IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Implementation And Analysis Of A Hidden Markov Model Based Speech Recognition System

[1]S. Niveditha, [2]Dr. Pradip K. Das, [3]Junaid Shafee
[1]Assistant Professor, [2]Professor, [3]Student
[1]Dept. of CSE, SRM Institute of Science and Technology, Vadapalani, Chennai, India
[2]Dept. Of CSE, Indian Institute of Technology, Guwahati, India
[3]Dept. of CSE, SRM Institute of Science and Technology, Vadapalani, Chennai, India

Abstract: This paper demonstrates a speech recognition system for isolated spoken digits (0–9) based on Hidden Markov Models (HMMs), demonstrating an end-to-end pipeline covering feature extraction, training, and testing. Raw speech signals are analyzed to obtain cepstral coefficients, which are subsequently vector quantized to generate observation sequences. Each digit is represented by an individual HMM trained with the Forward-Backward algorithm, Viterbi decoding, and iterative Baum-Welch re-estimation. In testing, the system predicts unseen speech samples by calculating the likelihood of observation sequences with respect to trained models and predicting the digit of highest likelihood. With high accuracy, the system shows the efficiency of statistical modeling in speech recognition and its promise for real-world use.

Index Terms: Speech Recognition, Hidden Markov Models, Machine Learning, Isolated Word Recognition, Feature Extraction, Linear Predictive Coding, Vector Quantization, Viterbi Algorithm, Baum-Welch Algorithm

I. INTRODUCTION

The introduction sets the establishment of speech recognition within the context of a historical area of research in artificial intelligence, with discussion on the way statistical modeling techniques revolutionized the study. It presents the difficulties inherent in speech recognition and the system presented in this project—an isolated word recognition system for English digits based on Hidden Markov Models (HMMs).

1.1 Background

Speech is the most natural and effective form of human communication. As the focus on human-computer interaction has increased, speech recognition technology has become a significant area of research in artificial intelligence. It allows machines to recognize and respond to spoken words, making interfaces more user-friendly.

Out of numerous applications, isolated word recognition — in which a word is uttered individually — is a building block and a precursor to more sophisticated speech processing systems. Identification of English digits (0–9) spoken correctly is a significant benchmark, particularly while testing early speech recognition systems.

1.2 The Evolution of Speech Recognition

Speech recognition has evolved over a several decades, starting with early speech recognition systems during the mid-20th century that could recognize just a few words of spoken language. These early systems made extensive use of basic template matching techniques, which were too inflexible and unstable. The major breakthrough came in the 1970s and 1980s when probabilistic models and statistical methods were introduced to approach speech as a time-sequential process.

The introduction of statistical techniques was a turning point in the development of the discipline. Linear predictive coding (LPC), and hidden Markov models (HMMs) helped model speech more precisely by recognizing its time variability and addressing pronunciation, duration, and speaker variation uncertainties. The early years set the stage for much of the methods used today, and provide the foundation for the system in this project.

1.3 Challenges in Speech Recognition

Speech recognition systems have to overcome many intricate challenges:

- **Speaker Variability:** Accent, pitch, and speech rate differences cause inconsistencies.
- **Time-Varying Nature of Speech:** The duration and rhythm of words may vary when spoken.
- **Background Noise and Artifacts:** Recordings from the real world contain noise that degrades the signal.
- **Subtle Phonetic Overlaps:** Adjacent sounds can affect one another, complicating segmentation and identification.

To meet these, effective models should be able to recognize sequential patterns, handle noise, and be adaptable to variations—chief advantages provided by the HMM-based method.

1.4 Hidden Markov Models for Speech

Hidden Markov Models (HMMs) are a probabilistic model that are especially suitable for speech modeling. They model speech as a sequence of hidden states that map to linguistic units (e.g., words or phonemes), and the observed outputs are feature vectors from the speech signal.

