KEY WORD SPOTTING USING NEURAL NETWORKS

by

MARAM TEJASWANTH REDDY	19BEC1019
GUTURU SIVA NAGA NIHITH	19BEC1097
SHREYANSH KUMAR	19BEC1246
SRIVAIKUNTHAN N	19BEC1462

Project Report

submitted

to

Dr. KIRUTHIKA V

SCHOOL OF ELECTRONICS ENGINEERING

in partial fulfilment of the requirements for the course of

ECE3009 – Neural Networks and Fuzzy Control

in

B. Tech. ELECTRONICS AND COMMUNICATION ENGINEERING

Vandalur – Kelambakkam Road

Chennai – 600127

JULY 2022

BONAFIDE CERTIFICATE

Certified that this project report entitled "KEY WORD SPOTTING USING NEURAL NETWORKS" is a bonafide work of **SRIVAIKUNTHAN–19BEC1462,TEJASWANTH-19BEC1019, NIHITH-19BEC1097 and SHREYANSH-19BEC1246 who** carried out the Project work under my supervision and guidance for **ECE3009-Neural Networks and Fuzzy Control.**

Dr. KIRUTHIKA V

Assistant Professor Senior Grade,

School of Electronics Engineering (SENSE),

VIT University, Chennai

Chennai – 600 127.

ABSTRACT

Spoken Keyword Spotting is the task of identifying predefined words (called as keywords) from speech. Rapid developments and research in the areas of voice-based interaction with machines has tremendously influenced the heavy adaptation of these technologies into everyday life.

With the development of devices such as Google Home, Amazon Alexa and Smartphones, speech is increasingly becoming a more natural way to interact with devices.

However, always-on speech recognition is generally not preferred due to its energy inefficiency and network congestion that arises due to continuous audio stream from millions of devices to the cloud. Processing such a large amount of audio stream will require more time and adds to the latency and can have privacy issues.

Keyword Spotting (KWS) provides an efficient solution to all the above issues. Modern day voice-based devices first detect predefined keyword(s) — such as ”OK Google”, ”Alexa” — from the speech locally on the device. On successfully detecting such words, a full scale speech recognition is triggered on the cloud (or on the device).

Since the KWS system is always-on, it is highly preferred to have low memory footprint and computation complexity, but with high accuracy and low latency. We explore using a hybrid system consisting of a Convolutional Neural Network and a Support Vector Machine for KWS task.

ACKNOWLEDGEMENT

We wish to express our sincere thanks and deep sense of gratitude to our project guide, **Dr. Kiruthika V,** Assistant Professor Senior Grade, School of Electronics Engineering, for her consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We are extremely grateful to **Dr. Susan Elias,** Dean of School of Electronics Engineering, VIT Chennai, for extending the facilities of the School towards our project and for her unstinting support.

We express our thanks to our Head of the Department **Dr. Mohanaprasad** K for his support throughout the course of this project.

We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the course.

We thank our parents, family and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

SHREYANSH KUMAR

TABLE OF CONTENTS

1. INTRODUCTION

1.1 OBJECTIVES AND GOALS

To provide a suitable solution for the KWS setting, we look at a hybrid system — consisting of a Convolutional Neural Network (CNN) and a Support Vector Machine (SVM). We train the CNN model to be a feature extractor that embeds the input into a suitable representation that properly captures the relevant information. We consider the output of the 256 dimensional penultimate dense layer (marked with arrow on figure below) as an embedding of the input feature. We train the OCSVM with these embedding as input. The performance of OCSVM is highly dependent on its hyperparameters values. To obtain the best performing OCSVM, we tune the hyperparameters using scikit-optimize library.

1.2 APPLICATIONS

- Widely used in AI driven bots like Google Assistant and Alexa
- Used in IOT based automation systems
- Used in the field of Security and Voice recognition

1.3 FEATURES

Our model successfully detects the Keyword “Marvin”for which it is trained. The data-set contains speech recordings of over 30 words with different dialects and the model can be implemented for a real time use case as well.The model has achieved an accuracy of over 96% that successfully classifies the Keyword for which it is trained.

