Automated Segmentation of Medical Images Using a Convolutional Neural Network

Installation Manual

Vancouver Coastal Health Research Institute

Document Purpose

To describe how to install the components required to run the platform developed to automatically segment regions of interest (ROIs) from medical imaging data.

Table of Contents

Software Installation

Download GitHub Repository

Go to https://download-directory.github.io/

Copy/paste the following link into the URL bar, then press enter to download the project's directory: https://github.com/bendourthe/artificial-intelligence/tree/master/deep-learning/image-processing/medical-autosegmentation-cnn

Once downloaded, extract the corresponding ZIP file to the location of your choice. Keep note of the location of the extracted repository (e.g. '*C:\Users\username\Documents*') as it will be required later.

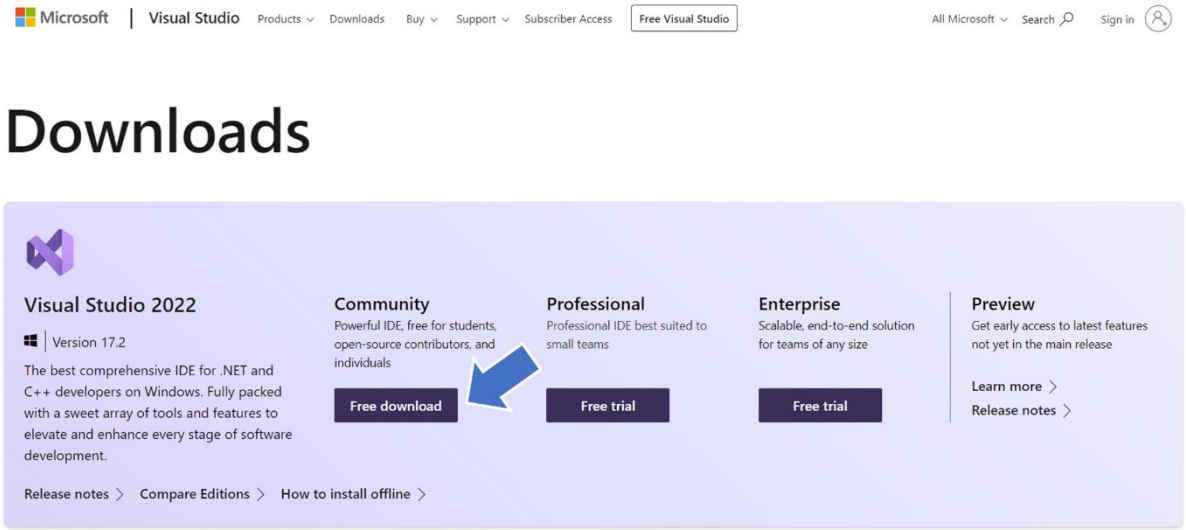
Requirements for GPU Computing

If your computer is equipped with a Graphics Processing Unit (GPU), you will need to install a few software to enable GPU computing, which can significantly speed up the computing process.

Visual Studio Installation

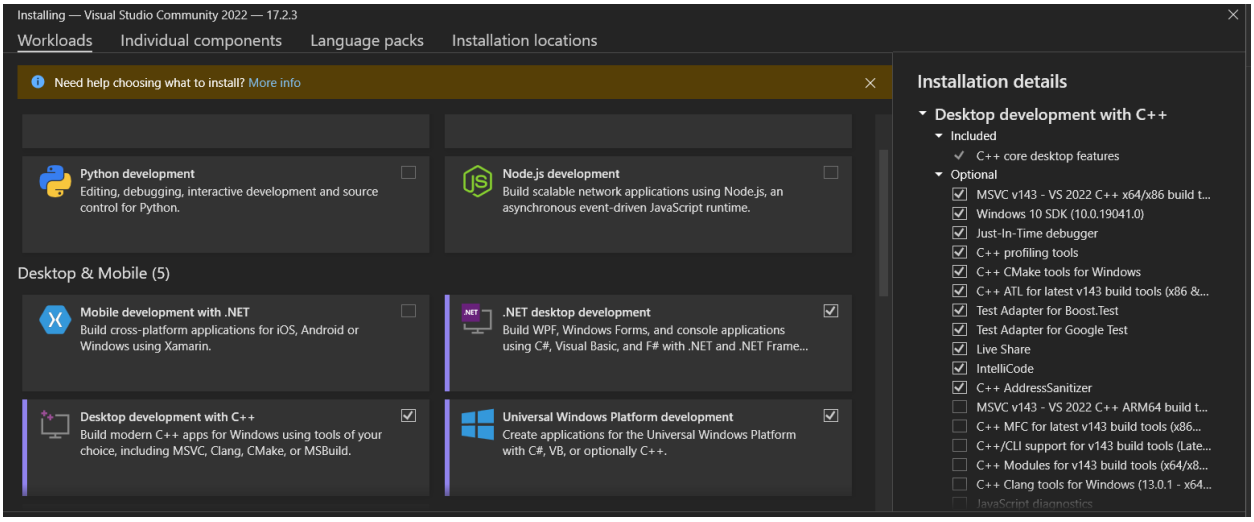
Go to: https://visualstudio.microsoft.com/downloads/

