ת.ז.

בחינות

מבוא לראייה ממוחשבת (10224) - אביב תש״ף

Introduction to Computer Vision (10224) - Spring 2020

Home Midterm

Published: 10.05.2020, 08:00

Due Date: 18.05.2020, 07:59

Late Policy: Late submission is not permitted. Late submitted files will not be accepted and / or checked.

Guidelines: All work will be individual and original for each student. Teamwork of any kind, answers or code from any source such as the internet, and / or from tutorials, help files etc., are forbidden. Only your own, individually written code and answers are permitted. Use of theoretical references and course material are permitted and encouraged. **All code will be written in python, with the usual code writing guidelines (PEP 008). All code will be submitted with the same submission guidelines for this course. All docs should be submitted in Word or PDF format. Scanned answers should be submitted as a single file PDF only.**

Any two or more works with part or whole “too similar” code or document will disqualify (midterm grade = 0) all relevant works, for both the source and the copy.

Personal interview related to your work knowledge (via Zoom or other video conferencing means):

The course team may choose to interview some or all of the submitting students (with online video and audio) to verify that the work was done personally, to explain parts of the code or of the accompanying documentation and results.

This work contains two parts:

Part 1: Written Problems (55%)

Please handwrite your response on the problem’s pages, (either in Hebrew or in English), and attach it as **a single PDF document**.

Part 2: Code with report (45%)

Please attach three types of files: Python code and a pdf / doc report document, data folder with input and output images.

בחינות

Part 1:

Problem 1 (35%) שאלה 1

For the following *6x6* 2D image ***f*** (**shown in bold black**), and the 3x3 filters ***h1, h2:***

$$f = \begin{bmatrix} 200 & 200 & 200 & 200 & 200 & 200 & 200 & 40 \\ 40 & \mathbf{200} & \mathbf{200} & \mathbf{200} & \mathbf{200} & \mathbf{200} & \mathbf{40} & 40 \\ 40 & \mathbf{40} & \mathbf{200} & \mathbf{200} & \mathbf{200} & \mathbf{40} & \mathbf{40} & 40 \\ 40 & \mathbf{40} & \mathbf{40} & \mathbf{200} & \mathbf{40} & \mathbf{40} & \mathbf{40} & 40 \\ 40 & \mathbf{40} & \mathbf{40} & \mathbf{40} & \mathbf{40} & \mathbf{40} & \mathbf{40} & 40 \\ 160 & \mathbf{160} & \mathbf{160} & \mathbf{160} & \mathbf{160} & \mathbf{160} & \mathbf{160} & 160 \\ 160 & \mathbf{160} & \mathbf{160} & \mathbf{160} & \mathbf{160} & \mathbf{160} & \mathbf{160} & 160 \\ 160 & 160 & 160 & 160 & 160 & 160 & 160 & 160 \end{bmatrix}$$

$$\boldsymbol{h1} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}; \qquad \boldsymbol{h2} = \frac{1}{5} \cdot \begin{bmatrix} 0 & -2 & -1 \\ 2 & 0 & -2 \\ 1 & 2 & 0 \end{bmatrix}$$

a. (3%) Name and shortly describe two differences between convolution and correlation operations:

1. ____

2. ____

b. (2%) Name and fully explain the function performed by ***h1***: ____

____\

____

c. (3%) Name and fully explain the function performed by ***h2***: ______

____

Calculate the following convolution results (**write and explain your calculations in your notebook / Added pages,** then, copy the results below). **Convolution** is marked by '*', and **Correlation** is marked by '⊗'.

Notes: If necessary, you may expand the values of ***f***, by mirroring (new pixels identical to adjacent existing pixels).

d. (7%) $g1 = f * h2$

e. (5%) $g2 = h1 * f$

f. (5%) $g3 = g1 * h1$

g. (5%) $g4 = h2 * f * h1$

h. (5%) $g5 = f \otimes h1$

d - ***g1***

e - ***g2***

f - ***g3***

g - ***g4***

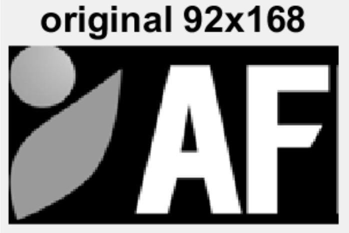
h - ***g5***

ת.ז.