2.LITERATURE REVIEW

Guoguo Chen at el.(2019) in the paper titled a "**Small Footprint keyword spotting using Neural Network**" deep neural network is trained to directly predict the keywords or sub word units of the keywords followed by a posterior handling method producing a final confidence score. Keyword recognition results achieve 45% relative improvement with respect to a competitive hidden Markov Model-based system, while performance in the presence of babble noise shows 39% relative improvement.

Axel Bergl at el(2021) in the paper titled a "**Keyword Transformer: A Self-attention model for Keyword Spotting**" introduces us to a simple architecture which outperforms more complex models that mixes convolutional, recurrent and attentive layers. KWT can be used as a drop-in replacement for these models, setting two new benchmark records on the Google Speech Commands dataset with 98.6% and 97.7% accuracy on the 12 and 35-command tasks respectively.

Changhao Shan at el(2019) in the paper titled a "**Attention based End to End**",the authors proposed an attention-based end-to-end neural approach for small-footprint keyword spotting (KWS), which aims to simplify the pipelines of building a production-quality KWS system. The model consists of an encoder and an attention mechanism. The encoder transforms the input signal into a high level representation using RNNs. Then the attention mechanism weights the encoder features and generates a fixed-length vector. Finally, by linear transformation and soft max function ,the vector becomes a score used for keyword detection. This paper also evaluates the performance of different encoder architectures, including LSTM, GRU and CRNN.

Seungwoo Choi at el(2018) in the paper titled "**Temporal Convolution for Real-time Keyword Spotting on Mobile Devices**" the authors aimed to implement fast and accurate models for real-time KWS on mobile devices. They have measured inference speed on the mobile device, Google Pixel 1, and provided quantitative analysis of conventional convolution based KWS models and their models utilizing temporal convolutions. Their proposed model achieved 385x speedup while improving 0.3% accuracy compared to the state-of-the art model.Through ablation study, They have demonstrated that temporal convolution is indeed responsible for the dramatic speedup while improving the accuracy of the model.

Sercan at el(2017) in the paper titled "**Convolutional Recurrent Neural Networks for small footprint keyword spotting**" the paper combines the strengths of convolutional layers and recurrent layers to exploit local structure and long-range context. The paper analyzed the effect of architecture parameters, and propose training strategies to improve performance. With only ~230k parameters, our CRNN model yields acceptably low latency, and achieves 97.71% accuracy at 0.5 FA/hour for 5 dB signal-to-noise ratio.

Tara N at el(2020) in the paper titled "**Convolutional Neural Networks for Small print Keyword Spotting"** the authors explored CNNs for a KWS task.They compared CNNs to DNNs when they limit number of multiplies or parameters. When limiting multiplies, they found that shifting convolutional filters in frequency results in over a 27% relative improvement in performance over the DNN in both clean and noisy conditions. When limiting parameters, they found that pooling in time results in over a 41% relative improvement over a DNN in both clean and noisy conditions.

3. EXISTING SYSTEM

Traditional approaches for KWS are based on Hidden Markov Models with sequence search algorithms. With the advances in deep learning and increase in the amount of available data, state-of-the-art KWS has been replaced by deep learning based approaches due to their superior performance. Several attempts have been made using Deep Neural Networks (DNNs) and Convolutional Neural Networks (CNNs) to perform KWS task adhering to constraint budgets on memory and power. CNNs turn to be a befitting candidate for KWS task, because they capture translational invariance with far fewer parameters than DNNs by averaging the outputs of hidden units in different local time and frequency regions. CNNs are also capable of embedding speech by transforming them from input representations to meaningful representation.

