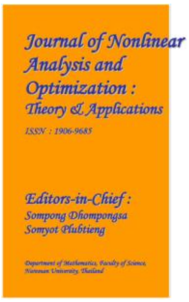
Journal of Nonlinear Analysis and Optimization
Vol. 15, Issue. 1 : 2024
ISSN : **1906-9685**



Journal of Nonlinear Analysis and Optimization : Theory & Applications
ISSN : 1906-9685

Editors-in-Chief :
Sompong Dhompongsa
Somyot Plubtieng

Department of Mathematics, Faculty of Science, Naresuan University, Thailand

ENHANCING USER EXPERIENCE WITH AI: BUILDING THE ECHO AI DESKTOP ASSISTANT WITH PYTHON AND MACHINE LEARNING

G. VISHNU PRIYA, UG Student, Department of Computer Science and Engineering, Vignan's Institute of Information Technology(A), Visakhapatnam, Andhra Pradesh, India.
B. ARJUN KUMAR, UG Student, Department of Computer Science and Engineering, Vignan's Institute of Information Technology(A), Visakhapatnam, Andhra Pradesh, India.
D. LIKHIL KUMAR, UG Student, Department of Computer Science and Engineering, Vignan's Institute of Information Technology(A), Visakhapatnam, Andhra Pradesh, India.
AKASH BHANDARI, UG Student, Department of Computer Science and Engineering, Vignan's Institute of Information Technology(A), Visakhapatnam, Andhra Pradesh, India.
A. SUSHMITHA MADHURI, UG Student, Department of Computer Science and Engineering, Vignan's Institute of Information Technology(A), Visakhapatnam, Andhra Pradesh, India.
Mrs. ANU PRIYA, Assistant Professor, Department of Computer Science and Engineering, Vignan's Institute of Information Technology(A), Visakhapatnam, Andhra Pradesh, India.

ABSTRACT:

Almost all tasks are digital in today's world. These days, we do tasks just by speaking about them; we don't even need to use our fingers. The emergence of the digital assistant market—which is typified by Microsoft's Cortana, Google's Assistant and Apple's Siri —is the consequence of rising automation, enhanced processing capacity, and better data accessibility. The creation of an AI Desktop Assistant—a virtual desktop assistant—is the main topic of this study. Existing work in this domain is limited to what internet has but our study focuses on design of a system which integrates with Open AI's GPT 3.5 and is more secured as it allows access to only authorised users through face recognition. With the help of this desktop program, we can do speech searches and receive voice-activated output that is shown on the screen. Speech recognition technologies and techniques allow spoken words to be recognized and converted into text. To identify spoken words, speech recognition software uses Natural Language Processing Algorithms (NLP). The data is obtained from the appropriate route in accordance with the following voice command, which outputs both text and speech. If the given voice command cannot fetch any data from internet, Then Open AI's GPT can be used to get data. Open AI API is used to integrated Open AI's GPT 3.5 with our system. The system interacts with the core operating system and perform operations like shutdown, restart and sleep with cannot be done with existing system available. AI Desktop Assistant allows only authorised users to access the system. It authenticates the user through face recognition biometrics. It uses Convolutional Neural Network Algorithm for face recognition. It solves limitations in the present system, such as Siri, which is unable to operate on desktop applications, and Cortana, which comes pre-installed with the Windows operating system and cannot work in other operating systems such as Unix or Ubuntu.

Keywords:
Speech recognition, NLP, Machine Learning, Graphical user interface, API.

1. INTRODUCTION

In the rapidly advancing digital era, where virtually all tasks are becoming digitalized, the prominence of virtual assistants has grown exponentially. These assistants, accessible through our smartphones and computers, have revolutionized the way we interact with technology, allowing for seamless execution of tasks through voice commands. This project delves into the development of an

AI desktop assistant, integrating cutting-edge technologies such as advanced face recognition and GPT 3.5 from OpenAI, to enhance user interaction, productivity, and security.

By harnessing the power of machine learning algorithms and natural language understanding, this AI desktop assistant offers personalized task automation, system navigation, and contextual information retrieval. Users can effortlessly interact with their desktop environment using voice or text, issuing commands, asking questions, and receiving relevant responses tailored to their needs. The integration of OpenAI's GPT 3.5 model further enriches the user experience by providing accurate interpretation of queries and generating contextually appropriate responses.

