Bildebehandling med Python og EzGraphics

I denne oppgaven skal dere jobbe med bildebehandling. På samme måte som vi jobbet med lyd tidligere, skal vi nå se på bilder. Vi kan bruke EzGraphics til alt vi trenger i disse oppgavene. Hvis du ikke får EzGraphics til å virke på din maskin, kan du legge EzGraphics-fila i samme mappe som python-fila.

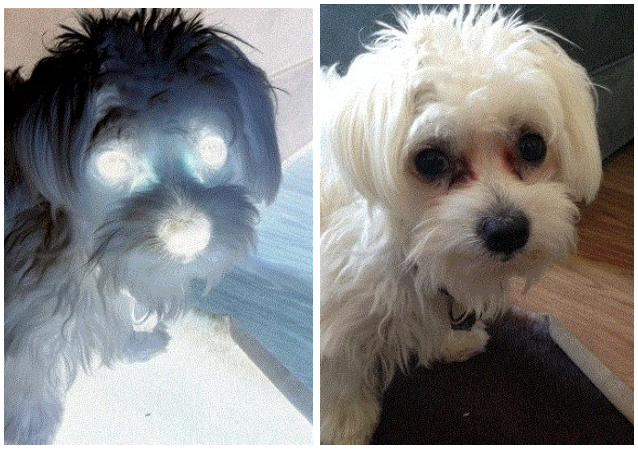
Du finner EzGraphics-fila i ei mappe på Fredagspython-sida.

Du kan lese fra side 212 til side 217, men vi gjentar det viktigste her:

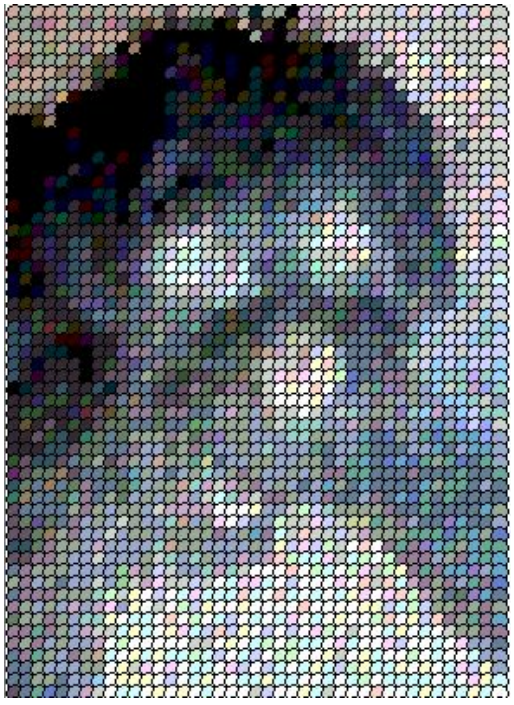
```
#bildeanalyse

#Her må vi importere GraphicsImage i tillegg til den vanlige GraphicsWindow.
from ezgraphics import GraphicsWindow,GraphicsImage

#Vi velger et filnavn. Dette er navnet på fila vi skal lese inn.
filename = "hund.gif"

#Oppretter en variabel som programmet kan bruke
image = GraphicsImage(filename)

#Vi har to metoder for å hente ut høyden og bredden til et bilde
width = image.width()
height = image.height()

#Vi går igjennom ei todimensjonal liste
#metodene getRed(row,col), osv,  henter fargen til en piksel på et gitt sted.

for row in range(height):
    for col in range(width):
        red = image.getRed(row,col)
        green = image.getGreen(row,col)
        blue = image.getBlue(row,col)
        #for å invertere trekker vi den gamle verdien fra 255:
        newRed = 255 - red
        newGreen = 255 - green
        newBlue = 255 - blue
        #setter ny piksel:
        image.setPixel(row, col, newRed, newGreen,newBlue)



#Vi lager et vindu og et lerret:
win = GraphicsWindow()
canvas = win.canvas()
#Her tegner vi bildet vi har laget

canvas.drawImage(image)
#venter på brukeren
win.wait()
```

```
#Vi kan også lagre bildet:
image.save("redigert_" + filename)
```

Her er et eksempel på bilde før(høyre) og etter(venstre) invertering.