Download the installer for the Community version by clicking on 'Free download':



Once downloaded, launch the executable file. During the installation, when asked to select Workloads, make sure to check the following boxes under the Desktop & Mobile section:

- Desktop development with C++
- Universal Windows platform development
- .NET desktop development



Then, click install and restart your computer when done.

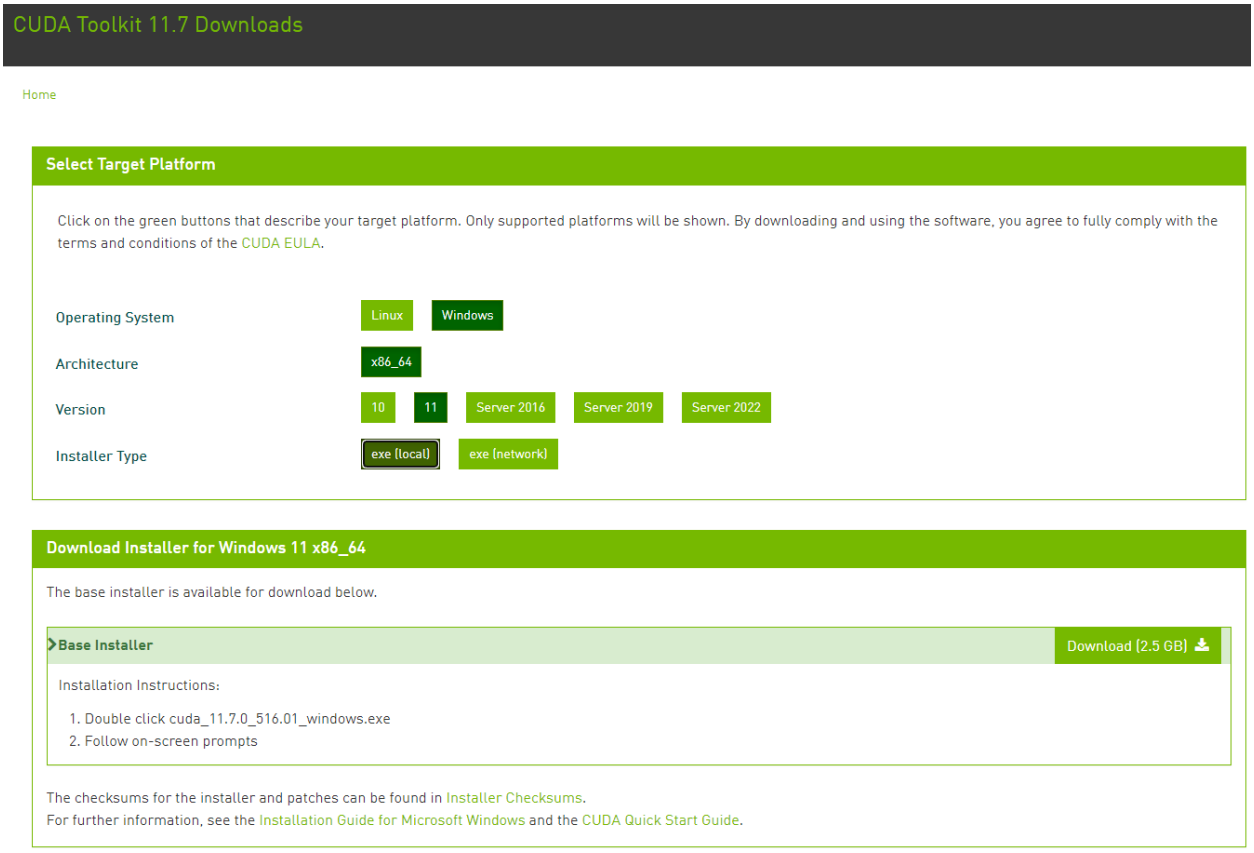
CUDA Toolkit, NVIDIA drivers and cuDNN Installation

CUDA Toolkit

CUDA is a parallel computing platform and application programming interface that allows software to use certain types of GPUs for general purpose processing.