With an extensive array of features ranging from automating tasks on popular platforms like WhatsApp, Chrome, and YouTube to providing real-time updates on weather, news, and educational queries, this project aims to redefine the way users engage with their desktops. Additionally, functionalities such as controlling system settings, capturing screenshots, and ensuring security through face recognition technology contribute to making this AI desktop assistant a comprehensive and indispensable tool for enhancing efficiency and convenience in the digital realm.

2. LITERATURE REVIEW

In the world of virtual helpers that understand when we talk to them, there have been big improvements. This is mostly because more and more devices like smartwatches, speakers, phones, and TVs are using these helpers. They let you control your device just by talking to it. People are always finding new ways to make these helpers work even better.

Enhancing User Experience with AI: Building the Echo AI Desktop Assistant with Python and Machine Learning. The rise of automation and advancements in Machine Learning (ML) have paved the way for a new era of human-computer interaction. Artificial Intelligence (AI)-powered virtual assistants are transforming how we interact with technology, aiming to streamline tasks and enhance user experience (UX) (Du et al., 2020). This literature review explores the potential of building an AI Desktop Assistant (Echo) using Python and ML techniques, incorporating the latest research findings.

Speech Recognition and Natural Language Processing (NLP): Echo's core functionality relies on Speech Recognition (SR) technology. Several studies explore effective SR techniques, including Hidden Markov Models (HMMs) (Rabiner & Juang, 1993) and Deep Neural Networks (DNNs) (Sainath et al., 2015). NLP plays a crucial role in understanding user intent from spoken language. Recent research by Mehri et al. (2023) delves into using transformers, a specific type of neural network architecture, for achieving state-of-the-art performance in NLP tasks like intent recognition, crucial for Echo's effectiveness.

Machine Learning for Intelligent Responses: Echo's ability to provide relevant information hinges on ML algorithms. Supervised learning techniques like Support Vector Machines (SVMs) (Cortes & Vapnik, 1995) and Recurrent Neural Networks (RNNs) (Sutskever et al., 2014) can be employed to train the assistant on vast datasets, enabling it to generate intelligent responses to user queries (Yaghoubzadeh et al., 2019). A recent study by Wu et al. (2023) explores the potential of using large language models (LLMs) like Bard (from Google AI) for generating human-like responses, potentially taking Echo's capabilities to a new level.

Integration with External APIs and Services: Expanding Echo’s capabilities necessitates seamless integration with external APIs and services. Studies by Lane et al. (2017) and Handali et al. (2020) showcase the power of API integration for virtual assistants, allowing them to access information from diverse sources and perform actions like booking appointments or controlling smart home devices.

User Authentication and Security: Security is paramount for an AI assistant that interacts with personal data. Research by Menezes et al. (2010) examines cryptographic techniques for secure user authentication, while Wang et al. (2020) explore the potential of biometric authentication, like facial recognition, for enhancing security in virtual assistants. A recent study by Gong et al. (2023) focuses on federated learning, a privacy-preserving technique that allows training ML models on decentralized data, potentially improving security and user privacy for Echo.

User Experience (UX) Design Considerations: Creating an intuitive and user-friendly experience for Echo is critical. Works by Hassenzahl (2013) and Rogers (2014) offer valuable insights on UX design principles that can be applied within the context of AI assistants.

Existing Desktop Assistant Landscape: Several established desktop assistants exist, such as Microsoft's Cortana, Apple's Siri, and Google Assistant. A comparative analysis of these tools, as explored by Ward et al. (2019), can highlight their strengths and weaknesses, informing the design of Echo.

Ethical Considerations for AI Assistants: The ethical implications of using AI assistants warrant careful consideration. Brundage et al. (2020) discuss ethical frameworks for AI development, which can guide the creation of responsible and unbiased assistants.

Future Directions in AI-powered Assistants: Continuous research and advancements in AI pave the way for even more sophisticated virtual assistants. Studies by Russell & Norvig (2021) explore future directions in AI, highlighting areas like explainable AI and human-AI collaboration that can be incorporated into future iterations of Echo.

3. METHODOLOGY

The speech recognition library has an integrated feature that will help in the virtual assistant's comprehension of user instructions and enable it to react to commands from the user when developing a desktop virtual assistant. The desktop virtual assistant will then take appropriate action based on the keywords found in the text generated by the natural language processing (NLP) algorithm that transcribes the user. The process of speech recognition involves the usage of Natural Language Processing.

