UBC

icord

Automated Segmentation of Medical Images Using a Convolutional Neural Network

Installation Manual

Vancouver CoastalHealth
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.

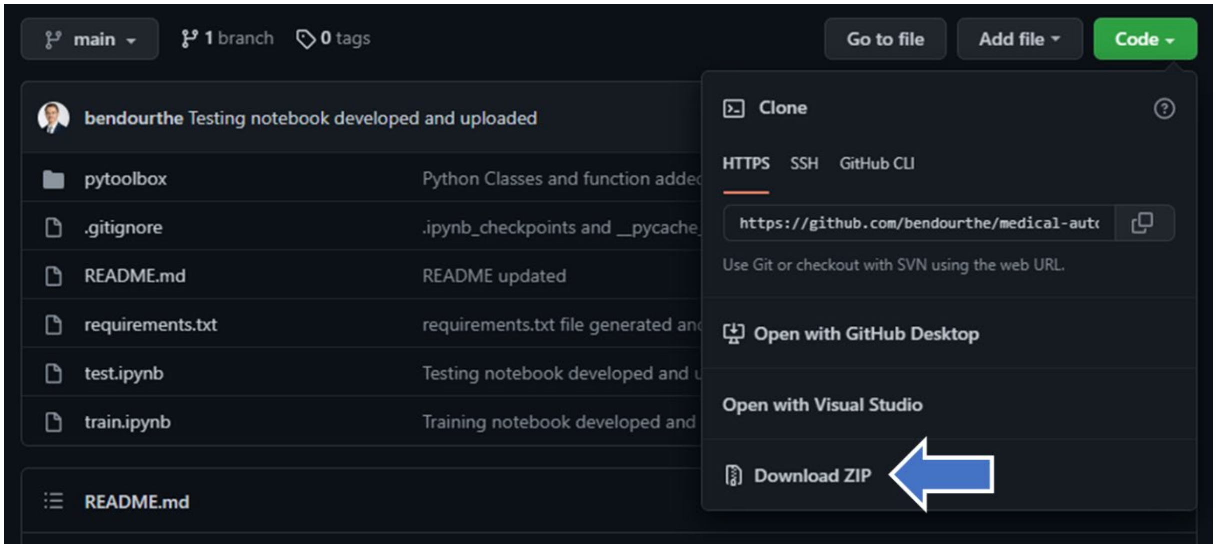
Table of Contents

Software Installation

Download GitHub Repository

Go to https://github.com/bendourthe/medical-autosegmentation-cnn

Click on '*Code*' in the top right corner of the screen and then click '*Download ZIP*':



bendourthe Testing notebook developed and uploaded

pytoolbox	Python Classes and function addec
.gitignore	.ipynb_checkpoints and __pycache_
README.md	README updated
requirements.txt	requirements.txt file generated anc
test.ipynb	Testing notebook developed and u
train.ipynb	Training notebook developed and

```
https://github.com/bendourthe/medical-autc
```

Use Git or checkout with SVN using the web URL.

> *Note: By using the 'Download ZIP' option, the downloaded copy of the repository will not be linked by Git, meaning that any changes that you may make in your local copy of the repository will not affect the main project.*

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.

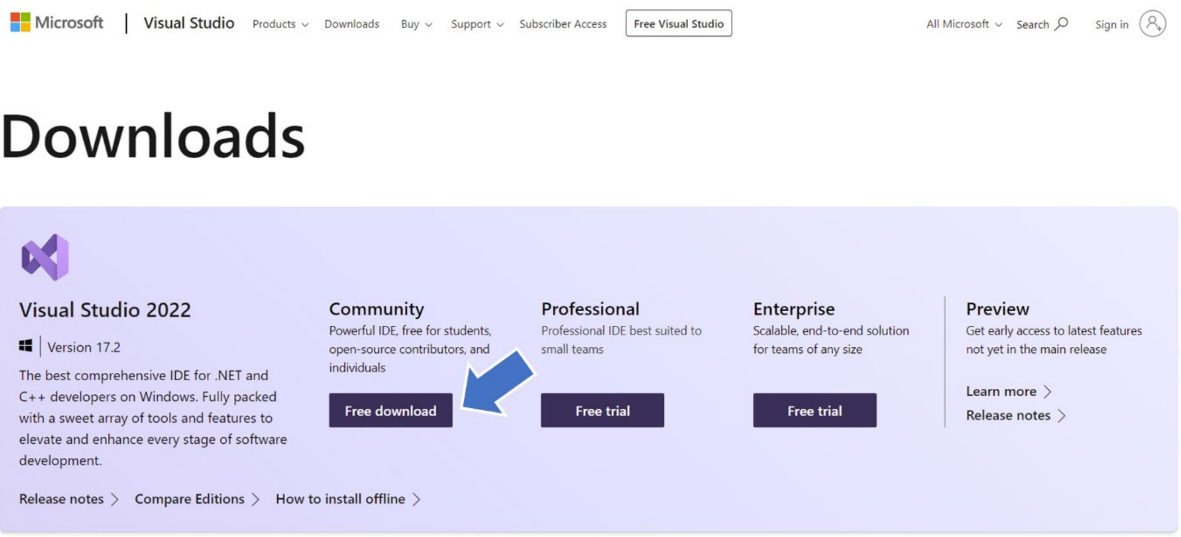
Requirements for GPU Computing

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':



Downloads

Visual Studio 2022

Version 17.2

The best comprehensive IDE for .NET and C++ developers on Windows. Fully packed with a sweet array of tools and features to elevate and enhance every stage of software development.

Community
Powerful IDE, free for students, open-source contributors, and individuals

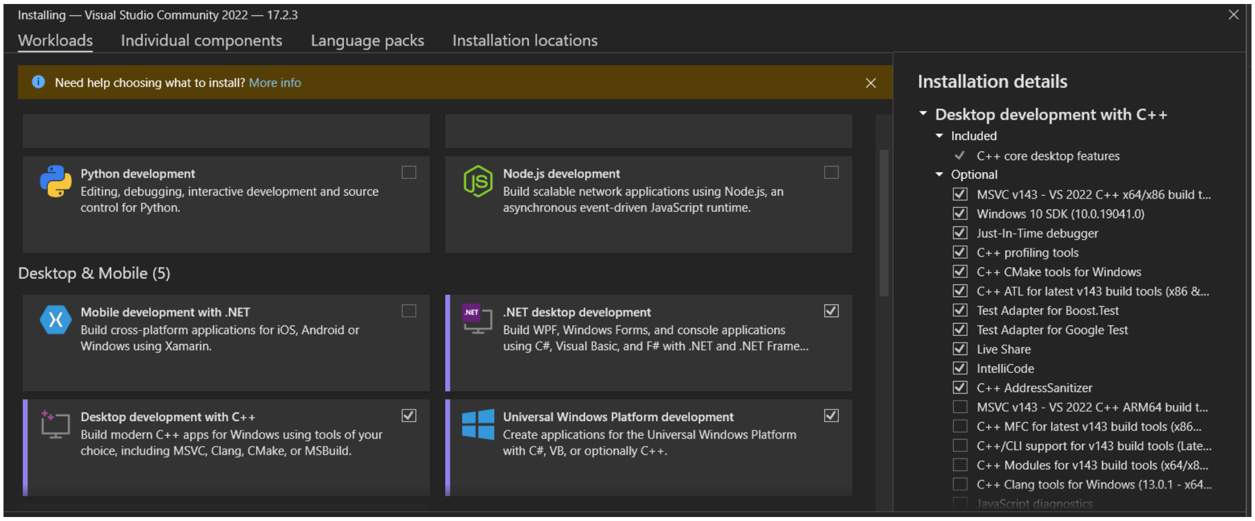
Professional
Professional IDE best suited to small teams

Enterprise
Scalable, end-to-end solution for teams of any size

Preview
Get early access to latest features not yet in the main release

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



Installing — Visual Studio Community 2022 — 17.2.3

Need help choosing what to install? More info

Python development □
Editing, debugging, interactive development and source control for Python.

Node.js development □
Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.

Desktop & Mobile (5)

Mobile development with .NET □
Build cross-platform applications for iOS, Android or Windows using Xamarin.

.NET desktop development ☑
Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET and .NET Frame...

Desktop development with C++ ☑
Build modern C++ apps for Windows using tools of your choice, including MSVC, Clang, CMake, or MSBuild.

Universal Windows Platform development ☑
Create applications for the Universal Windows Platform with C#, VB, or optionally C++.

Installation details

- Desktop development with C++
  - Included
    - ✓ C++ core desktop features
  - Optional
    - ☑ MSVC v143 - VS 2022 C++ x64/x86 build t...
    - ☑ Windows 10 SDK (10.0.19041.0)
    - ☑ Just-In-Time debugger
    - ☑ C++ profiling tools
    - ☑ C++ CMake tools for Windows
    - ☑ C++ ATL for latest v143 build tools (x86 &...
    - ☑ Test Adapter for Boost.Test
    - ☑ Test Adapter for Google Test
    - ☑ Live Share
    - ☑ IntelliCode
    - ☑ C++ AddressSanitizer
    - □ MSVC v143 - VS 2022 C++ ARM64 build t...
    - □ C++ MFC for latest v143 build tools (x86...
    - □ C++/CLI support for v143 build tools (Late...
    - □ C++ Modules for v143 build tools (x64/x8...
    - □ C++ Clang tools for Windows (13.0.1 - x64...
    - □ JavaScript diagnostics

Then, click install and restart your computer when done.

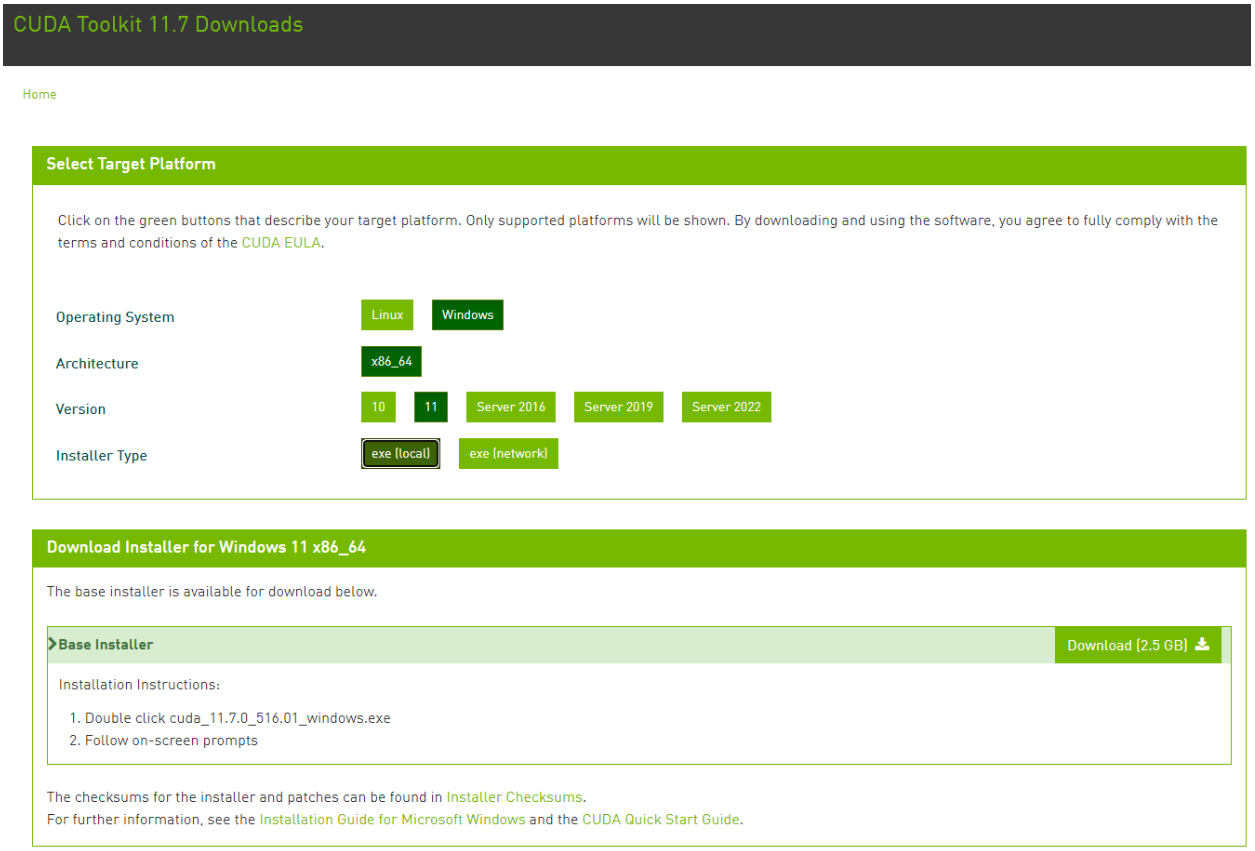
CUDA Toolkit, NVIDIA drivers and cuDNN Installation

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:



CUDA Toolkit 11.7 Downloads

Home

Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown. By downloading and using the software, you agree to fully comply with the terms and conditions of the CUDA EULA.

Operating System: Linux [✓ Windows]

Architecture: [✓ x86_64]

Version: 10 [✓ 11] Server 2016 Server 2019 Server 2022

Installer Type: [✓ exe (local)] exe (network)

Download Installer for Windows 11 x86_64

The base installer is available for download below.

Base Installer

Installation Instructions:

1. Double click cuda_11.7.0_516.01_windows.exe
2. Follow on-screen prompts

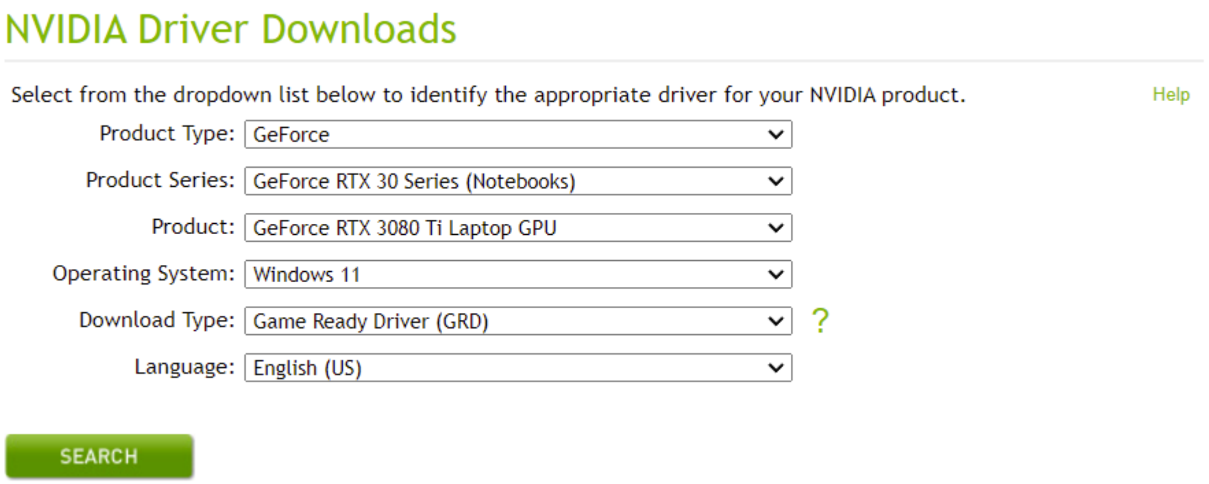
The checksums for the installer and patches can be found in Installer Checksums.
For further information, see the Installation Guide for Microsoft Windows and the CUDA Quick Start Guide.

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’.



NVIDIA Driver Downloads

Select from the dropdown list below to identify the appropriate driver for your NVIDIA product.

Product Type: GeForce

Product Series: GeForce RTX 30 Series (Notebooks)

Product: GeForce RTX 3080 Ti Laptop GPU

Operating System: Windows 11

Download Type: Game Ready Driver (GRD)

Language: English (US)

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

cuDNN

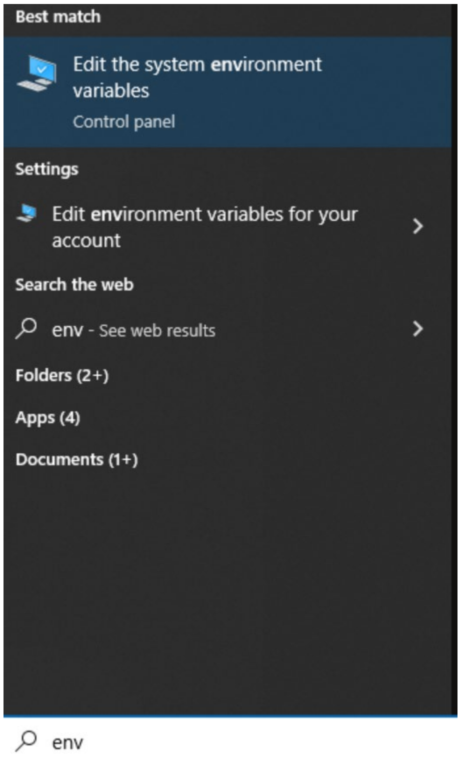
Go to: https://developer.nvidia.com/rdp/cudnn-download

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

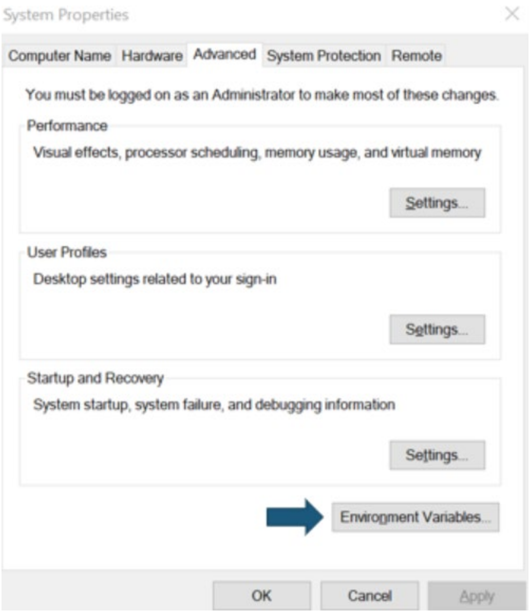
Once done, login and download the zip file for cuDNN. Once downloaded, extract the corresponding zip file to this location: *C:\Program Files\NVIDIA GPU Computing Toolkit*.

Adding CUDA and cuDNN to Environment Variables

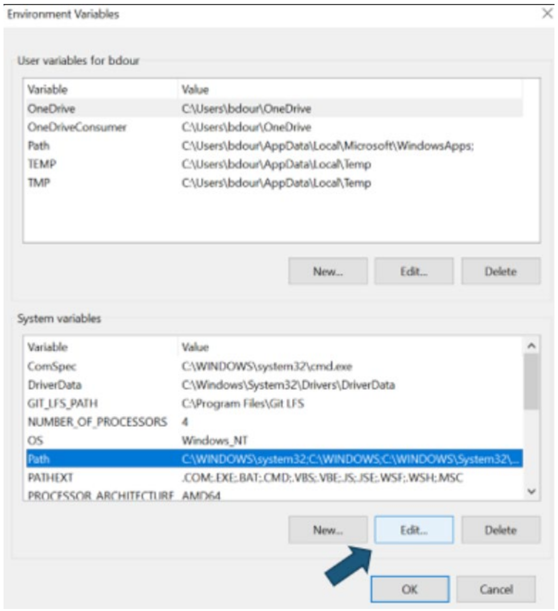
In Windows, click on the start button, then type ‘*env*’, then click on ‘*Edit the system environment variables*’:



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


System Properties
Computer Name
Hardware
Advanced
System Protection
Remote
You must be logged on as an Administrator to make most of these changes.
Performance
Visual effects, processor scheduling, memory usage, and virtual memory
Settings...
User Profiles
Desktop settings related to your sign-in
Settings...
Startup and Recovery
System startup, system failure, and debugging information
Settings...
Environment Variables...
OK
Cancel
Apply

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



Environment Variables

User variables for bdour

Variable	Value
OneDrive	C:\Users\bdour\OneDrive
OneDriveConsumer	C:\Users\bdour\OneDrive
Path	C:\Users\bdour\AppData\Local\Microsoft\WindowsApps;
TEMP	C:\Users\bdour\AppData\Local\Temp
TMP	C:\Users\bdour\AppData\Local\Temp

System variables

Variable	Value
ComSpec	C:\WINDOWS\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
GIT_LFS_PATH	C:\Program Files\Git LFS
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
Path	C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64

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

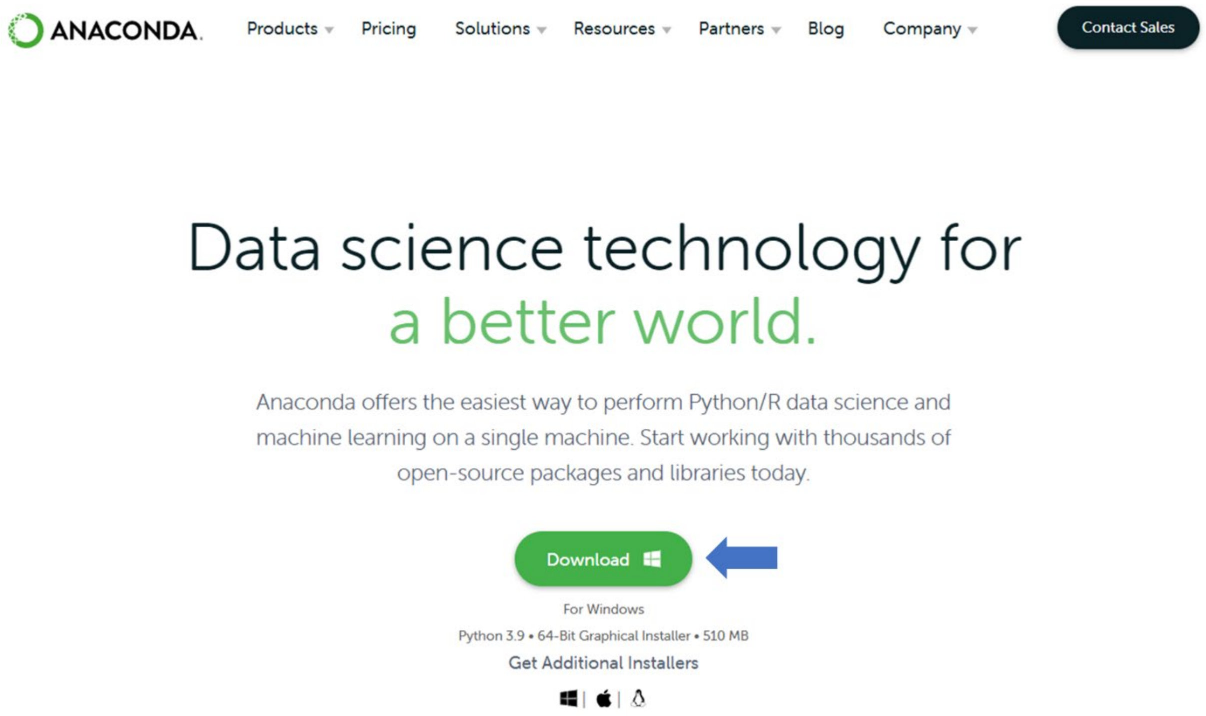
- C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.7\bin
- C:\Program Files\NVIDIA GPU Computing Toolkit\cudnn-windows-x86_64-8.4.1.50_cuda11.6-archive\bin

> *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:



Data science technology for a better world.

Anaconda offers the easiest way to perform Python/R data science and machine learning on a single machine. Start working with thousands of open-source packages and libraries today.

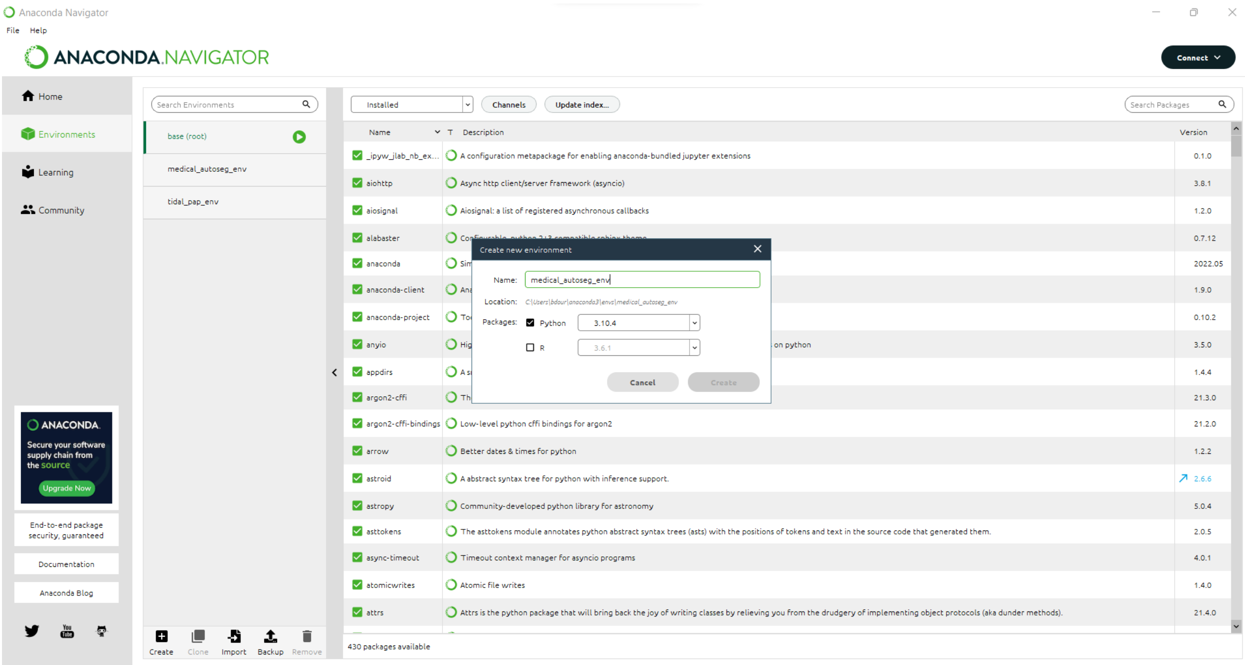
Once downloaded, launch the corresponding executable file, and follow all the default installation steps.

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.



Create new environment

Name: medical_autoseg_env

Location: C:\Users\bdour\anaconda3\envs\medical_autoseg_env

Packages: ☑ Python 3.10.4

□ R 3.6.1

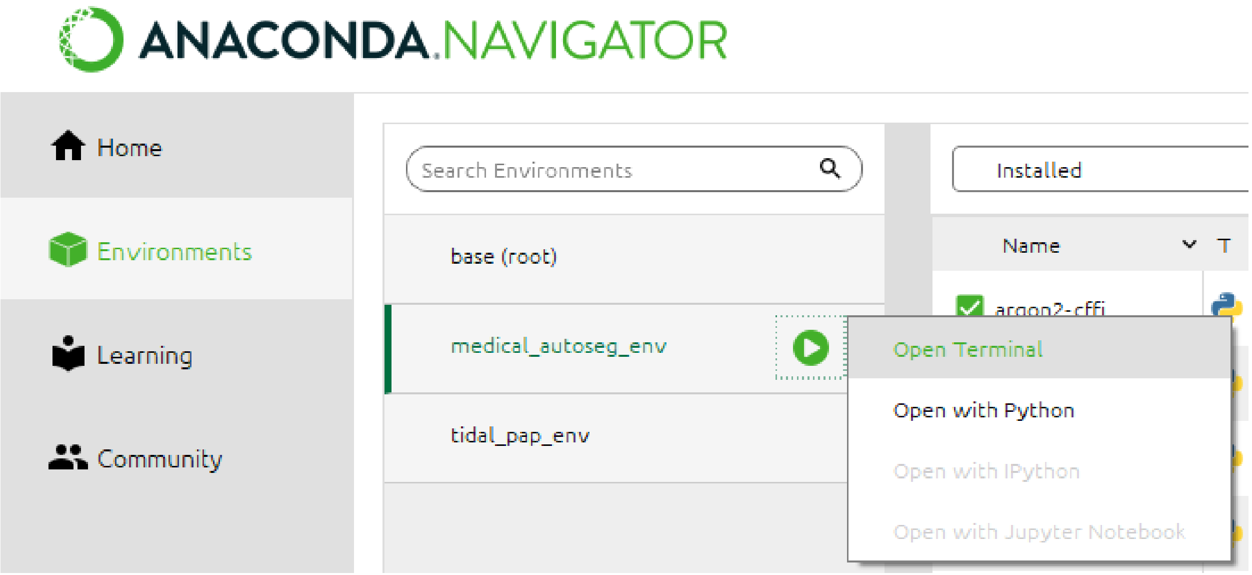
Cancel Create

Name	T	Description	Version
☑ _ipyw_jlab_nb_ex...	○	A configuration metapackage for enabling anaconda-bundled jupyter extensions	0.1.0
☑ aiohttp	○	Async http client/server framework (asyncio)	3.8.1
☑ aiosignal	○	Aiosignal: a list of registered asynchronous callbacks	1.2.0
☑ alabaster	○	Configurable, python 2+3 compatible sphinx theme	0.7.12
☑ anaconda	○	Sim	2022.05
☑ anaconda-client	○	Ana	1.9.0
☑ anaconda-project	○	To	0.10.2
☑ anyio	○	Hig ... on python	3.5.0
☑ appdirs	○	A s	1.4.4
☑ argon2-cffi	○	Th	21.3.0
☑ argon2-cffi-bindings	○	Low-level python cffi bindings for argon2	21.2.0
☑ arrow	○	Better dates & times for python	1.2.2
☑ astroid	○	A abstract syntax tree for python with inference support.	↗ 2.6.6
☑ astropy	○	Community-developed python library for astronomy	5.0.4
☑ asttokens	○	The asttokens module annotates python abstract syntax trees (asts) with the positions of tokens and text in the source code that generated them.	2.0.5
☑ async-timeout	○	Timeout context manager for asyncio programs	4.0.1
☑ atomicwrites	○	Atomic file writes	1.4.0
☑ attrs	○	Attrs is the python package that will bring back the joy of writing classes by relieving you from the drudgery of implementing object protocols (aka dunder methods).	21.4.0

430 packages available

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*').

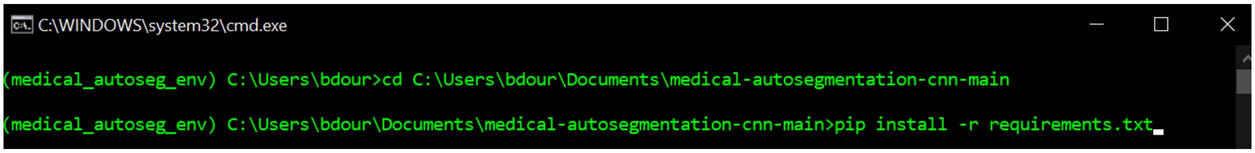
Once in the terminal, you will need to change your current directory to the location of your copy of the project repository. You can do this by entering the following command:

> *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*



```
(medical_autoseg_env) C:\Users\bdour>cd C:\Users\bdour\Documents\medical-autosegmentation-cnn-main

(medical_autoseg_env) C:\Users\bdour\Documents\medical-autosegmentation-cnn-main>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).

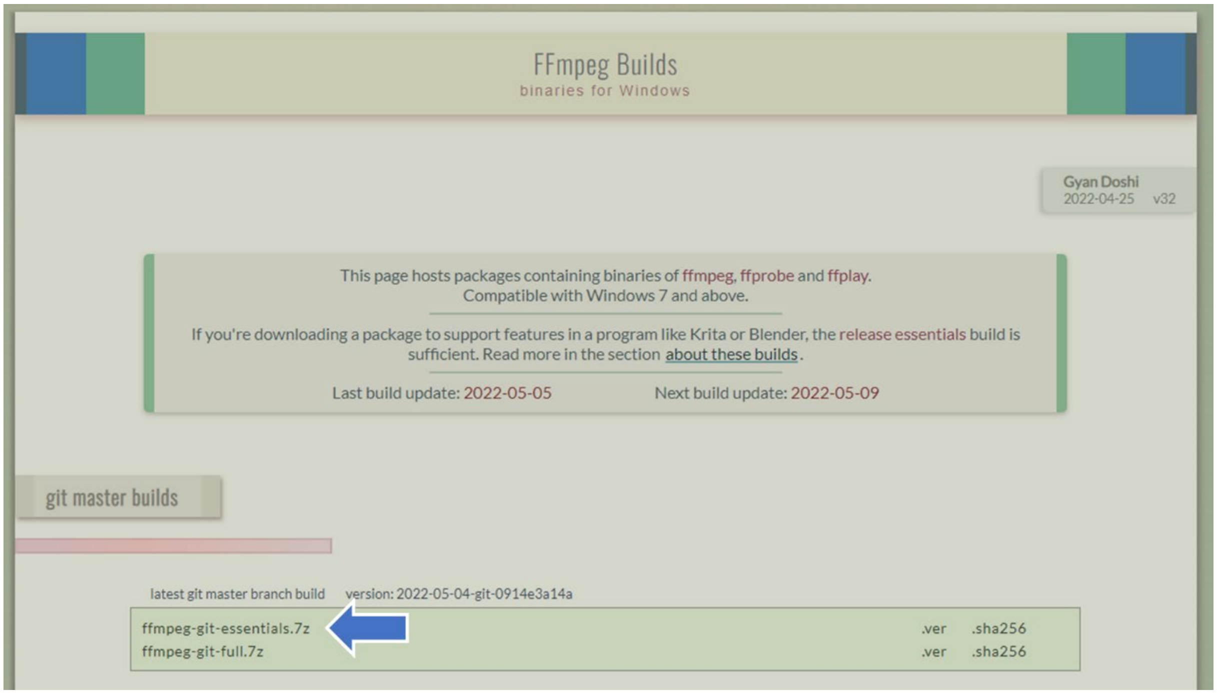
When done, close the terminal.

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:



FFmpeg Builds
binaries for Windows

Gyan Doshi
2022-04-25 v32

This page hosts packages containing binaries of ffmpeg, ffprobe and ffplay.
Compatible with Windows 7 and above.

If you're downloading a package to support features in a program like Krita or Blender, the release essentials build is sufficient. Read more in the section about these builds.

Last build update: 2022-05-05 Next build update: 2022-05-09

git master builds

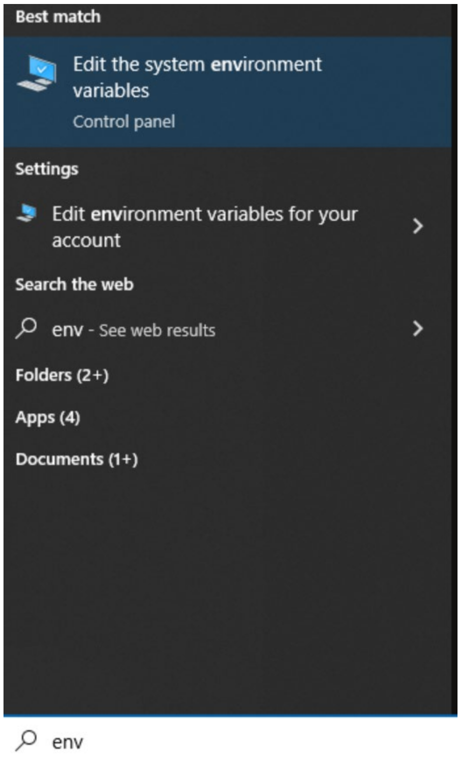
latest git master branch build version: 2022-05-04-git-0914e3a14a

ffmpeg-git-essentials.7z	.ver	.sha256
ffmpeg-git-full.7z	.ver	.sha256

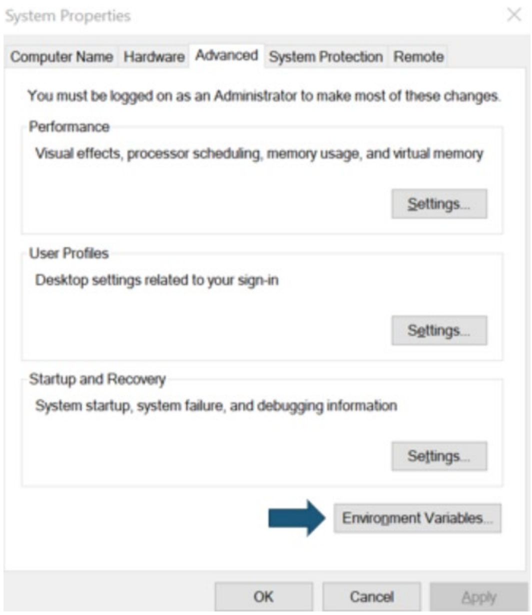
Once downloaded, extract the corresponding file to the location of your choice (e.g. '*C:\Users\bdour\Documents*'), then rename the extracted folder to '*ffmpeg*'.

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:



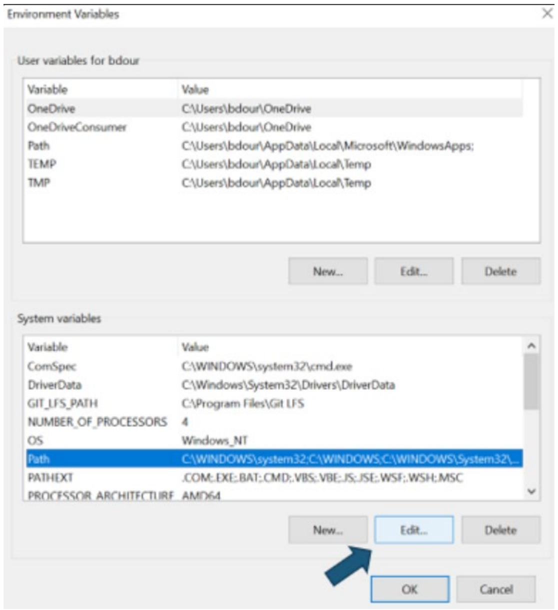
You must be logged on as an Administrator to make most of these changes.

Visual effects, processor scheduling, memory usage, and virtual memory

Desktop settings related to your sign-in

System startup, system failure, and debugging information

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



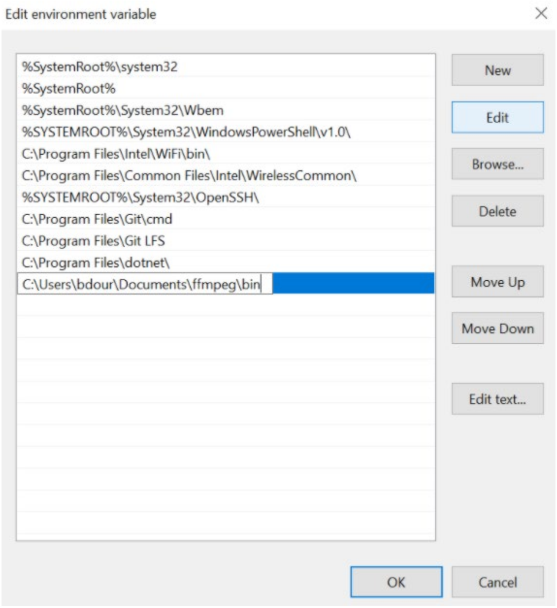
User variables for bdour

Variable	Value
OneDrive	C:\Users\bdour\OneDrive
OneDriveConsumer	C:\Users\bdour\OneDrive
Path	C:\Users\bdour\AppData\Local\Microsoft\WindowsApps;
TEMP	C:\Users\bdour\AppData\Local\Temp
TMP	C:\Users\bdour\AppData\Local\Temp

System variables

Variable	Value
ComSpec	C:\WINDOWS\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
GIT_LFS_PATH	C:\Program Files\Git LFS
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
Path	C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64

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:



```
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
C:\Program Files\Intel\WiFi\bin\
C:\Program Files\Common Files\Intel\WirelessCommon\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files\Git\cmd
C:\Program Files\Git LFS
C:\Program Files\dotnet\
C:\Users\bdour\Documents\ffmpeg\bin
```

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
```