4. PROPOSED SYSTEM

The approach we are taking to provide a suitable solution with this setting, we look at a hybrid system — consisting of a Convolutional Neural Network(CNN) and a Support Vector Machine (SVM). We train the CNN model to be a feature extractor that embeds the input into a suitable representation that properly captures the relevant information. We then train an One Class SVM (OCSVM), popularly used for outlier detection, with these embeddings as input. OCSVM is an unsupervised outlier detector particularly used in scenarios where there are huge imbalances in the dataset. We shall look at the implementation and training of KWS system. The KWS system is implemented in Python using Tensorflow framework and scikit-learn library. The input to the system is the log Mel Filterbank energies of the speech signal calculated with a window of length 25ms and stepsize 10ms. We use Tensoflow Dataset API for efficient handling of inputs

— generating input features on-the-fly since loading the entire dataset into memory would require high resources.

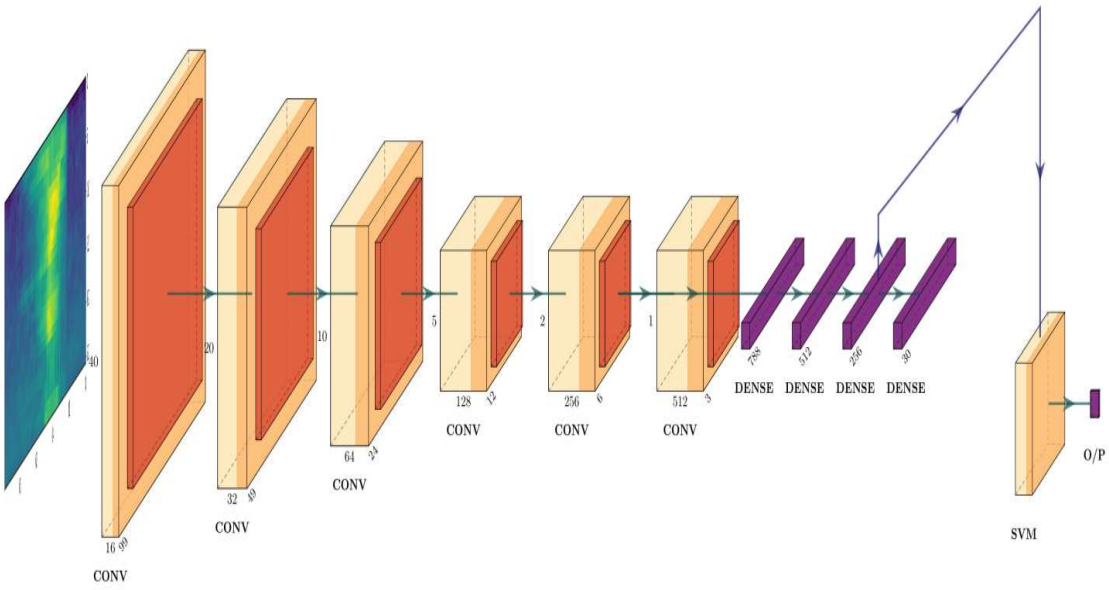


Figure 1-KWS Architecture

5. RESULTS AND DISCUSSION

First, we train a thirty-word classifier based on CNN architecture above using the training files in the dataset. The model achieved an training accuracy of 96.59% and validation accuracy of 95.56%. The deep network can be interpreted as a black box that transforms the input representations into meaningful representations, with which the final layer perform a classification. Since we have good performance in classification, we can safely assume that the model learns the proper representations for the input. We consider the output of the 256 dimensional penultimate dense layer as an embedding of the input feature.

We train the OCSVM with these embedding as input. We do not use the entire training set for this purpose. We use the validation set for this purpose. Since

OCSVM is an outlier detector, it needs only the positive samples to learn the maximum margin detector. The performance of OCSVM is highly dependent on its hyperparameters values. To obtain the best performing OCSVM, we tune the hyperparameters using scikit optimize library. We choose the best performing gp minimize() function for tuning. This tuning method approximates the cost function values (here, - F1Score) as a Gaussian process and minimizes it using Bayesian optimization. This method is far less expensive compared toother methods and performs empirically well.We train the OCSVM with positive samples from validation set with the hyper-parameters set with the optimal values. We then evaluate our model on the test set, whose results are summarized below in Fig4 . We can observe that we have a high value of detection and low value of false alarm which are typically desired of a KWS system.