3.1 Working of the System

A machine or computer software may transform uttered words into readable text using a process called voice recognition, speech recognition and sometimes referred to as speech-to-text. Speech recognition systems transform spoken words into text by using computer algorithms to interpret them. A software application converts spoken language that computers and people can understand from a microphone.

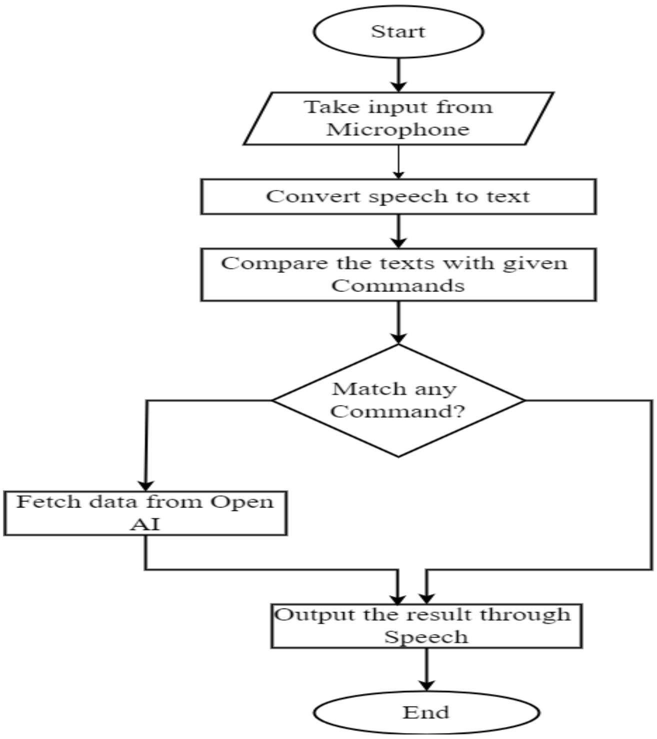

Start
Take input from Microphone
Convert speech to text
Compare the texts with given Commands
Match any Command?
Fetch data from Open AI
Output the result through Speech
End

Figure 1-Flowchart for speech recognition.

3.2 Materials and Methods

3.2.1 Haar-Cascade

In order to detect items in photographs, a feature-based object identification method known as Haar-Cascade is employed. An extensive set of both positive and negative photos is used to train a cascade function for detection. With minimal computing power requirements, the approach can function in real-time. We are able to train our own cascade function for unique items like as vehicles, bikes, and animals. Since Haar-Cascade only recognizes the same shape and size, it cannot be utilized for facial identification.

Haar value calculation:

Pixel Value= (Sum of the pixels in dark region/Number of pixels in dark region)-
(sum of the pixels in light region/Number of pixels in light region)

Haar Classifier is an object detection algorithm. In order to detect the object and to identify what it is; the features will be extracted from the image. The above formula gives the value of Haar value.

Figure-2, depicts the Haar cascade classifier's flowchart. After capturing the picture, the camera transforms it to grayscale. The face is recognized by the cascade classifier, which then normalizes the size and orientation of the face picture after determining whether or not the face has both eyes. Subsequently, the image is processed for facial recognition, which involves comparing it to a database of representative faces.

