Cairo University
Faculty of Engineering
Computer Engineering Department

Pattern Recognition and Neural Networks

Project Document

Handwriting Based Gender Classification

Spring 2022

Contents

1 Objectives

This project aims to put your understanding of machine learning algorithms into practice with a real world problem. **By the end of this project you should be able to:**

1. Analyze the problem, extract features, and choose the most appropriate preprocessing method (if necessary).
2. Assess performance of different machine learning techniques.
3. Compare and combine different machine learning components into one big solution.
4. Design and implement your own ML pipeline.

2 Project Description

In this project, you are required to implement a **Handwriting-Based Gender Classification System**. Given an input image for a piece of text; the target is to identify the gender of the writer whether male or female. Human handwriting is used to examine and investigate human characteristics in a variety of applications, such as bank check verification, personality profiling, historical document analysis, and criminological investigations.

You should implement a complete machine learning pipeline, *i.e.* the project should include (but not limited to) the following modules:

- Pre-processing Module.
- Feature Extraction/Selection Module.
- Model Selection and Training Module.
- Performance Analysis Module.

You will need to research about the topic, read research papers that tackle this application and similar applications, and do a literature survey that can help you identify the best approaches/ techniques that you can start with in order to improve the accuracy of your results.

You will work with a collection of images of English text segments. The collection consists of two datasets:

1. The **(ICDAR2013)** dataset used for the competition organized in the scope of the Twelfth International Conference on Document Analysis and Recognition. This dataset contains images of English and Arabic text paragraphs produced by about 400 writers: 46 % of them are males and the rest are females. Every writer produced four images: two in English and two in Arabic. We will work on the English portion of the dataset. The dataset

can be found here:

`https://www.kaggle.com/c/icdar2013-gender-prediction-from-handwriting/data`

2. CMP2023 dataset: You will build your own dataset containing images for English text segments written by CMP2023 students. Details of building the dataset will be discussed in class.
You will divide the data into:

- **Training set** to train your model.
- **Validation set** to tune the parameters of your model.
- **Test set** to report the results of your models.

3 Team Formation

A team should be formed of 4 to 5 members. All team members should attend the final project discussion. If one team member failed to show up in the project discussion, then they are regarded as zero contributors in this project.

4 Final Deliverables

By the end of this project, each team should submit the following:

1. A **PDF report** named [team number]-report.pdf that includes the following sections fully and clearly described, while describing all the work done and approaches adopted. You should include also the unsuccessful trials.
   (a) Literature review summarizes the approaches, features, and machine learning techniques used to tackle the handwriting-based gender classification problem in the papers you have read (3 papers or more).
   (b) Detailed description of your system including:
      i. Project Pipeline.
      ii. Preprocessing Module.
      iii. Feature Extraction/Selection Module.
      iv. Model Selection/Training Module.
      v. Performance Analysis Module.
      vi. (Optional) Any other developed modules.
      vii. **(BONUS 2 Marks)** A formal comparison between the modules tested out in the experiments and the criteria used for choosing the best one.

(c) Enhancements and Future work.

(d) Work Load Distribution

2. **Code** as a zipped folder with the format code [team number].zip containing all code developed with a *readme* file including all packages or libraries needed to run this code. It is highly recommended to provide an executable file beside the source code.

5 Rules and Instructions

All students must follow the following rules and instructions:

1. All projects are submitted online and discussed on premise in the classroom.

2. Don't print any document or report. All submissions are electronic.

3. There will be a late penalty for the final project submission.

4. Any sign of cheating or plagiarism **will not be tolerated.** If one team got caught cheating or plagiarizing from another team, both teams will receive a **ZERO** for the project grade, in addition to a penalty up to 50% of the project grade.

5. Workload should be distributed fairly and equally. Team members who did not contribute effectively in the project will be penalized.

6. All team members should attend the final discussion. A team member who fails to show up will get a ZERO in the discussion grade.

6 Grading Criteria

This project is a competition-based one. This means the better you work, the higher your grade! The grading criteria of this project is divided as follows:

1. **Report and Discussion: (20%):** This point includes the report submission, the quality of the report and the discussion with all team members. It also evaluates how all team members fully understand the details of the project and can elaborate on the examiner questions.

2. **Project Performance (80%):** Both results accuracy (65%) and running time (15%) will be taken into consideration by the given weights. However, **it's not a linear formula**; the ranking procedure will be based primarily on these two factors. Take care that your language choice might affect the running time. Don't rely only on the first "good" approach you find in a paper.

