Design and Development of Smart Electronic Pen

Project Team

Noman Yousaf	20P-0614
Aqil Umar	19P-0034
Haq Nawaz	20P-0057

Session 2020-2024

Supervised by

Dr. Ali Sayyed

Department of Computer Science

National University of Computer and Emerging Sciences
Peshawar, Pakistan

June, 2024

Student's Declaration

We declare that this project titled "*Design and Development of Smart Electronic Pen*", submitted as requirement for the award of degree of Bachelors in Computer Science, does not contain any material previously submitted for a degree in any university; and that to the best of our knowledge, it does not contain any materials previously published or written by another person except where due reference is made in the text.

We understand that the management of Department of Computer Science, National University of Computer and Emerging Sciences, has a zero tolerance policy towards plagiarism. Therefore, We, as authors of the above-mentioned thesis, solemnly declare that no portion of our thesis has been plagiarized and any material used in the thesis from other sources is properly referenced.

We further understand that if we are found guilty of any form of plagiarism in the thesis work even after graduation, the University reserves the right to revoke our BS degree.

Noman Yousaf	Signature: ________________
Aqil Umar	Signature: ________________
Haq Nawaz	Signature: ________________

Verified by Plagiarism Cell Officer
Dated:

Certificate of Approval

The Department of Computer Science, National University of Computer and Emerging Sciences, accepts this thesis titled *Design and Development of Smart Electronic Pen*, submitted by Noman Yousaf (20P-0614), Aqil Umar (19P-0034), and Haq Nawaz (20P-0057), in its current form, and it is satisfying the dissertation requirements for the award of Bachelors Degree in Computer Science.

Supervisor

Dr. Ali Sayyed Signature: ____________________

____

Mr. Haroon Zafar

FYP Coordinator
National University of Computer and Emerging Sciences, Peshawar

____

Mr. Fazle Basit

HoD of Department of Computer Science
National University of Computer and Emerging Sciences

Acknowledgements

We extend our heartfelt gratitude to all those who have supported and contributed to the successful completion of our Final Year Project (FYP).

At the forefront of our gratitude stands our esteemed supervisor, Sir Fazl-e-Basit. His steadfast support, astute guidance, and invaluable feedback have been pivotal throughout this journey. His commitment and discerning counsel have not only propelled our project forward but also shaped its trajectory towards successful fruition.

We also extend our sincere thanks to our co-supervisor, Dr. Muhammad Amin, whose expertise, thoughtful critiques, and continuous encouragement have enriched the depth and quality of our work. His mentorship has been a guiding light, inspiring us to strive for excellence and innovation.

Our appreciation further extends to our friends, whose unwavering support, camaraderie, and willingness to share ideas have been invaluable. Their presence has infused our journey with warmth, motivation, and collaborative spirit.

Lastly, we express our deepest gratitude to our families for their unwavering love, patience, and belief in us. Their constant encouragement and understanding have provided us with the strength and resilience needed to overcome challenges and pursue our aspirations.

To all who have contributed, directly or indirectly, to the success of this project, we offer our sincerest thanks and appreciation.

Thank you.

Noman Yousaf

Aqil Umar

Haq Nawaz

Abstract

Efficient real-time data gathering and interaction with AI models have been challenging due to a lack of context maintenance and prompt refinement. Unstructured data further complicates AI understanding and response. This study introduces a solution that integrates real-time data collection, refines user prompts, enhances GPT model interactions, and improves data comprehensiveness through media processing and text summarization. The objective is to enable context-enriched prompts for informative responses. Real-time data gathering and AI interaction inefficiencies necessitate an innovative solution. Current methods struggle with prompt refinement and context maintenance. Unstructured data compounds these challenges, hindering prompt comprehensibility.

Our solution addresses these issues by integrating real-time data collection, refining user prompts effectively, enhancing GPT model interactions for context-aware responses, and improving data comprehensiveness through media processing and text summarization. The goal is to enrich AI models, enabling context-aware prompts for informative responses.

The success criteria for this study involve the seamless integration of real-time data, enhanced prompt refinement, context-aware AI responses, and comprehensive data utilization. This solution, driven by the need for more efficient data integration and AI interactions, holds promise for diverse applications across domains.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Human history is close to linked to the history of writing instruments. Since the beginning of time, people have looked for ways to transmit ideas, record thoughts, and monitor happenings. In the past, they used sticks or sharp implements to carve symbols onto tablets of clay or stone. Writing changed as civilisations developed, and implements like quills and feathers gained popularity. During the Middle Ages, these ink-dipped writing implements were common and made writing smoother [1].

Writing entered a new era with the creation of the pen. Writing became more convenient and portable with the advent of metal nibs and ink cartridges, which eliminated the need for frequent dipping. Ballpoint pens gained popularity over time because they are dependable and easy to use. But as the digital era ushered in, conventional writing instruments encountered fresh difficulties. The simplicity and adaptability of digital technology needs to be combined with the intimate sense of handwriting[2].

Digital pens, which are instruments made to link handwriting with electronic devices, were created in response to this demand. Features like electronically storing sketches or transforming handwritten notes into digital text were available with early digital pens. By bridging the gap between the digital and physical worlds, these pens enabled users to operate more productively in contemporary settings[3].

By developing a advanced digital pen that offers more than just basic functionality, our project expands on this idea. The strength of contemporary technology is combined with the organic experience of traditional writing in the design of this pen. It records the hand's movements using an optical sensor and converts them into digital data. The pen accurately captures and saves user work in real time, whether they are taking notes or drawing designs.

The unique feature of this pen is its smooth computer integration. Once the handwritten material has been stored, users can examine, edit, or save it on a computer. This feature preserves the joy of handwriting while doing away with the necessity for paper. By doing this, the pen supports environmentally friendly behaviours by increasing convenience and lowering paper waste.

This device's utilisation of cutting-edge technology is at its core. As the user writes or draws, the optical sensor precisely records the hand's movements and records the x and y coordinates. This information is formatted for later display on a computer screen and saved on an SD card. This pen's accuracy and user-friendliness make it an effective tool for anyone who needs to swiftly and digitally record their ideas.

There are lots of domains in which this digital pen can be used. Professionals can rely on it for effective note-taking and document archiving, artists can easily make digital sketches, and educators and students can utilise it for learning and teaching. Because of its adaptability, it may be used for a wide range of jobs, from daily productivity requirements to creative initiatives.

Additionally, the project supports efforts to improve the efficiency of our operations. It offers a modern substitute for traditional writing instruments while lessening the environmental impact by minimising the demand for paper. This pen is more than simply a tool; it represents an innovation in the integration of sustainability, efficiency, and creativity.

In conclusion, this project is the next step in the development of writing instruments. It draws inspiration from the lengthy history of pens, modifies them to meet modern requirements, and produces a gadget that blends in seamlessly with the current digital environment. This pen seeks to transform writing, drawing, and creation for a more sustainable future by fusing tradition and technology.

1.1 Project Purposes:

- The project aims to revolutionize traditional sketching methods by developing a smart pen prototype that eliminates the need for paper or dedicated pads, thereby reducing paper waste and enhancing accessibility to digital sketching tools.
- Through the integration of optical sensor and microcontroller, the project seeks to capture hand movements in real-time, converting them into digital format and offering users a seamless transition from analog to digital drawing experiences.
- By providing users with a versatile and Eco-friendly tool for creative expression and note-taking, the project aims to enhance productivity and reduce environmental impact, ultimately paving the way for a more sustainable approach to digital drawing and design.

1.2 Project Domain:

- The project centers on the realm of digital sketching technology, with a primary focus on revolutionizing traditional sketching methods through the development of an innovative smartpen prototype.
- It involves the integration of advanced sensor technology, including optical sensor to capture hand movements and convert them into digital format seamlessly.
- The project domain extends to various fields such as digital design, creative arts, and productivity tools, aiming to address the challenges of traditional sketching methods and provide users with a versatile and Eco-friendly tool for creative expression.

1.3 Idea of Project:

- At its core, this project revolves around creating a smart pen that bridges the gap between traditional writing and modern digital technology.

- We aim to simplify the process of digitizing handwritten content by capturing hand movements in real-time and converting them seamlessly into digital format using advanced optical sensors.
- The idea is to offer users an innovative, Eco-friendly, and versatile tool that enables smooth and efficient transitions from analog to digital creativity, enhancing productivity and accessibility.

1.4 Scope of Project:

- The project's scope encompasses multiple domains, including creative arts, digital design, education, and professional note-taking.
- It aims to address the limitations of traditional sketching and writing methods by providing an intuitive and sustainable alternative that integrates seamlessly with existing digital ecosystems.
- The smart pen's design ensures versatility, making it suitable for artists, students, educators, and professionals seeking an effective tool for expression and productivity.
- By targeting industries that rely on digital content creation, the project has the potential to redefine workflows, reduce environmental impact, and expand access to innovative sketching and writing technologies.

Chapter 2

Review of Literature

2.1 Summaries

- In order to support environmentally friendly, paperless documentation, the paper [4]An Assistance to Paperless Documentation" offers a framework for digitising handwritten input. The ISWT technology preserves the organic feel of handwriting by using sensors, accelerometers, and gyroscopes to record writing motions and translate them into digital text or visuals. With advantages including less environmental impact and easy adaption to digital platforms, it seeks to optimise operations in business, personal, and educational settings. The technology offers a creative solution for effective and sustainable documentation by addressing issues like real-time accuracy and fluid rendering.

- The handwriting recognition system is presented in the paper [5] It uses a pen with a sensor that records writing data in real-time. The system recognises handwriting online without the need for a writer by using machine learning, more precisely a convolutional neural network (CNN). The suggested method allows handwriting to be efficiently digitised without the need for individual user calibration beforehand. Utilising sensor data, the system avoids the use of conventional optical character recognition (OCR) techniques and exhibits improved accuracy and resilience in a variety of writing styles. For real-time handwriting recognition and digitisation applications across multiple domains, this work has important ramifications.

- The paper [6] provides a thorough analysis of developments in autonomous writing pen technology. It examines several handwriting digitisation techniques and systems, highlighting their uses in the artistic, commercial, and educational domains. The survey discusses the advantages, drawbacks, and room for improvement of several important technologies, including sensor integration, real-time data collection, and handwriting recognition algorithms. The authors also discuss the difficulties in attaining writer adaptability, accuracy, and usability while providing suggestions for future developments in these technologies to enable smooth and effective handwriting digitisation.

- This paper [7] the Trail Making Test (TMT), a cognitive test is administered to older persons using digital pen technology. A cross-over research was done with 40 people (mean age 74.4 years), comparing performance with a digital pen and a traditional pencil. The test results revealed no discernible differences between the two approaches, proving that digital pens can successfully take the place of conventional instruments without sacrificing precision. The study also shown how automated digital pen analysis can increase productivity and objectivity, which makes it a good choice for cognitive evaluations in senior citizens.

- This paper [8] of a feed-forward neural network-based handwriting recognition system using an Arduino Uno. It describes how the system records handwritten inputs' X and Y coordinates, runs them through a trained neural network, and produces precise classification. The authors emphasise how the system's use of affordable technology might improve human-machine interaction, especially in environments with limited resources. The study also addresses the use of machine learning methods to boost handwriting recognition performance and makes recommendations for future developments to improve accuracy and usability in practical applications.

- The paper [9] presents a smart ink pen intended for the ecological evaluation of tremor features and age-related changes in handwriting. The gadget includes sensors to evaluate writing dynamics, including motion and pressure, and records accurate information on handwriting features. In IoT ecosystems, it uses Bluetooth technology to transmit data seamlessly, enabling real-time analysis and storage. The authors highlight how the pen can be used to track tremor patterns and age-related motor changes, both of which are important for the early diagnosis of neurological disorders. The device's flexibility for individualised health monitoring and treatment is suggested by the study, which also highlights the device's usability in everyday and clinical settings.

- The paper [10]introduces an Internet of Things (IoT)-enabled ink pen made for ecological handwriting monitoring in daily life. The device provides real-time data on handwriting dynamics by integrating multiple sensors to detect important metrics including writing force and acceleration. The authors examine how the pen might be used to evaluate handwriting performance in routine tasks, providing information about the accuracy and dependability of writing exercises. They go over how the pen may be used to track and assess variations in handwriting styles, which could help identify motor or cognitive issues early on. The study emphasises the pen's use in both clinical and non-clinical settings, showcasing its adaptability in health monitoring and its role in improving comprehension of handwriting in relation to neurological disorders or ageing.