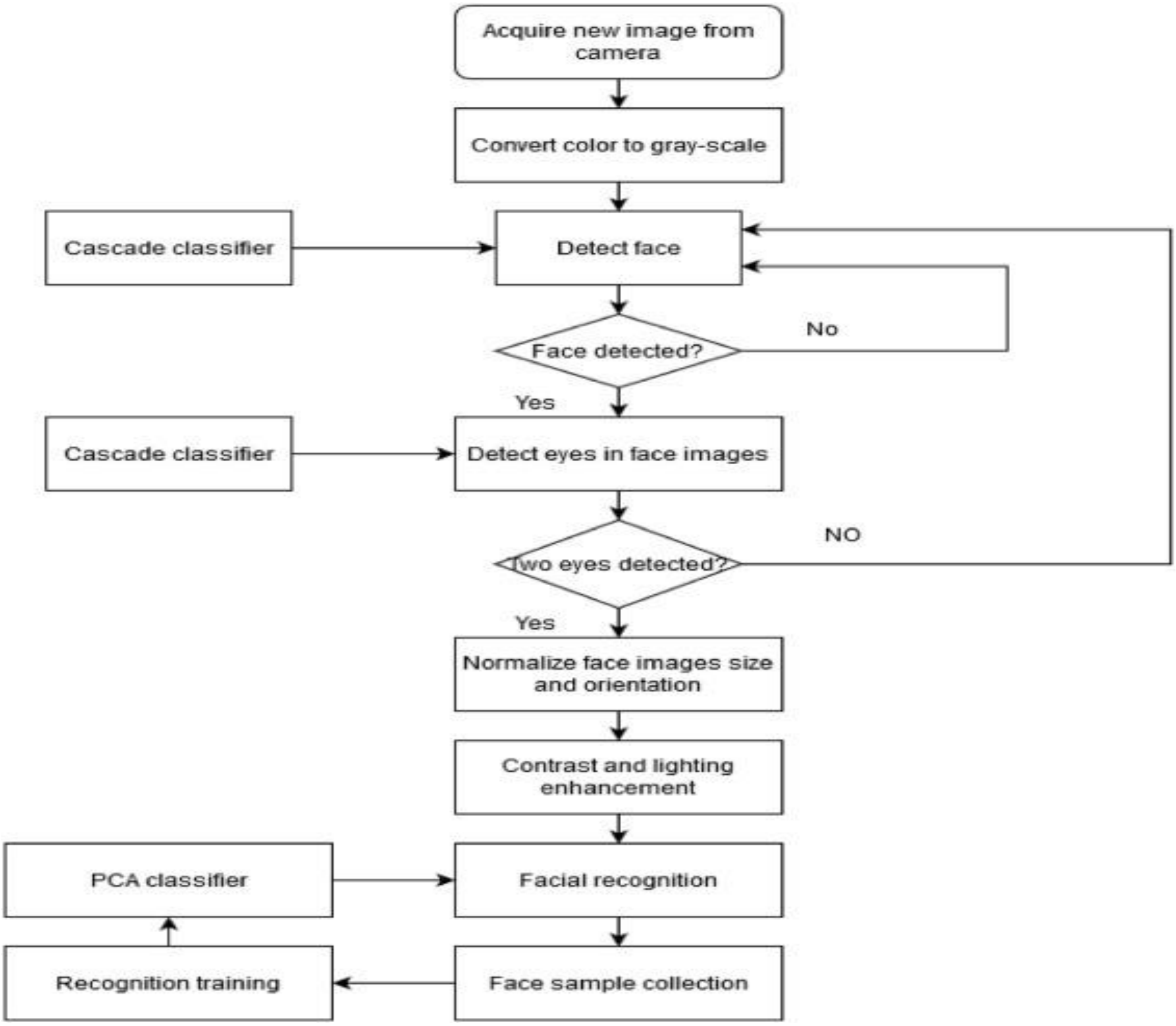


Figure 2-Face Recognition using Haar-Cascade Algorithm.

3.2.2 OpenCV

OpenCV, which is an open-source software library for computer vision and machine learning, is often referred to as the Open-Source Computer Vision Library. It was developed by Intel, it is now maintained by a community of developers under the OpenCV Foundation. The extensive open-source library OpenCV is used for image processing, computer vision, and machine learning. Real-time operations are critical to the functioning of modern systems, and it plays a significant part in them. It may be used to verify photos and videos in order to recognize persons, objects, and even handwriting. When connected with various libraries, such as NumPy, it may do specific analyses by analysing the OpenCV array structure. It employs vector space and applies mathematical operations to these characteristics in order to identify an image pattern.

Version 1.0 of OpenCV was released. Due to its BSD license, OpenCV can be used for both commercial and academic purposes without charge. It runs on Linux, Windows, Andriod, iOS, and Mac OS and offers interfaces in C++, C, Python, and Java. Real-time applications for computing efficiency were the primary emphasis throughout its creation. Every piece of code is developed in optimized C/C++ to fully use multi-core processing.

OpenCV allows you to perform various operations in the image.

Read the Image: OpenCV helps you to read the image from file or directly from camera to make it accessible for further processing.

Image Enhancement: You will be able to enhance image by adjusting the brightness, sharpness or contract of the image. This is helpful to visualize quality of the image.

Object detection: As you can see in the below image object can also be detected by using OpenCV, Bracelet, watch, patterns, faces can be detected.

This can also include to recognize faces, shapes or even objects.

Image Filtering: You can change image by applying various filters such as blurring or Sharpening.

