**CHAPTER 4**

**DEVELOPMENT PROCESS**

* 1. **REQUIREMENT ANALYSIS**

Requirements are a feature of a system or description of something that the system is capable of doing in order to fulfil the system’s purpose. It provides the appropriate mechanism for understanding what the customer wants, analysing the needs assessing feasibility, negotiating a reasonable solution, specifying the solution unambiguously, validating the specification and managing the requirements as they are translated into an operational system.

* + 1. **PYTHON:**

Python is a dynamic, high level, free open source and interpreted programming language. It supports object-oriented programming as well as procedural oriented programming. In Python, we don’t need to declare the type of variable because it is a dynamically typed language.

For example, x=10. Here, x can be anything such as String, int, etc.

Python is an interpreted, object-oriented programming language similar to PERL, that has gained popularity because of its clear [syntax](https://whatis.techtarget.com/definition/syntax) and readability. Python is said to be relatively easy to learn and portable, meaning its statements can be interpreted in a number of [operating system](https://whatis.techtarget.com/definition/operating-system-OS)s, including UNIX-based systems, Mac OS, MS-DOS, OS/2, and various versions of Microsoft Windows 98. Python was created by Guido van Rossum, a former resident of the Netherlands, whose favourite comedy group at the time was Monty Python's Flying Circus. The source code is freely available and open for modification and reuse. Python has a significant number of users.

**Features in Python**

There are many features in Python, some of which are discussed below

* Easy to code
* Free and Open Source
* Object-Oriented Language
* GUI Programming Support
* High-Level Language
* Extensible feature
* Python is Portable language
* Python is Integrated language
* Interpreted Language
  1. **ANACONDA**

Anaconda distribution comes with over 250 packages automatically installed, and over 7,500 additional open-source packages can be installed from [PyPI](https://en.wikipedia.org/wiki/Python_Package_Index" \o "Python Package Index) as well as the [conda](https://en.wikipedia.org/wiki/Conda_(package_manager)" \o "Conda (package manager)) package and virtual environment manager. It also includes a GUI, Anaconda Navigator,[[12]](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)#cite_note-12) as a graphical alternative to the command line interface (CLI).

The big difference between conda and the [pip package manager](https://en.wikipedia.org/wiki/Pip_(package_manager)) is in how package dependencies are managed, which is a significant challenge for Python data science and the reason conda exists.

When pip installs a package, it automatically installs any dependent Python packages without checking if these conflict with previously installed packages. It will install a package and any of its dependencies regardless of the state of the existing installation. Because of this, a user with a working installation of, for example, Google Tensorflow, can find that it stops working having used pip to install a different package that requires a different version of the dependent numpy library than the one used by Tensorflow. In some cases, the package may appear to work but produce different results in detail.

In contrast, conda analyses the current environment including everything currently installed, and, together with any version limitations specified (e.g. the user may wish to have Tensorflow version 2,0 or higher), works out how to install a compatible set of dependencies, and shows a warning if this cannot be done.

Open source packages can be individually installed from the Anaconda repository, Anaconda Cloud (anaconda.org), or the user's own private repository or mirror, using the conda install command. Anaconda, Inc. compiles and builds the packages available in the Anaconda repository itself, and provides binaries for Windows 32/64 bit, Linux 64 bit and MacOS 64-bit. Anything available on [PyPI](https://en.wikipedia.org/wiki/Python_Package_Index" \o "Python Package Index) may be installed into a conda environment using pip, and conda will keep track of what it has installed itself and what pip has installed.

Custom packages can be made using the conda build command, and can be shared with others by uploading them to Anaconda Cloud, [PyPI](https://en.wikipedia.org/wiki/Python_Package_Index) or other repositories.

The default installation of Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.7. However, it is possible to create new environments that include any version of Python packaged with conda.

### Anaconda Navigator

Anaconda Navigator is a desktop [graphical user interface (GUI)](https://en.wikipedia.org/wiki/Graphical_user_interface) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using [command-line commands](https://en.wikipedia.org/wiki/Command-line_interface). Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for [Windows](https://en.wikipedia.org/wiki/Windows), [macOS](https://en.wikipedia.org/wiki/MacOS) and [Linux](https://en.wikipedia.org/wiki/Linux).

The following applications are available by default in Navigator:[[16]](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)#cite_note-16)

* [JupyterLab](https://en.wikipedia.org/wiki/Project_Jupyter#JupyterLab)
* [Jupyter Notebook](https://en.wikipedia.org/wiki/Project_Jupyter#Jupyter_Notebook)
* QtConsole
* [Spyder](https://en.wikipedia.org/wiki/Spyder_(software))
* [Glue](https://en.wikipedia.org/wiki/Glue_(software))
* [Orange](https://en.wikipedia.org/wiki/Orange_(software))
* [RStudio](https://en.wikipedia.org/wiki/RStudio)
* [Visual Studio Code](https://en.wikipedia.org/wiki/Visual_Studio_Code)
  + 1. **JUPYTER NOTEBOOK**

Jupyter [Notebook](https://en.wikipedia.org/wiki/Notebook_interface) (formerly IPython Notebooks) is a [web-based interactive](https://en.wikipedia.org/wiki/Rich_Internet_application) computational environment for creating Jupyter notebook documents. The "notebook" term can colloquially make reference to many different entities, mainly the Jupyter [web application](https://en.wikipedia.org/wiki/Web_application), Jupyter Python web server, or Jupyter document format depending on context. A Jupyter Notebook document is a [JSON](https://en.wikipedia.org/wiki/JSON) document, following a versioned schema, containing an ordered list of input/output cells which can contain code, text (using [Markdown](https://en.wikipedia.org/wiki/Markdown)), mathematics, plots and rich media, usually ending with the ".ipynb" extension.

