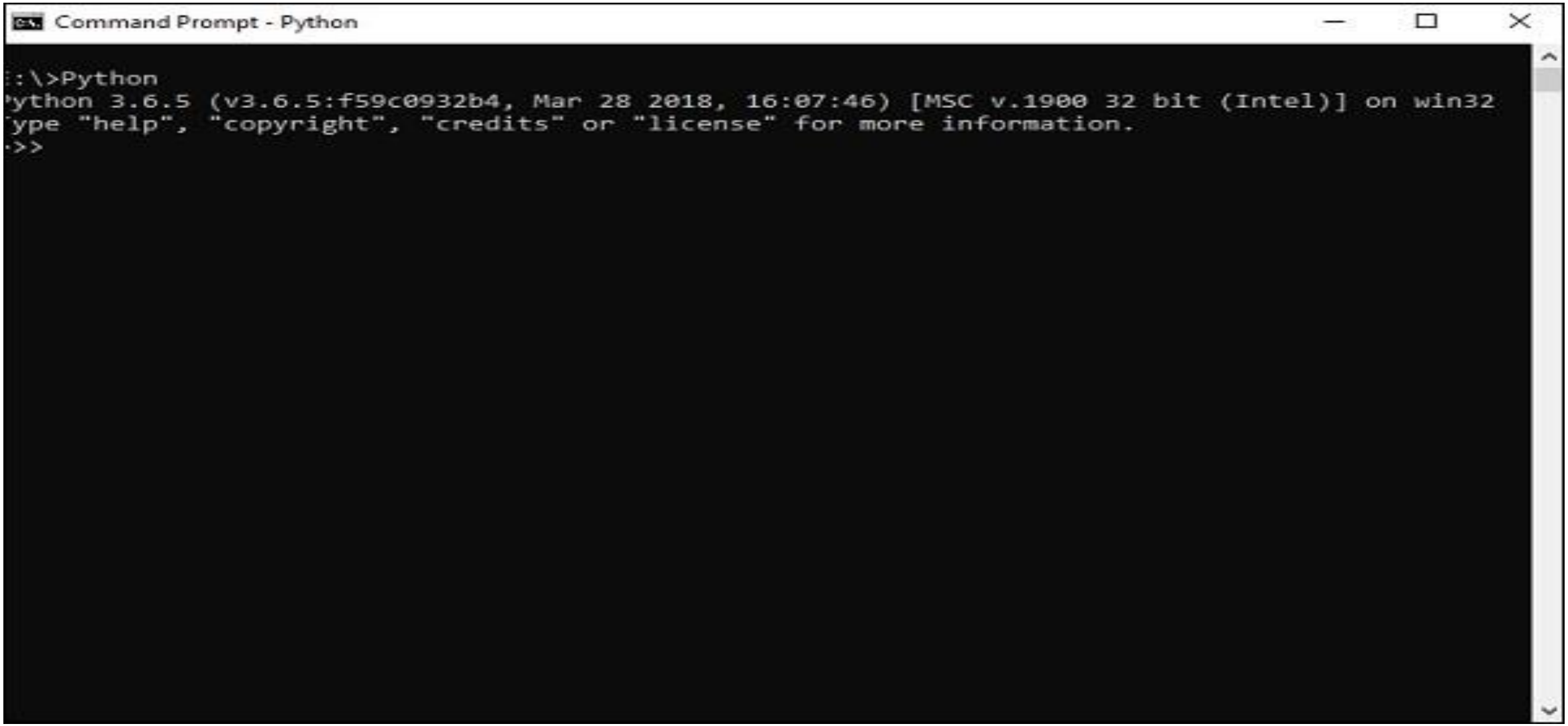
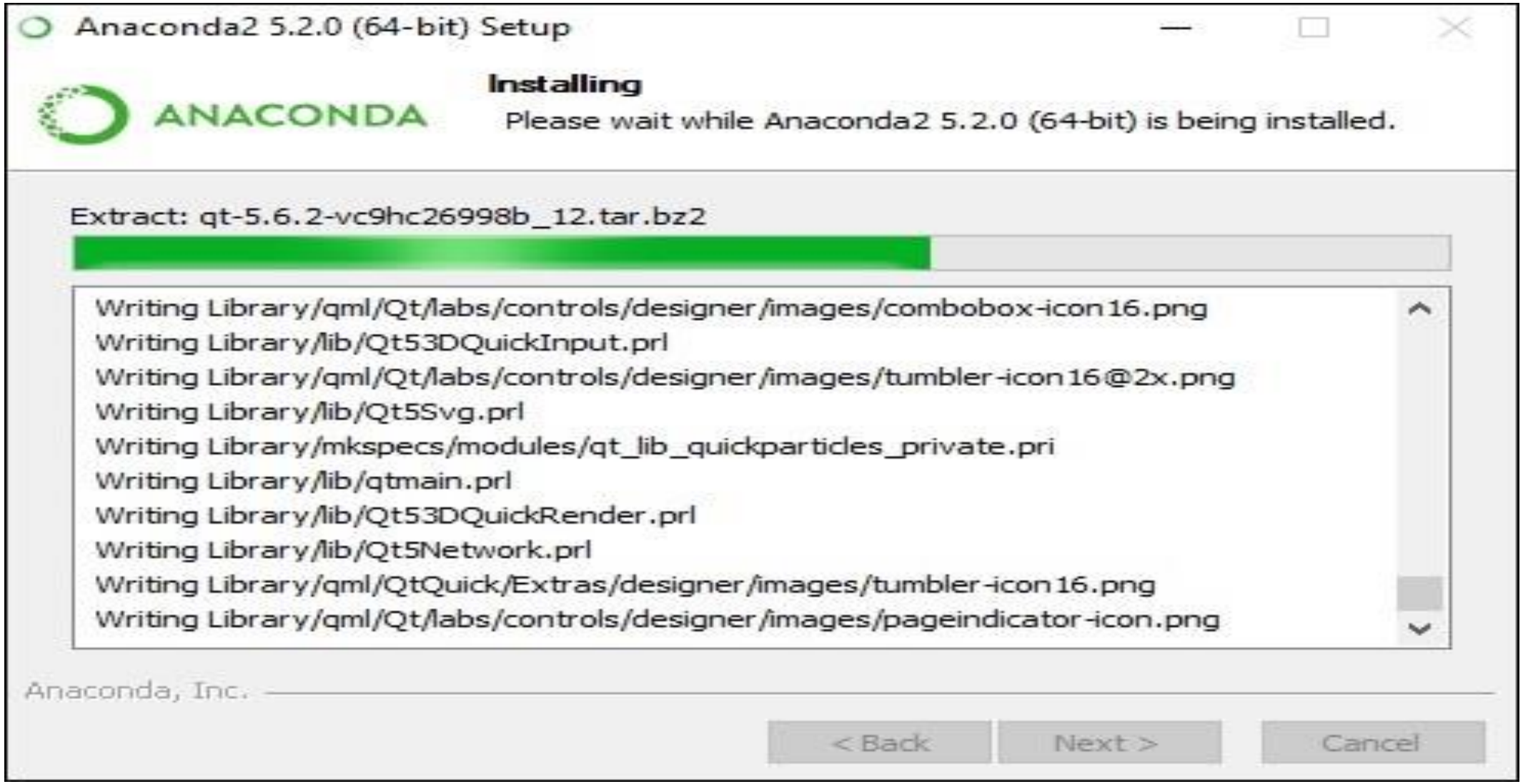
INSTALLING PYTHON PACKAGES