- The paper [11]An intelligent pen that uses an accelerometer to recognise characters. As the user writes characters on paper, the device records the pen's acceleration and sends this information to an AI algorithm for analysis. The system records acceleration measurements in three directions using a microcontroller and an off-the-shelf accelerometer. To find patterns in the acceleration data, the scientists employ Long Short-Term Memory (LSTM) cells. Because LSTMs can retain longer sequences, they are favoured over conventional Recurrent Neural Networks (RNNs). A dataset of numerical digits (0–9) was used to preprocess and train the neural network. The system's test accuracy of 47% greatly exceeded the baseline guesses accuracy of 10%. The study emphasises the possibility of additional speed enhancements and optimisation with

- The paper [12] demonstrates a deep learning-based method for character identification utilising motion data from handwriting that is recorded by a small smart pen. In contrast to conventional techniques that depend on optical character recognition or specialised equipment, this system tracks hand motion while writing using information from a six-channel inertial sensor and an inertial force sensor. The smart pen prototype has a microcontroller, force sensors, and an ink chamber. Several deep-learning models, including Vision Transformer (ViT), Deep Neural Network (DNN), Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM), were trained using preprocessed data from six volunteers. The ViT model had a validation accuracy of 99.05%, outperforming the other models. Its efficacy was further validated in real time. This research sets the stage for expanding the system to recognize a broader range of characters and involve more subjects.

- The paper [13] presents an innovative method that uses posture-aware sensing techniques to enhance pen + touch interactions on tablets. Flexibility is limited by the fact that current tablet interfaces frequently force users to interact with fixed device-centric UI widgets. In order to get around this, the authors suggest a system that adjusts to different hand grips, screen angles, palm placements, and hand approach directions while being used both fixed and mobile. To identify hand proximity and grab, the system combines three sensing modalities: electric field sensors surrounding the screen bezel, inertial motion sensors, and raw capacitance touchscreen images. By enabling posture-aware pen + touch interactions, these sensors give users a more natural and contextually sensitive interaction experience by enabling UI elements to dynamically adapt to their body, arm, hand, and grip location.

- The paper [14] introduces a digital pen made to recognise handwritten alphabets. The authors concentrate on creating a system that can recognise written characters in real time by combining deep learning algorithms with force and ink sensors. The digital pen can correctly identify and categorise characters as they are written by training the system on a variety of handwritten alphabets. This method has prospective uses in fields like interactive systems and personal digital assistants by fusing hardware and software advancements to enable real-time handwriting detection. Deep learning for character identification improves the pen's accuracy and versatility under various writing circumstances.

- The paper [15] examines interaction using pens in relation to human-computer interaction (HCI). It emphasises how pen-based interfaces have developed and how they are used in contemporary computer systems. The author explores the various ways in which pens are used as input devices, focusing on their role in handwriting recognition, digital drawing, and interaction with graphical user interfaces. The difficulties with pen-based interactions, including user adaptability, ergonomics, and accuracy, are also covered in the article. It also describes possible developments and future areas of inquiry in the area, with the goal of enhancing the incorporation of pen-based input in various technological contexts.

- The paper [16] gives a thorough analysis of hand gesture recognition technology, emphasising developments between 2018 and 2024. It encompasses a range of gesture detection techniques, such as sensor-based, vision-based, and hybrid systems. The authors look at important topics such feature extraction strategies, categorisation schemes, and assistive technology applications, especially those related to sign language recognition. The intricacy of task analysis, location awareness, and dynamic hand gesture detection in gesture-based systems is emphasised in the review. The difficulties and potential paths for enhancing precision, processing in real time, and incorporating gesture detection into commonplace applications are also covered.

- The paper [17] presents a cutting-edge digital pen device intended to improve communication between humans and computers. The study highlights how the pen can record handwriting and motions, providing creative ways to input and interpret data in real time. The authors describe the hardware and software integration of the system and its uses in the fields of education, the arts, and the workplace. Future directions include for enhancing its functionality for greater user interaction as well as increasing its accuracy and adaptability.

- The paper [18] presents a cutting-edge virtual pen input system that makes use of hand trajectory identification and audio tracking. The system employs time-of-flight (ToF) estimation and advanced signal processing algorithms to achieve fine-grained input precision. The V-Pen provides a contactless and effective input method by tracking hand motions via acoustic waves. The system is a promising tool for applications that require accurate handwriting and gesture input because of its accuracy in tracking and character identification, according to the study.

- The paper [19] showcases a technique for tracking hand gestures in real time with the goal of improving human-computer interaction. In order to increase tracking precision and responsiveness, the system combines data from several sensors using sensor fusion techniques. The method guarantees strong performance even under challenging conditions by integrating information from multiple sense modalities. The study highlights the system's potential for use in assistive technologies, robotics, and virtual reality by showcasing its ability to precisely recognise and track gestures.

- The paper [20] presents a novel motion-sensing system based on pressure sensors for hand gesture detection. This device uses a variety of sensors to record the pressure distribution on the wrist, allowing for precise and effective gesture recognition. This technique makes use of the distinct force patterns produced by various hand movements, providing a more reliable solution against ambient interference than conventional techniques that rely on accelerometers or vision-based tracking. Because the system is made for real-time applications, it can be used in assistive technologies and mouse controls, among other human-computer interfaces. Its wearable, small form factor improves usability and opens the door for dependable and smooth gesture-based input systems across a range of industries.

Sr. no	Year	Basic Idea	Methodologies	Results	Limitations
[4]	2020	Interactive Smart Writing Technology (ISWT) is designed to transform handwritten text into a digital format, allowing for seamless paperless documentation by instantly converting handwriting into digital data.	The system capture the user's handwriting. The data is then processed by specialized software that translates the written input into digital text, suitable for various applications.	ISWT effectively converts handwriting into digital text in real time, improving the speed and accuracy of document processing while reducing the reliance on traditional paper-based methods.	The technology's performance may degrade with varying handwriting styles, particularly those that are inconsistent or complex. Additionally, the system faces challenges in scaling to manage large amounts of handwritten data efficiently.
[5]	2020	The paper presents a system that uses a motion-sensing sensor pen to capture and digitize handwriting in real time, allowing for efficient writer-independent handwriting recognition without the need for specialized training for each user.	The system uses a motion-sensing sensor pen equipped with accelerometers and gyroscopes to capture real-time pen movements. The data is processed using convolutional neural networks (CNNs) for writer-independent handwriting recognition, converting the handwritten input into digital text for various applications.	The system successfully converts handwritten content into digital text with high accuracy, achieving robust performance across different writers and handwriting styles. The use of CNNs ensures real-time recognition with minimal latency, making it suitable for practical applications.	The system's performance may decrease with noisy data or when the pen's motion sensors are not properly calibrated. Additionally, the recognition accuracy can be impacted by complex or highly variable handwriting styles, and the system may face challenges in handling large-scale data.
[6]	2023	This paper provides a comprehensive survey on various technologies involved in the development of automatic writing pens, exploring the underlying principles, different types of sensors, and their applications in digitizing handwriting.	The authors analyze various writing pen technologies by reviewing sensor integration (e.g., capacitive, optical, and motion sensors), signal processing techniques, and machine learning algorithms used for handwriting recognition and digital conversion.	The paper highlights the diversity of sensor technologies and their effectiveness in capturing precise handwriting data. It also discusses how automatic pens can provide real-time handwriting conversion, leading to more efficient paperless documentation.	The paper notes the complexity in developing user-independent systems and the cost of integrating multiple sensor types. There are also challenges related to sensor accuracy under varying environmental conditions, such as different paper types or hand movements.
[7]	2021	This study explores the use of digital pen technology in conducting cognitive assessments with older adults, aiming to improve the accuracy and convenience of such assessments while reducing cognitive load.	The authors employed digital pens to capture handwriting during cognitive tasks and analyzed the resulting data to assess cognitive performance. The study involved a cross-over design, where participants used both traditional and digital pen methods to compare the effectiveness of each in terms of accuracy and user experience.	The results showed that digital pen technology facilitated accurate data capture and was well-received by the participants. It enabled easier tracking of cognitive performance, especially in tasks that required writing and drawing, and provided real-time feedback on performance.	The study noted limited generalize ability due to the focus on older adults and the specific cognitive tasks used. Additionally, issues with device calibration and potential technical difficulties with the digital pen were observed, which may affect the system's reliability in varied environments.

Table 2.1: Literature Review

Sr. no	Year	Basic Idea	Methodologies	Results	Limitations
[8]	2024	This paper presents a handwriting recognition system developed using an Arduino Uno microcontroller and a feed-forward neural network, aiming to enhance human-machine interaction by recognizing handwritten text in real-time.	Captures handwriting using a sensor and processes the data through a neural network for character recognition.The Arduino Uno is used to interface with the recognition model, where the data is trained to recognize handwritten characters, and results are displayed on a two-dimensional display.	The proposed system successfully recognized handwritten characters, demonstrating the feasibility of using the Arduino Uno and neural networks for real-time handwriting recognition, offering a low-cost, efficient solution for human-computer interaction.	The recognition accuracy may be influenced by the quality of input data and the variations in handwriting styles. Additionally, the system's performance might be limited by the processing power of the Arduino Uno, affecting the speed and complexity of recognition tasks.
[9]	2021	This paper introduces a smart ink pen designed to assess age-related changes in handwriting and detect tremor features, providing an ecological tool for monitoring motor and cognitive functions over time.	The smart ink pen integrates various sensors to capture handwriting dynamics, including pressure, velocity, and tremor characteristics. The data is transmitted via Bluetooth to a connected device for analysis, enabling the assessment of age-related changes in writing patterns.	Successfully captured and analyzed handwriting features, demonstrating its ability to assess tremor patterns and age-related changes in writing. This innovative tool provides valuable insights into motor function assessment in aging populations.	The pen's performance can be influenced by external factors such as battery life and sensor accuracy. Additionally, the system may face challenges in accurately differentiating between natural tremors and other movement variations, especially in more complex scenarios.
[10]	2020	This paper presents an IoT-enabled ink pen designed for ecological monitoring of daily handwriting activities, aiming to track and analyze writing patterns in real-world settings to better understand cognitive and motor behaviors.	The IoT ink pen incorporates sensors to capture data on force, acceleration, and other handwriting parameters. The data is transmitted to an external device for analysis, enabling real-time monitoring of writing behavior in natural, everyday situations.	The system effectively monitors handwriting activities and provides real-time feedback on writing dynamics, helping to assess cognitive and motor function. The data collected contributes to a more accurate understanding of daily life handwriting in various contexts.	The system's accuracy may be influenced by environmental factors such as sensor calibration and battery life. Additionally, real-time analysis can be limited by the volume of data processed and potential data storage constraints.
[11]	2019	The paper introduces DigiPen, an intelligent pen that utilizes an accelerometer to capture and recognize characters during writing, converting handwritten content into digital text.	The accelerometer in the DigiPen tracks motion and orientation during writing. Data from the pen is processed using algorithms to recognize characters in real-time, enabling digital character recognition without the need for a traditional keyboard or touchscreen.	The system successfully recognized handwritten characters with high accuracy, demonstrating the potential of using motion sensors for efficient character recognition in various applications, such as note-taking and digital writing.	The recognition accuracy may degrade with irregular writing styles or inconsistent pen pressure. Also, the pen's sensitivity to subtle hand movements may require calibration for optimal performance.
[12]	2022	This paper proposes a deep learning approach for character recognition from handwriting motion data captured by an IMU (Inertial Measurement Unit) and force sensors, aiming to improve the accuracy of handwriting recognition systems.	The IMU and force sensors capture handwriting motion, including acceleration, orientation, and pressure.These data are fed to deep learning model for character recognition, enabling the system to recognize and digitize handwriting in real-time.	The system achieved high accuracy in recognizing handwritten characters, even with variations in writing style and pressure, demonstrating the effectiveness of combining motion and force data.	The system may be influenced by sensor noise or variations in writing speed. Additionally, it may struggle with recognizing highly complex or irregular handwriting without further model training.

