Submission details

- This assignment should be done in pairs (contact the TA if this is a problem).
- The topics of this assignment are Template Matching and Object Detection.
- The submission date is **5/11/2016**. Please pay attention to the late submission policy.
- Coding should be done in Matlab or Python. We prefer Python 2, but Python 3 is possible.
- You are not required to use any specific function or library, unless stated otherwise. If in doubt, please contact the TA via email or the Piazza website.
- For submission, package up your code as a zip file. Include your written answers as a pdf file named writeup.pdf. Include graphs and images in your writeup, and please label them so we know which figures go with which sub-problem.
- Send the final zip file to the TA. Add the course number to the subject of the email.
- If you have any questions about this assignment, please contact the TA stinger@tx.technion.ac.il.

Task 1: Optical Character Recognition using Template Matching

Optical Character Recognition (OCR) is a method for extracting text from images. In complex problems, composed of handwritten characters, Machine Learning methods are used to train a classifier to distinguish between different characters. In this question we will deal with a much simpler problem, equivalent to a problem of creating electronic versions of old books. In such cases, the use of Template Matching is often sufficient and quite successful.

A short text by Edgar Allan Poe was scanned into the image text.jpg and was given to you without any additional information. We would like to learn a few details on this text.

1. Find out what is the font of the text using the images in the fonts folder and the SSD measure. Explain the process and show the results.
2. Find out what is the size of the font using the images in the sizes folder and the SSD measure. Show the results.
3. Find out how many times do the letters 't' and 'a' appear in the text, including capital letters. Do this by cutting the appropriate templates from the text itself (see previous items for good template examples) and use the SSD measure. Try to find a single rule for setting the threshold for the matching (it can be a function of the template).
    a. Show your templates.
    b. Explain your threshold selection
    c. Write how many times each of the letters appear. Did you locate all the appearances?
4. Consider the previous items and suggest (in short) an automated process for detecting and recognizing letters and words in such images. No need to code here.
5. Help Mr. Smoke-too-much read this text to his friends. Do this by replacing each appearance of the letter 'c' with the letter 'k'. Use the given templates and the SSD measure. Show the new text. Does it look natural? Can you read it to your partner?

Task 2: Object Detection in Images

In this task you will build a system for specific object detection. We're providing you with 3 datasets and it's ok to have a solution that's good only for this data. If you'd rather make it broader that's also fine. Creativity is a key factor in this assignment. Thinking out-of-the-box could lead to better performance. A possible idea is to search for relevant papers and follow the footsteps of one that you like. The detection performance will have some part in the grading (but not a big one).

1. Propose a method for finding a given template image in a set of query images. Any method you propose is fine, as long as you explain and motivate it clearly. Before you code it and test it, please write the following:
    a. The motivation to the method you propose.
    b. The guiding principles of your approach.
    c. In what cases do you expect it to succeed or fail?
    d. What are the advantages and disadvantages of your approach?
2. Implement the method and apply it to the 3 provided datasets. To evaluate the quality of the results draw on each query image green bounding-boxes of the detected templates.
3. Based on your experimental results, revisit item 1 and assess your predictions. Associate examples from your results with items on your list of pros & cons. Explain the success/failure cases you got.
4. Suggest a modification to your method that will improve at least one of the failure modes, and implement it. Did it work? Explain why.