En annen morsom ting dere kan gjøre er å lage overføre verdier fra bildet til et canvas, slik som vi er vant til. Hvis vi for eksempel tegner hver syvende piksel som en stor sirkel fylt med fargen til pikselen, får vi følgende bilde på lerretet. Merk at dette kan vi ikke lagre (men vi kan ta skjermdump!)



Noen oppgaver:
-Snu bildet opp ned
-Snu bildet 180 grader.

-gjør bildet svart-hvitt

Utvalgte oppgaver fra boka:
Oppgave P4.46:
Implement a “sunset” effect by increasing the red level of each pixeø of an image by 30 percent (up to a value of 255).

Oppgave P4.47
Add black vertical stripes to an image, each spaced five pixels from the next. The partial image below is enlarged to show the lines.

Oppgave P4.48 Add black diagonal stripes to an image, each spaced five pixels apart. The partial image below is enlarged to show the lines.

Oppgave P4.49
Rotate an image 90 degrees clockwise.

Oppgave P4.50
Repeat an image four times.

Oppgave P.4.51
Replicate each pixel of an image four times, so that the image is twice as arge and wide as the original, with “blocky” pixels. The partial image below is enlarged to show the pixels.

Oppgave P4.53
Detect edges in an image, coloring each pixel black or white depending on whether it is significantly different from the neighbouring pixel to the east, south and south-east. Average the red, green and blue components of the three neighbours. The compute

distance = $|red - red_{neighbours}| + |green - green_{neighbours}| + |green - red_{neighbours}|$
If that distance is < 30, color the pixel black, Otherwise, colour it white.

Fjerne støy i bilde

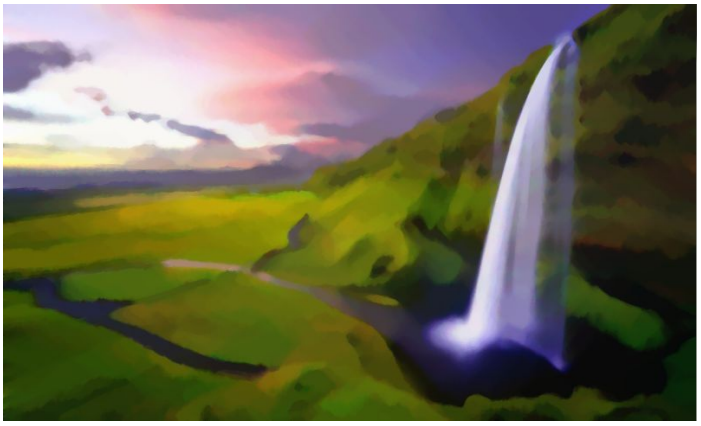
Et vanlig problem med bilder er det er støy (pixler som skiller seg veldig ut), og datamaskiner blir ofte brukt til å fjerne slik støy fra f. eks bilder fra overvåkingskameraer. Dette blir typisk gjort før man prøver å gjenkjenne objekter i bildet. Det finnes både veldig enkle og svært avanserte måter å fjerne støy på. Ideen bak alle er å forsøke å fjerne elementer fra bildet som er unaturlige. Ofte er dette finkornet støy, i form av enkelte pixler som skiller seg ut fra pixlene rundt.

Oppgave
Målet i denne oppgaven er å fjerne støy og skape et best mulig bilde ut av dette dette bildet (last ned fra http://folk.uio.no/ivargry/landscape_noisy_small.gif).

