



Cairo University  
Faculty of Engineering  
Computer Engineering Department

# Pattern Recognition and Neural Networks

## Project Document

### Hand Gesture Recognition



Spring 2023

# Contents

1	Objectives	2
2	Project Description	2
3	Team Formation	3
4	Research Team Proposal	3
5	Final Deliverables	3
6	Rules and Instructions	4
7	Grading Criteria	5
8	FAQ	5
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).
2. Assess performance of different machine learning techniques.
3. Design and implement your own ML pipeline.

## 2 Project Description

In this project, you are required to implement a **Hand Gesture Recognition System**. Given an image containing a single hand, your system is supposed to classify the hand gesture into **one of six digits** (from 0 to 5). Your system should try to handle different lighting effects and different **hand poses** 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.**

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 **Hand Gesture of the Colombian sign language** dataset, however you are required to only consider the digit classes (from 0 to 5) for both men and women. The dataset can be downloaded from <https://drive.google.com/drive/u/2/folders/1o9wzwaJVfrbpCFJ0rIyed1QvARh0JAtn>.

You will divide the dataset 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 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 Research Team Proposal

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.

### 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 and approaches adopted. 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* file including all packages or libraries needed to run this code and how to run the code. It is highly recommended to provide an executable file beside the source code.
3. **Model file** 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 do the test predictions, so make sure that your system can handle different conditions. It's advised to submit the model file as a **Pickle** file.

## 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 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. **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 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.

## 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** 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](#) 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 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 images in the **Hand Gesture of the Colombian sign language** dataset. You will receive a folder named **data**. Under this directory, you will find N test images. These images files will be named **1.png, 2.png, 3.png, ..., 10.png, 11.png, 12.png, ....., 100.png, 101.png, 102.png,...**etc.

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

- **\data**
  - 1.png
  - 2.png
  - .....

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 (e.g. lighting and hand pose).

### 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 5.

The format of this file should be as follows:

```

2
3
0
2
1
4
3
:
5
4

```

This means that: The hand gesture in the first image is classified as 2, whereas the hand gesture 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. This file should not include anything other than the model predictions as indicated.
3. Do not include the test case number.
4. Do not include any extra information.
5. 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 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