رواد مصر الرقمية

Microsoft Machine Learning

Final Project

Hand Gesture Recognition for Arabic Sign Language to Text

CAI2_AIS2_S4

Team Members:

- Seif Sherif Assad Ali
- Sama Ahmed ElSayed
- Yusuf Sobhy Sadek Elmeligy
- George Nemr Mellek Poqtor
- Omar Elsayed Elsayed Mousa

Table of Contents

1. Introduction 3

1.1 Project Overview 3

1.2 Motivation 3

2. Project Scope & Features 3

2.1 Dataset Selection & Preprocessing 3

2.2 Model Development & Training 3

2.4 Future Expansion & Enhancements 3

3. Techniques Used 4

4. Model Exploration 4

5. UI & Deployment 5

6. Implementation Plan 5

6.1 Project Timeline & Phases 5

7. Technical Implementation 6

7.1 Tools & Libraries 6

7.2 Model Architecture 6

7.3 Performance Metrics 6

8. Expected Outcomes & Future Work 7

8.1 Expected Outcomes 7

8.2 Future Work 7

9. Conclusion 7

1. Introduction

1.1 Project Overview

This project focuses on developing an AI-based Hand Gesture Recognition System that translates Arabic Sign Language (ArSL) into text using computer vision and deep learning techniques. With the increasing need for accessible communication tools, our goal is to create a real-time recognition system that bridges the gap between deaf or hard-of-hearing individuals and those who do not understand sign language.

The system utilizes image processing and machine learning models trained on a dataset containing RGB images of Arabic hand gestures representing different letters of the alphabet. The end goal is to accurately classify hand gestures and convert them into written text, which can later be extended into applications such as text-to-speech conversion or integration with conversational AI assistants.

1.2 Motivation

Sign language is the primary mode of communication for millions around the world. However, Arabic Sign Language (ArSL) lacks robust technological tools that facilitate seamless communication between sign language users and those who do not understand it.

Our motivation is to enhance accessibility and inclusion by developing a highly accurate and scalable Arabic Sign Language recognition system. This system enables real-time, gesture-to-text conversion, removing barriers between Arabic-speaking sign language users and the broader community.

2. Project Scope & Features

2.1 Dataset Selection & Preprocessing

- Used the Arabic Sign Language Dataset from Kaggle, consisting of RGB images representing hand gestures for different Arabic letters.
- Performed data cleaning, augmentation, normalization, and resizing to optimize model performance.

2.2 Model Development & Training

- Experimented with various models including SVM, XGBoost, FCN, ResNet-50, and Vision Transformer (ViT).
- Selected ViT as the final model due to its superior performance and 98% accuracy.
- Trained the model using PyTorch and Hugging Face libraries for flexibility and performance.

2.3 Real-Time Gesture Recognition & Deployment

- Integrated the trained model with a webcam input pipeline using OpenCV.
- Utilized Gradio to build an interactive, browser-based UI for real-time prediction.
- Implemented letter buffering and sentence reconstruction using Camel Tools.

2.4 Future Expansion & Enhancements

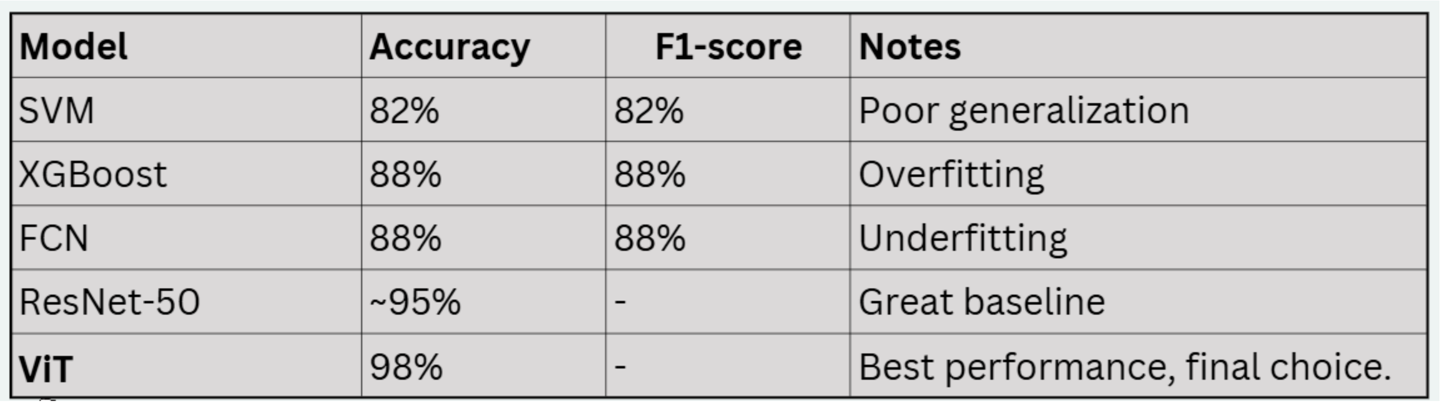
- Extend the system to recognize complete words and sentence structures.
- Add speech synthesis to provide voice output for recognized text.
- Deploy the system as a mobile app or browser extension for greater accessibility.

3. Techniques Used

- **Image Preprocessing**: Color space conversion, cropping, resizing
- **Hand Detection**: MediaPipe and OpenCV contour-based methods
- **Feature Extraction**: Deep learning-based feature encoders
- **Sequence Handling**: Letter buffering for word and sentence construction
- **Text Cleaning**: Arabic sentence reconstruction using Camel BERT Tools

4. Model Exploration

- Compared different models for gesture classification:
    - **SVM**: Moderate accuracy, poor scalability
    - **XGBoost**: Required manual feature extraction, cause overfitting.
    - **Ensemble Methods:**
        - Stack Classifier [SVM, XGBoost]->[SVM] (86.4%)
        - Stack Classifier [XGBoost,LGBM] ->[SVM] (88.8%)
    - **FCN**: Underfit, struggled with complex images
    - **ResNet-50**: Good performance (~95% accuracy)
    - **ViT**: Best performance with 98% accuracy and better generalization



Model	Accuracy	F1-score	Notes
SVM	82%	82%	Poor generalization
XGBoost	88%	88%	Overfitting
FCN	88%	88%	Underfitting
ResNet-50	~95%	-	Great baseline
ViT	98%	-	Best performance, final choice.

- Chose Vision Transformer (ViT) due to its attention-based architecture and strong performance on structured image data

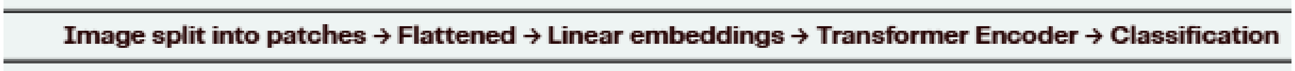
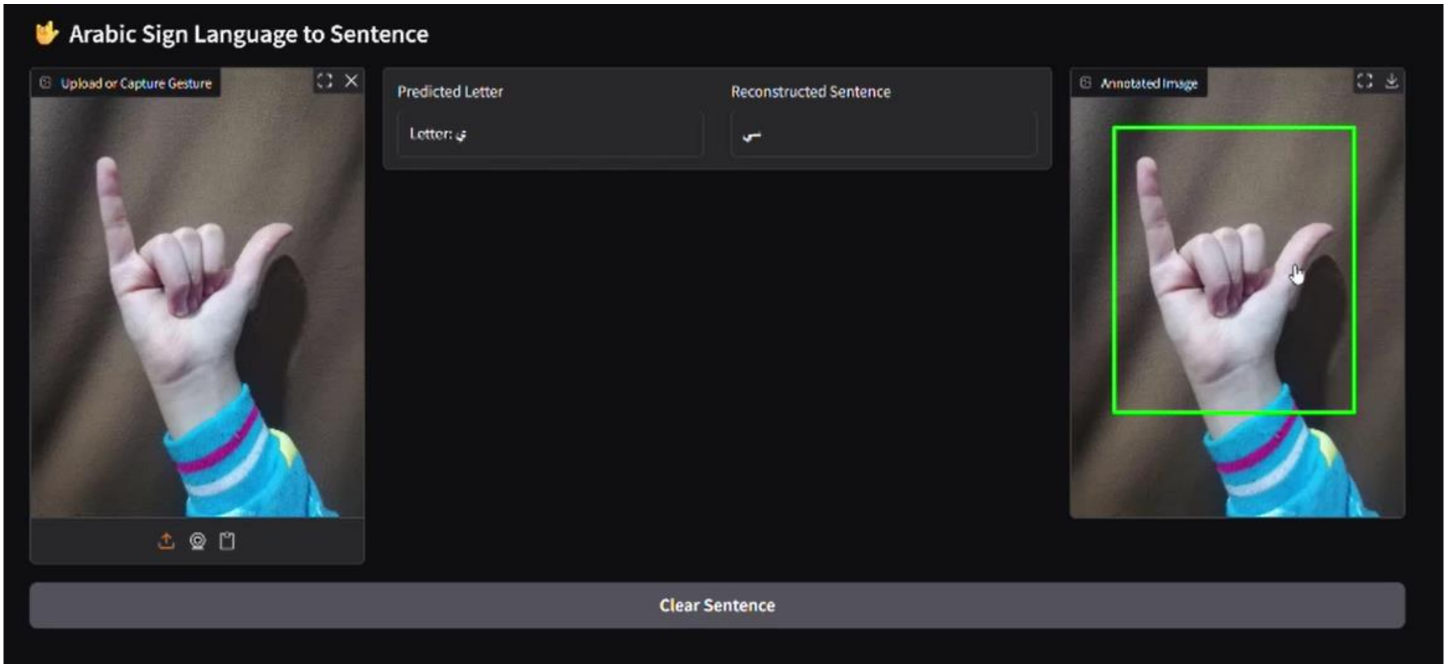


Image split into patches → Flattened → Linear embeddings → Transformer Encoder → Classification

5. UI & Deployment

- Built the front-end using Gradio for rapid prototyping and ease of use
- Supports image upload and real-time webcam snapshot
- Displays:
    - Detected Arabic letter
    - Cleaned sentence (buffered letters)
    - Annotated image with bounding box
- Includes a clear button to reset the sentence buffer
- Entire system runs locally in-browser without need for cloud deployment



Arabic Sign Language to Sentence

Letter: ي

ي

6. Implementation Plan

6.1 Project Timeline *& Phases*

Phase	Tasks
1. Planning & Research	Define objectives, research existing models, finalize dataset
2. Data Collection & Preprocessing	Clean and preprocess data, perform exploratory data analysis
3. Model Development & Training	Implement CNN, train model, optimize performance
4. Real-Time System Integration	Integrate model with webcam feed, create UI
5. Deployment & Testing	Deploy system, conduct real-world testing
6. Final Documentation & Presentation	Prepare final report, conduct live demonstration

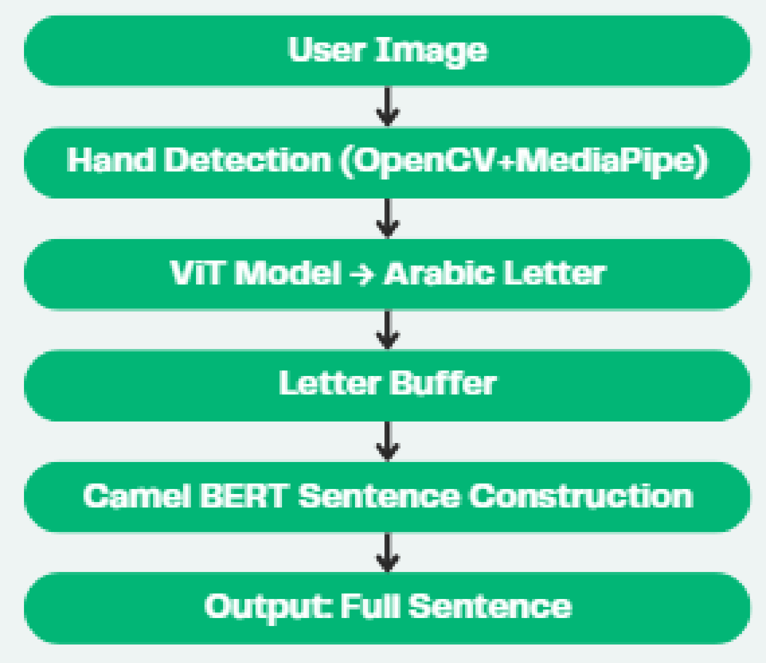
7. Technical Implementation

7.1 Tools & Libraries

- **Data Handling**: numpy
- **Image Processing**: OpenCV, PIL
- **Modeling**: PyTorch, transformers (ViT), scikit-learn
- **Deployment/UI**: Gradio
- **Text Processing**: Camel Tools

7.2 Model Architecture

- **Input**: 224x224 RGB image of cropped hand gesture
- **Feature Extraction**: Vision Transformer (ViT)
- **Classification**: Fully connected softmax layer to predict letter class
- **Output**: Predicted letter → buffered → cleaned sentence


User Image
Hand Detection (OpenCV+MediaPipe)
ViT Model → Arabic Letter
Letter Buffer
Camel BERT Sentence Construction
Output: Full Sentence

7.3 Performance Metrics

- **Accuracy**: Achieved 98% on test data
- **Latency**: Low-latency predictions for near real-time feedback
- **Robustness**: Model generalizes well across multiple gesture inputs

8. Expected Outcomes & Future Work

8.1 Expected Outcomes

- Deliver a high-accuracy gesture recognition model for Arabic Sign Language
- Achieve real-time prediction through optimized backend and lightweight model
- Provide an intuitive Gradio-based interface to promote usability
- Promote digital inclusivity for Arabic-speaking signers

8.2 Future Work

- Expand dataset to cover dialectical/regional gesture variations
- Integrate NLP for context-aware sentence reconstruction
- Deploy as a mobile application with offline capabilities

9. Conclusion

This project presents an innovative, real-time system for recognizing Arabic Sign Language gestures and converting them into coherent Arabic sentences. By combining the strengths of Vision Transformers, OpenCV preprocessing, and Camel Tools language processing within a Gradio-based user interface, the system enhances communication accessibility for Arabic-speaking deaf and hard-of-hearing individuals.

Our implementation demonstrates strong technical performance, and the system is ready for future extensions such as voice output, mobile deployment, and expanded language understanding. This project contributes meaningfully to the field of assistive technology and digital inclusion for the Arabic-speaking community.