# Automatic Speech Recognition Using Deep2Search Model

**Abstract**

This report presents the development and evaluation of an Automatic Speech Recognition (ASR) system implemented using a modified version of the Deep2Search ASR model by developed by baidu. The model aims to improve speech recognition accuracy under noisy conditions using the Akan dataset. I used a dataset of 1000 hours audio clips using only 1000 instances for the training to ensure my computing power meets the resources required to train the model. Unlike the original deep speech algorithm, I utilized mfcc features instead of spectrogram as mfcc is more generally compatible with speech recognition.

## Introduction

Automatic Speech Recognition (ASR) technology has significantly evolved, enabling machines to understand and transcribe human speech effectively. This project explores the use of Deep Speech 2, a deep learning framework, to enhance ASR systems. The main objective was to develop a robust ASR model that can operate effectively in noisy environments.

## Methodology

Dataset Description

The dataset comprised 1000 hours of audio in akan language. The dataset was split was into train, test and validation dataset for the creation of the model. Only 1000 instances were used for training and evaluating the model.

### Data Preprocessing

I applied the mel-frequency cepstral coefficients to the audio signals to extract relevant features. Normalization was then applied to ensure consistency in the data.

### Challenges

Challenge was mainly due to the lack of computing resources to train the model hence code runtime broke disconnects randomly or run out of resources since I used Google Colab. Streaming a large amount of data from Dropbox is also problematic as api sometimes breaks down as a result of the size of the dataset

## Model Architecture

The ASR model was based on a Residual convolutional neural network (RCNN) learn relevant audio features followed by a Bidirectional Recurrent Neural Network (BiRNN) layer, it is followed by a Bidirectional GRU then finally forwarded to a fully connected layer used to classify timesteps. This architecture was chosen for its ability to capture both temporal and spatial features from the speech input.

- **CNN Layer Normalization (CNNLayerNorm):**

Purpose: Normalizes the input spectrogram before passing it through the convolutional layers.

Implementation: Utilizes layer normalization to standardize activations across the features.

- **Residual CNN (ResidualCNN):**

Purpose: Extracts hierarchical features from the input mel-frequency cepstral coefficients.

Design: Consists of multiple residual blocks, each comprising 2 convolutional layers with residual connections.

Activation: Applies GELU (Gaussian Error Linear Unit) activation after each convolutional layer to introduce non-linearity and facilitate feature learning.

Normalization: Employs layer normalization before the GELU activation to stabilize the learning process.

- **Bidirectional GRU (BidirectionalGRU):**

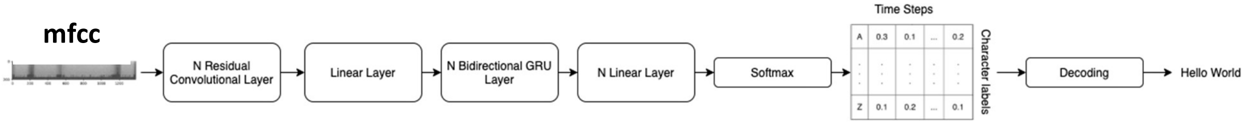
Purpose: Captures temporal dependencies in the speech sequence.

Design: Utilizes bidirectional GRU layers to process the input sequence in both forward and backward directions. This allows the model to capture context from past and future time steps simultaneously, giving it the ability to learn long-range dependencies in the speech signal.

Activation: Applies GELU activation after the GRU layers to introduce non-linearity.

Normalization: Utilizes layer normalization to stabilize the training process and improve convergence.

- **Speech Recognition Model (SpeechRecognitionModel):** This class integrates the above components to perform speech recognition.



**Architecture of the Deep Search Model**

## Training Process

We employed a combination of real-time data augmentation and a curriculum learning strategy to train the model. The optimizer used was Adam, with a learning rate of 0.001, reduced by 10% after every epoch where the validation loss plateaued.

## Evaluation Metrics

The primary metric for evaluating the model was the Word Error Rate (WER). Additional metrics included the Character Error Rate (CER) and the Sentence Error Rate (SER). Loss and Accuracy plots were also utilized. Unfortunately, they did not show significant impact as the size of dataset used was small due to my limited computational resources.

## Results

The model showed potential to be very powerful but could not train on the whole data due to computational resources. The model implemented is a state-of-the-art model with few modifications therefore would really help in training a model for the Akan language.

## Discussion

The reduction in the loss highlights the effectiveness of the combined Deep Search architecture and training strategies in handling noisy speech data. However, the model still struggles with very training due to difficult in streaming the data and computational resources.

## Conclusion

The Deep2Search-based ASR model demonstrated significant improvements in recognizing speech under noisy conditions. Future work will focus on integrating more robust noise handling mechanisms and exploring the impact of larger datasets on model performance.