To install CUDA, go to: https://developer.nvidia.com/cuda-downloads

Then, select the right requirements for your system and click the download button:

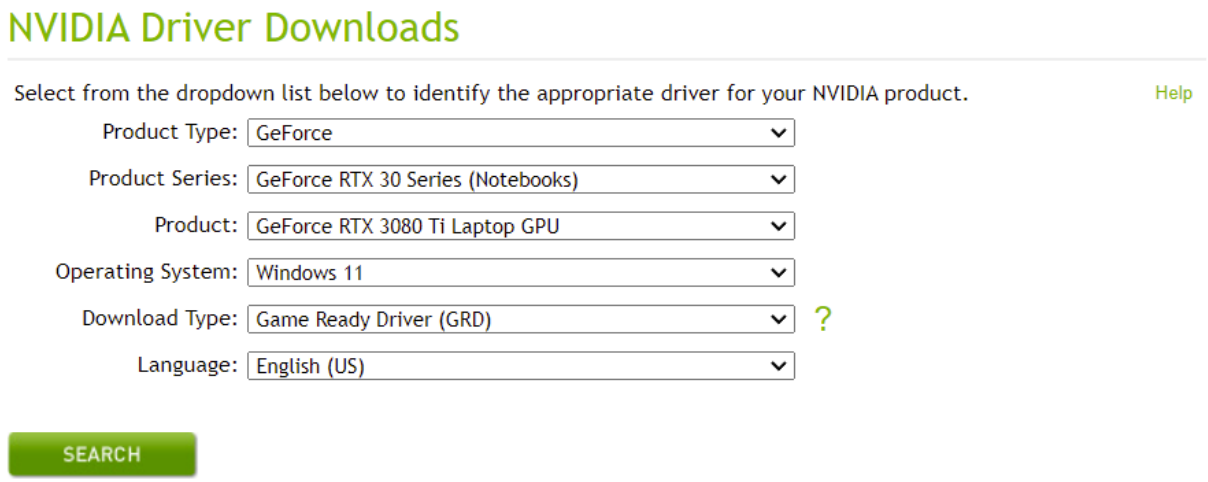


Once downloaded, launch the corresponding executable file, and follow all the default installation steps.

NVIDIA Graphics Card Driver

Go to: https://www.nvidia.com/download/index.aspx

Select the right product and operating system, then click 'SEARCH', and on the next page, click 'DOWNLOAD'.



Once downloaded, launch the corresponding executable file, and follow all the default installation steps.

cuDNN

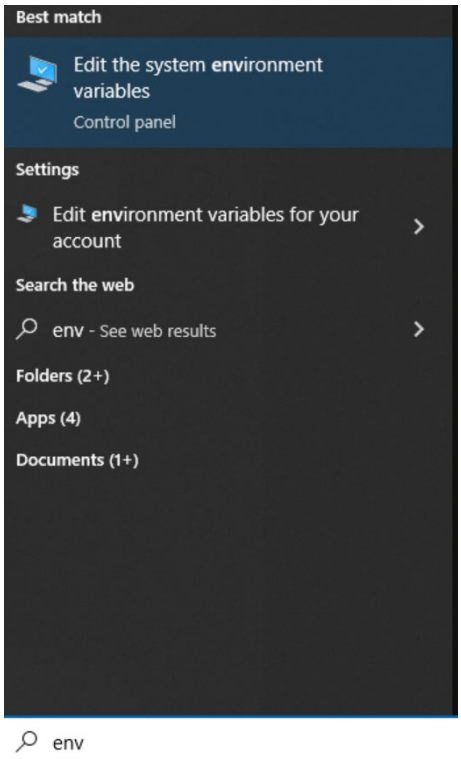
Go to: https://developer.nvidia.com/rdp/cudnn-download

If you do not have one already, you may need to create an NVIDIA developer account (free).

