VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SIGN LANGUAGE RECOGNITION USING CUDA AND TENSORFLOW

ASSIGNMENT- 1

MCSE503L- COMPUTER ARCHITECTURE AND ORGANIZATION

FALL SEMESTER 2024-25

SLOT: D2+TD2

PRESENTED BY:

JOHN ALEXANDER - 24MAI0004

BANSI BHUVA - 24MAI0012

TEJ KANJARIYA - 24MAI0030

Under the guidance of

PROF. SAIRABANU J

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING, VIT UNIVERSITY

PROBLEM STATMENT: SIGN LANGUAGE RECOGNITION USING CUDA AND TENSORFLOW

ABSTRACT :-

This research paper aims to enhance the performance of sign language recognition systems by leveraging parallel processing techniques through CUDA and TensorFlow. Although the specific implementation details are still under development, the primary focus will be on utilizing various types of Convolutional Neural Networks (CNNs) to improve the accuracy and efficiency of recognizing sign language gestures. The anticipated contributions include a robust framework that not only accelerates processing times but also increases the overall recognition accuracy, thereby facilitating better communication for the hearing-impaired community.

KEYWORDS :-

CUDA, Parallel Processing, Sign Language Detection, Convolutional Neural Networks (CNNs), Performance Improvement.

INTRODUCTION :-

Sign language recognition is a critical area of research that aims to bridge communication gaps for the hearing-impaired community. Traditional methods often struggle with accuracy and real-time processing, which are essential for effective communication. This study investigates the application of parallel processing techniques using CUDA alongside deep learning frameworks like TensorFlow to enhance the performance of sign language recognition systems. The primary objective of this research is to implement parallel processing to improve the performance of sign language detection systems. By exploring different architectures of CNNs, we aim to identify the most effective models that can handle the complexities of sign language gestures. This research is significant as it not only seeks to advance the technical capabilities of sign language recognition systems but also aims to contribute to the inclusivity and accessibility of communication tools for the deaf and hard-of-hearing individuals. Through this study, we hope to provide insights into the integration of cutting-edge technologies in developing efficient and effective sign language recognition systems, ultimately enhancing user experience and interaction.

LITERATURE REVIEW:

Title	Author Name	Journal and Date	Key Findings	Advantages	Disadvantages	Future Work
Mime Recognition for Indian Sign Language	Shrivarshaa Sakhamuri; Koppula Praneeta; Pidugu Jahnavi; Anuradha Chinta	2022 3rd International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT) 20 March 2023	The model uses Convolutional Neural Networks (CNNs) to achieve 98.27% accuracy in recognizing and converting Indian Sign Language gestures into text and vice versa.	Provides fast and precise conversion of gestures or signs to text and text to signs.	It only handles single sign recognition, lacking support for sentence-level sign detection. It has not yet been converted into a mobile application for broader accessibility.	Future work will be, to convert the graphical user interface of our developed model into a mobile application and to provide sentence-level sign detection.
Breaking Down Communication Barriers: Real-Time Sign Language Recognition Using CNN & Flask-Based API	Lakshmi Bhavani Nekkanti; Priyanga A; A. Mary Posonia; J. Albert Mayan	2023 International Conference on Circuit Power and Computing Technologies (ICCPCT) 22 September 2023	Real-time recognition of sign language system using image processing & CNN to decode gestures in video footage. Flaskbased API is provided that can be distributed to cloud computing, making solution widely accessible.	The system incorporates image processing methods include pixel brightness changes, fourier transforms, image filtering & classification, and Fourier & image restoration. Model performed at 95% accuracy on the validation set, with recall and precision of 0.95 & 0.94, respectively.	Current systems only recognize individual gestures, limiting natural and spontaneous conversation. High computational resource demands may impact efficiency.	Develop systems to identify full phrases for smoother communication. Integrate visual feedback for enhanced interaction. Improve computational efficiency to reduce resource usage. Creating a mobile app for real-time communication.
American Sign Language Real Time Detection Using TensorFlow and Keras in Python	Amsaprabhaa M; H Saadhvi Sree; Jayashre; Kavini Muthamizhvalavan; Nidhi Gummaraju; Padmajaa S	2024 3rd International Conference for Innovation in Technology (INOCON)	Python based sign language detection system using image recognition and CNN to accurately interpret ASL gestures. The detected gestures are converted to text and displayed, offering a scalable solution for improved accessibility and inclusivity for the hearing-impaired community, implemented with OpenCV and TensorFlow.	It analyze both the spatial and temporal characteristics of hand movements, extracting meaningful features and patterns that represent different ASL gestures. By leveraging computer vision techniques, gesture analysis algorithms, and a user-friendly GUI, the system improving accessibility.	System accuracy depends on high contrast between the hand color and background. Lack of contrast can reduce recognition accuracy.	Improve recognition accuracy by focusing on hand feature extraction instead of threshold images. Develop more advanced training models for higher accuracy in varied environments. Expand the system's capabilities for more natural communication, such as recognizing full phrases.

