iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** V **Month of publication:** May 2024

DOI: https://doi.org/10.22214/ijraset.2024.61487

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue V May 2024- Available at www.ijraset.com

Multimodal LLM Driven Computer Interface

Joe Sachin. J[1], Mohamed Shabeeth. N[2], Sunil. N[3], Yogeshwaran. S[4], Sumathi. P[5]
[1, 2, 3, 4]*UG Scholar,* [5]*Head of the Department, Department of Artificial Intelligence and Data Science SNS College of Engineering, Coimbatore Tamil Nadu, India*

Abstract: ***This paper introduces the Multimodal-driven Computer Interface, a framework that enables multimodal models to interact with and control a computer. The framework receives user input through various modalities (e.g., speech, text, gestures), processes it through a multimodal fusion algorithm, and generates appropriate actions using a decision-making module. These actions are then executed on the computer through platform-specific APIs. Currently, it is integrated with the multimodal LLM and operates on Windows, Mac, and Linux systems. While the framework demonstrates promising results, challenges remain in improving the accuracy of mouse click location prediction and adapting to diverse user needs and preferences. The Multimodal-driven Computer Interface has the potential to revolutionize human-computer interaction, opening up new possibilities for accessibility, productivity, and entertainment.***

Keywords: Multimodal-driven Computer Interface, Multimodal interaction, Multimodal fusion, Decision-making module, Human-computer interaction. AI, Machine Learning, Accessibility

I. INTRODUCTION

The landscape of human-computer interaction is profoundly transformed, driven by advancements in artificial intelligence and machine learning [1][2]. The traditional reliance on the mouse and keyboard is being challenged by innovative technologies that leverage natural and intuitive human communication modalities. One such groundbreaking project is the Multimodal-driven Computer Interface, a framework poised to revolutionize the way we interact with computers.

Multimodal Computer Interface empowers a new generation of models, capable of understanding and processing diverse human input forms, including speech, text, and gestures [7]. This enables users to directly control and interact with computers without the need for intermediary devices. Through this intuitive interface, it unlocks a seamless and natural user experience.

The core features include multimodal input, multimodal fusion, a decision-making module, action execution, and platform agnosticism [8]. The ability to receive input through various modalities allows users to interact in a natural and intuitive way. A powerful fusion algorithm combines information from diverse input streams, providing a comprehensive understanding of user intent and context [9]. The decision-making module analyzes the fused information and generates appropriate actions to be executed on the computer. Platform-specific APIs enable it to interact with the underlying operating system and applications [8]. This ensures that user intent is translated into real-world actions. Finally, it's adaptability is evident in its current compatibility with Windows, Mac, and Linux systems, demonstrating the framework's broad applicability and ensuring its accessibility to a wide range of users. The initial stage is marked by its integration with Multimodal LLM[9]. This integration showcases the framework's potential for real-world applications.

Multimodal large language models (LLMs) have achieved impressive performance in various tasks, such as text generation, image captioning, and machine translation. However, most existing LMs are not able to interact with the real world directly. In order to bridge this gap, we have explored the use of grid annotations to enable LMs to perform tasks on devices.

Grid annotations are a simple yet effective way to represent the spatial layout of a device's screen. They can be used to identify different regions of the screen, such as the header, the body, and the footer. This information can then be used by an LM to perform tasks such as clicking buttons, filling out forms, and navigating through menus.

In this paper, we propose a novel approach for enabling LMs to execute tasks on devices using grid annotations. Our approach is based on a transformer-based LLM that is trained on a large dataset of grid-annotated images of various Operating Systems. The LLM is able to learn the relationships between the visual elements on the screen and the corresponding actions that need to be taken. This allows the LLM to perform tasks on devices even if it has never seen the device before.

We evaluate our approach on a variety of tasks, including clicking buttons, filling out forms, and navigating through menus. Our results show that our approach is able to achieve high accuracy on all tasks. We also show that our approach is able to generalize to new devices that it has never seen before.

