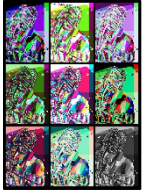


Maddie Villella

Completed



Originals



```
def collage():
    pict=makePicture(getMediaPath("picture.jpg"))
    copy1=makeEmptyPicture(232,320)
    factor=0.167
    #scales original picture down by a factor of 0.167
    scale(pict,copy1,factor)

    copy2=duplicatePicture(copy1)
    copy3=duplicatePicture(copy1)
    copy4=duplicatePicture(copy1)
    copy5=duplicatePicture(copy1)

    canvas=makeEmptyPicture(699, 968)           #creates empty canvas to put collage on

    #draws border around the original picture in the collage
    drawBorder(copy1)
    copy(copy1, canvas, 0, getHeight(canvas)-getHeight(copy1)-5)

    #takes the opposite of each of the current values for red, green, and blue of the
    #original picture
    negative(copy1)
    #draws border around the first negative modified picture in the collage
    drawBorder(copy1)
    copy(copy1, canvas, getWidth(copy1), getHeight(canvas)-getHeight(copy1)-5)

    #averages red, green, and blue and replicates it so each color component has the
    #same value of the original picture
    grayScale(copy2)
    #draws border around the grayscale modified picture in the collage
    drawBorder(copy2)
    copy(copy2, canvas, getWidth(copy2)*2, getHeight(canvas)-getHeight(copy2)-5)

    #swaps out the different color values of the original picture
    colorSwapping(copy3)
    #draws border around the first color swapping modified picture in the collage
    drawBorder(copy3)
    copy(copy3, canvas,0, getHeight(canvas)-(getHeight(copy3)*2)-5)

    #lightens the colors of the pixels of the original picture
    lighten(copy4)
    #draws border around the lightened modified picture in the collage
    drawBorder(copy4)
    copy(copy4, canvas, getWidth(copy4),getHeight(canvas)-(getHeight(copy4)*2)-5)

    #takes the opposite of each of the current values for red, green, and blue of the
    #lightened modified picture
    negative(copy4)
```

← means the line is continued on the next line

```

#draws border around the second negative modified picture in the collage
drawBorder(copy4)
copy(copy4, canvas, getWidth(copy4)*2,getHeight(canvas)-(getHeight(copy4)*2)-5)

#swaps out the different color values of the second negative modified picture
colorSwapping(copy4)
#draws border around the second color swapping modified picture in the collage
drawBorder(copy4)
copy(copy4, canvas, 0, getHeight(canvas)-(getHeight(copy5)*3)-5)

blurPict=blur(copy3)           #blurs the first color swapping modified picture
#lightens the colors of the pixels of the blurred picture
lighten(blurPict)
#draws border around the lightened blurred modified picture in the collage
drawBorder(blurPict)
copy(blurPict, canvas, getWidth(copy1), getHeight(canvas)-(getHeight(copy1)*3)-5)

#swaps out the different color values of the first negative modified picture
colorSwapping(copy1)
#draws border around the third color swapping modified picture in the collage
drawBorder(copy1)
copy(copy1, canvas, getWidth(copy5)*2, getHeight(canvas)-(getHeight(copy5)*3)-5)
#this copies original and modified pictures to correct position on the canvas

signature=makePicture(getMediaPath("signature.png"))
newSig = scaleSig(signature, 0.15)      #scales signature down by a factor of 0.15
#tells where to put the signature on the canvas and what color to make the
#signature
addSig(canvas,newSig, 230, -7, black)

drawBorder(canvas)           #adds a border around the canvas

explore(canvas)

def addSig(canvas, signature, toX, toY, color):           #puts signature onto canvas
    toYStart=toY
    for x in range(0, getWidth(signature)):
        toY=toYStart
        for y in range(0, getHeight(signature)):
            p=getPixel(signature,x, y)
            if(getRed(p)<255 and getGreen(p)<255 and getBlue(p)<255):
                setColor(getPixel(canvas, toX, toY), color)
                toY=toY+1
            toX=toX+1
    return (canvas)

def scale(pict, copy1, factor):           #scales original picture down
    sourceX = 0                           #sets x to 0
    #loops through x coordinates from 0 to width times the scaling factor
    for targX in range(0, int(getWidth(pict)*factor)):
        sourceY = 0                       #sets y to 0
        #loops through y coordinates from 0 to height times the scaling factor
        for targY in range(0, int(getHeight(pict)*factor)):
            color = getColor(getPixel(pict, int(sourceX), int(sourceY)))
            setColor(getPixel(copy1, targX, targY), color)
            #increments y coordinate of pic by one divided by the scaling factor
            sourceY = sourceY + 1.0 / factor

