Tejas: Your Personal AI Assistant for Seamless PC Control

Piyush Raj
BE-CSE
1201piyushraj@gmail.com

Priyanshu Kumar
BE-CSE
priyanshu345kumar@gmail.com

Nikhil Kumar Tiwari
BE-CSE
nikhiltiwariofficial2608@gmail.com

Divyanshu
BE-CSE
22bcs10238@cuchd.in

Harshal Jain
CSE, Chandigarh University
harshal.e18792@cumail.in

***Abstract*—Developed for smooth PC control through voice and text, Tejas AI is a sophisticated program integrated with advanced natural language processing that allows users to interact with their systems conversationally. Users can give commands and obtain system information with real-time feedback for better interaction and experience. Tejas AI performs several tasks and provides real-time updates as requested. Its complex architecture features a powerful execution engine that utilizes text-to-speech and speech recognition libraries for command processing and communication. The system's reliability and effectiveness are ensured through rigorous deployment methodologies, real- world scenario testing, unit testing, integration testing, and user acceptability testing. Users of the system are provided streamlined access to their computers, giving consumers greater AI control over their environments, thus proving the use of AI for routine tasks. This paper examines teh AI's methodology, system architecture, and practical applications to illustrate its significance in intelligent automation systems.**

***Index Terms*—Artificial Intelligence (AI) , Natural Language Processing (NLP) , Robotic process automation (RPA) , Memory-info Availability (Minfo.A) , Memory Availability (MA) ,Memory-info Total (Minfo.T) , Memory Total (MT) , Memory-info Used (Minfo.U) , Memory Used (MU)**

I. INTRODUCTION

In recent years, artificial intelligence (AI) has further automated processes, improved decision making, and increased the efficiency of systems in various sectors. The incorporation of AI into commonplace technologies is particularly noteworthy because it paves ways for the effortless engagement of humans and machines. With the continuous development of AI technologies, their application is ubiquitous, and AI is being integrated as personal assistants, allowing users to control and manage their devices much more efficiently. A notable example is the creation of AI-enabled assistants for personal computers (PCs), which provide an interactive platform to manage tasks, increase productivity, and enhance user experience.

The development of Robotic Process Automation (RPA) and Intelligent Automation (IA) have shown the potential of AI in automating simple tasks beyond routine, predefined processes. These developments have been facilitated by more advanced systems capable of performing higher-order thinking and dealing with unstructured environments, thereby augment- ing the functions of personal AI assistants (RPA Review, 2022) [] 1. In addition, advances in Machine Learning (ML), Natural Language Processing (NLP), and computer vision make it possible for AI systems to be fully autonomous and situationally aware (AI and Robotics, 2023) [] 2. This level of autonomy is essential for personal assistants to manage calendars, respond to requests, and perform automated tasks on personal computers.

By automating everyday tasks, improving resource management, and streamlining complex tasks, personal AI assistants for PC control can improve productivity. Personal AI assistants with embedded AI have the potential to enhance user experience, optimized resources, and better usability of the device for the user. When aiming to create a personal AI for PC control there are challenges that arise in the development, user take up, and corporate buy-in for integration into existing systems (Future of Work, 2023) [3]. This paper will discuss the development and potential of personal AI for seamless PC control, identifying key technologies needed, barriers to developing these systems, and anticipated opportunities to embed AI into computer control systems. While the growth of AI-powered personal assistants for PC control is an intellectuals leap in HCI as it combines automation and AI with user intelligence, it has potential to influence human interaction with technology, through improved productivity, reduced human input and promote seamless workflows. The objective of the research is to investigate the potential for personal AI assistants by examining their existing capabilities, challenges of developing personal AI assistants, and other future developments. Examining AI and PC control we can understand AI's transformative role into the future of computing.

There are several problems that impact existing systems: More and more possibilities are being explored to use AI technology to create AI systems for PC control in order to perform routine tasks and automate workflows, which could potentially lead to a more productive person using a PC (i.e., the kids are able to use virtual assistants to do their work in a way they do not have to think about). However, while

Identify applicable funding agency here. If none, delete this.

