****

**School of Computing Science & Engineering**

**MCSE503L-Computer Architecture and Organisation**

Digital Assignment-2 Report

Submitted By:

24MAI0004 - John Alexander

24MAI0012 - Bansi Buva

24MAI0024 - Tej Kanjaria

## Sign Language Recognition using CUDA and Tensorflow

## Abstract

Sign language serves as an essential communication medium for individuals with speech and hearing impairments, though real-time translation and accessibility remain challenging. This project explores a high-performance deep learning approach to American Sign Language (ASL) recognition through Convolutional Neural Networks (CNNs) with accelerated model inference using CUDA and TensorRT. The model, trained on a dataset of ASL letters (A–Z) and digits (0–9), efficiently classifies 36 unique hand signs with high accuracy.

The ASL recognition system combines a custom-trained CNN model with OpenCV and TensorFlow for real-time classification. During training, CUDA-enabled GPUs accelerated the computation-intensive tasks, significantly reducing training time and enabling hyperparameter optimization for improved accuracy. In the live video feed, the deployment of TensorRT optimizes the model for low-latency inference, providing near-instantaneous predictions on hand gestures captured from a webcam. To further enhance real-time performance, a hand-detection module isolates hand regions, minimizing background interference and improving model focus.

**Keywords:** Sign Language Recognition, American Sign Language (ASL), Convolutional Neural Network (CNN), Real-Time Classification, CUDA Acceleration, TensorRT Optimization, Hand Tracking, Deep Learning, Accessibility Technology, Computer Vision

##### Introduction

The ability to communicate effectively is fundamental to human interaction, yet individuals with speech and hearing impairments face significant barriers due to limited accessibility in mainstream communication. American Sign Language (ASL), a structured visual language that uses hand gestures, facial expressions, and body language, serves as an essential medium for many within these communities. However, there remains a considerable need for tools that translate ASL gestures into spoken or written language in real time, allowing for seamless communication between ASL users and non-signers.

Advancements in deep learning and computer vision offer promising solutions to bridge this communication gap. Convolutional Neural Networks (CNNs), widely recognized for their proficiency in image classification, have shown significant potential in gesture and sign language recognition tasks. By integrating CNNs with real-time hand detection and tracking, ASL gestures can be recognized with a high degree of accuracy in dynamic environments. However, processing these gestures in real time, particularly in live video feeds, demands substantial computational resources.

To address this challenge, this project leverages CUDA-enabled GPUs and TensorRT optimization to enhance the efficiency of model training and real-time inference. CUDA, a parallel computing platform, accelerates the training of deep learning models on GPUs, significantly reducing training time and enabling complex hyperparameter tuning. TensorRT, a high-performance deep learning inference library, further optimizes the CNN model for low-latency predictions, allowing for fast and efficient recognition of ASL gestures in a live streaming environment.

##### Contribution of the work

This work contributes a high-performance, real-time American Sign Language (ASL) recognition system that utilizes deep learning techniques, GPU acceleration, and model optimization. Key contributions include:

**Real-Time ASL Gesture Recognition Pipeline**: We develop a robust pipeline that integrates a Convolutional Neural Network (CNN) for multi-class classification of 36 ASL gestures (letters and digits) in live video, with OpenCV-based hand detection and tracking to isolate hand gestures from background noise.

**Accelerated Model Training**: By leveraging CUDA-enabled GPUs, we significantly reduce training time and enable advanced hyperparameter tuning, resulting in an accurate and efficient CNN model.

**Optimized Inference with TensorRT**: For live-streamed recognition, TensorRT optimizations minimize latency, allowing for near-instantaneous predictions, thus enhancing the system’s practical applicability in real-world scenarios.

**Accessible Communication Technology**: This project lays foundational work toward accessible, scalable ASL translation technology, bridging communication gaps for speech and hearing-impaired individuals and advancing real-time language translation systems.