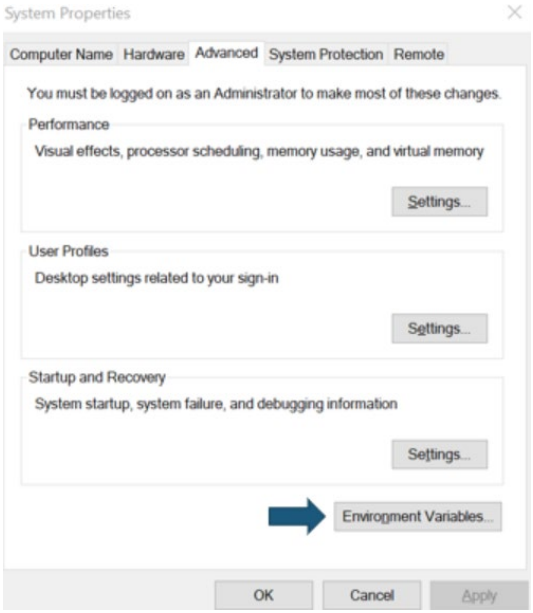
Once done, login and download the zip file for cuDNN. Once downloaded, extract the corresponding zip file to this location: *C:\Program Files\NVIDIA GPU Computing Toolkit*.

Adding CUDA and cuDNN to Environment Variables

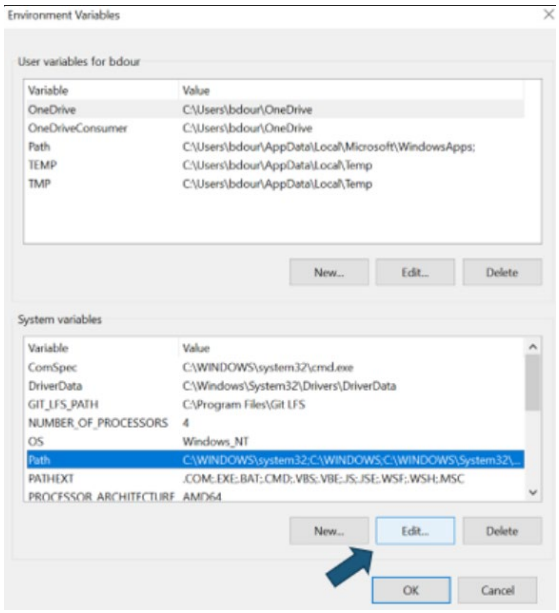
In Windows, click on the start button, then type '*env*', then click on '*Edit the system environment variables*':



This will open the '*System Properties*' window. Click on '*Environment Variables*' in the bottom right:



This will open the '*Environment Variables*' window. Click on '*Path*' under '*System variables*', then click '*Edit...*':



This will open the '*Edit environment variable*' window. Click on '*New*', then paste the path to CUDA an cuDNN bin folders:

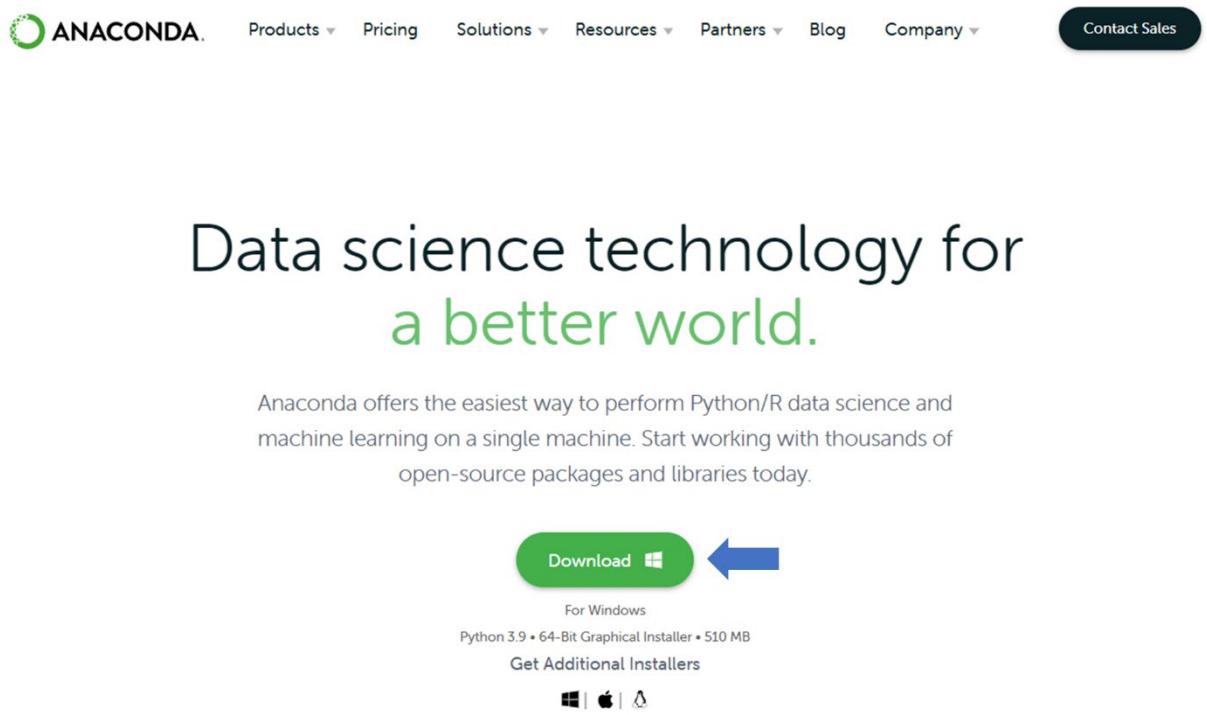
- C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.7\bin
- C:\Program Files\NVIDIA GPU Computing Toolkit\cudnn-windows-x86_64-8.4.1.50_cuda11.6-archive\bin

