National College of Ireland

“Hybrid Audio Event Recognition Using CNN and Transfer Learning: Comparative Study on Custom and Google Speech Commands Datasets”

MSc Research Project
MSCDAD_C

Hrushikesh Kumthekar
Student ID: 23313731

School of Computing
National College of Ireland

Supervisor: Eamon Nolan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Hrushikesh Nitin Kumthekar
Student ID:	x23313731
Programme:	MSCDAD_C **Year:** 2024-25
Module:	MSC Research Practicum
Supervisor:	Eamon Nolan
Submission Due Date:	11-08-2025
Project Title:	Hybrid Audio Event Recognition Using CNN and Transfer Learning: Comparative Study on Custom and Google Speech Commands Datasets
Word Count:	

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Hrushikesh Nitin Kumthekar

Date: 11-08-2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	

“Hybrid Audio Event Recognition Using CNN and Transfer Learning: Comparative Study on Custom and Google Speech Commands Datasets”

Eamon Nolan[1], Hrushikesh Kumthekar[2]
National College of Ireland, D01 K6W2, Ireland

ABSTRACT

The current project deeply analyses how data size emphasizes on reliability as well as reliability in field of keyword spotting (KWS) models for detection of distress words related audio circumstances. Phrases like “save me” or “help me” as well as gunshot shots were identified moreover is also vital for crucial for execution of real-time emergency response systems. Typically, deep learning models are usually effective when trained on smaller size datasets and how model performance is influenced by data scale and quality. Two datasets were used for this research. A custom-made dataset of approximately of size ~300 .wav audio samples were generated using text-to-speech tools and public sound libraries to simulate emergency situations. Concurrently, the large dataset Google Speech Commands (GSC) comprising of roughly 65,000 audio samples was used to benchmark the executed system. The implemented two CNN and pre-trained YAMNet models with classifiers like SVM were trained and evaluated across these two used datasets. Mel spectrograms being standard pre-processing technique was being applied in both cases thus revealing results which showed clear trend. The custom-made CNN achieved high to moderate accuracy across both datasets but ineffective to generalize correctly. Although model demonstrated moderate accuracy predictions were inaccurate frequently. On the other hand, GSC custom trained CNN model achieved higher accuracy and was able to produce more correct predictions which was reliable during real-time testing. This finding emphasizes the limitations of solely reliant upon evaluation metrics and highlights usage of deeper value of large, vast dataset in build-up of robust models. This end results suggest that quality and size of data not only impact accuracy but also actual prediction capabilities. Future work focuses on improvement in variability of small datasets and realism though augmentation techniques, collection of real-world audio samples and validating models within more complex and noisy environments to ensure dependable KWS system.

Keywords - Keyword Spotting, Dataset Size, Distress Detection, Emergency Audio, Deep Learning, Small Dataset, Model Reliability, CNN, YAMNet, SVM, Mel Spectrogram, Data Augmentation, Real-world Testing, Dataset Quality

1. INTRODUCTION

During recent years, speech recognition has seen keyword spotting (KWS) system as core field of research in particular for real-time audio detection circumstances. Keyword spotting (KWS) aims to detect one-shot keyword spoken word or phrases from continuous audio input, thus enabling voice-activation control in devices and also providing crucial functionality in

emergency response systems and safety. Existing work in field of KWS is vastly focused on large-scale, general-purpose datasets such as Google Speech Commands (GSC) which enables usage of complex architectural models like convolutional neural networks (CNN) and transformers which enables higher training accuracy. These models show consistent impressive performance when sufficient data is channel. However, gap related to KWS system in research in low-resource environments specifically where data input is small in size and highly specific like distress related words or phrases or data being collected under real-world constraints. This project examines this gap by determining how data size has overall impact on performance as well as generalization ability of KWS models. Generally, higher accuracy is common benchmark for model evaluation, it does not showcase always real-world alignment particularly when taken into account smaller datasets where overfitting is at risk. Project compares models trained with pre-trained on custom made dataset for distress related audio files and gunshots sound with large scale GSC dataset. The text-to-speech (TTS) was use to create custom dataset which had synthetic phrases like “help me” and “save me” and sound effects of gunshots thus totaling altogether ~300 samples. The custom trained CNN achieved accuracy of 78.26% but yet failed frequently to predict words correctly during real-time testing. Conversely, GSC model trained CNN achieved higher test accuracy 92.84% which produced remarkable results and reliable predictions in real-world scenarios. This imbalance forms crucial foundation for identifying role of data quantity and diversity in training robust KWS systems. The problem lies in application of this system in real world safety and emergency systems which makes it important. For instance, to monitor environments where signs of danger like shouting for help or gunfire must be able to detect such events accurately even when trained on small scale datasets. This is highly relevant in systems like smart homes, surveillance systems or safety wearable devices. Performance could be downgrade if such systems are built using small, handcrafted datasets when generalization and noise variability are not taken into account, thus their performance may be dangerously misleading. Moreover, institutions and public relying on such systems could affect by false alarms, worse missed alerts during actual emergencies. This research also addresses need for studying and contributing to growing need for more reliable and robust AI models for low-resource setting conditions, where amount of data is impractical and ineffective near to impossible. As in recent literature [1][2][7], KWS under insufficient dataset conditions is a recognized challenge also an unexplored in terms of practical implications. The central argument of this implemented project addresses the questions like: How data size affects the generalization, performance and reliability of keyword spotting models in emergency and distress audio events? Furthermore, to explore this question project is structured around the following objective: Investigate current approaches to KWS and role of dataset size in model training. Design and build custom keyword dataset on distress phrases and emergency sounds using Text-to-Speech and open-source audio of gunshot fire. Implement CNN and YAMNet-based models trained on both custom and GSC dataset. At last, evaluate model performance based on standard metrics like accuracy, confusion matrix and assess its real-world practical reliability. This project constrained by several limitations. Most significantly, the custom-made dataset is small in size and generated artificially lacking natural acoustic diversity of real-world speech. Whereas efforts were made to simulate natural sounds using higher quality TTS and sound effects, background noise, speaking styles, variations in speaker accents limits the generalizability of end results. To add that, comparison between models was constrained on basis on time and computational resources as cross-validation and full scale hyperparameter optimization were outside the project scope. It is also taken into account that input audio for testing as well as

training is pre-processed and free from majority of clutter, quality issues, with issues like clipping and excessive background noise interference. The dissertation is organized as follows: Chapter 2 represents the literature review of current KWS techniques, datasets and challenges with consideration of low-resource scenarios and emergency audio event detection. Chapter 3 emphasizes on methodology used in data collection, pre-processing, model designing and at last training workflows. On the other hand, Chapter 4, discusses the design and implementation of both experimental setups-custom and GSC based with their model architectures and their training configuration. Chapter 5 represents results which includes quantitative evaluation and qualitative analysis of prediction performance. Chapter 6 puts forward the limitations of achieved results their correlation with existing research in field. Finally, Chapter 7 conclusion of work, offers future research and directions such as need for real word data collection, noise robust training and deployment scenarios. As a part of project, Graphical User Interface (GUI) was also developed to replicate real-time testing of trained models. This GUI directly allows direct input from microphone, on device and decision to choose keyword spotting model with live display of predicted labels with their timestamps and confidence levels. By integration of user-friendly front-end with background models' system was evaluated in realistic scenarios thus folding gap between offline evaluation and practical deployment.