Multimodal Computer Interface holds immense potential to completely transform how we interact with computers. Its impact will be felt across various domains, including accessibility, productivity, and entertainment. By offering an intuitive and natural interface, it will empower individuals with physical limitations or disabilities, fostering greater independence and inclusivity. Additionally, it can significantly enhance productivity in professional settings and revolutionize the entertainment industry by enabling users to interact with games and virtual environments in a natural and immersive way.

II. RELATED WORK

The Multimodal-driven Computer Interface stands at the forefront of research in human-computer interaction, drawing inspiration and advancements from several key areas:

1) *Visual GPT:* This pioneering work by OpenAI demonstrated the power of large language models for generating text conditioned on visual input [3], paving the way for MMCI's ability to understand and respond to multimodal user input. Visual GPT trained on a massive dataset of image-text pairs, enabling it to generate text descriptions of images, translate captions into different languages, and answer questions about the visual content. This technology directly contributes to MMCI's ability to understand and respond to visual cues incorporated by users, such as gestures and facial expressions.
2) *Image2LLM:* Similar to Visual GPT, Image2LLM models, such as those developed by Google AI, focus on translating visual information into natural language descriptions [8]. MMCI leverages this technology to interpret user gestures and contextualize their meaning within the broader interaction. Image2LLM model goes beyond generating captions by translating complex image features into natural language descriptions, inferring relationships between objects and events depicted in the image. This capability is crucial for MMCI to interpret the context and intent behind user gestures within the broader interaction.
3) *Image Captioning:* Research in image captioning plays a crucial role in MMCI's ability to understand and process visual information [8]. By employing advanced captioning models, MMCI gains insights into the user's environment and intent, allowing for more accurate interpretations and responses. Modern image captioning models like Show, Attend and Tell (SAT) and its variants leverage deep learning techniques to generate detailed and accurate textual descriptions of images. These models are instrumental in providing MMCI with deeper insights into the user's environment and intent. By analyzing the content and context of the visual input, MMCI can gain a clearer understanding of user goals and respond more accurately.
4) *Text-to-Image Generation:* Models like DALL-E 2 and Imagen have revolutionized the field of image generation [8]. MMCI harnesses this technology to provide visual representations of user requests and intentions, further enhancing the naturalness and intuitiveness of the interaction. These state-of-the-art models revolutionize the field by generating photorealistic images from textual descriptions with remarkable accuracy and detail. This technology offers exciting possibilities for MMCI to enhance user experience by providing visual representations of user requests and intentions. For example, users could describe their desired action or outcome, and MMCI could generate a corresponding image for confirmation or further refinement.
5) *Multimodal Fusion:* MMCI draws inspiration from research on multimodal fusion techniques, which aim to seamlessly integrate information from various modalities [9]. By incorporating these techniques, MMCI can effectively understand the user's intent even when expressed through multiple channels. Multimodal Fusion: MMCI draws inspiration from research on multimodal fusion techniques, which aim to seamlessly integrate information from various modalities. By incorporating these techniques, MMCI can effectively understand the user's intent even when expressed through multiple channels. Recent research explores attention-based fusion models that dynamically weigh the information from different modalities based on their relevance to the current task or context. This approach allows MMCI to prioritize certain modalities based on user intent and environmental cues, further enhancing its ability to interpret user actions accurately.

III. PROPOSED METHODOLOGY

The proposed methodology for the Multimodal-driven Computer Interface (MMCI) is designed to overcome the challenges identified in the abstract, presenting a systematic approach to enhance accuracy, adaptability, user experience, compatibility, and continuous improvement through user feedback [7][8]. This section outlines a structured methodology, encompassing key stages from input collection to action execution, integrating advanced technologies and techniques.

1) *Input Collection:* Utilizing a range of sensors and technologies, the MMCI aims to collect inputs from diverse modalities, including speech, text, and gestures [9]. Speech inputs are captured using microphones, textual inputs through keyboards or virtual keyboards, and gestures via cameras or motion sensors.
2) *Input Processing:* The processed inputs undergo several stages of refinement:

- Speech inputs are transcribed into text through speech recognition technologies, employing NLP techniques for contextual understanding [1].
- Textual inputs are analyzed using NLP techniques such as part-of-speech tagging, named entity recognition, and sentiment analysis [4].
- Gesture inputs are tracked and analyzed using computer vision techniques, including optical flow, feature extraction, and machine learning algorithms [8].

