Untitled-1

April 17, 2024

[ ]:
```
from os import path

if not path.exists('utils.py'): # If running on colab: the first time download␣
 ↪and unzip additional files
    !wget https://biolab.csr.unibo.it/samples/fr/files.zip
    !unzip files.zip


import utils  # Run utils.py for helper functions
import fingerprint as fp
import math
import os
import importlib
import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt
#import PyQt5.QtWidgets as qtw
from utils import *
from ipywidgets import interact

%cd /workspaces/fingerprint/
```

```
/workspaces/fingerprint
```

```
/home/vscode/.local/lib/python3.10/site-packages/IPython/core/magics/osm.py:417:
UserWarning: using dhist requires you to install the `pickleshare` library.
  self.shell.db['dhist'] = compress_dhist(dhist)[-100:]
```

[ ]:
```
importlib.reload(fp)
importlib.reload(utils)
```

[ ]:
```
<module 'utils' from '/workspaces/fingerprint/A2/utils.py'>
```

[ ]:
```
#Show the resultant image for the first element of the database
readFingerPrints = fp.read_fingerprints()
GxList, GyList, Gx2List, Gy2List, GmList = fp.calc_sobel(readFingerPrints)
_, filename = readFingerPrints[0]
```

```
show((Gx2List[0], f'Gx**2 {filename}'), (Gy2List[0], f'Gy**2 of {filename}'),␣
 ↪(GmList[0], f'Gradient Magnitude {filename}'))

print(readFingerPrints[0][0].shape)
```

```
<IPython.core.display.HTML object>
```

```
(300, 300)
```

[ ]:

```
w, h = readFingerPrints[0][0].shape
fingerprint = readFingerPrints[0][0]
cropWidth = int(0.3 * w)
cropHeight = int(0.3 * h)
endRow = (h - cropHeight) // 2
startRow = endRow - 80
rowLength = len(fingerprint[startRow:endRow])
# startY = (h - cropHeight) // 2
# endX = startX + 80
middleColumn = (w - cropWidth) // 2
startColumn = middleColumn - 25
endColumn = middleColumn + 25
columnLen = len(fingerprint[startColumn:endColumn])
endY = (h - cropHeight) // 2
#endY = startY + cropHeight
print(startRow, endRow, startColumn, endColumn)
region = fingerprint[startRow:endRow, startColumn:endColumn]

print("Row Lenght = %d" % rowLength)
print("Column Length = %d" % columnLen)
show(region)
```

```
25 105 80 130
Row Lenght = 80
Column Length = 50
```

```
<IPython.core.display.HTML object>
```

[ ]:

```
# Integral over a square window for 102_5.tif
sumGm = fp.sum_Gm(GmList)

show(sumGm[0], f'Integral of the Gradient Magnitude of {filename}')
```

```
<IPython.core.display.HTML object>
```

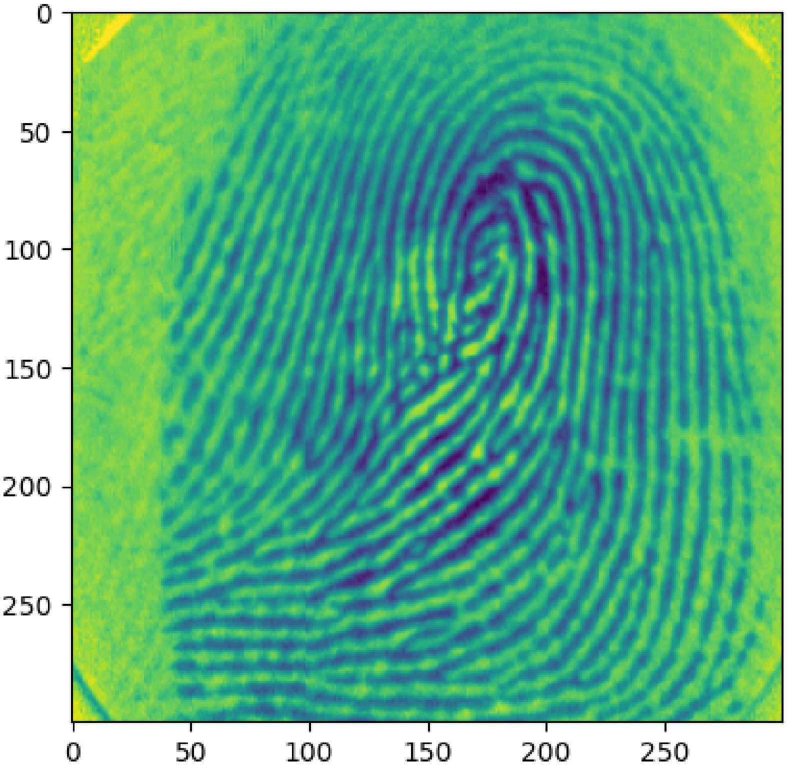
[ ]:

```
# Simple thresholding for segmenting the fingerprint pattern

thres = fp.threshold_mask(sumGm)
```

```
show((readFingerPrints[0][0], f'Original Image of {filename}'), (thres[0],␣
 ↪f'Threshold Mask of {filename}'), (cv.merge((thres[0],␣
 ↪readFingerPrints[0][0], readFingerPrints[0][0])), f'Mask Backprojected to␣
 ↪the Original Image of {filename}'))
plt.imshow(readFingerPrints[0][0])
plt.show()
```

```
<IPython.core.display.HTML object>
```