1.INSTALL TENSORFLOW:

- **Step 1 – Verify the python version being installed.**

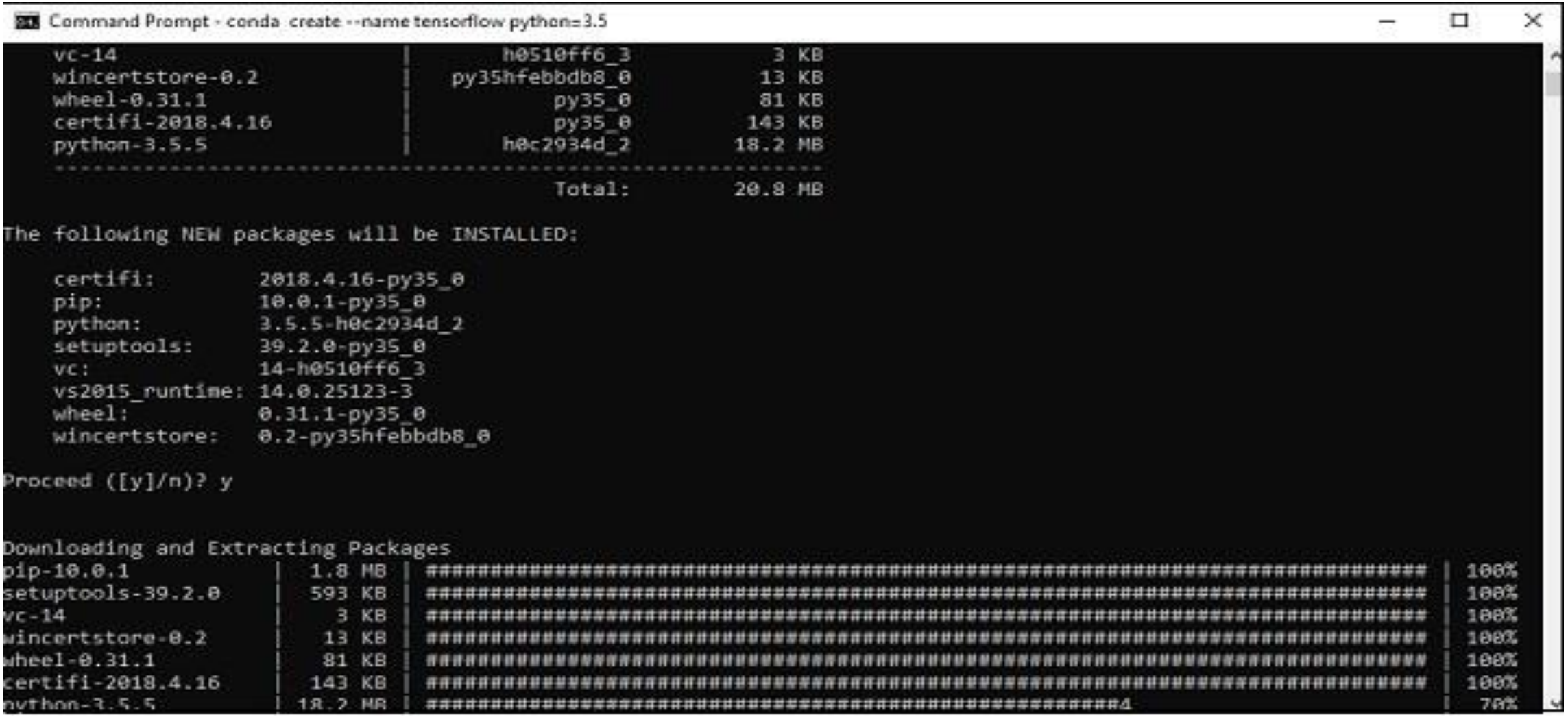


Step 2: Install Anaconda