**5. IMPLEMENTATION**

For any System to implement we need to have some basic requirements that fulfil our needs. The basic requirements needed to accomplish the project requirements are given below:

**Software requirement**:

Language : python

Operating System : Above Windows 7

Data base : MNIST data base

Software : Syder,Anakonda

**Hardware requirement**:

Hard ware required for the system are:

* Processor : Intel core I3 above
* Memory : Above 4 GB RAM
* Speed : 2.4GHz

**Functional Requirements**:

* The system should process the input given by the user only if it is an image file.
* System should detect the digits present in the image.
* System should retrieve digits present in the image and display it to the user.

**Non-Functional requirements**:

* Performance : System should recognized the hand written Character inwith a accuracy of 90% and above
* Flexibility : It provides user to load Image easily.

**Anaconda Navigator:**

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda® distribution that allows you to launch applications and easily manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository. It is available for Windows, macOS and Linux.

In order to run, many scientific packages depend on specific versions of other packages. Data scientists often use multiple versions of many packages, and use multiple environments to separate these different versions.

The command line program conda is both a package manager and an environment manager, to help data scientists ensure that each version of each package has all the dependencies it requires and works correctly.

Navigator is an easy, point-and-click way to work with packages and environments without needing to type conda commands in a terminal window. You can use it to find the packages you want, install them in an environment, run the packages and update them, all inside Navigator.

The following applications are available by default in Navigator:

* [JupyterLab](https://jupyterlab.readthedocs.io/en/stable/)
* [Jupyter Notebook](https://jupyter.readthedocs.io/en/latest/)
* [QTConsole](https://qtconsole.readthedocs.io/en/latest/)
* [Spyder](https://www.spyder-ide.org/)
* [VSCode](https://code.visualstudio.com/docs)
* [Glueviz](http://glueviz.org/en/stable/)
* [Orange 3 App](http://orange.biolab.si/docs/)
* [Rodeo](http://rodeo.yhat.com/docs/#requirements)
* [RStudio](http://docs.rstudio.com/)
* Advanced conda users can also [build your own Navigator applications](https://conda.io/docs/build_tutorials/app.html).

System requirement for installing anaconda are listed below:

* License: it can used for free of cost.
* Operating System:Windows vista or newer,64-bit macOS 10.10+ or Linux and others.
* System architecture:64-bit x 86, 32-bit x 86 with windows or Linux , or Power8.
* Minimum 3 GB disk space to download and install.

Anaconda navigator can install in windows in the following way.

1. Open the anaconda navigator in ang web browser.
2. Download the Anakonda installer..
3. Click Next.
4. Read the licensing terms and click “I Agree”.
5. Select an install for “Just Me” unless you’re installing for all users (which requires Windows Administrator privileges) and click Next.
6. Select a destination folder to install Anaconda and click the Next button.
7. Choose whether to add Anaconda to your PATH environment variable. We recommend not adding Anaconda to the PATH environment variable, since this can interfere with other software. Instead, use Anaconda software by opening Anaconda Navigator or the Anaconda Prompt from the Start Menu.

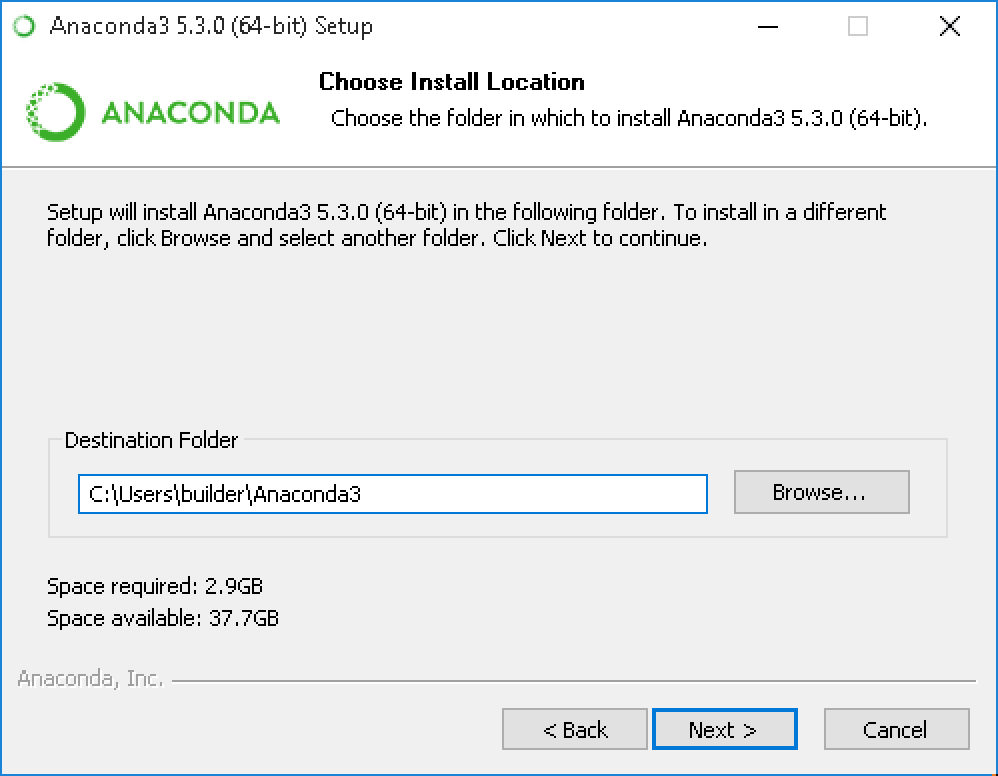


Figure 5.1 Selection a destination folder

1. Choose whetherto the PATH environment variable, since this can interfere with other software. Instead, use Anaconda software by opening Anaconda Navigator or the Anaconda Prompt from the Start Menu.