virtual assistants have been adopted widely in personal and business applications, existing AI assistants have considerable limitations. A primary limitation is that there is very little true context-aware understanding, and this can lead to suboptimal task execution based on incorrect command comprehension (e.g., robotic process automation, RPA). An additional limitation relates to rule-based automation systems that are highly rigid. Existing solutions are good at recognizing tasks and automating in repetitive and highly structured environments, however, when a task becomes dynamic and the user must engage cognitive decision-making and flexibility (e.g., intelligent automation), there is little existing systems can do to extend automation. Finally, as it relates to AI systems, existing agents with rule-based automation cannot process unstructured data or handle context-aware computing so that the agents create true intelligent, autonomous solutions that can deal with user requirements that are rich and complex (e.g., AI in robotics). This paper aims to improve upon the deficiencies listed, on behalf of the user, by proposing development of a personal AI assistant capable of controlling a PC. This artificial intelligence we envision would utilize advanced AI modalities like natural language processing (NLP), machine learning (ML) and real time decision making to allow easier communication ability between the user and PC through a better, user-focused interface. This solution would more effectively fulfill user intentions by not allowing due to misinterpretation commands to occur, and would allow for less robotic interactions with a PC.

II. LITERATURE REVIEW

Artificial intelligence (AI) and robotic process automation (RPA), with a focus on how these technologies might improve engineering and automated business workflows' decision-making (DM) capabilities. Intelligent automation (IA), which combines RPA, AI, and soft computing, improves decision-making, system reliability, and operational efficiency to revolutionize traditional DM. While AI uses machine learning and natural language processing to process unstructured data, RPA reduces operational mistakes and manages repetitive operations. The study outlines important research areas, benefits, implementation issues, and future applications to improve the discipline. It demonstrates the promise of IA for automated DM, fault diagnosis, and knowledge extraction in complicated environments[1].Artificial intelligence (AI) is incorporated into robotics, with a particular emphasis on the transition from automated to autonomous systems. It gives a general introduction to robotics and artificial intelligence, covering key ideas such as computer vision, machine learning, and decision-making. The study explores AI techniques including reinforcement learning and natural language processing while discussing different forms of automation, their advantages, and disadvantages. It also covers issues related to develop- ment, ethical considerations, and autonomy levels. A review of the current applications of artificial intelligence in Industry 4.0 looked at overall results and recommendations for future work, and provided case studies of AI augmented autonomous robots in the transportation and health care industries identifying challenges and opportunities for advancement. The report demonstrates that business is on a trajectory toward integration of artificial intelligence (AI) and automation at increasing rates, as implementation of AI and automation is disrupting the traditional way of carrying out business processes. Businesses are leveraging existing technologies in AI with automation processes to become more effective and efficient. The report indicates that AI and automation can provide a competitive advantage for organizations when both technologies are aligned in the organizations processes and the synergies released have the potential for organizations to achieve new degrees of operational performance and innovativeness in their respective industries. The review suggests that to be successful in organizations, presence and integration of AI and automation in business processes need to be embraced by organizations[3].The growing applications of automation beyond industry, such as in transportation, agriculture, and healthcare. Robotics is based on intelligence and adaptability under unstructured environments - robotics is primarily concerned with a system's sensors and actuators performing autonomously or semi-autonomously; automation depends on productivity, reliability, and efficiency under controlled environments. The author advocates for a second distinction: in automation research, quality is of paramount importance while in robotics research feasibility is of the most interest - demonstrating new capabilities. In this argument, potential linkages such as quality and feasibility are highlighted in both automation and robotics, and this article attempts to dismiss the notion that these two domains are entirely different[4]. This research study looks at the disruptive nature of automation and artificial intelligence (AI) technologies on various industries, such as manufacturing, healthcare, finance, logistics, transportation, and education. In assessing the benefits and consequences associated with the advent of these technologies, it discusses adoption rates and current trends. The article considers policy frameworks, ethical considerations, and the impacts for individuals and organizations. The article also offers predictions and future scenarios in the hope of revealing possible outcomes in an ever-changing environment. The findings made clear how critical it is for stakeholders to collaborate towards together to meet challenges and leverage opportunities brought about by AI and automation[5]. Automation and artificial intelligence (AI) are transforming organizations, enhancing productivity and personalization but posing ethical and legal questions. There is a range of consequences fields such as manufacturing, healthcare, and finance, which underscore stakeholder engagement. The literature cites improvements with AI and operational flexibility and decision-making, but underscored differences in the efficacy of speech recognition compared to normal hearing individuals and hearing impaired individuals. Nevertheless,