Table 2.2: Literature Review

Sr. no	Year	Basic Idea	Methodologies	Results	Limitations
[13]	2019	This paper explores a posture-aware interaction system that integrates pen and touch inputs on tablets, using electric field sensing to detect grip and posture for more natural and precise interaction.	The system uses electric field sensing to capture the user's grip and posture during pen-and-touch interactions on a tablet. This data is then used to adjust the interaction model, making it more responsive to the user's physical context.	The system enables accurate posture-aware interactions, improving the usability and precision of pen-and-touch input, especially for applications requiring complex gestures and writing tasks.	The system's performance can be influenced by the tablet's environmental conditions and sensor limitations, which may affect its sensitivity in certain use scenarios.
[14]	2021	This paper presents a digital pen system designed for real-time handwritten alphabet recognition, leveraging force sensors and deep learning for accurate character recognition.	The system uses a digital pen equipped with force sensors to capture handwriting data. The data is then processed using deep learning techniques for alphabet recognition.	The system demonstrates real-time character recognition, achieving high accuracy in recognizing handwritten alphabets, enabling efficient conversion of handwritten input to digital form.	The accuracy of the system may be affected by variations in handwriting styles and sensor noise, which can reduce performance in certain scenarios.
[15]	2023	Focusing on how pen input can enhance human-computer interaction (HCI) by providing intuitive and efficient ways to interact with digital systems.	various pen-based technologies and their applications in HCI, including stylus-based input systems, touchscreens, and digitizing tablets, analyzing their benefits in user interaction.	Pen-based interactions are shown to improve user experience in tasks such as drawing, note-taking, and signature verification, offering more natural and precise input compared to traditional mouse or keyboard-based systems.	Like accuracy in recognition, usability across different environments, and device limitations in terms of sensor performance and gesture recognition.
[16]	2024	The technologies from 2018 to 2024, emphasizing their application in areas like sign language recognition, assistive technologies, and human-computer interaction.	Research on vision-based, sensor-based, and hybrid-based gesture recognition techniques, covering feature extraction, classification methods, and the use of dynamic hand gestures.	More accurate, real-time systems capable of recognizing complex gestures and improving interaction in assistive devices and location awareness systems.	Gesture complexity, sensor reliability, and real-time processing, especially in environments with variable lighting or background noise.
[17]	2023	Mr. Pen, a system designed to enhance pen-based interaction through the integration of digital pen technology, aimed at improving handwriting recognition and supporting various human-computer interaction applications.	Sensors to capture writing patterns, enabling real-time digitization and recognition of handwriting, with an emphasis on accuracy and user experience.	Improved handwriting recognition accuracy and performance, offering a potential solution for applications in education, assistive technology, and digital note taking.	Affected by sensor calibration, environmental conditions, and individual variations in handwriting, which may impact the real-time recognition and overall usability of the system.
[18]	2024	V-Pen, an acoustic-based system that combines hand tracking and time-of-flight estimation for fine-grained virtual pen input.	Uses acoustic signals and signal processing algorithms to track hand movements for precise character recognition and trajectory estimation.	V-Pen provides high accuracy in tracking hand movements, suitable for virtual pen input applications.	The system's performance may degrade in noisy environments or with multiple objects interfering with the acoustic signals.

Table 2.3: Literature Review

Sr. no	Year	Basic Idea	Methodologies	Results	Limitations
[19]	2021	Paper presents a real-time hand gesture tracking system for human-computer interaction using multi-sensor data fusion to enhance tracking accuracy and responsiveness.	Multiple sensors (e.g., accelerometers, gyroscopes, and cameras) to track hand gestures, using data fusion algorithms for real-time processing	System achieves high precision in tracking hand gestures in real-time, offering improved performance over traditional single-sensor methods.	Challenges with sensor calibration and may be less effective in complex environments with multiple overlapping gestures.
[20]	2019	This paper proposes a gesture recognition system based on pressure-sensor-based motion sensing to recognize wrist gestures for human-computer interaction.	Uses pressure sensors mounted on the wrist to measure pressure distribution while performing hand gestures. These pressure patterns are processed to recognize gestures in real-time.	Successfully detects hand gestures with high accuracy and can be used as an alternative to traditional gesture recognition methods, such as those using cameras or motion sensors.	Sensor placement and wrist movement constraints, and it may face challenges in complex gesture recognition

Table 2.4: Literature Review

Chapter 3

Project Vision

3.1 Problem Statement

- This project addresses the limitations of conventional sketching and note-taking methods, which primarily rely on paper, lack digital integration, and contribute to environmental waste. Additionally, there are challenges in effectively converting handwritten content into a digital format that fits seamlessly into modern workflows.

3.2 Business Opportunity

This project presents a compelling business opportunity by developing a smart pen that bridges the gap between traditional analog writing and modern digital solutions. As industries evolve, the demand for innovative tools that offer both functionality and sustainability is on the rise. The smart pen can meet this demand by combining the tactile experience of traditional writing with the efficiency of digital technology.

The market for such a device spans multiple sectors, including creative arts, education, and professional environments. Artists and designers can use the pen for digital sketching, while students and teachers benefit from the seamless integration of handwritten notes into digital formats. This versatility makes the product valuable across diverse fields, enhancing productivity and creativity.

Moreover, the shift towards eco-friendly solutions is driving the need for tools that reduce paper consumption without compromising on performance. By offering a sustainable alternative to conventional writing instruments, the smart pen aligns with global efforts to minimize environmental impact while providing an advanced, efficient solution for everyday tasks.

3.3 Objectives

- Develop a smart pen that accurately tracks hand movements in real time and converts them into digital formats, providing a seamless transition from traditional writing to digital documentation.
- Minimize paper usage by offering a sustainable alternative for note-taking, sketching, and creative tasks, contributing to eco-friendly practices.
- Incorporate advanced optical sensors and microcontroller technology to ensure precision, enhance productivity, and deliver an intuitive and smooth user experience.

3.4 Project Scope

- The project's scope encompasses the design and prototyping of a smart pen equipped with optical sensors to track hand movements. It also involves the creation of software for real-time digital conversion and storage. The project serves a wide range of applications, including creative arts, digital design, education, and professional note-taking, providing a modern alternative to traditional methods.

3.5 Constraints

- The project must adhere to constraints such as budget limitations, time restrictions, and technical feasibility during the development phase. It must also comply with environmental standards and ensure proper alignment with handwriting.

3.6 Stakeholders Description

- The stakeholders in this project include developers, end-users (such as artists, students, and teachers), and industry collaborators. Each group plays a critical role in the success of the project.

3.6.1 Stakeholders Summary

The stakeholders in this project include a variety of groups. Developers play a crucial role in designing and creating both the smart pen and its accompanying software. End-users such as students, teachers, artists, and executives are the primary beneficiaries, utilizing the tool to enhance their work in education, creative fields, and organizational tasks. Collaborators and industry partners also contribute by providing valuable insights and ensuring that the product integrates seamlessly into the broader digital ecosystem.

3.6.2 Key High-Level Goals and Problems of Stakeholders

- Developers aim to create an innovative, reliable, and user-friendly product. End-users seek a versatile tool that improves creative and professional workflows while minimizing environmental impact. Industry collaborators focus on ensuring the product's compatibility with current systems and its potential for scalability in diverse applications.

Chapter 4

Software Requirements Specifications

This chapter outlines the functional and non-functional requirements for the smart pen project.

4.1 List of Features

- Real-time Hand Movement Tracking:
- Data Conversion and Storage:
- Integration with Digital Platforms:
- User Profile and Customization:
- Eco-friendly Functionality:

4.2 Functional Requirements

4.2.1 Real-time Hand Movement Tracking

- The smart pen shall track hand movements in real-time using the KA8 optical sensor, which captures LED light.

- The system shall capture and process these movements into x and y values, which represent the hand's position and movement in space.

4.2.2 Data Conversion and Storage

- The pen shall convert captured hand movement data (x, y values) into a digital format in real time.
- The digital data shall be stored on an SD card in text format for later retrieval.
- The stored data can be transferred to digital platforms for viewing and further processing.

4.2.3 Integration with Digital Platforms

- The pen shall seamlessly integrate with various digital platforms, including computers and other devices capable of running Streamlit.
- Data stored on the SD card shall be retrieved and displayed on a computer screen via a Streamlit web application.

4.3 Quality Attributes

4.3.1 Performance

- The pen shall have low-latency in tracking hand movements and converting them to digital output.
- The system should provide smooth and fast data transfer to connected devices, including data retrieval from the SD card.
- The pen must be capable of supporting multiple user interactions simultaneously.

4.3.2 Usability

- The pen shall have a user-friendly interface that is intuitive for both beginners and experienced users.

- The system shall provide a seamless user experience when interacting with both the pen and digital platforms.

4.4 Non-Functional Requirements

4.4.1 Performance

- The pen shall track hand movements with minimal latency and convert them into digital writing in real-time.

- Data transfer from the pen to connected digital platforms shall occur with minimal delay.

- The pen must support multiple simultaneous users in a shared environment without performance degradation.

4.4.2 Usability

- The user interface shall be intuitive, responsive, and easy to navigate.

- The system should provide a seamless and enjoyable experience for users across all devices.

4.5 Non-Functional Requirements

4.5.1 Performance

- The pen shall track hand movements with minimal latency and convert them into digital writing in real time.
- Data transfer from the pen to connected digital platforms shall occur with minimal delay.
- The pen must support multiple simultaneous users in a shared environment without performance degradation.

4.5.2 Usability

- The user interface shall be intuitive, responsive, and easy to navigate.
- The system should provide a seamless and enjoyable experience for users across all devices.