> ***Note:*** *These paths may vary based on software version and where you decided to extract cuDNN (does not have to be in the specified location, just easier to find in the future if within the NVIDIA GPU Computing Toolkit).*

Python Installation

Anaconda Installation

Go to https://www.anaconda.com/ and click on the '*Download*' button:



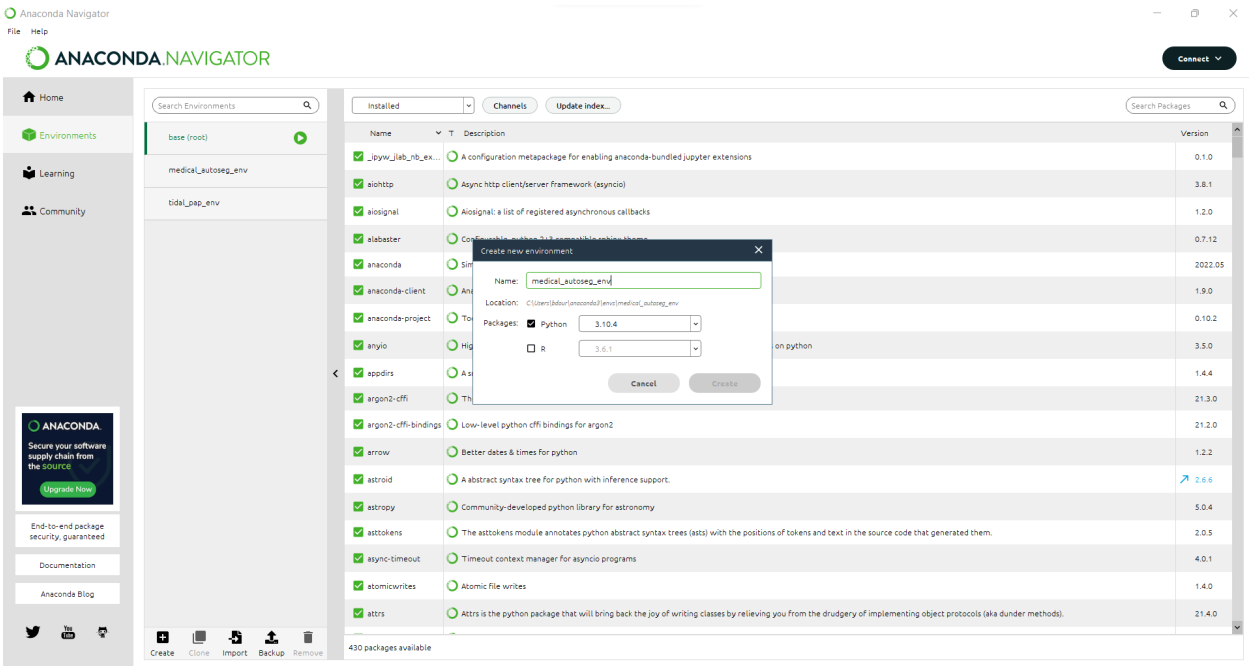
Once downloaded, launch the corresponding executable file, and follow all the default installation steps.

Python Environment Management

To ensure that the platform runs as expected, you will need to create a new Python environment, where the right version of Python and each required library will be installed.

