CPSC 425 Assignment 2 （due October 10, 2018）

Quan Zhang
48486154

2 def MakePyramid(image, minsize), in which “image” should be an predefined image and minsize should be a positive number of size. This function creates a pyramid for an image, returns a list including the original PIL image followed by all the PIL images of reduced size, using a scale factor of 0.75 from one level to the next.

Input: MakePyramid(img1, 300)

Output:
```
[<PIL.JpegImagePlugin.JpegImageFile image mode=RGB size=1500x1000 at
0x12758DD68>,
        <PIL.Image.Image image mode=RGB size=1125x750 at 0x1277A14A8>,
        <PIL.Image.Image image mode=RGB size=843x562 at 0x1270D1860>,
        <PIL.Image.Image image mode=RGB size=632x421 at 0x1270D1630>,
        <PIL.Image.Image image mode=RGB size=474x315 at 0x1270D14E0>]
```
Output is a pyramid which stops when any further reduction in size will make a dimension of the image smaller than minsize.

(implemented in file a2.py as def MakePyramid(image, minsize))

3 ShowPyramid(pyramid) Joins the images of image pyramid into a single horizontal image and displays them with image.show().

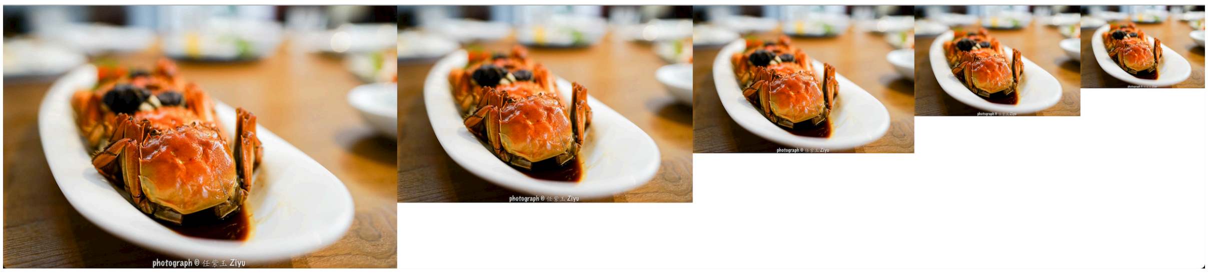
Input: ShowPyramid(MakePyramid(img1, 300))

Output: figure(2.1)

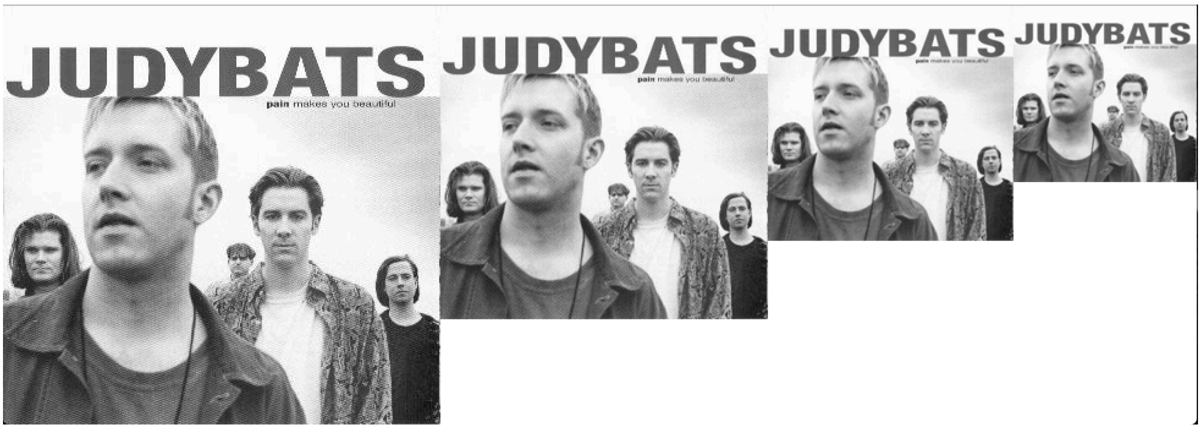
Input: ShowPyramid(MakePyramid(img2, 120))

Output: figure(2.2)

(implemented in file a1.py as ShowPyramid(pyramid))