2. LITERATURE REVIEW

Real-time applications like voice assistants, low-power devices as well as emergency alert systems is core application of Keyword Spotting within research field of speech recognition. Overall goal of KWS system is to identify pre-defined sets of keywords from audio streams in an efficient and reliable manner with frequent constraints like limited processing power or to operate system within noisy environment. The primary objective of this project is to explore already work done on KWS, to understand implemented methodology used and their limitations when taken into account small, custom datasets. This literature directly associates to research question put forward in this dissertation: How data size has impact on effectiveness and reliability of KWS models notably in distress detection tasks? Recent studies have conducted thoroughly study on various KWS models ranging from CNN based architectural models to more advanced transformer-based attention-driven models. For instance, Momeni et al. [1] proposed a solution for audio-video keyword spotting method that utilizes both speech and visual aspect input to improve accuracy. Their study demonstrated outstanding results in controlled conditions but mandatory complex multi-modal data making it inferior in practical conditions where audio is only input emergency systems. On the other hand, Berg et al. [3] used Keyword transformer, a self-attention-based model designed for improved accuracy and varied context handling in KWS tasks. Foremost, such models are heavily dependent on large scale datasets and thus making them inefficient to work when working on small dataset. Several, other works have attempted to work with KWS with usage of small-scale dataset or in low-resource settings. Gok et al. [8] explored supervised learning models which were used on pre-trained model embeddings in few shots KWS application, thus showcased improved performance with use of minimal labelled data. However, when it comes to practical deployment complexity of training and fine-tuning such models can put barrier, especially for applications like emergency systems needing fast setup. Likewise, Rusci and Tuytelaars [15] proposed solution using Tiny KWS models with very few samples with on device customization on these models. Their work focused feasibility factor but still required heavy

hardware support and lacks focus on crucial application cases like distress words detection. Moreover, techniques like data augmentation and dataset design within KWS application have also been investigated. Herreilers et al. [9] implemented transformer embeddings in low-resource languages and found out contrastive learning helps but real-world application requires more robust generalization across different noise profiles. On the other hand, Garai and Samui [7] conducted a comprehensive review of small-footprint and efficient KWS and emphasized the trade-off between model size and accuracy, stressing importance of dataset diversity in deriving usable results. Most of existing studies have advanced in terms of model design and performance optimization, but lack critical application focus on distress-specific keyword spotting in small, targeted datasets. Most recent work consisting of keywords like "yes", "no" or "stop" which was trained on large public datasets like Google Speech Commands. Such high-quality datasets offer sufficient variety, speaker diversity with clean labelling thus supporting high performing models even when used relatively small-scale architecture [20]. But on the other hand, real-world emergency detection involves far more complex acoustic cues inclusive of urgent phrases or environmental sounds like gunshots. These scenarios are not covered within in mainstream datasets. Also, current literature often demonstrated high accuracy with reliability but overlooks situations where models perform during validation but showcases little to no promising results on real-word prediction. This is a key limitation this dissertation addresses in this paper. In this conducted study, a custom-made dataset consisting of distress keywords was created with help of text-to-speech and open-source audio libraries for gunshot fire samples [23] [24], which equates to total ~300 samples. Although models trained on these custom-made datasets showed promising accuracy of 78.26% but when deployed on real-time prediction tasks derived poor results on the other hand same model trained on GSC datasets, with an accuracy of 92.84% achieved dependable results, supported by stronger generalization in real-world application. This reveals a gap in the literature: as most of the attention is been given to model optimization and architecture design but influence of dataset size and diversity on real-world KWS reliability has been unexplored. Furthermore, few studies have focused on keywords like emergency specific or critical keywords which consist of unique acoustic and conceptual properties that general datasets are unable to achieve. To conclude, significant progress has been made to building effective and accurate KWS models but when taken into consideration use of large, well-labelled and diverse dataset. Few studies address issue of small-scale dataset and domain specific especially in context of safety critical applications. This dissertation aims to fill this gap by comparing performance across both CNN and YAMNet-based models trained on large scaled publicly available dataset and small-scale custom dataset. It emphasizes that reported accuracy and real-world deployment with argument of data size, quality and relevance are as well important if not then model complexity when building KWS for emergency applications. This work contributes to existing studies on low-resource speech recognition and offers suitable solution when real world deployment taken into account where data scarcity is normal.

3. RESEARCH METHODOLOGY

The main objective of this research is to find how dataset size has overall impact on performance, reliability and generalizability of keyword spotting (KWS) models when trained for detection of distress related audio cues. Implemented methodology was designed to be fully reproducible and documented systematically to ensure that other researcher can replicate entire process with no hassle with use of same tools and datasets. The research conducted two main

experimental tracks: one focused on a custom-built dataset which consists of distress related audio files and gunshot sounds and with that using GSC datasets for benchmarking purpose. Foremost step was to collect data. The cusom dataset was mix up of distress audio spoken words like “save me” and “help me” created using TTSMP3.com text-to-speech service [23] and gunshot fire sounds were sourced from Pixabay’s free sound effects library [24].

Ambulance	Danger	Emergency	Fire	Gunshot-sound	Gunshot-spoken
Help	Hostage	I am bleeding	I am hurt	I am scared	Please help
Police	Save me	Stop	Trapped		

Table 1.0 Keywords list in Custom Built Dataset

The fetched sound .wav files from these websites were classes into sub-folders where each folder (e.g., “help me”, “gunshot”) contained corresponding audio files. Distribution of this file was visualized using seaborn library to verify dataset balance. Properties like audio’s sample rate and duration were derived using python’s librosa library to ensure consistency across all samples.

Yes	Down	Left	On	Stop
No	Up	Right	Off	Go

Table 2.0 Used Keywords list from Google Speech Dataset

On the other hand, GSC dataset was uncompressed from its original format .tar.gz archive and was structured according to its actual class labels such as “yes”, “no”, “stop” and “go”. A subset of these classes was taken into consideration instead of using whole classes for training and class distribution was plotted of this class to analyze balance. Both of this dataset was standardize by resampling them into uniform sample rate of 16kHz to ensure comparability. Next followed by data preparation, Librosa library was used to visualize waveforms and mel spectrograms were generated for two datasets. Human speech is best represented when used Mel spectrograms because they effectively represent frequency patterns which is common step in speech and audio classification applications. These derived spectrograms were resized and normalized to ensure consistency and were fed to CNN models as input features. Implementation part was conducted on Python via Jupyter notebook which consisted libraries like Numpy, Pandas, librosa, Matplotlib, Seaborn, Sci-kit learn and Tensorflow. CNN models were built up using Tensorflow/ Keras with a structure comprising of convolutional layers, ReLU activations, max pooling with dropout layers to overcome issue of overfitting. For multi-class classification softmax layer was used. For evaluation purpose dataset was split across standard size 70% for training with 15% for validation and 15% for testing. Moreover, CNN was baseline model for both of the datasets with pre-trained YAMNet model were also tested on custom made datasets which is based on MobileNet. YAMNet takes audio waveforms as input and turns them into embeddings which can be passed to traditional classifiers like Support Vector Machine and Logistic Regression using Scikit-learn library. Such solution allowed for

hybrid pipeline where deep feature extractor (YAMNet) is combined with lightweight classical models which is useful for low-resource systems. Feature extraction using YAMNet consisted of segmenting each audio files into frames and averaging embedded vectors across duration to produce a fixed-size input vector. Derived embeddings were used to train classifiers which logs like accuracy, loss and confusion matrices. Confusion matrices helped for identifying misclassifications, to know whether models were simply overfitting or understanding classes boundaries. On the other hand, for measurement, primary metric was classification accuracy but confusion matrices and sample level predictions were critically important to understand prediction correctness. This was necessary as for custom dataset the high accuracy did not yield higher output in real-time predictions. For instance, the custom trained CNN yield an accuracy of 78.26% yet during interference it frequently misidentified inputs clips. Contrary, GSC-trained model, with an accuracy of 92.84% consistently predicted accurate labels with audio was fed during testing. The central theme in observation was divergence, it showed that smaller datasets cause inflated evaluation metrics due to overfitting or lack of representative test samples. Also, demonstrated that relying on solely on numerical accuracy without considering real world conditions makes model behave irregular. For the GSC dataset experimentation, CNN model was trained on Mel spectrograms from multiple classes which resulted higher generalization performance. This model was built upon same architecture parameters as custom dataset to ensure fair comparison. Same classification metrics were used to benchmark the results. Overall, having same methodology and comparing same models across different size dataset enabled investigation into how data volume and variety of data affects the real-world outcome. All code was fabricated according to CRISP-DM methodology within ipynb notebook including separate cells for data preprocessing, visualization, model training and evaluation thus ensuring reproducibility at end. This methodology reflects real world constraints where labelled audio data for uses like distress detection is scarce or unavailable. By implementing both synthetic and large-scale models, this study not only identifies influence of dataset size but also highlights critical impact when it comes to real world deployment of KWS systems.

