

CPSC425: Assignment 2 (Python v2.7.15)

Alec Xu
38108130 (p7g9)

February 11, 2020

Part 1: Face Detection

NOTE: All testing scripts presented in appendix at end

Question 2 & 3

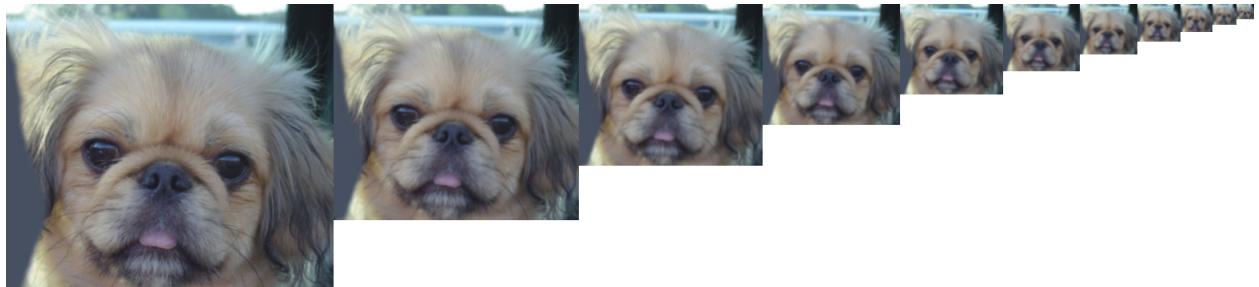


Figure 1: RGB Gaussian pyramid constructed using dog image. Scale = 0.75, minsize = 20

Gaussian blurring was applied for each level of the pyramid. To get the scaled dimensions for each layer, we cast straight to int, ignoring decimals.

Question 4

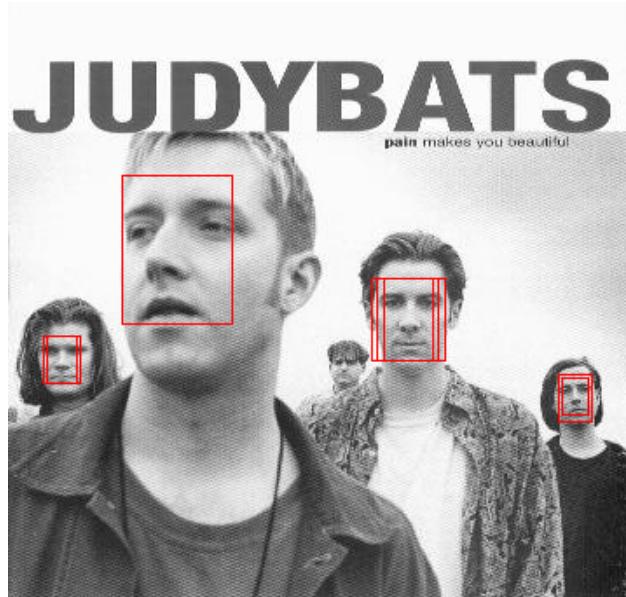


Figure 2: Face detection using template on judybats image. Pyramid constructed with same parameters as above. Threshold for template matching = 0.76, templateWidth = 15

Question 5

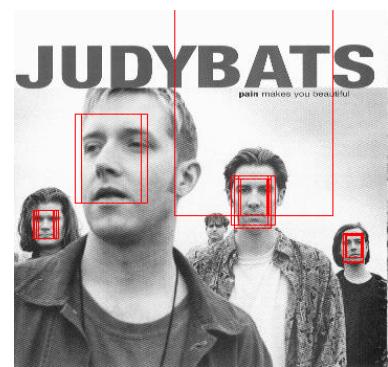
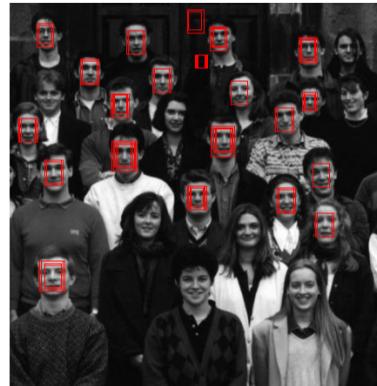


Figure 3: Faces matched using 0.69 as threshold

- Using 0.70 as the threshold, we get 15 missed faces and 3 not faces
- Using 0.69 as the threshold, we get 14 missed faces and 11 not faces
- Using 0.68 as the threshold, we get 11 missed faces and 17 not faces

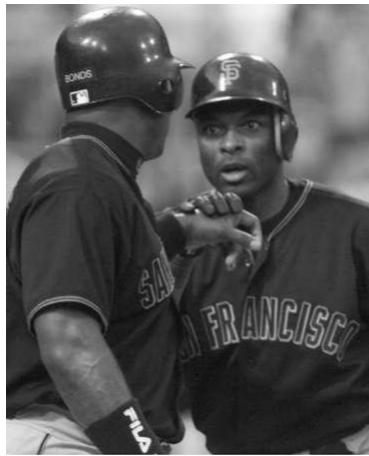


Figure 4: More faces matched using 0.69 as threshold

Question 6

- judybats: $4/5 = 80\%$
- students: $19/27 = 70\%$
- tree: $0/0$
- family: $0/3 = 0\%$
- fans: $0/3 = 0\%$
- sports: $0/1 = 0\%$

The template describes a white, young-adult, male at a frontal angle, with overhead lighting. It seems that the algorithm is best at detecting that. For most of the unrecognized faces, they were at a different angle, different lighting, female, or different ethnicity.