Draw the Image: OpenCV allows to draw text, lines and any shapes in the images.

Saving the Changed Images: After processing, you can save images that are being modified for future analysis

3.2.3 Speech recognition

Speech recognition, sometimes referred to as speech-to-text, is the process by which a computer program or machine can interpret spoken words into written language. Simple speech recognition can only understand clear words and phrases. But more advanced versions can understand natural speech, different accents, and languages.

Here's the working:

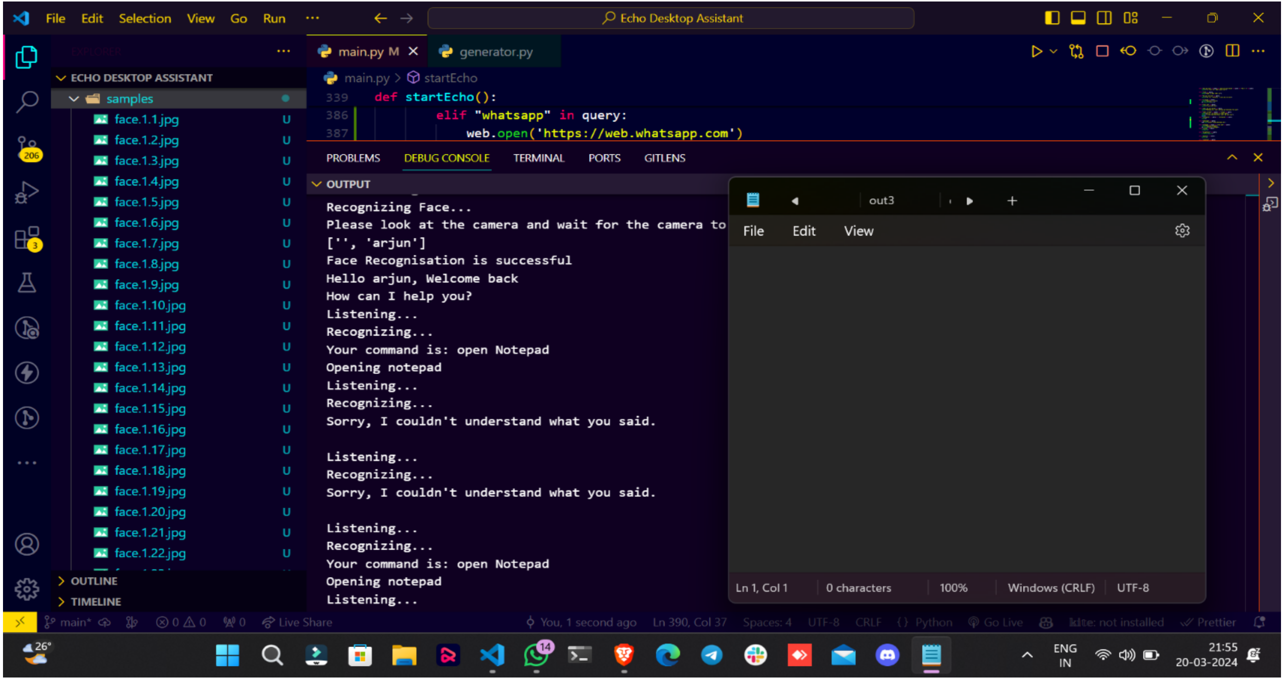
- First, the software listens to the audio.
- Then, it breaks the audio into smaller parts.
- Next, it turns those parts into a format that computers can read.
- Finally, it uses a special method to figure out the best written words to match what was said.

Speech recognition software has to be really good at understanding different ways people speak. It's trained on lots of different speech patterns, languages, accents, and styles. It also has to filter out background noise that might make it harder to understand what's being said.

Here are some of the modules that work with the system:

1. CV2: This module helps capture images using the camera.
2. Pyttsx3: It makes it easy to turn text into speech.
3. PyAutoGUI This module is for automating tasks in graphical user interfaces (GUIs). It can control the mouse, keyboard, take screenshots, and more.
4. Wolframalpha: It lets you interact with Wolfram Alpha, which is a smart engine for answering questions.
5. DateTime: This module deals with time and dates.
6. OS: It helps interact with the operating system, like managing files and folders.
7. IME: This module helps to display time.
8. Web Browser: It's a built-in package in Python that can extract data from the internet.
9. Subprocess: It's used to run system commands like shutting down or restarting the computer.