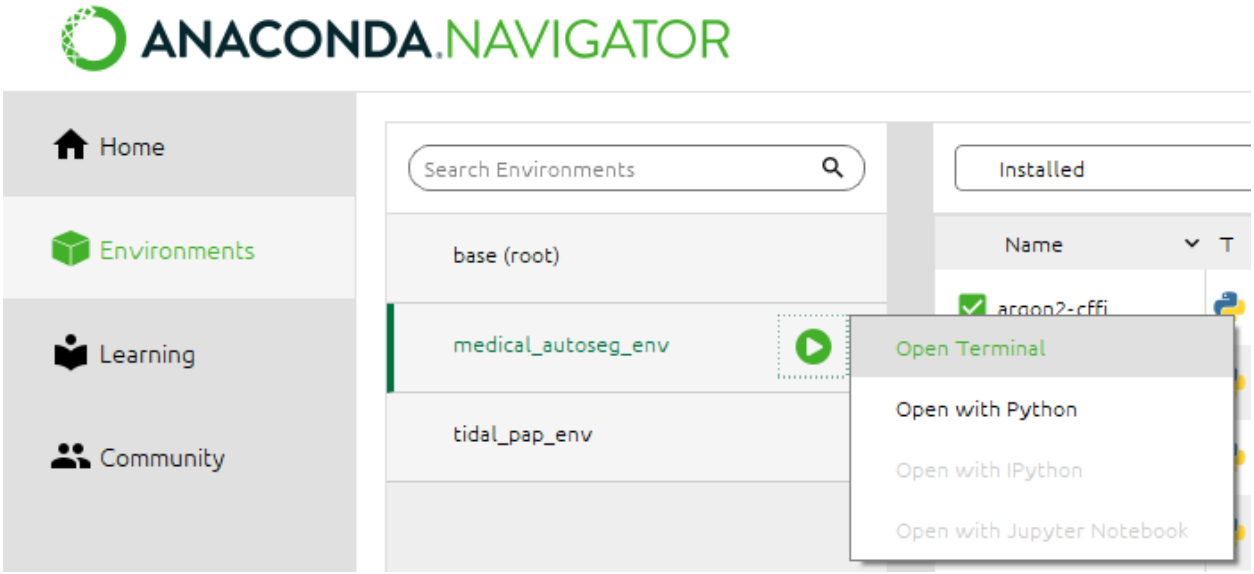
To do so, open the Anaconda Navigator, click on the '*Environments*' tab on the top left corner, then click on the '*Create*' icon in the bottom left of the screen. Give your environment a name (e.g. '*medical_autoseg_env*'), then select the right version of Python (i.e. 3.10), then click create.

It will take a few minutes for the environment to be completely setup.



Python Packages Installation

Once the environment has been completely setup, click on the '*Play*' button next to the environment name, then click '*Open Terminal*'.



This will automatically open a terminal where the environment is active and where your current directory is set to your user folder (i.e. '*C:\Users\username*').

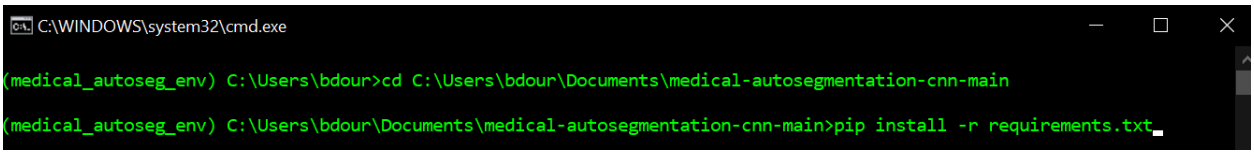
Once in the terminal, you will need to change your current directory to the location of your copy of the project repository. You can do this by entering the following command:

> *cd path*

Where '*path*' is the exact path towards the copy of the repository that you extracted (e.g. '*C:\Users\bdour\Documents\medical-autosegmentation-cnn-main*').

Once in the right directory, enter the following command to automatically install all the packages specified in the requirements.txt file.

> *pip install -r requirements.txt*



This will install most of the basic packages required to run this platform. When done, type the following command in the terminal to install PyTorch:

> *conda install pytorch -c pytorch*

The terminal will ask you to confirm the installation of all the dependencies, type 'y' and press enter to continue the installation.

When done, type the following command to install torchvision:

> *pip install torchvision*

When done, type the following command to add your new environment to Jupyter Lab (so the code can run within this environment in Jupyter Lab):

> *ipython kernel install --user --name=kernel_name*

Where '*kernal_name*' should be changed to the name of the environment that you created (i.e. --name=medical_autoseg_env).

