**3**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Introduction to Internship**

**3.1 Internship Summary**

**3.2 Purpose**

**3.3 Objective**

**3.4 Scope**

**3.5 Technology and Literature Review**

**3.6 Internship Planning**

**3.6.1 Project Development Approach and Justification**

**3.6.2 Project Effort and Time, Cost Estimation**

**3.6.3 Roles and Responsibilities**

**3.6.4 Group Dependencies**

**3.7 Internship Scheduling**

**3.1 Internship Summary**

This Internship is focused on developing a gesture recognition system using machine learning algorithms to accurately interpret and classify hand gestures in real-time. The system will be trained on a dataset of hand gesture images and will use convolutional neural networks to identify gestures in real-time. The application of this technology can range from sign language recognition to controlling devices with hand gestures, offering a more natural and intuitive user interface.

The project involves collecting a large dataset of hand gestures performed by various individuals, and then training a deep learning model using convolutional neural networks (CNNs) to recognize and classify the gestures. The CNN model is optimized using various techniques, including data augmentation and transfer learning, to improve its accuracy and robustness.

**3.2 Purpose**

The purpose of the project is to make user-friendly application that allows users to interact with a computer or device through hand gestures. This could have a wide range of applications, from gaming and entertainment to industrial and medical fields. Overall, the project aims to improve human-computer interaction by creating a more natural and intuitive interface.

Hand gestures are a powerful means of nonverbal communication that are used to convey meaning, emotion, and intention. They are an integral part of human communication and play an important role in social interaction, cultural expression, and even cognitive development.

One of the primary purposes of hand gestures is to convey information that is not easily expressed through words alone. For example, a simple thumbs-up or thumbs-down gesture can convey approval or disapproval, respectively, without the need for verbal communication. Similarly, pointing gestures can be used to indicate objects or locations, while waving can be used to signal greeting or farewell.

**3.3 Objective**

First objective of this project is to create a complete system to detect, recognize and interpret the hand gestures through computer vision .

Second objective of the project is therefore to provide a new low-cost, high speed and color image acquisition system.

Overall, the objective of hand gesture recognition is to enhance human-computer interaction by providing a more natural and intuitive interface. This has the potential to improve user experience and enable new forms of interaction and control, leading to a wide range of applications and benefits across various fields.

**3.4 Scope**

The use of hand gestures as a means of human-computer interaction has become increasingly popular in recent years, and the scope for this technology is wide and varied.

The scope of this project is to build a real time gesture classification system that can automatically detect gestures in natural lighting condition. In order to accomplish this objective, a real time gesture-based system is developed to identify gestures.

The scope for hand gesture recognition technology is vast, and its applications can be seen across a range of industries, improving efficiency, convenience, and user experience.

**3.5 Technology and Literature Review**

The ability to recognize and interpret hand gestures can greatly enhance human-computer interaction by providing a more natural and intuitive interface. In this literature review, we will discuss the various technologies and techniques used in hand gesture recognition.

One of the most commonly used technologies in hand gesture recognition is cameras . These cameras capture depth information, allowing for accurate tracking of hand movements in three dimensions. Machine learning algorithms such as convolutional neural networks (CNNs) are then used to classify the gestures based on the captured data.

Another approach to hand gesture recognition is using wearable sensors such as gloves or wristbands. These sensors can capture various parameters such as finger flexion and acceleration, which can be used to recognize different hand gestures. However, these systems can be more cumbersome to use and may require calibration.

**Technology:**

* In this project I use Python, C Sharp and Unity
* Python Packages likes OpenCV and Mediapipe.

**Python:**



Fig 3.5.1 Python

(Courtesy of https://en.wikipedia.org/wiki/Python\_%28programming\_language%29)

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance.

Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

**C Sharp:**



Fig 3.5.2 C Sharp

(Courtesy of https://www.pluralsight.com/paths/c-10)

C# is a modern, object-oriented programming language developed by Microsoft. It is commonly used for building Windows desktop applications, games, mobile apps, and web applications. C# syntax is similar to other C-style languages, making it relatively easy to learn for those familiar with Java, C++, or JavaScript.

C# includes a wide range of features, including garbage collection, type safety, and automatic memory management, making it a powerful language for building robust and scalable applications. It also supports asynchronous programming with the async and await keywords, making it easy to write responsive and efficient applications.