organizations that want to ensure a responsible transition to a more automated future must embrace hiring qualified personnel and developing ethical frameworks to capitalize on benefits and address challenges as technology continues to develop[6].This chapter will present the application of the tkinter package in python by demonstrating the construction a simple graphical user interface (GUI) for an industrial vision system. We will begin with a contextual overview of the various elements of a GUI as provided by tkinter, such as buttons, labels, and text fields, with brief descriptions of each. These elements will form the basis for the entire user experience. By the end of the chapter, readers will have the knowledge to use tkinter to create a GUI that achieves an effective and efficient experience for vision system use, improving the function and user interaction for industry applications[7].Although English is a dynamic language, this article addressed the need of a refreshed English lexicon for natural language processing (NLP) applications; the addition of new words and phrases is needed to enhance the dictionary in respect to accuracy for some NLP applications, such as part-of-speech tagging or machine translation. To make these enhancements, this study presented the idea of NLP++, in conjunction with distributed big data analytics platforms such as HPCC systems. NLP strategies and dictionary refinement studies are presented, demonstrating that use of NLP++ significantly improves the accuracy and completeness of the English lexicon [8]. Our focus in this study is to highlight the diversity of hardware and software in the Internet of Things (IoT), which limits innovation due vary- ing operating systems. We present Ambience, a new clean slate operating system designed in direct response to these issues, by adopting microservices as a common application programming model, and unifying resource scales. The use of ambientity also benefits IoT applications in that it enhances deployment portability and isolation. We describe the architec- ture and implementation evaluated with real-vs-ideal installa- tions and microbenchmarks. The results show that Ambience has comparable or better performance than other comparable operating systems across hardware platforms, from 64MHz microcontrollers to modern x86-64 servers [9]. Lastly we highlight the issues presented by the heterogeneous hardware and software of the IoT, that impede innovation because of different operating systems.We present Ambience, a new ”clean slate” operating system that leverages resource scales and supports microservices as a general application programming model to address these challenges. By emphasizing Internet of Things applications, Ambience enhances isolation, portability, and deployment configuration. We present an architecture and implementation layer of detail, assessment with micro benchmarks and from real-world installations. The results demonstrate that Ambience performs over a wide range of hardware resources, from 64MHz microcontrollers to modern x86-64 servers, with performance levels consistent with, or better than existing operating systems[10].

III. METHODOLOGY

The process for creating Tejas, the AI helper, is thoroughly explained in this part. It covers the instruments, technology, and procedures used to guarantee the assistant’s effective growth and peak performance. The project’s design, implementation, and testing phases are all covered by the steps that are listed.

A. Technology Infrastructure

The respective technology infrastructure was established with the explicit intention of establishing a powerful tech- nology infrastructure for, including enabling Artificial Intelli- gence, system integration and automation as part of providing Tejas AI support. Because of Python’s versatility and abun- dance of packages supporting both automation and artificial intelligence, it was selected as the programming language. The Speech Recognition package was used to allow our AI to recognize and act on commands in human oral expressions [6]. In addition, we used the Natural Language Toolkit (NLTK) to process natural language inputs so Tejas AI could recognize responses to voice and text commands [6]. I implemented a GUI (graphical user interface) using Tkinter for user input via text [7]. This allows a fairly decent user experience for individuals who may not be interested in or comfortable with voice commands. Python’s os and subprocess libraries gave us the ability to run operating system commands allowing Tejas AI to access the operating system, perform tasks like launching applications, read files, and monitor CPU, RAM and disk usage. Finally, we used Pyttsx3 to include text to speech functionality and feedback to the user. Additionally, we also employed libraries like psutil, to monitor system resources like CPU and memory to help provide real-time visibility for Tejas AI [6].

B. System Design And Architecture

The overall architecture of Tejas AI was developed with an ailment to modularity and scalability. Each component of the system architecture is responsible for a different part of the operation, allowing the system to be developed as multiple independent blocks to follow Engineering principles [2]. The system architecture consists of multiple flows: input, command processing, command execution, and feedback [8]. First, there is the Input Module that collects and captures user inputs either by listening to a prompted challenge through a microphone (voice command) or by receiving it visually through the GUI (type command) [6][7]. After this Input Module captures the command, it sends the command on to the Natural Language Processing (NLP) Module. This Module will analyse the command and interprets the intent of the user to breakdown and identify the task the user wants to facilitate [8]. After this breakdown, the system will send the command onto the Execution Engine, that module will take raw commands as well as sequences of system commands and interpret them into operating system ’system-level action’ that will be performed, e.g. creating files and directories, starting programmes, and performing system shut down operations,

etc. Similar to the previous module feedback is delivered back to the user after Feedback Mechanism Module receives back the commitment from the Execution Engine for feedback. Type to speech (audio) feedback as well as GUI visual confirmation feedback are provided as feedback mechanisms. Moreover, because of its modular built, the system will always be fully adaptable to future changes and upgrades provide remaining as unintrusive as possible allowing the incremental addition of new integrations and new features.

