TransLingualVisionary

Kavish Purani, Neeraj Ramesh, Sandra Serbu
Team E6

Use Case

Problem Difficult for deaf or hard of hearing (HOH) individuals to participate in live digital environments (online meetings, live streams, etc.)

Lack of widespread understanding of American Sign Language (ASL) often requires the hearing impaired to rely on assistance from translators to communicate.

Solution A real-time ASL speech to English text translator on a user friendly web application

Design Requirements

Requirement	Metric
Recognize when a user is signing	~95% sign recognition rate
Correctly identify ASL words	Recognize 2000 words at ~80% accuracy
Correctly interpret ASL semantics	Translate identified clusters of words into full english sentences with a BLEU score of ~40%
Classification Distance	Recognize and retain accuracy of the classification model up to 4-5 feet away from the camera.
Text Accessibility	Display and collect the ASL Speech in an accessible user format that can be easily found and read.
Overall Latency ~ real time	Present visual feed and translation on web UI within ~3 seconds

Solution Approach

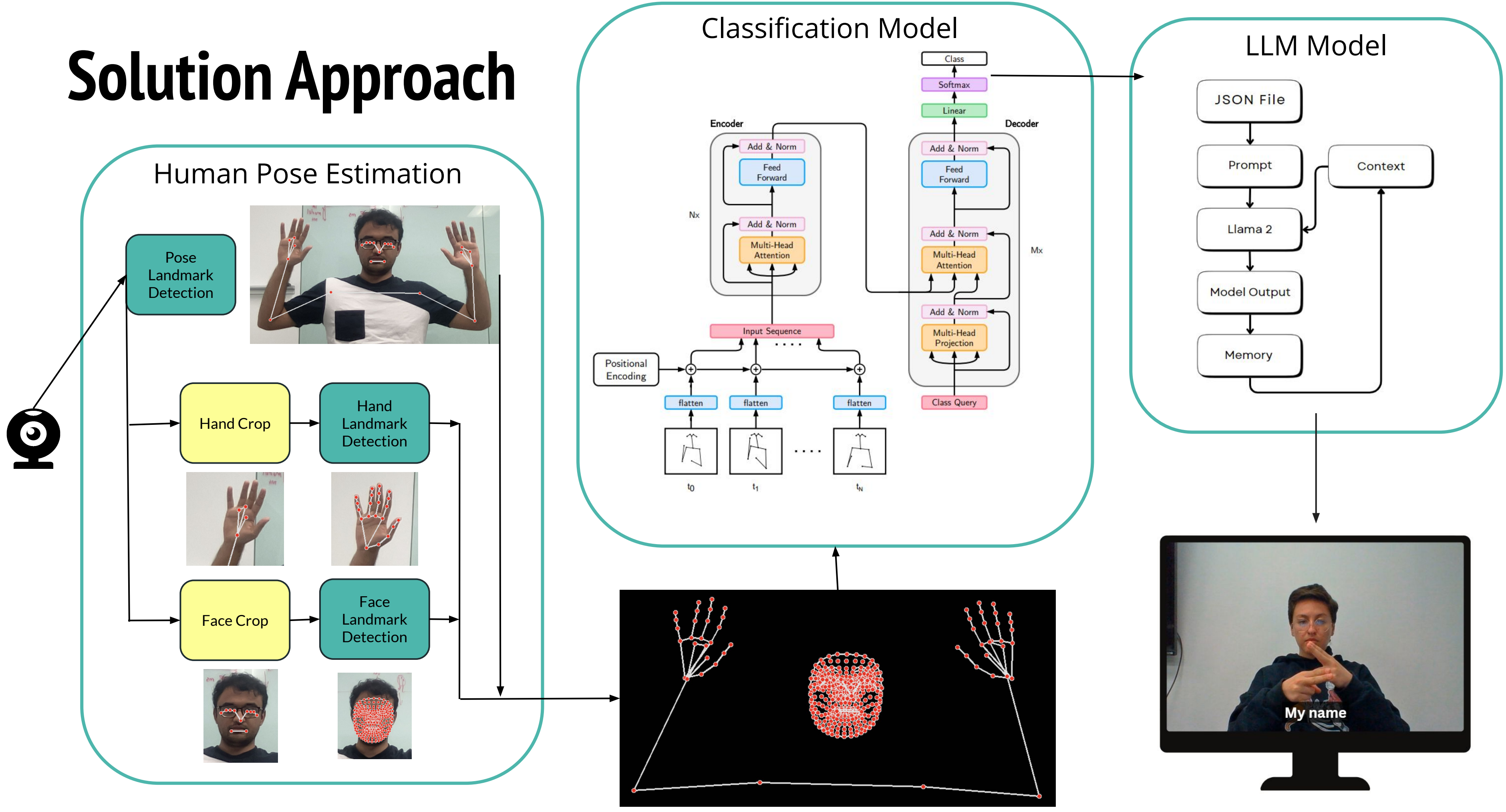
Welfare Accessible Technology serves to foster Autonomy

