Assignment 2: Face Detection in a Scaled Representation

Question 1 & Question 2

Script interaction pasted below, commented version shown at the end of the pdf:

jupyter MakePyramid Last Checkpoint: 9 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 2

Run Code

In [6]:
```
from PIL import Image, ImageDraw
import numpy as np
import math
from scipy import signal
import ncc

def MakePyramid(image,minsize):
    im = Image.open(image)
    Pyramid = [];
    while(im.size[0]>=minsize and im.size[1]>=minsize):
        Pyramid.append(im)
        im = im.resize((int(im.size[0]*0.75),int(im.size[1]*0.75)),Image.BICUBIC)
        im.show()
    return Pyramid
```

Out[6]:
```
[<PIL.JpegImagePlugin.JpegImageFile image mode=L size=358x342 at 0x1C0D1C62D0>,
 <PIL.Image.Image image mode=L size=268x256 at 0x1C0D1C6210>,
 <PIL.Image.Image image mode=L size=201x192 at 0x1C0D1C6310>,
 <PIL.Image.Image image mode=L size=150x144 at 0x1C0D1C6350>,
 <PIL.Image.Image image mode=L size=112x108 at 0x1C0D1C6390>,
 <PIL.Image.Image image mode=L size=84x81 at 0x1C0D1C63D0>,
 <PIL.Image.Image image mode=L size=63x60 at 0x1C0D1C6410>,
 <PIL.Image.Image image mode=L size=47x45 at 0x1C0D1C6450>,
 <PIL.Image.Image image mode=L size=35x33 at 0x1C0D1C6490>]
```

In [ ]:

jupyter ShowPyramid Last Checkpoint: 2 hours ago (autosaved)

Logout

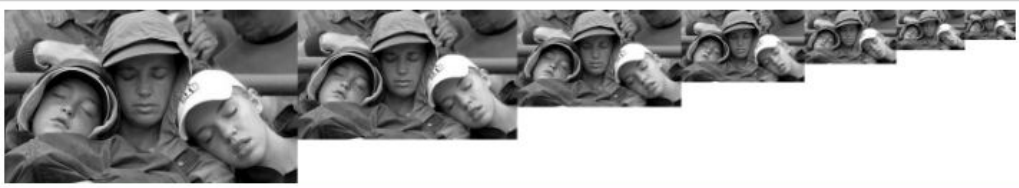
File Edit View Insert Cell Kernel Widgets Help Trusted Python 2

Run Code

In [13]:

```
from PIL import Image, ImageDraw
import numpy as np
import math
from scipy import signal
import ncc


def MakePyramid(image,minsize):
    im = Image.open(image)
    Pyramid = [];
    while(im.size[0]>=minsize and im.size[1]>=minsize):
        Pyramid.append(im)
        im = im.resize((int(im.size[0]*0.75),int(im.size[1]*0.75)),Image.BICUBIC)
    return Pyramid


def ShowPyramid(pyramid):
    x = 0
    y = 0
    for im in pyramid:
        x = x + im.size[0]
        y = max(im.size[1],y)

    image = Image.new("L",(x,y),255)
    offset_x = 0
    for im in pyramid:
        image.paste(im,(offset_x,0))
        offset_x = offset_x + im.size[0]
    image.save("showpyramid.png",png)
    image.show()
    return image

pyramid = MakePyramid("faces/fans.jpg",40)
ShowPyramid(pyramid)
```

Out[13]:

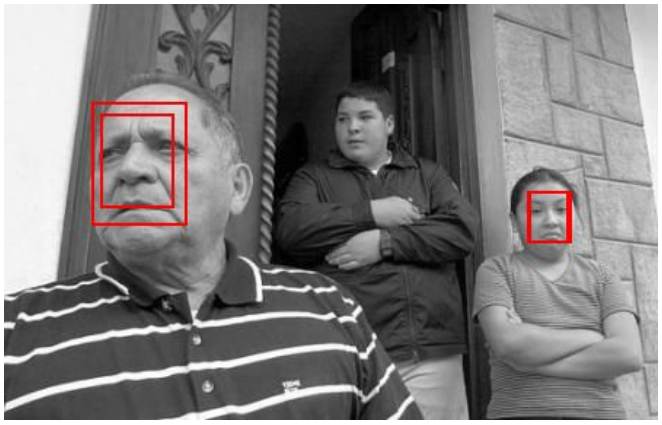


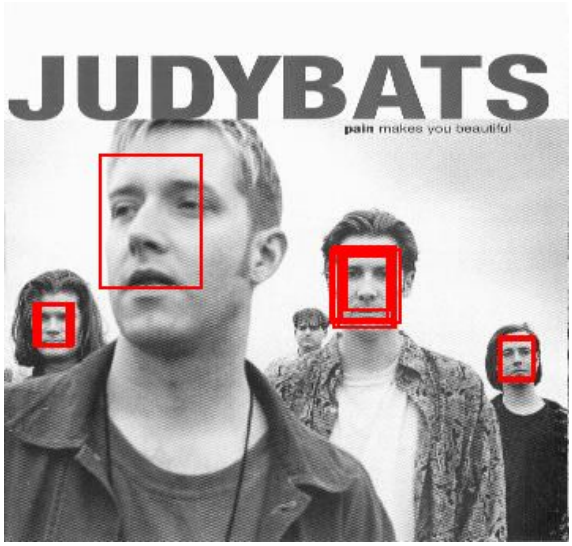
In [ ]:

Question 4 & Question 5

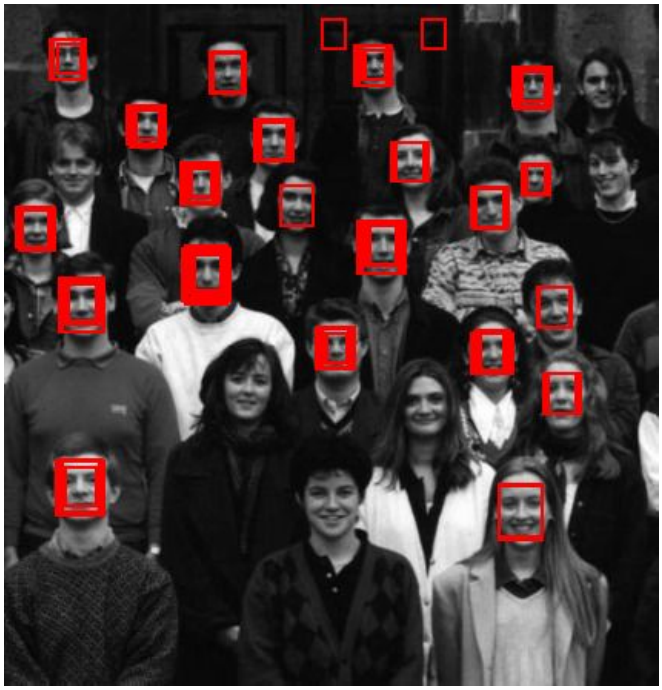
Try out Threshold	False Positive	False Negative
0.6	15	10
0.61	11	12
0.62	6	12

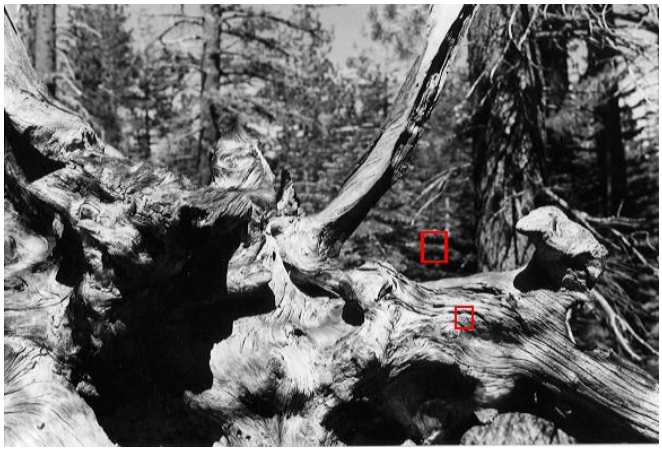
Final selected threshold = 0.61
False positive = 11
False negative = 12




JUDYBATS
pain makes you beautiful


BONDS
FRANCISCO
FILA





Question 6 Recall Rate

Image	True positives	Recall rate
family	2	2/3
fans	0	0/3
judybats	4	4/5
sports	0	0/1

students	21	21/27
tree	0	0/0

It's obvious that the two pictures have the lowest recall rate is relatively dark and the face orientation are much different than the template. So the NCC method has a very low recall rate on some images may due to different orientation, lighting conditions and also may cause by other factors that make the face dramatically different than the template.