framework in our system.

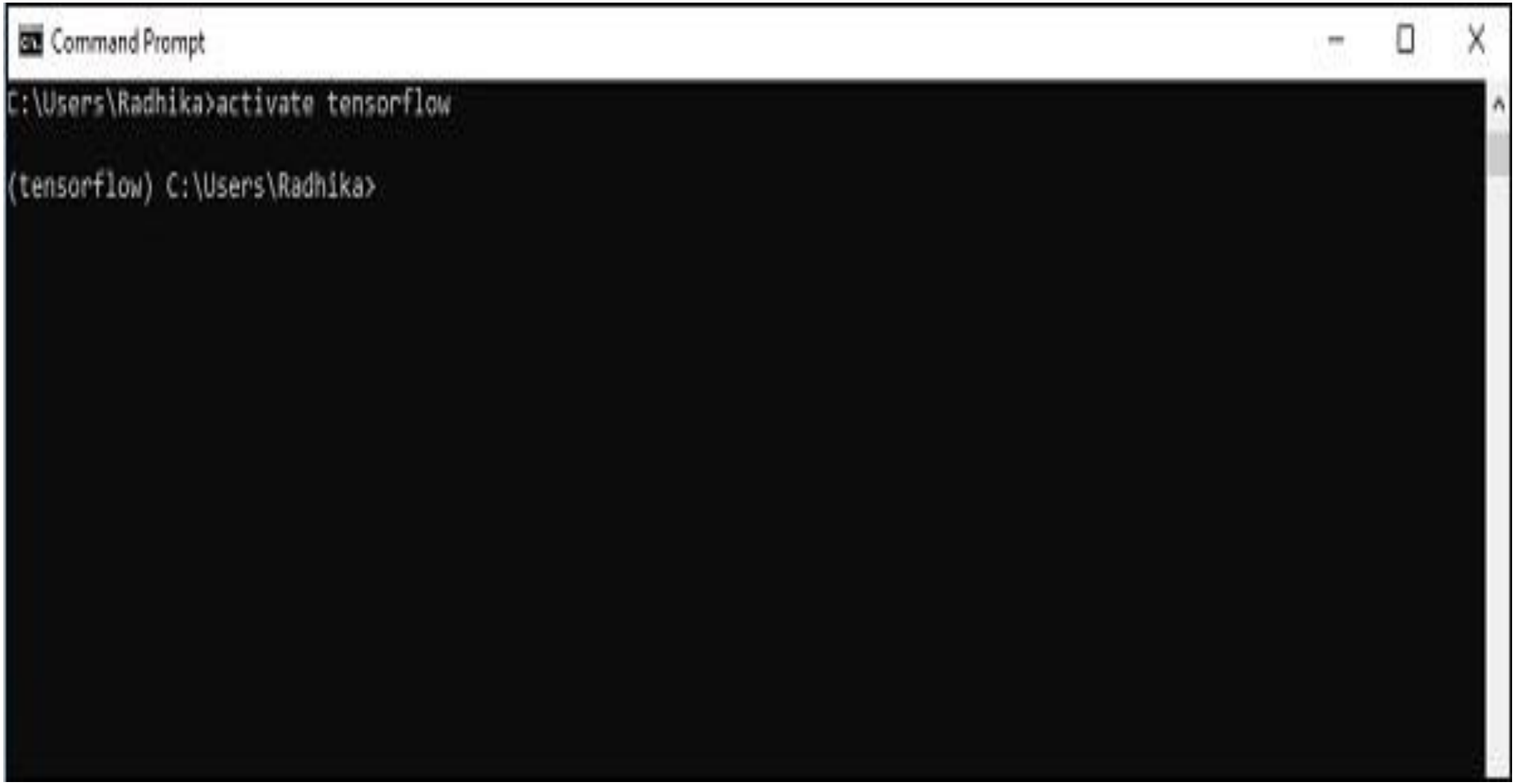


Step 3 – Execute the following command to initialize the installation of TensorFlow

