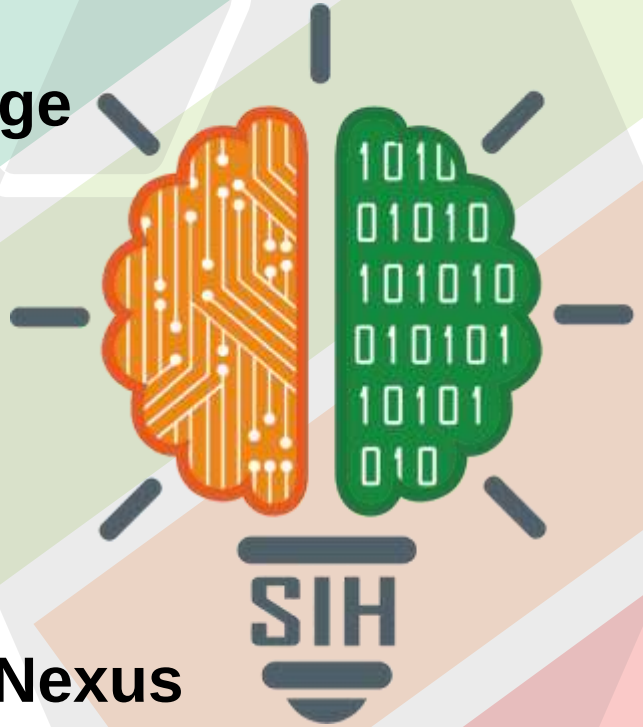
SMART INDIA HACKATHON 2024

SMART INDIA HACKATHON 2024

- **Problem Statement ID – 1680**
- **Problem Statement Title- Few Shot Language Agnostic Keyword Spotting System**
- **Theme- System Automation**
- **PS Category- Software**
- **Team ID-**
- **Team Name (Registered on portal)- Neural Nexus**
- **Team Mentor- Mr. H.S. Pannu**



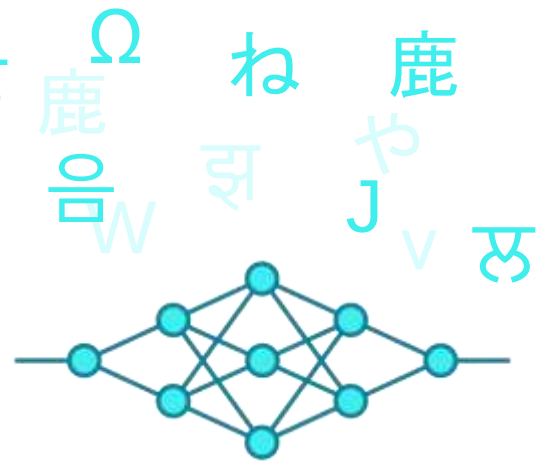
FEW SHOT TRANSFER LEARNING AUTOMATION

SMART INDIA HACKATHON 2024

WHAT OUR SOLUTION DOES AND HOW:

Provides a framework to train a keyword spotting model with very few audio samples

- Uses Large Multilingual Embedding Model.
- Keyword Databank to spot known, unknown or cross-language keywords.
- Audio samples are spliced, pre-processed and individually analysed.
- Penultimate layer is used as feature vector.
- Databank trained on target and non-target keywords.

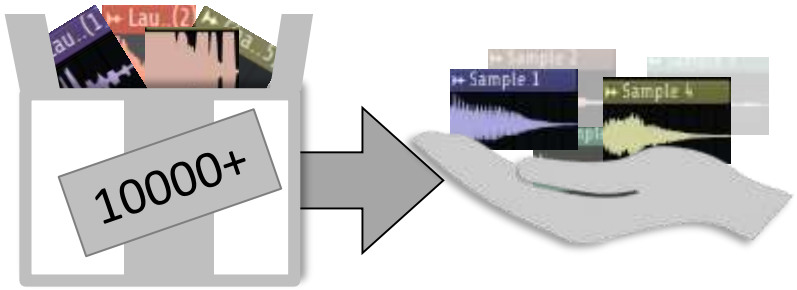


HOW IT TACKLES THE PROBLEM:

- Pertaining on large corpus allows embedding to learn from audio samples.
- This will help us ensure that words that haven't been known can also be trained in relatively low resources

HOW IT DIFFERS FROM OTHER SOLUTIONS:

- Other approaches uses thousands of keyword samples while ours will only need a handful.



TECHNOLOGIES TO BE USED

Programming Languages

Python: Will be used for implementing ML models because it has a rich set of **ML libraries** and frameworks like **TensorFlow** and PyTorch.

Machine Learning Frameworks

TensorFlow: Will be used for **developing and training** ML models required for keyword spotting.