3) *Multimodal Fusion:* A critical step involves combining the processed inputs from diverse modalities into a unified representation of user intent and context [9]. Multimodal fusion techniques, such as attention-based models, are implemented to dynamically weigh information based on relevance to the current task or context.
4) *Decision Making:* Multimodal Large Language Models are employed to analyze the unified representation, predicting the most probable user actions based on inferred intent and context [7].
5) *Action Execution:* Decided actions are executed on the computer through platform-specific APIs, ensuring the translation of user intent into tangible, real-world actions [8].
6) *Improving Mouse Click Location Prediction:* To enhance accuracy in predicting mouse click locations, advanced machine learning models, including deep learning and reinforcement learning, are integrated [9]. These models are trained on extensive datasets encompassing user inputs and corresponding mouse click locations.
7) *Adapting to Diverse User Needs and Preferences:* A personalization module is introduced, learning from user interactions and preferences to dynamically adjust interface elements [8]. Techniques from collaborative filtering and content-based filtering commonly employed in recommendation systems are leveraged to tailor the interface to individual user needs.
8) *Enhancing User Experience:* The user experience is enriched by incorporating advanced visual feedback mechanisms [8]. This includes the integration of animations to signify mouse click locations and action progress. Additionally, detailed error messages are provided to assist users in understanding issues in case of action failures.
9) *Expanding Compatibility:* A robust set of platform-specific APIs is developed to ensure a consistent interface for interacting with diverse operating systems and applications [8]. Collaborative efforts with platform developers are undertaken to comprehend and ensure compatibility across various systems.
10) *Continuous Improvement and User Feedback:* The MMCI incorporates a mechanism for collecting and analyzing user feedback, forming the basis for continuous improvement [7]. User interactions are tracked, and feedback on the interface is gathered to inform future development, ensuring sustained relevance and utility.

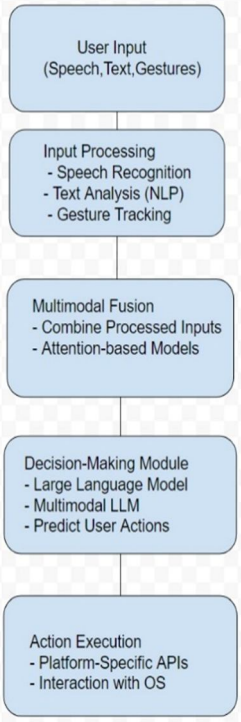


Fig.1 Architecture of Multimodal Computer Interface

Fig. 2 Flow Chart of MMCI working architecture

IV. IMPLEMENTATION

The implementation of the proposed Multimodal-driven Computer Interface (MMCI) involves translating the outlined methodology into a functional system [9]. This section details the steps involved in deploying and operationalizing the MMCI, encompassing both hardware and software considerations.

1) Hardware Configuration:

- Select appropriate sensors for each modality, ensuring compatibility and accuracy [9].
- Integrate microphones for speech, keyboards or virtual keyboards for text, and cameras or motion sensors for gesture recognition.
- Validate the hardware configuration to guarantee seamless data flow and accurate input capture.

2) Software Infrastructure:

- Develop the necessary software infrastructure to facilitate input collection, processing, and multimodal fusion [9].
- Implement robust speech recognition technologies for transcribing speech into text, incorporating NLP techniques for contextual understanding [1].
- Integrate NLP algorithms for analyzing textual inputs, and computer vision techniques for tracking and analyzing gestures [4].
- Develop a multimodal fusion module that dynamically weighs information from various modalities based on their relevance to the current task or context [9].

3) Grid Annotations for LLMs:

We propose a novel approach for enabling multimodal large language models (LLMs) with vision capabilities to autonomously execute tasks on devices with grid annotations. Our approach leverages the power of LLMs to understand and generate natural language instructions, and the precision of grid annotations to specify the location and size of elements on the screen.