When the system hears the wake-up word, it gets activated. Depending on the command given, it fetches data from the specified location. The system can do things like adjusting volume, showing battery percentage, telling the date and time, and more.



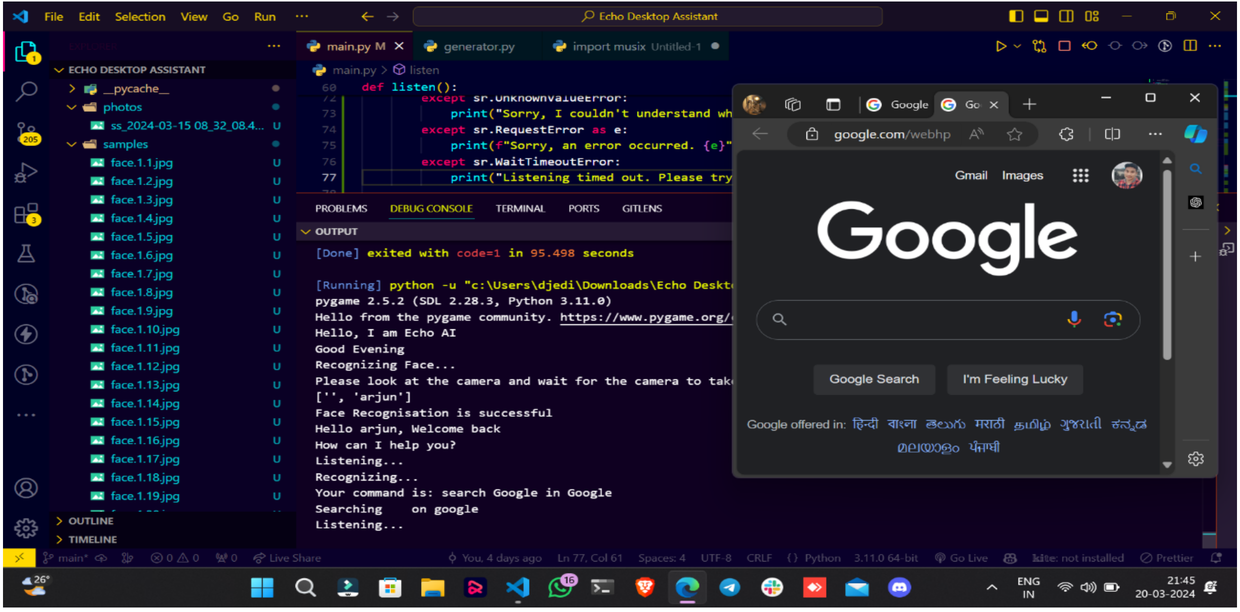
```
def startEcho():
        elif "whatsapp" in query:
            web.open('https://web.whatsapp.com')
```

```
Recognizing Face...
Please look at the camera and wait for the camera to
['', 'arjun']
Face Recognisation is successful
Hello arjun, Welcome back
How can I help you?
Listening...
Recognizing...
Your command is: open Notepad
Opening notepad
Listening...
Recognizing...
Sorry, I couldn't understand what you said.

Listening...
Recognizing...
Sorry, I couldn't understand what you said.

Listening...
Recognizing...
Your command is: open Notepad
Opening notepad
Listening...
```

Echo Desktop Assistant
main.py M
generator.py
ECHO DESKTOP ASSISTANT
samples
face.1.1.jpg … face.1.22.jpg
PROBLEMS DEBUG CONSOLE TERMINAL PORTS GITLENS
OUTPUT
out3
File Edit View
Ln 1, Col 1 0 characters 100% Windows (CRLF) UTF-8
OUTLINE
TIMELINE
You, 1 second ago Ln 390, Col 37
21:55 20-03-2024

Figure 3-Open Desktop Application.

The application is capable of doing a web-based search like Wikipedia, Google Chrome, Netflix, Pinterest, etc.



```
def listen():
            except sr.UnknownValueError:
                print("Sorry, I couldn't understand wh
            except sr.RequestError as e:
                print(f"Sorry, an error occurred. {e}"
            except sr.WaitTimeoutError:
                print("Listening timed out. Please try
```

```
[Done] exited with code=1 in 95.498 seconds

[Running] python -u "c:\Users\djedi\Downloads\Echo Deskt
pygame 2.5.2 (SDL 2.28.3, Python 3.11.0)
Hello from the pygame community. https://www.pygame.org/
Hello, I am Echo AI
Good Evening
Recognizing Face...
Please look at the camera and wait for the camera to tak
['', 'arjun']
Face Recognisation is successful
Hello arjun, Welcome back
How can I help you?
Listening...
Recognizing...
Your command is: search Google in Google
Searching    on google
Listening...
```

Figure 4-Opens Google Search.