Keras: Will be used as a **high-level API** for TensorFlow to simplify the process of building and prototyping neural networks.

Audio Processing Libraries

FFmpeg: Will be used to convert audio files to a common bitrate and to perform batch **audio manipulation tasks** like trimming, merging and **normalizing** audio files.

Librosa: Will be used for **audio analysis** and manipulation like resampling audio to a uniform sample rate and trimming silence from the start and end of audio clips. It is used to convert an audio file to **spectrogram**.

Hardware

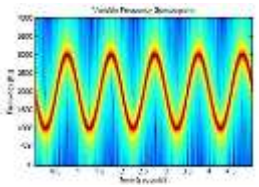
GPUs (Graphics Processing Units): Will be used for training ML models on **Google Colab** or an AI lab machine as they can handle high parallel processing workloads.

METHODOLOGY AND PROCESS

FLOW DIAGRAM

Dataset Preparation

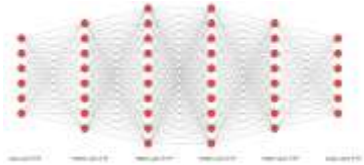
1) Open-source dataset utilization (like Common Voice)
2) Preprocessing



Model

1)Conversion of audio files into spectrograms
2) Input fed into a CNN

Training Process



Trained Model

Trained Model

Few Shot training

Fine-tune the model in a few-shot learning setup

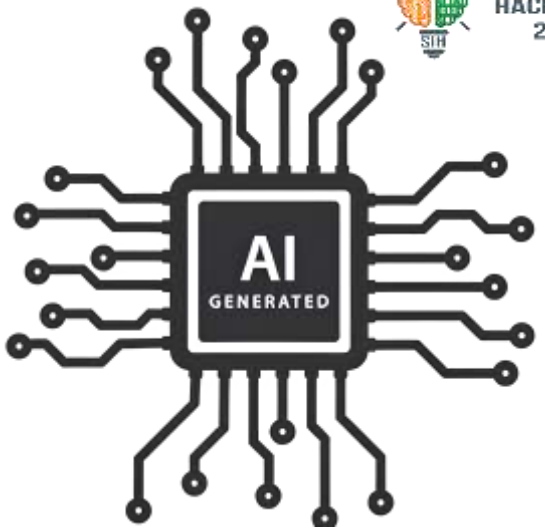
Few shot language agnostic keyword spotting system

Testing

Evaluate using metrics like accuracy, precision, recall, and F1-score

Technological stack

FEASIBILITY AND VIABILITY



Feasibility Analysis

- **Scalability**: The system can efficiently handle **multiple languages**, whether they were trained on the model or not.
- **Performance**: Previous iterations have obtained an **F1 score of as high as 0.75**.
- **Real-time Application**: Well-suited for real-time applications, such as **voice-activated assistants** or automated transcription services.
- **Market Viability**: Given the increasing demand for multilingual AI solutions, this system has strong potential for **commercial success**.

Problems

- **Limited Data Representation**
- **False Acceptance and Rejection Rates**
- **Streaming and Real-time Processing**
- **Cultural and Contextual Variability**
- **Bias and Fairness**
- **Integration with Existing Systems**

Solutions

- Use data augmentation techniques such as pitch shifting, noise injection, and time-stretching.
- Implement adaptive thresholding mechanisms that dynamically adjust the acceptance criteria based on the confidence level.
- Use lightweight and efficient architectures to ensure real-time processing capabilities without sacrificing accuracy.
- Develop context-aware models that take into account surrounding words or phrases.
- Implement bias detection and mitigation techniques, such as fairness-aware training algorithms or re-weighting methods.
- Design the system with flexible APIs and modular components that can easily integrate with existing platforms.