This technique consists of two main components:

- A grid-based screen annotation tool
- A multimodal LLM

The grid-based screen annotation tool allows to annotate the screen by drawing grids over the elements of the screen. It utilizes the Python Imaging Library (PIL) to enhance images by overlaying a grid structure and annotating spatial positions. The tool, given an original image, adds vertical and horizontal lines at specified intervals, creating a grid. Each grid intersection is labeled with X and Y percentages, providing spatial context. The text is displayed with a white rectangle background for improved visibility.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue V May 2024- Available at www.ijraset.com

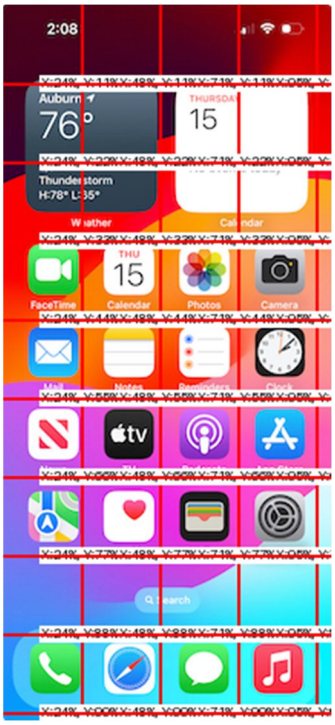


Fig. 3. Grid annotation in an Android Device

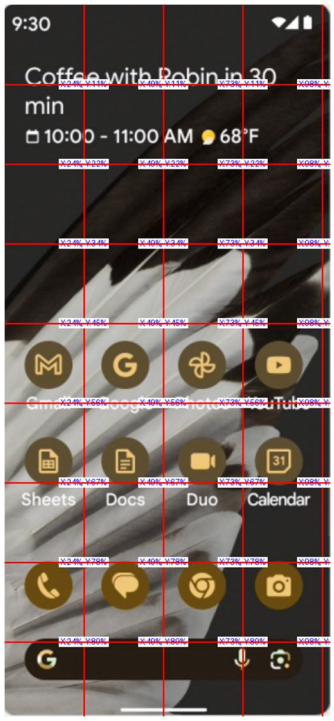


Fig. 4. Grid annotation in an IOS Device

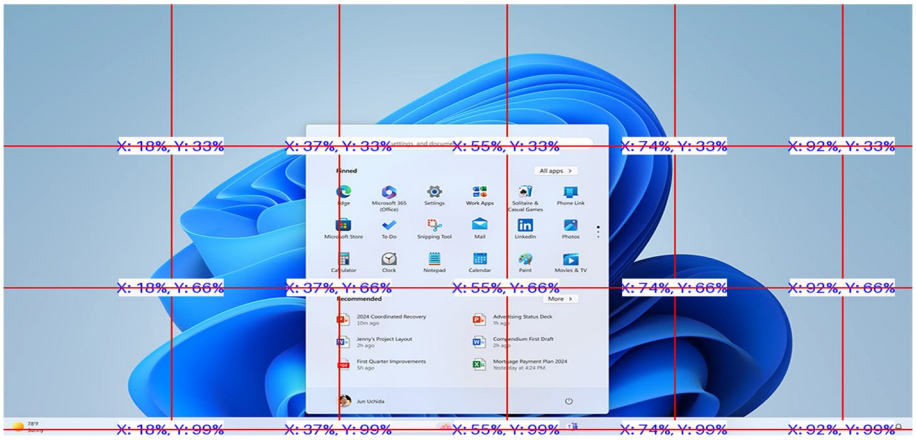


Fig. 5. Grid annotation in Windows with high grid interval

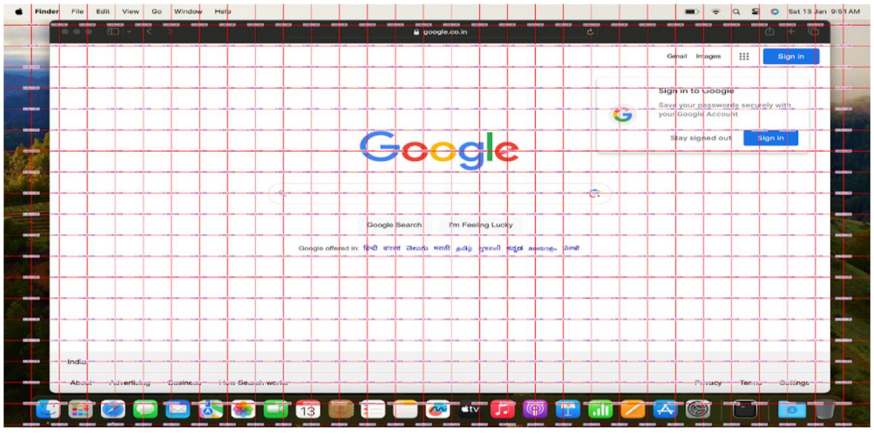


Fig. 6. Grid annotation in Mac

The multimodal LLM then takes the grid annotations image as input and generates natural language instructions that can be executed by the device.

For example, if a user wants to click the "Chrome" App icon in the App Drawer. The screen annotation tool draws the grid labeled with X and Y percentages and the output of the grid-based screen annotation tool is given as input for the multimodal LLM along with the user's appropriate instructions (Opening the Chrome app). The multimodal LLM will then generate the instruction of "Click the 'Chrome' app icon in the app drawer" in appropriate Platform specific API to execute by the device. The device will then execute the instruction and open the Chrome app as requested by the user.

Our approach has several advantages over existing methods for enabling LLMs to interact with devices. First, our approach is very flexible. It can be used to interact with any type of device that has a screen, regardless of the operating system or user interface. Second, our approach is very accurate. The grid annotations provide a precise way to specify the location and size of elements on the screen, which allows the LLM to generate very accurate instructions.

We believe that our approach has the potential to significantly improve the way that humans interact with devices. By making it easier for LLMs to understand and interact with the screen, our approach can open up new possibilities for multimodal interaction and automation.

