

Tkinter

December 6, 2018

Catalin Stoean
catalin.stoean@inf.ucv.ro
<http://inf.ucv.ro/~cstoean>

1 Citim o imagine si ii aplicam ponderare in GUI folosind Tkinter

```
In [22]: from tkinter import *
         from PIL import Image
         from PIL import ImageTk
         import tkinter.filedialog
         import cv2

         # Stabilim dimensiunile etichetelor cu poze in functie de
         # lungimea dorita pentru eticheta si de dimensiunile pozei
         def stabilesteInaltimea(lungimePanou, lungimePoza, inaltimePoza):
             inaltimePanou = (lungimePanou * inaltimePoza) / lungimePoza
             return int(inaltimePanou)

         # Metoda ce urmeaza sa fie apelata de catre butonul de incarcare a pozei
         def selectareImagine():
             '''
             Punem global pentru cele doua variabile de mai jos pentru a putea
             modifica valorile lor in cadrul acestei metode.
             '''

             global labelA, labelB
             # citim poza de la o locatie selectata de utilizator
             cale = tkinter.filedialog.askopenfilename()

             if len(cale) > 0: #daca avem o cale
                 poza = cv2.imread(cale)
                 '''
                 Citim h si w de la poza.
                 Stabilim lungimea etichetei.
                 In functie de acestea calculam inaltimea etichetei.
                 Redimensionam apoi poza in functie de w si h ale etichetei.
                 '''
                 h, w, _ = poza.shape
```

```

lungimePanou = 300
inaltimePanou = stabilesteInaltimea(lungimePanou, w, h)
poza = cv2.resize(poza, (lungimePanou, inaltimePanou))
gri = cv2.cvtColor(poza, cv2.COLOR_BGR2GRAY)
_,binara = cv2.threshold(gri, 127, 255, cv2.THRESH_BINARY)

#Transformam poza in RGB pentru PIL
poza = cv2.cvtColor(poza, cv2.COLOR_BGR2RGB)

# Trecem poza in PIL
poza = Image.fromarray(poza)
binara = Image.fromarray(binara)

# Transformam poza in format tkinter
poza = ImageTk.PhotoImage(poza)
binara = ImageTk.PhotoImage(binara)

# Daca nu este nimic incarcat in etichete
if labelA is None or labelB is None:
    # Punem imaginea originala in prima eticheta (stanga)

    labelA = Label(image = poza, width = lungimePanou,
                    height = inaltimePanou)
    labelA.image = poza
    #padx si pady adauga pixeli pe orizontala si verticala langa obiect
    labelA.pack(side="left", padx=10, pady=10)

    # Imaginea ponderata va fi pusa in a doua eticheta
    labelB = Label(image=binara, width = lungimePanou,
                    height = inaltimePanou)
    labelB.image = binara
    labelB.pack(side="right", padx=10, pady=10)
# Daca avem deja poze in etichete, trebuie sa folosim "configure"
else:
    # actualizam etichetele
    labelA.configure(image = poza)
    labelB.configure(image = binara)
    labelA.image = poza
    labelB.image = binara

return

root = Tk()
root.title('Ponderare binara')
labelA = None
labelB = None
# Butonul care apeleaza metoda "selectareImagine"
btn = Button(root, text="Incarca o imagine", command=selectareImagine)

```

```

#adaugam butonul in partea de jos a ferestrei
btn.pack(side="bottom", fill="both", expand="yes", padx="10", pady="10")

# Lansam interfata
root.mainloop()