בחינות

Question Number 2 (20%) שאלה 2

על מקטע מהלוגו של אפקה (באיור משמאל), ביצעו סינון במרחב התמונה ע״י קונבולוציה עם מסנן אחד בכל פעם. גודל התמונה נתון והוא כפי שמופיע באיור. ישנם 4 מסננים בגדלים שונים, שרשום באיור 2 (המסננים לפני סיבוב ב-180 מעלות). יש להתאים בין כל מסנן (1-4) לבין תמונת התוצאה המתאימה (A-D). התמונה המוצגת עשויה לכלול מספרים שליליים, חיוביים ואפס. יש לנמק את בחירתכם.



א. ערכי מסנן מספר 1

א. (3%) רשום בטבלה משמאל את ערכי מסנן מספר 1 :

ב. (2%) מה סוג המסנן? ______________________________

ג. (3%) מה השימוש המתאים למסנן מסעיף א? ____________________

____

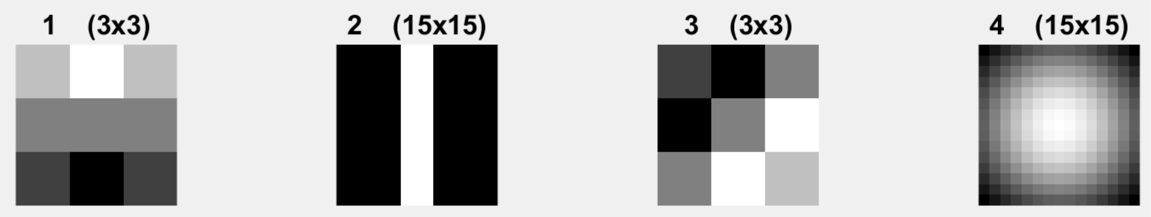
ד. (12%) התאם בין המסנן לתמונת התוצאה.

כל התאמה נכונה תזכה ב 3%, וכל התאמה שגויה תוריד 3%, כך שלא מומלץ לנחש.

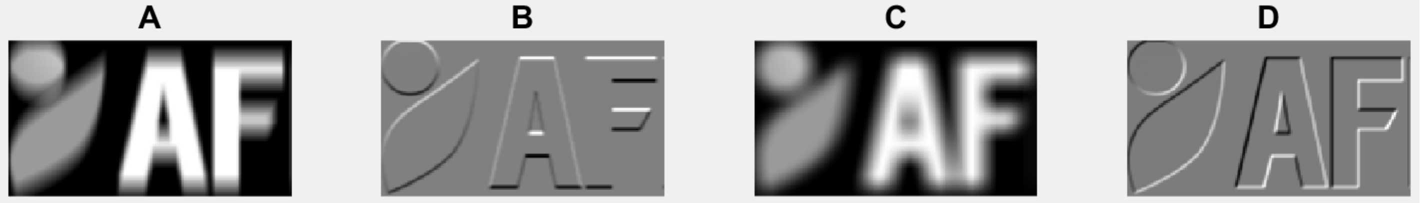
(ציון מינימלי לסעיף : 0%). (רמז : ניתן לבדוק במחברת את תשובתכם על תמונת ניסיון לפי בחירתכם, ולהעזר בה בנימוק).

האות הרשומה מעל לתמונת התוצאה	המספר הרשום מעל למסנן	נימוק לתשובה – חובה לנמק\ להסביר את בחירתכם (ניתן להרחיב במחברת)
	1	
	2	
	3	
	4	

תמונות המסננים (בסוגריים – מידות המסננים בפיקסלים):



תמונות התוצאה (לפי קנה מידה שונה):



Part 2:

Goal

Convolutional Networks are based on the **correlation operation** (**convolution** without 180 degrees rotation of the filter kernel). In this work you will write and use the basic convolution operation, including Stride and Padding. The code will be implemented in **numpy**.
More reference to the required operations can be found in the course material.
Create your own convolution operation, observe the results, and demonstrate it.

1 Assumptions

The function is ***result = myCorr (Im, Ker, S, P, N, Norm, CORR, ACTV)***

Your input is an image, ***Im***, with shape: ***WxHxC***, where ***W*** is the image width in pixels, ***H*** is the height, and ***C*** is the number of channels.

You should also provide the weights of the kernel, ***Ker*** of size ***FxFxC***, as an input.