4. DESIGN & IMPLEMENTATION

These two implemented parallel pipelines within designed project aim to evaluate influence of dataset size on keyword spotting (KWS) performance following comparative framework. One pipeline is based on GSC dataset while other pipeline is based upon custom made dataset containing distress-related voice commands with gunshots fire sounds. Both pipelines share same architecture thus ensuring consistency and fairness in comparison with variations within data volume its content and embedding strategies. This whole project follows CRISP-DM methodology in which modelling is central theme followed after objectives and data understanding with data preparation. The system architecture across both pipelines starts with similar approach to put forward audio pre-processing followed by feature extraction using Mel spectrograms for CNN or YAMNet embeddings for traditional machine learning models like Support Vector Machine or Logistic Regression. The core input for this project is .wav audio files. It is pre-processed to standardize its sampling rate and then converted into time-frequency representation (spectrogram) and then passed through either a deep learning model or feature extractor with classical classifier. The outcomes of this system are predicted keyword label representing either distress word phrase or command. Firstly, the custom dataset pipelines traverses through each class folder each one of those representing a keyword or sound category

and loads that .wav files using python's librosa library. All audio inputs files were converted to 16kHz to ensure consistency. Initially data understanding steps were performed like showing samples of numbers per each class, waveform structure with sample duration to ensure quality control. Mel spectrograms are then generated for each of the audio clips which then converted into image-like arrays suitable for CNN input and stored into arrays using NumPy.

$$X(m,\omega) = \sum_{n=-\infty}^{\infty} x[n] \cdot w[n-m] \cdot e^{-j\omega n}$$

Above formula of Short-Time Fourier Transform (STFT) decomposes audio into the overlapping time windows to reveal frequency content over time of period.

$$m = 2595 \cdot \log_{10}\left(1 + \frac{f}{700}\right)$$

The Mel Frequency Mapping formula transforms a frequency fff in hertz with its corresponding value on the Mel scale, which is way how humans perceive pitch.

$$\text{Log-Mel}(f,t) = \log(\text{MelSpectrogram}(f,t) + \epsilon)$$

The Log-Mel Spectrogram formula converts Mel spectrogram values into a logarithmic scale, thus enhancing feature representation by aligning more closely with human loudness perception.

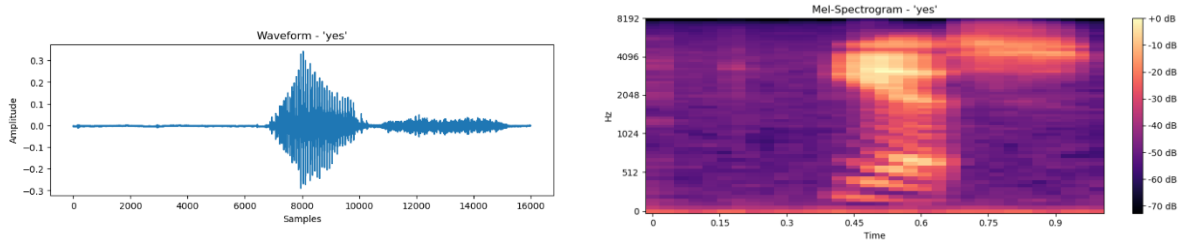

Fig 1.0 Waveform and Mel Spectrogram – "yes" (Google Speech Commands Dataset)

This figure 1.0 shows both the raw waveform and its corresponding Mel spectrogram for the keyword "yes" from GSC dataset. Waveform captures amplitude variations over time on the other hand mel spectrogram represents same signal in time frequency domain thus revealing clear spectral patterns. These features benefit from the dataset's high variability which is result of multiple speakers and diverse recording conditions.

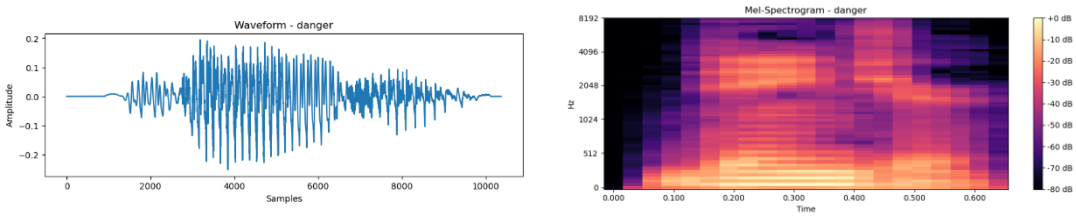

Fig 2.0 Waveform and Mel Spectrogram – "danger" (Custom Built Dataset)

Above figure 2.0 raw waveform as well as its corresponding Mel spectrogram for the keyword "danger" from custom distress audio. The derived waveform showcases amplitude pattern generated by text-to-speech synthesis, while the Mel spectrogram showcases its time-frequency representation. As compared to GSC dataset, the signal here shows lesser variability due to low number of speakers and controlled recording conditions. This performed Mel spectrogram transformation ensures that time frequency domain is leveraged for better keyword identification. The implemented CNN model used for classification in custom pipeline follows a standard architecture with two or three convolutional layers, each being followed by a ReLU activation function and max pooling. To overcome overfitting dropout layers were used and output was flattened before passing into dense softmax layer that outputs class probabilities. Model was compiled using categorical cross entropy as loss function and optimized with Adam optimizer. For CNN training was conducted over multiple epochs with validation on 15% of data typically 70% and rest of 15% for testing purpose. The performance was measured through accuracy scores, loss values with confusion matrices. A secondary pipeline on custom dataset involved use of YAMNet, a pre-trained audio embedding model based on MobileNet architecture. Afterwards, raw audio waveforms are passed via YAMNet, which outputs 1024-dimensional feature embeddings. This further are averaged out over duration of audio file to produce fixed length vectors. The output features are then feed to classical classifiers like SVM and Logistic Regression using Scikit-learn. Computed models were evaluated on basis of their accuracy and confusion matrices. Core reason for implementation of pre-trained embeddings were to test whether this can compensate for lack of dataset volume. In GSC pipeline, dataset is extracted from its compressed archive and relevant classes such as "yes", "no", "down", "left", "up", "right", "on", "off", "stop", "go" were selected. Each class contained up to thousands of samples which were loaded, resampled and visualized to understand waveform patterns. On the other hand, for custom pipeline, Mel spectrograms were computed using librosa library, which were then normalized and resized for compatibility with CNN input. The CNN of both custom dataset and GSC is typically same comprising of multiple convolutional layers, pooling with dropout. The model is trained on same configuration parameters, including optimizer, batch size and loss function. Fig 3.0 is complete workflow of project.

