DataGuardian: AI-Powered Multi-Modal Visual and Textual Data Anonymization System

Nikita B Emberi, Jason Yoo, Karamjeet Singh Gulati

1 Introduction

1.1 Objective and Problem Statement

In today's era of heightened digital surveillance, protecting personally identifiable information (PII) has become a critical challenge. DataGuardian is an AI-powered system designed for real-time anonymization of multi-modal data, addressing the need to protect both visual and textual information while maintaining data utility. Unlike traditional solutions that handle modalities separately, our system integrates state-of-the-art vision models (GPT-4 Vision) with advanced natural language processing (NLP) techniques for seamless, context-aware anonymization.

The primary challenges addressed include:

- **Multi-Modal Integration**: Combining visual anonymization (e.g., face blurring) with text redaction for comprehensive privacy protection.
- **Real-Time Processing**: Ensuring instantaneous anonymization with a response timeout of 700ms, optimized for performance.
- **Privacy-Utility Balance**: Configurable anonymization operators maintain usability, masking personal information, financial data, contact details, and location data with placeholders.

1.2 Research Motivation and Innovation

Current privacy solutions suffer from significant technical gaps, including limited language support and insufficient recognition capabilities. DataGuardian overcomes these limitations with multilingual NLP support for English, Spanish, French, German, Russian, and Dutch, alongside a universal fallback model. Its multi-layered recognition system employs pattern-based recognition, context-aware enhancements, and adaptive confidence scoring for improved accuracy.

This research introduces several innovations:

- **Unified Framework**: Integrates visual and textual anonymization into a cohesive system.
- **Real-Time Processing**: Optimized pipelines for concurrent entity recognition and anonymization.
- **Extensible Architecture**: Modular design supporting updates to recognition patterns, custom operators, and language models.
- **User-Centric Interface**: Interactive Gradio-based interface featuring real-time webcam processing and immediate anonymization feedback.

1.3 Applications

DataGuardian has broad applications in:

- **Healthcare**: Anonymizing medical imaging and patient records.
- **Social Media**: Ensuring privacy in content moderation.
- **Document Processing**: Securing sensitive business documents.
- **Research**: Facilitating privacy-compliant data sharing.

2 Methodology

2.1 Data Processing

The DataGuardian system employs a robust and efficient framework for real-time data processing and anonymization, seamlessly integrating multiple components to ensure reliable performance. Visual data is processed through a structured pipeline, beginning with an input capture mechanism that leverages Gradio's streaming API to enable real-time frame capture from webcams. The system supports native resolution with a default frame rate of 30 FPS. The image processing pipeline incorporates key steps such as RGB-to-BGR color space transformation, resolution standardization, orientation correction using NumPy, and lossless JPEG compression, ensuring high-quality output. Processed images are systematically stored using a UUID-based naming convention within a hierarchical directory structure, supported by automated cleanup protocols and robust error-handling mechanisms to maintain operational stability.

Text analysis is powered by natural language processing (NLP) and supports multiple languages, including English (*en_core_web_lg*), Spanish (*es_core_news_md*), French (*fr_core_news_sm*), German (*de_core_news_sm*), Russian (*ru_core_news_sm*), Dutch (*nl_core_news_sm*), and a universal fallback model (*xx_sent_ud_sm*). This framework ensures robust multi-language capabilities suitable for diverse use cases.