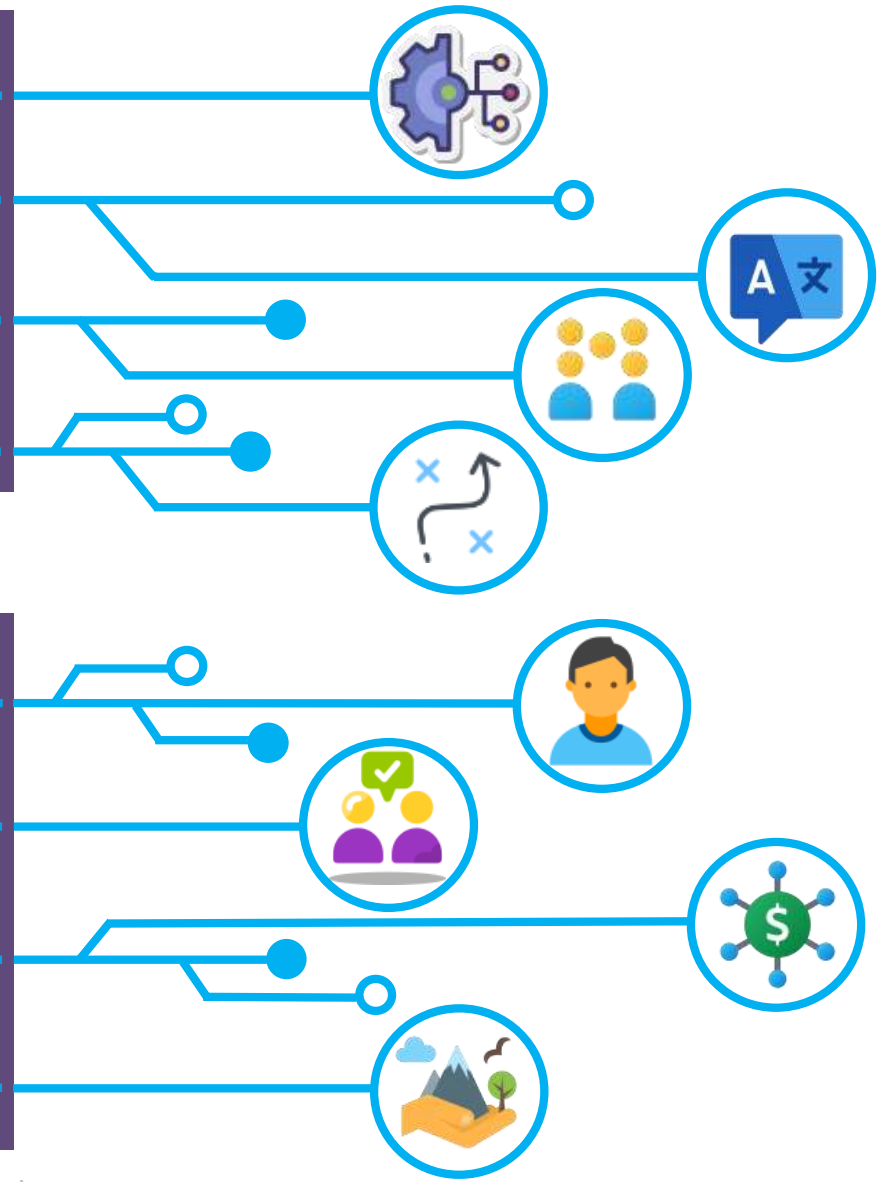
IMPACT AND BENEFITS

IMPACT

- Improved transcription technology by focusing on relevant sections.
- Generalization to new languages beyond the ones used in training.
- Repurposes general speech recognition datasets through forced alignment highlighting value of crowd-sourced data.
- Enhances logistics clustering by identifying key terms and helps in locating and integrating value-added services.

BENEFITS

- Keyword spotting recognizes user commands and enables seamless interactions.
- Bridges language barriers enabling smoother and faster communication.
- Reduces the need for extensive data collection unlike traditional keyword spotting models, potentially lowering costs.
- Avoids retraining, so, less energy intensive model updates: a more sustainable approach.



RESEARCH AND REFERENCES

References and Citation:

- Mazumder, M., Colby, B., Meyer, J., Warden, P., & Reddi, V.J. (2021). Few-Shot Keyword Spotting in Any Language. *Neural Information Processing Systems.* https://browse.arxiv.org/pdf/2104.01454v4
- Mazumder, M., Chitlangia, S., Banbury, C., Kang, Y., Ciro, J. M., Achorn, K., Galvez, D., Sabini, M., Mattson, P., Kanter, D., Diamos, G., Warden, P., Meyer, J., & Reddi, V. J. (2021). Multilingual Spoken Words corpus. *Neural Information Processing Systems*. https://openreview.net/pdf?id=c20jiJ5K2H
- TensorFlow, "TinyConv," https://github.com/tensorflow/ tensorflow/blob/master/tensorflow/examples/speech commands/ models.py, 2024
- Keras, "EfficientNet B0 to B7", https://keras.io/api/applications/efficientnet/, 2024
- HuggingFace, "google/speech_commands", https://huggingface.co/datasets/google/speech_commands
- TensorFlow, "Simple audio recognition: Recognizing keywords", https://www.tensorflow.org/tutorials/audio/simple_audio
- TensorFlow, "tensorflow/examples/speech_commands/test_streaming_accuracy.cc", https://github.com/tensorflow/tensorflow/blob/master/tensorflow/examples/speech_commands/test_streaming_accuracy.cc
- TensorFlow, "Audio recognition using transfer learning", https://codelabs.developers.google.com/codelabs/tensorflowjs-audio-codelab/index.html#0