Cairo University
Faculty of Engineering
Computer Engineering Department

Pattern Recognition and Neural Networks

Project Document

Arabic Font Recognition

Spring 2024

Contents

1 Objectives **2**

2 Project Description **2**

3 Team Formation **3**

4 Bonus: Research Paper **4**

5 Final Deliverables **4**

6 Rules and Instructions **5**

7 Grading Criteria **5**

8 FAQ **6**

9 Delivery Instructions **7**

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) and assess the performance of different machine learning techniques.
2. Be able to explore the literature to experiment with different methods and potentially contribute with novel methods.
3. Be able to design and implement your own ML pipeline for real-world problems.

2 Project Description

In this project, you are required to implement an **Arabic Font Recognition System**. Given an image containing a paragraph written in Arabic, your system is supposed to classify the paragraph into one of four fonts (from 0 to 3).

Font Code	Font Name
0	Scheherazade New
1	Marhey
2	Lemonada
3	IBM Plex Sans Arabic

Your system should try to handle variations as much as possible. You should implement a complete machine learning pipeline, *i.e.* the project should include (but not limited to) the following modules:

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

Preprocessing module should handle different paragraphs formats in images like:

- Text alignment
- Text size/weight
- Image blur
- Brightness variation
- Text color
- Text rotation
- Salt and pepper noise

You will be expected to research the topic, read research papers that tackle this problem and similar problems, and do a literature survey that can help you identify the best approaches or techniques that you can start with in order to improve the accuracy of your results.

The dataset can be downloaded from `https://www.kaggle.com/datasets/breathemath/fonts-dataset-cmp`.

You will divide the dataset into:

- **Training set** to train your model.
- **Validation set** to tune hyperparameters and choose between models.

We will be providing a **test set** to report the results of your models.

You are **required** to deploy your model on a hosting service (e.g., Azure or Heroku), so that it can be used by others to determine the Arabic font provided in a screenshot. It suffices if this is only a server with no GUI.

3 Team Formation

A team should be formed of 3 to 4 members. No team should have more than 4 members under any condition. Please register your team number in the provided spreadsheets. 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 Bonus: Research Paper

Optionally, one of the teams can work on an innovative research idea related to the project problem. The team will be directly supervised by the course instructor, and their main goal is to **publish a research paper** on a new contribution related to the project idea. The team **will not be a part of the competition** and their grades will not be affected by the final leaderboard. However, they will be graded based on their work on the research paper.

The teams interested in working on the research project should **submit a proposal** to the course instructor explaining their motivation and proposed contribution. The proposals will be reviewed, then the selected teams can start working on their ideas.

Grading for the research paper will be out of 130% of the project's weight, if your work gets the full mark you will be awarded with 30% bonus grades.

5 Final Deliverables

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

1. A **PDF report** that includes the following sections fully and clearly described, while describing all the work done, approaches adopted and results obtained. You should include also the unsuccessful trials.

   (a) Project Pipeline.

   (b) Preprocessing Module.

   (c) Feature Extraction/Selection Module.

   (d) Model Selection/Training Module.

   (e) Performance Analysis Module.

   (f) (Optional) Any other developed modules.

   (g) Enhancements and Future work.

   The report should include also a workload distribution between team members.

2. **Code** as a zipped folder with the format code [team number].zip containing all code developed with a *README* explaining in detail how to run the code (training or inference via one of the models). A *requirements.txt* file or equivalent should be available to sort out any needed dependencies. It is highly recommended to provide an executable file beside the source code.

3. **Model file** for the final model should be included with your code and in the correct path to execute the code. This model should be loaded into your code upon execution and used to make test predictions with the *testing script* mentioned later in this document. It's advised to store the model file as a `pickle` file.

4. **Server test file** which will be a simple Python script that, given an image, sends a POST request to the server where the model is deployed and returns the prediction. It should be possible to run this file independent of the rest of the code.

6 Rules and Instructions

All students must follow the following rules and instructions:

1. All projects are submitted on 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.
7. Each team member should be able to deeply and comprehensively explain their contribution to the project.

7 Grading Criteria

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. **Code Review (10%):** The quality of the written code and the structure of the project will be taken into consideration by comparing with industry standards.
3. **Project Performance (65%):** Both results accuracy (55%) and running time (10%) will be taken into consideration by the given weights. However, it's not a linear formula but the ranking procedure will be based only on these two factors. Take care that your language choice might affect the running time. This project is a competition-based one.

4. **Model Deployment (5%):** The model shall be tested on arbitary input and show the output in an appropriate manner.

8 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 or shallow networks** such as *Bayesian Classifiers, KNN, Linear/Logistic Regression, Support Vector Machines, Principal Component Analysis, Neural Networks (with two hidden layers as a maximum), etc.*

   You are NOT ALLOWED to use **deep learning** techniques or **pretrained** models *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 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 (at least) training and validation sets. The test set is identical to the provided dataset 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.

9 Delivery Instructions

9.1 Input Format

The project will be evaluated using our own test set. The test set is identical to the one provided to you. You will receive a folder named `data`. Under this directory, you will find N test images. These images files will be named **1.jpeg, 2.jpeg, 3.jpeg, ..., 10.jpeg, 11.jpeg, 12.jpeg, ..., 100.jpeg, 101.jpeg, 102.jpeg,**....etc.

The hierarchy can be visualized as follows (for the first two test cases):

- **\data**
  - 1.jpeg
  - 2.jpeg
  - .....

Make sure that you process the test images in an **increasingly ordered sequence**. You will output the results in the same order as the input. The submitted model file will be run on all test images, so you need to make sure that the system can handle different conditions (i.e., preprocess images).

9.2 Output Format

You should generate **TWO** text files:

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

9.2.1 Results

In the first file **results.txt**, you should output the prediction of the model for every test case of the N test cases in a single line. The prediction of the model is a number from 0 to 3.

The format of this file should be as follows:

```
2
3
0
2
1
3
3
⋮
2
1
```

This means that: The font in the first image is classified as 2, whereas the Arabic font in the second image is classified as 3 and so on.

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. You can ensure your script runs correctly by first trying it on training data and reproducing the predictions
3. This file should not include anything other than the model predictions as indicated.
4. Do not include the test case number.
5. Do not include any extra information.
6. Do not print all the test cases in one single line.

9.2.2 Time

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

The format of this file should be as follows:

```
30.00
35.205
32.320
40.150
35.121
38.982
26.233
⋮
20.654
```

This means that: For the first test image, your code ran in **30.00 seconds**, for the second test image, your code ran in **35.25 seconds** and so on.

Important Notes:

1. For every iteration, place a timer **after** reading 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 iteration 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 **three 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