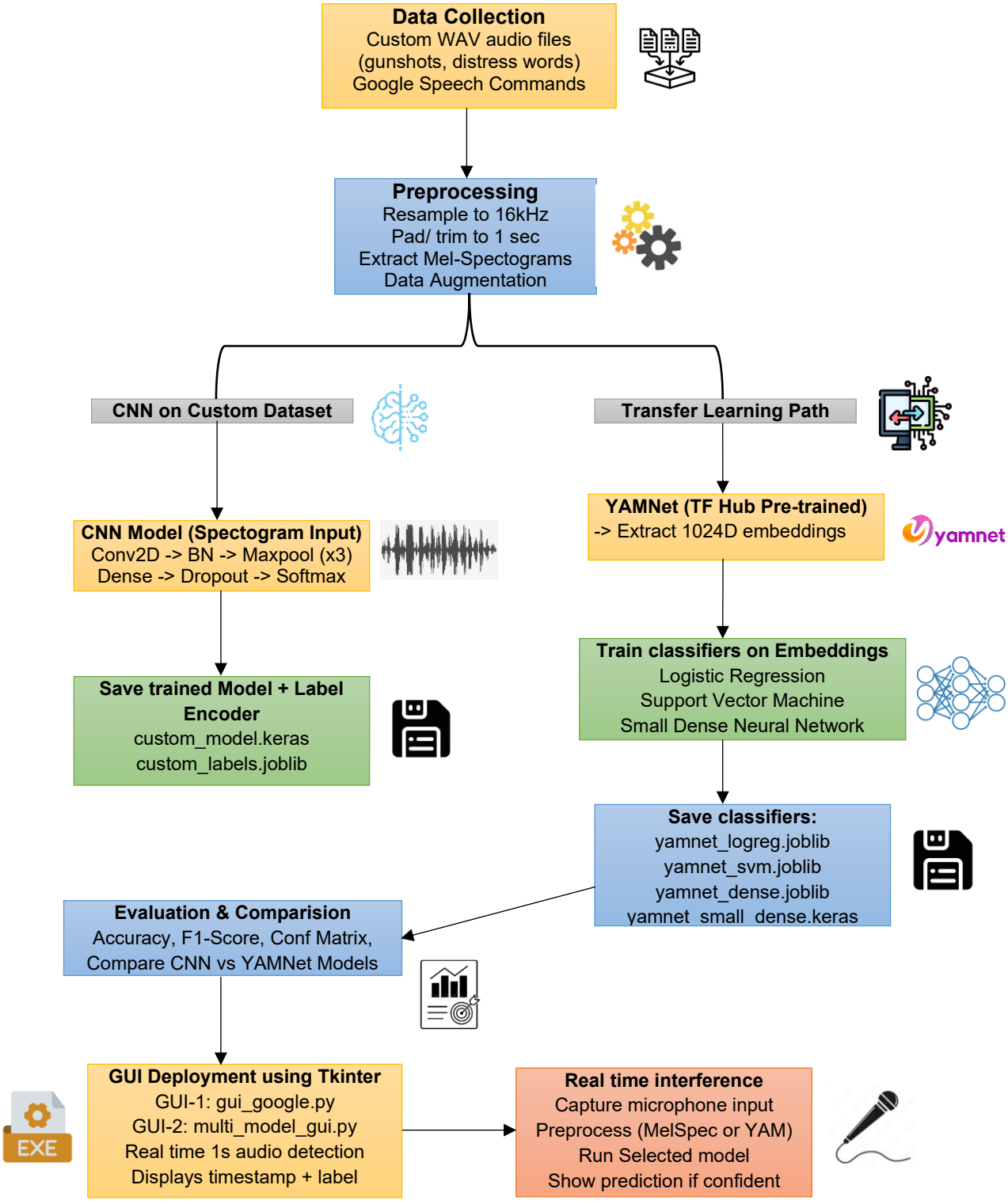

Figure 3.0 Complete Architecture of KWS project

To make sure transparency and reproducibility exact model architectures and hyperparameters with configurations used in both pipelines are summarized below for better understanding. The

Convolutional Neural Network (CNN – Mel Spectogram Input) architecture processes Mel spectrograms as 2-D images to enable to model learn spatial patterns in time frequency domain. The CNN network consists of three convolutional layers with increasing filter counts (32, 64, 128) and kernel of size 3x3, each being followed by ReLU activation to introduce non-linearity. 2x2 Max pooling layer futher reduces dimensionality after each convolution in order to preserve dominant frequency features with dropout layers to prevent overfitting. The resulted feature maps are then flattened and connected to a dense softmax layer to result output class probabilities. The model was trained on Adam optimizer with learning rate of 0.001 and batch size of 32 and for 30 epochs with categorical cross-entropy as loss function. YAMNet + Logistic Regression model uses pretrained YAMNet architecture, which is based on MobileNet and trained on large-scale Audioset to extract high level audio embeddings. With each audio clip split across frames, processed by YAMNet to generate 1024-dimensional embeddings which then averaged out to produce single fixed length feature vector. With L2 regularization of C1=1.0 is then trained on these derived embeddings. Solver is set to 'lbfgs' with maximum of 1000 iterations to ensure convergence. This keeps classifier lightweight for low-resource deployment thus benefit from transfer learning. In case of SVM model, embeddings from YAMNet served as feature set, with use of Radial Basis Function (RBF) kernel handled nonlinear decision boundaries in the 1024-dimensional space. In SVM regularization parameter C was set o 1.0 and kernel coefficient gamma was set to 'scale' for automatic adjustment based on input dimensionality. Model then was optimized to balance class separation, though its more prone to data imbalance and noise when training data is limited. Also, for YAMNet + Small Dense Neural Networks YAMNet embeddings were used as input into a fully connected feedforward neural network. The architecture followed two hidden layers with 128 and 64 units specifically which was further follow up being by ReLU activation for non-linearity. Dropout Layer at rate of 0.3 were applied between hidden layers to reduce overfitting. Softmax being output layer for multi-class probability estimation with network being trained on Adam optimizer with learning rate at 0.001 with batch size of 32 for 50 epochs. This showcases power of YAMNet embeddings with adaptability of small deep neural networks. These parameters were kept same across datasets to ensure valid comparison allowing performance differences based on dataset size rather than model hyper tuning. Moreover, key difference lies in dataset volume and variability of training data which impacts stability and generalization of model. In case of GSC model, training logs with results consistently demonstrated smoother learning curve with better convergence and more reliable performance on test set (92.84% accuracy and 0.84 F1 score). The project was completely implemented on Python using Jupyter Notebooks. For deep learning, TensorFlow and Keras were utilized to build up and train models. Librosa library from python was utilized for audio processing, including waveform analysis, resampling and Mel spectrogram feature extraction. On the other hand, Scikit-learn was used for traditional classifiers and evaluation metrics such as classification reports and confusion matrices were used. Visualization libraires like Seaborn and Matplotlib were used to plot distributions, waveforms with performance graphs. Project was implemented considering less focused on model complexity and steady focus on input diversity. This aligns with core objective of project to determine whether dataset size affects performance of model rather than model architecture itself. In nutshell, both pipelines followed consistent CNN structures and identical training routines allowing for apple-to-apple comparison. On contrary, custom dataset models often achieved high accuracy scores 78.26% with 0.77 F1 score for CNN; 55.73% accuracy with 0.53 F1 score for YAMNet + Logistic Regression; 49.2% accuracy with 0.49 F1 Score for YAMNet + SVM and at last 54.1%

accuracy with 0.51 F1 score for Small Dense Neural Network, considering this results model performed poorly when tested with unseen samples exposing risk of overfitting due to limited availability of data diversity. To sum up, the implementation was performed on one small scale dataset and on the other hand one large public scale dataset. Both were evaluated under identical metrics. The methodology confirmed that larger datasets outputs to more robust and reliable keyword spotting systems, even when using relatively simpler models. The practical results are clear which emphasizes that in application like distress audio detection data quality and variety are often more impactful than architectural sophistication. In addition to model implementation further project was advanced to GUI based implementation using Tkinter-based graphical user interface which enable interactive and real-time keyword spotting system. Different two GUI system was implemented for custom dataset and GSC dataset. Multi-Model GUI supporting multiple backends including CNN (Mel spectrogram) Transfer Learning models YAMNet with Logistic Regression, Support Vector Machine. Lastly Dense Neural Networks for custom dataset. In case of GSC dataset, a single streamlined solution was made solely consisting of implemented CNN model. Both systems capture one-second of audio segments from inbuilt pc microphone which then are process using Mel spectrograms extraction or YAMNet embedding generation. These features are the passed on to selected model for interference. Predictions along with timestamps, model names and confidence levels are displayed across screen. GUI system runs interference in backend to main steady responsiveness and supports continuous detection. This designed allowed models to being trained on stress-related audio events where conditions change like background noise or speaker and pronunciation. GUI serves as both deployment prototype and qualitative testing tool.