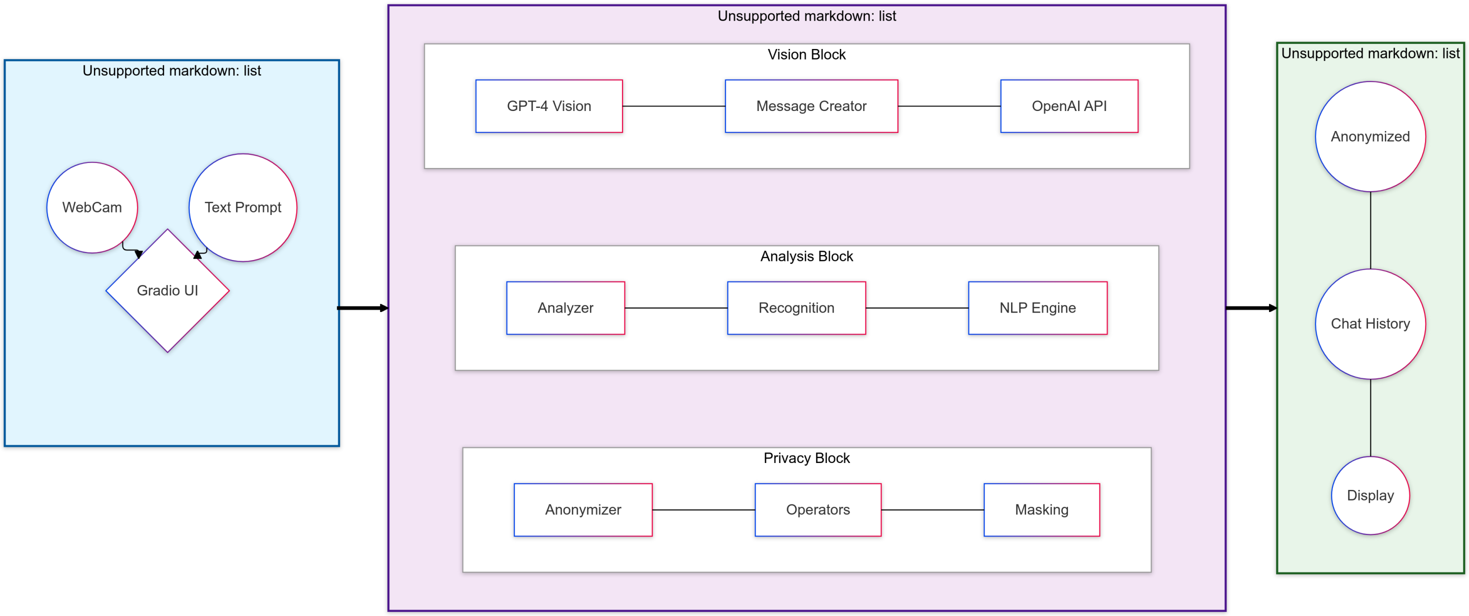

Figure 1: AI-Supported Visual Identification & Data Anonymization Architecture: Three-stage pipeline showing data flow from input capture through AI processing to anonymized output. The system integrates GPT-4 Vision capabilities with Presidio-based anonymization in three main blocks: input handling, core processing, and output management.

2.2 Architecture

The proposed architecture implements a privacy-preserving visual recognition system through a streamlined three-stage pipeline, as illustrated in Figure 1. At its core, the system orchestrates real-time data processing while maintaining strict privacy controls through advanced anonymization techniques.

The Input Layer establishes the foundation of the pipeline through a Gradio-based interface, managing concurrent streams of webcam captures and text prompts. This initial stage preprocesses the multimodal inputs, preparing them for subsequent analysis while maintaining data integrity and format consistency.

The Output Layer culminates the pipeline by orchestrating the presentation of processed data through a managed chat interface. This stage ensures all displayed information adheres to privacy requirements while maintaining conversational context. The architecture's modular design facilitates straightforward maintenance and allows for future extensions of recognition capabilities or anonymization rules.

This implementation demonstrates the practical integration of advanced AI vision capabilities with privacy-preserving technologies, offering a robust foundation for sensitive data handling in visual recognition systems. The architecture's emphasis on modularity and privacy preservation makes it particularly suitable for applications requiring real-time processing of sensitive visual and textual content.

2.3 Processing Pipeline and Output Generation

The processing pipeline is designed to integrate several key modules to handle visual and textual data effectively. The image processing module, implemented in Python, would convert color spaces, correct orientations, and store files using error-resilient mechanisms. Vision models, such as GPT-4 Vision, would be integrated with optimized configurations, including a token limit of 500, a low temperature (0.1) for focused outputs, and advanced error-handling strategies like timeout management and fallback mechanisms.

Entity recognition would employ specialized recognizers to identify patterns, such as phone numbers, emails, URLs, and personal names, with a confidence threshold of 0.2 and context enhancement for improved accuracy. Anonymization would be achieved through operator configurations that replace sensitive entities with generic placeholders (e.g., `-MASKED_EMAIL_RELATED-`).

Processed outputs would be formatted as JSON or HTML and delivered through an interactive interface that would provide real-time feedback, status indicators, and a chat functionality. The system would ensure data integrity by maintaining chat history, tracking image references, and cleaning temporary files after processing.