Overall commented scripts and tests:

Part 1

jupyter FindTemplate Last Checkpoint: 5 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 2

Run Code

In [212]:
```
from PIL import Image, ImageDraw
import numpy as np
import math
from scipy import signal
import ncc

def MakePyramid(image,minsize):
    im = Image.open(image)
    #create pyramid array
    Pyramid = []
    #stop creating pyramid if the width or the height of the image are less then minsize
    while(im.size[0]>=minsize and im.size[1]>=minsize):
        Pyramid.append(im)
        #resize picture by the scale of 0.75
        im = im.resize((int(im.size[0]*0.75),int(im.size[1]*0.75)),Image.BICUBIC)
    return Pyramid

def ShowPyramid(pyramid):
    x = 0
    y = 0
    #set the size for the horizontal image
    for im in pyramid:
        x = x + im.size[0]
        #use the largest height of pyramid as the height of the horizontal image
        y = max(im.size[1],y)

    #create a new image with proper heigh and width, use white as background color
    image = Image.new("L",(x,y),255)
    offset_x = 0

    for im in pyramid:
        image.paste(im,(offset_x,0))
        #use offset_x to make sure pyramid listed horizontally
        offset_x = offset_x + im.size[0]
    image.save("outputs/showpyramid.png",'PNG')
    image.show()
    return image

def FindTemplate(pyramid,template,threshold):
    #resize the template to fix width 15px,and scale the height accordingly
    template_size = 15
    template = Image.open(template)
    template = template.resize((int(template_size),int(template.size[1]*template_size/template.size[0])),Image.BICUBIC)
    image = pyramid[0].convert('RGB')
    scale = 1
```

Part 2

jupyter FindTemplate Last Checkpoint: 6 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 2

Run Code

```
    for im in pyramid:
        ncc_result = ncc.normxcorr2D(im,template)
        #get all the locations that are aboved threshold
        above_threshold = np.where(ncc_result>=threshold)
        x = above_threshold[1]
        y = above_threshold[0]
        #adjust template by the right scale factor
        backToScale = 1/scale
        xoffset = template.size[0]*backToScale/2
        yoffset = template.size[1]*backToScale/2
        arraySize = len(x)
        for i in range(arraySize):
            #adjust x and by by the right scale factor
            xscaled = x[i]*backToScale
            yscaled = y[i]*backToScale
            #get the position for top-left,top-right,bottom-left,bottom-right
            x1 = xscaled-xoffset
            y1 = yscaled-yoffset

            x2 = xscaled+xoffset
            y2 = yscaled-yoffset

            x3 = xscaled+xoffset
            y3 = yscaled+yoffset

            x4 = xscaled-xoffset
            y4 = yscaled+yoffset
            #draw line based on the positions
            draw = ImageDraw.Draw(image)
            draw.line((x1,y1,x2,y2),fill="red",width=2)
            draw.line((x2,y2,x3,y3),fill="red",width=2)
            draw.line((x3,y3,x4,y4),fill="red",width=2)
            draw.line((x4,y4,x1,y1),fill="red",width=2)
        scale = scale *0.75
    return image
```

Part 3

jupyter FindTemplate Last Checkpoint: 6 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

```
            draw.line((x1,y1,x2,y2),fill="red",width=2)
            draw.line((x2,y2,x3,y3),fill="red",width=2)
            draw.line((x3,y3,x4,y4),fill="red",width=2)
            draw.line((x4,y4,x1,y1),fill="red",width=2)
        scale = scale *0.75
    return image


def tests():

    template_path = "faces/template.jpg"
    minsize = 10
    threshold = 0.61

    #test ShowPyramid
    pyramid = MakePyramid("faces/judybats.jpg",minsize)
    ShowPyramid(pyramid)

    #test Template Matching
    pyramid = MakePyramid("faces/judybats.jpg",minsize)
    image = FindTemplate(pyramid,template_path,threshold)
    image.save("outputs/Judybats1.png",'PNG')

    pyramid = MakePyramid("faces/fans.jpg",minsize)
    image = FindTemplate(pyramid,template_path,threshold)
    image.save("outputs/Fans2.png",'PNG')

    pyramid = MakePyramid("faces/family.jpg",minsize)
    image = FindTemplate(pyramid,template_path,threshold)
    image.save("outputs/Family3.png",'PNG')

    pyramid = MakePyramid("faces/sports.jpg",minsize)
    image = FindTemplate(pyramid,template_path,threshold)
    image.save("outputs/Sports4.png",'PNG')

    pyramid = MakePyramid("faces/students.jpg",minsize)
    image = FindTemplate(pyramid,template_path,threshold)
    image.save("outputs/Students5.png",'PNG')

    pyramid = MakePyramid("faces/tree.jpg",minsize)
    image = FindTemplate(pyramid,template_path,threshold)
    image.save("outputs/Tree5.png",'PNG')

    return

tests()
```