5. RESULTS AND ANALYSIS

This section represents the final outcome of our experimental pipelines develop in this conducted study. The first pipeline used the custom-made dataset with 300 audio files which holds distress words with a combination of gunshot fire sound. The second pipeline utilized Google Speech Commands (GSC) dataset, which consist over 65,000 labelled samples from 35+ classes. Both pipelines were executed on same baseline architecture and training procedures to allow for a fair comparison of the effects of dataset volume on model's performance, generalization and real-world usability. The CNN executed on custom made dataset achieved an accuracy of 78.26% which is indicating promising results considering such a small dataset. The corresponding analysis of confusion matrix Fig 4.0 revealed that model performed well on distinguishing certain classes although it had phonetically similar categories like "help me" and "save me". Meanwhile CNN model trained on GSC dataset achieved superior accuracy of 92.84% as shown in Fig 5.0. However, it demonstrated superior prediction capability in real-world deployment. Despite having such accuracy and trained on CNN model it produced confident and consistent predictions when new samples were taken into account especially those were not included in training. The reported accuracy and actual outcome of model is one the most important finding of this conducted study. It emphasizes the hypothesis that volume and variability is critical that raw performance metrics in real-world deployment.

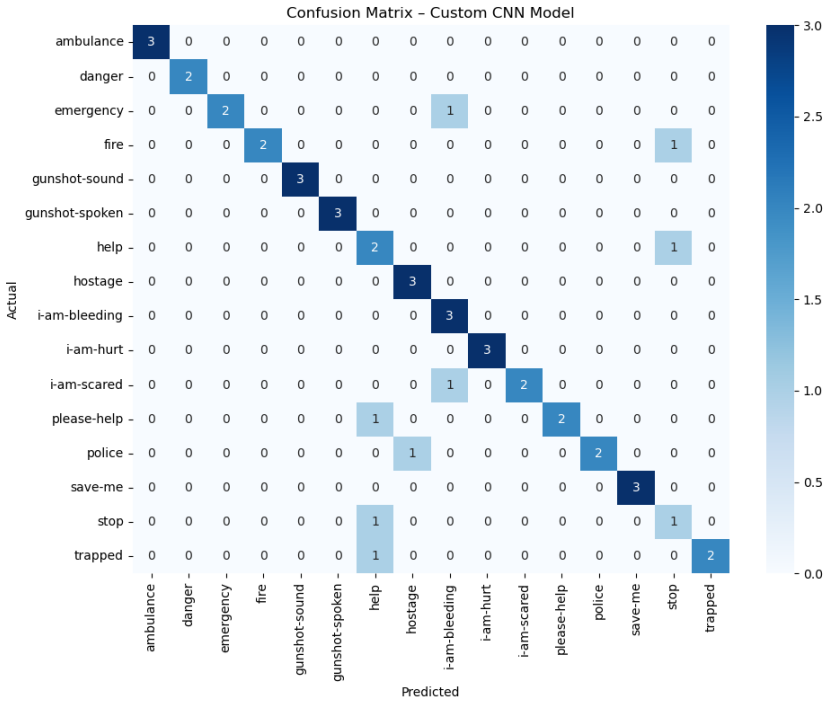

Figure 4.0 Confusion Matrix – Custom CNN on Custom Built Dataset

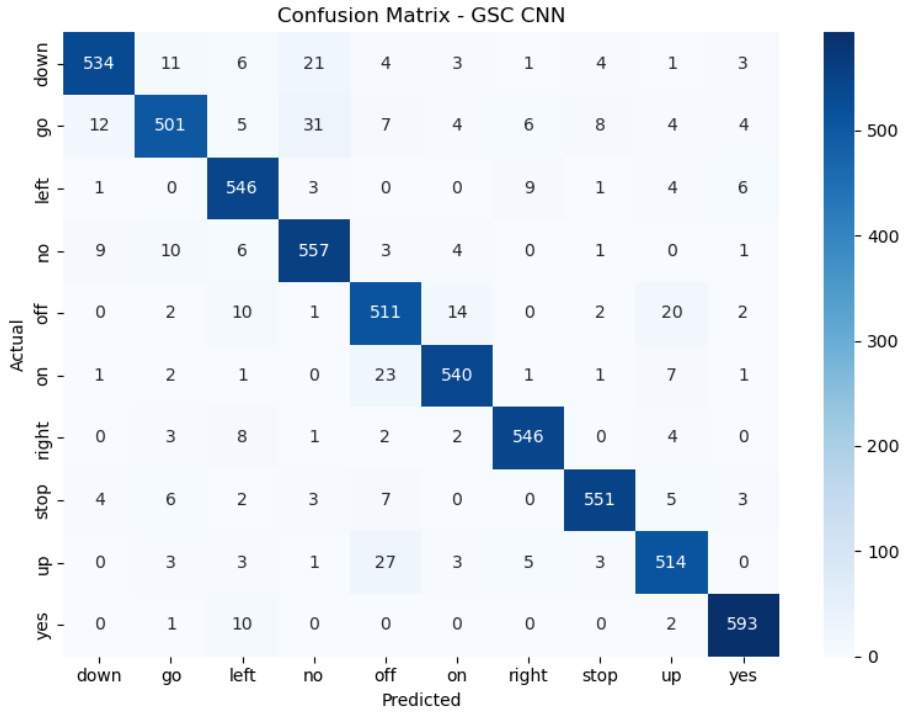

Figure 5.0 Confusion Matrix – Custom CNN on Google Speech Dataset

Model Type	Dataset	Test Accuracy	Avg. F1 Score	Generalization (Observed)	Notes
CNN	Google Speech Dataset	**92.84%**	**0.84**	*Strong*	Baseline
CNN	Custom made audio dataset	**78.26%**	**0.77**	*Moderate*	Baseline
YAMNet + Logistic Reg	Custom made audio dataset	**55.73%**	**0.53**	*Poor*	TL (shallow)
YAMNet + SVM	Custom made audio dataset	**49.2%**	**0.49**	*Poor*	TL (classical ML)
YAMNet + Dense NN	Custom made audio dataset	**54.1%**	**0.51**	*Poor-Moderate*	TL (deep)

Table 3.0 Comparative Model Performance Summary

The executed CNN model reached higher training accuracy and maintained a high validation accuracy during training. Moreover, the widen gap within training and validation in later on epochs showcased overfitting. On contrary, the GSC trained CNN showed slower although stable convergence with closer alignment between training and validation curves, thus demonstrating better generalization. The outcome showcased that CNN model performed well in controlled evaluations, it lacked robustness. Factors like background noise, voice and speed downgraded the prediction accuracy in real-world scenarios. On the other hand, GSC model's more moderate accuracy was complemented by its strong resilience to ongoing audio variability thus highlighting real-world value on data diversity. The custom YAMNet + SVM pipeline further support these findings. Although it reached accuracy of 49.2%, it suffered from behaviour like inconsistent predictions. Unlike CNNs which learns spatial patterns in spectrograms, the SVM classifier with YAMNet embeddings showed instability with small shifts within audio input likely due to lack of intra-class variation on training data. As positive outcomes, the conducted study successfully showcased that even basic CNN model have higher performance on smaller datasets. The implemented pipeline was modular also reusable and efficient which was built upon open-source tools such as TensorFlow, Scikit-learn and Librosa. In terms of resource use and speed both models were being lightweight in nature and well suited for deployment on low resource hardware. Moreover, several limitations with challenges were also observed like custom dataset was synthetic, generated from TTS and open-source sound libraries. Despite normalization its acoustic variability was low. Overfitting was central findings in custom model where it showed higher accuracy in validation and on the other hand poor performance in real world deployment. The speaker diversity lack in custom dataset restricted model's ability to generalize to unheard audio. Augmentation strategies were not

