Northeastern University

**CSYE7105 HIGH-PERFORMACE PARALLEL MACHINE LEARNING & AI**

**PROJECT PROPOSAL**

TEAM 19

PROJECT TITLE:

**High-Performance ASR System Development with Parallel GPU Computing using PyTorch Distributed**

MEMBERS:

**ARNAB CHAKRABORTY**

**RISHABH INDORIA**

# Introduction

Automatic Speech Recognition (ASR) systems are revolutionizing human-computer interaction, enabling speech-driven applications like virtual assistants and automated captioning. The LibriSpeech dataset, featuring audiobooks with diverse speakers and real-world noise, provides a valuable resource for training and evaluating ASR models. This project focuses on utilizing PyTorch Distributed to harness the power of parallel GPU computing for accelerated model training.

## Background

ASR systems play a crucial role in bridging the gap between spoken and written language. They power functionalities like voice search and dictation software, and models like OpenAI's Whisper demonstrate the potential of transformer-based architectures for speech-to-text tasks. PyTorch Distributed provides a scalable solution for parallelizing computations across multiple GPUs, making it ideal for training large models like transformers.

## Motivation

The rise of large language models (LLMs) and their multimodal counterparts needs seamless integration with speech data. This project is driven by the need for high-performance ASR systems that can accurately recognize speech across various domains. By leveraging parallel GPU computing with PyTorch Distributed, we aim to accelerate model training and improve the scalability of ASR solutions. This approach enables us to train larger models on larger datasets, leading to enhanced performance and accuracy.

## Goal

Our primary goal is to develop a state-of-the-art ASR system that achieves exceptional accuracy on the LibriSpeech data set. We propose implementing the Conformer architecture, which exploits both the long-range dependencies captured by transformers and the local context learned by convolutional neural networks (CNNs) and leveraging PyTorch Distributed to distribute computations across multiple GPUs. By utilizing parallel GPU computing, we aim to accelerate training and achieve faster convergence, ultimately leading to improved ASR performance.

# Methodology

## Data Preparation

* Load the audio files from the LibriSpeech dataset, ensuring that each file is accurately tagged with metadata such as book, chapter, speaker, and corresponding text transcription.
* Preprocess the audio data, including resampling, normalization, and data augmentation techniques to improve model robustness.
* Partition the dataset into training, validation, and test sets, following established practices to ensure a fair and representative evaluation.

## Research

Conduct a comprehensive review of recent research papers on Automatic Speech Recognition (ASR) architectures, focusing on state-of-the-art models like transformers and conformers.

## Model Implementation

Implement the deep learning model based on the conformer architecture using PyTorch, incorporating essential components such as convolutional layers, transformer layers, and attention mechanisms. Configure PyTorch Distributed to distribute and perform parallel computations across GPUs.

## Training

* Configure PyTorch Distributed to distribute computations across CUDA-enabled GPUs on Northeastern's Discovery cluster, harnessing the computational power of parallel processing for accelerated model training.
* Train the Conformer model on the preprocessed training set, employing optimization algorithms such as Adam and loss functions like Connectionist Temporal Classification (CTC) Loss or Hybrid Attention-Based End-to-End (HAT) Loss.
* Implement techniques such as gradient clipping, learning rate scheduling, and early stopping to improve model convergence and prevent overfitting.

## Evaluation

Evaluate the trained Conformer model's performance on the validation set, using standard ASR metrics such as Word Error Rate (WER) and Character Error Rate (CER).

## Hyperparameter Tuning

* Perform parallel hyperparameter tuning experiments to optimize the model's performance further, exploring different configurations for learning rates, batch sizes, and model architectures.
* Leverage techniques such as random search, grid search, or Bayesian optimization to efficiently navigate the hyperparameter space and identify optimal configurations.

# Description of Data Source

The LibriSpeech dataset consists of approximately 1000 hours of read English speech, derived from audiobooks. It provides a valuable resource for training and evaluating ASR models, with carefully segmented and aligned audio files.

# Data Source

* LibriSpeech dataset: <https://www.openslr.org/12>
* https://huggingface.co/datasets/librispeech\_asr