ShuffleNetv2-YOLOv3: a real-time recognition method of static sign language based on a lightweight network	Sun, Shiniu ,Han, Lisheng,Wei, Jie,Hao, Huimin, Huang, Jiahai, Xin, Wenbin, Zhou, XuA, Kang, PengA	Springer Conference 12 January 2023	The ShuffleNetv2-YOLOv3 lightweight model improves sign language recognition speed and accuracy on mobile and embedded platforms by using ShuffleNetv2 as the backbone of YOLOv3 and Convolutional neural network,Gesture recognition,Object detection techniques.	This model is optimized for real-time, static gesture recognition, providing a strong foundation for mobile applications. It achieves high F1 (99.1%) and mAP (98.4%) scores, with detection speeds of 54 FPS on GPU, significantly outperforming models like YOLOv3-tiny and SSD.	Real-time performance on mobile terminals is hindered by large and complex network architectures, and achieving a balance between model accuracy, size, and speed remains challenging.	Combining lightweight architectures with advanced techniques like Transformers, integrating additional features such as depth and hand motion tracking, optimizing models for diverse mobile and embedded platforms, and exploring multimodal systems that incorporate vision along with other input methods to enhance recognition capabilities.
Indian Sign Language Recognition: A Comparative Analysis Using CNN and RNN Models	S Renjith; Rasmi Manazhy	2023 International Conference on Circuit Power and Computing Technologies (ICCPCT) 22 September 2023	This study compares CNN and RNN for Indian Sign Language (ISL) recognition. CNN captures spatial features, while RNN excels in temporal dynamics. Both models demonstrate high accuracy, precision, recall and F1 scores.	CNN: Efficient in detecting local spatial patterns and structures in visual data. RNN: Effective in capturing temporal dynamics and the sequential nature of gesture data.	CNN: Ineffective in modeling temporal relationships and gesture order. RNN: Computationally intensive and slower due to its sequential processing requirements.	Focus on hybrid CNN-RNN models, incorporating depth information and hand motion tracking, exploring advanced techniques like Transformers, and developing multimodal systems for enhanced ISL recognition.

Title	Author Name	Journal & Date	Key Findings	Advantages	Disadvantages	Future Work
Parallel Training of Deep Neural Networks with GPU Accelerators	Dean, J.; Corrado,G.; Monga, R.	Advances in Neural Information Processing Systems (NIPS), 2012	Parallel training on GPUs significantly reduces model convergence time.	Faster model convergence.	Requires specialized knowledge of parallel processing.	Improving scalability by optimizing training on larger distributed GPU clusters, which would reduce inter-GPU communication delays.
Using GPUs for Real-Time Hand Gesture Recognition in Sign Language	Molchanov , P.; Yang, X.; Gupta, S.	IEEE Conference on Computer Vision and Pattern Recognitio n (CVPR), 2015	Significant speedup in gesture recognition for sign language using GPU	Real-time recognition capability.	Computational overhead when handling large-scale datasets. .	Hybrid GPU-accelerated approaches combined with cloud computing for handling large volumes of data with faster processing and accuracy.
Accelerating neural network architecture search using multi-GPU high-performance computing	Liu, Y.; Wang, R.; Sun, S.	IEEE Transactions on Image Processing, 2019	Demonstrated speedup in image segmentation using GPU-optimized CNNs.	Substantial improvements in segmentation speed.	Trade-offs in accuracy when using faster models.	Using advanced search algorithms optimized for multi-GPU setups to reduce the search time for finding the best-performing neural network architectures
Real-Time Gesture-Based Sign Language Recognition System	Jeet Debnath, Praveen Joe I R	2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS), IEEE	The system leverages Python, OpenCV, MediaPipe Holistic, and an LSTM neural network to recognize American Sign Language (ASL) gestures with high accuracy in real-time.	Real-time recognition with low latency and high accuracy.	Computationall y intensive and potential limitations in handling diverse lighting conditions and environments	Plans to integrate natural language processing (NLP) for two-way communication between sign language and spoken language

Title	Author Name	Journal & Date	Key Findings	Advantages	Disadvantaes	Future Work
Improving Automatic Sign Language Translation with Image Binarisation and Deep Learning	Mahmudul Haque, Syma Afsha, Tareque Bashar Ovi, Hussain Nyeem	2021 5th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT)	The paper proposes a 2D-CNN model trained with binarized American Sign Language (ASL) images, achieving a 99.99% accuracy using the SL MNIST dataset. Image binarization and dynamic learning rate (LR) adjustments were used to optimize the model's performance.	Higher accuracy compared to previous models.	Limited to static gesture datasets; no motion-based gesture handling	Incorporate motion-based gestures and use sequential models like RNNs or LSTMs to improve real-world sign language translation.
Slowfast Network for Continuous Sign Language Detection	Junseok Ahn∗, Youngjoon Jang∗, Joon Son Chung	IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)	The SlowFast network, originally designed for video recognition, was adapted to sign language recognition. It uses two pathways: a slow pathway to capture low-motion features and a fast pathway to capture fine-grained, quick motions, leading to a significant improvement in recognizing continuous sign language sequences.	High accuracy in recognizing both slow and fast gestures in sign language; able to handle complex, dynamic gestures due to its multi-speed temporal feature extraction.	Requires high computational resources, making it unsuitable for low-power or mobile devices; training time is longer due to the dual-path network.	Focus on reducing the model size and improving its computational efficiency for real-time applications, particularly on mobile platforms and edge devices.

Sign Language Recognition and Translation Method based on VTN (Video Transformer Network)	Wuyang Qin, Xue Mei *, Yuming Chen, Qihang Zhang, Yanyin Yao, Shi Hu	2021 International Conference on Digital Society and Intelligent Systems (DSInS)	Video Transformer Network (VTN) demonstrates a strong ability in capturing long-range temporal dependencies, making it highly suitable for translating video sequences of sign language into text. The method outperforms traditional CNN-RNN approaches by focusing on self-attention mechanisms that effectively model the temporal dependencies in video frames.	High precision in capturing temporal relationships in sign sequences; more accurate translation of complex, multi-step signs; better generalization due to the transformer architecture.	Transformer models are resource-intensive and require large datasets for effective training; the model may struggle with real-time performance without optimization.	Develop techniques for model compression and optimization for faster inference; investigate how the model handles low-resource sign languages and small datasets, improving inclusivity and accessibility.
SignTrack: Advancements in Real-Time Sign Language Processing for Inclusive Computing with Optimized AI	Stavros Piperakis, Maria Papatsimouli, Vasilis Argyriou, Panagiotis Sarigiannidis, and George F. Fragulis	2024 13th International Conference on Modern Circuits and Systems Technologies (MOCAST)	SignTrack leverages optimized AI models to achieve real-time sign language processing, aiming to create inclusive computing platforms that cater to deaf and hard-of-hearing individuals. The system reduces latency and improves the speed of sign language recognition, making it practical for real-time applications like video conferencing and translation.	High-speed processing with minimal latency; tailored for real-time applications, such as live sign language translation; designed with inclusivity in mind, making computing more accessible to deaf users.	Struggles with recognizing complex, nuanced gestures or less common sign languages; while the model is fast, its accuracy may lag behind when handling less structured signs or multi-hand gestures.	Expand support to more sign languages and gestures, particularly in regions with less available data; further optimize the AI to balance speed and accuracy, ensuring real-time applications can scale across different contexts.
Hardware Parallel Structure for Convolution Computing in Image Processing	Reyes Perez Melesio*, Moises Arredondo-Velázquez† , Javier Diaz-Carmona*	2024 47th International Conference on Telecommunications and Signal Processing (TSP)	A hardware architecture designed for parallelizing convolution operations can drastically improve the speed and performance of image processing tasks. By employing multiple processing units simultaneously, the structure significantly reduces the time required for convolution-heavy tasks like object detection and image classification.	Substantial speed improvements in convolutional operations; suitable for real-time applications like image and video processing; reduces the bottleneck in convolutional layers, especially in high-resolution image processing tasks.	High hardware development and implementation cost; increased design complexity as it requires specialized hardware, limiting its flexibility compared to software-based approaches.	Investigate more energy-efficient designs to reduce power consumption in large-scale deployments; explore ways to make the architecture more accessible for a broader range of applications, such as in low-power or embedded systems

Sign Language Recognition System Using Convolutional Neural Network (CNN)	Ms Pavan Kumar, Pasam Chanakya, Saranya Velusamy	2024 IEEE International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS)	CNN-based sign language recognition systems are highly effective in classifying static hand gestures from images. The system can identify different sign gestures with good accuracy, making it suitable for systems that recognize isolated signs, such as in alphabet-based sign languages.	High accuracy in classifying static hand gestures; relatively straightforward implementation with well-established CNN architectures; easy to train on image datasets, and robust when dealing with variations in lighting and background.	Struggles to capture the temporal dynamics necessary for continuous or dynamic sign language recognition; limited to static gestures or individual frames, making it less useful for fluid sign language communication.	Incorporate sequential models like RNNs or transformers for continuous gesture recognition; develop hybrid architectures that can handle both static and dynamic signs, improving overall system robustness in real-world applications.

PARALLEL PLATFORM DETAILS:

SOFTWARE REQUIREMENTS:

- Python 3.9–3.11
- pip version 19.0 or higher for Linux (requires manylinux2014 support) and Windows. pip version 20.3 or higher for macOS.
- Windows Native Requires Microsoft Visual C++ Redistributable for Visual Studio 2015, 2017 and 2019

The following NVIDIA® software are only required for GPU support.

- NVIDIA® GPU drivers version 450.80.02 or higher.
- CUDA® Toolkit 11.8.
- cuDNN SDK 8.6.0.

INSTALLATION STEPS:

Install NVIDIA Drivers and CUDA Toolkit:

A: visit the NVIDIA website and download the latest GPU drivers for your specific GPU model.

https://www.nvidia.com/Download/index.aspx

b. Install the GPU drivers.

c. Download the CUDA Toolkit that is compatible with your GPU and operating system from the NVIDIA CUDA Toolkit download page.

d. Follow the installation instructions provided in the CUDA Toolkit documentation.

Install cuDNN:

a. Visit the NVIDIA cuDNN download page.

b. Download the cuDNN library that matches your CUDA version.

c. Follow the installation instructions provided in the cuDNN documentation.

Create a Virtual Environment (Optional but Recommended):

It's a good practice to create a virtual environment to isolate your TensorFlow installation from other Python packages. We can use virtualenv or conda for this purpose.

Using virtualenv

pip install virtualenv

virtualenv myenv

source

myenv/bin/activate or

```bash

# Using conda

conda create -n myenv python=3.8

conda activate myenv

```

Install TensorFlow with GPU support:

We can install TensorFlow using pip, specifying the GPU version:

bash

pip install tensorflow-gpu

This command will automatically install the appropriate version of TensorFlow that is compatible with installed CUDA and cuDNN versions.

Verify TensorFlow Installation:

We can verify that TensorFlow is correctly installed and configured to use the GPU by running a simple Python script:

```python
import tensorflow as tf

# Check if a GPU is available and if TensorFlow is
using it if tf.test.is_gpu_available():

    print("GPU is available.")

    print("TensorFlow is using GPU:",
tf.test.gpu_device_name()) else:

    print("No GPU available. TensorFlow is running on CPU.")
```

Running this script should confirm that TensorFlow is using your GPU for computation.

Remember that the specific package versions and installation procedures may change over time, so it's essential to refer to the official TensorFlow documentation and NVIDIA documentation for the most accurate and up-to-date instructions for environment in 2023.

Install NVIDIA Drivers and CUDA Toolkit:

a. Visit the official NVIDIA website to download the latest GPU drivers for your specific GPU model: NVIDIA Drivers.
b. Install the GPU drivers by following the instructions on the NVIDIA website.
c. Download the CUDA Toolkit that matches your GPU and operating system from the NVIDIA CUDA Toolkit download page: CUDA Toolkit
d. Follow the installation instructions provided in the CUDA Toolkit documentation.

Install cuDNN:

a. Visit the NVIDIA cuDNN download page: cuDNN.
b. Down https://developer.nvidia.com/cuda-downloadsload the

cuDNN library that corresponds to your CUDA version and GPU model.

c. Follow the installation instructions provided in the cuDNN documentation.

Create a Virtual Environment (Optional but Recommended):

It's a good practice to create a virtual environment to isolate your TensorFlow installation from other Python packages. We can use virtualenv or conda for this purpose.

Virtualenv: Virtualenv Installation

Conda: Conda Installation

Install TensorFlow with GPU Support:

We can install TensorFlow with GPU support using pip by specifying the GPU version:

```bash

pip install tensorflow-gpu

```

This command should automatically install the appropriate version of TensorFlow that is compatible with installed CUDA and cuDNN versions.

https://developer.nvidia.com/cudnn

Verify TensorFlow Installation:

After installation, verify that TensorFlow is correctly using the GPU by running a Python script like the one mentioned in the previous response. Ensure that TensorFlow is using your GPU for computation.

HARDWARE REQUIREMENTS:

· A working laptop with at least 8GB of RAM.
· High speed Internet.