NOVA SCHOOL OF SCIENCE & TECHNOLOGY

Advanced Topics in Digital Image Processing

LESSON 9 - Pattern Recognition & Thermal Imaging

Objective:

In this class students should familiarize themselves with the algorithms of Template Matching, Viola-Jones for facial recognition and Histogram of Oriented Gradients (HOG) and evaluate some of their potentialities for image processing. Also, students will try out a Thermal Camera.

Procedure:

0 - Thermal Imaging – Use the Thermal Camera from the laboratory to analyze one equipment (computer, WiFi AP, charger, mobile, etc.) and also capture a face to try with the face detection algorithms. Download the images from the camera.
Note: while you are waiting for the IR camera continue with the other tasks.

1 - Template Matching Algorithm - apply the template matching algorithm on the available images to search for one of the template images. Test with the different calculation methods (Sum square difference, Correlation, Correlation Coefficient). (OpenCVs functions *cv.matchTemplate and cv.minMaxLoc, cv.drawMarker)*

2 - Viola-Jones algorithm – Apply the Viola-Jones algorithm on the available images for face recognition and show the detected face (rectangle) over the image. (functions of the *CascadeClassifier - detectMultiScale, cv.rectangle)*

3 – HOG algorithm - apply the HOG algorithm on the available images for recognizing pedestrians and show the detected pedestrian over the image. (functions of the *HOGDescriptor, cv.rectangle*)

4 – Apply on a video sequence - Use the Capture component of the OpenCV library to acquire images from a webcam or video recorded file.

a. Develop a routine that shows the webcam video in continuous mode and applies the face detection algorithm and then defocus the detected face.
b. Develop a routine that shows the pedestrians video in continuous mode and applies the detection algorithm.

OpenCV Functions

- ***Template Matching***

```
result = cv.matchTemplate(
    image,
    template,
    method,
    mask = None
)
```

Returns an image (single channel only) with the result of the Template Match.

Parameters:

Image – original image (WxH)

Temp - template image (twxth)

Result – image of type – size = W-tw+1xH-th+1

Method – calculation method (TM_SQDIFF, TM_SQDIFF_NORMED, TM_CCORR, TM_CCORR_NORMED, TM_CCOEFF, TM_CCOEFF_NORMED)

Mask – apply only on certain pixels

- ***minMaxLoc***

```
minVal, maxVal, minLoc, maxLoc = minMaxLoc()
```

Returns the value and location of the minimum and maximum values of the image for each channel.

- ***Viola Jones***

```
haar = cv.CascadeClassifier("haarcascade_frontalface_alt2.xml");

faces = haar.detectMultiScale(
        img, scaleFactor = 1.4,
        minSize = (20, 20),
        maxSize = (width // 2, height // 2) )
```

Returns a list of rectangles that are listed as face. Allows you to set minimum and maximum size.

- ***Histogram of Oriented Gradients (HOG)***

```
hog = cv.HOGDescriptor()
hog.setSVMDetector(cv.HOGDescriptor.getDefaultPeopleDetector() )
pedestrians = hog.detectMultiScale(imgOriginal)

for (x,y,w,h) in pedestrians[0]:
  img = cv.rectangle(img, (x,y), (x+w,y+h), (0, 0, 255), 4)
```

Returns a list of detected objects (with rectangle), in this case of people.