**LITERATURE REVIEW:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | TITLE | AUTHOR NAME | JOURNAL AND  DATE | KEY FINDINGS | ADVANTAGE | DISADVANGE | FUTURE ENHANCEMENT |
| 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. |
| 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 usng | 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. |

**Existing System Description:**

Existing systems for American Sign Language (ASL) recognition primarily fall into three categories: glove-based solutions, vision-based solutions, and hybrid approaches. Each has advantages and limitations, particularly in accuracy, cost, and applicability for real-time, large-vocabulary recognition.

1. **Glove-Based Solutions**: These systems use specialized gloves equipped with sensors (e.g., flex sensors, accelerometers, and gyroscopes) to capture hand movements and finger positions. The sensors relay motion data, which is processed to recognize individual signs. While glove-based solutions can achieve high accuracy, they are often limited by cost, restricted mobility, and the inconvenience of wearing hardware. Examples include sensor gloves like the SignAloud Glove and AcceleGlove.
2. **Vision-Based Solutions Using Handcrafted Features**: Vision-based ASL recognition systems use cameras to capture hand gestures and apply image processing techniques to detect and classify gestures. Earlier systems relied on handcrafted features, such as shape, color, and edge detection, combined with classical machine learning algorithms like Support Vector Machines (SVMs) or Hidden Markov Models (HMMs) for classification. However, these approaches struggled with variability in lighting, background, and individual differences in hand shape, which often led to lower accuracy.
3. **Vision-Based Solutions Using Deep Learning**: Recent advances in deep learning have popularized Convolutional Neural Networks (CNNs) and other deep architectures for gesture recognition. CNNs can learn complex visual features automatically from raw image data, improving accuracy and robustness over handcrafted methods. Systems like Google's Teachable Machine allow non-experts to train custom models for simple sign recognition, but these models are often not optimized for real-time performance and have limited sign vocabularies.
4. **Hybrid Approaches**: Some systems combine glove-based sensors with vision-based recognition to improve accuracy and reliability. By combining motion data from gloves with visual data from cameras, these systems can capture more nuanced aspects of ASL gestures. However, such systems are complex, expensive, and generally unsuitable for wide-scale use.

**DATASET :** https://www.kaggle.com/datasets/ayuraj/asl-dataset/data

The data set is a collection of images of alphabets from the American Sign Language, separated in 36 folders which represent the various classes.

The data set contains 2515 images which are 400x400 pixels. There are 36 classes.

The 36 classes contain 0-9 and A-Z alphabets.

### Proposed System Architecture with module wise description

The proposed system architecture for real-time American Sign Language (ASL) recognition is based on a Convolutional Neural Network (CNN) model integrated with a hand detection system and optimized for fast inference. The architecture is designed to capture hand gestures from a live video feed, preprocess the images, classify them into corresponding ASL signs, and display the results with high accuracy and low latency. The system is modular and consists of the following key components:

1. Hand Detection Module

Description: This module uses the cvzone.HandTrackingModule library to detect and track a single hand in each video frame. The module identifies hand landmarks and bounding boxes to accurately isolate the region containing the hand gesture.

Key Functions:

Captures real-time video feed and identifies the presence of a hand.

Extracts the bounding box around the hand and applies padding to create a consistent area for gesture recognition.

Ensures stable hand tracking to reduce noise and improve classification accuracy.

2. Background Processing Module

Description: After detecting the hand, this module processes the background of the detected hand to increase contrast and clarity. It applies a mask to isolate the hand and set the background to black.

Key Functions:

Sets the area outside the detected hand to black, improving model focus on the gesture.

Enhances the model’s ability to detect hand shape and features by removing irrelevant background elements.

3. Image Preprocessing Module

Description: The cropped hand image is preprocessed to match the CNN’s input specifications (200x200 pixels, grayscale). Preprocessing also includes normalization and resizing.

Key Functions:

Resizes the cropped hand image to 200x200 pixels.

Normalizes pixel values to the [0, 1] range to enhance training consistency.

Converts the image to grayscale, reducing complexity while retaining essential gesture information.

4. Convolutional Neural Network (CNN) Classification Module

Description: This core module utilizes a CNN architecture to classify the processed hand image into one of the 36 ASL classes (26 letters and 10 digits). The CNN has three main convolutional blocks for feature extraction, followed by fully connected layers to generate the classification output.

Key Functions:

Extracts spatial features of the hand gesture through multiple convolutional and pooling layers.

Uses dense layers to map the extracted features to 36 output classes, with a softmax layer for final classification.

Outputs the predicted label for each gesture.

5. CUDA and TensorRT Optimization Module

Description: This module leverages CUDA and TensorRT for faster model inference, essential for real-time applications. CUDA enables parallel processing on GPU, while TensorRT optimizes the model for efficient execution.

Key Functions:

Converts the trained model into a TensorRT-optimized version, reducing inference time.

Enables the system to process frames at high speeds without compromising accuracy.

Ensures smooth, real-time gesture recognition by leveraging GPU acceleration.

6. Prediction Display Module

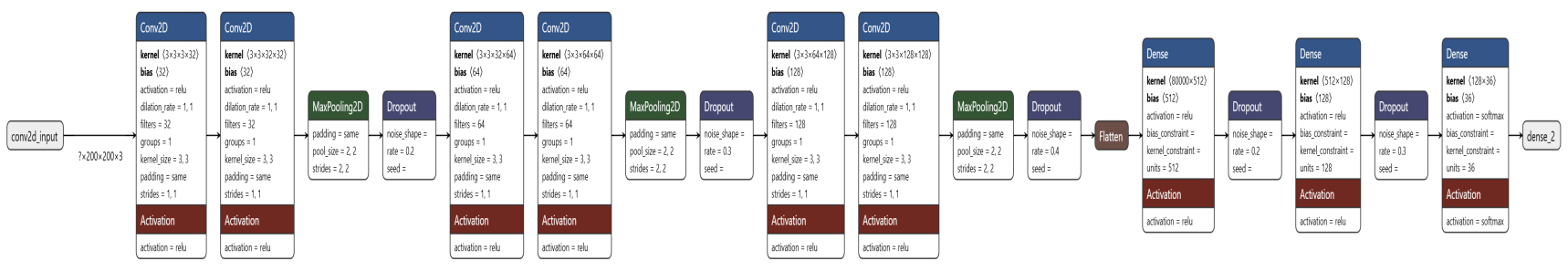
Description: This module overlays the predicted gesture label on the live video feed, making it easy for the user to view the recognition results in real-time.

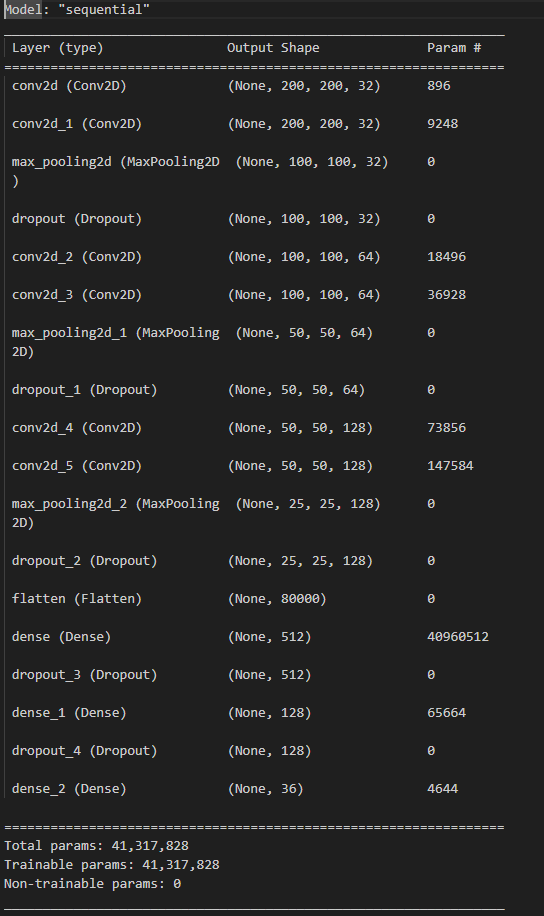
Key Functions:

Displays the predicted ASL gesture label at the top of the bounding box around the hand.

Highlights the bounding box with a colored border for visual clarity.

Provides immediate feedback to the user by updating the prediction continuously with each frame.

.



#### Parellelization using CUDA and TensorRT

#### 1. ****Parallelized Model Inference Using CUDA****

* **Overview**: CUDA (Compute Unified Device Architecture) enables general-purpose parallel processing on NVIDIA GPUs, allowing the system to execute many operations simultaneously rather than sequentially. By harnessing the power of CUDA, the CNN model can perform matrix operations, convolutions, and other complex computations on large batches of data in parallel, significantly reducing computation time.
* **Application in ASL Recognition**:
  + **Matrix Multiplications in Convolution Layers**: Convolutional operations, which involve a significant number of matrix multiplications, are parallelized using CUDA. Each filter operation across different portions of an image is computed simultaneously on the GPU.
  + **Batch Processing**: CUDA enables batch processing of input frames, which improves model inference speed, especially for frames captured in quick succession. This batching allows the model to perform more than one classification in parallel, increasing the frame rate without compromising accuracy.

#### 2. ****Model Optimization with TensorRT for Parallel Execution****

* **Overview**: NVIDIA TensorRT is a high-performance deep learning inference optimizer and runtime library, designed to maximize the efficiency of deep learning models on NVIDIA GPUs. TensorRT optimizes neural networks by pruning and fusing layers and by quantizing the model for lower precision (e.g., FP16 or INT8), which can reduce memory usage and speed up inference.
* **Application in ASL Recognition**:
  + **Layer Fusion and Kernel Auto-Tuning**: TensorRT combines layers and finds the best kernel implementations for the hardware, reducing the number of separate computations. This is especially useful in the CNN architecture, where consecutive convolutional layers and dense layers benefit from reduced overhead due to fewer kernel launches.
  + **Dynamic Tensor Memory Allocation**: By dynamically allocating memory only when necessary, TensorRT minimizes memory bottlenecks, allowing the system to handle multiple frames and model executions in parallel with less latency.
  + **Precision Reduction**: TensorRT can convert model parameters to lower precision (e.g., from FP32 to FP16), which significantly decreases computation time while retaining a high level of accuracy. This optimization allows the system to handle real-time inference requirements for ASL recognition.

#### 3. ****Parallel Video Frame Processing Pipeline****

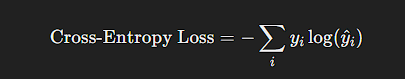
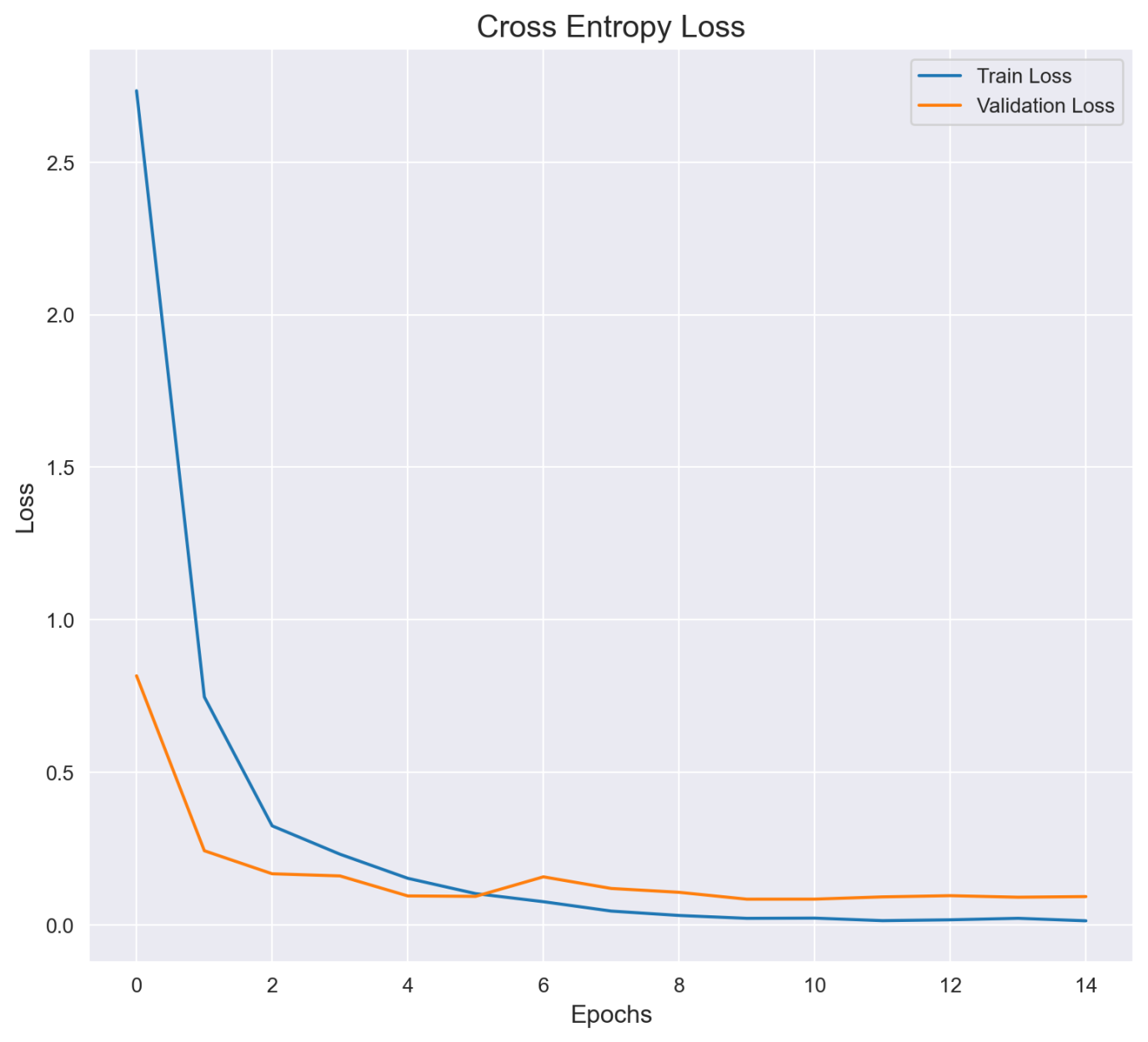
* **Overview**: The system is designed to process frames in a parallelized pipeline, which splits the tasks of hand detection, preprocessing, and classification across multiple stages that can run concurrently. By parallelizing the frame processing pipeline, the system minimizes idle times between frames.
* **Application in ASL Recognition**:
  + **Concurrent Hand Detection and Preprocessing**: While one frame is being classified by the CNN, the next frame undergoes hand detection and preprocessing, ensuring that there’s minimal delay between receiving new input and generating a prediction.
  + **Data Transfer Parallelism**: The parallel pipeline also handles data transfer between the CPU (capturing video frames) and the GPU (inference) in parallel, with frame batches continuously sent to the GPU, preventing CPU-GPU communication from becoming a bottleneck.

**GITHUB LINK** : https://github.com/John-Alex07/Sign-Language-Recognition

### **METRICS FOR MODEL EVALUATION:**

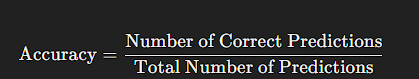
### Cross-Entropy Loss

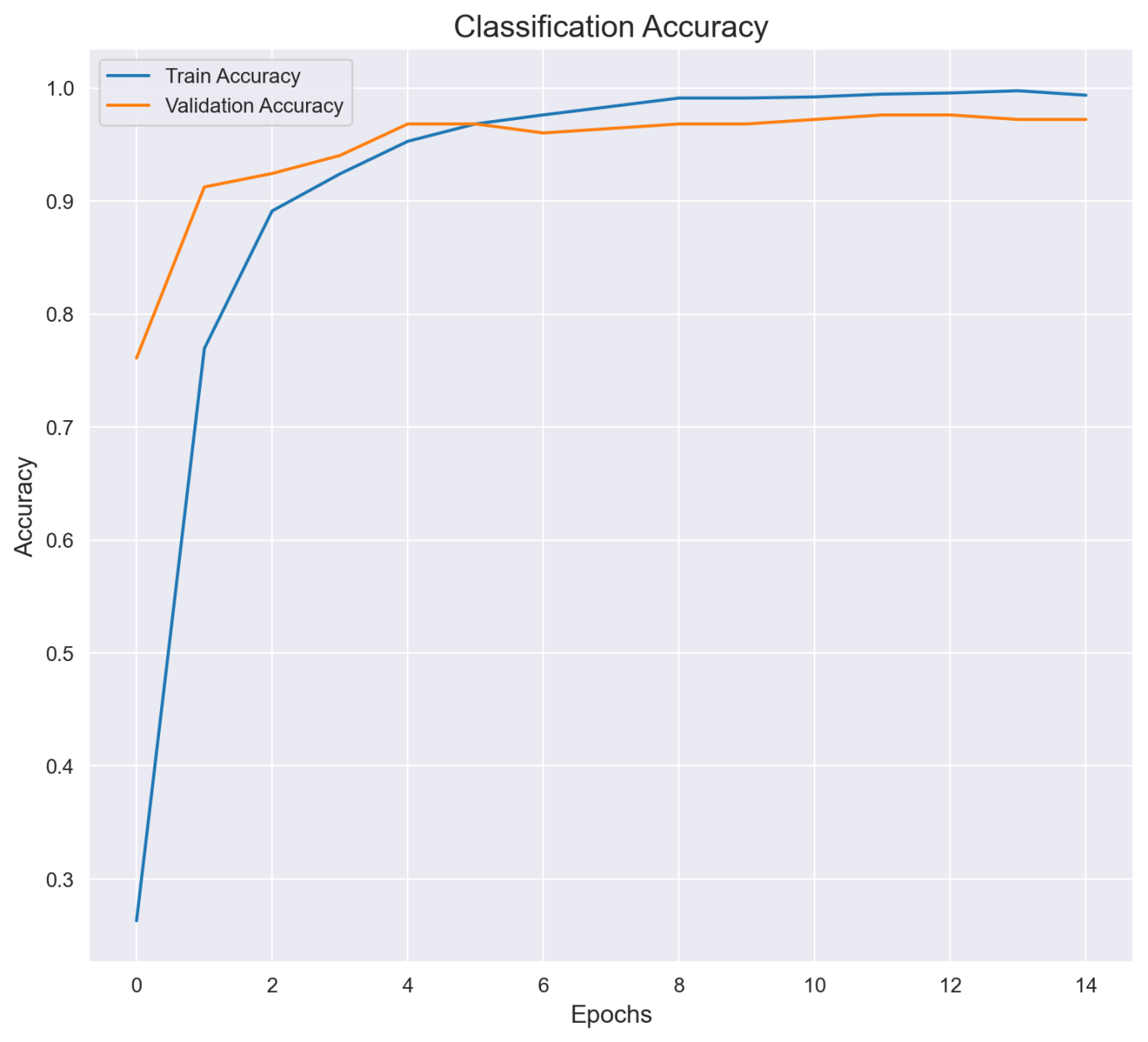
Cross-entropy loss measures the performance of a classification model whose output is a probability value between 0 and 1. It calculates the logarithmic difference between the predicted and true label probabilities.



**Accuracy**

Accuracy is the ratio of correctly classified instances to the total instances, and it represents the model's overall effectiveness.

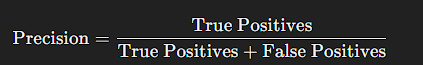




**Precision, Recall, and F1-Score**

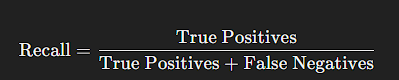
These three metrics are useful, especially in cases where class imbalances exist or some classes are more challenging than others to classify.

**Precision**

 Precision calculates the proportion of true positive predictions out of all positive predictions made by the model.

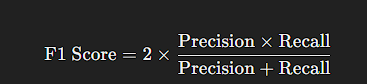
**Recall (Sensitivity)**

Recall calculates the proportion of true positive predictions out of the actual positive instances in the dataset.

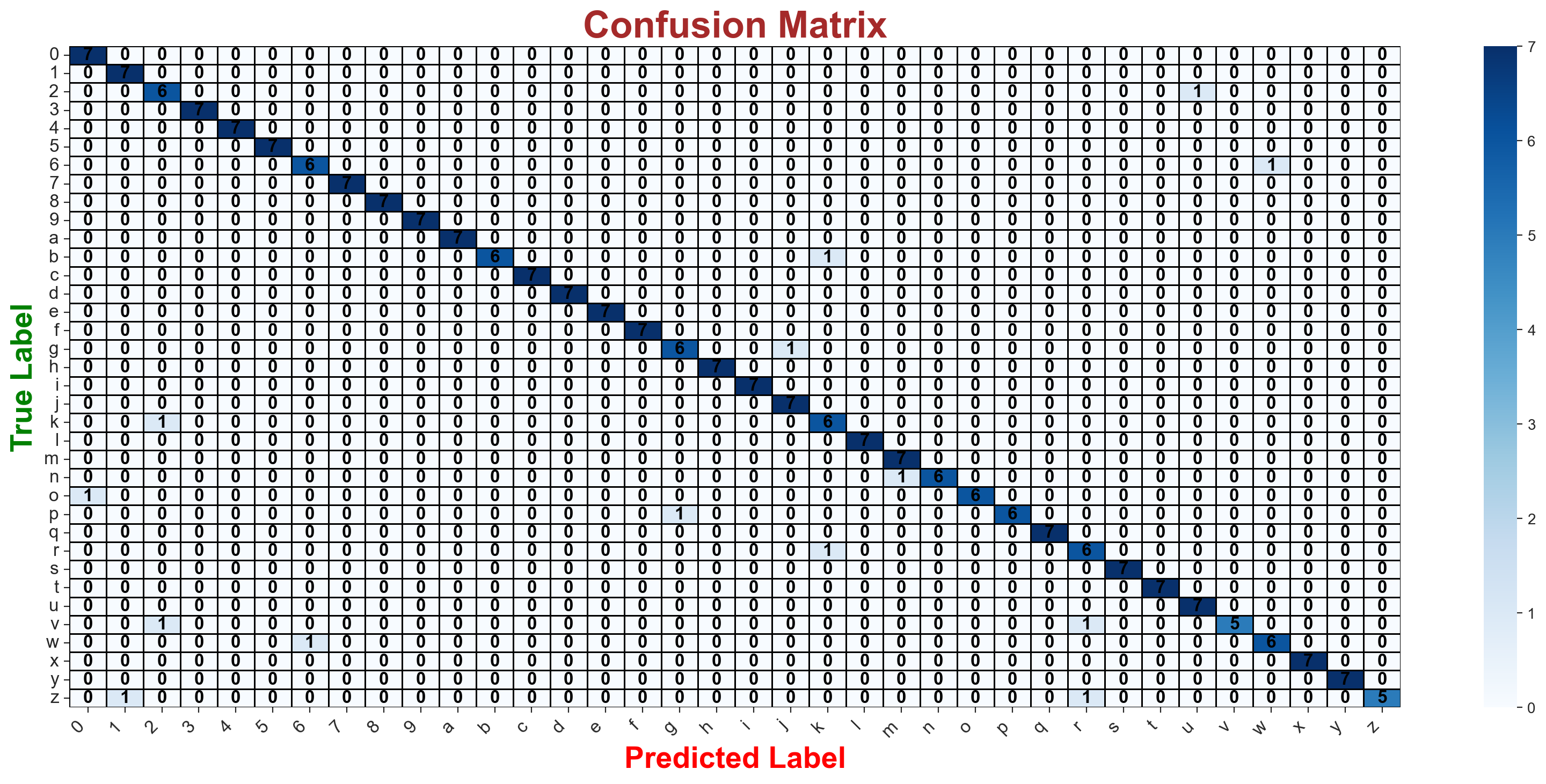


**F1-Score**

The F1-score combines precision and recall into a single metric, representing the harmonic mean of the two.



**Confusion Matrix**

The confusion matrix is a table that visualizes the performance of the model by showing the true positive, false positive, true negative, and false negative counts for each class.

### Conclusion

The Sign Language Classification project demonstrates the effective application of parallel computing techniques to enhance the performance of deep learning models. By leveraging CUDA and GPU computing, we can significantly accelerate the training and inference processes, enabling the model to handle the computationally intensive tasks required for recognizing American Sign Language (ASL) signs in real time.

Utilizing TensorRT for inference optimization allows for further improvements in the model's responsiveness and efficiency. TensorRT optimizes the trained model by reducing its size and increasing its inference speed without compromising accuracy. This is particularly crucial for real-time hand sign recognition, where latency must be minimized to provide seamless communication.

In conclusion, the integration of parallel computing strategies, including CUDA and TensorRT, within GPU computing frameworks not only enhances the model’s training efficiency but also ensures that it meets the demands of real-time applications. The comparative training times—**7,320 seconds on CPU versus just 960 seconds on GPU**—underscore the advantages of GPU utilization, making this approach a valuable tool for sign language interpretation and facilitating better communication for the hearing impaired.

**Acknowledgement.**

## REFERENCES

1. Park, C., Lu, Y., Saha, S., Xue, T., Guo, J., Mojumder, S., Apley, D. W., Wagner, G. J., & Liu, W. K. (2023). Convolution hierarchical deep-learning neural network (C-HiDeNN) with graphics processing unit (GPU) acceleration. Springer conference, 31 May.
2. Jiaduo, J., & Gong, J. (2023). 3D Ultrasound Tomography Image Reconstruction Algorithm by GPU. CVIPPR conference, 19 June.
3. Lupión, M., Cruz, N. C., Sanjuan, J. F., Paechter, B., & Ortigosa, P. M. (2022). Accelerating neural network architecture search using multi-GPU high-performance computing. Springer conference, 1 December.
4. Erciyas, A., & Barisci, N. (2022). Improving detection and classification of diabetic retinopathy using CUDA and MASK RCNN. Springer conference, 14 August.
5. Gray, B. M. H., Habal, P. E., & See Tiong, T. (2021). Dog disease recognition using image segmentation and GPU enhanced convolutional neural network. IEEE conference.
6. Jian, Y., Han, C., Zhou, H., & Chinchin. (2022). GPU allocation strategy for deep learning. IEEE conference.
7. Abed, A., & Jaload, A. A. (2022). GPU based multiple face recognition using YOLO5X. IICCIT conference.
8. Zhuang, B., & Shen, C. (2022). Structured binary neural network for image recognition. Springer conference, 22 June.
9. Miyamoto, T., & Fukushima, R. (2021). A property management system using image recognition by YOLO. IEEE conference, August.