conda create --name tensorflow python = 3. 5

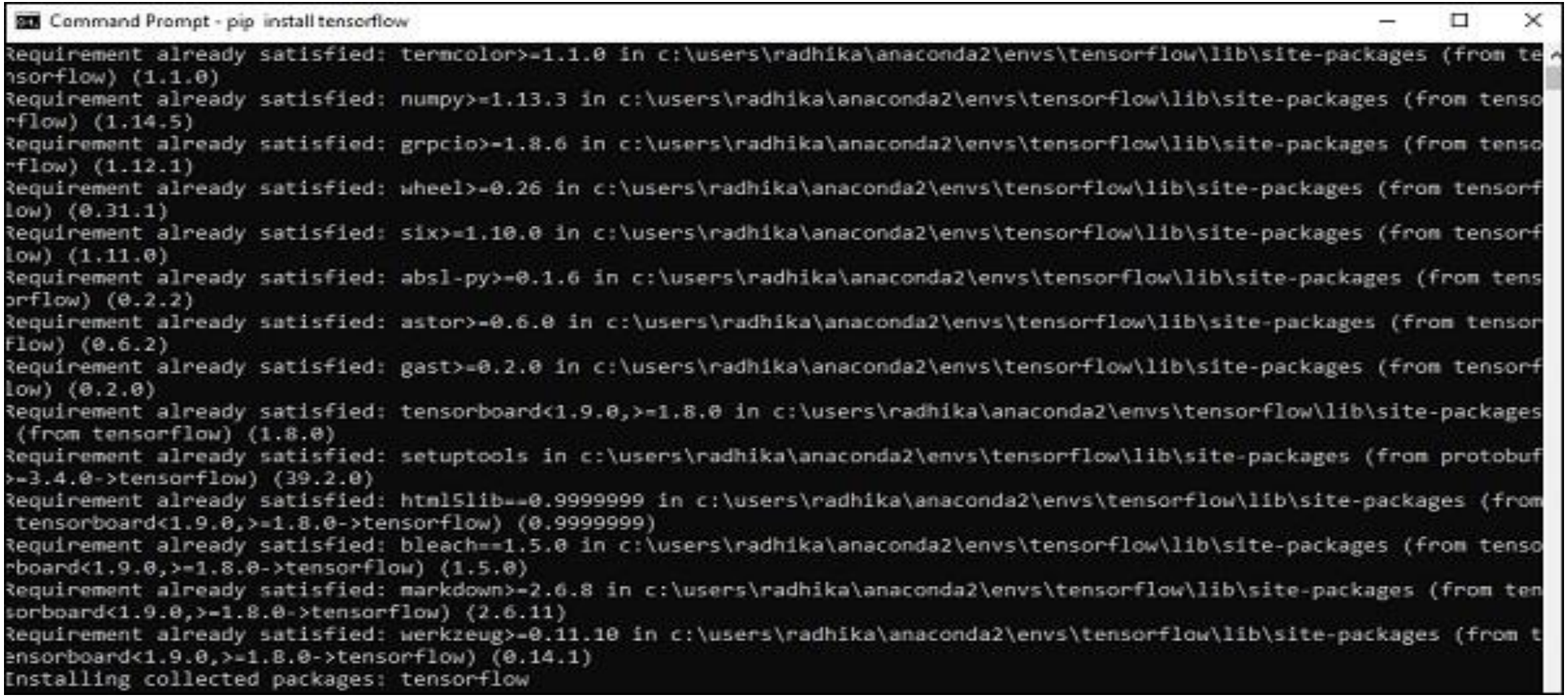


Step 4: ACTIVATE TENSORFLOW



Step 5 – Use pip to install “Tensorflow” in the system.

- **pip install tensorflow**
- **pip install tensorflow-gpu**



“HELLO WORLD” IN TENSORFLOW :

```
C:\Users\Radhika>activate tensorflow

(tensorflow) C:\Users\Radhika>python
Python 3.5.5 |Anaconda, Inc.| (default, Apr  7 2018, 04:52:34) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorflow as tf
>>> hello = tf.constant('Hello, Tensorflow!')
>>> sess = tf.session()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: module 'tensorflow' has no attribute 'session'
>>> sess = tf.Session()
2018-06-28 11:12:04.586763: I T:\src\github\tensorflow\tensorflow\core\platform\cpu_feature_guard.cc:140] Your CPU suppo
rts instructions that this Tensorflow binary was not compiled to use: AVX2
>>> print(sess.run(hello))
b'Hello, Tensorflow!'
>>>
```

2.Keras Installation Steps:

- **Step 1: Create virtual environment**

- **Virtualenv is used to manage Python packages for different projects.**

For Windows using this command

```
py -m venv keras
```

Step 2: Activate the environment

FOR WINDOWS \env\Scripts\activate

Step 3: Python libraries

NUMPY :

pip install numpy

Collecting numpy Downloading
https://files.pythonhosted.org/packages/cf/a4/d5387a742045
42a60ad1baa84cd2d3353c330e59be8cf2d47c0b11d3cde8/
numpy-3.1.1-cp36-cp36m-macosx_10_6_intel.
macosx_10_9_intel.macosx_10_9_x86_64.
macosx_10_10_intel.macosx_10_10_x86_64.whl (14.4MB)
|  |