4.6 UML Diagrams

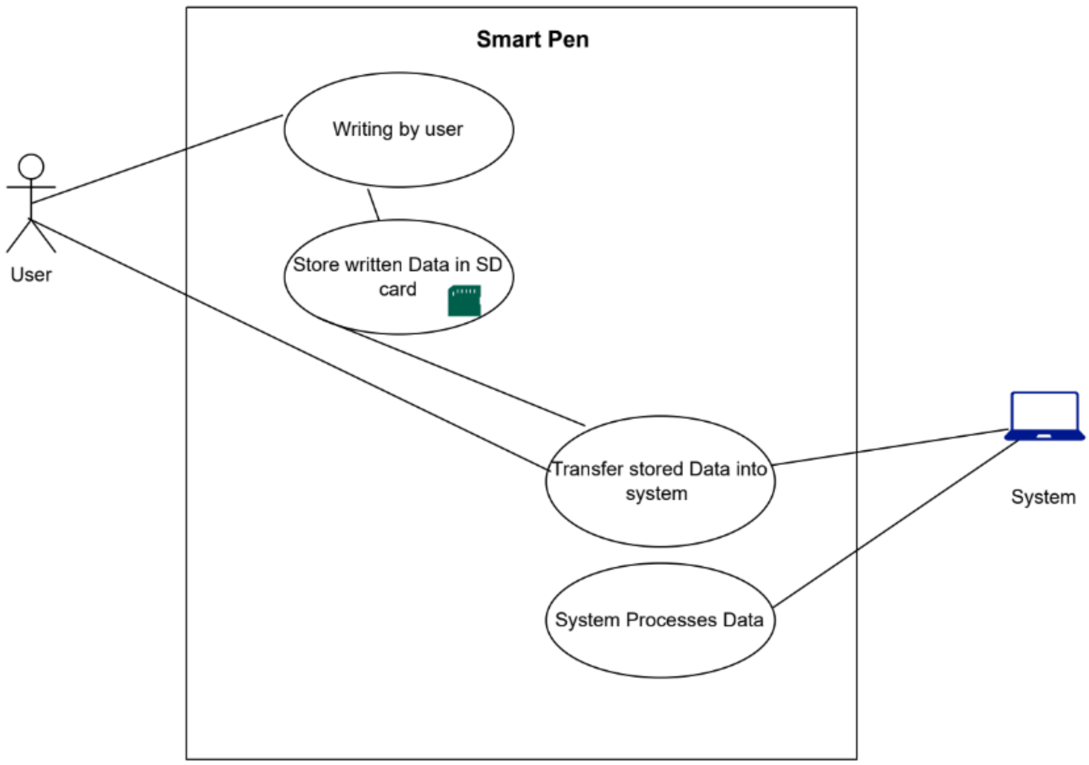

Figure 4.1: Use Case Diagram

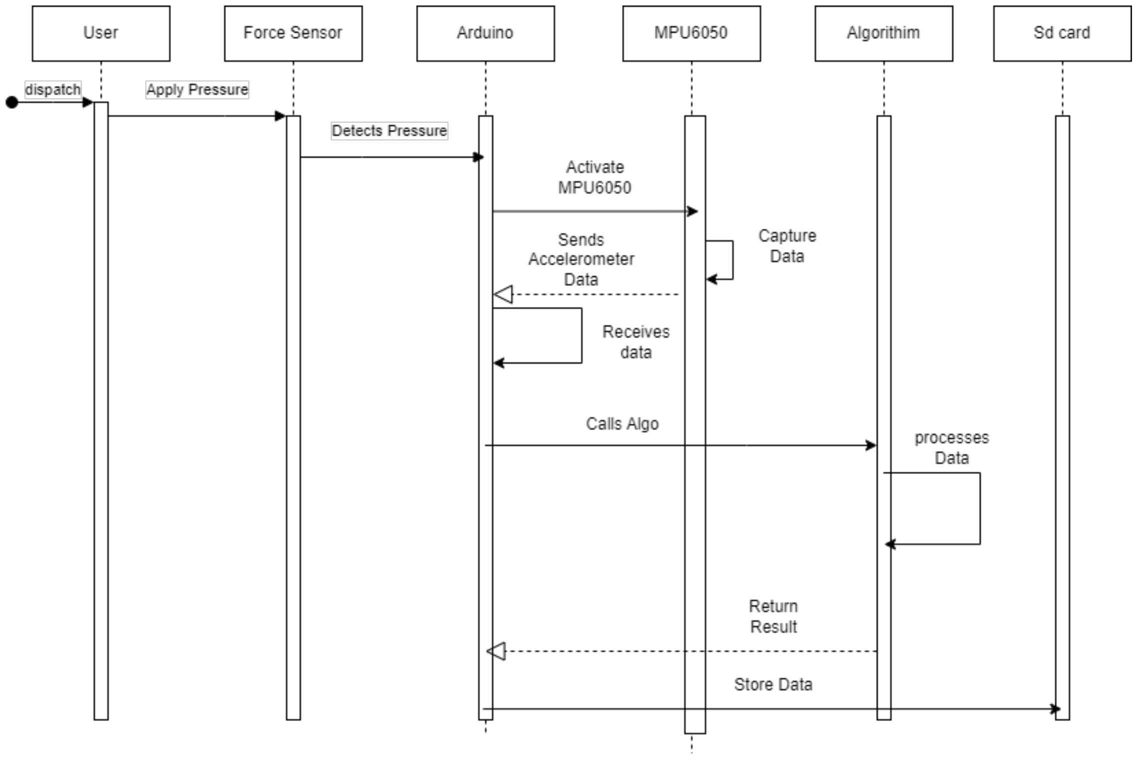

Figure 4.2: Sequence Diagram

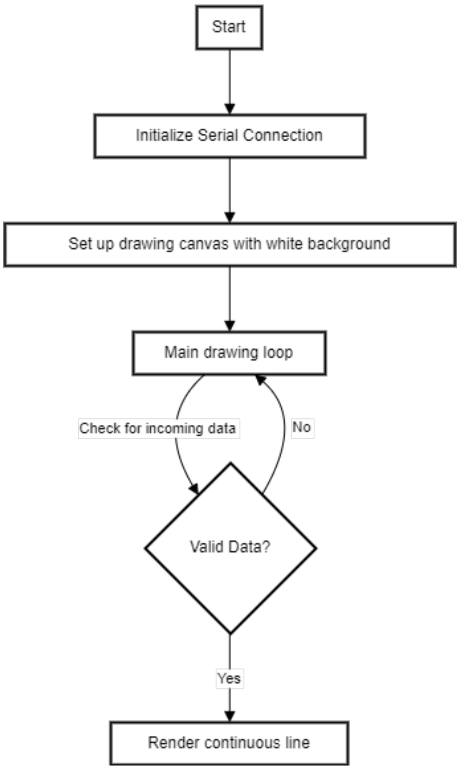

Figure 4.3: Flow Chart

4.7 User Interface

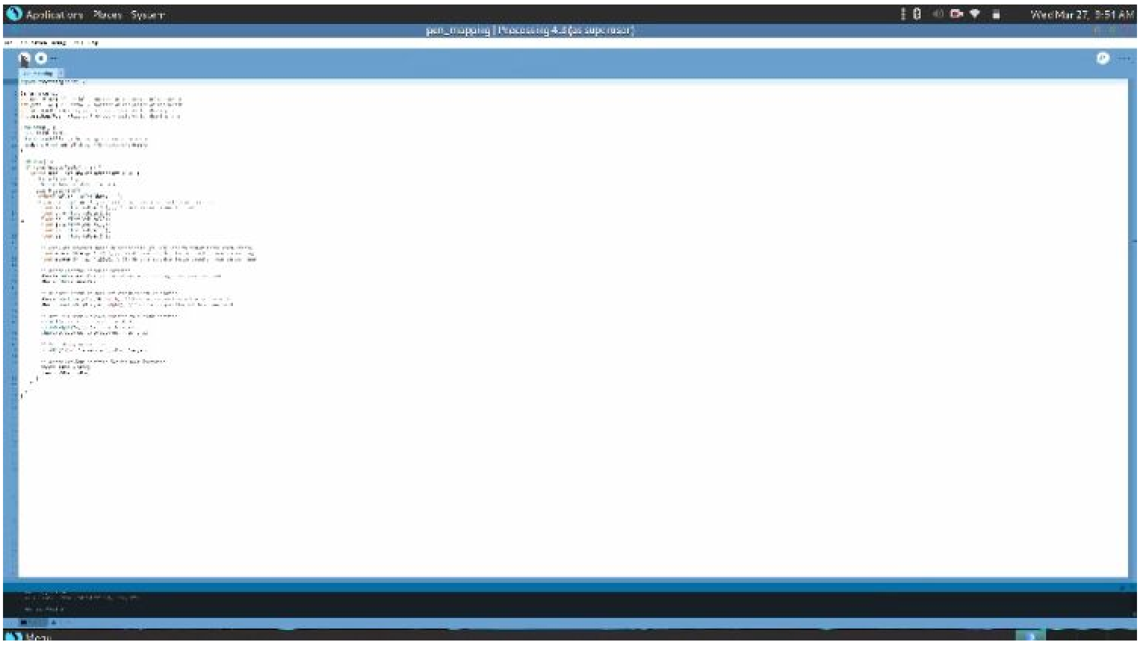

Figure 4.4: Screen no:1

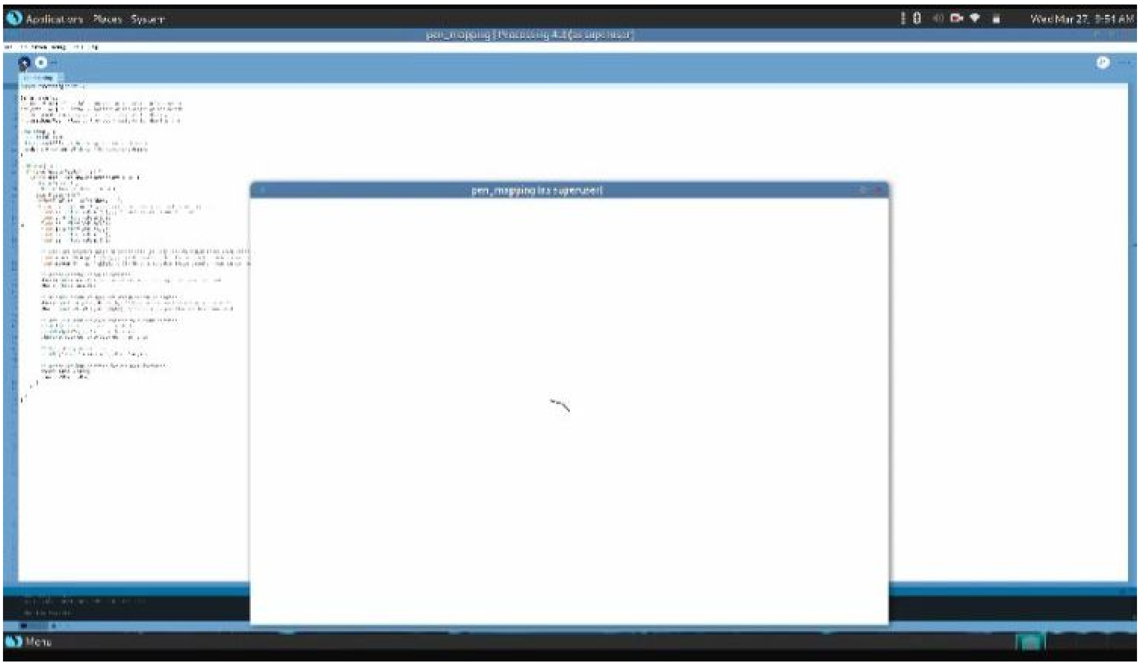

Figure 4.5: Screen no:2

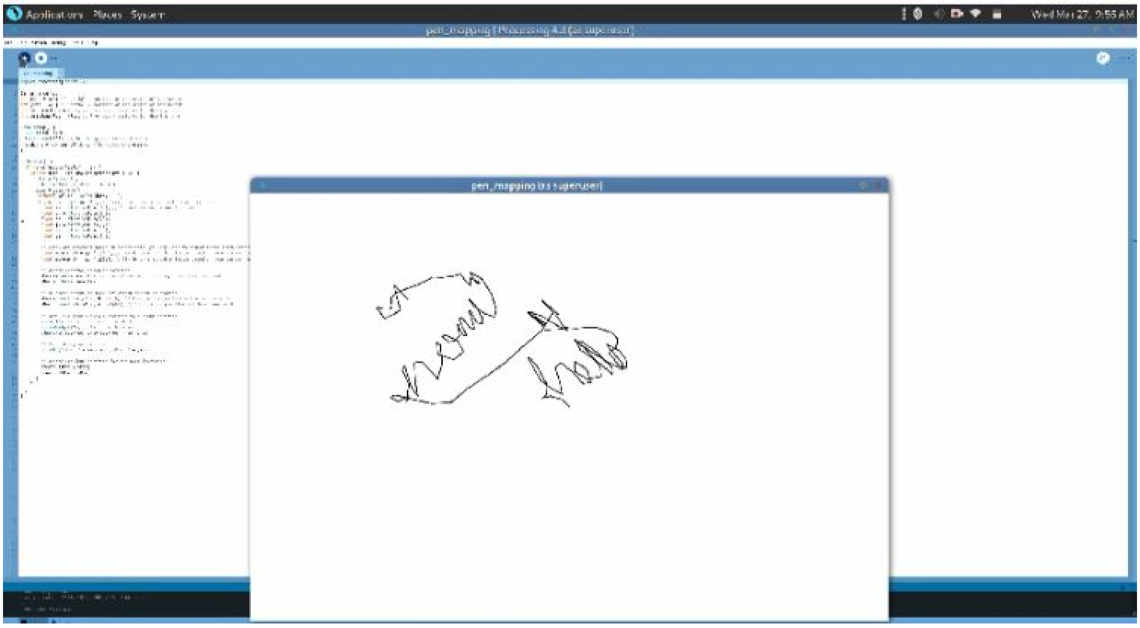

Figure 4.6: Screen no:3

Chapter 5

Iteration 1

5.0.1 Midterm FYP 1

The goal of the Midterm FYP 1 phase is to establish a solid foundation for the smart pen project, ensuring that all critical components are ready for successful execution. This phase includes thorough research to define the project's overall direction, selecting the most appropriate models, and identifying the most effective approaches to meet the project's objectives. Additionally, the initial structure of the project will be created, setting a clear framework to guide the following stages of development and integration of the digital pen.

5.1 Hardware Setup and Functional Modules

To ensure the success of the smart pen project, the hardware was designed and configured into distinct functional modules, each focusing on specific tasks required for motion tracking, data capture, and system integration. The following subsections detail the individual components and their purpose within the system.

5.1.1 MPU6050 Sensor Integration

The MPU6050 sensor combines an accelerometer and a gyroscope to capture motion data, providing the foundation for pen movement tracking.

- **Accelerometer:** Measures acceleration along the X, Y, and Z axes to detect the direction and intensity of movement.
- **Gyroscope:** Measures angular velocity to detect rotations of the pen.
- **I2C Communication:** The sensor communicates with the Arduino Uno via the I2C protocol, ensuring efficient data transmission.
- **Calibration:** Initial calibration was performed to minimize sensor drift and ensure accurate readings.

5.1.2 Arduino Uno and SD Card Module

The Arduino Uno serves as the core microcontroller for data acquisition, with an SD card module attached for local data storage.

- **Data Processing:** Reads motion data from the MPU6050 sensor and performs basic filtering to reduce noise.
- **SD Card Module:** The SD card module is connected to the Arduino Uno to log motion data for later processing. This enables offline storage and retrieval of pen movement data.
- **Serial Communication:** Transfers processed data to the Processing IDE via a serial connection for real-time sketching.

5.1.3 Battery, Power Booster, and Charging Module

To ensure portability and uninterrupted operation, a battery system with a power booster and charging module was incorporated.

- **Battery Pack:** A 3.7V rechargeable battery was used to power the system.
- **Power Booster:** Boosts the battery output to 5V to meet the power requirements of the Arduino Uno and attached modules.
- **Charging Module:** Allows convenient recharging of the battery without disassembly.