C# has a large and active developer community, and Microsoft provides extensive documentation and support for the language. It is commonly used in enterprise settings, particularly for building applications that run on the .NET framework. With its many features and wide range of applications, C# is a versatile and powerful language for modern software development.

**Unity & Unity Hub:**



Fig 3.5.3 Unity

(Courtesy of https://scottishgames.net/2021/03/31/case-study-unity/)

Unity is a cross-platform game engine and development tool used to create 2D and 3D games, simulations, and interactive experiences. It provides a user-friendly interface and a vast array of features such as physics, audio, scripting, and animation tools, as well as a wide range of supported platforms, including Windows, macOS, Linux, iOS, Android, and many more. Unity allows developers to easily create and deploy their projects across multiple platforms with ease, making it an excellent choice for indie developers, small studios, and large game development companies alike. Its versatility and flexibility make it one of the most popular game engines in the industry.

**3.6 Internship Planning**

Internship planning is one of the major tasks that are performed during the development of the project. Using Internship planning, the task of finding the size of the project is done and with that total amount of time and cost required for project development is calculated.

Planning of this project was done using a special approach. After getting the project definition, upper-level analysis was performed first. That analysis was confined to the whole project level. That analysis gave the idea about the size and the structure of the project and using that analysis information, planning of the project was done.

**3.6.1 Project Development Approach and Justification**

The approach to develop the software system should follow some systematic way i.e., Software Development Life Cycle. Using the upper-level analysis and the environment of the project, which lifecycle model would fit properly for this project was judged. After deciding the proper software development lifecycle model, the development of this project according to the model was done.

**WATERFALL MODEL:**

The Waterfall Model was the first Process Model to be introduced. It is very simple to understand and use. In a Waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases. The waterfall model is the earliest SDLC approach that was used for software development.

In “The Waterfall” approach, the whole process of software development is divided into separate phases. The outcome of one phase acts as the input for the next phase sequentially. This means that any phase in the development process begins only if the previous phase is complete. The waterfall model is a sequential design process in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Conception, Initiation, Analysis, Design, Construction, Testing, Production/Implementation, and Maintenance.

As the Waterfall Model illustrates the software development process in a linear sequential flow; hence it is also referred to as a Linear-Sequential Life Cycle Model.

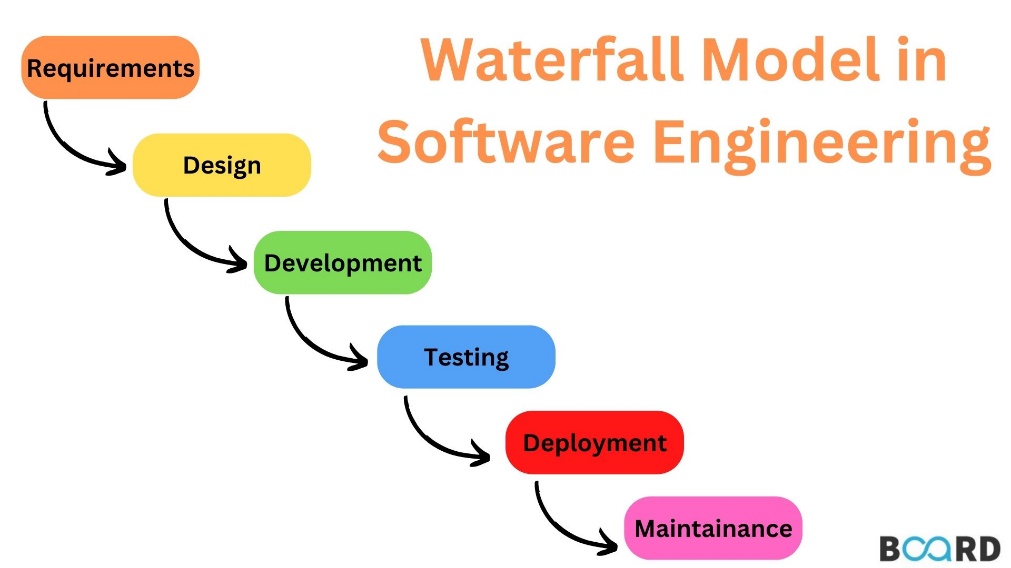


Fig 3.6.1 Waterfall Model

(Courtesy of https://www.boardinfinity.com/blog/a-quick-guide-to-arrays-in-python/)

**3.6.2** **Project Effort and Time, Cost Estimation**

* **Cost Estimation:**

The cost estimation for gesture recognition can vary depending on several factors, including the complexity of the system, the technology used, and the specific application. Here are some factors that could influence the cost:

* + **Hardware:**

The cost of hardware components such as sensors, cameras, and microprocessors can vary based on the quality and type of components used.