PANDAS: pip install pandas

- **could see the following response,**
- **Collecting pandas Downloading https://files.pythonhosted.org/packages/cf/a4/d5387a74204542a60ad1baa84cd2d3353c330e59be8cf2d47c0b11d3cde8/ pandas-3.1.1-cp36-cp36m-macosx_10_6_intel.macosx_10_9_intel.macosx_10_9_x86_64.macosx_10_10_intel.macosx_10_10_x86_64.whl (14.4MB)**
  |[redacted]| 14.4MB 2.8MB/s

MATPLOTLIB :pip install matplotlib

- Collecting matplotlib Downloading https://files.pythonhosted.org/packages/cf/a4/d5387a74204542a60ad1baa84cd2d3353c330e59be8cf2d47c0b11d3cde8/ matplotlib-3.1.1-cp36-cp36m-macosx_10_6_intel.macosx_10_9_intel.macosx_10_9_x86_64.macosx_10_10_intel.macosx_10_10_x86_64.whl (14.4MB)
  
  | 14.4MB 2.8MB/s

Keras Installation Using Python

- Pip install keras

Quit virtual environment

DEACTIVATE

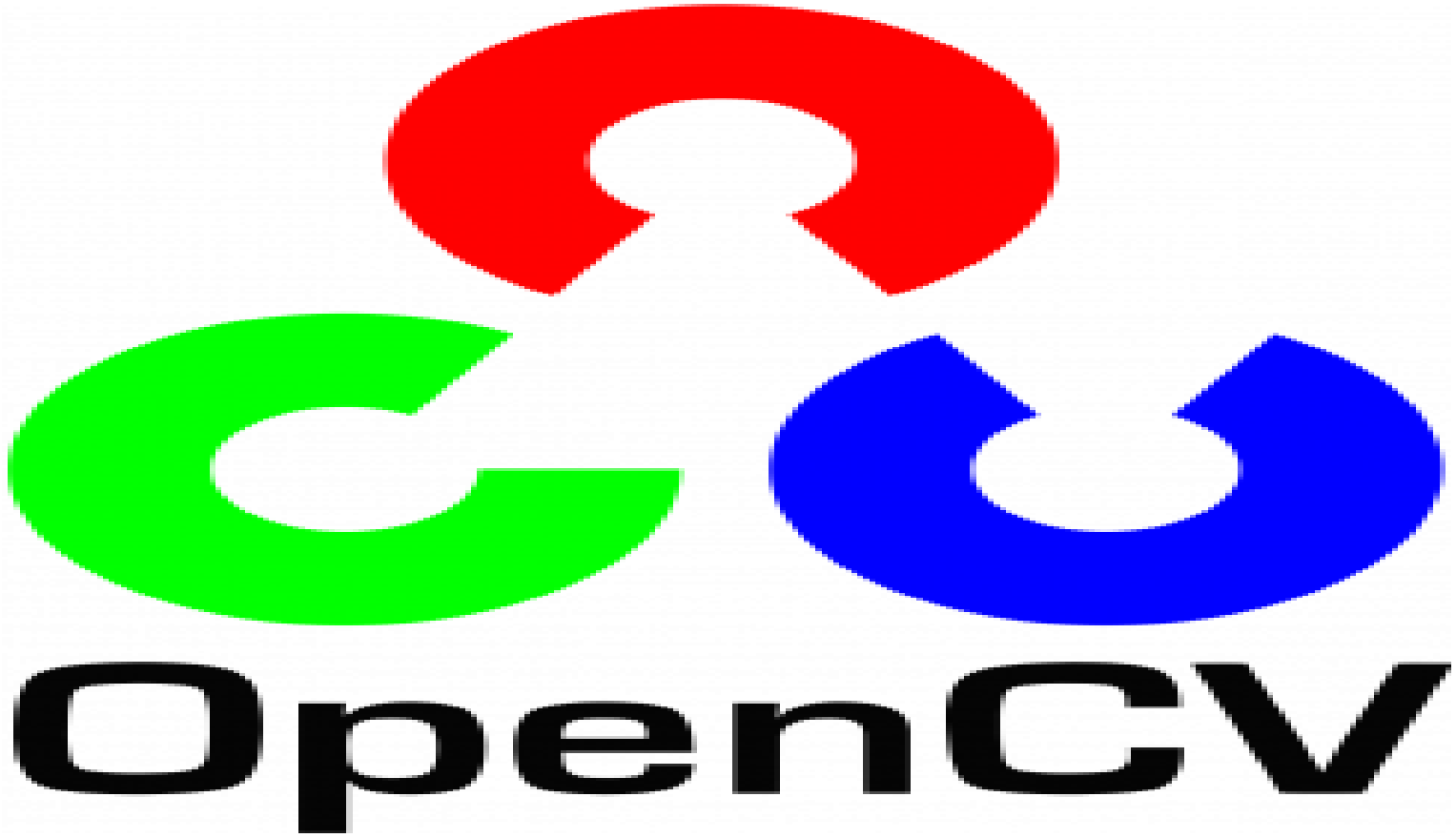
INSTALL CONDA ENVIRONMENT

conda create --name PythonCPU

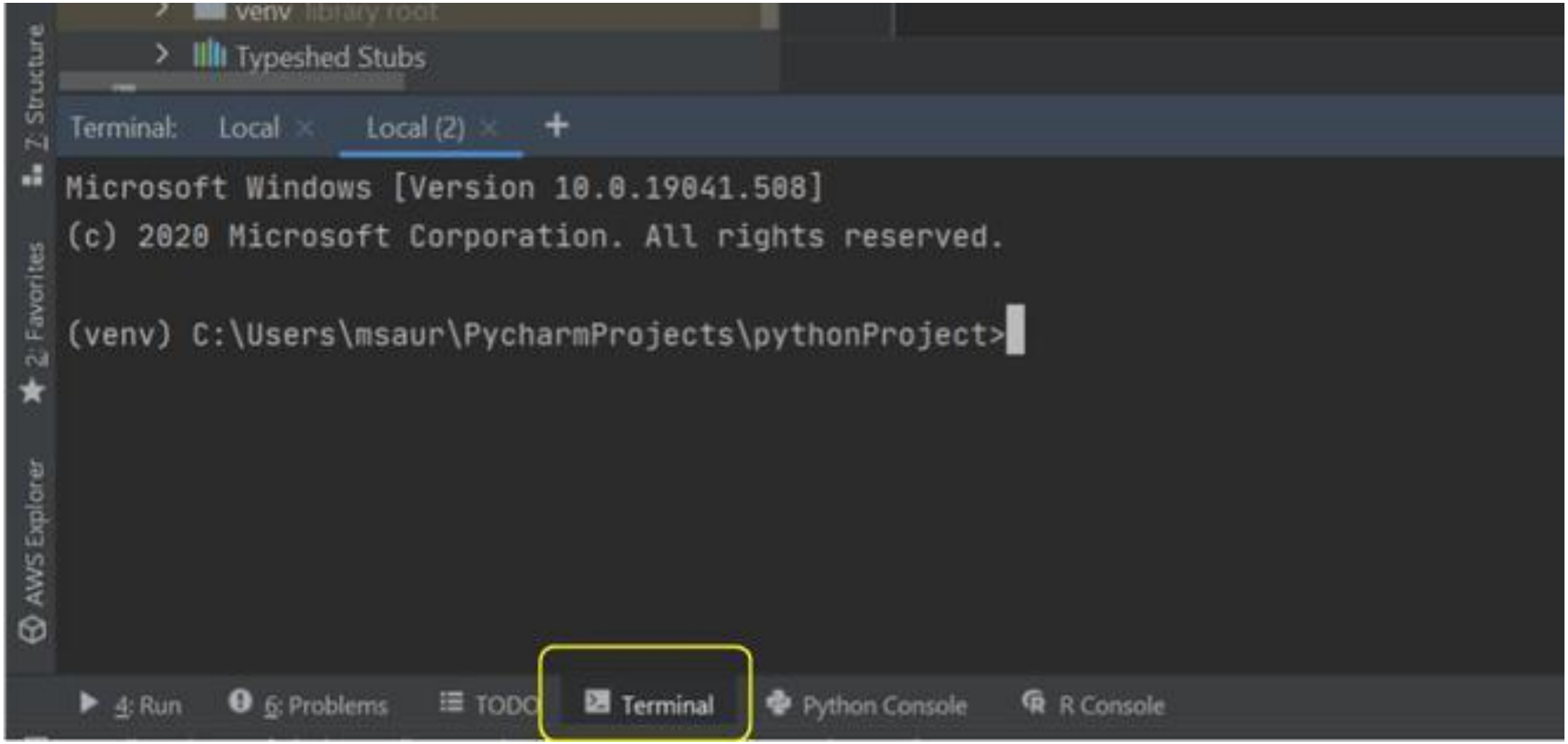