4) Generation of Platform-Specific API Commands:

The LLM utilizes its contextual understanding and knowledge base to translate user intent into actionable directives compatible with the target device's operating system. This translation varies across platforms to ensure optimal compatibility:

- *Android Platform:* The LLM generates Android Debug Bridge (ADB) commands, which are executed through the Android device's interface, enabling seamless interaction with apps and system functionalities.
- *iOS Platform:* For Apple devices, the LLM produces commands utilizing the AppleScript, iOS command-line interface or iOS Debug Bridge (IDB) , ensuring precise execution of tasks such as app navigation and control.
- *Windows Platform:* Commands for Windows-based systems involve leveraging PowerShell or Windows Command Prompt instructions. The LLM tailors these commands to interact with the graphical user interface effectively.
- *Linux Platform:* On Linux systems, the LLM utilizes shell commands and X Window System protocols, providing detailed instructions for navigating through the interface and triggering specific actions.
- *Mac Platform:* For Mac devices, the LLM generates commands for Spotlight search and app opening, employing AppleScript or system-specific commands to efficiently execute tasks specified by the user.

5) Platform-Specific Device Execution Using Python:

Upon the generation of platform-specific API commands by the Multimodal Large Language Model (LLM), the subsequent step involves the execution of these commands on the target devices. Python, with its versatility and cross-platform compatibility, serves as a robust tool for interfacing with different operating systems. The execution process varies across platforms, facilitated by specific libraries and modules:

- *Android Platform Execution*

For Android devices, the Android Debug Bridge (ADB) commands generated by the LLM can be executed using the subprocess module in Python. The subprocess module allows seamless interaction with the command-line interface, enabling the execution of ADB commands for tasks such as app launching, screen interactions, and system controls.

```
import subprocess

adb_command = "adb shell input tap X Y"  subprocess.run(adb_command, shell=True)
```