C. Execution Flow

To ensure the reliability, accuracy, and overall performance of Tejas AI, the system went through a comprehensive testing and deployment process. Unit testing was the first step in the testing process, where individual modules were tested on their own such as the execution engine, speech recognition, and natural language processing. The purpose of this step was to determine that the module worked correctly and could perform the given functions without error. After unit tests were performed successfully, the system proceeded through an integration testing phase as a way to test that each module could communicate with each other and work fully integrated as one system. The performance of each module when working with the Input Module, NLP processing, and the Execution Engine was then identified so that any long- term overall feedback could be identified and avoid any pauses or delays in the execution of commands such as executing inferences. Performance testing was conducted to determine how the system would respond when processing multiple instructions and how well it would respond under load. This testing enabled the system to work efficiently without overload when completing multiple instructions simultaneously. Follow- ing the performance testing, a group of users engaged in user testing with the Tejas AI in real-world scenarios, providing their feedback on performance; accuracy; and ease of usability. Based on the feedback user testing provided, the NLP model was enhanced, and the system was fine-tuned for performance. For deployment purposes, Tejas AI was produced into an executable file by means of PyInstaller, making the ON-RAMP process easy to install and run on Windows systems. Users now have the ability to install the AI on their own PC with ease, with no clunky install process and/or difficulty.

1) Input Capture: Tejas AI can capture input in two dif- ferent ways: text and speech. The system uses a microphone that is always" on" to listen for spoken commands, recording spoken input and transcribing it to text. Additionally, users have much more control over how they interact with the AI since they can enter commands directly into the GUI.

2) Input Conversion: The aim of Tejas AI's input con- version stage is to convert the data that you have collected into a format that can then be processed. If you used a voice command then the SpeechRecognition library accurately recognizes the audio and converts it to text. As the Natural Language Processing module converts text input into coded commands, this efficient ability to analyze commands takes place, and there is no need for the conversion.

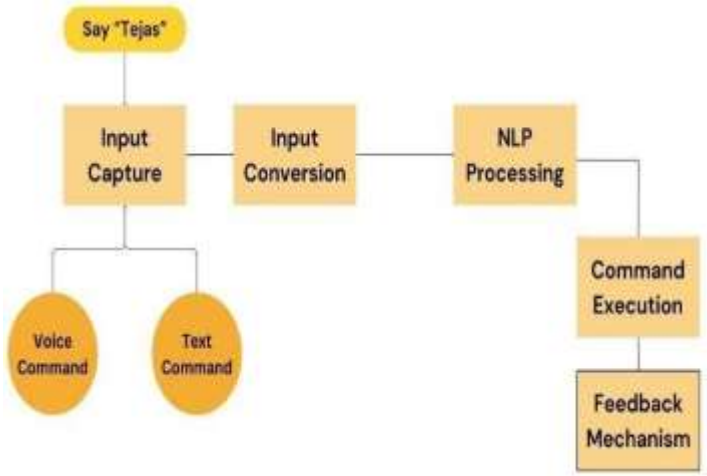
3) Natural Language Processing (NLP):

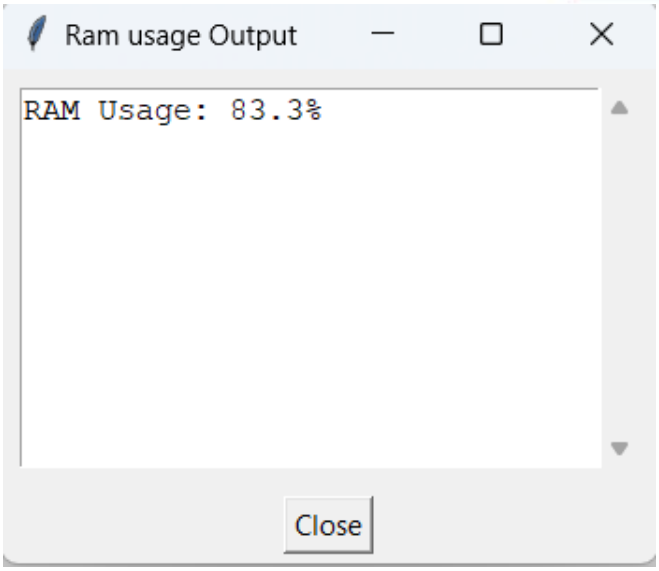
- **Tokenization:** The input text is separated into distinct tokens, or individual words or phrases, to allow for a thorough analysis and understanding of the command structure.
- **Intent Recognition:** The system analyzes the tokens to determine the user's intent and selects the best course of action based on keywords and contextual clues present in the command.
- **Parameter extraction:** By retrieving any additional data or parameters needed to execute the command, this procedure enables the system to do operations (such file naming) accurately.
- **Command Structuring:** The processed data is organized for the Execution Engine to understand, hence preparing the command for the operating system to run.
- **Equations:** Tejas AI provides users with real-time infor- mation about the number of resources their computer is utilizing by using system performance monitoring. These equations are used to calculate the key metrics:

