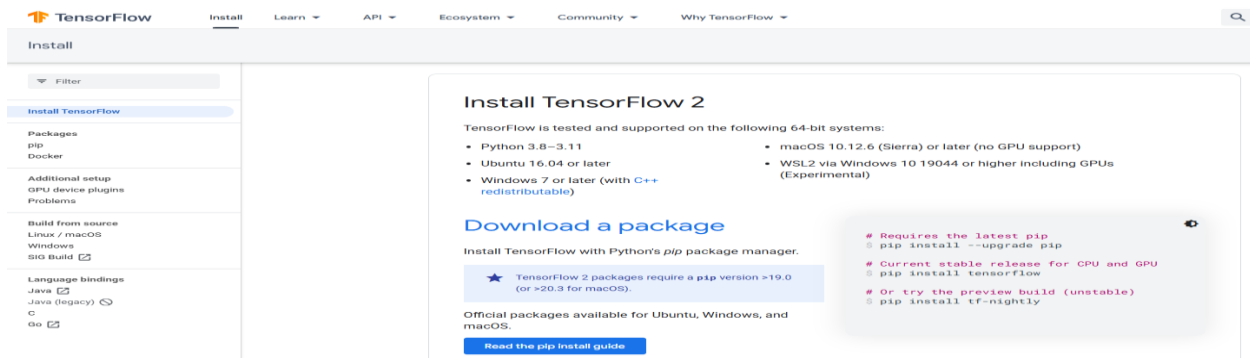


Installation instructions for implementations

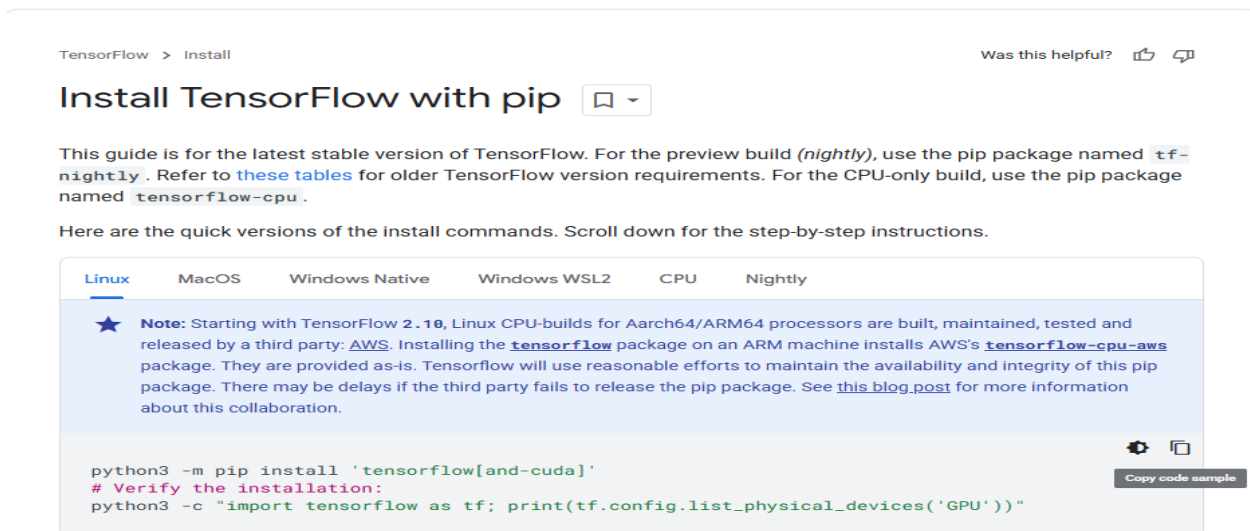
In this document, we will see the process of setting up TensorFlow with GPU support to significantly optimize model training times for scratch based CNN implementation.

First go to following link, which is “tensorflow.org” <https://www.tensorflow.org/>

Within the side-bar there will be a section which is called as “Install” click install after visiting the page:



After clicking install, there will be a section called as “pip” on the left side-bar, click pip.