5.1.4 LED Indicator Module

To provide feedback during operation, an LED module was integrated:

- **Power Indicator:** Displays whether the pen is powered on.
- **Data Transmission Indicator:** Flashes during data capture and transmission.
- **Error Feedback:** Signals if there are any issues in communication or sensor functionality.

5.2 Software Modules

The coding phase involves the development of core software modules to handle real-time data processing and user interaction.

5.2.1 Arduino Communication and Data Acquisition Module

This module focuses on establishing robust communication between the Arduino Uno and the MPU6050 sensor. The Arduino code is written to:

- Continuously read accelerometer and gyroscope data.
- Filter noise using a low-pass filter for smoother motion tracking.
- Log real-time data to the SD card for offline processing.
- Transmit real-time data via serial communication for sketching.

5.2.2 Real-Time Processing and Sketching Module

Using Processing IDE, this module visualizes the pen's movements in real time. It includes:

- **Canvas Rendering:** Displays motion data as visual sketches.
- **Motion Mapping:** Translates accelerometer and gyroscope readings to x, y positions on the canvas.
- **Drawing Modes:** Includes options for freehand sketching or predefined patterns.

5.2.3 Sketch Saving and Export Module

This module enables the user to save their work as a digital file:

- Converts canvas data into PDF format.
- Ensures high-quality rendering of saved sketches for documentation and sharing.

5.2.4 Integration and Workflow Management Module

This module ensures smooth data flow between the hardware and software components:

- Manages data transfer from Arduino to Processing.
- Handles synchronization between user inputs and visual output.
- Provides error handling to detect and resolve any data transmission issues.

5.3 Limitations and Areas for Improvement

5.3.1 Hardware Challenges

- **Sensor Noise:** The MPU6050 introduces substantial noise, affecting accuracy.

- **Writing Misalignment:** The output is not perfectly aligned with the user's hand movement due to sensor limitations.
- **Limited Processing Power:** The Arduino Uno's capabilities limit complex real-time processing.
- **Power Efficiency:** Despite the power booster, battery life could be improved.

5.3.2 Future Enhancements

To address these challenges, future iterations could consider:

- **Sensor Upgrade:** Switching to an optical sensor (e.g., KA8) to enhance accuracy and eliminate noise.
- **Advanced Microcontroller:** Upgrading to Teensy 4.1 for built-in SD card support, better processing power, and USB shield compatibility.
- **Improved Power Management:** Using a higher-capacity battery or incorporating a power optimization module for longer operation times.
- **Feedback Mechanism:** Adding haptic feedback to enhance user interaction.
- **Modular Design:** Simplifying the integration of components for easier assembly and maintenance.

5.4 Coding

The coding phase is divided into key modules that are essential for the project's functionality and integration. These modules ensure real-time data collection, interaction between hardware and software, and the ability to generate and store sketches based on user input.

5.4.1 Arduino Communication and Data Acquisition Module

Listing 5.1: Arduino Code for MPU6050 Communication

```
#include <Wire.h>
#include <MPU6050.h>
#include <SD.h>

MPU6050 mpu;
int16_t ax, ay, az, gx, gy, gz;
const int chipSelect = 4;

void setup() {
  Serial.begin(9600);
  Wire.begin();
  mpu.initialize();

  if (!SD.begin(chipSelect)) {
    Serial.println("SD␣Card␣initialization␣failed!");
    while (1);
  }
}

void loop() {
  mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);

  File dataFile = SD.open("motion_data.txt", FILE_WRITE);
  if (dataFile) {
    dataFile.print(ax); dataFile.print(",");
    dataFile.print(ay); dataFile.print(",");
    dataFile.print(az); dataFile.print(",");
    dataFile.print(gx); dataFile.print(",");
```

```
      dataFile.print(gy);  dataFile.print(",");
      dataFile.println(gz);
      dataFile.close();
    } else {
      Serial.println("Error␣opening␣file");
    }

    delay(100);
}
```

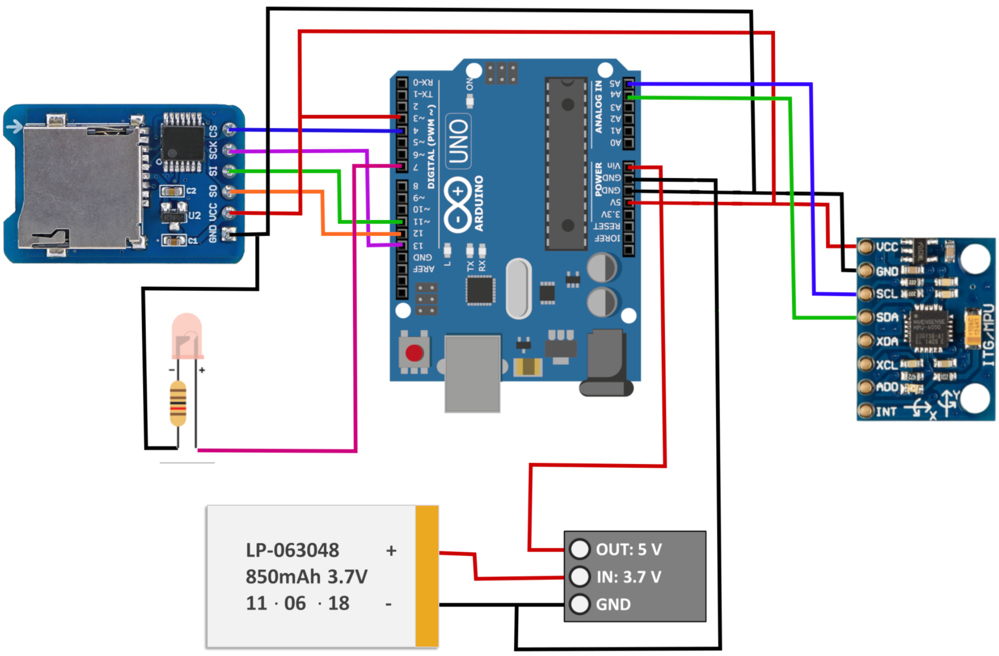

Figure 5.1: Architectural Circuit Diagram

Chapter 6

Iteration 2

6.1 FYP 1 Final

The Final phase of FYP 1 marks the transition from research to practical implementation of the smart pen project. Following extensive research, a roadmap was developed to guide the design and development stages. The initial phase involved constructing the pen's basic structure and focusing on core components. This phase emphasized sensor data acquisition, real-time data processing, and sketching functionalities to ensure the smart pen's major features were thoroughly tested.

6.2 Test Cases

A series of test cases were executed to ensure the smart pen's components were functioning correctly and reliably.

6.2.1 Data Acquisition and Sensor Module Test Case

Test cases were designed to verify the data acquisition process between the Arduino and the MPU6050 sensor. Key test scenarios included:

- Ensuring accurate capture of acceleration and rotational data across the X, Y, and Z axes.

- Verifying the Arduino's reliable transmission of sensor data via the serial connection.
- Testing the sensor's accuracy and responsiveness by simulating hand movements and comparing results with expectations.

6.2.2 Real-Time Processing and Sketching Test Case

Test cases for real-time data processing and sketching included:

- Verifying the Processing software's ability to convert sensor data into real-time canvas movements.
- Testing sketching fluidity by simulating various drawing motions.
- Ensuring minimal delay and distortion when mirroring the pen's movements on the canvas.

6.2.3 Sketch Saving and Export Test Case

Test cases for sketch saving and export involved:

- Ensuring sketches were correctly saved as PDF files.
- Verifying visual fidelity and integrity of exported sketches.
- Testing export functionality in multiple file formats for compatibility and ease of sharing.

6.2.4 Integration and Workflow Management Test Case

Test cases for module integration included:

- Ensuring seamless data flow between Arduino, Processing, and PDF export functions.
- Verifying the system's error handling, such as managing sensor malfunctions or communication breakdowns.
- Testing overall system performance using complex sketch data.

6.3 Tools, Frameworks, and Models

Various tools, frameworks, and programming languages were used to ensure optimal performance and functionality of the smart pen.

6.3.1 Development Tools

- Arduino IDE: The main development environment for programming the Arduino and integrating sensors.
- Processing IDE: Used for visualizing real-time pen movements on the digital canvas.
- PDF Libraries: Utilized to export sketches in PDF format.

6.3.2 Languages

- C++: Used for programming the Arduino and processing sensor data.
- Java: Used in Processing IDE for data processing and rendering sketches.

6.3.3 Frameworks and Libraries

- Arduino Libraries: Libraries for interfacing with the MPU6050 sensor.
- Processing Libraries: Libraries for real-time interaction and graphical rendering.

- PDF Export Libraries: Libraries for exporting sketches in PDF format.

6.3.4 Additional Tools

- Git and GitHub: Used for version control and collaborative development.
- Google Colab: Utilized for testing and optimizing sensor data.

6.4 Code Snippet

Below is a code snippet showing the integration of the MPU6050 sensor with the Arduino and Processing IDE for real-time sketching.

6.4.1 Arduino Code

```
#include <Wire.h>
#include <MPU6050.h>

MPU6050 mpu;
int16_t ax, ay, az, gx, gy, gz;
int irValue = 0; // IR sensor value (1 = off, 0 = on)
float accelScaleX = 1, accelScaleY = 1, accelScaleZ = 1;
float gyroScaleX = 0.7, gyroScaleY = 0.5, gyroScaleZ = 1;

void setup() {
  Serial.begin(9600);
  Wire.begin();
  mpu.initialize();
}

void loop() {
  mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
  irValue = digitalRead(3); // IR sensor on pin 3

```

```
  float scaledAx = ax * accelScaleX;
  float scaledAy = ay * accelScaleY;
  float scaledAz = az * accelScaleZ;
  float scaledGx = gx * gyroScaleX;
  float scaledGy = gy * gyroScaleY;
  float scaledGz = gz * gyroScaleZ;

  Serial.print(scaledAx);
  Serial.print(",");
  Serial.print(scaledAy);
  Serial.print(",");
  Serial.print(scaledAz);
  Serial.print(",");
  Serial.print(scaledGx);
  Serial.print(",");
  Serial.print(scaledGy);
  Serial.print(",");
  Serial.print(scaledGz);
  Serial.print(",");
  Serial.println(irValue);

  delay(100);
}
```

Listing 6.1: Arduino Code for Data Acquisition and Communication

6.4.2 Processing IDE Code

```
import processing.serial.*;
import processing.pdf.*;

Serial arduino;
int xPos = 100, yPos = 200;
boolean shouldScroll = false;

ArrayList<Integer> xPositions = new ArrayList<Integer>();
ArrayList<Integer> yPositions = new ArrayList<Integer>();
ArrayList<Integer> isDrawing = new ArrayList<Integer>();

void setup() {
  size(1800, 1200);
  background(255);
  arduino = new Serial(this, "/dev/ttyACM0", 9600);
}

void draw() {
  if (arduino.available() > 0) {
    String data = arduino.readStringUntil('\n');
    if (data != null) {
      data = data.trim();
      String[] values = split(data, ',');
      if (values.length == 7) {
        float ax = float(values[0]);
        float ay = float(values[1]);
        float az = float(values[2]);
        float gx = float(values[3]);
        float gy = float(values[4]);
        float gz = float(values[5]);
        int ir = int(values[6]);

        float movementX = (gy + 1200) / 200;
        float movementY = (gx - 1100) / 250;

```

```
        xPos += int(movementX);
        yPos += int(-movementY);

        xPos = constrain(xPos, 0, width - 20);
        yPos = constrain(yPos, 0, height - 20);

        if (ir == 1) {
          xPositions.add(xPos);
          yPositions.add(yPos);
          isDrawing.add(1);
        }

        if (xPos > 1500 && yPos > 900) {
          shouldScroll = true;
        }

        if (shouldScroll) {
          background(255);
          xPos = 100;
          yPos = 200;
          xPositions.clear();
          yPositions.clear();
          isDrawing.clear();
          shouldScroll = false;
        }
      }
    }
  }

  for (int i = 0; i < xPositions.size(); i++) {
    int x = xPositions.get(i);
    int y = yPositions.get(i);
    int drawing = isDrawing.get(i);

    if (drawing == 1) {
      fill(0);
      ellipse(x, y, 6, 6);
```

```
    } else {
      if (i > 0) {
        int prevX = xPositions.get(i - 1);
        int prevY = yPositions.get(i - 1);
        stroke(0);
        strokeWeight(4);
        line(prevX, prevY, x, y);
      }
    }
  }
}

void keyPressed() {
  if (key == 's' || key == 'S') {
    PGraphicsPDF pdf = (PGraphicsPDF) createGraphics(
    width, height, PDF,
    "/home/nomi/gabbar_"
    + "is_back.pdf");
    pdf.beginDraw();
    pdf.background(255);
    for (int i = 0; i < xPositions.size(); i++) {
      int x = xPositions.get(i);
      int y = yPositions.get(i);
      int drawing = isDrawing.get(i);
      if (drawing == 1) {
        pdf.fill(0);
        pdf.ellipse(x, y, 6, 6);
      } else {
        if (i > 0) {
          int prevX = xPositions.get(i - 1);
          int prevY = yPositions.get(i - 1);
          pdf.stroke(0);
          pdf.strokeWeight(4);
          pdf.line(prevX, prevY, x, y);
        }
      }
    }
```

```
    pdf.endDraw();
    pdf.dispose();
  }
}
```

Listing 6.2: Processing Code for Real-Time Sketching and Export

Chapter 7

Iteration 2

7.1 FYP 2 Midterm

7.1.1 Challenges in FYP 1

During FYP 1, the initial implementation faced significant challenges:

- **Accuracy Issues**: The MPU6050 sensor struggled to provide precise readings, resulting in a lack of fidelity between human hand movement and pen output.