$$(Minfo.A)/1024 * 1024 = MA \quad (1)$$

$$(Minfo.T)/1024 * 1024 = MT \quad (2)$$

$$(Minfo.U)/1024 * 1024 = MU \quad (3)$$





4) Command Execution: The Command Execution section of Tejas AI is critical for interpreting user commands as tasks for the operating system to execute. The Execution Engine will take action once the Natural Language Processing (NLP) module has identified the goal of the user and extracted the relevant parameters [8]. The Identification of Action step begins here, wherein the system uses the processed command to identify the exact task that needs to be executed. For example, if a user says," o p e n a browser," this command will be understood as a command for the system to launch the web browser. Depending on what the user is asking for, the Execution Engine can launch applications, along with other system management functions like file management and configuration settings on the system level. The flexible nature of Tejas AI allows it to handle so many requests from users. After performing the command, the system next enters the Feedback Mechanism, in which it communicates back information about the outcome of its action. Feedback is critical to allowing for a smooth exchange because it indicates if the command was either implemented or how the command may have been mis implemented. These actions in combination ensure Tejas AI not only implements commands accurately but also gives feeds the user at each step and improves the overall experience.

5) Feedback Mechanism: Tejas AI's Feedback Mechanism is an essential element of the communication and interaction between the AI and users, after a command is executed. The purpose of this phase is to clearly state the results of users' requests and enhance the user experience by ensuring they are informed and involved with the system throughout their interactions. Once the Execution Engine completes a task, such as opening an application or retrieving system data, user feedback will be provided in two mediums, visually and auditorily.

The graphical user interface (GUI) used in Tejas AI gives visual feedback, visually showing instantaneous confirmations like" Browser opened successfully" or" Current CPU usage: 25 percent". Having such easy visual confirmations allows users to easily check the outcome ofthe action and be sure the requests are completed satisfactorily. Tejas AI also uses the text-to-speech capabilities of the Pyttsx3 library to provide verbal feedback. This feature creates an option for users that might wish to have the confirmation read aloud, instead of only visually. The AI can now vocalize the outcomes of the orders. The system might say," Opening browser", or" Memory usage displayed", for example, to make it easier for the user to follow— especially if the user is completing other tasks at the same time.

Error handling is also effectively taken care of by the Feedback Mechanism. Any time a command fails for any reason (i.e., application could not be found) or does not get enough permissions granted, the system immediately lets users know both verbally and visually that there is an error. To make sure users are aware of the failure and can take any appropriate corrective action, errors like "Error: Application not found," will not only be brief and clear but will let the user know that their command was incorrect. Generally speaking, Tejas AI's Feedback Mechanism informs users throughout the entire process of contact. This method improves usability while simultaneously providing a clear more intuitive experience by providing both types of feedback - verbal and visual. In the end, both types of feedback will improve user satisfaction and the confidence to manage their own tasks.

D. Execution Flow

The deployment and testing of Tejas AI are essential steps for the system to function as envisaged in real-world applications, bringing seamless PC automation and control to users. There are many parts to the deployment of Tejas AI when installing it into users' systems. The application is fully packaged with required dependencies, such as Text-to- Speech and Speech Recognition libraries, and is packaged in an easy-to-use installer to make installing across different operating system family types as easy as possible, and ensures the users will not have difficulty installing the application by providing an installer. Extensive documentation has also been authored to aid the user experience, including installation instructions, troubleshooting tips, and usage information. Testing becomes a

major effort after the deployment of Tejas AI. This process is important to expose any bugs or performance issues that may appear for the user using the system in normal operation.
The testing phase has many facets:

- **Unit Testing:** Tejas AI's distinct components, such as the Execution Engine and the NLP module, are tested independently to make sure they all function as intended. This makes it feasible to ensure that specific functions—like the carrying out and interpreting of commands—occur as intended.
- **Intent Recognition:** To determine the user's intent, the system analyzes the tokens and selects the appropriate action based on contextual cues and keywords present in the command.
- **Parameter extraction:** By retrieving any additional data or parameters needed to execute the command, this procedure enables the system to do operations (such file naming) accurately.