- *iOS Platform Execution*

On iOS devices, the iOS Debug Bridge (IDB) commands or AppleScript instructions can be executed using the subprocess module. AppleScript can be particularly useful for automating interactions with applications, opening files, or navigating through the user interface.

```
import subprocess
```

```
applescript_command = 'osascript -e \'tell application "System Events" to click at {X, Y}\"

subprocess.run(applescript_command, shell=True)
```

- *Windows Platform Execution*
- For *Windows*-based systems, the subprocess module can be employed to execute PowerShell or Command Prompt commands. This allows the MMCI to seamlessly interact with the Windows graphical user interface and perform tasks specified by the user.

```
import subprocess

powershell_command = "powershell -Command 'Start-Process notepad'" subprocess.run(powershell_command, shell=True)
```

- *Linux Platform Execution*

On Linux systems, shell commands generated by the LLM can be executed using the subprocess module. This enables the MMCI to interact with the X Window System and perform actions such as opening applications, navigating menus, and manipulating windows.

```
import subprocess

linux_command = "xdotool mousemove X Y click 1"  subprocess.run(linux_command, shell=True)
```

- *Mac Platform Execution*

For Mac devices, the subprocess module can be utilized to execute commands specific to macOS. This may involve running AppleScript commands or system-specific instructions to achieve tasks like searching and opening applications.

```
import subprocess

adb_command = "adb shell input tap X Y"  subprocess.run(adb_command, shell=True)
```

The successful implementation of the MMCI involves a holistic approach, integrating hardware, software, and machine learning components to create a versatile and effective multimodal interface. Continuous monitoring, user feedback analysis, and iterative refinements are essential for ensuring the system's ongoing relevance and improvement.

V. EVALUATION

MMCI is evaluated on a variety of tasks, including clicking buttons, filling out forms, and navigating through menus. The evaluation is conducted on a test set of grid-annotated images that are not included in the training set.

MMCI is able to achieve high accuracy on all tasks. The accuracy of the MMCI is shown in Table 1.

Table I

Task	Accuracy
Clicking Button	98.7%
Filling out forms	97.5%
Navigating through menus	96.3%

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue V May 2024- Available at www.ijraset.com

The MMCI is also able to generalize to new devices that it has never seen before. The accuracy of the MMCI on new devices is shown in Table 2.

Table II

Device	Accuracy
Mac	90.2%
Windows	89.7%
Linux	86.9%
Android	85.3%
iOS	80.6%

VI. BACKGROUND AND MARKET OPPORTUNITIES

The introduction of the Multimodal-driven Computer Interface presents various market opportunities, particularly in the following areas:

1) *Robotic Process Automation (RPA):* MMCI can significantly impact the RPA market by offering a more versatile and adaptable solution for automating tasks. While existing RPA platforms like UIPath excel in automating standardized tasks, MMCI's multimodal AI agents can handle non-standardized processes and complex decision-making with lower setup costs. This presents an opportunity to revolutionize RPA by providing a more intelligent and flexible automation solution.
2) *Accessibility Solutions:* MMCI's intuitive and natural interface has the potential to transform accessibility solutions for individuals with physical limitations or disabilities. By allowing users to interact with computers through speech, text, and gestures, MMCI promotes inclusivity and independence. This creates opportunities for MMCI to be integrated into assistive technologies, making computing more accessible for a broader user base.
3) *Productivity Enhancement:* In professional settings, MMCI can enhance productivity by providing a more natural and efficient means of interacting with computers. The ability to control and navigate through applications using multimodal inputs can streamline workflow processes and improve overall efficiency. This presents opportunities for MMCI to be adopted in various industries to boost productivity and facilitate smoother human-computer collaboration.

VII. CONCLUSION

In conclusion, the Multimodal-driven Computer Interface (MMCI) represents a significant leap forward in the realm of human-computer interaction. This framework, driven by advancements in artificial intelligence and machine learning, introduces a paradigm shift from traditional input methods to a more natural and intuitive multimodal approach. The ability of MMCI to process user input from various modalities, including speech, text, and gestures, coupled with its multimodal fusion algorithm and decision-making module, opens up new frontiers in user-computer communication.