- **Excessive Noise**: The sensor introduced substantial noise, causing distortions and jittery output during sketching.

- **Misalignment of Writing**: Due to sensor limitations, the output failed to align with the user's actual hand movement accurately.

- **Hardware Limitations**: The Arduino Uno lacked advanced features such as built-in SD card support and compatibility with USB shields, leading to inefficiencies in data handling.

7.1.2 Change in Approach

To overcome these limitations, we adopted a new approach in Iteration 2, featuring two major changes:

- **Optical Sensor (KA8)**: The MPU6050 sensor was replaced with an optical sensor (KA8) paired with an LED light. The optical sensor detects reflected light from the writing surface, capturing precise x, y coordinates of the pen's movement.
- **Teensy 4.1 Microcontroller**: The Arduino Uno was replaced with a Teensy 4.1 microcontroller due to its built-in SD card slot, support for USB shields, and superior processing capabilities.

The new workflow involves the optical sensor capturing pen movement data, which is then processed and stored as a **.txt file** on the SD card integrated with the Teensy 4.1.

7.1.3 Reasons for Changing the Approach

The decision to shift technologies was motivated by the following reasons:

- To achieve higher accuracy by eliminating noise and distortion in sensor data.
- To improve the alignment of writing, ensuring a more natural and intuitive user experience.
- To simplify and streamline data storage workflows with the built-in SD card support in the Teensy 4.1.
- To enable better scalability and compatibility through the use of USB shields and enhanced hardware capabilities.

7.1.4 Benefits of the New Approach

The updated approach provided several tangible benefits:

- **Precise Alignment**: The optical sensor ensures the pen's output closely mirrors the user's hand movements.
- **Minimal Noise**: The replacement of the MPU6050 sensor with the optical sensor eliminates noise, resulting in cleaner and more accurate writing or sketches.
- **Efficient Data Handling**: The Teensy 4.1 simplifies data processing and storage, reducing latency and hardware complexity.
- **Enhanced Usability**: The system's improved accuracy and alignment make it more user-friendly and suitable for creative applications.

7.1.5 Backend System: Optical Data Pipeline

The new system features a backend pipeline optimized for real-time optical data processing. This pipeline consists of the following components:

- **Tier 1: Optical Sensor and LED Module**
  - The LED module illuminates the writing surface.
  - The optical sensor (KA8) captures reflected light, translating movement into precise x, y coordinates.
- **Tier 2: Teensy 4.1 Processing and Storage**
  - The Teensy 4.1 processes the captured x, y coordinates.
  - Data is stored in **.txt format** on the SD card integrated with the Teensy.
- **Tier 3: Data Visualization and Analysis**
  - The stored data can be retrieved for visualization using tools like **Streamlit**.
  - The visualization displays the recorded pen movements, enabling users to review or edit their sketches digitally.

7.2 Conclusion

The transition to an optical sensor and Teensy 4.1 represents a significant improvement in the smart pen's development. By addressing the challenges faced in FYP 1, the new system delivers superior accuracy, alignment, and efficiency. The backend system ensures seamless data processing and storage, paving the way for future enhancements and expanded functionality.

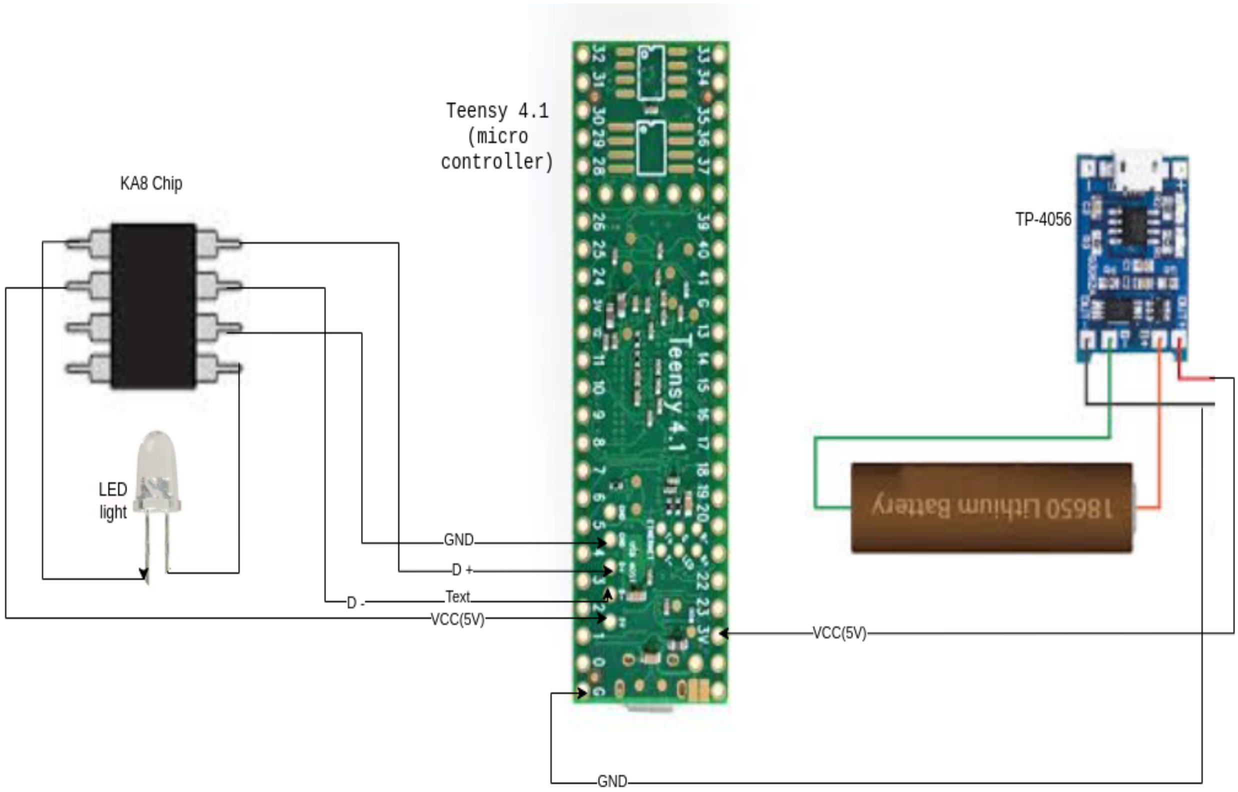

Figure 7.1: Circuit Daigram

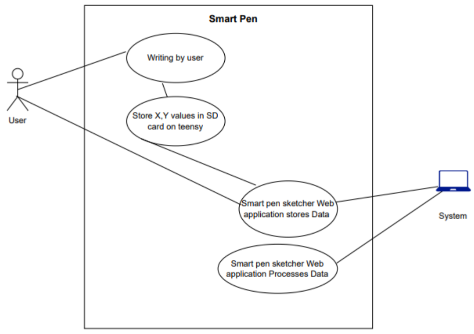

Figure 7.2: Use Case Daigram

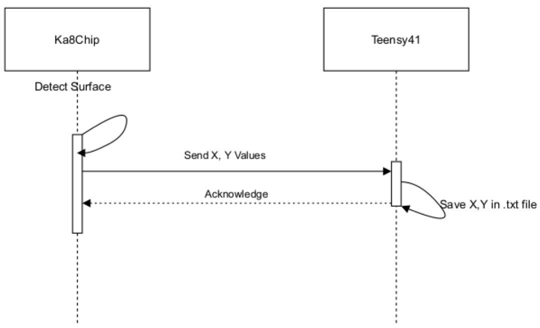

Figure 7.3: Sequence Daigram

Chapter 8

Iteration 4

8.1 FYP 2 Final

8.1.1 Smart Pen Sketcher - Final Implementation

In this phase, we focused on integrating the backend functionality with the web interface for our Streamlit-based "Smart Pen Sketcher" application. This application allows users to upload files containing coordinates (x, y, and flags) and generate sketches by connecting the points. The integration enables a smooth and interactive user experience, where the backend processes the coordinates and generates plots, and the web interface displays the results in real-time.

The core logic involves plotting coordinates on a customizable page layout, with options to adjust plot settings like dot color and size. Additionally, users can download the resulting sketches as PDF files. This integration represents a major step in streamlining the user's workflow for generating sketches and creating PDF reports.

8.1.2 Code Snippet

```
import streamlit as st
import pandas as pd
import matplotlib.pyplot as plt
```

```
import io
from matplotlib.backends.backend_pdf import PdfPages
from datetime import datetime

# Hide the default Streamlit header and set padding
st.markdown(
    """
    <style>
    .stApp > header {display: none;}
    .stApp > div {padding-top: 0;}
    </style>
    """,
    unsafe_allow_html=True
)

# Function to read coordinates from a file
def read_coordinates(file):
    try:
        coordinates = pd.read_csv(file, header=None)
        if coordinates.shape[1] != 3:
            st.error("Uploaded file must contain exactly 3 columns
(x, y coordinates, and a flag).")
            return None
        return coordinates
    except Exception as e:
        st.error(f"Error reading the file: {e}")
        return None

# Function to create plots based on coordinates
def create_plots(coordinates, dot_color, dot_size, page_width,
page_height):
    figures = []
    fig, ax = plt.subplots(figsize=(page_width, page_height))
    ax.set_xlim(0, page_width * 100)
    ax.set_ylim(0, page_height * 100)
    ax.axis('off')  # Hide axes
    ax.invert_yaxis()  # Invert the y-axis
```

```

    last_x, last_y = 0, 0  # Start from the top-left corner

    for i in range(len(coordinates)):
        x, y, flag = coordinates.iloc[i]

        # Only draw if the flag is 1
        if flag == 1:
            # Draw line from the last point to the current point
            ax.plot([last_x, x], [last_y, y], color=dot_color, linewidth=dot_size)
            last_x, last_y = x, y
        else:
            # Reset last_x and last_y if the flag is not 1
            last_x, last_y = x, y  # This will skip drawing

    # Append and close the last figure only if it has drawn lines
    figures.append(fig) if last_x != 0 or last_y != 0 else plt.close(fig)  # Close if empty

    return figures
```

Listing 8.1: Code for reading coordinates and creating plots

This code snippet defines a function to read coordinates from a CSV file and then plots the coordinates on a page layout. If the flag in the data is set to 1, it connects the points with lines; otherwise, it skips plotting that point. The plot's appearance, such as dot color and size, as well as the page size, can be customized by the user. Additionally, the plot can be exported as a PDF.

