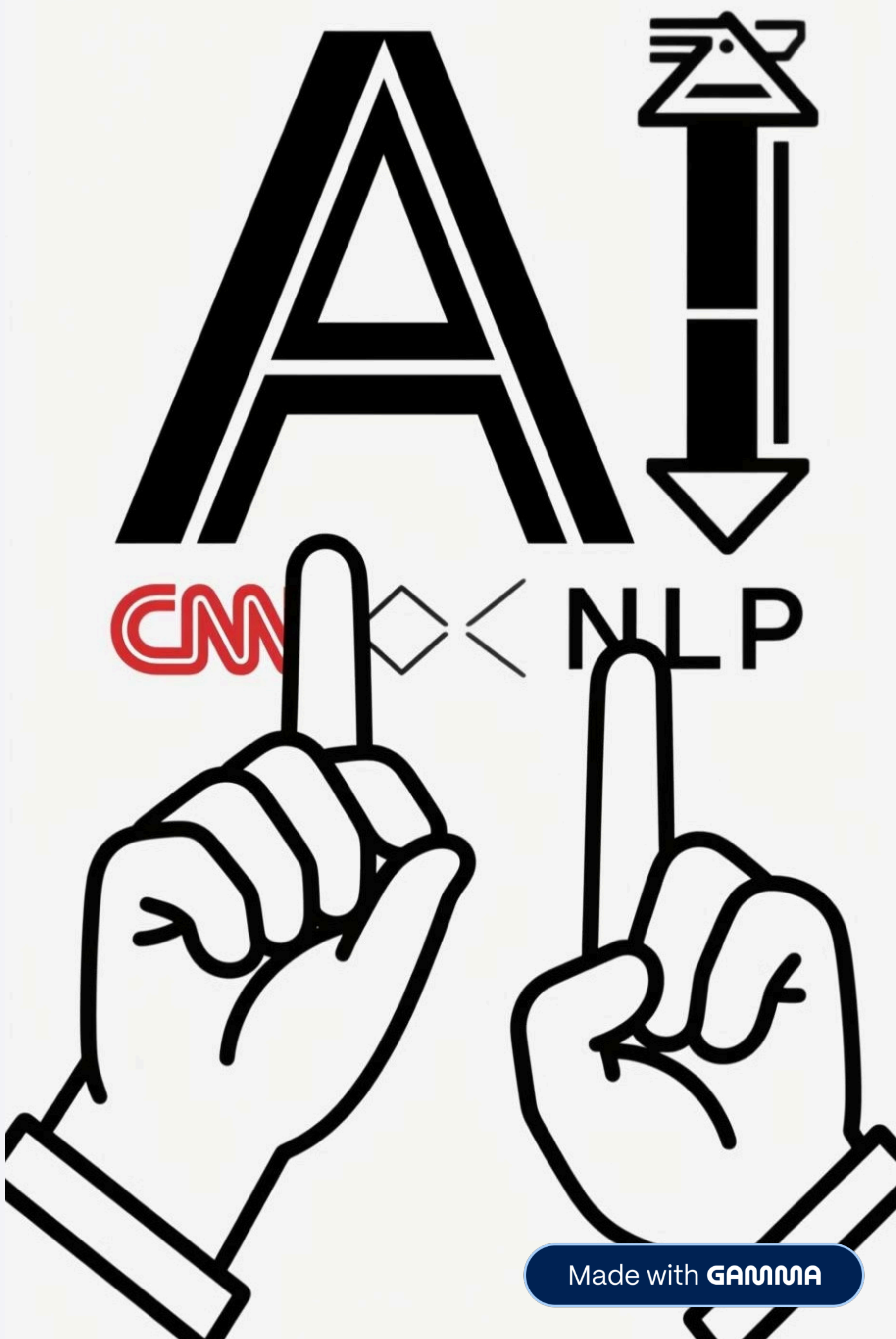
AI-Based ASL Interpreter

Using CNN Classification and NLP Naturalization for a Terminal-Based Prototype



Introduction: Bridging Communication Gaps

American Sign Language (ASL) is vital for millions, yet communication barriers persist. AI systems translating sign language into natural English text can significantly reduce these barriers.

This project presents a terminal-based ASL translation pipeline. It integrates computer vision (CV) for handshape recognition with natural language processing (NLP) for smoothing raw letter sequences into grammatically coherent English. The prototype focuses on the ASL alphabet (fingerspelling) using a Convolutional Neural Network (CNN) to classify static gesture images.

Problem Definition: The Need for Accessible ASL Tools

Traditional ASL interpretation relies on human interpreters, who are not always available. Many assistive tools are complex, focusing on full-body pose estimation or mobile apps.

Fingerspelling, using static handshapes for letters, offers a simpler domain for effective computer vision models. Even a terminal-based model converting gestures to text provides meaningful accessibility benefits for:

- Early-stage ASL learners
- Deaf individuals communicating with non-signers
- Researchers studying gesture recognition
- Developers building larger sign-language systems

Importance and Relevance: Why This Project Matters

Accessibility

Provides an automated ASL-to-English tool, supporting equitable communication.

AI Integration

Demonstrates how CV and NLP can work together in a full pipeline.

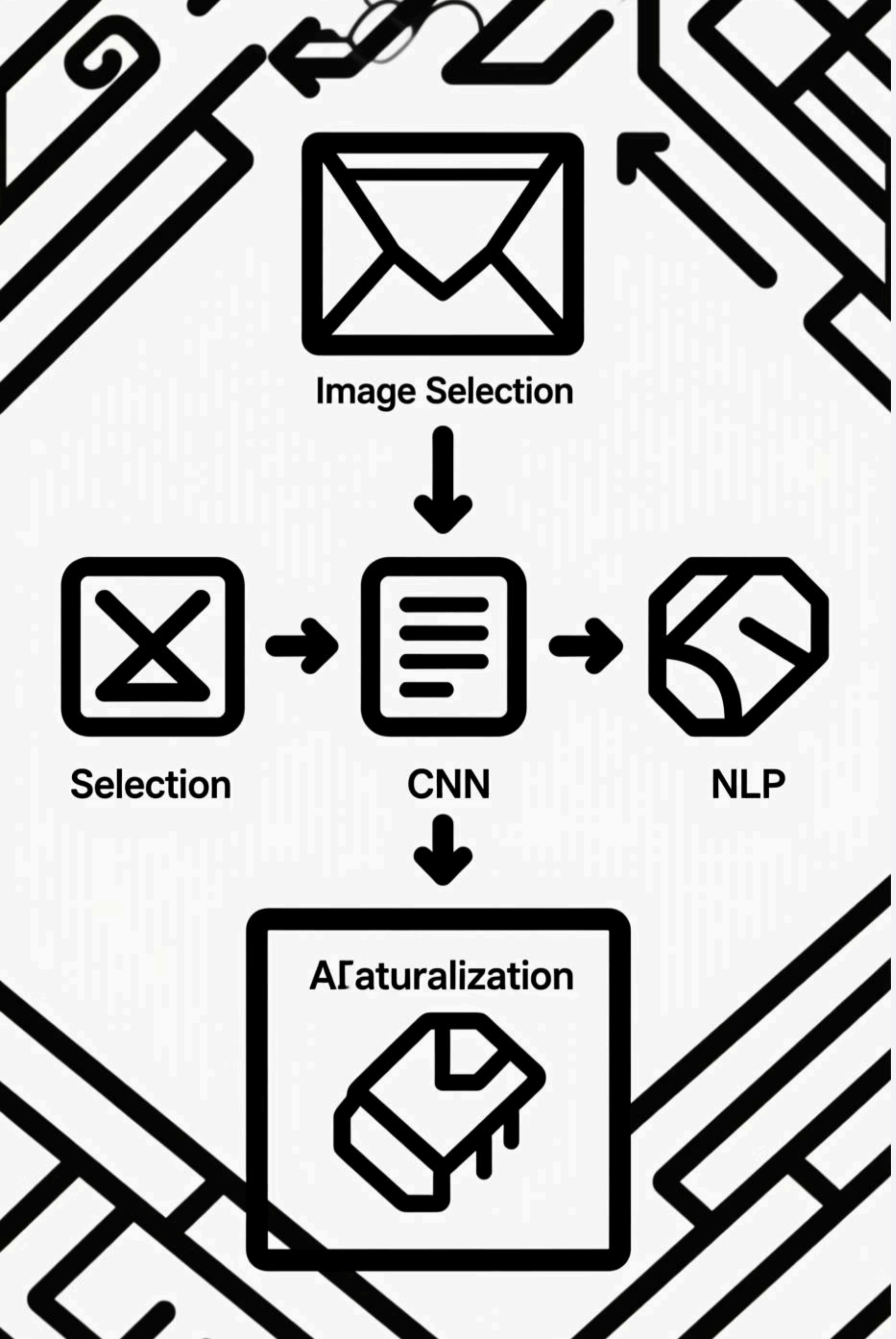
Practical Feasibility

Focuses on alphabetic signs, making the project achievable within a semester.

Course Alignment

Satisfies CSCI 402 requirements by implementing an AI model, using real data, and analyzing results.

The terminal-only interface avoids GUI complexity while delivering a functional AI system.



AI Methodology: System Overview

The system follows a three-stage pipeline:

Image Selection

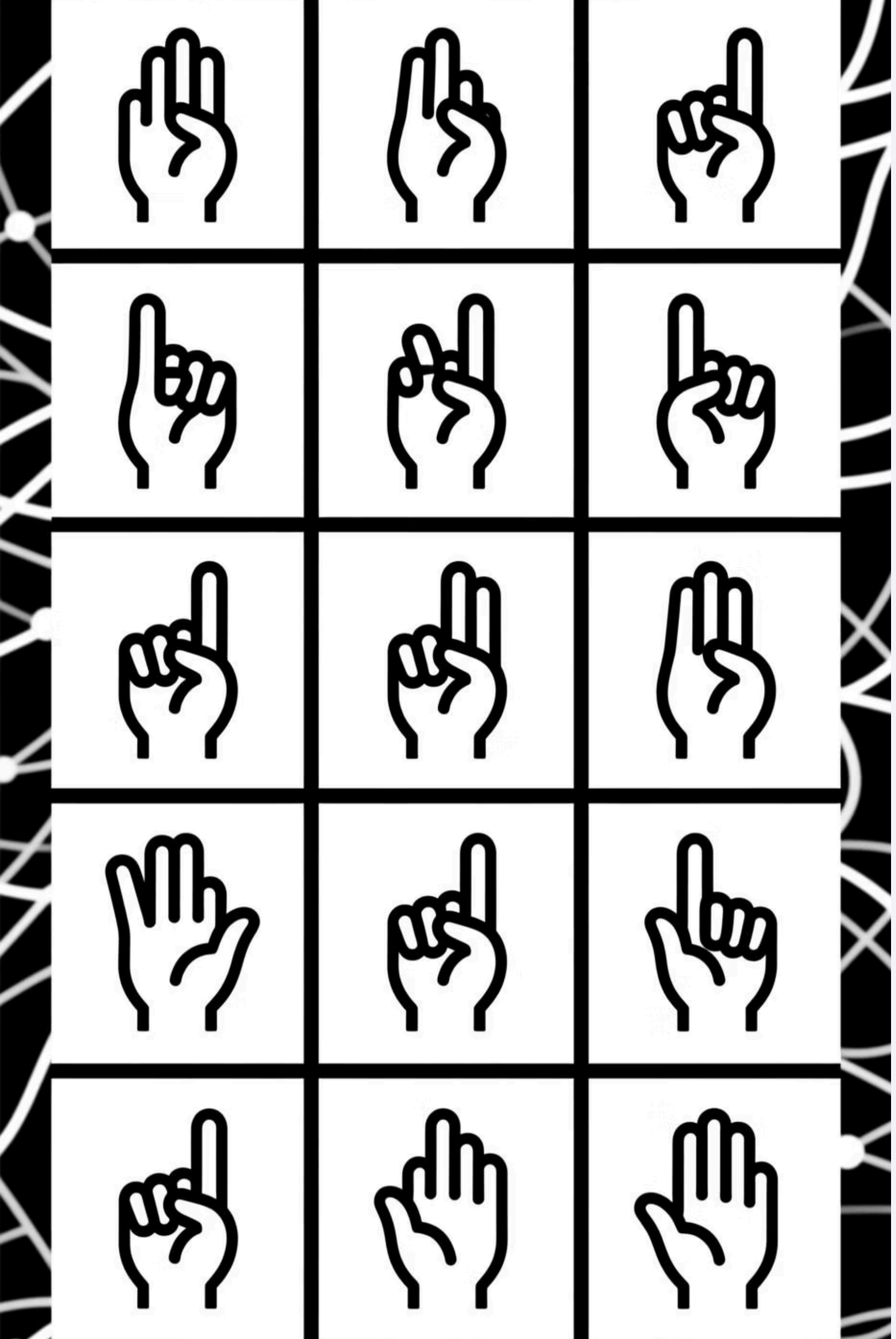
A script extracts labeled ASL images from the Sign Language MNIST dataset.

CNN-Based Gesture Recognition

A TensorFlow CNN model classifies each input image as one of 24 ASL alphabet classes (A–Y, excluding J and Z).

NLP Naturalization

A GPT-based NLP module converts raw letter sequences into coherent English, correcting spacing, capitalization, and grammar.



Dataset and CNN Architecture

Dataset: Sign Language MNIST

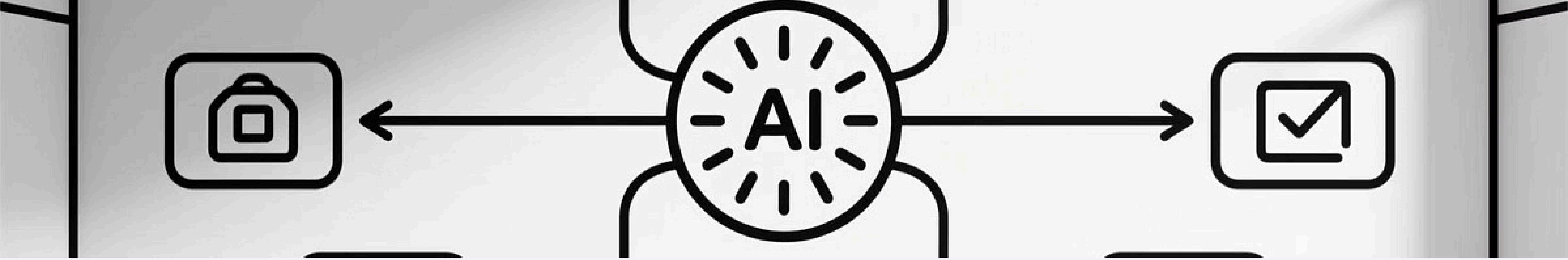
- 28×28 grayscale images
- 27,455 training samples
- 7,172 test samples
- Labels mapped to 24 letters (J & Z excluded due to motion)

This dataset is ideal for CNN training due to its consistency and image size.

CNN Model Structure

- Normalization layer (scales pixel intensities)
- Two convolutional blocks (Conv2D + ReLU + MaxPooling)
- Dropout (reduces overfitting)
- Dense layers (including a 128-unit hidden layer)
- Softmax output (predicts class probabilities for 24 letters)

Trained with Sparse categorical cross-entropy, Adam optimizer, 10–20 epochs, and a batch size of 32.



Inference Pipeline and NLP Integration

The inference process is handled by `infer_and_nlp.py`, which loads the trained CNN, processes test images, and predicts letters. A per-image confidence threshold is applied, forming raw sequences like "ITHANKU".

Load CNN & Images

Loads the trained CNN model and selected test images.

Predict Letters

Classifies each image to predict its corresponding ASL letter.

Apply Confidence

Filters predictions based on a confidence threshold (e.g., --min_conf).

Form Raw Sequence

Combines predicted letters into a raw sequence (e.g., ITHANKU).

If an OpenAI key is provided, a GPT-based NLP module converts this raw sequence into coherent English, demonstrating the power of integrating deep learning with generative AI.

Results: High Accuracy and Naturalization

CNN Classification Performance

After training on Sign Language MNIST:

- Training accuracy: ~98–99%
- Test accuracy: ~94–96%

This confirms the model generalizes well. Common errors include confusions between visually similar letters (A vs S, N vs M).

Inference Demonstration: "I THANK YOU"

Using selected test images, the system successfully predicted the sequence "ITHANKU".

I	0.9999
T	0.6865
H	0.9467
A	0.9999
N	0.9954
K	0.9276
U	0.9991

The GPT module then correctly naturalized "ITHANKU" to "I THANK YOU."

Strengths, Limitations, and Future Work

Strengths

- End-to-end AI pipeline (CV → text → natural language)
- High letter-level accuracy
- Consistent phrase interpretation
- Simple terminal interface
- Modular code structure

Limitations

- Fingerspelling only (no full ASL grammar)
- No real-time camera input yet
- Dependent on Sign Language MNIST variability
- GPT output requires API key

Future Work

- Expand to dynamic letters (J/Z) using video CNN
- Add webcam inference and mobile app UI
- Incorporate ASL grammar rules
- Train on real ASL datasets beyond MNIST

Conclusion: A Foundation for AI-Driven Accessibility

This project successfully demonstrates a working terminal-based ASL interpreter, integrating a CNN model for handshape recognition, a letter-sequence reconstruction pipeline, and a GPT-based NLP module for fluent English translation.

Testing results, including the "I THANK YOU" sample, show reliable performance on the Sign Language MNIST test distribution and the ability to generate natural English. This work highlights the integration of computer vision and natural language processing in an accessible, lightweight AI system, laying a strong foundation for more advanced ASL-to-speech systems and AI-driven accessibility technology.