3 Current Progress

Our team has made steady progress in developing the DataGuardian system, working on key components to enhance functionality. We have implemented a user interface using Gradio, which includes real-time webcam integration for video capture, a responsive chat interface supporting multi-line input, and a chat history display. These features have been developed through collaborative efforts, aiming to create a straightforward and user-friendly experience.

On the backend, we have integrated GPT-4 Vision for visual data analysis, configuring the API for basic functionality and adding response handling to manage common scenarios. The image processing pipeline, developed incrementally, now includes modules for color space conversion, orientation correction, and secure storage using a UUID-based file naming system. These components represent our team's ongoing work toward building a functional and efficient system.

Privacy protection mechanisms are in place, utilizing entity recognition to identify and anonymize sensitive information such as numbers, credit cards, emails, URLs, phone numbers, and personal names. Anonymization rules replace identified entities with placeholders like `-MASKED_PERSON_RELATED-` and `-MASKED_EMAIL_RELATED-`, ensuring data privacy while maintaining structure.

Multi-language support has been successfully implemented with models for English, Spanish, French, German, Russian, and Dutch. Core system parameters, including a maximum token limit of 500, low temperature (0.1) for focused outputs, and penalties for diverse responses, have been configured to optimize processing. The server operates locally with up to 10 threads, ensuring stability and scalability.

Integration testing has confirmed the functionality of webcam capture, real-time image processing, and the file management system. The file management system successfully organizes temporary storage, configuration files, and processed results in a structured directory hierarchy. Pending tasks include implementing anonymization, chat responsiveness, multi-language processing, enhanced error handling, response time optimization, extended language support, additional entity recognition patterns, user authentication, and advanced privacy configuration options.

4 Current Results

The system was tested on a local server (`127.0.0.1:8800`) with a maximum of 10 threads, real-time processing enabled, and active error tracking. Key testing parameters included a 700ms response time threshold, memory usage monitoring, entity recognition accuracy, and multi-language processing validation.

The implemented system features a web-based interface built using the Gradio framework, which serves as the primary interaction point for users. The interface has been designed with a dual-panel layout: a webcam input section on the left and an interactive chat interface on the right. The webcam panel enables real-time video streaming and frame capture, with images processed through OpenCV before being sent to GPT-4 Vision for analysis. This visual data pipeline is intended to ensure efficient handling of input data while maintaining high-quality image processing standards. However, due to current system errors, we have not yet been able to demonstrate the fully functional implementation.

The chat interface component has been developed to implement a robust communication system where users can interact with the AI model through text inputs. Responses from the GPT-4 Vision model are designed to pass through a comprehensive anonymization pipeline powered by Microsoft's Presidio framework. This pipeline detects sensitive information such as personal names, email addresses, phone numbers, and other identifiable data, replacing them with placeholders like `-MASKED_PERSON_RELATED-`. While this feature has been coded, errors in the current implementation prevent us from providing proof of its functionality in the form of screenshots at this time. The interface is designed to run on a local server (`localhost:8800`) with WebSocket connections enabling real-time updates and smooth interaction between components.

The system architecture follows a modular design pattern, enabling independent operation of components while ensuring seamless integration. Using Python's advanced libraries such as OpenCV for image processing, Presidio for data anonymization, and OpenAI's GPT-4 Vision API for intelligent analysis, the system is designed to provide real-time processing capabilities with a focus on privacy and security. Further testing and refinements are ongoing to address current errors and achieve full operational efficiency.

Testing involved:

- **Visual Processing**: Validated frame capture, image quality, color conversion, and storage efficiency.
- **Entity Recognition**: Assessment of personal, numerical, and contact information detection
- **Anonymization**: Preliminary validation of anonymization patterns, with error resolution still in progress
- **Ongoing Development**: Most of the code is complete, with additional refinements and optimization in progress.

