

00660121 - Medical Diagnostics HW0: Setup

TA in charge: Mattan Hoory*
Faculty of Biotechnology and Food Engineering
Technion - Israel Institute of Technology

Spring 2025

Submission Due: 7.6.25 (Non-mandatory)

1 Introduction

This initial homework assignment is designed to help students enrolled in the course “Medical Diagnostics” (00660121) prepare their environment for working with machine learning tools. The assignment is not for submission but ensures students can run notebooks, install the required tools, and execute a short code correctly. The goal is to identify and mitigate technical barriers before the graded assignments. These will involve both practical tasks (e.g., running and modifying machine learning models) and theoretical questions. They make up a significant part of your final grade, so it’s important to ensure you’re ready!

It is expected that it will be the first time you encounter some of these tools, so this assignment is meant to help you get familiar with them. It is also expected that you will use one of the many freely available online resources and LLMs to help you get started. In any case where something is unclear, feel free to ask for help in the course forum, directly after the tutorial or in the workshops.

*Thanks to Prof. Roee Amit and the Technion Faculty of Biotechnology & Food Engineering for their support.

While the tools listed here represent a recommended setup, especially for those new to machine learning workflows, you're free to use any environment you're comfortable with. The core of this course is not about mastering any one tool or library, but about understanding the principles and reasoning behind machine learning methods which are ideas that remain useful even as technologies change.

HW0 is **not** graded or submitted. While submission is in pairs, all students are expected to complete it before the first tutorial on June 8th.

2 Obtaining the Course Materials

The course materials, including the assignments and tutorials, are available on GitHub.

You can access them at: <https://github.com/mathoory/00660121-medical-diagnostics>

To get the course materials, you can either:

1. Download the repository as a ZIP file:

- Go to the GitHub repository link.
- Click on the green “Code” button.
- Select “Download ZIP”.
- Extract the contents of the ZIP file to a folder on your computer.

2. Clone the repository using Git:

- Install Git if you haven't already. You can download it from <https://git-scm.com/downloads>.
- Open a terminal or command prompt.
- Navigate to the directory where you want to clone the repository.

- Run the following command:

```
git clone https://github.com/mathoory/00660121-medical-diagnostics.git
```

2.1 Working with Git (Optional but Recommended)

If you are not familiar with Git, it is a version control system that allows you to track changes in your code and collaborate with others. Working with Git is recommended for this course, as it helps you manage your assignments and collaborate.

If you want to manage your assignments using Git, here are the basic steps to get started:

1. **Install Git:** If you haven't already, download and install Git from <https://git-scm.com/downloads>.
2. **Configure Git:** Open a terminal or command prompt and set your name and email:

```
git config --global user.name "Your Name"  
git config --global user.email "your.email@example.com"
```

3. **Create a new repository:** Navigate to the directory where you want to create your repository and run:

```
git init
```

4. **Add files:** Add files to your repository using:

```
git add .
```

5. **Commit changes:** Commit your changes with a message:

```
git commit -m "Initial commit"
```

6. **Push to GitHub:** If you want to push your local repository to GitHub, create a

new repository on GitHub and follow the instructions provided there to link your local repository to the remote one.

7. **Learn more:** If you're new to Git, there are many resources available online. A good starting point is the [Git documentation](#) or the [Pro Git book](#). a good tutorial can be found [here](#).

3 Writing Code

For the assignments in this course you will need to write code, while you are scored based on your written report, your code will also be submitted and is part of the evaluation. You can write code in any environment you prefer, but we recommend using a Jupyter notebook or a Python script.

There are three recommended platforms for running code in this course:

1. **Local on your PC** – Conda + VS Code (full control, offline access)
2. **Google Colab** – free cloud notebooks with optional GPU
3. **Kaggle Notebooks** – free cloud notebooks persistent storage and GPU quotas

We recommend to familiarize yourself with more than one of these platforms, as they each have their own advantages and disadvantages.

3.1 What Are Graphical Processing Units (GPUs)?

GPUs are specialized hardware designed to accelerate computations, particularly for tasks involving large amounts of data and parallel processing. They are especially useful in machine learning and deep learning, where they can significantly speed up training times for models.

Generally, GPUs are not required for this course, but they can be helpful for running larger models or datasets. For the most part, any code you write can and will run on a CPU (like any other software or code you use). If you wish to take advantage of the speedup that GPUs can provide, you can use cloud services like Google Colab or Kaggle Notebooks, which offer free access to GPUs.

In any case where a GPU will be required, it will be clearly stated in the assignment. And in such a case we will try to provide suitable options for running the code on a GPU.