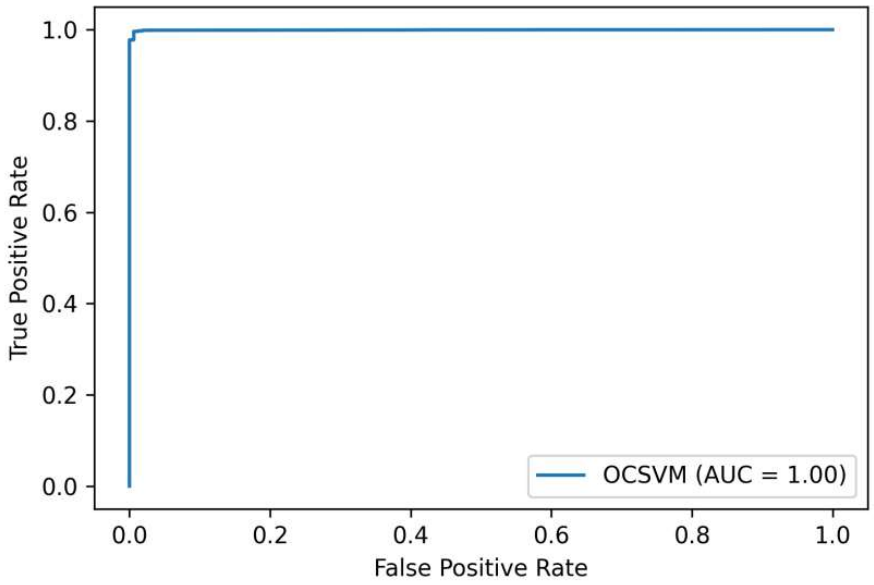


Figure 2-ROC Curve

Fig 2 shows the Receiver Operating Characteristics (ROC) and Fig 3 below shows the Precision Recall graph of the classifier. We can observe that the area under the curve for both curves are 1, which is desirable, but the use of these

graphs for a imbalanced set is debatable.

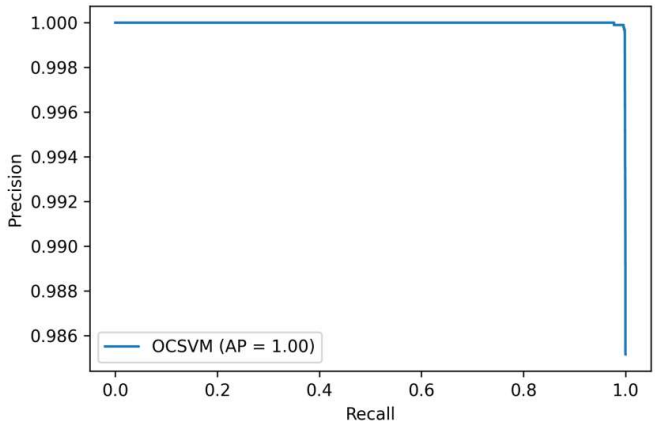


Figure 3-Precision Recall

We train the OCSVM with positive samples from validation set with the hyper-parameters set with the optimal values. We then evaluate our model on the test set, whose results are summarized below in Fig4 . We can observe that we have a high value of detection and low value of false alarm which are typically desired of a KWS system.