Jupyter Notebook can connect to many kernels to allow programming in different languages. By default, Jupyter Notebook ships with the IPython kernel. As of the 2.3 release[[11]](https://en.wikipedia.org/wiki/Project_Jupyter#cite_note-releasenote23-11)[[12]](https://en.wikipedia.org/wiki/Project_Jupyter#cite_note-releasenote20-12) (October 2014), there are currently 49 Jupyter-compatible kernels for many programming languages, including [Python](https://en.wikipedia.org/wiki/Python_(programming_language)), [R](https://en.wikipedia.org/wiki/R_(programming_language)), [Julia](https://en.wikipedia.org/wiki/Julia_(programming_language)) and [Haskell](https://en.wikipedia.org/wiki/Haskell_(programming_language)).

The Notebook interface was added to IPython in the 0.12 release[[14]](https://en.wikipedia.org/wiki/Project_Jupyter#cite_note-releasenote012-14) (December 2011), renamed to Jupyter notebook in 2015 (IPython 4.0 – Jupyter 1.0). Jupyter Notebook is similar to the notebook interface of other programs such as [Maple](https://en.wikipedia.org/wiki/Maple_(software)), [Mathematica](https://en.wikipedia.org/wiki/Mathematica), and [SageMath](https://en.wikipedia.org/wiki/SageMath" \o "SageMath), a computational interface style that originated with Mathematica in the 1980s. According to [*The Atlantic*](https://en.wikipedia.org/wiki/The_Atlantic), Jupyter interest overtook the popularity of the Mathematica notebook interface in early 2018.

* 1. **RESOURCE REQUIREMENTS:**

**SOFTWARE REQUIREMENTS**:

|  |  |
| --- | --- |
| Operating System | Windows 7or later |
| Simulation Tool | Visual Studio Code |
| Documentation | Ms – Office |

**HARDWARE REQUIREMENTS:**

|  |  |
| --- | --- |
| CPU type | I5 |
| Ram size | 4GB |
| Hard disk capacity | 80 GB |
| Keyboard type | Internet keyboard |
| Monitor type | 15 Inch colour monitor |
| CD -drive type | 52xmax |

* 1. **PROPOSED SYSTEM**

Facial Emotion Recognition for Paralyzed Individuals (FERPI) employs the Convolutional Neural Networks (FERC) algorithm to discern emotions through facial expressions. Tailored for paralyzed individuals, this algorithm finds applications in medical treatment, education, police investigations, and human-robot interfaces. Utilizing the FER-2013 dataset and tools like OpenCV, Keras, and Tensor Flow, the system is trained for precise emotion identification. The project emphasizes the critical role of emotions in biomedical engineering, psychology, neuroscience, and health, offering a communication and security solution for paralyzed individuals. Real-time email notifications ensure timely expression of emotions, extending the technology to all vehicle applications for a comprehensive impact.

**Advantages of the system: -**

* Recognizing facial emotions aids in security by identifying intentions, particularly beneficial in fields like law enforcement.
* Implementation in the education sector could provide tools for understanding and responding to students' emotional states.
* Improved recognition contributes to more effective and intuitive interactions between humans and robots.
  1. **SYSTEM DESIGN**

**SYSTEM ARCHITECTURE**

**Data collection**

**Data Pre Processing**

**Model implementation**

**Deep Learning**

**Loading the training video into CNN algorithm**

**Face Deduction using Webcam**

**Emotion Detection**

**ANGRY**

**HAPPINESS**

**SAD**

**SURPRISE**

* 1. **SYSTEM MODULES**
* Module 1: Data collection
* Module 2: Data Pre processing
* Module 3: Model implementation
* Module 4: Loading the trained model into GUI
* Model 5: Deduction
* **Module 1 : Data Collection:**Involves collecting image data, specifically targeting a curated subset of facial expression images meticulously chosen to augment model training by excluding potentially misleading content. This approach ensures the optimal training of our facial emotion deduction system, enhancing its accuracy and reliability.
* **Module 2: Data Pre-Processing**

In our project, data preprocessing assumes a pivotal role in elevating image quality for efficient analysis. Various techniques, including noise reduction, contrast enhancement, resizing, color correction, segmentation, and feature extraction, are systematically employed to optimize the dataset, ensuring it is well-prepared for subsequent processing and thorough analysis.

* **Module 3: Model implementation:**

The datasets were trained using Deep learning method with Open CV algorithm. By using this we can able to predict the values of accuracy.

* **Module 4: Loading the trained model into GUI**
* The trained model is loaded into the frame work called Tkinter.
* It is used to show the result in the GUI form
* **Module 5: Deduction**
* Finally, the user can view their face via webcam.
* It can deduct face emotion happy, sad, angry, etc...