[ ]:
```
orientationsList, strengthsList = fp.ridge_orientation(GxList, GyList, Gx2List,␣
 ↪Gy2List)
show((draw_orientations(readFingerPrints[0][0], orientationsList[0],␣
 ↪strengthsList[0], thres[0], 1, 16)), f'Orientation Image {filename}')
```

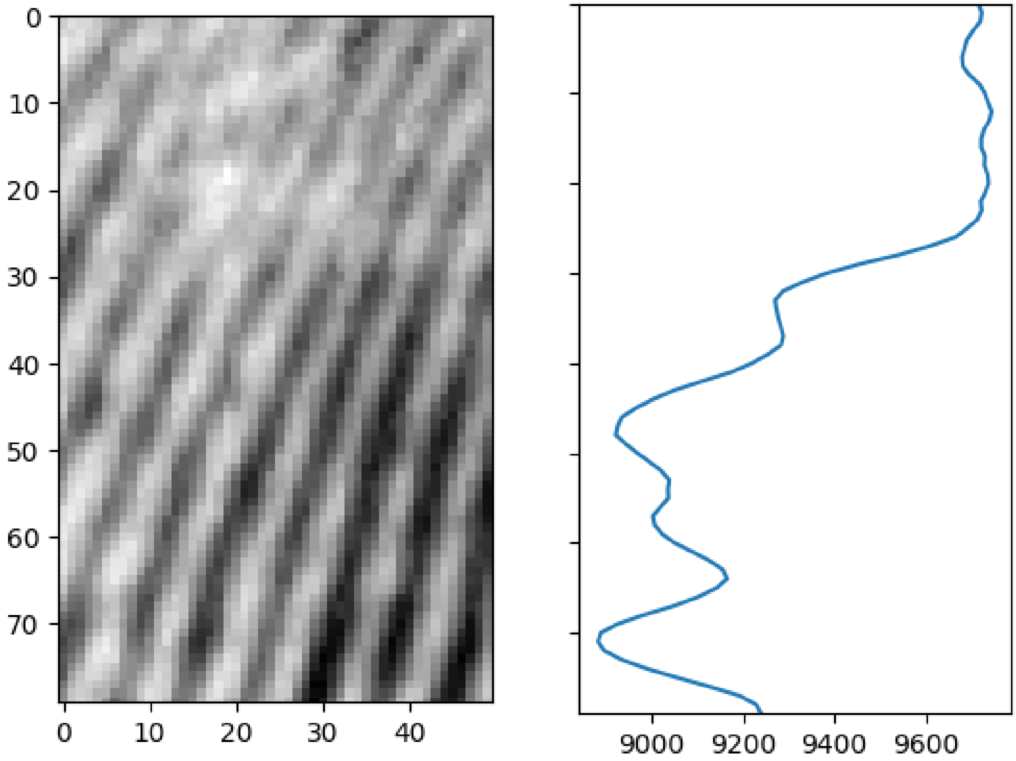
```
<IPython.core.display.HTML object>
```

[ ]:
```
regionList, locMaxList, xSigList = fp.ridge_frequency(readFingerPrints)
show(regionList[0])
print(xSigList[0])
x = np.arange(regionList[0].shape[0])
f, axarr = plt.subplots(1, 2, sharey = True)
axarr[0].imshow(regionList[0], cmap = 'gray')
```

```
axarr[1].plot(xSigList[0], x)
axarr[1].set_ylim(regionList[0].shape[0] - 1, 0)
plt.show()
```