3.2 What Are Notebooks?

A notebook file (`.ipynb`) is a document that mixes executable code, text, and figures. You can open notebooks locally (VS Code, Jupyter Lab) or online (Colab, Kaggle). If required, we will provide a starter notebook for each assignment, which you can run and modify. For example, `HW0.ipynb` includes a small executable task to make sure you have everything set up correctly.

3.3 Running Code Locally

Working on your own machine gives you faster file access, full-featured editors, and easier debugging. The recommended setup combines **Conda** for package isolation with **VS Code** as the editor.

3.3.1 Conda

To avoid version conflicts and to keep the course software separate from anything already installed on your machine, we recommend working inside a **Conda virtual environment**. A virtual environment is an isolated copy of Python plus any additional libraries you choose;

it does **not** interfere with your system-wide Python or other projects.

To use your Conda environment, you can either use the terminal to execute code directly on the environment, or through an integrated development environment (IDE) such as VS Code that allows you write the code and the directly run it on the environment.

3.3.1.1 Install Miniconda (Py 3)

Download the latest *Miniconda* installer from <https://conda.io/miniconda.html> and follow the [official instructions](#). After installation, open a new terminal so that the `conda` command is on your PATH.

3.3.1.2 Create the course environment

Each assignment folder (cloned from GitHub) contains an `environment.yml` file listing the exact third-party packages we use:

```
# from the root directory of the assignment  
conda env create -f environment.yml
```

Conda will download the specified versions of Python and all required libraries and register a new environment (e.g. `hw0-00660121`).

3.3.1.3 Activate the environment

```
conda activate hw0-00660121
```

Your prompt will now show the environment name:

```
(hw0-00660121) user@host:~/...$
```

Always confirm this prefix before running notebooks or scripts. Opened a new terminal?

Activate again.

3.3.1.4 Keep it up to date

```
conda env update
```

Run from the assignment root when a new task is released; this synchronises packages with the latest `environment.yml`.

3.3.1.5 Troubleshooting

- **ModuleNotFoundError** or odd import errors usually mean the correct environment is not active. Run `conda activate hw0-00660121` and retry.
- List all environments with `conda env list`.

3.3.2 VS Code

Download from <https://code.visualstudio.com>. VS Code is useful for debugging, Git integration, and group work.

Recommended extensions

- **Python** · **Jupyter** · **Pylance**
- **GitHub Copilot** (free for students)
- **Live Share** (optional for pair work)

3.3.2.1 Running code with VS Code

1. **Open the assignment folder** – *File* → *Open Folder* (hw0-00660121).
2. **Open a terminal** – *View* → *Terminal* (‘Ctrl+‘).
3. **Activate** the Conda environment: `conda activate hw0-00660121`.

4. Run notebooks or scripts.

- `.ipynb` – interactive cells.
- `.py` – whole scripts.

5. Select the correct kernel in notebooks:

```
Python 3.x ('hw0-00660121': conda)
```

3.4 Running Code Online

Cloud notebooks remove local installation hassles and provide on-demand GPUs.

3.4.1 Kaggle

Kaggle is a platform for data science competitions and collaboration, offering free cloud-based Jupyter notebooks with persistent storage and GPU access.

- Sign in at <https://www.kaggle.com>.
- On the left pannel, click on **Create**
- Choose **New Notebook**, or upload an existing notebook.
- You can upload .zip files as datasets by click **Upload**
- To connect to a GPU, scroll down on the righthand side to **Session Options** and select **Accelerator = GPU**. Please note that you are limited to 30 hours of GPU usage per week, so use it wisely.
- By default, if not run is made for 40 minutes, the session will be disconnected and all variables and files will be deleted. You can change this by choosing a **Persistence** that suits you needs.
- Press the **Turn on button** to run the machine and start coding!

3.4.2 Google Colab

A free online Jupyter environment that runs in the cloud. Particularly useful if you do not want local installs or need a GPU.

- Go to <https://colab.research.google.com>.
 - Either open a **New notebook** or **Upload** an existing notebook.
 - Runtime **Change runtime type** *Hardware accelerator = GPU* (if required).
-

Choose the platform that feels most comfortable. The course will evaluate your understanding, not the tool you pick.

More information can be found in the online tutorials for each platform (and on many other online resources such as YouTube, Medium articles, etc.):

- [Kaggle Notebooks](#)
- [Google Colab](#)

4 Self-Check

By the first tutorial, make sure you can:

- ☐ Obtain the course materials from GitHub
- ☐ Open, edit and run a notebook comfortably (see HW0.ipynb provided in the course materials you downloaded or cloned from GitHub) using one of the methods described above

SUPPLEMENTARY MATERIAL

HW0.ipynb: a Colab notebook to test your setup (.ipynb file)

HW0 environment.yaml: Conda environment file to install PyTorch and related packages for local use