```

2 Adaugam si un slider

```

In [21]: from tkinter import *
        from PIL import Image
        from PIL import ImageTk
        import tkinter.filedialog
        import cv2

# Stabilim dimensiunile etichetelor cu poze in functie de
# lungimea dorita pentru eticheta si de dimensiunile pozei
def stabilesteInaltimea(lungimePanou, lungimePoza, inaltimePoza):
    inaltimePanou = (lungimePanou * inaltimePoza) / lungimePoza
    return int(inaltimePanou)

# Metoda ce urmeaza sa fie apelata de catre butonul de incarcare a pozei
def selectareImagine():
    '''
    Punem global pentru cele doua variabile de mai jos pentru a putea
    modifica valorile lor in cadrul acestei metode.
    '''
    global labelA, labelB, gri, binara
    # citim poza de la o locatie selectata de utilizator
    cale = tkinter.filedialog.askopenfilename()

    if len(cale) > 0: #daca avem o cale
        poza = cv2.imread(cale)
        '''
        Citim h si w de la poza.
        Stabilim lungimea etichetei.
        In functie de acestea calculam inaltimea etichetei.
        Redimensionam apoi poza in functie de w si h ale etichetei.
        '''
        h, w, _ = poza.shape
        lungimePanou = 300
        inaltimePanou = stabilesteInaltimea(lungimePanou, w, h)
        poza = cv2.resize(poza, (lungimePanou, inaltimePanou))
        gri = cv2.cvtColor(poza, cv2.COLOR_BGR2GRAY)
        _,binara = cv2.threshold(gri, s.get(), 255, cv2.THRESH_BINARY)

        #Transformam poza in RGB pentru PIL
        poza = cv2.cvtColor(poza, cv2.COLOR_BGR2RGB)

```

```

# Trecem poza in PIL
poza = Image.fromarray(poza)
binara = Image.fromarray(binara)

# Transformam poza in format tkinter
poza = ImageTk.PhotoImage(poza)
binara = ImageTk.PhotoImage(binara)

# Daca nu este nimic incarcat in etichete
if labelA is None or labelB is None:
    # Punem imaginea originala in prima eticheta (stanga)

    labelA = Label(image = poza, width = lungimePanou,
                    height = inaltimePanou)
    labelA.image = poza
    labelA.pack(side="left", padx=10, pady=10)

    # Imaginea ponderata va fi pusa in a doua eticheta
    labelB = Label(image=binara, width = lungimePanou,
                    height = inaltimePanou)
    labelB.image = binara
    labelB.pack(side="right", padx=10, pady=10)
    # Daca avem deja poze in etichete, trebuie sa folosim "configure"
else:
    labelA.configure(image = poza)
    labelB.configure(image = binara)
    labelA.image = poza
    labelB.image = binara

return

# Metoda ce urmeaza sa fie apelata de catre slider
def schimbaPrag(val):

    global labelB, gri
    if labelB == None: #Nu am citit inca poza initiala daca nu avem nimic in labelB
        pass
    else: #Recalculam imaginea binara si o punem in eticheta B
        _,binara = cv2.threshold(gri, int(val), 255, cv2.THRESH_BINARY)

        binara = Image.fromarray(binara)

        binara = ImageTk.PhotoImage(binara)
        labelB.configure(image = binara)
        labelB.image = binara

    return

root = Tk()

```

```

root.title('Ponderare binara')
labelA = None
labelB = None
gri = None # imaginea gri este globala ca sa o folosim in schimbaPrag
# Butonul care apeleaza metoda "selectareImagine"
btn = Button(root, text="Incarca o imagine", command = selectareImagine)
# adaugam butonul in partea de jos a ferestrei
btn.pack(side="bottom", fill="both", expand="no", padx="10", pady="10")
# adaugam un slider si pentru acesta o metoda
s = Scale(root, from_=0, to=255, orient=HORIZONTAL, command = schimbaPrag)
s.set(100)
s.pack()
# Lansam interfata
root.mainloop()

```

Un tutorial cu informatii succinte si exemple relativ la interfete in Python cu Tkinter se gaseste aici: https://www.python-course.eu/python_tkinter.php

3 Exerciții

1. Faceti un program cu GUI care sa citeasca o poza si sa contina butoane radio care sa permita selectarea din mai multe tipuri de ponderare. Rezultatul este afisat intr-o a doua eticheta.
2. Realizati un program cu GUI care sa citeasca o poza si o culoare, apoi sa produca o imagine binara care sa transforme toti pixelii din imaginea initiala care sunt apropiati sub un anumit prag de culoarea selectata in alb, iar pe ceilalti in negru. Pragul se stabileste printr-un slider.

Capturi de ecran