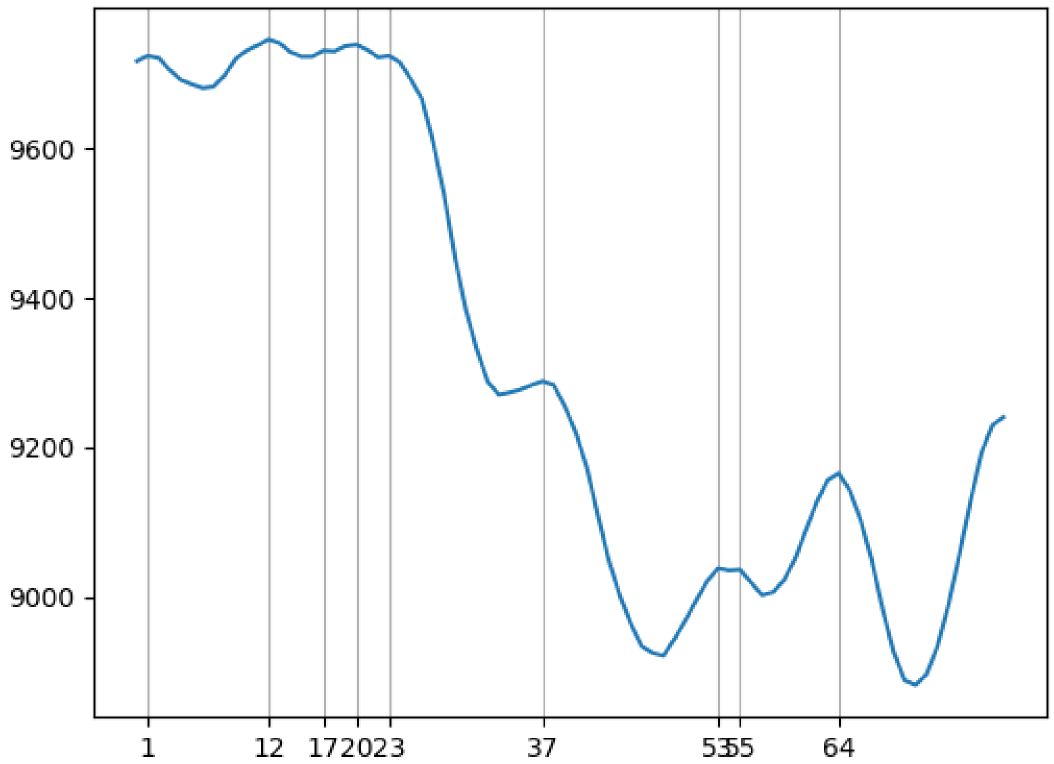
```
<IPython.core.display.HTML object>
```

```
[9716 9723 9720 9704 9691 9685 9680 9682 9697 9719 9730 9737 9745 9740
 9728 9722 9722 9730 9729 9736 9738 9731 9721 9723 9714 9691 9665 9608
 9538 9454 9383 9330 9287 9270 9273 9277 9283 9288 9283 9255 9220 9174
 9111 9050 9003 8965 8934 8925 8921 8943 8968 8995 9021 9038 9035 9036
 9019 9002 9006 9022 9050 9089 9127 9156 9165 9142 9102 9049 8983 8926
 8888 8882 8896 8934 8989 9056 9127 9192 9229 9240]
```



[ ]:

```
# Find the indices of the x-signature local maxima
plt.plot(x, xSigList[0])
plt.xticks(locMaxList[0])
plt.grid(True, axis='x')
plt.show()
```



[ ]:

```
distanceList, ridgePeriodList = fp.ridge_period(locMaxList)
distances = distanceList[0]
ridgePeriod = ridgePeriodList[0]
print("distance array of %s = %s" % (filename,distances))
print("average ridge period of %s = %s" % (filename, ridgePeriod))
```

```
distance array of 102_5.tif = [11  5  3  3 14 16  2  9]
average ridge period of 102_5.tif = 7.875
```

[ ]:

```
gaborBankList = fp.gabor_bank(ridgePeriodList)
gaborBank = gaborBankList[0]
gaborBank2 = gaborBankList[1]


#print(gaborBank2)
#print(gaborBank2)
show(*gaborBank)
```

```
<IPython.core.display.HTML object>
```

[ ]:

```
fFPList, nfFBList = fp.filter_fingerprint(readFingerPrints, gaborBank)
show(nfFBList[0], *fFPList[0])
```

```
<IPython.core.display.HTML object>
```