Based on the host machine choose appropriate OS, since my machine is windows we'll go through with the windows installation steps. For other machines you can follow the similar steps based on the provided instructions. For windows machines choose the Windows Native section. TensorFlow 2.10 was the last TensorFlow release that supported GPU on native-Windows. In order to use more upgrade versions you have to follow the instructions of Windows WSL2. The

GPU supported package will be installed in order to train our scratch-based CNN model on GPU. So within the main sidebar of the PC you have to see the related Nvidia control panel, which is generally located on the lower right side of the screen. I have Quadro P1000 within my PC. When we scroll down in the pip page you'll see software requirements part which is start with **Microsoft Visual C++ Redistributable for Visual Studio 2015, 2017 and 2019**, if you don't have this version please go to this page and follow the instructions based on the supported architecture.

Software requirements

- Python 3.9–3.12
- pip version 19.0 or higher for Linux (requires `manylinux2014` support) and Windows. pip version 20.3 or higher for macOS.
- Windows Native Requires [Microsoft Visual C++ Redistributable for Visual Studio 2015, 2017 and 2019](#)

The following NVIDIA® software are only required for GPU support.

- [NVIDIA® GPU drivers](#)
 - `>= 525.60.13` for Linux
 - `>= 528.33` for WSL on Windows
- [CUDA® Toolkit 12.3](#).
- [cuDNN SDK 8.9.7](#).
- (Optional) [TensorRT](#) to improve latency and throughput for inference.

After that click on NVIDIA GPU drivers link, in order to download the GPU support for tensorflow.

Manual Driver Search

Search by product, product type or series

NVIDIA RTX PRO / RTX / Quadro

Quadro Series

Quadro P1000

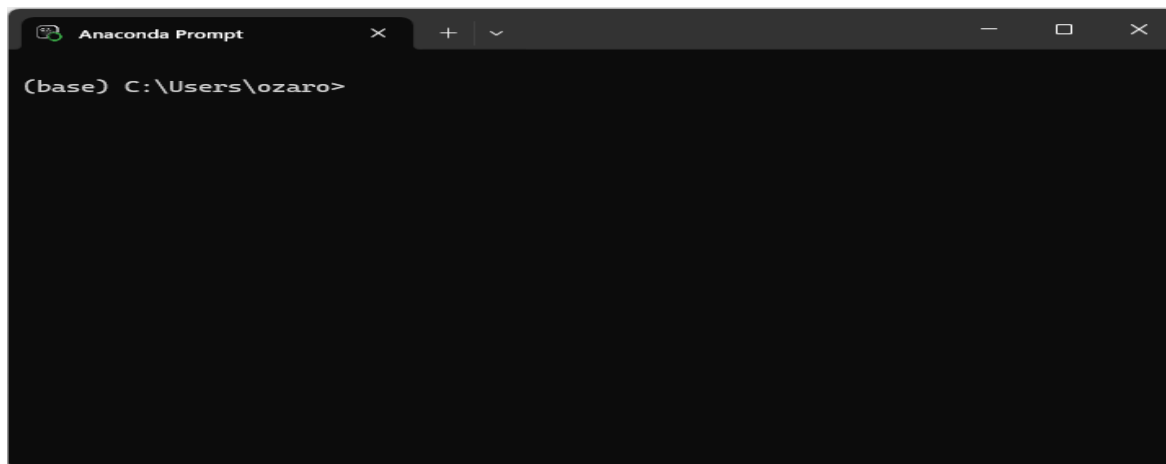
Windows 10 64-bit

English (US)

Find

Choose an appropriate driver based on the Nvidia device that your host machine has. Since my PC has Quadro P1000, I choose the related sections for this device. After clicking “Find” and “View” respectively, you will reach the download page after that click download. And run the .exe file which will be on “download section” within the PC. This is the driver that’ll be used in order to run the model on GPU.