7 FAQ

1. **What programming language(s) can I use?**
   You may use any programming language of your choice!

2. **Is there any restriction on the techniques or approaches to use in any phase of the project?**
   You are free to use any approach or technique you find appropriate for the problem in hand. It's actually your task to find out the best combination of techniques that will yield the best results. However, you are **limited only to classical machine learning methods** such as *Bayesian Classifiers, KNN, Linear/Logistic Regression, Neural Networks (with two hidden layers as a maximum), Support Vector Machines, Principal Component Analysis, etc.*
   You are NOT ALLOWED to use **deep learning** techniques *e.g. Convolutional Neural Networks, Recurrent Neural Networks, etc.*

3. **How to find research papers for this topic?**
   There are many resources on the web tackling this problem. You can start by creating an account on EKB (Egyptian Knowledge Bank) with your university provided student email address, which will give you wide access to a huge number of research papers and journals. You can also use the Academic search engines such as Microsoft Academic, Semantic Scholar, or Google Scholar) a lot of articles are free for public.

4. **How will our project be tested?**
   We will test with our own test set (you have not seen before) in order to standardize the way all teams are graded and ranked. However, you should divide your dataset into Training, Validating and Testing set. The test set is identical to the **CMP2023** database but it is not directly extracted from it.

5. **How will our project be ranked?**
   Please refer to the Grading Criteria section for the grading criteria. The teams will be ranked according to some formula comprising the results accuracy as well as the time taken to generate the results, with the larger weight for the results accuracy. For example, Team X had results with accuracy 92.0%, with running time = 5 seconds, will be ranked above team with accuracy 80.0%, with running time = 2 seconds.

8 Delivery Instructions

8.1 Input Format

The project will be evaluated using our own made test set. The test set is identical to the images in the **CMP2023** dataset. You will receive a folder named data. Under this directory, you will find N-test images in *.png* format named **001, 002, 003..100, 101, 102....etc**

The hierarchy can be visualized as follows (assuming 5 test samples):

- **\data**
  - **001.png**
  - **002.png**
  - **003.png**
  - **004.png**
  - **005.png**

8.2 Output Format

You should generate **TWO** text files:

1. **results.txt**
2. **time.txt**

Make sure you read the test data in the order they are given since you need to output the values in the exact order.

8.2.1 Results

In the first file **results.txt**, you should output the prediction of the model for every test case of the N test cases each in a single line. The prediction of the model is either 1 for male and 0 for female.

The format of this file should be as follows:

```
1
0
0
1
0
1
0
⋮
1
0
```

This means that: For the first test sample **001.png**, your model predicted it belongs to **class 1: male**, whereas in for the second test sample **002.png**, your model predicted it belongs to **class 0: female** , and so on and so forth.

Important Notes:

1. Make sure that you output the result for the test cases in the same order as the input. This mistake is very common as you might think you are ingesting the test cases in order, however, you find you are reading the test cases in an incorrect way that you are processing folder 10 before 01 and so on.
2. This file should not include anything other than the model predictions as indicated.
3. Each line contains exactly **ONE** prediction
4. Do not include the test case number.
5. Do not include any extra information.
6. Again, Do not print all the test cases in one single line.

8.2.2 Time

In the second file **time.txt**, You should output the time taken (in seconds) by every iteration of the N iterations each in a single line.

The format of this file should be as follows:

```
30.00
35.25
32.32
40.15
35.12
38.98
26.23
⋮
20.65
```

This means that: for the first image, your code ran in **30.00 seconds**, for the second image, your code ran in **35.25 seconds** and so on until the Nth image where your code ran in **20.65 seconds**.

Important Notes:

1. For every test image, place a timer **after** reading the training images and the test image, and **after** generating the prediction for the test image. You should output the difference between the two times (i.e. the time taken for a single image to be processed in the complete pipeline without any I/O overhead).

2. The numbers in the above example are just random numbers and have nothing to do with the actual running times. These numbers are **NOT** a reference and they are just for illustration.

3. The running times should be rounded to **two decimal places**.

4. This file should not include anything other than the model running times as indicated.

5. Do not include the test case number.

6. Do not include any extra information.

7. Do not print all the test cases in one single line.

8. Do not print the results in the console. You have to generate the two files results.txt and time.txt