```

← means the line is continued on the next line

```

sourceX = sourceX + 1.0 / factor

def scaleSig(pict, factor):
    #scales signature picture down
    copy1 = makeEmptyPicture(int(getWidth(pict) * factor), int(getHeight(pict) * factor))
    sourceX = 0
    for targX in range(0, int(getWidth(pict) * factor)):
        sourceY = 0
        for targY in range(0, int(getHeight(pict) * factor)):
            color = getColor(getPixel(pict, int(sourceX), int(sourceY)))
            setColor(getPixel(copy1, targX, targY), color)
            sourceY = sourceY + 1.0 / factor
            sourceX = sourceX + 1.0 / factor
    return copy1

#copy function that allows each picture to be copied onto the canvas
def copy(source, target, targX, targY):
    targetX = targX
    for sourceX in range(0, getWidth(source)):
        targetY = targY
        for sourceY in range(0, getHeight(source)):
            px=getPixel(source, sourceX, sourceY)
            tx=getPixel(target, targetX, targetY)
            setColor(tx, getColor(px))
            targetY=targetY + 1
            targetX=targetX + 1

#creates negative by taking the opposite of each of the current values for red,
#green, and blue of the picture
def negative(pict):
    for pixel in getPixels(pict):
        r=getRed(pixel)
        g=getGreen(pixel)
        b=getBlue(pixel)
        negColor=makeColor(255-r, 255-g, 255-b)
        setColor(pixel, negColor)

#averages red, green, and blue and replicates it so each color component has the same
#value of the picture
def grayScale(pict):
    for px in getPixels(pict):
        intensity=(getRed(px)+getGreen(px)+getBlue(px))/3
        grayColor=makeColor(intensity, intensity, intensity)
        setColor(px, grayColor)

def colorSwapping(pict):
    #swaps out the different color values of the picture
    for p in getPixels(pict):
        r=getRed(p)
        g=getGreen(p)
        b=getBlue(p)
        setRed(p, b)
        setGreen(p, r)
        setBlue(p, g)
        #changes the red value to the original blue value
        #changes the green value to the original red value
        #changes the blue value to the original green value

```

← means the line is continued on the next line

```

def lighten(copy4):
    #this lightens the colors of the pixels of
    the picture
    for x in range(0,getWidth(copy4)):
        for y in range(0,getHeight(copy4)):
            px=getPixel(copy4,x,y)
            color=getColor(px)
            color=makeLighter(makeLighter(color))
            setColor(px,color)

def blur(pict):
    #blurs the picture by reducing the pixelation
    blurPict=duplicatePicture(pict)
    for x in range(1, getWidth(pict)-1):
        for y in range(1, getHeight(pict)-1):
            top = getPixel(pict,x,y-1)
            left = getPixel(pict,x-1,y)
            bottom = getPixel(pict,x,y+1)
            right = getPixel(pict,x+1,y)
            center = getPixel(blurPict,x,y)
            #calculates avg color values
            newRed=(getRed(top)+ getRed(left) + getRed(bottom) + getRed(right)+ ↵
                    getRed(center))/5
            newGreen=(getGreen(top) + getGreen(left) + getGreen(bottom) + ↵
                    getGreen(right) + getGreen(center))/5
            newBlue=(getBlue(top) + getBlue(left) + getBlue(bottom) + getBlue(right)+ ↵
                    getBlue(center))/5
            setColor(center, makeColor(newRed, newGreen, newBlue))
    return blurPict

def drawBorder(pict):
    #creates a black border
    width=getWidth(pict)
    height=getHeight(pict)
    for px in getPixels(pict):
        x=getX(px)
        y=getY(px)
        if (x<5 or x>width-6 or y<5 or y>height-6):
            setColor(px,black)

```

↵ means the line is continued on the next line