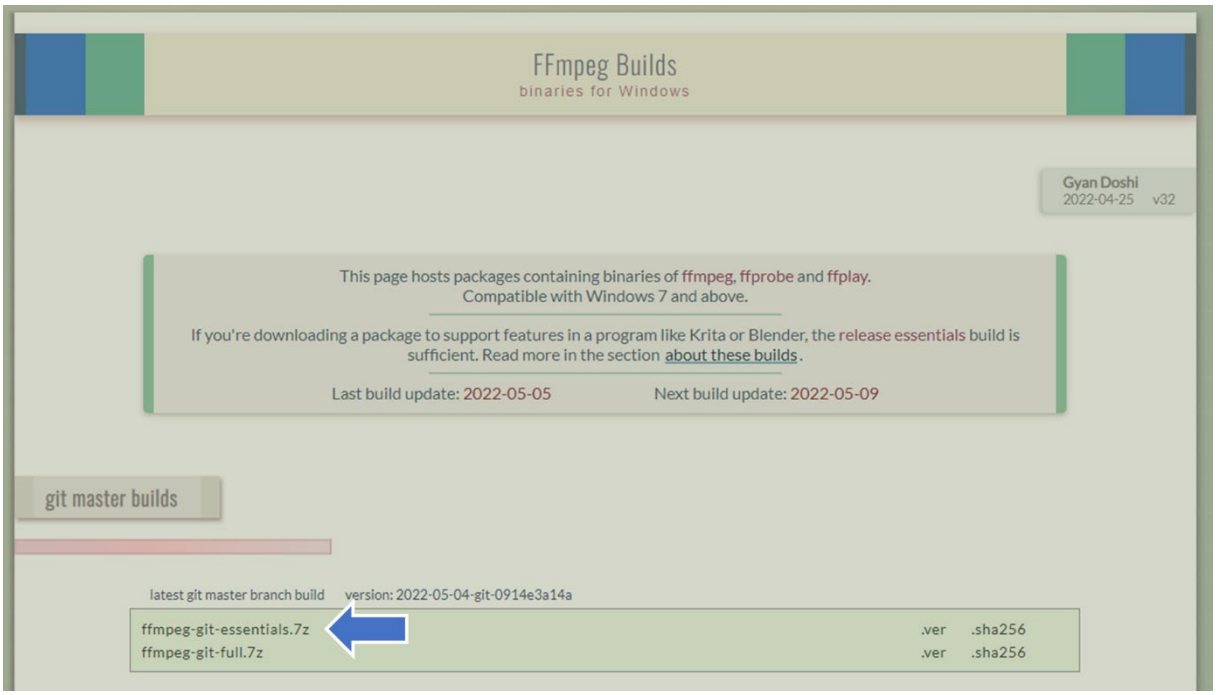
When done, close the terminal.

FFmpeg Installation

FFmpeg is a free and open-source software project consisting of a suite of packages and programs for handling video, audio, and other multimedia files and streams. Although not really necessary to train and use the autosegmentation platform, the testing notebook has the functionality to display an animated figure illustrating each input image along with the resulting segmentation on top, which requires the installation of FFmpeg to be work.

Download FFmpeg

Go to https://www.gyan.dev/ffmpeg/builds/ and download the '*essentials*' release:

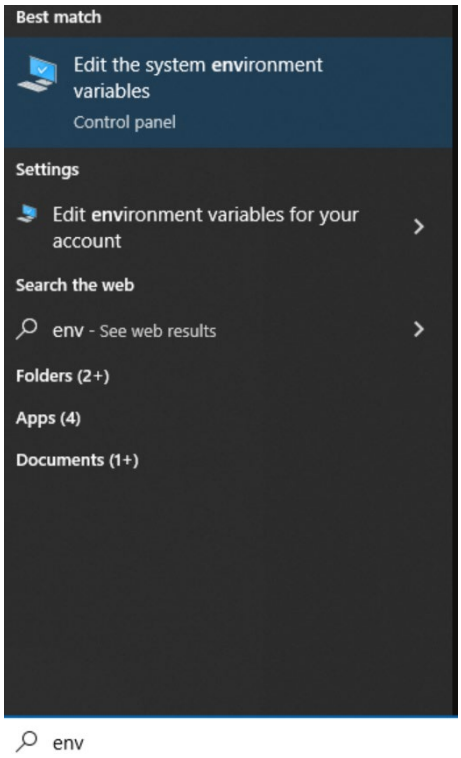


Once downloaded, extract the corresponding file to the location of your choice (e.g. '*C:\Users\bdour\Documents*'), then rename the extracted folder to '*ffmpeg*'.