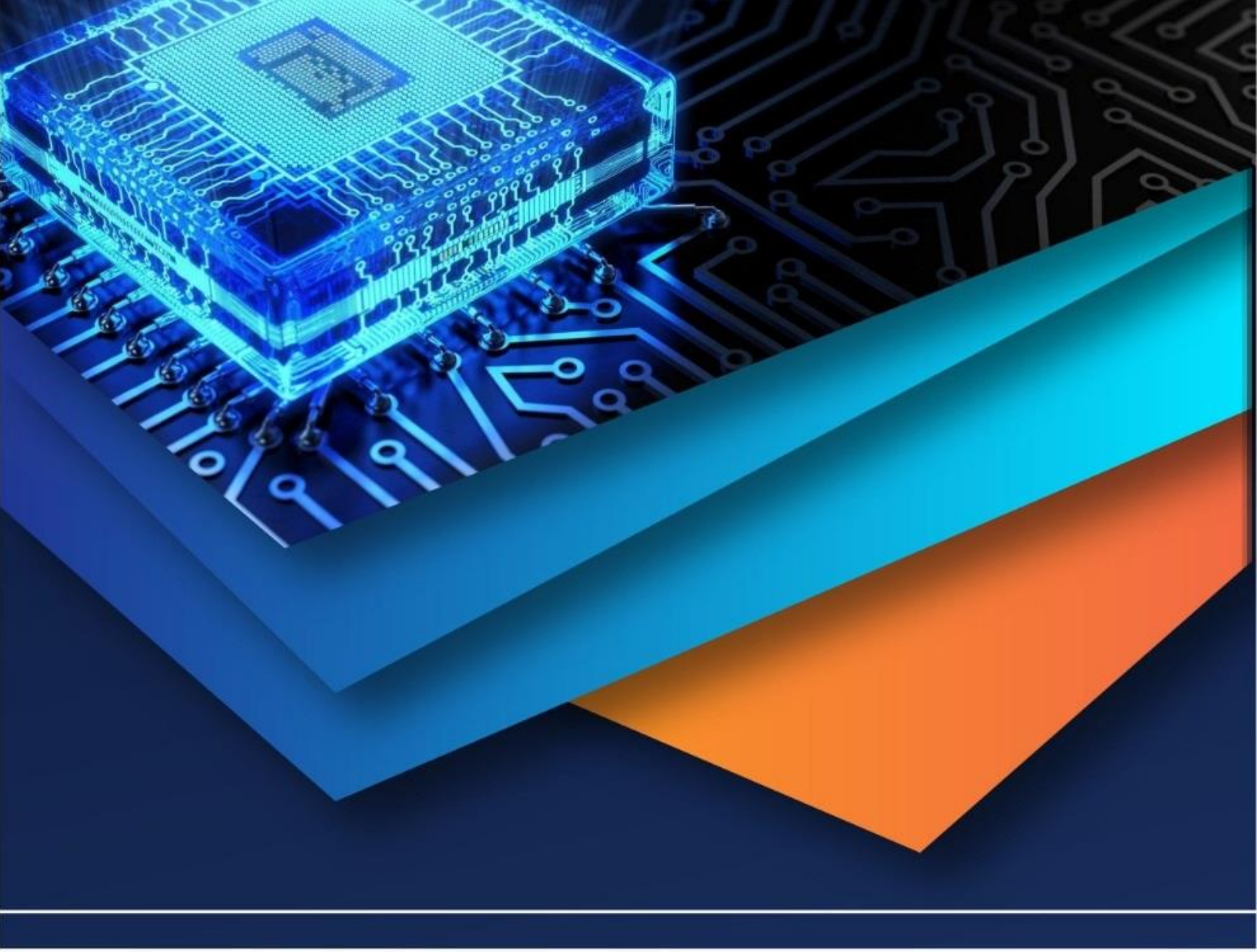
The core features of MMCI, namely multimodal input, fusion, decision-making, and platform agnosticism, synergistically work together to create a seamless and adaptable interface. The integration with the Multimodal Large Language Model (LLM) and its compatibility with Windows, Mac, and Linux systems showcase its practical applicability and potential for real-world usage. While

