Digital Accessibility Using Local Large Language Models

Bhanu Arya
The University of Texas
Austin, Texas, USA
bhanuarya@utexas.edu

ABSTRACT

Digital accessibility ensures that all individuals, including those with disabilities, can use digital content and platforms. This paper explores a novel system that leverages locally hosted large language models (LLMs) and vision-language models to enhance accessibility. Our approach combines speech-to-text, screen capture, LLM processing, and text-to-speech to create an intelligent, privacy-focused accessibility assistant. Unlike traditional screen readers, our system can understand, simplify, and explain on-screen content, providing context-aware narration for users with visual and cognitive impairments. By deploying quantized models locally, we ensure data privacy while maintaining performance on CPU-only systems. We present implementation details, evaluation results across diverse digital content, and future directions for improving accessibility through AI assistance.

KEYWORDS

digital accessibility, large language models, vision-language models, screen readers, privacy-preserving AI

ACM Reference Format:
Bhanu Arya. 2025. Digital Accessibility Using Local Large Language Models. In . ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Digital accessibility refers to the inclusive practice of making digital content, platforms, and tools usable by all individuals, including those with disabilities. This encompasses a broad range of considerations, such as providing alternative text for images, ensuring video content has captions or transcripts, and creating navigation systems operable by keyboard or screen readers. The goal is to ensure equitable access to digital resources, aligning with broader objectives of diversity, equity, and inclusion [1].

Visual accessibility supports users with vision impairments, ranging from mild vision loss to complete blindness. It also covers color blindness and low vision, which affect how users perceive and interact with digital content. As of 2020, approximately 1.1 billion people worldwide had some form of vision impairment, a number projected to rise to 1.8 billion by 2050 due to aging populations and lifestyle changes [3]. Cognitive impairments, too, affect a substantial portion of the population, with global prevalence rates ranging from 5.1% to 41%, depending on region and age group.

To address these barriers, the World Wide Web Consortium (W3C) developed the Web Content Accessibility Guidelines (WCAG) to establish standards for making Web content more accessible. The latest version, WCAG 2.2, introduces new success criteria organized into three conformance levels: A (minimum), AA (midrange), and AAA (highest) [5]. These guidelines aim to ensure content is perceivable, operable, understandable, and robust. However, real-world implementation often falls short, and many websites remain partially or entirely inaccessible due to poor design, inconsistent enforcement, or lack of awareness.

Screen readers, widely used for accessibility, convert on-screen text into synthesized speech or Braille output. While effective in many scenarios, they have notable limitations: difficulty interpreting complex layouts, an inability to describe images without alternative text, and a steep learning curve requiring memorization of intricate navigation commands. Furthermore, screen readers can hinder the user experience; for example, reading a one-page document (300 to 500 words) typically takes 1.5 to 3 minutes [6]. In addition, inaccessible websites and applications further limit their effectiveness.

Recent advances in Large Language Models (LLMs) and vision-language models offer promising solutions. LLMs can understand and generate natural language, while vision-language models interpret images and extract contextual meaning. When deployed locally using quantization, these models improve privacy by ensuring that all data remain on the user's device [7]. Integrating LLMs with tools like speech-to-text, screen capture, and text-to-speech technologies enables the creation of intelligent pipelines that can simplify, explain, and narrate on-screen content, transforming traditional screen readers into context-aware, AI-assisted accessibility companions.

This project explores the development of such a system using a local LLM to interpret screen content (e.g., web pages or PDFs), simplify or explain it, and provide output through speech. The goal is to bridge the gap between static content and accessible comprehension, particularly for users with visual and cognitive impairments.