By properly completing these deployment and testing processes, Tejas AI is positioned to provide users with reliable, efficient, and seamless control and automation capabilities, increasing productivity and user satisfaction.

IV. RESULT

The "Tejas AI Assistant and PC Control" project has successfully produced an excellent and intuitive graphical user interface (GUI) program that enhances user experience and engagement with personal computers. Instead of the burdensome and often frustrating experience of working with personal computers, this technology characterizes communication as improved efficiency and access, by integrating improved voice command recognition to automate processes and manage applications with ease. Speed, accuracy in executing voice commands, and the ability to decide to take actions hands-free are dominant outcomes. The program is created with the user in mind, establishing an intuitive interface that incorporates different levels of user technical skill. Strong security protocols ensure the protection of sensitive data and user privacy is maintained through projects such as the use of data 'encryption' and user authentication. Preliminary user testing received positive theoretically validated appraisals of the operating flexibility and performance of the overall system, with satisfaction reported as high. The application is modularly structured allowing potential scaling, as not all user features or functions have been implemented, allowing updates/changes to be implemented based on user input and advances in technology. This research indicates a practical application of artificial intelligence in everyday computing and improves the prospects of automation and smart technology in the future.

V. CONCLUSION

Through the incorporation of sophisticated voice recognition and automation features into an intuitive user interface, the Tejas AI Assistant and PC Control project seeks to completely transform how people interact with their computers. Key concerns such system compatibility, security, privacy, and user experience have been successfully solved by the project through meticulous feature evaluation, design, and selection.The study shows how artificial intelligence (AI) may improve routine computer work by enabling voice commands to automate procedures, control apps, and adjust system settings. For both pros and casual users, the system offers a dependable and effective solution by putting user demands first and maintaining strong security safeguards.Furthermore, scalability is guaranteed by the modular design, which permits future improvements and the addition of new features in response to user input and technical developments. By following industry best practices for data security and privacy, the system builds user confidence and guarantees adherence to pertinent laws.In essence, the Tejas AI Assistant and PC Control project lays the groundwork for future advancements in AI-driven automation and smart computing solutions while also improv- ing user interaction with technology. This project's successful completion opens the door for additional study and advance- ment in the area, resulting in a more sophisticated and effective digital environment.

REFERENCES

[1] K. K. H. Ng, C. H. Chen, C. K. M. Lee, and R. J. Jiao, "A systematic lit- erature review on intelligent automation: Aligning concepts from theory, practice, and future perspectives," DOI: 10.1016/j.aei.2021.101246.
[2] A. Rayhan, "Artificial Intelligence in Robotics: From Automation to Autonomous Systems," DOI: 10.13140/RG.2.2.15540.42889.
[3] S. S. Shekhar, "Artificial Intelligence in Automation," vol. 4, no. 6, June 2019.
[4] K. Y. Goldberg, "What Is Automation?"
[5] A. Rayhan, "The Future of Work: How AI and Automation Will Transform Industries," DOI: 10.13140/RG.2.2.36092.51848.
[6] E. M. Johnson and E. W. Healy, "The Optimal Speech-to-Background Ratio for Balancing Speech Recognition With Environmental Sound Recognition," DOI: 10.1097/AUD.0000000000001532.
[7] M. A. Kadhar and K. G. Anand, "Graphical User Interface with OpenCV and tkinter," DOI: 10.1007/979-8-8688-0097-96.
[8] J. Chikkarangaiah, A. Uday, D. De Hilster, S. Gangadhar, and J. Shetty, "Enhancing the English natural language processing dictionary using natural language processing++," DOI: 10.11591/ijai.v13.i3.pp3466- 3477.
[9] F. Bakir, S. Wang, T. Ekaireb, J. Pearson, C. Krintz, and R. Wol- ski, "Ambience: an operating system for IoT microservices," DOI: 10.55056/jec.786.
[10] I. Seilonen, T. Pirttioja, P. Appelqvist, and K. Koskinen, "An Approach to Process Automation Based on Cooperating Subprocess Agents," DOI: 10.1007/978-3-540-45185-3-22.