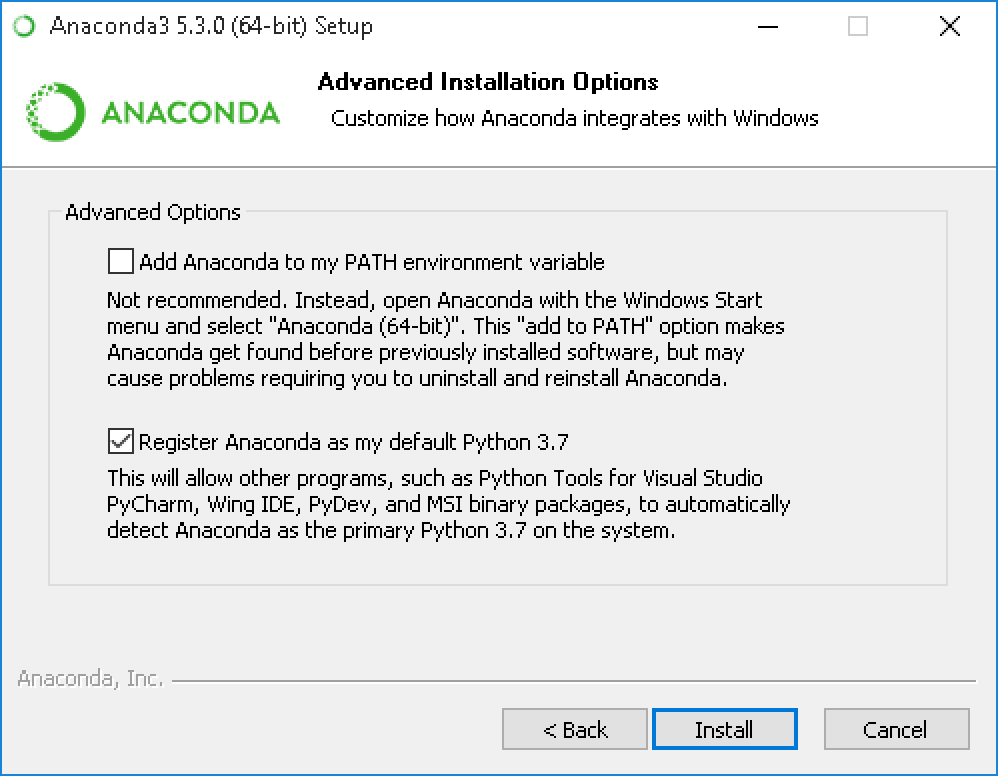


Figure 5.2Choosing whether to register Anaconda as your default Python.

1. Choose whether to register Anaconda as your default Python. Unless you plan on installing and running multiple versions of Anaconda, or multiple versions of Python, accept the default and leave this box checked.
2. Click the Install button. If you want to watch the packages Anaconda is installing, click Show Details.
3. Click the Next button.
4. Optional: To [install VS Code](https://docs.anaconda.com/anaconda/user-guide/tasks/integration/vscode/), click the Install Microsoft VS Code button. After the install completes click the Next button.