2 RELATED WORK

2.1 AI-Driven Accessibility Review and Gaps

Chemnad and Othman (2024) conducted a systematic review of 43 academic articles published between 2018 and 2023, categorizing AI-based accessibility solutions [8]. Examples include AI-Vision smart glasses, which recognize facial expressions and colors, and tactile gloves with webcam systems, allowing users to perceive

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Conference'17, July 2017, Washington, DC, USA
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

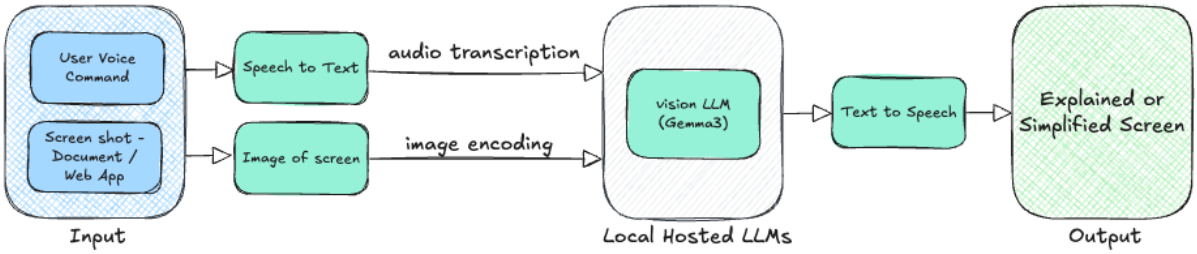

Figure 1: System architecture showing the flow from user input (voice commands or screen captures) through speech-to-text processing, LLM interpretation, and text-to-speech output.

environments through tactile feedback guided by computer vision. However, the review emphasized a gap in simplifying complex textual information, particularly for web or document formats. Cognitive accessibility remains underrepresented, and reliance on cloud-based AI introduces privacy concerns—an issue locally deployed LLMs can address.

2.2 Human-Centered AI Accessibility Applications

Borunda et al. (2024) developed an Android prototype, SmartGuide, integrating YOLOv7, DepthAnything-V2, Blip2, and LLMs such as GPT-4 and LLaMA to provide real-time scene understanding and audio descriptions [9]. Unlike generic outputs, their system generated rich, contextual descriptions (e.g., describing an office waiting area rather than merely stating "a woman with dark hair"). Participatory design workshops with visually impaired users highlighted the demand for accurate, user-centered multimodal interaction—principles that guide the present project's design.

2.3 AI-Assisted Content Simplification and Text Interpretation

Chemnad and Othman also noted developments in automated braille caption generation and summarization of social media content for accessibility. However, most systems were domain-specific and lacked general-purpose tools capable of comprehending arbitrary digital content such as scientific papers, news articles, or complex PDFs. This creates an opportunity for systems like ours, which integrate STT, LLMs, and TTS into a seamless, privacy-respecting accessibility pipeline.

3 METHODOLOGY

Our project introduces a Python-based accessibility assistant that uses locally hosted large language models (LLMs) to simplify or explain screen content through audio. The system, illustrated in Figure 1, is designed for users with visual or cognitive impairments and operates entirely offline.

3.1 Input Processing

Users initiate the system either through a voice command or by capturing a screenshot of their current screen:

Table 1: Technology Stack

Component	Technology
Programming Language	Python
Frontend UI	Streamlit
Local Model Host	Ollama
Vision-Language Model	Gemma3 (4-bit)
Speech-to-Text	SpeechRecognition
Text-to-Speech	pyttsx3

- **Voice Command:** Captured using a microphone and converted to text via a Speech-to-Text (STT) engine. Commands like "simplify this screen" guide LLM behavior.
- **Screen Capture:** Screenshots are base64-encoded and processed by Gemma3, a vision-language model, to generate contextual natural language descriptions.

3.2 LLM Processing

For privacy, all models are deployed locally using the Ollama framework. A 4-bit quantized version of Gemma 4B is used, minimizing memory usage and enabling CPU-only operation.

Gemma 3 models, developed by Google based on Gemini technology, are multimodal, support 140+ languages, and feature a 128K token context window. For this project, the Gemma 4B variant was chosen for its balance between performance and resource efficiency.

3.3 Output Generation

The LLM-generated text is converted into speech using pyttsx3, an open-source Text-to-Speech (TTS) engine. The end-to-end interaction is packaged within a Streamlit-based frontend for ease of use and real-time feedback.