* + **Software:**

The cost of developing software to recognize and interpret gestures can also vary depending on the complexity of the system.

* + **Integration:**

The cost of integrating the gesture recognition system with other components, such as a user interface or a larger system, can also impact the overall cost.

* + **Manufacturing:**

The cost of manufacturing the gesture recognition system will depend on the volume of units produced and the manufacturing process used.

* + **Application:**

The specific application of the gesture recognition system can also influence the cost. For example, a gesture recognition system used in a medical device may have higher development and manufacturing costs compared to a gesture recognition system used in a gaming application.

* **Project Effort and Time:**

The effort and time required for a gesture recognition project can vary depending on the scope of the project, the complexity of the system, and the experience level of the development team. Here are some factors that could influence the effort and time required:

* **Research and design:**

Depending on the application and technology used, a significant amount of time may be required to research and design the gesture recognition system. This could include researching available sensors and algorithms, designing the hardware and software components, and creating a user interface.

* **Development:**

Developing the software and hardware components of the gesture recognition system can take a significant amount of time. This could include programming the gesture recognition algorithms, integrating the sensors and microcontrollers, and developing a user interface.

* **Testing and Validation:**

Testing and validation of the gesture recognition system is an essential step in the development process. This can include testing the system's performance in various environments, validating the accuracy of the gesture recognition algorithms, and ensuring the system's safety and reliability.

* **Integration:**

Integrating the gesture con recognition system with other components, such as a user interface or a larger system, can also require significant effort and time.

* **Documentation:**

Creating documentation for the gesture recognition system, such as user manuals, technical specifications, and testing reports, can also require a significant amount of time.

**3.6.3 Roles and Responsibilities**

As an individual working on a gesture recognition project, your role and responsibilities may include:

* **Research:**

Conducting research to understand the technology, available sensors, algorithms, and other components needed to develop a gesture recognition system.

* **Design:**

Designing the hardware and software components of the gesture recognition system, including selecting and integrating the sensors and microcontrollers, developing algorithms for gesture recognition, and creating a user interface.

* **Development:**

Developing the software and hardware components of the gesture recognition system, including programming the gesture recognition algorithms, integrating the sensors and microcontrollers, and developing a user interface.

* **Testing and Validation:**

Testing and validating the gesture recognition system's performance, accuracy, safety, and reliability. This could include testing the system in various environments and ensuring that it meets the project requirements and specifications.

* **Documentation:**

Creating documentation for the gesture recognition system, including user manuals, technical specifications, and testing reports.

* **Project Management:**

Managing the project timeline, resources, and budget, including coordinating with team members, stakeholders, and vendors as needed.

* **Communication:**

Communicating progress, issues, and challenges with team members, stakeholders, and other relevant parties, and providing regular updates on the project's status.

**3.6.4 Group Dependencies**

As working on this project, there are certain dependencies such as:

* + Availability of resources
  + Support from stakeholders
  + Time constraints
  + Budget constraints

**3.7 Project Scheduling**

* Project Scheduling consists of identifying the tasks needed to complete the project, determine the dependency among different tasks, plan the starting and ending dates for various tasks and determine the chain of tasks that determine the duration of the projecting Project scheduling we decide the order in which to do the tasks.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Task | 13/2 | 20/2 | 27/2 | 06/3 | 13/3 | 20/3 | 27/3 | 03/4 | 10/4 | 17/4 | 24/4 | 01/5 | 13/5 |
| Research work |  | | |  | | | | | | | | | |
| Data Collection |  | |  | | |  | | | | | | | |
| Design Model |  | | |  | | |  | | | | | | |
| Implement |  | | | |  | | | | |  | | | |
| Test a Model |  | | | | | |  | | | | |  | |
| Optimization |  | | | | | | |  | | | | |  |
| Documentation |  | | | | | | | | | | |  | |

Fig 3.7.1 Gantt Chart