8.1.3 Code Snippet

```
# Streamlit app
st.title("Smart Pen Sketcher")

# File Upload section
st.sidebar.header("File Upload Options")
```

```
uploaded_file = st.sidebar.file_uploader("Choose a file with x,y, flag coordinates", type=["csv", "txt"])

coordinates_data = None

if uploaded_file is not None:
    coordinates_data = read_coordinates(uploaded_file)

# Settings Section
if coordinates_data is not None:
    st.sidebar.header("Settings")
    with st.sidebar.expander("Plot Settings", expanded=True):
        dot_color = st.color_picker("Choose Dot Color", "#010b13")
        dot_size = st.slider("Choose Dot Size", min_value=1, max_value=20, value=2)

        st.subheader("Choose Page Size")
        page_size_option = st.selectbox("Select Page Size", options=["A4", "Letter", "Custom"])

        standard_sizes = {
            "A4": (8.268, 11.693),
            "Letter": (8.5, 11)
        }

        if page_size_option in standard_sizes:
            page_width, page_height = standard_sizes[page_size_option]
        else:
            page_width = st.number_input("Page Width (inches)", value=8.268, min_value=0.1)
            page_height = st.number_input("Page Height (inches)", value=11.693, min_value=0.1)

    # Create plot and display it with a spinner
    with st.spinner("Creating plot... Please wait."):
        figures = create_plots(coordinates_data, dot_color,
```

```
dot_size, page_width, page_height)

        for fig in figures:
            st.pyplot(fig)

        # Automatically generate the PDF when the figures are ready
        pdf_buffer = io.BytesIO()
        with PdfPages(pdf_buffer) as pdf:
            for fig in figures:
                pdf.savefig(fig)
                plt.close(fig)  # Close each figure after saving to PDF
        pdf_buffer.seek(0)  # Reset buffer position to the beginning

        # Initialize session state for PDF name if not already set
        if 'pdf_name' not in st.session_state:
            st.session_state.pdf_name = f"mno-{datetime.now().strftime('%Y%m%d_%H%M%S')}"

        # Editable PDF name input
        pdf_name_input = st.sidebar.text_input("PDF File Name", st.session_state.pdf_name, key="pdf_name_input")

        # Update PDF name in session state on change
        if pdf_name_input:
            st.session_state.pdf_name = pdf_name_input.strip()

        # Store the PDF buffer in session state
        st.session_state.pdf_buffer = pdf_buffer

        # Download button for the generated PDF
        if st.session_state.pdf_buffer is not None:
            pdf_name_final = st.session_state.pdf_name.strip() or f"mno-{datetime.now().strftime('%Y%m%d_%H%M%S')}"
            st.sidebar.download_button(
                label="Download PDF",
```

```
                data=st.session_state.pdf_buffer,
                file_name=f"{pdf_name_final}.pdf",
                mime="application/pdf"
            )
```

Listing 8.2: Streamlit app for file upload and settings

This code defines a simple Streamlit app that enables the user to upload a file with x, y coordinates and a flag. It also allows customization of plot settings, including dot color, size, and page layout. Once the plot is generated, it can be downloaded as a PDF file.

8.2 Images

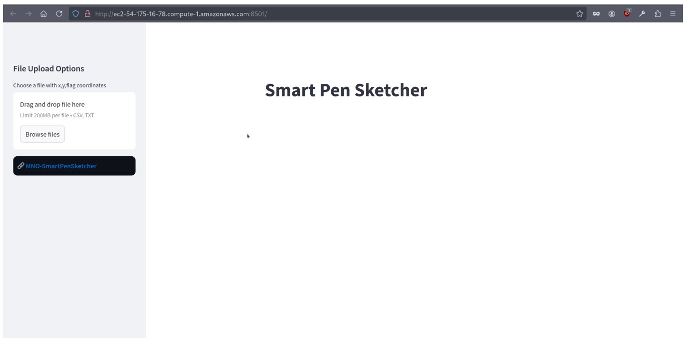

Figure 8.1: The main page of the web interface that appears when you open the application.

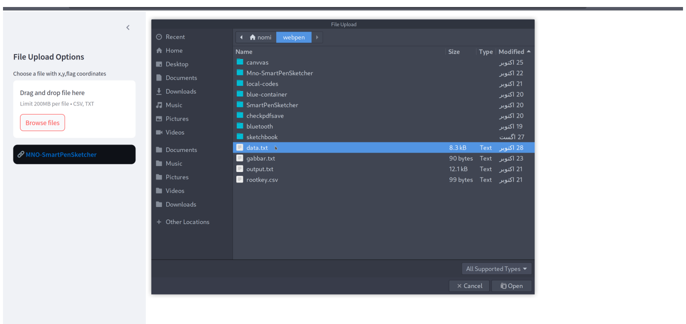

Figure 8.2: The screen shown after clicking on the "Upload File" button to upload a coordinate file.

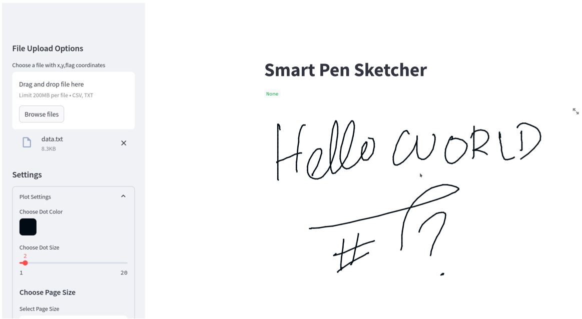

Figure 8.3: The view displaying the coordinates mapped and sketched from the chosen file, showing the generated plot.

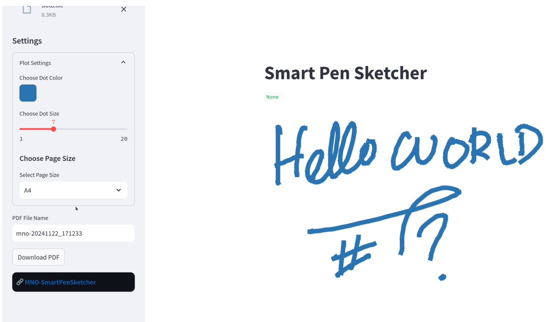

Figure 8.4: The interface where you can adjust the pen size and color, with the updated results shown on the screen.

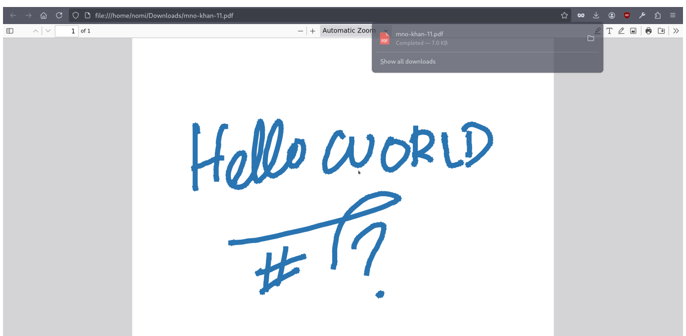

Figure 8.5: The saved PDF (with a user defined name) opened in the browser, showing the generated plot based on the coordinates from the uploaded file.

Chapter 9

Implementation Details

9.1 Technologies Used

9.1.1 Frontend

The frontend of the Smart Pen Sketcher application is developed using the **Streamlit** framework. This framework provides a simple and powerful interface to create interactive web applications. Through Streamlit, users can upload files containing pen movement data (in CSV or TXT format), adjust plot settings (e.g., dot size, color, page size), and visualize the generated sketches in real-time. The frontend also allows users to download the resulting sketch as a PDF file, providing a seamless and intuitive user experience.

9.1.2 Backend

The backend is implemented in **Python**, leveraging various libraries to handle the data processing and coordinate visualization. Core libraries used include:

- **Matplotlib**: for plotting coordinates and drawing sketches.
- **Pandas**: for managing and processing CSV data (coordinates and flags).
- **Streamlit**: not only serves the frontend but also handles data flow, file uploads, and

PDF generation.

The backend's core functionality involves:

- Reading coordinate data from uploaded files.
- Generating visualizations of pen movements based on x, y coordinates and flags.
- Storing the processed data in memory before generating and serving PDFs.

The backend is tightly integrated with the frontend, ensuring real-time data processing and visualization.

9.2 System Architecture

The system architecture is designed to efficiently manage the flow of data from the user's pen movements to the final output, a downloadable PDF containing the generated sketch.

9.2.1 Client-Side Application

The **Streamlit** application serves as the primary interface through which users interact with the system. Users upload CSV or TXT files containing coordinates and flags representing pen movements. The application enables customization of the sketch by providing options such as dot size, color, and page layout (e.g., A4, Letter, or custom size). Once the user adjusts the settings and uploads their data, the frontend communicates with the backend to process the file and generate the corresponding sketch.

9.2.1.1 File Upload and Data Validation

Upon file upload, the system checks the integrity of the data, ensuring it contains the correct number of columns (x, y, and a flag column). In the case of invalid or incomplete data, the system provides error messages to help the user correct their input before proceeding.

9.2.2 Backend System

The backend processes the uploaded coordinates and generates visualizations based on the given settings. It performs the following tasks:

- **Reading Coordinates**: The uploaded file is read and parsed into a Pandas DataFrame for easy manipulation and processing.
- **Plot Generation**: The coordinates are plotted on a defined page layout (A4, Letter, or custom) using **Matplotlib**. The plot generation includes connecting points where the flag equals 1, representing active pen movements.
- **PDF Generation**: After plotting, the sketch is saved as a PDF file. The **Matplotlib** library integrates with **PdfPages** to allow multiple pages to be generated and saved into a single PDF file.

The backend operates seamlessly with the frontend, ensuring that the user receives instant feedback as their data is processed.

9.2.3 Data Flow

The data flow can be broken down into several distinct stages:

1. **User Upload**: The user uploads a file containing x, y coordinates and flags (indicating whether to draw or skip a point).
2. **Backend Processing**: The backend reads the file, validates it, and processes the coordinates to generate the plot.
3. **Real-time Visualization**: The plot is displayed in the frontend as the user adjusts settings (e.g., dot color, dot size, page size).
4. **PDF Generation**: Once the user is satisfied with the plot, the backend generates a PDF, which can be downloaded directly from the web interface.

9.2.4 Modular Architecture

The system architecture follows a modular approach, allowing for easy updates and scalability. The following modules are defined:

- **File Handling Module**: Responsible for managing file uploads, reading data, and validating input.
- **Plotting Module**: Handles the creation of visualizations based on the data received from the file.
- **PDF Generation Module**: Responsible for creating downloadable PDFs from the generated plots.
- **Frontend Module**: Interacts with the user, presenting options and displaying visual feedback, and integrates with the backend.

Each module can be independently modified or enhanced, providing flexibility for future iterations or improvements.

9.3 Error Handling and Validation

A key aspect of the system is ensuring that the user inputs valid data. Several error handling and validation mechanisms are in place:

- **File Validation**: The system checks that the uploaded file contains three columns (x, y, and a flag column). If the file format is incorrect, the user is notified.
- **Coordinate Validation**: Each coordinate is validated to ensure it lies within the expected range for the page layout. Invalid coordinates are flagged, and users are prompted to correct them.
- **User Feedback**: If an error occurs, such as a missing column or invalid input, the system provides clear and actionable error messages to guide the user through correcting the issue.

These measures ensure that users have a smooth experience with minimal disruptions.

9.4 Security Considerations

While the system is primarily a local tool for generating sketches, security considerations still apply to the uploaded data:

- **File Security**: The system ensures that uploaded files are not stored long-term and are processed in memory to minimize the risk of file-based security threats.
- **Session Management**: Although not using full user authentication, the system ensures that all data processing is done within the user's session to avoid unauthorized access to the uploaded files and generated sketches.

9.5 User Experience (UX) Considerations

The **Smart Pen Sketcher** prioritizes user experience with an intuitive and responsive interface. Key UX considerations include:

- **File Upload Interface**: The system provides an easy-to-use file upload option with clear instructions on what file formats are supported.
- **Interactive Settings**: Users can interactively change the plot settings, including dot color, size, and page layout, with real-time visual feedback.
- **Error Messages**: The system provides clear, user-friendly error messages, ensuring users understand what went wrong and how to resolve it.

The design is intended to make the sketching process straightforward, reducing the learning curve for users and enhancing overall usability.

9.6 Version Control

To manage the development of the application, **Git** and **GitHub** were used for version control. This allowed the team to:

- Track changes and maintain a history of the codebase.
- Collaborate efficiently on different features of the project.
- Ensure that all code updates were properly integrated and tested before deployment.

The use of version control ensured a structured and reliable development process.

9.7 Conclusion

The implementation of the **Smart Pen Sketcher** has successfully addressed the challenges encountered in previous iterations, providing a robust and user-friendly application for generating and visualizing sketches. By integrating a real-time plotting system, data validation, and PDF export functionality, the system delivers a seamless experience for users looking to digitize their pen-based drawings.

Chapter 10

Test Cases

Test Case 1: Valid Coordinate Input

Objective: Ensure that the system correctly processes and visualizes valid coordinate data from the KA8 sensor.
Method: Upload a CSV or TXT file containing valid x, y coordinates and flags from the KA8 sensor into the Smart Pen Sketcher.
Expected Outcome: The system processes the file, generates a valid plot, and displays the pen movements accurately on the user interface.

Test Case 2: Invalid Coordinate Data Handling

Objective: Verify that the system correctly handles invalid coordinate data.
Method: Upload a file with missing or corrupted x, y coordinates and flags (e.g., non-numeric values or incomplete rows).
Expected Outcome: The system returns an error message indicating the data is invalid and prompts the user to upload a valid file.