After these installations please go to “**Anaconda**” following link : <https://www.anaconda.com/> Click “Free Download” and type you e-mail to receive the installation links. Once you’ve download and install your Anaconda within the main windows panel you’ll see a folder as Anaconda(anaconda3) After open the anaconda Prompt:



Type nvidia-smi to verify the driver is successfully downloaded.

```
(base) C:\Users\ozaro>nvidia-smi
Sat May 31 13:15:10 2025
```

NVIDIA-SMI 573.06			Driver Version: 573.06			CUDA Version: 12.8		
GPU	Name	Perf	Driver-Model	Bus-Id	Disp.A	Volatile	Uncorr.	ECC
Fan	Temp		Pwr:Usage/Cap		Memory-Usage	GPU-Util	Compute	M. MIG M.
0	Quadro P1000	P0	N/A / N/A	00000000:01:00.0	On	2%	Default	N/A
0%	67C			1307MiB / 4096MiB				N/A

Processes:							
GPU	GI ID	CI ID	PID	Type	Process name	GPU Memory Usage	
0	N/A	N/A	300	C+G	...lus\logioptionsplus_agent.exe	N/A	
0	N/A	N/A	464	C+G	...4__8wekyb3d8bbwe\ms-teams.exe	N/A	
0	N/A	N/A	5012	C+G	...indows\System32\ShellHost.exe	N/A	
0	N/A	N/A	8788	C+G	...0.3240.92\msedgewebview2.exe	N/A	
0	N/A	N/A	9724	C+G	...em32\ApplicationFrameHost.exe	N/A	
0	N/A	N/A	10068	C+G	...0_x64__8j3eq9eme6ctt\IGCC.exe	N/A	
0	N/A	N/A	10728	C+G	...4__8wekyb3d8bbwe\ms-teams.exe	N/A	
0	N/A	N/A	13092	C+G	...0.3240.92\msedgewebview2.exe	N/A	
0	N/A	N/A	13524	C+G	...t\Edge\Application\msedge.exe	N/A	
0	N/A	N/A	15544	C+G	...Chrome\Application\chrome.exe	N/A	
0	N/A	N/A	19144	C+G	...y\StartMenuExperienceHost.exe	N/A	
0	N/A	N/A	19348	C+G	...SnippingTool\SnippingTool.exe	N/A	
0	N/A	N/A	19876	C+G	...cemodV3\app\last\Voicemod.exe	N/A	
0	N/A	N/A	21440	C+G	..._cw5n1h2txyewy\SearchHost.exe	N/A	
0	N/A	N/A	23716	C+G	...Chrome\Application\chrome.exe	N/A	
0	N/A	N/A	23736	C+G	...xyewy\ShellExperienceHost.exe	N/A	
0	N/A	N/A	26008	C+G	...yb3d8bbwe\WindowsTerminal.exe	N/A	
0	N/A	N/A	29720	C+G	...5n1h2txyewy\TextInputHost.exe	N/A	
0	N/A	N/A	30184	C+G	C:\Windows\explorer.exe	N/A	
0	N/A	N/A	30840	C+G	...ntrolPanel\SystemSettings.exe	N/A	
0	N/A	N/A	31540	C+G	...crosoft\OneDrive\OneDrive.exe	N/A	
0	N/A	N/A	32092	C+G	...ms\Microsoft VS Code\Code.exe	N/A	
0	N/A	N/A	32796	C+G	...8bbwe\PhoneExperienceHost.exe	N/A	
0	N/A	N/A	33364	C+G	...ice\root\Office16\WINWORD.EXE	N/A	


```
WARNING: infoROM is corrupted at gpu 0000:01:00.0
(base) C:\Users\ozaro>
```

Now after that we should create environment for implementation. **Type: conda create -n “environment_name” python==3.10** You can type any environment name, and any python version that you are going to use within the project. But keep in mind that the versions of libraries should be consistent with the used python versions. The given python version is the version which will be used within the newly created environment. Once you’ve successfully created the environment you’ve to type **conda activate “environment_name”** also in order to deactivate an environment type **conda deactivate**. Once you’ve deactivate the environment you’ll go back to the base environment which is actually the default environment. You can also type **conda env list** to list all the conda environments.

```

(base) C:\Users\ozaro>conda activate tensorflow_env

(tensorflow_env) C:\Users\ozaro>conda env list

# conda environments:
#
base                  C:\Users\ozaro\anaconda3
tensorflow_env        * C:\Users\ozaro\anaconda3\envs\tensorflow_env
tf_gpu                C:\Users\ozaro\anaconda3\envs\tf_gpu
vggish-cnn            C:\Users\ozaro\anaconda3\envs\vggish-cnn
vggish_env            C:\Users\ozaro\anaconda3\envs\vggish_env
vggish_tf_env         C:\Users\ozaro\anaconda3\envs\vggish_tf_env
tensorflow_env        c:\Users\ozaro\anaconda3\envs\tensorflow_env
vggish-cnn            c:\Users\ozaro\anaconda3\envs\vggish-cnn
vggish_tf_env         c:\Users\ozaro\anaconda3\envs\vggish_tf_env

(tensorflow_env) C:\Users\ozaro>

```

The star “*” shows us the currently working environment, which is activated before.