inclusively used which could have overcome limitations of small dataset. This finding supports issue that custom models performed well worse in real world deployment despite looking better on paper. This project used multi-metric evaluation strategy that took into account: Accuracy – the percentage of correctly predicted samples; Confusion matrices – visual representation of misclassification while Prediction confidence observed distribution of predicted probabilities and Real-world deployment showcased model's behaviour when exposed to completely unseen audio inputs. The core used metrics were commonly used in recent KWS research [1] [3] [8] [9]. From point of view of comparison GSC CNN's F1 score of 0.84 closely replicates outcomes from well conducted research as seen in KWS benchmarks [18] [20]. Contrary, these findings showed that pre-trained models like YAMNets are highly dependent on data quality and quantity without sufficient variation in dataset even strong feature extractors cannot generalize effectively. Also, findings support and confirm what recent papers have highlighted: data often trumps more complex models [7] [8] [9]. A simple CNN trained on rich dataset with 65,000 samples outperformed a way better hybrid pipeline trained on 300 samples even though it used deeper embeddings and theoretically superior feature space. Without extensive use of statistical tests such as t-tests were not considered due to dataset constraints, consistent trends across multiple runs, models and architectures indicated superior robustness in conclusion. If scaled further enough future work could apply cross-validation and confidence interval analysis to reinforce these findings. The overall result strongly reinforces the core thesis that dataset size plays crucial role in determining KWS model's reliability even when smaller models are taken into account. This emphasized that real-world performance is not always in measurement of accuracy metrics, especially when conducted on small, homogenous test sets. This highlights the misalignment need for practical validation protocols that go beyond test accuracy. As we explore further discussion, several key themes emerged. First, data diversity has crucial role in effectiveness of keyword spotting systems especially in emergency and safety-critical applications. Secondly, synthetic datasets, while useful for rapid prototyping are ineffective without real-world acoustic variability as they not capture complexity and unproductiveness of actual environmental conditions. Finally, it is essential to validate models not solely on common metrics but also real-world deployment where environment varies. To ensure reported performance translate into practical reliability. The GUI implementation was used to perform live testing of each trained model. This approach allowed to answer qualitative assessment of prediction stability, latency with robustness under variation of real-world conditions. For instance, GUI logs revealed that custom made CNN on custom made dataset reported misclassified similar sounding phrases when spoken in unfamiliar voice despite higher testing accuracy. On the other hand, model trained on GSC maintained consistent predictions even in noisy environments. This live observation supports wider assessment which supports conclusion that larger, more diverse dataset yield more reliable keyword spotting performance. These findings will be further discussed in next section where we assess the strengths and limitations of models with how this conducted study contributes to broader field of keyword spotting in constrained data scenarios. Fig 6.0 is real time deployment of Google Speech dataset KWS system comprising of buttons like START, STOP with addition of Time and Detected keywords while Fig 7.0 demonstrates use of custom dataset-based models which has similar results like START, STOP also Model selection with found Keyword and Confidence level.

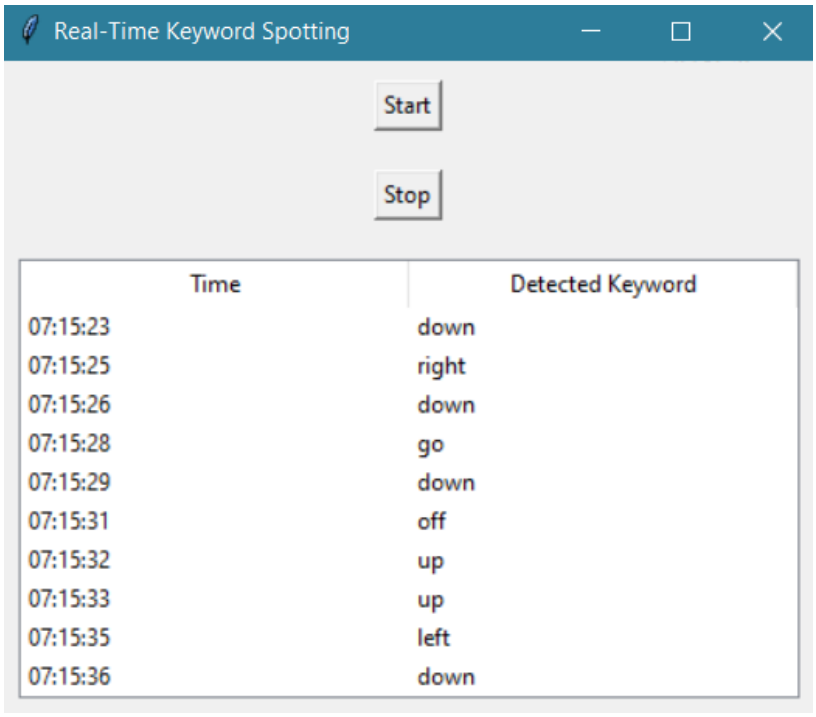

Figure 6.0 Real-time Keyword Spotting for Google Speech Dataset

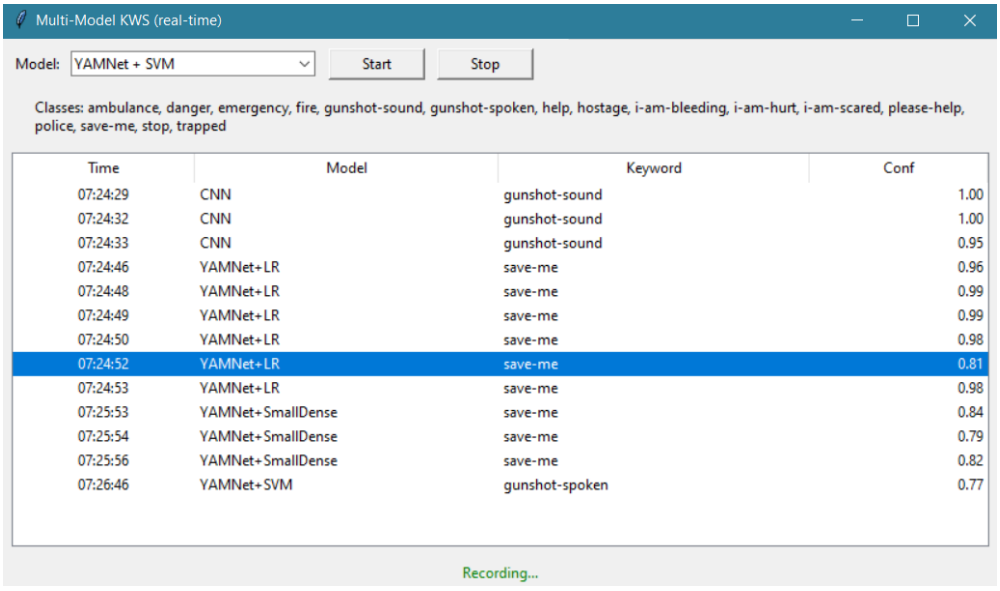

Figure 7.0 Real-time Keyword Spotting for Custom Built Dataset

6. DISCUSSION

The overall aim of the conducted study is to determine how dataset size influences the reliability, performance and generalizability of keyword spotting (KWS) systems. We developed two separate pipelines one trained on custom based distress audio datasets which