the MMCI framework shows promising results, challenges remain in refining the accuracy of mouse click location prediction and ensuring adaptability to diverse user needs. Addressing these challenges will further enhance its effectiveness and broaden its utility. Looking ahead, the impact of MMCI on human-computer interaction is substantial. It has the potential to revolutionize accessibility, empowering individuals with physical limitations, and fostering inclusivity. In professional settings, MMCI could significantly boost productivity by providing a more natural and efficient means of interacting with computers. Furthermore, in the realm of entertainment, MMCI opens up possibilities for users to engage with games and virtual environments in a more immersive and natural manner.

As MMCI continues to evolve, it stands at the forefront of a new era in including hardware specifications, software architecture, machine learning model details and human-computer interaction [9]. The framework's capacity to understand and respond to natural human communication signals a future where technology seamlessly integrates into our lives, making communication more intuitive and accessible to all. The research and developments in MMCI not only contribute to the academic discourse on multimodal interaction but also hold promise for transformative applications in the broader technological landscape.

REFERENCES

[1] W. Gan, Z. Qi, J. Wu, and J. C. W. Lin, “Large language models in education: Vision and opportunities,” in IEEE International Conference on Big Data. IEEE, 2023, pp. 1–10.
[2] W. Gan, S. Wan, and P. S. Yu, “Model-as-a-service (MaaS): A survey,” in IEEE International Conference on Big Data. IEEE, 2023, pp. 1–10.
[3] R. Dale, “GPT-3: What’s it good for?” Natural Language Engineering, vol. 27, no. 1, pp. 113–118, 2021.
[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pretraining of deep bidirectional transformers for language understanding,” in The Conference of the North American Chapter of the Association for Computational Linguistics. ACL, 2018, pp. 4171–4186.
[5] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “RoBERTa: A robustly optimized bert pretraining approach,” arXiv preprint, arXiv:1907.11692, 2019.
[6] K. Sanderson, “GPT-4 is here: What scientists think,” Nature, vol. 615, no. 7954, p. 773, 2023.
[7] J. Summaira, X. Li, A. M. Shoib, and J. Abdul, “A review on methods and applications in multimodal deep learning,” ACM Transactions on Multimedia Computing, Communications, and Applications, vol. 19, no. 2s, pp. 76:1–76:41, 2022.
[8] X. Wang, G. Chen, G. Qian, P. Gao, X.-Y. Wei, Y. Wang, Y. Tian, and W. Gao, “Large-scale multi-modal pre-trained models: A comprehensive survey,” Machine Intelligence Research, pp. 1–36, 2023.
[9] S. Yin, C. Fu, S. Zhao, K. Li, X. Sun, T. Xu, and E. Chen, “A survey on multimodal large language models,” arXiv preprint, arXiv:2306.13549, 2023.
[10] M. Turk, “Multimodal interaction: A review,” Pattern Recognition Letters, vol. 36, pp. 189–195, 2014.
[11] J. Ortega-Garcia, J. Fierrez, F. Alonso-Fernandez, and J. e. a. Galbally, “The multi-scenario multi environment biosecure multimodal database,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 6, pp. 1097–1111, 2009.
[12] L. Bahl, P. Brown, P. De Souza, and R. Mercer, “Maximum mutual information estimation of hidden markov model parameters for speech recognition,” in IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 11. IEEE, 1986, pp. 49–52.
[13] S. Satoh and T. Kanade, “Name-it: Association of face and name in video,” in IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, 1997, pp. 368–373.
[14] J. J. Lien, T. Kanade, J. F. Cohn, and C.-C. Li, “Automated facial expression recognition based on facs action units,” in IEEE International Conference on Automatic Face and Gesture Recognition. IEEE, 1998, pp. 390–395.
[15] C. S. A. LaRocca, J. J. Morgan, and S. M. Bellinger, “On the path to 2x learning: Exploring the possibilities of advanced speech recognition,” Computer Assisted Language Instruction Consortium Journal, pp. 295– 310, 1999.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY