

**SIGN LANGUAGE RECOGNITION USING CUDA AND TENSORFLOW**

**ASSIGNMENT- 1**

**MCSE503L- COMPUTER ARCHIETECTURE AND ORGANIZATION**

**FALL SEMESTER 2023-24 SLOT: D2+TD2**

**PRESENTED BY:**

**Under the guidance of**

**PROF. SAIRABANU J**

# SCHOOL OF COMPUTING SCIENCE AND ENGINEERING, VIT UNIVERSITY

**PROBLEM STATMNT: SIGN LANGUAGE RECOGNITION USING CUDA AND TENSORFLOW**

**ABSTRACT:** Plant diseases pose a significant threat to global food security and agricultural sustainability. Accurate and timely disease diagnosis is crucial for disease management and crop protection. This research paper explores the utilization of GPU (Graphics Processing Unit) computing to enhance the efficiency and accuracy of plant disease prediction models. In this study, we propose a novel approach that leverages the parallel processing capabilities of GPUs to accelerate the processing of large-scale plant disease datasets and improve the performance of machine learning algorithms. We present a comprehensive analysis of the GPU-accelerated disease prediction framework, which includes data preprocessing, feature extraction, model training, and real-time inference.

**KEYWORDS:** GPU(Graphics Processing Unit), Disease Management, Crop Protection

# INTRODUCTION:

In an era characterized by burgeoning global population growth and heightened concerns about food security, the health and vitality of agricultural crops hold a pivotal role in ensuring sustainable and resilient food production systems. However, the agricultural sector faces relentless challenges posed by the omnipresent threat of plant diseases. These diseases can cause substantial crop losses, affecting not only the livelihoods of farmers but also the availability and affordability of food for communities worldwide. Addressing the timely and accurate detection of plant diseases has, therefore, become an imperative focus of agricultural research and innovation.

Traditionally, the identification of plant diseases has relied heavily on visual inspections by human experts. While this method can be effective, it is inherently labor-intensive, time- consuming, and subject to human error. Furthermore, as the scale of agricultural operations expands to meet the demands of a growing population, the need for more efficient and scalable disease detection solutions becomes increasingly urgent.

Recent advancements in the field of artificial intelligence, particularly in machine learning and computer vision, offer a transformative potential for revolutionizing plant disease detection. Machine learning algorithms have demonstrated remarkable capabilities in analyzing complex datasets, making them particularly well-suited for the task of identifying subtle and often imperceptible signs of plant diseases from images and sensor data. These algorithms can learn and generalize from large and diverse datasets, enabling automated and accurate disease identification across a variety of plant species and disease types.

This research paper endeavors to explore the burgeoning intersection of plant pathology and machine learning, aiming to provide a comprehensive overview of the state-of-the-art techniques and methodologies in plant disease detection using machine learning approaches. It delves into the challenges and opportunities that arise in this dynamic field, discusses the potential for real-world applications, and highlights the broader implications for sustainable agriculture and global food security.

In this context, we will review the current literature, analyze prominent case studies, and elucidate the technical intricacies involved in developing machine learning-based solutions for plant disease detection. Moreover, we will consider the practical considerations of implementing these technologies in agricultural settings, including scalability, cost-

management. effectiveness, and accessibility for resource-limited farming communities.

Ultimately, the fusion of plant pathology and machine learning holds the promise of transforming our ability to detect and manage plant diseases swiftly and effectively. By harnessing the power of data-driven techniques, we can empower farmers with tools that not only enhance crop health but also contribute to the resilience and sustainability of our global food supply chain. This research paper endeavours to shed light on this compelling convergence and its potential to shape the future of agriculture and plant disease.

**LITERATURE REVIEW:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | TITLE | AUTHOR NAME | JOURNAL AND  DATE | KEY FINDINGS | ADVANTAGE S | DISADVANGE S | FUTURE ENHANCEMENT  S |
| 1  . | Convolution hierarchical deep-learning neural network (C-HiDeNN)  with graphics processing unit (GPU)  acceleration | Chanwook Park, Ye Lu Sourav Saha. Tianju Xue, Jiachen Guo, Satyajit Mojumder, Daniel W, Apley, Gregory J, Wagner,  Wing Kam Liu | Springer coenferenc e  on  31st May, 2023 | Proposed a C- HiDeNN framework based on GPU computation. Although the interpolants are based on a linear finite element mesh, nodal connection information allows for higher reproduction orders. | C-HiDeNN  using s, a, and p variables. combining self- consistent clustering analysis with concurrent multiscale theory.the GPU programming of the C-HiDeNN theory utilizing is geometric analysis. | GPU  programming greatly reduces the computational burden of C- HiDeNN.  Commercial FEM software running on CPU is slower than the most recent version of C- HiDeNN with a small patch size. | C-HiDeNN with variable s, a, p parameters can be studied. A combination with concurrent multiscale theory such as self- consistent clustering analysis is another future goal. Finally, C- HiDeNN theory and its GPU programming will be extended to  isogeometric analysis. |
| 2  . | 3D Ultrasound Tomography Image Reconstruction Algorithm by GPU | Jiaduo, J, Gong | CVIPPR  conference  19th June 2023 | Uses TVL3 algorithm to reconstruct the original image from the information collected by the UT system for clinical use. | Investigated the acceleration of the transmission tomography using TVL3.  TVL3 is an iterative optimization algorithm which performs augmentation on large scale  images. | Used GPU to parallelize the two-matrix vector multiplication separately then it was found that the GPU site is faster, but it takes a lot of time in data transfer. | Multi GPU method can be used further to improve the acceleration effect and the parallel ability to get better results. |
| 3 | Accelerating neural network architecture search using multi‑GPU high‑performanc e computing | Marcos Lupión, N. C. Cruz, Juan F. Sanjuan, B. Paechter3 Pilar M. Ortigosa | Springer conference  1st December 2022 | This work describes how the teaching- learning based optimization algorithm has been adopted for designing neural network exploriting a multi -GPU high performance computing environment. | This strategy obtains a speedup factor of 90 for the first stage using 96 CPU cores, and a speedup factor of 3.83 for the second one using 4 GPUs. | TLBO generally need to evaluate numerous solutions, which is computationally. demanding, and also need GPUs to run in reasonable times. | To implement a mechanism to group disabled large altogether to have permanently simplified architectures and reduces the search space. Finally different optimization methods will be compared within the proposed framework of continuous neural network encoding and decoupled oracle-based  evaluation. |
| 4  . | Improving detection and classification of  diabetic | Abdussamed Erciyas, Necattin  Barisci | Springer conference | In this detection phase of the method,  Gradient Based | Diabetic Retinopathy classification  was made using | Using Faster RCNN does not completely  cover the lesion | This strategy can be used to diagnose more  diseases and can |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | retinopathy using CUDA and MASK RCNN |  | 14th August, 2022 | edge detection method developed with Mask RCNN and CUDA was used instead of faster RCNN. In the classification phase, only Image Nat models were used. | the GPU. Some improvements have been made to the method to improve the lower results in some pretrained models. In the detection phase, the background in the DR image was extracted using CUDA and Gradient Method. | in all cases. As the spaces between the box and the lesion increase when the lesion spreads thinly and diagonally within the retina, the empty space inside the box becomes larger than the lesion area. This causes difficulties in learning the  lesion pattern. | contribute heath sector for earlier diagnosis of a particular disease. |
| 5  . | Dog Disease recognition using Image Segmentation and GPU Enhanced convolutional Neural Network. | Beau Gray M.Habal Pierre Edwin See Tiong. | IEEE  conference 2021 | The researchers have developed asystem for detection and diagnosis of dog skin diseases using image processing techniques and achieved in creating a model with the application of (CNN)  enhanced by (GPU) for an enhanced maximum  speed of training. | Creation of system for detection and diagnosis of skin disease using image processing techniques and achieved in creating a model with the application of CNN enhances by graphics processing unit. | Processing speed is slow and can be improved by using latest version of CUDA to have a higher score level. | Latest ML algorithms can be used to get more accuracy with a Multi-GPU system. |
| 6  . | GPU Allocation Strategy for Deep Learning | Yingwanchan JianChen Han,Huan Zhou Chinchin. | ,IEEE 2022 | We propose a scheduling framework for GPU clusters which improves performance and reduces energy  consumption of clusters. | It is adapted to various environment and more effective than other methods especially when cluster is busy. | Huge energy consumption of clusters, The unpredictable DLT task completion time. | Research can be done where the task competition time and cost both can be reduced parallelly using GPU allocation strategies. |
| 7  . | GPU Based Multiple Face recognition using YOLO5X | Ali Abed,Ali Abdulghafar Jaload. | IICCIT 2022 | The face recognition system is a deep neural networks adopted for learning effective features and parameters to obtain faster and more reliable  algorithms. | The system could be considered as a building block for all commercial applications that need face recognizers such as security and surveillance civil and military  systems. | Due to the high computational complexity of deep learning algorithms, multiple face recognition is another challenge. | As a future work, low cost, energy efficient real time multiple face recognizer designed in this work can be integrated with any security, attendance or surveillance system. |
| 8  . | Structured Binary Neural Network for Image Recognition. | Bohan Zhuang, Chunhua Shen | Springer conference  22nd June 2022 | Implemented OpenCL(simila r with CUDA) and is a common programming language on the embedded platforms to implement the GPU  acceleration. | Have explored highly efficient and accurate CNN  architectures with binary weights and activations. They have proposed to  directly decompose the | A new approach for quantization can be used. | To employ the latent-free optimizer for BNNs that directly updates the binary weights. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | full-precision network into multiple groups and each group is  approximated. |  |  |
| 9  . | A property Management System using image recognition by YOLO | Taiki MIYAMOTO  , Ryo FUKUSHIM A | IEEE  conference  August, 2021 | YOLO model is implemented along with OCR to detect text from the image. | The preprocessing of images enhances the accuracy of recognizing texts in OCR. | Noise of image causes failure in the OCR process and YOLO draws lines vertically and horizontally only around the  objects. | Better scope for property management system using YOLO algorithm and object detection. |

**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](https://support.microsoft.com/help/2977003/the-latest-supported-visual-c-downloads) [2015, 2017 and 2019](https://support.microsoft.com/help/2977003/the-latest-supported-visual-c-downloads)

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

* [NVIDIA® GPU drivers](https://www.nvidia.com/drivers) version 450.80.02 or higher.
* [CUDA® Toolkit 11.8](https://developer.nvidia.com/cuda-toolkit-archive).
* [cuDNN SDK 8.6.0](https://developer.nvidia.com/cudnn).

# 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>

1. Install the GPU drivers.
2. Download the CUDA Toolkit that is compatible with your GPU and operating system from the NVIDIA CUDA Toolkit download page.
3. Follow the installation instructions provided in the CUDA Toolkit documentation.

## Install cuDNN:

1. Visit the NVIDIA cuDNN download page.
2. Download the cuDNN library that matches your CUDA version.
3. 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.

bash

# 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:

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

## Install cuDNN:

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

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

1. 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.