**Assignment UCS547**

**The functionality is as follows**

The object does not cover the face initially

When the face is clicked the object starts covering the face

When the object is clicked it stops following the face and can be dragged and moved anywhere

Note:2 files paths have to be changed **face\_classifier** and **object** which are present at lines 8 and 10

*# KUNAL NARANG 102003058*

*from* tkinter *import* \*

*from* PIL *import* Image, ImageTk

*import* cv2

*# Create an instance of TKinter Window or frame*

face\_classifier = cv2.CascadeClassifier('F:\Edge ai\class 4\haarcascade\_frontalface\_default.xml')

object = cv2.imread("F:\Edge ai\images\img1.png")

x\_cord = 0

y\_cord = 0

clicked = 0

width = 0

height = 0

mouse\_x = 0

mouse\_y = 0

faces = []

win = Tk()

win.geometry("700x350")

label =Label(win)

label.place(*x*= 0, *y*= 0)

objlabel =Label(win)

cap= cv2.VideoCapture(0)

def changeMouseCord(*event*):

    global mouse\_x, mouse\_y

    mouse\_x,mouse\_y = *event*.x,*event*.y

def moveImage(*event*):

    global mouse\_x, mouse\_y,x\_cord,y\_cord

*# print(event.x -mouse\_x,event.y-mouse\_y)*

    difx,dify = *event*.x -mouse\_x,*event*.y-mouse\_y

    x\_cord ,y\_cord = x\_cord+difx,y\_cord+dify

    objlabel.place(*x*= x\_cord, *y*= y\_cord)

def stopFaceDetection(*event*):

    global clicked,mouse\_x,mouse\_y

    clicked+=1

    mouse\_x = *event*.x

    mouse\_y = *event*.y

    print(mouse\_x,mouse\_y)

def showImage(*event*):

    global clicked

    global faces

*for* (x,y,w,h) *in* faces:

*if*(*event*.x>x and *event*.x<x+w and *event*.y>y and *event*.y<y+h):

            clicked+=1

def changeCord(*x*,*y*):

    global x\_cord, y\_cord

    x\_cord=*x*

    y\_cord=*y*

def setDimension(*w*,*h*):

    global width,height

    width = *w*

    height = *h*

def show\_frames():

   global faces

   frame = cap.read()[1]

   gray = cv2.cvtColor(frame , cv2.COLOR\_BGR2GRAY)

   faces = face\_classifier.detectMultiScale(gray, 1.3, 5)

*# Get the latest frame and convert into Image*

   frameInvert= cv2.cvtColor(frame,cv2.COLOR\_BGR2RGB)

   frameimg = Image.fromarray(frameInvert)

*# Convert image to PhotoImage*

   frameimgtk = ImageTk.PhotoImage(*image* = frameimg)

   label.imgtk = frameimgtk

   label.configure(*image*=frameimgtk)

*if*(clicked < 2):

*for* (x,y,w,h) *in* faces:

                resizedObj = cv2.resize(object,(w,h))

                objectInvert= cv2.cvtColor(resizedObj,cv2.COLOR\_BGR2RGB)

                objectimg = Image.fromarray(objectInvert)

                objlabel.place(*x*= 0, *y*= 0)

*if*(clicked >= 1):

                    objectimgtk = ImageTk.PhotoImage(*image* = objectimg)

                    objlabel.imgtk = objectimgtk

                    objlabel.configure(*image*=objectimgtk)

                    objlabel.place(*x*= x, *y*= y)

                changeCord(x,y)

                setDimension(w,h)

   label.after(30, show\_frames)

objlabel.bind('<B1-Motion>',moveImage)

objlabel.bind('<ButtonRelease-1>',changeMouseCord)

label.bind('<Button-1>', showImage)

objlabel.bind('<Button-1>', stopFaceDetection)

show\_frames()

win.mainloop()