Once you’ve installed the GPU driver, we have to setup the GPU within the environment

```
type: conda install -c conda-forge cudatoolkit=11.2 cudnn=8.1.0
```

After downloaded the cuda toolkit for TensforFlow type : pip install --upgrade pip

Now we’ve successfully created our environment. Now we’ll download the required libraries which will be used within the implementation. For the sake of simplicity we’ll use requirement.txt file to download the necessary libraries. Here is the file we’ll be used.

```

≡ requirement.txt
1  tensorflow==2.10.0
2  scikit-learn==1.3.0
3  numpy==1.24.3
4  matplotlib==3.7.2
5  seaborn==0.13.0
6  pandas==2.1.0
7  streamlit
8  librosa==0.10.1

```

Once you’ve located this file within your project folder, you have to change your directory where this file is located in order to make installations.

```

(tensorflow_env) C:\Users\ozaro>cd C:\Users\ozaro\Desktop\CENG562\term_project

(tensorflow_env) C:\Users\ozaro\Desktop\CENG562\term_project>

```

After type “**dir**” you’ll see the files which is located in the given folder.

```
(tensorflow_env) C:\Users\ozaro\Desktop\CENG562\term_project>dir
Volume in drive C has no label.
Volume Serial Number is 5E5E-14FD

Directory of C:\Users\ozaro\Desktop\CENG562\term_project

05/27/2025  10:42 PM    <DIR>          .
05/31/2025  12:02 AM    <DIR>          ..
05/03/2025  08:01 PM          1,323,632 blues.000000.wav
05/03/2025  08:02 PM    <DIR>          dataset
05/09/2025  09:12 PM    <DIR>          genres_original
05/21/2025  07:01 PM          9,874 mel_features.py
05/10/2025  01:28 AM      5,980,553 music_genre_classification_notebook.html
05/03/2025  11:47 PM           133 requirement.txt
05/03/2025  08:01 PM          1,323,632 rock.000000.wav
05/10/2025  12:32 AM      86,294,096 Trained_model.h5
05/10/2025  12:33 AM      86,294,096 Trained_model.keras
05/21/2025  10:13 PM     152,647,272 Trained_model_vggish.h5
05/21/2025  10:13 PM     152,632,506 Trained_model_vggish.keras
05/10/2025  12:45 AM           2,479 training_hist.json
05/21/2025  10:13 PM           2,451 training_hist_vggish.json
05/31/2025  12:33 PM      5,658,478 training_model.ipynb
05/27/2025  10:42 PM      2,648,842 training_model_VGGish.html
05/21/2025  10:14 PM      2,331,530 training_model_vggish.ipynb
05/21/2025  09:14 PM           3,535 vggish_input.py
05/21/2025  07:01 PM           2,028 vggish_params.py
05/21/2025  09:15 PM    <DIR>          __pycache__
                16 File(s)      497,155,137 bytes
                5 Dir(s)   71,178,571,776 bytes free

(tensorflow_env) C:\Users\ozaro\Desktop\CENG562\term_project>
```

Once we’ve located our file we should install the “**requirement.txt**” type

```
pip install -r requirement.txt
```

This file contains all the required libraries used in the implementation. After the installation now the environment is ready for use. I used jupyter notebook for implementations, so in order to use the environment within the jupyter notebook you’ve to choose the right environment to work with. Otherwise you’ll may not be able to import necessary libraries which will be used during implementation.

These installations are all about the scratch-based CNN implementation. For creating an environment for transfer learning base approach we can use the same steps in order to create an environment for VGGish based notebook.

We'll not use GPU in order to train the VGGish based model, so we don't have to download the CUDA toolkit for VGGish based implementation. All the implementations are done via jupyter notebook using Visual Studio Code.