5 Next Steps

The focus remains on finalizing core functionalities, optimizing performance, and preparing for system scalability. Key tasks include:

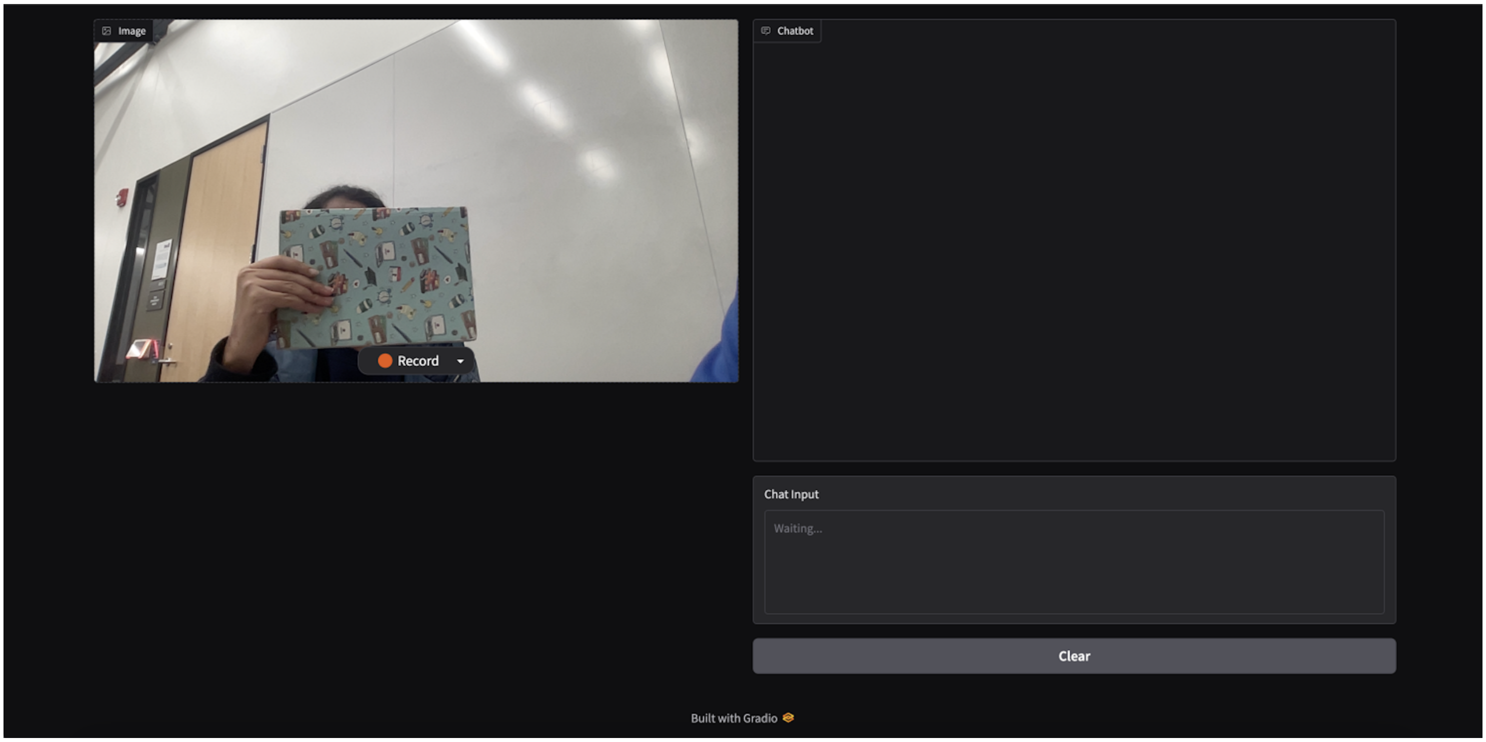

Figure 2: Gradio Web Interface for AI-Powered Visual Identification & Data Anonymization System

5.1 Development Completion

Implement Gaussian blur to enhance image anonymization and finalize context-aware entity recognition. Extend the system's capabilities by adding new recognition patterns.

5.2 Performance Optimization

Optimize image compression and memory management to reduce latency and address memory leaks. Improve thread pooling for concurrency and enhance error recovery to handle edge cases.

5.3 System Expansion

Integrate additional language models, introduce custom entity definitions, and implement advanced privacy rules. Add user authentication to secure system access.

5.4 Interface Enhancements

Develop real-time statistics visualization and a configuration interface. Enable batch processing and improve anonymized data presentation for better usability.

5.5 Long-Term Vision

Incorporate machine learning for automated rule generation, privacy policy automation for GDPR/HIPAA compliance, and cloud deployment for enterprise integration. Research advanced context-aware privacy techniques to further enhance anonymization.

6 Team Membership and Attestation of Work

Karamjeet Singh Gulati, Nikita B. Emberi, and Jason Yoo have contributed to the project's progress.