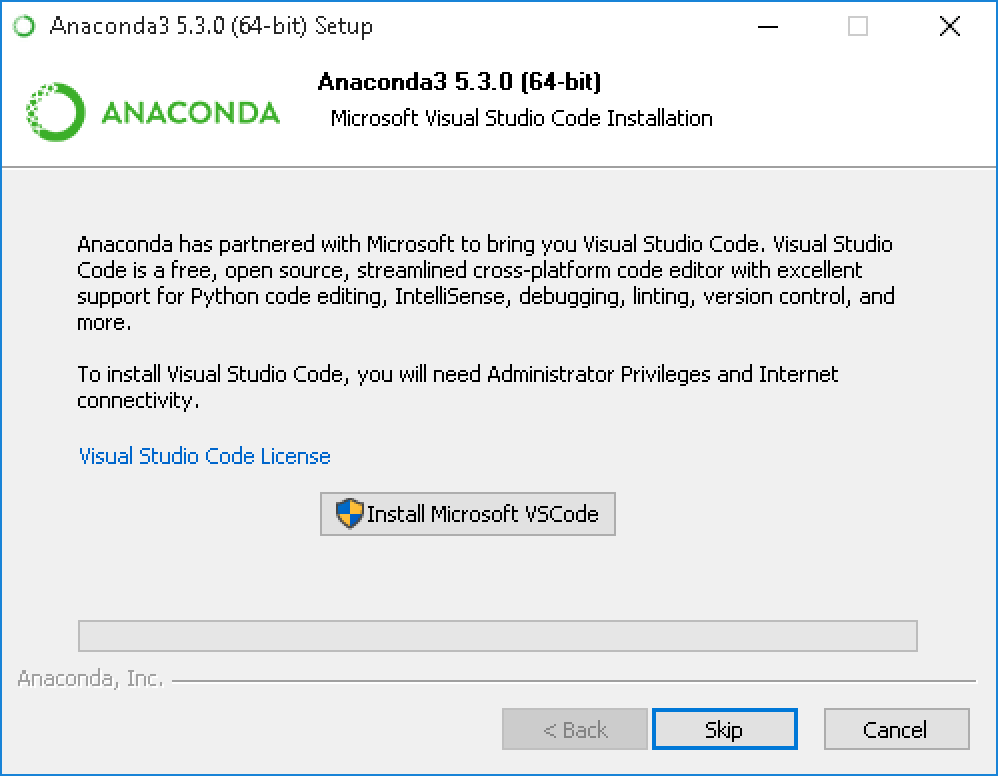


Figure 5.3 Installing or skipping the Vs code or to install Anaconda without VS Code

1. After a successful installation you will see the “Thanks for installing Anaconda” dialog box:

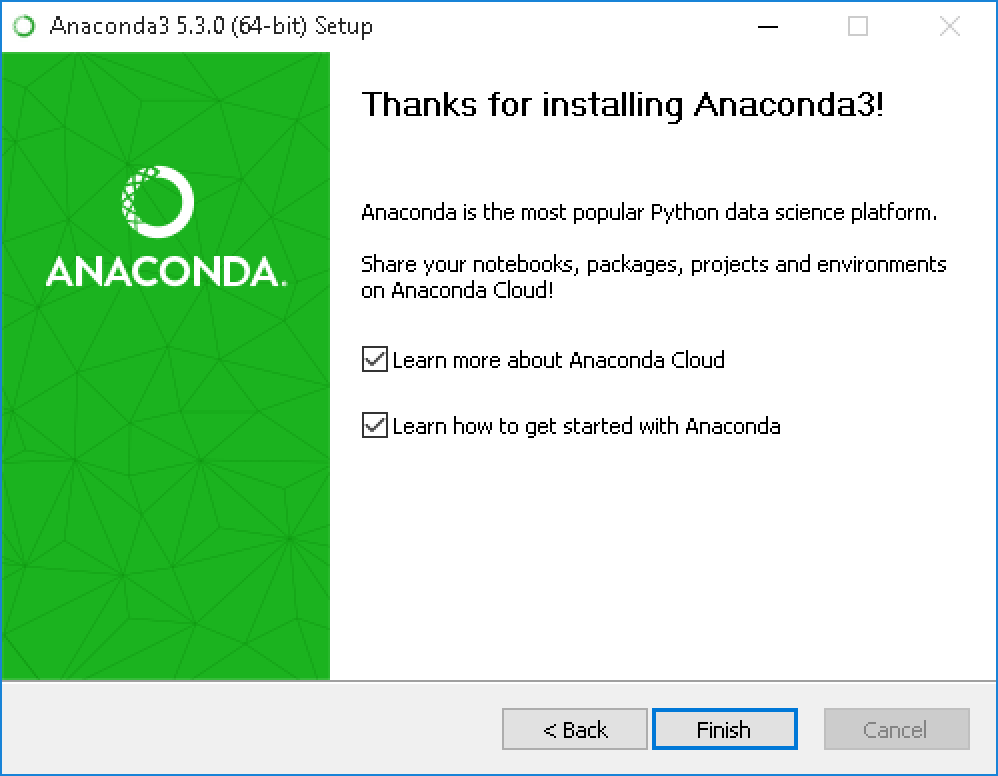


Figure 5.4 Indication of successful installation

1. If you wish to read more about Anaconda Cloud and how to get started with Anaconda, check the boxes “Learn more about Anaconda Cloud” and “Learn how to get started with Anaconda”. Click the Finish button.