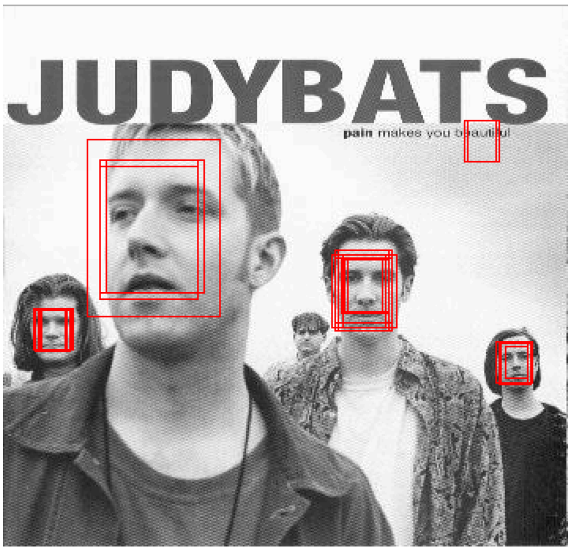


figure(3.1) image pyramid



figure(3.2) image pyramid

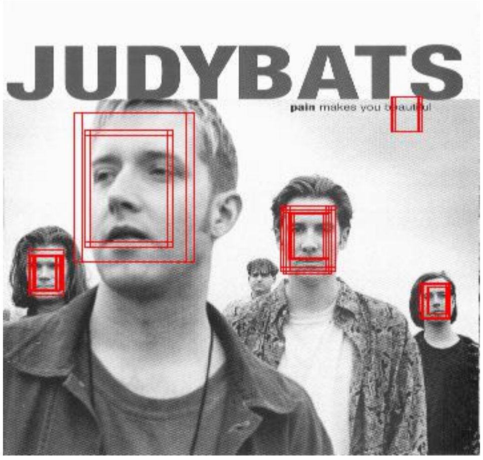
4 FindTemplate(pyramid, template, threshold), Finds and marks all locations in pyramid at which the normalized cross correlation of the template with the image is above the threshold.
Input: we define a threshold,
pyramid = MakePyramid(img, 20),
template = Image.open(templateLoc),
FindTemplate(pyramid, template, threshold)
Output: This function returns a PIL image of the largest image in the pyramid marked with red rectangle's corresponding to the locations of template matches.



figure(4) FindTemplate applies on threshold 0.5861

(implemented in file a2.py as FindTemplate(pyramid, template, threshold))

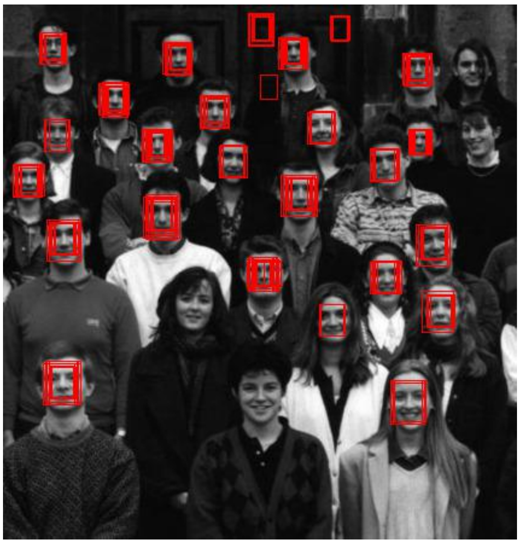
5 So the threshold I choose is 0.5715.
This threshold gives me the following results:



figure(4) judybats applies on threshold 0.5715

number of non-faces seen as faces: 1

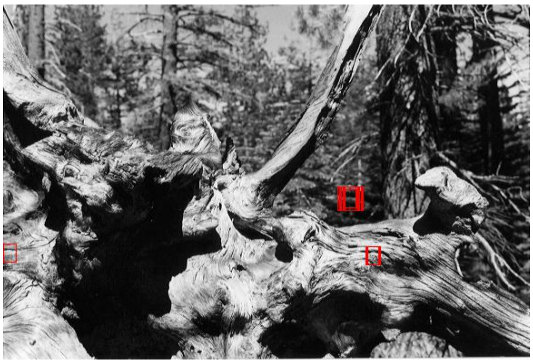
missed faces: 1



figure(4) students applies on threshold 0.5715

number of non-faces seen as faces: 3

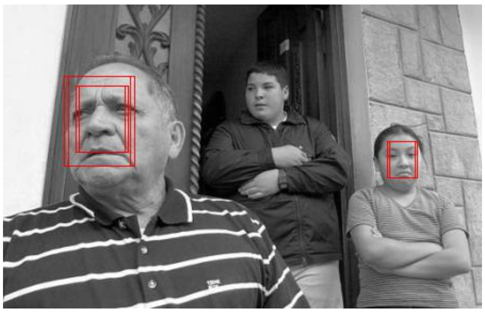
missed faces: 4



figure(4) tree applies on threshold 0.5715

number of non-faces seen as faces: 3

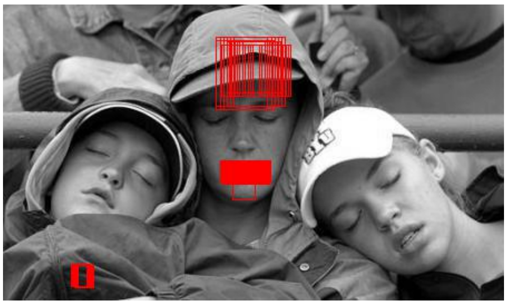
number of missed faces: 0



figure(4) family applies on threshold 0.5715

number of non-faces seen as faces: 0

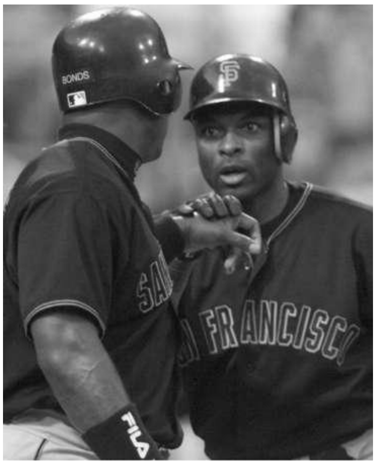
number of missed faces: 1



figure(4) fans applies on threshold 0.5715

number of non-faces seen as faces: 3

number of missed faces: 3



figure(4) sports applies on threshold 0.5715

number of non-faces seen as faces: 0

number of missed faces: 1

False positive = false negative = 10

6 Recall rate is True positive / (True positive + False negative).

R for judybats = 4/(4+1) = 0.8

R for students = 23/(23+4) = 0.85

R for tree is not applicable

R for family = 2/(2+1) = 0.66

R for fans = 0/3 =0

R for sports = 0

NCC method needs to normalize its two inputs correctly. So when the faces in some photos have the same angle (vertical in this case) to the temple, and have a good size (not too small), and not facing another direction (looking at the front), and have similar skin color, the method recognize them better. In this case, faces in judybats, students, and family are mostly recognized, but faces in fans and sports are not recognized.