consists of ~300 audio samples and another on the Google Speech Commands (GSC) dataset with over 65,000 samples. Thus, fully focused on impact on data volume while keeping modelling techniques largely constant. The conducted experiment used Convolutional Neural Networks (CNNs) and YAMNet embeddings with classical classifiers across custom based dataset. The outcomes showcased revealing insight: a model's high accuracy on test dataset does not always correlate with its real-world performance. In fact, the custom dataset with accuracy of 78.26% failed to generalize with new inputs samples that were only slightly differed from its training data. Audio clips such as minor pitch variations or background noise frequently resulted in incorrect predictions. This outcome unfolded the fragility of models when trained on limited and synthetic data even when evaluation metrics appear appealing. On contrary, GSC trained model with test accuracy of 92.84% demonstrated far better real-world reliability. GSC managed to consistently better correct predictions across a wide variety of inputs. This demonstrates that model behavior is more dependent on dataset diversity rather than model architecture alone. Considering executed methodology, project successfully produced a reproducible workflow using open-source tools with standardized audio pre-processing techniques like Mel spectrograms and consistent CNN design across both datasets. The implementation of transfer learning on improving custom dataset further improved comparative depth of the study. However, it became proved that tools cannot compensate for lack of dataset diversity. When used small size datasets even sophisticated models fails to extract robust generalized features. Overall, method was full proof for comparative analysis but limited experimental range due to dataset constraints. While models worked accurately in isolated validation scenarios, their practical usability was reveled during broader testing highlighting critical gap that many machine learning studies overlooked. This study findings are strongly aligned with growing body of literature which puts importance of large, varied dataset over complex model architectures. For example, Chiu et al. [8] demonstrated that small-footprint models perform competitively when proper trained on rich datasets in keyword spotting. Also, Tang et al. [9] highlighted that transformer-based models can only surpass CNNs when paired with sufficient large and diverse rich training samples. This study supports these conclusions. Although, exploring both CNNs and feature rich embeddings from YAMNet, the custom trained models consistently underperformed in real world deployment. The core issue was neither model design or model selection it was data insufficiency. This emphasizes findings from studies like [7] and [10] that pre-trained models and advanced learning algorithms provide diminishing returns in low-data regimes. Moreover, the success of GSC trained model aligned with benchmark studies like Warder [1], who explored strong keyword detection using CNNs trained on GSC. While many researchers have tried to build better models through novel architectures or with loss functions. This study reinforces the idea that data is the ultimate key in small sample learning scenarios. One of the unexpected findings in this conducted study was inverse relationship between reported accuracy and real-world performance. Typically, higher accuracy tends to demonstrate better model behavior. However, in the case of custom dataset this was not matter. The discrepancy can be put forward to overfitting. Model memorized patterns in the custom training dataset and validation split but failed to produce results in real world deployment. Minor changes such as text-to-speech engine or slight change in background noise disrupted model's ability to classify correctly. In such scenario, accuracy score becomes misleading, suggesting strong performance but still robustness was actually lacking. Another unexpected result involved the YAMNet embeddings. These high-dimensional features extracted using MobileNet like architecture trained on AudioSet. While this improved SVM performance on GSC dataset, although it had limited

value on custom based dataset. This again emphasizes reality of embeddings are only good as variability and representativeness of input data. Since our custom dataset lacked diverse acoustic conditions, the embeddings which were extracted was too homogenous to add any meaningful distinction for classification. This supports our outcomes that dataset characteristics often overpower model complexity in practical KWS applications. While model hyper tuning and architectural experiments are common in literature, this study emphasizes that real world reliability is determined upstream at the data level. On the contrary, the GSC based results are widely applicable. GSC dataset includes samples from multiple speakers, environments and devices. Models trained on such kind of data showed robustness and reliability, and the pipeline used on this dataset can be adapted to other large scale available datasets. On the other hand, custom based dataset findings are limited in generalizability. Since, audio was generated using synthetic methods which lacked speaker diversity and it does not represent the variety of multiple different real like key scenarios occurred in distress situations. As such, the models trained on such kind of data are likely to underperform beyond this scope. However, lessons learned from these datasets are widely applicable. These findings underscore the importance of ensuring data diversity, incorporation speaker and vary background with real world testing apart from validation set and recognizing where high evaluation metrics may mask overfitting in actual model behavior. This revealed insights are highly applicable to any domain where small datasets are used in fields like healthcare, security or embedded systems with fewer training samples. The strengths of conducted study comprise of reproducible pipeline, with both model pipelines employing consistent pre-processing, architecture and evaluation metrics. The incorporation of real-world testing also demonstrates its importance in analysis under realistic variations. The cross comparison of both CNNs and YAMNet models with classical classifiers provided deeper insights in impact of data size. Finally, the study also maintained consistent evaluation approach focusing not only on accuracy but also on model's behavior and prediction confidence. Addressing weakness of conducted study include use of synthetic dataset which consist of limited speakers and minimal background variation, real recorded voice could have improved realism. Additionally, no cross-validation was performed as single validation splits may not fully reflect behavior of models trained on small datasets. Without data augmentation techniques like pitch shifting, noise injection or speed perturbation was likely to contribute overfitting. Moreover, limited statistical testing was carried out due to small sample size prevented application of significance tests. Although these limitations, offers realistic portrayal of challenges faced with working with minimal data size and underscores why large-scale datasets are not merely advantageous but also essential for tasks like keyword spotting. By enabling direct input from microphone, and live display of predictions the implemented GUI allowed qualitative insights that numerical metrics alone could not provide. For example, detection logs in real time deployment of custom based dataset produced unstable or fluctuating predictions for repeated inputs while GSC trained models remained consistent. The findings supported that importance of real-world deployment in keyword spotting research. The GUI bridges this gap by offering to identify weakness that may only emerge under live conditions thus bridged gap between laboratory results and practical deployment of system. In conclusion, the discussed points revealed that model reliability in keyword spotting is not guaranteed with accuracy alone. When data is limited model may perform well in validation sets yet fails dramatically when deployment for real time usage. Conversely, model trained with large scale dataset with feature rich audio like Google Speech Commands set can achieve higher accuracy while proving dependable, robust predictions in varied environments. Conducted study strongly supports the notion that diversity and size of dataset are key

foundation in safety critical applications like distress audio detection. The limitations of this project include use of synthetics dataset with absence of cross validation, which can be future work. While its methodological consistency and honest evaluation make it valuable contribution to the domain of speech recognition research where data is limited.

7. CONCLUSION

This thoroughly research investigates how data size and quality has influence on performance as well as generalization capabilities of keyword spotting (KWS) models, particularly in safety-critical applications like distress audio detection. The primary research question explored whether custom made dataset could produce a reliable KWS model or if larger scale data was required to achieve real world results. To answer this question, two parallel pipelines were executed, first pipeline consisting of custom-made dataset which comprised of 300 audio files including distress words and gunshots sounds. The second was Google Speech Commands (GSC) dataset, which contained over 65,000 labelled audio samples. Both pipelines were consistent with modelling techniques primarily Convolutional Neural Networks (CNNs) and in some experiments, pre trained embeddings from YAMNet combined along classical classifiers like Logistics Regressions and SVMs. Each dataset pre-processed audio to extract Mel spectrogram features. Models were trained and evaluated using standard classification metrics such as accuracy, F1 score and confusion matrices. Additionally, both pipelines were tested on real world audio inputs to compute generalization and reliability beyond test set. The project successfully met its objective by completing end-to-end model development with evaluation and their comparison. It answered research question with proper evidence: although models trained on small datasets can show high accuracy on test sets, they frequently fail under real world conditions. Custom based dataset CNN model achieved score of 78.26% accuracy but could not classify slightly varied or noise inputs. On contrary, GSC trained CNN with test accuracy score of 92.84% performed far superior reliably in practice, confidently and correctly predicting real-world audio inputs. Additionally, the transfer learning experiments on the custom-made audio dataset YAMNet + Logistic Regression model achieved 55.73% of accuracy with an average F1 score of 0.53 showcasing poor generalization despite using pre-trained embeddings. The YAMNet + SVM achieved slightly lower accuracy of 49.2% and F1 score of 0.49 indicating poor generalization. On the other hand, YAMNet + Dense NN yielded 54.1% of accuracy with an F1 of 0.51, reflecting poor-to-moderate generalization. These results proved that when working with small scale dataset transfer learning approaches fail to overcome lack of data variability and volume. The overall findings are that dataset size and diversity are foremost influential rather than architectural choice when choose to build robust KWS system. Even simpler CNNs outperform transfer learning models when use varied data. This supports previous studies suggesting that model alone does not compensate for data scarcity. For future work, several directions are suggested. First, enhancing custom dataset with real recordings from number of multiple speakers and diverse backgrounds which could demonstrate realism. Secondly, incorporating advanced data augmentation strategies like pitch shifting, noise injection or voice conversion which could improve generalization without further requirement of more data. Finally, addition of transfer-based learning from larger pre-trained models or experimenting with transformer-based architectures to strike balance between data efficiency and model performance. To sum up, this conducted study highlights the overlook topic in machine learning: good data beats clever models. In keyword spotting and similar tasks, investing in such datasets will likely yield greater improvements than simply