Part 2: Image Blending

Question 3



Figure 5: Laplacian pyramid of orchid with scale = 0.75, minsize = 20



Figure 6: Laplacian pyramid of violet with scale = 0.75, minsize = 20

Question 4



Figure 7: Laplacian pyramid of orchid reconstructed into Gaussian.



Figure 8: Laplacian pyramid of violet reconstructed into Gaussian.

Question 5

Figure 9: Gaussian pyramid of orchard mask.

Question 6

Figure 10: Blended orchid and violet.

Question 7

Figure 11: Blended cups, blended tomato and apple

Appendix A: Part 1 Testing Script

```
from hw2functions import *
from PIL import Image
import numpy as np

# Part 1

# Question 2 & 3
im = Image.open("dog.jpg").convert('RGB')
imArr = np.asarray(im, dtype=np.float32)
GPyramid = MakeGaussianPyramid(imArr, 0.75, 20)
ShowGaussianPyramid(GPyramid)

# Question 4
template = Image.open("hw2part1/template.jpg").convert('L')
template = np.asarray(template, dtype=np.float32)

judyBatsIm = Image.open("hw2part1/judybats.jpg").convert('L')
judyBatsArr = np.asarray(judyBatsIm, dtype=np.float32)

pyramid = MakeGaussianPyramid(judyBatsArr, 0.75, 20)

FindTemplate(pyramid, template, 0.76, 15)

# Question 5
# Using 0.70 as the threshold, we get 15 missed faces and 3 not faces
# Using 0.69 as the threshold, we get 14 missed faces and 11 not faces
# Using 0.68 as the threshold, we get 11 missed faces and 17 not faces
template = Image.open("hw2part1/template.jpg").convert('L')
template = np.asarray(template, dtype=np.float32)

Arrs = []
imgNames = ["judybats", "students", "tree", "family", "fans", "sports"]
for i in range(0, len(imgNames)):
    img = Image.open("hw2part1/" + imgNames[i] + ".jpg").convert('L')
    Arrs.append(img)

pyramids = []
for i in range(0, len(Arrs)):
    pyramids.append(MakeGaussianPyramid(Arrs[i], 0.75, 20))
    FindTemplate(pyramids[i], template, 0.69, 15)
```

Appendix B: Part 2 Testing Script

```
from hw2functions import *
from PIL import Image
import numpy as np
# Part 2

# Question 2 & 3
imOrchid = Image.open("./hw2part2//orchid.jpg").convert('RGB')
imViolet = Image.open("./hw2part2//violet.jpg").convert('RGB')

imOrchidArr = np.asarray(imOrchid, dtype=np.float32)
imVioletArr = np.asarray(imViolet, dtype=np.float32)

# Define uniform scale and minsize
scale = 0.75
minsize = 20

LPyramidOrchid = MakeLaplacianPyramid(imOrchidArr, scale, minsize)
LPyramidViolet = MakeLaplacianPyramid(imVioletArr, scale, minsize)

ShowLaplacianPyramid(LPyramidOrchid)
ShowLaplacianPyramid(LPyramidViolet)

# Question 4
ShowGaussianPyramid(ReconstructGaussianFromLaplacian(LPyramidOrchid))
ShowGaussianPyramid(ReconstructGaussianFromLaplacian(LPyramidViolet))

# Question 5
imMask = Image.open("./hw2part2//orchid_mask.bmp").convert('RGB')
imMaskArr = np.asarray(imMask, dtype=np.float32)

GPyramidMask = MakeGaussianPyramid(imMaskArr, scale, minsize)
ShowGaussianPyramid(GPyramidMask)

# Question 6
# Make minsize smaller for better blends
minsize = 8

LPyramidOrchid = MakeLaplacianPyramid(imOrchidArr, scale, minsize)
LPyramidViolet = MakeLaplacianPyramid(imVioletArr, scale, minsize)
GPyramidMask = MakeGaussianPyramid(imMaskArr, scale, minsize)

LPyramidBlend1 = BlendPyramids(LPyramidOrchid,LPyramidViolet,GPyramidMask)
```

```
ShowGaussianPyramid(ReconstructGaussianFromLaplacian(LPyramidBlend1))
```

```
# Question 7
```

```
imMask = np.asarray(Image.open("./hw2part2//tomato_mask.bmp").convert('RGB'), dtype=np.float32)
im1 = np.asarray(Image.open("./hw2part2//tomato.jpg").convert('RGB'), dtype=np.float32)
im2 = np.asarray(Image.open("./hw2part2//apple.jpg").convert('RGB'), dtype=np.float32)
```

```
im1 = MakeLaplacianPyramid(im1, scale, minsize)
im2 = MakeLaplacianPyramid(im2, scale, minsize)
imMask = MakeGaussianPyramid(imMask, scale, minsize)
```

```
LPyramidBlend2 = BlendPyramids(im2,im1,imMask)
ShowGaussianPyramid(ReconstructGaussianFromLaplacian(LPyramidBlend2))
```

```
imMask = np.asarray(Image.open("./hw2part2//blue_cup.jpg").convert('RGB'), dtype=np.float32)
im1 = np.asarray(Image.open("./hw2part2//green_cup.jpg").convert('RGB'), dtype=np.float32)
im2 = np.asarray(Image.open("./hw2part2//cup_mask.bmp").convert('RGB'), dtype=np.float32)
```

```
im1 = MakeLaplacianPyramid(im1, scale, minsize)
im2 = MakeLaplacianPyramid(im2, scale, minsize)
imMask = MakeGaussianPyramid(imMask, scale, minsize)
```

```
LPyramidBlend3 = BlendPyramids(im2,im1,imMask)
ShowGaussianPyramid(ReconstructGaussianFromLaplacian(LPyramidBlend3))
```