INSTALL SPYDER :conda install spyder

conda install -c anaconda keras

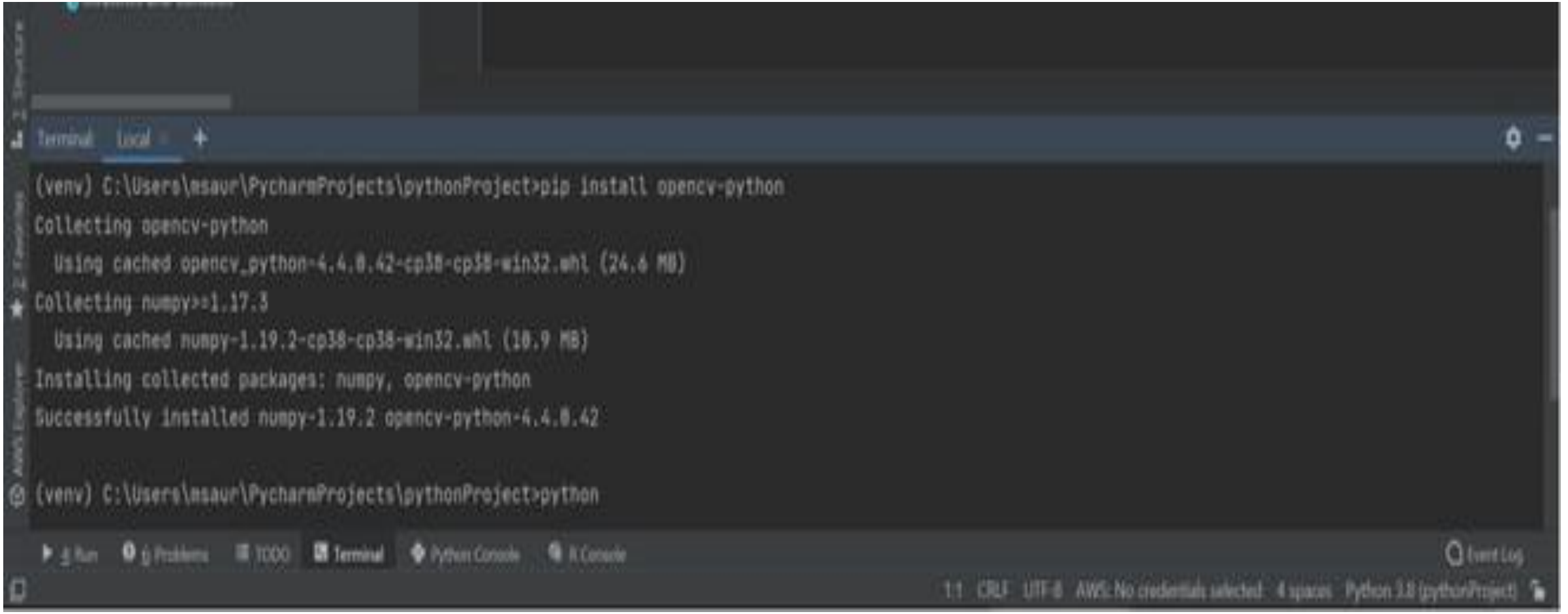
3.INSTALL OPEN CV



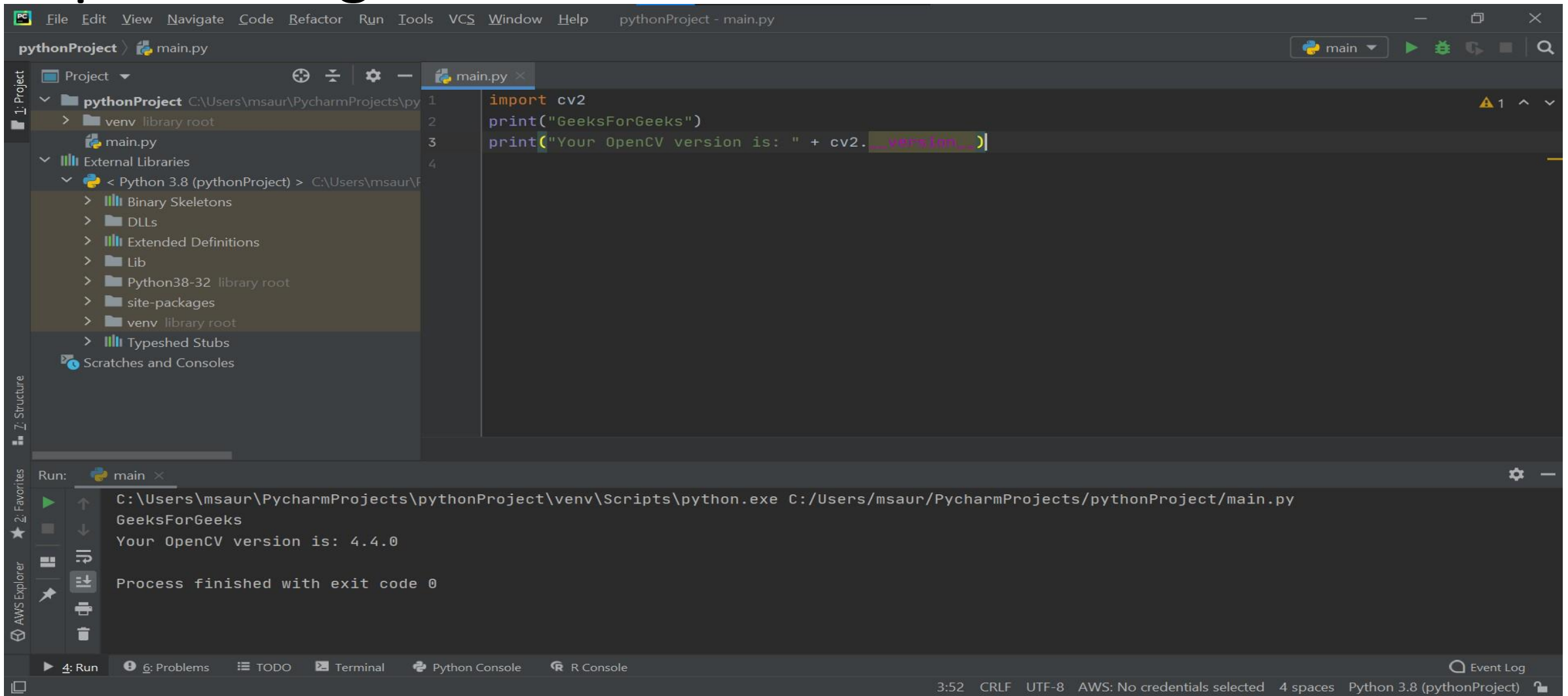
1) Go to the terminal option at the bottom of the IDE window



2) The pip (package manager) can also be used to download and install OpenCV. To install OpenCV, just type the following command:



3) Now simply import OpenCV in your python program in which you want to use image processing functions.