Skriv kode som går gjennom hver pixel i bildet og forsøker å avgjøre om pixel-verdien er støy eller ikke. Dette kan typisk gjøres ved å sammenligne verdien med pixlene rundt. Hvis pixelen er støy, så kan du endre pixel-verdien. Du kan ta utgangspunkt i følgende kode:

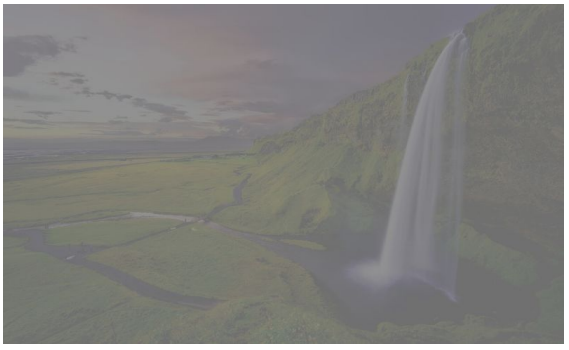
```
from ezgraphics import GraphicsWindow,GraphicsImage, ImageWindow
from statistics import median

image = GraphicsImage("landscape_noisy_small.gif")
width = image.width()
height = image.height()

def hent_pixelfarge(bilde, x, y, farge):
  if farge == "blue":
    return bilde.getBlue(y, x)
  elif farge == "red":
    return bilde.getRed(y, x)
  else:
    return bilde.getGreen(y, x)

def fjern_noise(bilde, farge, x, y):
  verdier_rundt = []
  for i in range(x-2, x+3):
    for j in range(y-2, y+3):
      if i < 0 or i >= width or j < 0 or j >= height:
        continue
      if i != x and j != y:
        verdier_rundt.append(hent_pixelfarge(bilde, i, j, farge))

  return int(median(verdier_rundt))

for x in range(width):
  for y in range(height):
    image.setPixel(y, x, (fjern_noise(image, "red", x, y), fjern_noise(image, "green", x, y), fjern_noise(image,
"blue", x, y)))

win = GraphicsWindow(width, height)
canvas = win.canvas()
canvas.drawImage(image)
win.wait()
```

Bildet under er laget ved å kjøre denne koden med en veldig enkel løsning der hver pixel alltid settes til gjennomsnittet av pixlene rundt. Det gjør at all støyen blir borte, men i tillegg blir også bildet veldig “smoothet”. Prøv å få til en bedre løsning enn dette!



Øke kontrast i bilde

Målet i denne oppgaven er å øke kontrasten i bildet under og få et så bra resultat som mulig. Last ned bildet herifra: http://folk.uio.no/ivargry/landscape_low_contrast.gif



Bilder med lav kontrast har gjerne pixelverdier som skiller seg lite fra hverandre. Vi ønsker å skape større skille mellom verdiene, f. eks ved å gjøre små verdier enda lavere og store verdier større.

Denne kjernekoden prøver på en veldig enkel måte å gjøre bildet skarpere:

```
from ezgraphics import GraphicsWindow,GraphicsImage, ImageWindow

image = GraphicsImage("landscape_low_contrast.gif")
width = image.width()
height = image.height()

def hent_pixler(bilde, x, y):
  red = bilde.getRed(y, x)
  green = bilde.getGreen(y, x)
```

```
    blue = bilde.getBlue(y, x)
    return (red, green, blue)

def mer_kontrast(pixelverdi):
    if pixelverdi < 128:
        return int(pixelverdi / 1.1)
    else:
        return int(256 - (256 - pixelverdi)/1.1)
    return ny_verdi

for x in range(width):
    for y in range(height):
        red, green, blue = hent_pixler(image, x, y)
        image.setPixel(y, x, (mer_kontrast(red), mer_kontrast(green), mer_kontrast(blue)))


win = GraphicsWindow(width, height)
canvas = win.canvas()
canvas.drawImage(image)
win.wait()
```

Funksjonen “mer_kontrast” tar her en pixel-verdi og deler verdien på 1.1 hvis den er under 128 (altså gjør den lavere) og gjør det omvendt hvis den er over 128 (gjør den høyere). Resultatet blir ikke så bra, men det gir et skarpere bilde:

Oppgave

Ta utgangspunkt i koden over, og skriv din egen funksjon for å øke kontrasten i bildet. Prøv å få et skarpt bilde som ser naturlig ut. Med en god funksjon bør resultatet kunne bli noe sånt som dette: