Introduction to Image Analysis

HT 2023

Mini-project:
CAPTCHA Classification

It is time to solve a complete image analysis task, relying on your knowledge acquired during the course, combined with your own ideas and creativity!

You will solve **CAPTCHA - "Completely Automated Public Turing test to tell Computers and Humans Apart".**

CAPTCHAs are distorted images containing letters and digits used in cyber-security to determine if the user of a software or website is human or not. The images are designed in a way that makes it difficult for a computer program to read the letters and digits in the images, but it is still possible for a human to determine the content. Your task will be to write a computer programme which can solve one CAPTCHA.

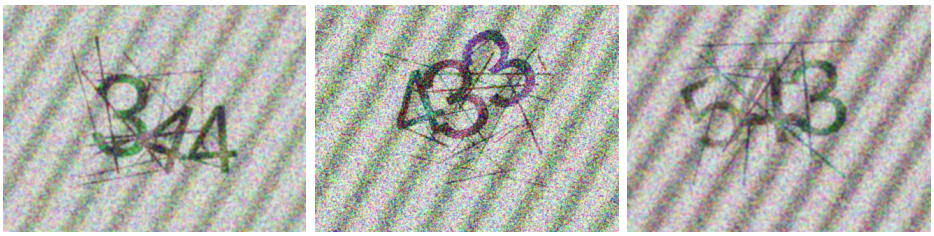
Formalities

- **Team up - one more time**! Join/form groups of two, in Studium (under People → Mini-project teams). Now, you may form the teams with any of the course participants, including those you have already collaborated on one of the computer exercises. Initiate communication as soon as possible and **collaborate within your team!**
- **You are expected to solve the task, to prepare a written project report and to orally present your solution and results.** One report is required per team. **Both** team-members are expected to participate in the presentation.
- **Your report should follow the template** provided in Studium, under 'Modules' → 'Labs - general' → 'Lab report template' (you are welcome to adjust its name to "Mini-project report"). The suggested content is included below, in this task description. More details about the **oral project presentation** will be provided in due time in Studium.
- Please, hand in your report via Studium.
- Follow the status of your reports in Studium, under 'Assignments' and 'Grades'.
- **Evaluation of the submitted solutions: Once a day, until December 12, 16:00. Final (mandatory) evaluation for all: Wednesday, December 13, 16:00.**
- **Report submission deadline: Draft by December 13, 2023, 18:00; Final version by December 22, 20:00.**

Getting started

Download the project material from Studium, 'Modules' → 'Mini-project' → 'Mini-project Assignment.

1 Project Description



(a) (b) (c)

Figure 1: Some examples of the CAPTCHA images you will work with.

The project aims at classifying the individual digits in a set of provided CAPTCHA images. Figure 1 shows some examples of the type of images you are going to work with. Your task is to write a function that takes a CAPTCHA image as input and provides, at the output, a vector containing the three digits in that image. The provided material includes a training set (which you may further divide into training and validation sets) consisting of 1200 images with corresponding labels. The end goal is to apply your solution to classify the images in a held-out test set, which you do not have access to – this part will be performed by us.

Requirements for your classifier are:

1. Classification of all the images in the provided training set should take less than 6 minutes (excluding the time needed to train the classifier).
2. Your classifier should reach at least 50% classification accuracy on the held out test set.
3. Your solution must not include/rely on the MATLAB function `ocr` (nor on any readily available digit classifier). You may wish, however, to compare results of your solution with those obtained by `ocr`.
4. You are allowed to use existing generic components, such as general classification and feature extraction functions in MATLAB.

Start with visual inspection of the images provided in the training set (in the `Train` directory). Try to identify the main challenges your classifier will need to address.

1. ***What are the the main properties of the images that will affect your solution?***
2. ***Which (main) challenges do you envision? What are your initial ideas how to tackle them?***

Include the answers to these questions into you report.

The provided material includes:

- skeleton-script for your classifier, `myclassifier.m`
- evaluation script, `evaluate_classifier.m`
- `labels.txt` file (in the `Train` folder).

You should use `myclassifier.m` to build your own solution. Note that the methods you have learned about during the course and practiced during the computer exercises are sufficient to complete the task, but you are welcome to use some other - excluding deep neural networks - if you find that suitable. Your original ideas how to solve the task are very welcome!

The evaluation script, `evaluate_classifier.m`, loops through the images in the training set, classifies each image and calculates the overall accuracy. You should use it during your method development, to evaluate your current solutions (you are encouraged to split the data so as to utilize a validation set). Your submitted `evaluate_classifier.m` script should be executable after the path to the data is provided – please, make sure that this is the case.

The `labels.txt` file consists of four columns. The first one contains the image number and the following three include the labels for the three numbers appearing in that image.

We will evaluate your (via Studium) submitted classifier(s) on the held-out test data and inform you (in a comment in Studium) about the score we got. You are allowed to submit one solution per day for evaluation, until Dec 13 (as stated in **Formalities**). Furthermore, to make the project a bit more exciting, all results reaching above 50% accuracy will be posted on a high score list in Studium, in the Mini-project Module. You might wish to choose a Team Name for this ranking - let us know about it, when you submit your solution for evaluation (otherwise we will use the team numbers). If you do not want your final results to be displayed, please let us know (in a comment to your submitted solution)!

Last day before the project presentations, on Dec 13 (at 16:00 latest), you should submit your best classifier for the final evaluation on a by-then-held-out test set. These results will be announced during the presentation days and should be included in the final version of your project report.

Your final report you should include:

- **Introduction**
  - Briefly describe the problem.
  - Report on the results of your initial data analysis and include responses to the two questions stated above.
- **Method**
  - Describe the approach you have taken to solve the problem. Motivate all the steps and provide detailed explanations of each of them.
  - When suitable, use references, with motivation and explanation for using the selected approaches from the literature.
- **Implementation**
  - Comment the main details of your implementation. Include your final MATLAB files: your final classifier `myclassifier.m` and `evaluate_classifier.m` which we can use to run and evaluate your classifier on the test data. The `myclassifier.m`

will have access to the training data (in the `Train` directory), in case you are relying on that for a learning-based/template-based approach. Remember to attach any auxiliary files, if you rely on such.

In other words, **we should be able to run your code, in a directory containing two sub-directories, `Train` (same as you have) and `Test` (with a similar layout as `Train`), and get classification results with essentially *no* changes of your code.**

- **Results**
  - Report the accuracy you have achieved on your training and validation sets.
  - Provide information about the required training and prediction time.
  - Include any other information about the performance of your method that you find relevant.
- **Discussion**
  - Comment on your work and results. Was your initial analysis good and did your initial idea work well? If not – what did you have to change and why?
  - Discuss pros and cons of your submitted solution (including implementation). Are there any issues not addressed well by your method? Any particularly good ideas you would like to mention?
  - Do you have ideas how to further improve your solution?
  - Include any further comments you may find relevant.
- **References**
  - List the main references, in case you used any.

Do not hesitate to ask questions, if you face problems!