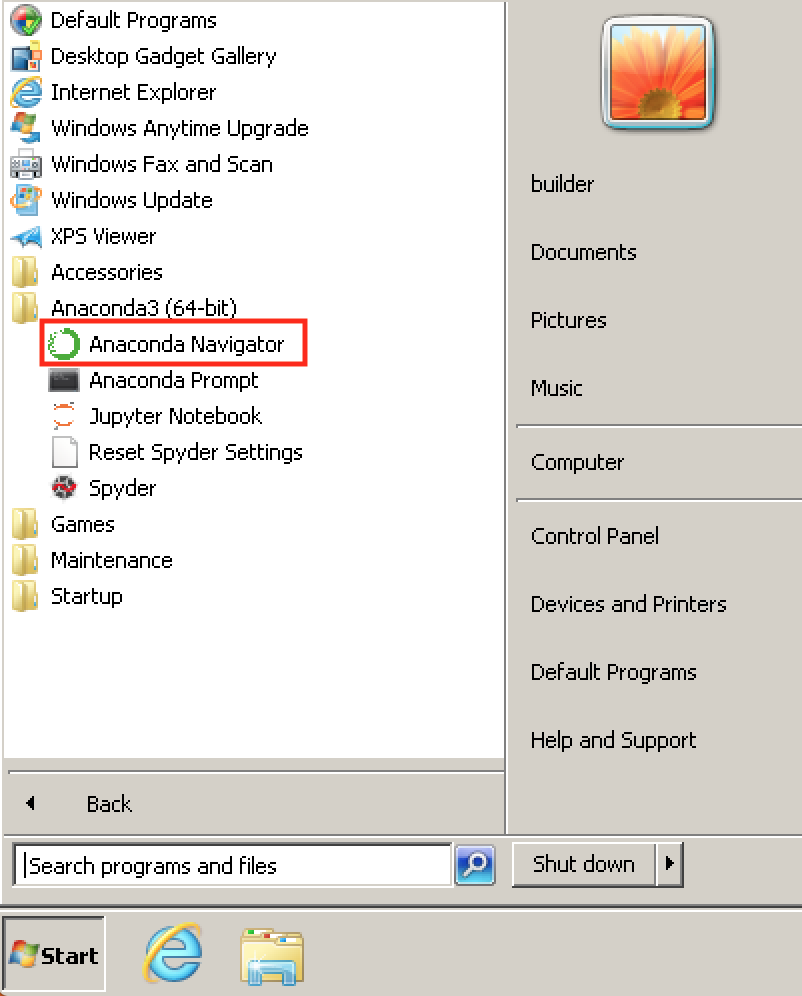


Figure 5.5 Verification by opening Anaconda Navigator from Windows Star menu

1. After your install is complete, verify it by opening Anaconda Navigator, a program that is included with Anaconda: from your Windows Start menu, select the shortcut Anaconda Navigator. If Navigator opens, you have successfully installed Anaconda. If not, check that you completed each step above, then see our Help page.

**SPYDER**

* Spyder is the Scientific Python Development Environment.
* Spyder is an open souce cross-platform interactive development Environment (IDE) for scientific programming in the Python language. Spyder integrates with a number of prominent packages in the scientific Python stack, including NumPy, SciPy, Matplotlib, pandas, IPython, [SymPy](https://en.wikipedia.org/wiki/SymPy) and Cython, as well as other open source software. It is released under the MIT license.
* Initially created and developed by Pierre Raybaut in 2009, since 2012 Spyder has been maintained and continuously improved by a team of scientific Python developers and the community.
* Spyder is extensi and Ubuntu.
* Spyder uses Qt for its GUI, and is designed to use either of the [PyQt](https://en.wikipedia.org/wiki/PyQt" \o "PyQt) or [PySide](https://en.wikipedia.org/wiki/PySide" \o "PySide) Python bindings. QtPy, a thin abstraction layer developed by the Spyder project and later adopted by multiple other packages, provides the flexibility to use either backend.

**General Features:**

The main editor features which are being provided are:

* syntax coloring for Python, C/C++, Fortran;
* breakpoints and conditional breakpoints (debugger: pdb);
* powerful dynamic code introspection features (powered by rope), including code completion and calltips;
* integrated pylint code analysis;
* class and function browser;
* code outline explorer: functions, classes, if/else/try/... statements;
* occurrence highlighting;
* to-do lists (TODO, FIXME, XXX);
* errors and warnings with real-time code analysis (provided by pyflakes);
* opportunity to run a whole script or any portion of it from the editor;
* code completion and automatic link to documentation through the Object Inspector.

[**Editor**](https://docs.spyder-ide.org/editor.html)**:**

* Customizable syntax and highlighting themes.[Debugging](https://docs.spyder-ide.org/debugging.html) breakpoints and conditional breakpoints (through ipdb integration).
* Interactive execution: Run line/selection, run cell, run file, re-run and debug
* Run configuration settings:
  + Working directory selection andCommand line options
  + Run in Current/dedicated/external console choice and Automatically clear variables or enter debugging
* Outline Explorer: Navigate cells, functions, classes, blocks, and more Real-time code introspection features (powered by rope and jedi):
  + On-demand (Tab) and “dot” automatic code completion
  + Automatic popup calltips showing function signatures
  + Go-to-definition for any symbol: Functions, classes, attributes, etc. (Ctrl-Click or Ctrl-G by default)

**Basic Console features:**

* all consoles are executed in a separate process;
* open Python interpreters or basic terminal command windows;
* many Python and Ipython consoles work simultaniously;
* Python interpreter;
* running Python script;
* User Module Deleter ( forces the Python interpreter to reload modules completely when executing a Python script);
* variable explorer: GUI-based editors for a lot of data types (numbers, strings, lists, arrays, dictionaries, etc.); import/export data from/to a lot of file types (text files, !NumPy files, MATLAB files); multiple array/list/dict editor instances at once, thus allowing to compare variable contents; data visualization.

Other features include history log, Object inspector (provides documentation or source code that may be displayed as an html page due to the rich text mode, powered by sphinx), and Project Explorer (support Pydev project import). Spyder is a powerful IDE for scientific programming in Python. It is efficient and free alternative to MatLab.

**Python**

After your install is complete, verify it by opening Anaconda Navigator, a program that is included with Anaconda: from your Windows Start menu, select the shortcut Anaconda Navigator. If Navigator opens, you have successfully installed Anaconda. If not, check that you completed each step above, then see our Help page.