focusing on model architecture. This project found that dataset size and diversity has greater impact on real world reliability of keyword spotting models than reported accuracy and model complexity. The primary aim of this project was to investigate how data size affects performance, generalizability and reliability of KWS systems in safety critical use cases. This comparison of results of two pipelines confirms this.

REFERENCES

[1] L. Momeni, T. Afouras, T. Stafylakis, S. Albanie, and A. Zisserman, "Seeing wake words: Audio-visual Keyword Spotting," 2020, arXiv. doi: 10.48550/ARXIV.2009.01225.

[2] R. Vygon and N. Mikhaylovskiy, "Learning Efficient Representations for Keyword Spotting with Triplet Loss," in Speech and Computer, vol. 12997, A. Karpov and R. Potapova, Eds., Cham: Springer International Publishing, 2021, pp. 773–785. doi: 10.1007/978-3-030-87802-3_69.

[3] A. Berg, M. O'Connor, and M. T. Cruz, "Keyword Transformer: A Self-Attention Model for Keyword Spotting," in Interspeech 2021, ISCA, Aug. 2021, pp. 4249–4253. doi: 10.21437/Interspeech.2021-1286.

[4] C. Cioflan, L. Cavigelli, and L. Benini, "Boosting keyword spotting through on-device learnable user speech characteristics," 2024, arXiv. doi: 10.48550/ARXIV.2403.07802.

[5] Y. Xi, H. Li, B. Yang, H. Li, H. Xu, and K. Yu, "TDT-KWS: Fast and Accurate Keyword Spotting Using Token-and-Duration Transducer," in ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, Korea, Republic of: IEEE, Apr. 2024, pp. 11351–11355. doi: 10.1109/ICASSP48485.2024.10446909.

[6] P. Jakuš and H. Džapo, "Implementing Keyword Spotting on the MCUX947 Microcontroller with Integrated NPU," 2025, arXiv. doi: 10.48550/ARXIV.2506.08911.

[7] S. Garai and S. Samui, "Advances in Small-Footprint Keyword Spotting: A Comprehensive Review of Efficient Models and Algorithms," 2025, arXiv. doi: 10.48550/ARXIV.2506.11169.

[8] A. Gok, O. Buyuksolak, O. E. Okman, and M. Saraclar, "Enhancing Few-shot Keyword Spotting Performance through Pre-Trained Self-supervised Speech Models," 2025, arXiv. doi: 10.48550/ARXIV.2506.17686.

[9] J. Herreilers, C. Jacobs, and T. Niesler, "Low-resource keyword spotting using contrastively trained transformer acoustic word embeddings," 2025, arXiv. doi: 10.48550/ARXIV.2506.17690.

[10] B. Liu et al., "A 3.8- μ W 10-Keyword Noise-Robust Keyword Spotting Processor Using Symmetric Compressed Ternary-Weight Neural Networks," IEEE Open J. Solid-State Circuits Soc., vol. 3, pp. 185–196, 2023, doi: 10.1109/OJSSCS.2023.3312354.

[11] Z.-E. Wu, S.-J. Chan, Y. A. Wubet, and K.-Y. Lian, "DLiGRU-X: Efficient X-Vector-Based Embeddings for Small-Footprint Keyword Spotting System," IEEE Access, vol. 13, pp. 23498–23507, 2025, doi: 10.1109/ACCESS.2025.3536470.

[12] M. Luo, D. Wang, X. Wang, S. Qiao, and Y. Zhou, "Error-Diffusion Based Speech Feature Quantization for Small-Footprint Keyword Spotting," IEEE Signal Process. Lett., vol. 29, pp. 1357–1361, 2022, doi: 10.1109/LSP.2022.3179208.

[13] P. H. Pereira, W. Beccaro, and M. A. Ramírez, "Evaluating Robustness to Noise and Compression of Deep Neural Networks for Keyword Spotting," IEEE Access, vol. 11, pp. 53224–53236, 2023, doi: 10.1109/ACCESS.2023.3280477.

[14] M. Pudo, M. Wosik, and A. Janicki, “Improved Small-Footprint ASR-Based Solution for Open Vocabulary Keyword Spotting,” IEEE Access, vol. 12, pp. 91289–91299, 2024, doi: 10.1109/ACCESS.2024.3421605.
[15] M. Rusci and T. Tuytelaars, “On-Device Customization of Tiny Deep Learning Models for Keyword Spotting With Few Examples,” IEEE Micro, vol. 43, no. 6, pp. 50–57, Nov. 2023, doi: 10.1109/MM.2023.3311826.
[16] D. Seo, H.-S. Oh, and Y. Jung, “Wav2KWS: Transfer Learning From Speech Representations for Keyword Spotting,” IEEE Access, vol. 9, pp. 80682–80691, 2021, doi: 10.1109/ACCESS.2021.3078715.
[17] K. R, V. K. Kurmi, V. Namboodiri, and C. V. Jawahar, “Generalized Keyword Spotting using ASR embeddings,” in Interspeech 2022, ISCA, Sep. 2022, pp. 126–130. doi: 10.21437/Interspeech.2022-10450.
[18] P. Warden, “Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition,” 2018, arXiv. doi: 10.48550/ARXIV.1804.03209.
[19] T. Khan, “A Deep Learning-Based Gunshot Detection IoT System with Enhanced Security Features and Testing Using Blank Guns,” IoT, vol. 6, no. 1, p. 5, Jan. 2025, doi: 10.3390/iot6010005.
[20] A. M. Rostami, A. Karimi, and M. A. Akhaee, “Keyword spotting in continuous speech using convolutional neural network,” Speech Communication, vol. 142, pp. 15–21, Jul. 2022, doi: 10.1016/j.specom.2022.06.001.
[21] E. Van Der Westhuizen, H. Kamper, R. Menon, J. Quinn, and T. Niesler, “Feature learning for efficient ASR-free keyword spotting in low-resource languages,” Computer Speech & Language, vol. 71, p. 101275, Jan. 2022, doi: 10.1016/j.csl.2021.101275.
[22] G. P. Usha and J. S. R. Alex, “Advanced grad-CAM extensions for interpretable aphasia speech keyword classification: Bridging the gap in impaired speech with XAI,” Results in Engineering, vol. 24, p. 103414, Dec. 2024, doi: 10.1016/j.rineng.2024.103414.
[23] “Free Text-To-Speech for 28+ languages & MP3 Download | ttsMP3.com.” Accessed: Aug. 06, 2025. [Online]. Available: https://ttsmp3.com/
[24] Pixabay, "Sound Effects," *Pixabay*, [Online].
Available: https://pixabay.com/sound-effects.
[25] “speech_commands | TensorFlow Datasets,” TensorFlow. Accessed: Aug. 06, 2025. [Online]. Available: https://www.tensorflow.org/datasets/catalog/speech_commands