4 RESULTS

The prototype system was evaluated across diverse digital content types, including web pages, PDFs, and application interfaces. All components functioned offline, ensuring complete user privacy.

4.1 Functional Highlights

- **LLM Responsiveness:** Gemma-3 (4-bit) processed content and generated summaries within 30–40 seconds on CPU-only hardware, supporting near real-time interaction.

- **Visual Content Interpretation:** Successfully generated natural language descriptions of complex scenes.
- **Text-to-Speech Output:** Speech generation using pyttsx3 ensured fluid and intelligible communication.

4.2 Use Case Scenarios

- **Web Accessibility:** For dense scientific webpages, the system summarized content, e.g.,
  "This article explains how cholesterol affects the heart. It says higher levels increase risk, especially in older people."
- **Image-Based Description:** For UI dashboards, the system described:
  "A dashboard with buttons labeled 'Start', 'Settings', 'Help', and 'Exit', and a status bar showing system uptime."

4.3 System Strengths

- **Privacy:** All processing remains local, eliminating cloud dependencies.
- **Low Cost:** Functions efficiently on CPU-only systems without GPU requirements.
- **Offline Operation:** Fully functional without internet connectivity.
- **Data Flexibility:** Supports input from text, images, and voice.

4.4 Limitations

- **Speech Recognition Noise:** Noisy environments affected transcription accuracy.
- **Visual Interpretation Challenges:** Occasional over-description or misinterpretation of complex visual layouts.
- **Synthetic Voice Quality:** The TTS engine produced speech perceived as less natural.
- **Limited Multilingual and Dialogue Support:** Currently monolingual and single-turn.
- **No Feedback Mechanism:** Lacks user correction integration for continuous improvement.

5 FUTURE WORK

To enhance the system's usability and effectiveness, the following improvements are proposed:

- **Interactive Dialogue:** Enable conversational interactions with the LLM for deeper exploration.
- **Multilingual Support:** Extend capabilities to multiple languages.
- **User Feedback Integration:** Capture and incorporate user corrections to fine-tune responses.
- **Enhanced OCR and Layout Detection:** Integrate advanced tools such as PaddleOCR for better visual parsing.
- **Mobile or Browser Extension Deployment:** Package the assistant as a browser add-on or mobile app.
- **Real-Time Screen Monitoring:** Allow proactive content detection and narration.
- **Ethical Considerations:** Investigate and mitigate potential biases in LLM-generated descriptions.

6 CONCLUSION

This project demonstrates the feasibility of combining locally hosted large language models with speech and image processing tools to build an intelligent, privacy-focused accessibility assistant. By capturing and interpreting on-screen content, and responding via simplified speech, the system serves as a context-aware, AI-powered alternative to traditional screen readers.

Unlike linear text narrators, this solution enables customized comprehension through summarization, explanation, and visual scene description, greatly improving accessibility for users with visual or cognitive impairments. The deployment of quantized LLMs enables lightweight, CPU-only operation, ensuring resource efficiency and strict user privacy.

The source code is available at GitHub Repository:
https://github.com/utbhanuarya/HighRiskProject.

REFERENCES

[1] Digital Accessibility - Oxford Review
[2] Disability Types - Yale Usability Guide
[3] Vision Loss Statistics - Statista
[4] Cognitive Impairment Study - PMC
[5] WCAG 2.2 Guidelines - W3C
[6] Introduction to Screen Readers - AbilityNet
[7] Transparency and Accessibility of LLMs - arXiv
[8] Chemnad, K., & Othman, A. (2024). Digital accessibility in the era of artificial intelligence—Bibliometric analysis and systematic review. Frontiers in Artificial Intelligence, 7:1349668. https://doi.org/10.3389/frai.2024.1349668
[9] Borunda, L., Gipe-Lazarou, A., & Meng, N. (2024). "I WANT": Agency and Accessibility in the Age of AI. ACSA 2024 International Conference: Inflections. Querétaro, Mexico.