**Main features of Python:**

* Uses an elegant syntax, making the programs you write easier to read.
* Is an easy-to-use language that makes it simple to get your program working. This makes Python ideal for prototype development and other ad-hoc programming tasks, without compromising maintainability.
* Comes with a large standard library that supports many common programming tasks such as connecting to web servers, searching text with regular expressions, reading and modifying files.
* Python's interactive mode makes it easy to test short snippets of code. There's also a bundled development environment called IDLE.
* Is easily extended by adding new modules implemented in a compiled language such as C or C++.
* Can also be embedded into an application to provide a programmable interface.
* Runs anywhere, including Mac OS , [Windows](https://www.python.org/downloads/windows/), [Linux](https://docs.python.org/3/using/unix.html), and [Unix](https://docs.python.org/3/using/unix.html), with unofficial builds also available for [Android](https://wiki.python.org/moin/Android) and iOS.

**Some programming-language features of Python are:**

* A variety of basic data types are available: numbers (floating point, complex, and unlimited-length long integers), strings (both ASCII and Unicode), lists, and dictionaries.
* Python supports object-oriented programming with classes and multiple inheritance.
* Code can be grouped into modules and packages.
* The language supports raising and catching exceptions, resulting in cleaner error handling.
* Data types are strongly and dynamically typed. Mixing incompatible types (e.g. attempting to add a string and a number) causes an exception to be raised, so errors are caught sooner.
* Python contains advanced programming features such as generators and list comprehensions.
* Python's automatic memory management frees you from having to manually allocate and free memory in your code.

**IPython** (Interactive Python) is a [command shell](https://en.wikipedia.org/wiki/Shell_(computing)) for interactive computing in multiple programming languages, originally developed for the [Python programming language](https://en.wikipedia.org/wiki/Python_(programming_language)), that offers [introspection](https://en.wikipedia.org/wiki/Introspection_(computer_science)), [rich media](https://en.wikipedia.org/wiki/Rich_media), shell syntax, [tab completion](https://en.wikipedia.org/wiki/Tab_completion), and history. IPython provides the following features:

* Interactive shells (terminal and [Qt](https://en.wikipedia.org/wiki/Qt_(framework))-based).
* A browser-based [notebook interface](https://en.wikipedia.org/wiki/Notebook_interface) with support for code, text, mathematical expressions, inline plots and other media.
* Support for interactive data visualization and use of GUI toolkits.
* Flexible, embeddable interpreters to load into one's own projects.
* Tools for [parallel computing](https://en.wikipedia.org/wiki/Parallel_computing).

## Parallel computing:

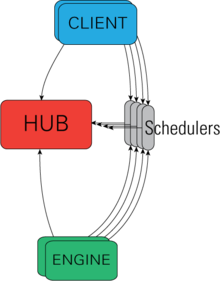
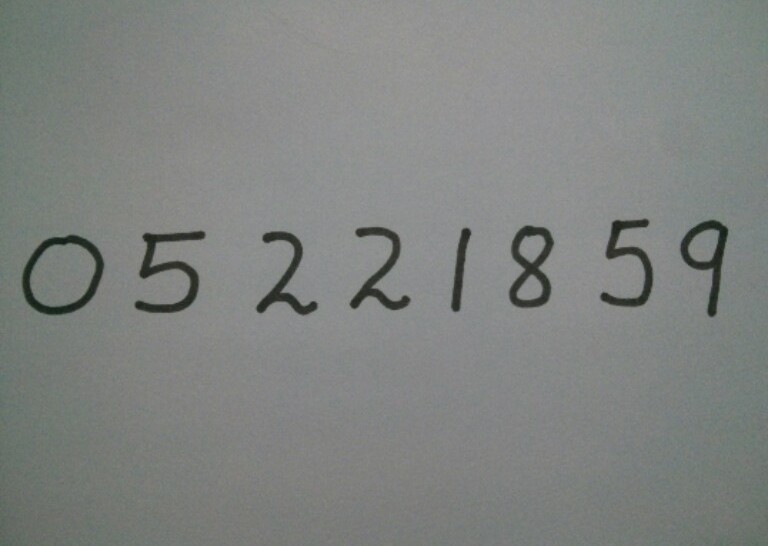
[](https://en.wikipedia.org/wiki/File:IpythonArchitecture.png)