Test Case 3: Custom Page Size Adjustment

Objective: Ensure the system can handle custom page sizes (e.g., custom width and height) for visualizing sketches.
Method: Adjust the page size settings in the Smart Pen Sketcher and upload a valid coordinate file.
Expected Outcome: The system successfully adjusts the sketch visualization according to the user-defined page size, and the sketch remains properly scaled and positioned.

Test Case 4: Plot Customization with Dot Size and Color

Objective: Test the ability to customize plot visuals, such as dot size and color.
Method: Upload a valid coordinate file and adjust the dot size and color settings in the Smart Pen Sketcher.
Expected Outcome: The system updates the plot to reflect the customized dot size and color, displaying the pen movements as requested by the user.

Test Case 5: Generation of PDF File from Plot

Objective: Verify that the system generates and downloads a PDF containing the visualized sketch.
Method: Upload a valid coordinate file, adjust settings, and request the download of the generated plot as a PDF.
Expected Outcome: The system generates a PDF containing the plot and provides a download link for the user to save the file.

Test Case 6: Performance Under Large Data Load

Objective: Test the system's performance and stability when handling a large volume of coordinate data.
Method: Upload a CSV file containing thousands of coordinates and request the visualization and PDF generation.
Expected Outcome: The system processes the file and generates the plot within an acceptable time frame, without crashing or experiencing significant delays.

Test Case 7: Multi-Page PDF Generation

Objective: Ensure that the system correctly handles the creation of multi-page PDFs when the data exceeds one page size.
Method: Upload a large file with sufficient coordinates to span multiple pages. Generate and download the resulting PDF.
Expected Outcome: The system generates a multi-page PDF, with each page containing a portion of the plot based on the defined page size.

Test Case 8: Coordinate Range Validation

Objective: Verify that the system correctly handles coordinates outside the expected range for the plot.
Method: Upload a coordinate file containing x, y values that exceed the defined boundaries of the page layout.
Expected Outcome: The system either adjusts the coordinates to fit within the valid range or prompts the user with a warning message about the invalid coordinates.

Test Case 9: Missing Flag Values Handling

Objective: Ensure that the system handles missing flag values (used to indicate whether to draw a point or not).
Method: Upload a file with missing or empty flag values and request the plot generation.
Expected Outcome: The system either assigns a default flag value or provides an error message indicating that the flag column is incomplete.

Test Case 10: User Interface Responsiveness

Objective: Verify the responsiveness of the Smart Pen Sketcher interface when interacting with large files or multiple settings adjustments.
Method: Upload a large coordinate file, adjust various plot settings (e.g., dot size, color), and observe the system's responsiveness.
Expected Outcome: The interface remains responsive, with settings adjustments reflected in real-time, and no significant lag during the plotting process.

Test Case 11: Invalid File Format Handling

Objective: Test the system's ability to handle unsupported file formats (e.g., non-CSV or non-TXT files).
Method: Upload a file with an unsupported format (e.g., PNG, JPG) and request the system to process it.
Expected Outcome: The system returns an error message indicating that the file format is unsupported and prompts the user to upload a valid CSV or TXT file.

Test Case 12: Real-Time Plot Update during File Upload

Objective: Ensure that the system updates the plot in real-time as the file is uploaded and processed.
Method: Upload a small coordinate file and observe the plot updating dynamically as the file is processed.
Expected Outcome: The plot is updated as soon as data is uploaded, showing the pen movements in real-time as the file is being read.

Test Case 13: Error Recovery for File Upload Failures

Objective: Verify that the system recovers gracefully from file upload errors (e.g., file corruption or timeout).
Method: Simulate a failed file upload (e.g., corrupted file or network timeout).
Expected Outcome: The system displays an error message informing the user of the failure and allows them to retry the upload.

Test Case 14: Handling of Extreme Coordinate Values

Objective: Test how the system handles extreme coordinate values (e.g., very large or very small numbers).
Method: Upload a file containing extreme x, y values and request the system to generate a plot.
Expected Outcome: The system either adjusts the coordinates to fit within the valid plot range or prompts the user to modify the coordinate values to acceptable ranges.

Chapter 11

Conclusions and Future Work

11.1 Goals Achieved

11.1.0.1 Accurate Coordinate Capture and Visualization

Our project has successfully developed a system that captures and visualizes coordinates in real-time using the KA8 sensor and the Smart Pen Sketcher. The system is able to accurately track pen movements and plot corresponding coordinates, ensuring precision in representing handwritten sketches. This allows for efficient data capture and immediate visual feedback, which can be useful in various applications such as design, art, and education.

11.1.0.2 Customizable User Experience

The Smart Pen Sketcher provides a customizable platform for users to adjust various parameters, such as dot size, color, and page size. This flexibility enables users to personalize their sketching experience and generate visualizations that suit their specific needs, whether for professional design or casual use. Additionally, the PDF generation feature allows users to easily export and share their sketches.

11.1.0.3 Real-Time Interaction with Data

By integrating the KA8 sensor with the Smart Pen Sketcher, the system enables real-time interaction with coordinate data as the user writes. This dynamic feature offers immediate feedback and allows users to adjust their writing and sketching techniques as needed. The system's ability to process large volumes of data and generate high-quality visualizations in real time further enhances the user experience.

11.2 Future Work

11.2.0.1 Integration with Advanced Gesture Recognition

To further enhance the user experience, future work could focus on integrating advanced gesture recognition capabilities. By adding the ability to recognize specific pen movements or gestures (such as drawing shapes or symbols), the system could allow for more sophisticated input, enabling users to sketch complex shapes or trigger specific actions with custom gestures.

11.2.0.2 Cloud Integration for Data Storage and Sharing

Future versions of the Smart Pen Sketcher could integrate with cloud storage platforms to allow users to store their sketches and coordinates online. This would not only ensure data backup but also enable easy sharing of sketches between devices, making it accessible from anywhere. Cloud integration would also facilitate collaboration among multiple users on the same sketch or design.

11.2.0.3 Enhanced Plotting Features and 3D Visualization

An exciting direction for future development is the integration of 3D plotting capabilities. Currently, the system supports basic 2D plotting of coordinates, but incorporating 3D models could open up new possibilities for applications in fields like architecture, engineering, and gaming. Additionally, the ability to customize the plotting style, such as adding textures or shadows, could provide users with more advanced visualization tools.

11.2.0.4 Support for Multimodal Inputs

To increase the versatility of the system, future work could explore incorporating multimodal inputs. This might include voice commands, where users could issue verbal instructions to adjust settings or navigate through the Smart Pen Sketcher. This would enhance the hands-free interaction, making the system more accessible to users with different needs and preferences.

11.2.0.5 Integration with Augmented Reality (AR)

Integrating augmented reality (AR) into the Smart Pen Sketcher could provide users with a more immersive experience. By superimposing digital sketches onto physical surfaces, users could visualize their designs in real-world contexts. This could be particularly useful for artists, designers, or architects who need to visualize their ideas on actual locations or objects.

11.2.0.6 Refinement of Data Processing and Optimization

As the volume of data captured by the KA8 sensor increases, future improvements in data processing algorithms will be necessary to optimize performance. More efficient algorithms for coordinate processing could reduce processing time and allow the system to handle larger data sets with minimal latency. Additionally, improving the accuracy of coordinate alignment will enhance the overall precision of the plot generation, making it suitable for more professional applications.

11.2.0.7 User Feedback and Usability Testing

Future development could involve gathering user feedback and conducting extensive usability testing. This would allow the team to identify areas for improvement in terms of user interface design, system performance, and overall user satisfaction. Regular updates and iterations based on user feedback would help in refining the Smart Pen Sketcher and ensuring it meets the diverse needs of its users.

11.3 Security

- Data transmission between the pen, SD card, and digital devices must be encrypted to ensure privacy.
- Secure user authentication and authorization mechanisms shall be implemented for managing accounts.
- Regular security updates shall be applied to the pen's software to address any vulnerabilities.

11.4 Scalability

- The pen's system shall be designed for scalability to accommodate future software updates and integration with new digital platforms.
- The design should allow easy integration with additional digital tools and services.

11.5 User Authentication and Authorization

- Access to the pen's settings and data shall be restricted based on user authentication.
- Secure session management, including automatic logout after inactivity, shall be implemented.

Bibliography

[1] P. America, "The history of pen," https://pen.org/the-history-of-pen, 2023.

[2] L. Gershon, "Ballpoint pen csi," https://daily.jstor.org/ballpoint-pen-csi, 2023.

[3] M.-W. et al., "Digital pen technology for conducting cognitive assessments: A cross-over study," *Springer*, 2023.

[4] S. U. Rehman, S. A. Khan, I. U. Khan, H. Mustafa, and M. A. Shaikh, "Interactive smart writing technology (iswt): An assistance to paperless documentation," *International Journal of Smart Technologies*, vol. 12, no. 2, pp. 123–135, 2020.

[5] M. Wehbi, T. Hamann, J. Barth, and B. Eskofier, "Digitizing handwriting with a sensor pen: A writer-independent recognizer," in *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2020, pp. 295–300.

[6] R. Ganesan, S. S. Sardar, and M. Marimuthu, "Survey paper on technologies of automatic writing pen," in *AIP Conference Proceedings*, vol. 2790, no. 1. AIP Publishing, 2023.

[7] A. Heimann-Steinert, A. Latendorf, A. Prange, D. Sonntag, and U. Müller-Werdan, "Digital pen technology for conducting cognitive assessments: a cross-over study with older adults," *Psychological Research*, vol. 85, no. 8, pp. 3075–3083, 2021.

[8] C. M. Sluitman and S. Mohamaddan, "Arduino uno based handwriting recognition system using a feed-forward neural network," in *2024 9th International Conference on Mechatronics Engineering (ICOM)*, 2024, pp. 402–405.

[9] F. Lunardini, D. D. Febbo, M. Malavolti, M. Cid, M. Serra, L. Piccini, A. L. G. Pedrocchi, N. A. Borghese, and S. Ferrante, “A smart ink pen for the ecological assessment of age-related changes in writing and tremor features,” *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–13, 2021.

[10] D. D. Febbo, F. Lunardini, M. Malavolti, A. Pedrocchi, N. A. Borghese, and S. Ferrante, “Iot ink pen for ecological monitoring of daily life handwriting,” in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*, 2020, pp. 5749–5752.

[11] A. Agarwal and S. S. Belur, “Digipen: An intelligent pen using accelerometer for character recognition,” in *Emerging Research in Computing, Information, Communication and Applications*, N. R. Shetty, L. M. Patnaik, H. C. Nagaraj, P. N. Hamsavath, and N. Nalini, Eds. Singapore: Springer Singapore, 2019, pp. 161–170.

[12] T. T. Alemayoh, M. Shintani, J. H. Lee, and S. Okamoto, “Deep-learning-based character recognition from handwriting motion data captured using imu and force sensors,” *Sensors*, vol. 22, no. 20, 2022.

[13] Y. Zhang, M. Pahud, C. Holz, H. Xia, G. Laput, M. McGuffin, X. Tu, A. Mittereder, F. Su, W. Buxton, and K. Hinckley, “Sensing posture-aware pen+touch interaction on tablets,” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ser. CHI ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 1–14.

[14] M. Shintani, J. H. Lee, and S. Okamoto, “Digital pen for handwritten alphabet recognition,” in *2021 IEEE International Conference on Consumer Electronics (ICCE)*, 2021, pp. 1–4.

[15] B. Signer, “Pen-based interaction,” in *Handbook of Human Computer Interaction, Major Reference Work, Springer Nature, 2023*. Springer, 2023.

[16] A. Osman Hashi, S. Zaiton Mohd Hashim, and A. Bte Asamah, “A systematic review of hand gesture recognition: An update from 2018 to 2024,” *IEEE Access*, vol. 12, pp. 143 599–143 626, 2024.

[17] A. J. Eshtiwi and A. A. Al-Qerem, “Mr. pen,” 2023.

[18] W. Han, Y. Li, H. Yu, and J. Jia, “V-pen: An acoustic-based fine-grained virtual pen input system using hand tracking,” *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–11, 2024.

[19] J. Li, X. Liu, Z. Wang, T. Zhang, S. Qiu, H. Zhao, X. Zhou, H. Cai, R. Ni, and A. Cangelosi, “Real-time hand gesture tracking for human–computer interface based on multi-sensor data fusion,” *IEEE Sensors Journal*, vol. 21, no. 23, pp. 26 642–26 654, 2021.

[20] Y. Zhang, B. Liu, and Z. Liu, “Recognizing hand gestures with pressure-sensor-based motion sensing,” *IEEE Transactions on Biomedical Circuits and Systems*, vol. 13, no. 6, pp. 1425–1436, 2019.