.464485296306298

Vi kan beregne kompresjonsraten ved bruk av følgende likningen

$$CR = \frac{b}{c}$$

der b er den faste antall biter i den ukomprimerte datamengden, og c er gjennomsnittlige antall biter per symbol i den komprimerte datamengden. For å beregne hvor sparsomme vi er for plass vi bruker

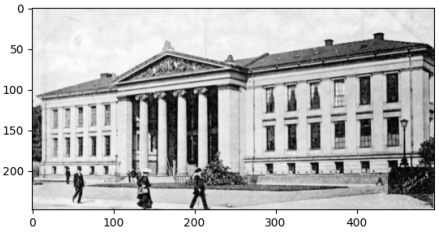
$$R = 1 - \frac{1}{CR} = 1 - \frac{c}{b}$$

. Følgende tabell viser de forskjellige verdiene for kompressjonsraten og redundans for de forskjellige q verdiene:

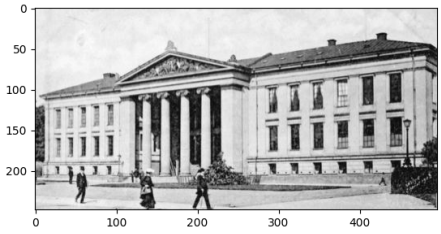
q values	Kompresjonsraten	Redundans i prosent
0.1	8	87.5
0.5	8	87.5
2	8	87.5
8	∞	100
32	∞	100

Fra de verdiene jeg får for kompressjon og redundanse vi får uendelig for q = 8 og q = 32, fordi deres entropi er rundt av til 0, og da blir kompresjonsraten delt på null og da går hele kompressjon mot ∞, det betyr at hver piksel innholder nesten ingen informasjon. I redundanse får vi 100 prosent

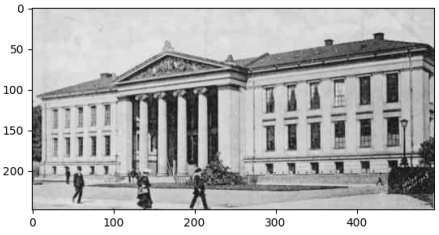
på begge to, som betyr at vi kan fjerne alt uten å miste relevant informasjon. For q verdiene 0.1 , 0.5 og 2 er de verdiene for kompressions raten og redundanse helt like. Dermed i prinsipp, mister vi like mye informasjon om vi bruker hvilken som helst av de tre. Dermed kan vi velge høyere verdier for q og få like mye plasssparing, derfor hadde jeg valgt q = 2. En annen ting er at entropien for q = 2 som ikke er rundt av er lavere enn både q = 0.1 og q = 0.5. Så lavere entropi, men like mye lagrningplass. Der q = 2 filteren gjør at vi ikke mister informasjon i kompressions prosessen, men der høyere q verdi vi har det dårligere oppløsning får vi i bildet. Men hadde vi ikke rundet av hadde vi fått en proposjonal forhold mellom q verdiene og kompresjonsraten. Det gir mening siden filter bruker av q gir oss dårligere oppløsning i bildet som betyr mindre informasjon, som da øker kompresjonsraten siden filtret resulterer til mindre informasjon.



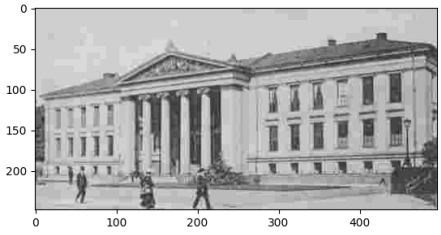
(a) q = 0.1



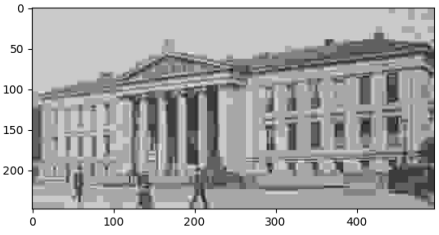
(b) q = 0.5



(c) q = 2



(d) q = 8



(e) q = 32