Figure 5.6 Architectural View of IPython's parallel machinery

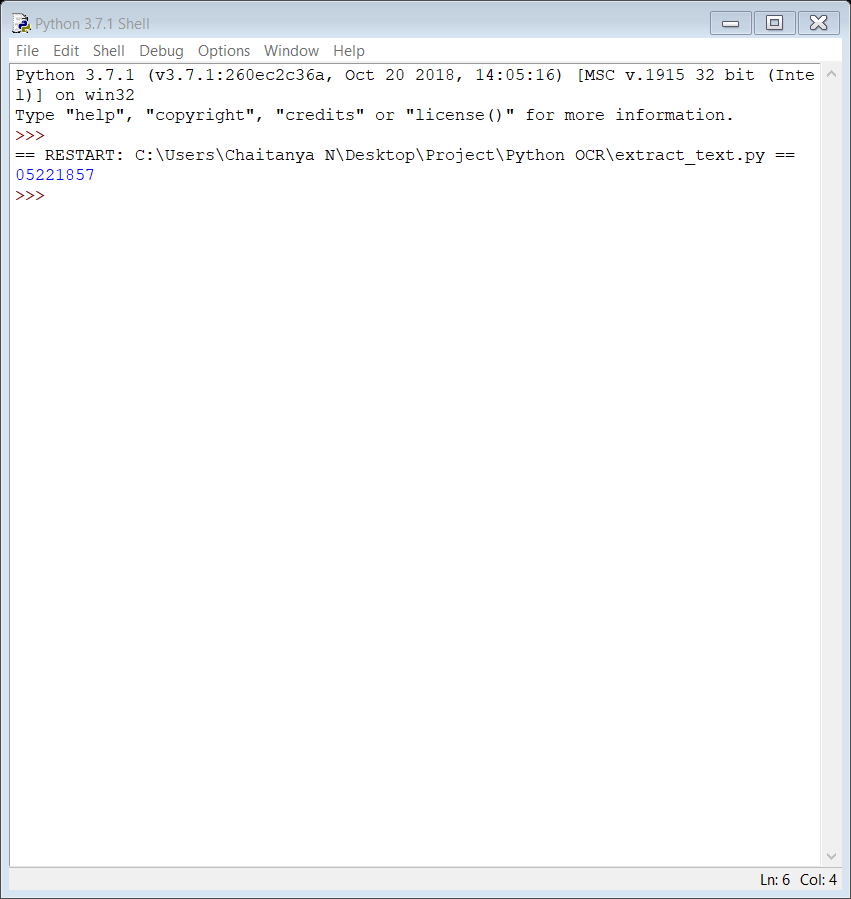
IPython is based on an architecture that provides parallel and distributed computing. IPython enables parallel applications to be developed, executed, debugged and monitored interactively, hence the I (Interactive) in IPython.[[3]](https://en.wikipedia.org/wiki/IPython#cite_note-3) This architecture abstracts out parallelism, enabling IPython to support many different styles of parallelism[[4]](https://en.wikipedia.org/wiki/IPython#cite_note-4) including:

* Single program, multiple data ([SPMD](https://en.wikipedia.org/wiki/SPMD)) parallelism
* Multiple program, multiple data ([MIMD](https://en.wikipedia.org/wiki/MIMD)) parallelism
* Message passing using [MPI](https://en.wikipedia.org/wiki/Message_Passing_Interface)
* [Task parallelism](https://en.wikipedia.org/wiki/Task_parallelism)
* [Data parallelism](https://en.wikipedia.org/wiki/Data_parallelism)
* Combinations of these approaches
* Custom user defined approaches

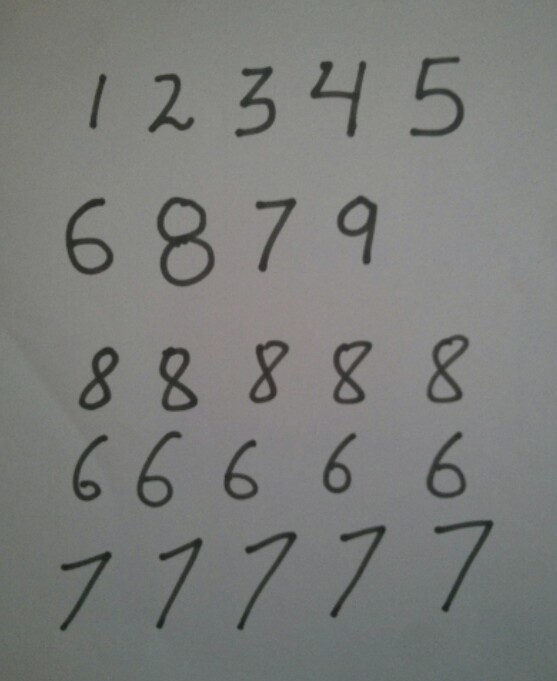
**Digit recognition using Optical Character recognizer**