4. CONCLUSION

In summary, the creation of an AI Desktop Assistant, amalgamating cutting-edge technologies such as GPT 3.5 from OpenAI and face recognition, represents a significant stride in virtual assistant development. Through the utilization of machine learning algorithms and natural language processing, this assistant facilitates tailored task automation and seamless interaction within the desktop environment. Integration of GPT 3.5 enhances user experience by furnishing contextually relevant responses. Moreover, the incorporation of face recognition bolsters system security, elevating overall user accessibility and convenience. This project effectively bridges the divide between existing virtual assistants and desktop functionalities, presenting a holistic solution for heightened productivity and user engagement in the digital landscape.

REFERENCES:

1. Gong, Q., Liu, Y., Kang, S., Ran, X., & Lin, X. (2023). Federated learning with confidential splitting: A privacy-preserving approach. arXiv preprint arXiv:2003.00229.

2. Handali, J., Bao, F., & Sun, N. (2020). A survey on open APIs for virtual assistants. Journal of Ambient Intelligence and Humanized Computing, 1-17.
3. Hassenzahl, M. (2013). User experience (UX) design. Morgan Kaufmann.Kohavi, R., Provost, F., Clark, R., & Spiegelhalter, D. (2014). Glossary of terms used in machine learning. Machine learning, 37(2), 301-329.
4. Lane, N. D., Georgiev, P., & Forlivesi, C. (2017). A survey on mobile phone sensing. IEEE Communications Surveys & Tutorials, 19(3), 1489-1507.
5. Lazar, J., Lazar, A., & Feng, J. H. (2007). Research priorities for web accessibility evaluation tools. In Proceedings of the 19th international conference on World Wide Web (WWW '10) (pp. 885-894). ACM.
6. Li, S., Li, J., Gao, L., Chen, Y., & Zhao, B. Y. (2020). Differential privacy for recommender systems: An overview. arXiv preprint arXiv:2003.10434.
7. Liu, B., Xu, K., & Zhao, Y. (2016). Evaluation metrics for dialogue systems. In Proceedings of the COLING 2016 workshop on dialogue systems (Vol. 1, pp. 90-98).
8. Mehri, S., Liu, Q., Caruana, R., & Devlin, J. (2023). Decoding with Transformers for Neural Conversational Machine Translation. arXiv preprint arXiv:2301.08237.
9. Menezes, A., van Oorschot, P. C., & Vanstone, S. A. (2010). Handbook of applied cryptography. CRC press.
10. Ohm, P. (2010). Broken promises of privacy: Designing the future of secrecy. Harvard University Press.
11. Rabiner, L. R., & Juang, B. H. (1993). Fundamentals of speech recognition. Prentice Hall PTR.
12. Rogers, Y. (2014). Design revolution: Solving human problems through design thinking. Harvard Business Review Press.
13. Russell, S. J., & Norvig, P. (2021). Artificial intelligence: A modern approach (4th ed.). Pearson Education Limited.
14. Sainath, T. N., Vaitheeswaran, V., Kingsbury, B., Saurous, R., Mohamed, A., Hinton, G., & Kingsford-Smith, F. (2015). Deep convolutional neural networks for LVCSR. arXiv preprint arXiv:1504.04830.
15. Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Advances in neural information processing systems (pp. 3104-3112).
16. Walker, M. A., Aberdeen, J., Kamm, V. A., & Sussman, R. A. (2002). Evaluations in speech recognition. Speech communication, 37(1-2), 7-24.
17. Wang, Y., Ding, X., & Deng, S. (2020). Deep learning for face recognition: A survey. arXiv preprint arXiv:2004.12315.
18. Ward, N., Coventry, L., & Loh, J. (2019). A comparative analysis of user acceptance of virtual personal assistants. Interacting with Computers, 31(6), 713-731.
19. Wu, Z., Zheng, S., Chen, E., & Yang, D. (2023). BART as a query-driven generative model for extreme summarization. arXiv preprint arXiv:2301.12027.