Important for users to see the accuracy of their intended speech

Health and Safety
Data Security Exposure

TLV runs locally, there is low to no risk of a malicious actor overhearing sensitive information

Solution Approach



Trade-offs and Decisions

Overall Pipeline	HPE + Transformer Pipeline	Single Transformer
Pros	- Lightweight transformer (fewer parameters) - Removes extra detail	- Single model ⇒ Easier pipeline to train
Cons	- Need to modify data between models	- Can capture extraneous detail - Far more model parameters

Classification Architecture	Transformer	RNN
Pros	- Captures short and long term dependencies via self-attention - Parallel processing	- Lighter model - Simpler architecture
Cons	- Heavier model	- Exploding/Vanishing Gradient - Sequential Processing

Trade-offs and Decisions (cont.)

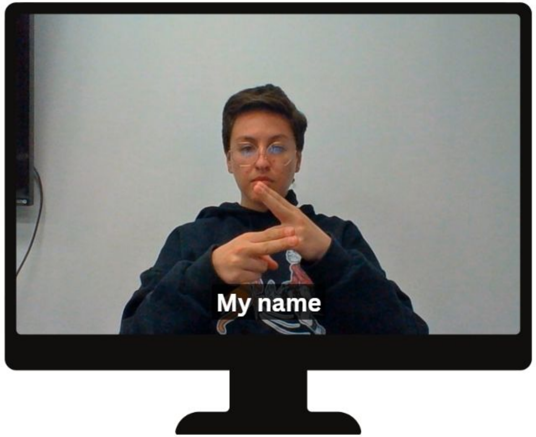
HPE	Jetson	FPGA	
Pros	- More CPU processing power - Models on device limits communication latency	- Opportunity to optimize models - Splitting models between devices to prevent excessive compute cost	
Cons	- More models running Jetson could decrease compute speed - Tighter space restriction due to multiple models on device	- Higher performance dependent on quantizability of models - Communicating between devices would increase latency	
FPS	~ 17 fps	Unaccelerated	Accelerated
		~3 fps	~25 fps*

LLM	Llama 2	GPT-4
Pros	Free and open source and LLM access	API access → Ease of use and offloaded computation
Cons	Runs locally– computation time may be slower and setup	Requires credits to access model

*calculated using different HPE model than ours; didn't use said model due to lack of necessary landmarks

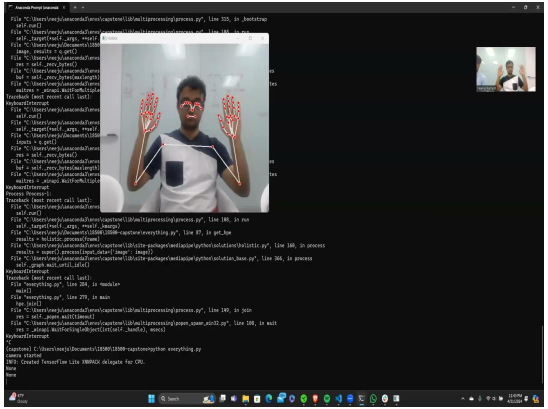
Complete Solution and Demonstration

Simple User Display



TLV on Local Resources

Classification



Reporting Quantitative Results

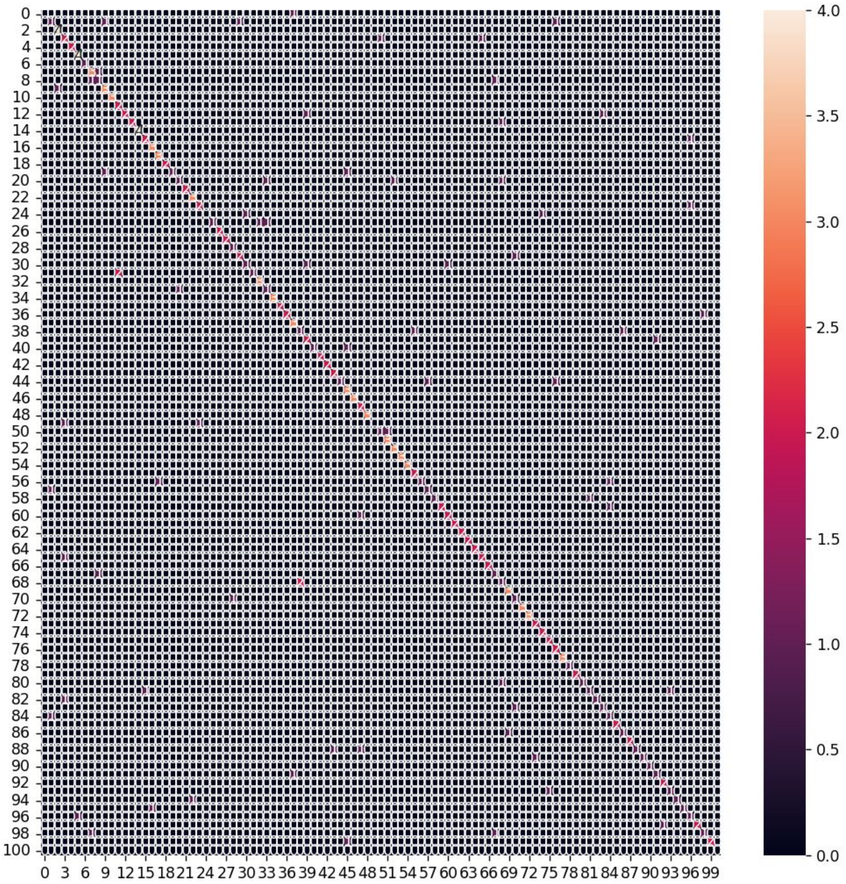
Metric	Tests	Goal	Results
Recognition Rate	Calculate how often the model provides a sign classification when a user is signing. *	~95%	~98%
Word Classification Accuracy	Split data into training and validation sets and then calculate how often the model's output and the desired output is the same.	Training ~95% Validation ~85%	Training ~ 97.82% Validation ~ 72.44%
Inference Accuracy	Calculate accuracy of inference (how often the user's sign and the model's word is the same)	~80%	~55%
Overall Latency	Run timer from beginning of HPE to classification output	~ 3 seconds	~ 2.2 seconds
Unit Latency	Measure latency via inter-component timestamps during live inferencing for various and signs	HPE: ~600 ms Classification: ~800ms	HPE: ~65 ms Classification: ~12 ms

*Tendency towards false positives (preferred over false negatives); optimize to prevent excessive false positive rate

Quantitative Results

Confusion matrix of validation accuracy. Model prediction on the x-axis, true label on the y-axis. Frequency shown through heatmap.

Training ~ 97.82%
Validation ~ 72.44%



Technical Challenges

- HPE Model isn't quantizable on FPGA
  - Running on Jetson
- No API access through OpenAI
  - Running LLM locally using Llama 2
- False positives with word recognition
  - Thresholding softmax to prevent classification that the model is not "confident" about
- Extraneous frames in training set affecting classification model
  - Pruning training set based on whether there is a hand in the frame
- Frame count of inferencing
  - Training transformer based on set frame count that we are going to use for inferencing
  - Pruning training should allow for leniency with set frame count

Project Management

	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W13	W13
	2/5	2/12	2/19	2/26	3/11	3/18	3/25	4/1	4/8	4/15	4/22	4/29
Presentation And Report												
Proposal	SKN											
Design Review				SKN								
Final										SKN	SKN	SKN
Hardware												
FPGA Ramp-up	K	K	K	K	K							
Camera I/O		K	K									
Model Verification		K										
Model Implementation			K									
Model Implementation Benchmarking				K								
I/O Testing and Benchmarking					K	K	K					
Mediapipe Implementation								K				
Software												
Find and test datasets	SN											
Mediapipe Implementation on Jetson		SN							K	K		
Testing and Optimization of Mediapipe										K	K	
Developing Transformer model			SN	SN	N	N	N	N				
Testing and Optimization				SN					N	N	N	N
Prompt Engineering LLM		SN	SN	SN	S	S	S	S	S			
Fine Tuning Local LLM Model								S	S	S	S	
Testing LLM From Jestion Word Classification												S
Integration of Word Classification and LLM Models												SKN
Simple Web App												
Developement						K	K					
Testing								K	K	K	K	K
Final Integration												
Testing									KN	KN	KN	SKN
Slack									SKN	SKN	SKN	SKN

SKN	Sandra, Kavish, Neeraj
K	Kavish
SN	Sandra, Neeraj
N	Neeraj
S	Sandra
KN	Kavish, Neeraj

Remaining work

- Retrain word classification
- Integration on local device
- User interface