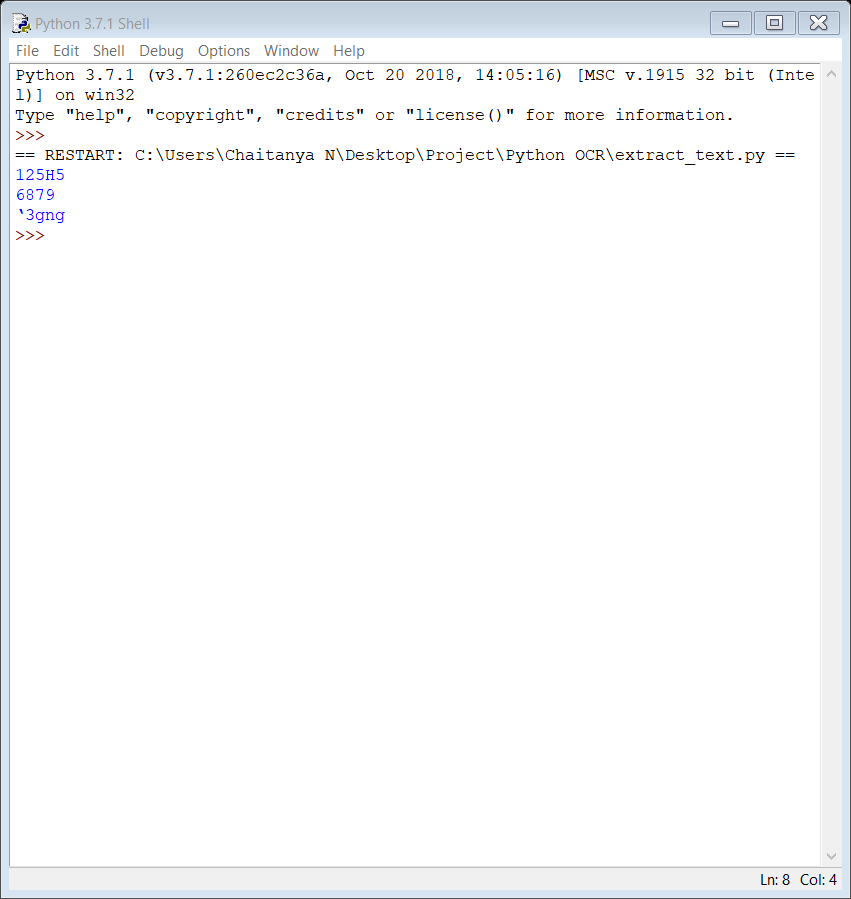


The above is given as an input to the optical character recognizer using the python tesseract OCR for digit recognition. The output of the optical character recognizer is shown in the below figure



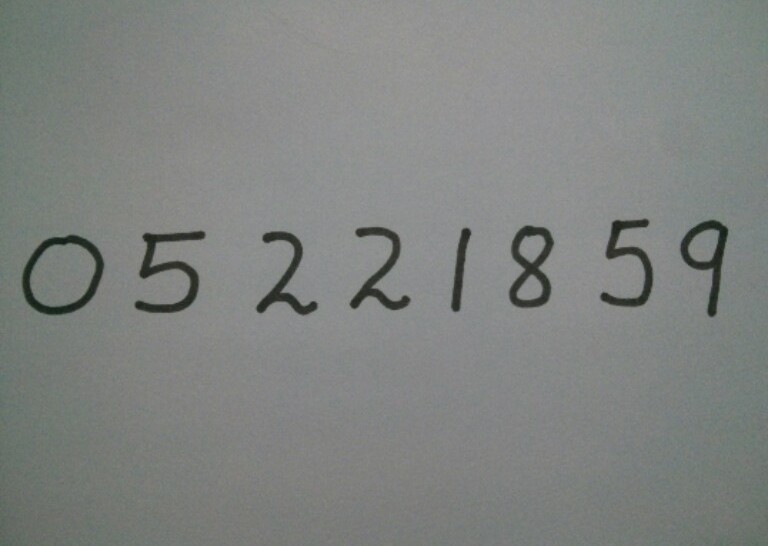
If the second input to the optical character recognizer is the image given below then the output of the optical character recognizer is as shown in the figure



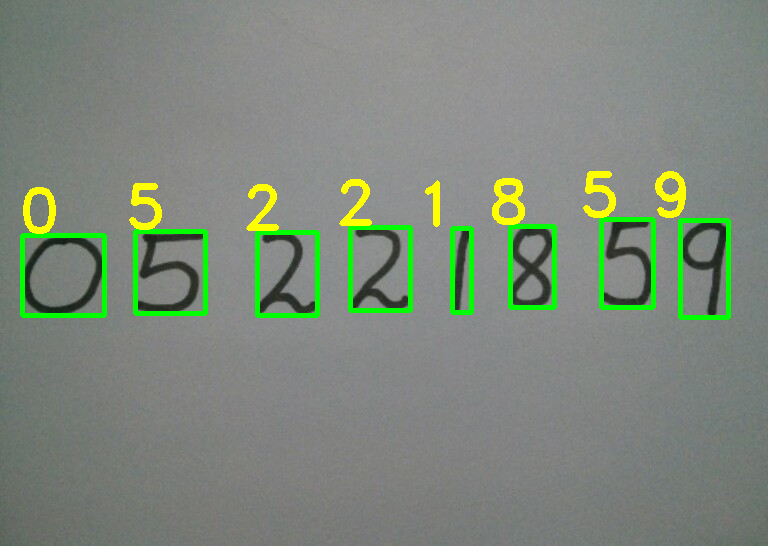


This is the output for the second image using optical character recognizer

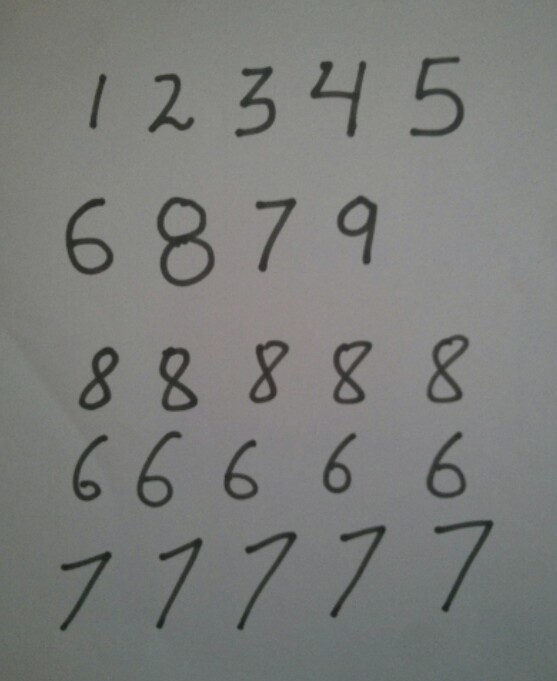
**Neural networks using the Support Vector Machine algorithm**