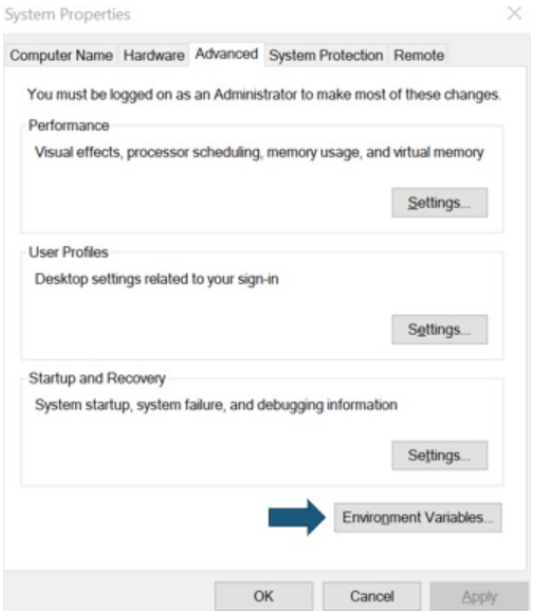
Add FFmpeg to the System's Environment Variables

When done, you need to add the path to the bin folder located in the ffmpeg directory (e.g. '*C:\Users\bdour\Documents\ffmpeg\bin*') to your system's

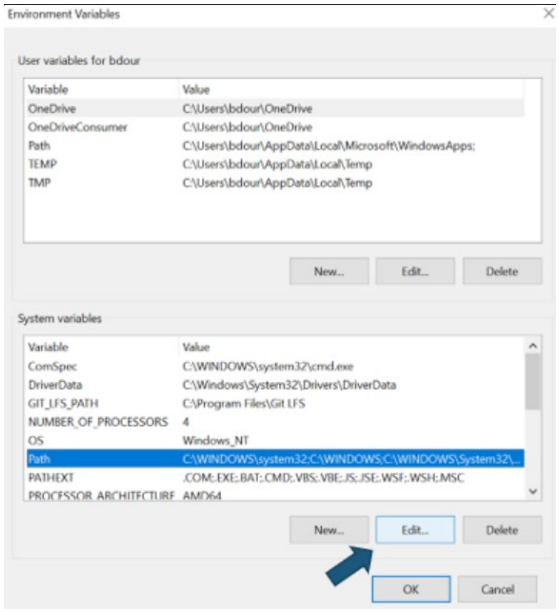
environment variables. To do so, click on the start button, then type '*env*', then click on '*Edit the system environment variables*':



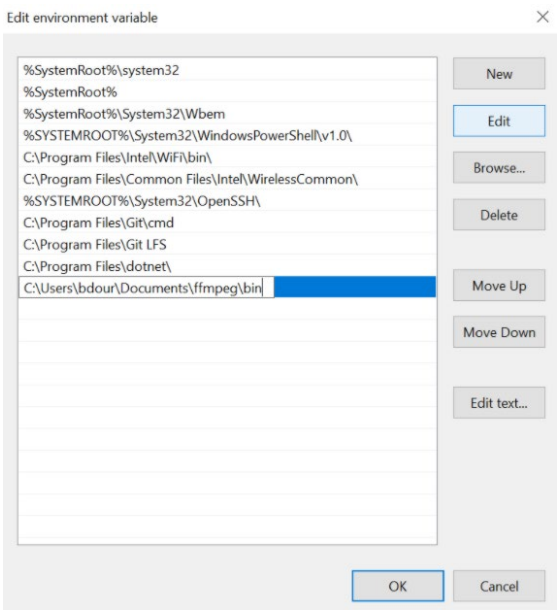
This will open the '*System Properties*' window. Click on '*Environment Variables*' in the bottom right:



This will open the '*Environment Variables*' window. Click on '*Path*' under '*System variables*', then click '*Edit…*':



This will open the '*Edit environment variable*' window. Click on '*New*', then paste the path to the bin folder located in the ffmpeg directory that you extracted:



Click OK for all the windows that were recently open to save this new path in the system environment variables.

Add FFmpeg to Python Environment

In the Anaconda Navigator, click on the '*Environments*' tab on the top left corner, then click on the '*Play*' button next the environment name, then click '*Open Terminal*'.

Copy and paste the command below in the resulting terminal, which will automatically install FFmpeg in your Python environment.

> *conda install -c conda-forge ffmpeg*