3. pip install pillow

- Merging Two or More Images
- IMAGES USED

1

2

3

```
From PIL import Image

img_01 = Image.open(“digit-number-img-0.jpg”)
img_02 = Image.open(“digit-number-img-1.jpg”)
img_03 = Image.open(“digit-number-img-2.jpg”)
img_04 = Image.open(“digit-number-img-3.jpg”)

img_01_size = img_01.size
img_02_size = img_02.size
img_03_size = img_02.size
img_02_size = img_02.size
```

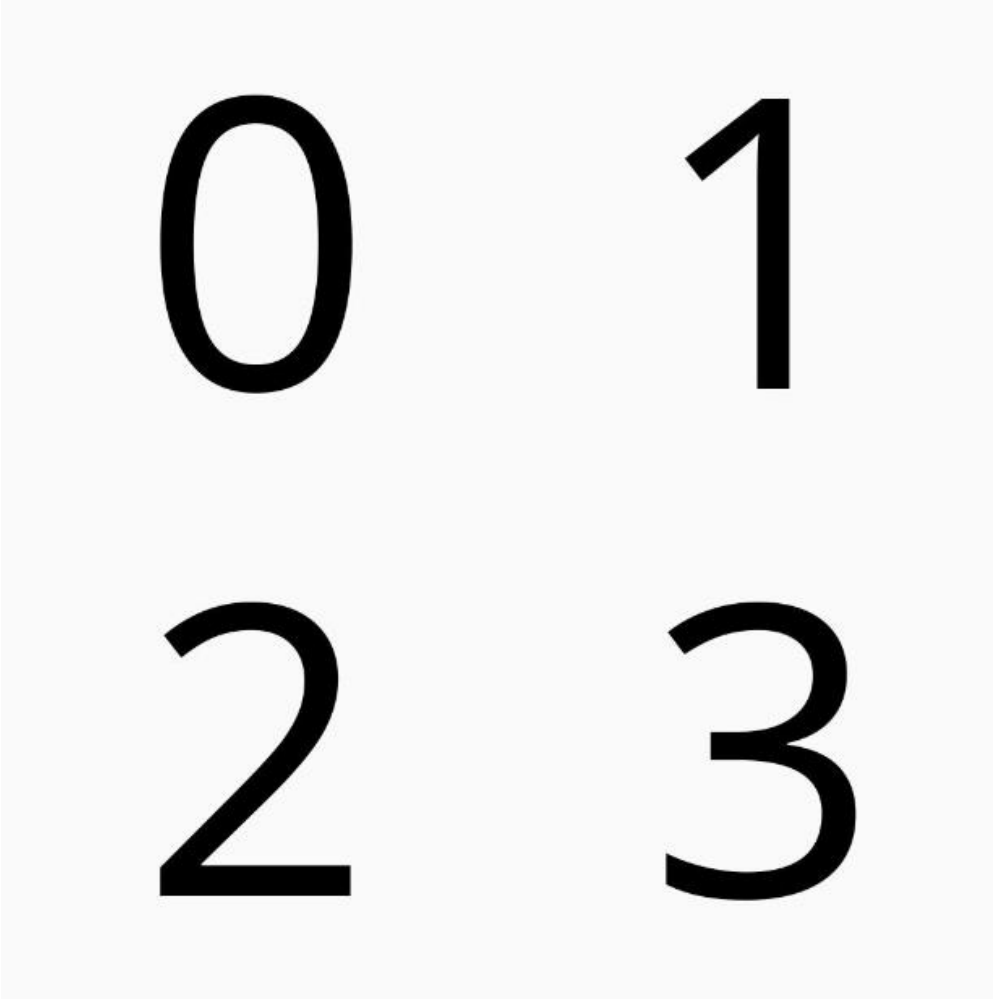
```
Print('img 1 size: ', img_01_size)
print('img 2 size: ', img_02_size)
print('img 3 size: ', img_03_size)
print('img 4 size: ', img_03_size)

new_im = Image.new('RGB', (2*img_01_size[0],2*img_01_size[1]),
(250,250,250))

new_im.paste(img_01, (0,0))
new_im.paste(img_02, (img_01_size[0],0))
new_im.paste(img_03, (0,img_01_size[1]))
new_im.paste(img_04, (img_01_size[0],img_01_size[1]))

new_im.save("merged_images.png", "PNG")
new_im.show()
```

OUTPUT



CREATE A THUMBNAIL

- IMAGE USED

```
From PIL import Image

# creating a object
image = Image.open(r”image.jpg”)
MAX_SIZE = (100, 100)

# Creating the thumbnail
image.thumbnail(MAX_SIZE)

image.show()
```

OUTPUT