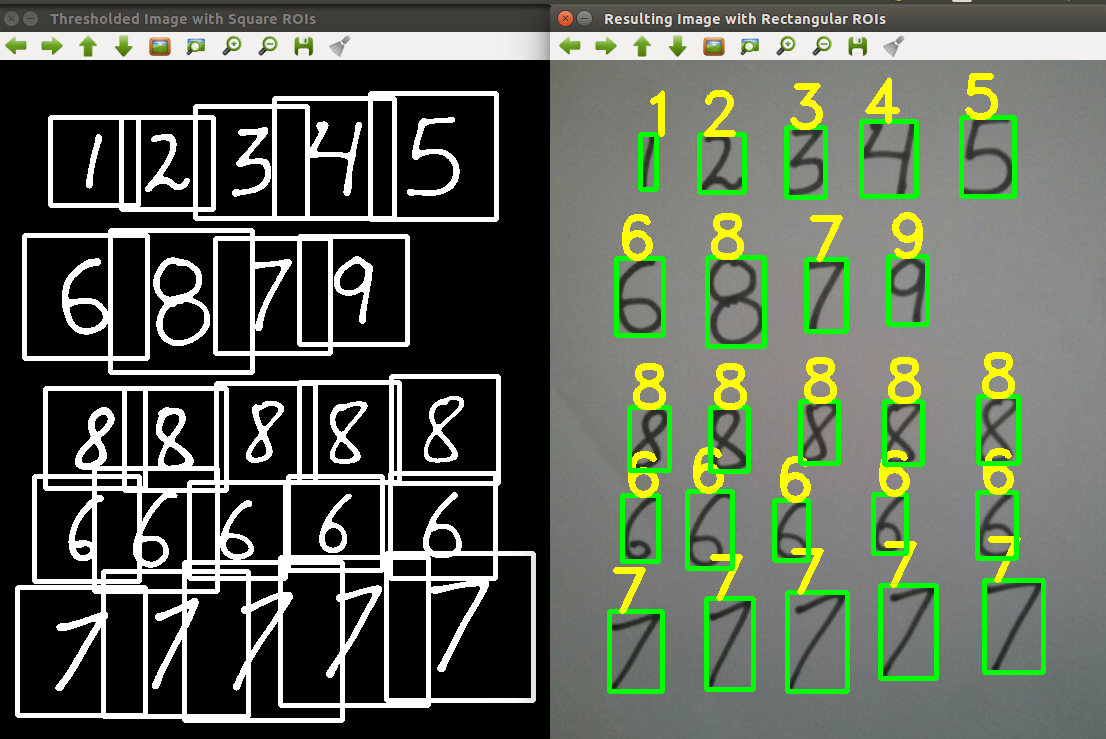


The above is given as an input to the neural network using the SVM algorithm and MNIST database for digit recognition. The output of the neural network is shown



If the second input to the neural network is the image given below then the output of the neural network is as shown in the figure





This is the output for the second input to the neural network

**APPLICATIONS**

1. **Institutional repositories** are digital collections of the outputs created within a university or research institution. It is an online locale of intellectual data of an institution, especially a research institution where it is collected, preserved and aired. It helps to open up the outputs of an institution and give it visibility and more impact on worldwide level.

2. **Invoice imaging** is widely used in many businesses applications to keep track of financial records and prevent a backlog of payments from pilling up. In government agencies and independent organizations, OCR simplifies data collection and analysis, among other processes.

3.**Automatic number plate recognition** [6] is used as a mass surveillance technique making use of optical character recognition on images to identify vehicle registration plates. ANPR has also been made to store the images captured by the cameras including the numbers captured from the license plate.

4. The **legal industry** is also one of the beneficiaries of the OCR technology. OCR is used to digitize documents and directly entered into a computer database.

5. Another important application of OCR is in **banking**, where it is used to process cheques without human involvement. Cheque can be inserted into a machine where the system scans the amount to be issued and the correct amount of accessed as necessary

6. **Healthcare** has also seen an increase in the use of OCR technology to process paperwork. Healthcare professionals always have to deal with large volumes of forms for each patient, including insurance forms as well as general health forms. To keep up with all of this information, it is useful to input relevant data into an electronic database that can be accessed as necessary.

**CONCLUSION**

This concentrates on adaptive methods that operate directly on size-normalized images. Compares the relative merits of Neural Network based handwritten digit recognition system and Optical Character Recognizer. The problems faced are due to diverse text patterns and various background interferences and the image distortion. The printed documents and handwritten documents are transformed to ASCII files for the purpose of compact storage, editing, fast retrieval and other file manipulations through the use of neural networks and optical character recognition.

So Neural networks are used for classification and identification as it is faster and it is a more efficient than optical character recognition. The accuracy of Neural networks was 7% superior than optical character recognition. The Neural network consists of data set extraction, pre-processing, classification, recognition and post-processing stages.

It is widely used as a form of information entry from printed paper data records, whether passport documents, invoices, bank statements, computerized receipts, business cards, mail, printouts of static data, or any suitable documentation. By this we can automate the repeated work.