Stride ***S***, Padding ***P***, and the number of kernels to apply, ***N***.

Additional hyper parameter is **Norm** - normalization of the image patch (subtracting the patches mean - see slide 11 of lecture 5 – also appears in the appendix of this document).

Additional hyper parameters are:

CORR - whether to perform **conv** or **corr,** and

***ACTV* whether to apply a nonlinear activation function (none and RELU)**.

Your returned output image (activation map) will be of shape ***W1xH1x1.***

Kernel examples should be taken from **Problem 2 of part 1**, from the example slide (Einstein's eye matching in the appendix and lecture). Choose more examples (RGB and gray scale) to show functionality of your work, as described later.

2 Objectives

The objectives are to expand the Lab 2 work: to create the forward path of a convolutional layer (activation map), by adding to the convolution (correlation) **Stride**, **Padding** and **Normalization** (**Normalized Cross Correlation to enable template matching**), followed by an activation non linearity. Then to apply multiple kernels to a single input image / activation map.

A submission package for this part, should include a descriptive report in **word and pdf documents**, a **code file** (html notebook for python) and **input and resulted output images** should be submitted. In the document, all operations should be explained and demonstrated in the doc file, as well as an explanation of your parameters selection and implementation decisions and considerations / reasoning (see below).

3 Work Details

3.1 Code

In order to meet the above objectives, two main functionalities are required:

- **Create a single convolution operation** from scratch, using numpy basic matrices and matrix operations only, **without any** image processing libraries, such as cv2, PIL, etc.
- Note that Using built in image processing functions for any part of the calculation and application (such as `conv2d`, or `cv2.matchTemplate(),#TM_CCORR_NORMED method`, is forbidden. However, using *imshow*, *imread* and *imwrite* is permitted.
  The validation of this part should be done against built-in functions, such as the ones mentioned above.
  NOTES:
    - The operation here should include: Calculating the index change, if relevant, then applying the intensity levels.
    - Output image should have the correct size / dimensions, based in the input size and hyper parameters.
    - Operation will work on all types of images: grey level / RGB / Float / UINT8. The functions will return a signed version of the input, and will be scaled during presentation with imshow (use of package functions for image type conversion is allowed)
    - Verification procedure / function should be created and performed, by comparing to the built-in functions, and calculating differences in results. In your report, relate to this note in detail.
    - Padding will be performed with zeros (also after normalization).
- **Demonstrate the template matching ability of the function you created, with different types and sized of template patches and images.**

3.2 Multiple sets of convolutional filters operated on a single image

Demonstrate the use of the above function for creating a convolution layer, set of filters. In other words, operate N Filters on the same image, to create a layer of depth N. (output shape ***W1xH1xN***). Add an activation layer per your choice (RELU is sufficient).

Cascade two layers of the above, to demonstrate the operation.

3.3 Expected output images for this part:

Implement and apply convolution / correlation layer. No weights calculation and backpropagation are required. Apply at least 10 **different filters / kernels** (edge detection, specific pattern detector – e.g. a 45 degrees line, Gaussian blur, etc.), operated on 5 **different types and sizes of images**, with **different hyper parameters** (e.g. filter size, stride, padding, Xcorr) generated form each original image. **Operate on (1) color Lena image, (2) Afeka Logo (3) MNIST digits. (4) 5 ImageNet examples (5) Where's Waldo**.

בחינות

In your report, **show only selected samples, and not all outputs. Show your understanding by selecting a few distinguishing examples,** rather than giving all outputs without references. **You may show “good” and “bad” hyper parameters or filter types.** You may also show **different patch sizes as templates for different types of images (e.g. MNIST digits vs. Lena image vs. where’s Waldo). If required you may add other images.**

4 Grading and Submission Guidelines for Part 2 (45%):

4.1 Work Breakdown:

Your work will include three parts:

1. **Preparing a working solution** with code attached (The default submission will run either in Google Colab or in the installed vlab environment (CV2020), with the same software versions as installed in class).
2. **Writing a project report** (see details below).
3. Preparing and submitting a sample inputs and outputs demonstrating the capabilities of your work
4. **Personal interview** – optional, per the course team decision. Details on the first page.

4.2 Submission Package

For each HW exercise / Project you must submit

1. A Project Report Document (details below)
2. <u>**A zip file**</u> containing the following files and folders:
   - The **Project Report** file in Microsoft Word **(.doc / .docx**) format and **pdf** format
   - A **README.TXT** file that contains the student name, and explains how to run your code, including (special / system) requirements, and notes to the instructor or TA.
   - A **code folder**, containing your code – a single notebook is acceptable.
   - A **data\ folder**, with your input and output (results) files (such as images), and any other file you may find relevant for your submission. Should also include intermediate results / images, used in your report.
   - A **ref/** clip/ **optional folder** containing reference papers, codes and other supporting documents or links.

4.3 Grading for The Midterm

- Working code: **23%** (coding style and efficiency will influence grading)
- Written Project Report: **22%.**

4.4 Code Guidelines

- The code must be modular, robust and understandable, correctly formatted as well as with meaningful parameter names. Document your code as necessary.
- Please follow all the software engineering rules that you've learned before.
- Attach your code to the project report's **Appendix**, and to the **code/** (**code**) folder.

4.5 Project Report Guidelines

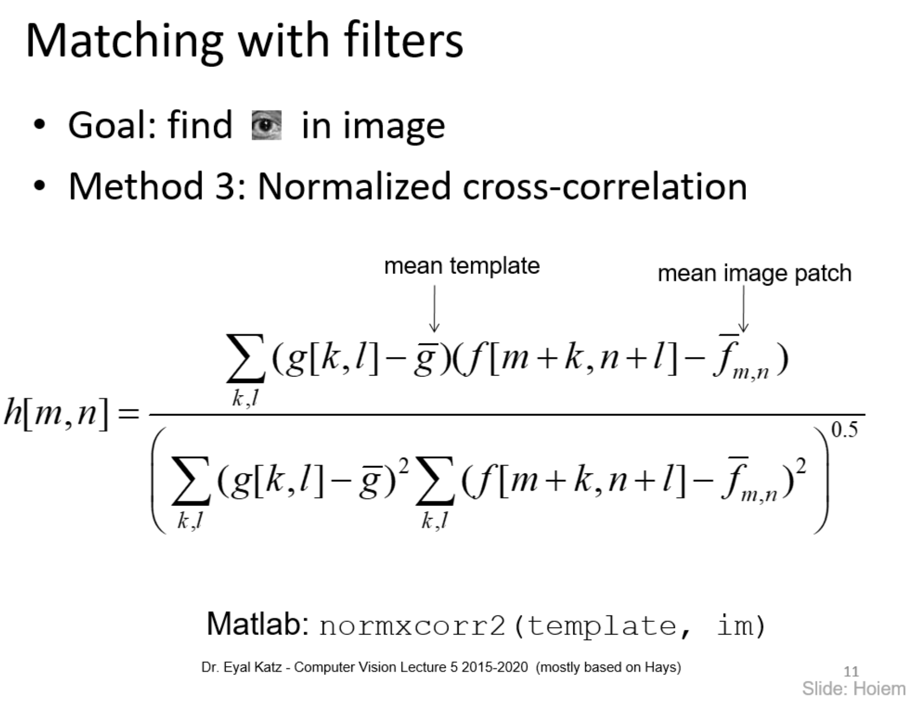
Your Project **Report** should be a Word document (plus a pdf doc), summarizing your work and results. **Note: The reader should appreciate your efforts and achievements, by reading only, thus get motivated to running your code.**
The report should be self-contained (no need to read instructions to understand your work), and include the following chapters:

- (5%) **Title page** with 1-2 images. Should include Afeka logo, the project name (**Midterm**), course name and year, your name, submission date, and **one or two images summarizing your project results**.
- (10%) **Abstract** – **5 lines to half a page** summarizes the problem needed to be solved, the required solution functionality, the algorithm or method, and the results you got. 1-3 images are optional here, provided they are not a repetition of the ones on the title page.
- (15%) **Introduction / Background / Related Work** (existing solutions) – **a few lines to half a page**, explaining the problem to programmers, less proficient in computer vision: Motivation: why solving this problem is interesting and what are the challenges. In existing solutions you may refer to existing code / algorithms you need to implement, replace, or improve. **In this section, please define, using matrices and / or mathematical formulation, the operations, and the hyper parameters' influence.**
- (30%) **The proposed solution / method / Algorithm** – **at least half a page**, of text (a must!!!), flowcharts, and images, explaining the method or algorithm you've implemented, solving the problem, with reference to your code. If relevant, include images for each relevant stage (intermediate results) of the process / algorithm. Show the intermediate results on one typical example only. If you use parameters such as thresholds, show results with correct and incorrect selection of parameters. Try explaining how to select the parameters.
- (20%) **Results – at least half a page** of the input and output images / models. Show your results on ALL images or cases. No need to show intermediate results here. (If you processed a video file or live video stream, put a typical input and output frame(s), which demonstrate your results**. In the html version,** link them to the input and output video files).
- (10%) **Discussion** – here you may expand or explain some of the problems you have encountered, any interesting findings, or open problems and future possible work that may be done to continue your submitted project.
- (10%) **Summary and Conclusion** – **up to 10 lines** – summarize the problem, solution, and achieved results. If applicable, enlighten the uniqueness of your solution.
- **References** – a list of external links and publications you have referred to in your report. If none, please leave the chapter title only. Use the Harvard citing and reference system.
- **(no credit, -20% if does not exist) Appendix**- copy your documented code / notebook in a readable format here.

Enjoy Your Work, **Dr. Eyal Katz and the Course Team**

5 Appendix

- Following are the relevant slides for normalized cross correlation.



Matching with filters

- Goal: find 👁 in image
- Method 3: Normalized cross-correlation

mean template

mean image patch

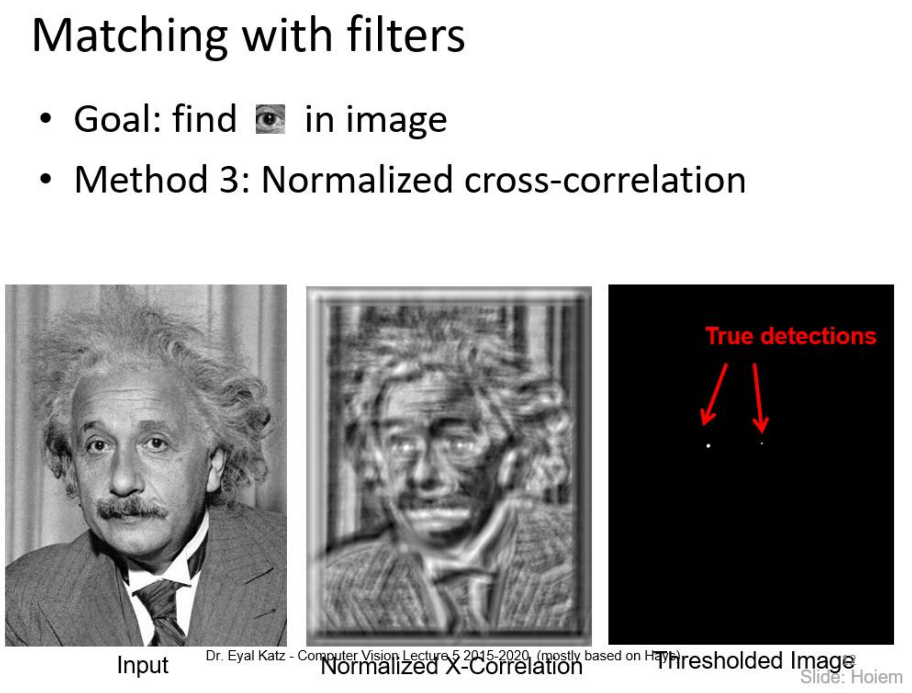
$$h[m,n]=\frac{\sum_{k,l}(g[k,l]-\bar{g})(f[m+k,n+l]-\bar{f}_{m,n})}{\left(\sum_{k,l}(g[k,l]-\bar{g})^2\sum_{k,l}(f[m+k,n+l]-\bar{f}_{m,n})^2\right)^{0.5}}$$

Matlab: `normxcorr2(template, im)`

Dr. Eyal Katz - Computer Vision Lecture 5 2015-2020 (mostly based on Hays)

11

Slide: Hoiem



Matching with filters

- Goal: find 👁 in image
- Method 3: Normalized cross-correlation

Input

Normalized X-Correlation

Thresholded Image

Dr. Eyal Katz - Computer Vision Lecture 5 2015-2020 (mostly based on Hays)

Slide: Hoiem

בהצלחה