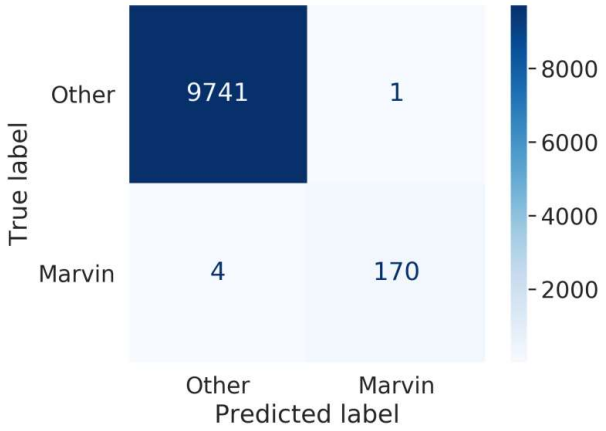


Figure 4-Test set: Confusion Matrix

Performance Metrics of KWS system:

Model size	11.4MB
Model size (Quantized)	978KB
Real Time Factor (RTF)	1.7ms
Accuracy	0.9995
Precision	0.9942
Recall (True Detection Rate)	0.9770
F1 Score	0.9855
Matthews Correlation Coefficient	0.9853
False Alarm Rate (FAR)	0.0001
False Alarm per Hour (FA/Hr)	0.0003
True Rejection Rates (TRR)	0.9998
False Rejection Rates (FRR)	0.0229
True Rejection Rates (TRR)	0.9998

6. CONCLUSION AND FUTURE WORK

A small footprint reconfigurable CNN-OCSVM based KWS system is proposed in this work. The raw performance numbers shows that this model outperforms many of the models in the literature. However, a direct comparison is not meaningful be cause of the differences in the datasets and the actual keywords. To prove this claim a lot of further work is necessary, such as performance analysis under noise and far-field conditions.The model can be experimented for multiple sets of Keywords to test the accuracy and robustness of the model

REFERENCES

1. Guoguo Chen, Carolina Parada, Georg Heigold 2019. To develop a keyword spotting system with a small memory footprint, low computational cost, and high precision using deep neural networks.

2. Axel Berg1, Mark O'Connor, Miguel Tairum Cruz 2021.Keyword TRANSFORMER: A Self-Attention Model FOR Keyword Spotting

3. Changhao Shan, Junbo Zhang, Yujun Wang, Lei Xie 2019.Attention-based End-to-End Models for Small-Footprint Keyword Spotting

4. Seungwoo Choi , Seokjun Seo , Beomjun Shin , Hyeongmin Byun, Martin Kersner, Beomsu Kim, Dongyoung Kim , Sungjoo Ha 2018.Temporal Convolution for Real-time Keyword Spotting on Mobile Devices

5. Sercan Ö. Arık, Markus Kliegl, Rewon Child, Joel Hestness, Andrew Gibiansky , Chris Fougner, Ryan Prenger, Adam Coates 2017.Convolutional RNN for small footprint Keyword Spotting

6. Tara N. Sainath, Carolina Parada 2020.Convolutional Neural Networks for Small-footprint Keyword Spotting

APPENDIX 1

BIODATA

Name :Srivaikunthan N

Mobile Number :7397231872

E-mail :srivaikunthan.n2019@vitstudent.ac.in

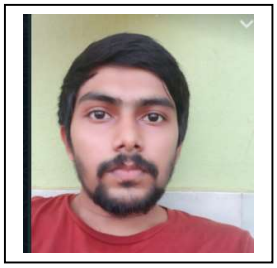
Permanent Address:Chennai Tamil Nadu

Name :Maram Tejaswanth Reddy

Mobile Number :9347495679

E-mail :maram.tejaswanthreddy2019@vitstudent.ac.in

Permanent Address:Addanki,Andhra Pradesh

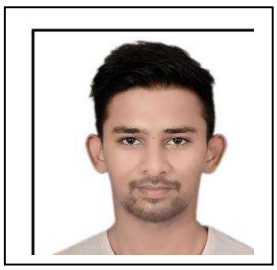


Name :Guturu Siva Naga Nihith

Mobile Number :9440009679

E-mail :guturusiva.naganihith2019@vitstudent.ac.in

Permanent Address:Nellore,Andhra Pradesh



Name :Shreyansh Kumar

Mobile Number :7004251110

E-Mail :shreyansh.kumar2019@vitstudent.ac.in

Permanent Address :Patna,Bihar