HMMs enable:

- Capturing the temporal structure of speech
- Handling variable-length inputs

- Efficient training, decoding, and evaluation algorithms

This project uses HMMs to model every English digit (0–9) with discrete observation sequences based on vector-quantized cepstral coefficients.

1.5 Objectives and Contributions

The goal of this project is to design an isolated word speech recognition system for English digits based on HMMs with the following specific contributions:

- Design of a full speech processing pipeline consisting of feature extraction, vector quantization, and HMM modeling.
- Training of separate HMMs for each digit using Forward-Backward and Baum-Welch algorithms.
- System performance evaluation on unseen speech samples with multiple speakers to verify speaker-independence.
- Application of the Viterbi algorithm for decoding and optimization of state sequence.
- Comparison of recognition rates between digits and examination of errors.

The system points to the significance and efficacy of traditional HMM-based methods for simple speech recognition tasks even in the age of deep learning solutions.

II. LITERATURE REVIEW

In the last few decades, speech recognition has undergone a transition from deterministic template-based to strong statistical models. HMM development and usage have been a turning point, providing a principled means of modeling speech's probabilistic and sequential structure. This part summarizes key papers that have impacted isolated and multi-digit recognition by HMMs.

Early basic work on HMMs in speech recognition is presented in Rabiner's classic tutorial [1]. The article presents the mathematical derivation of HMMs and outlines key algorithms like the Forward-Backward algorithm, Viterbi decoding, and Baum-Welch re-estimation. These algorithms facilitate accurate modeling of speech temporal variability, rendering HMMs over static pattern-matching methods of prior art.

Extending the theoretical model into application form, Rabiner et al. [2] proposed a model-based connected-digit recognition system that could utilize either HMMs or template-based references. Their system shows that sequences of digits, when trained on whole-word models based on continuous speech as opposed to isolated speech, yield dramatically better accuracy. Using cepstral analysis, segmental k-means clustering, and level-building algorithms, the system is able to obtain more than 98% accuracy in speaker-dependent environments and up to 96.6% in speaker-independent situations.

Wilpon et al. [3] investigated the application of HMM-based recognizers in actual telephony systems. Their paper describes the combination of HMMs with digital signal processing (DSP) hardware for use in voice-based credit card verification and other applications. The application of cepstral and delta-cepstral features, along with dynamic programming for sequence alignment, was very effective in speaker and channel variability conditions. This work is a key bridge between laboratory-grade models and commercially feasible systems.

For keyword spotting applications to spontaneous speech, yet another study by Wilpon et al. [4] offered an HMM-based keyword spotting system. With the use of distinct HMMs for modeling both target keywords and non-keyword (garbage) speech, their system proved to be highly accurate at isolated word recognition within unconstrained spoken input. Such ability is vital in real-time applications where isolated digit recognition might be inserted into more complex command structures.

In a broader and more theoretical context, Bahl et al. [5] cast speech recognition as a maximum likelihood decoding problem, laying the groundwork for statistical language modeling and decoding in a noisy communication channel. Their work introduces Markov sources for acoustic and linguistic modeling and suggests dynamic programming-based search strategies for optimal decoding in constrained or natural language tasks. While centered on continuous speech recognition, this effort makes important contributions to probabilistic modeling techniques employed in all HMM-based systems.

These efforts together provided the foundations for durable, small-vocabulary speech recognition systems, especially for digit recognition. The use of domain-specific feature extraction (e.g., cepstral coefficients), sophisticated model training algorithms, and statistical decoding has been successful for both isolated and connected speech tasks, even in adverse real-world environment.

The tutorial by Rabiner, dated 1989, is one of the most influential and most-cited papers in speech recognition literature [1]. The paper presents Hidden Markov Models (HMMs) as a versatile statistical tool for modeling time-sequential data, such as speech signals. It presents a clear and simple description of the fundamental HMM elements, namely states, observation symbols, transition probabilities, and emission probabilities.

Rabiner's treatment rigorously illustrates the three basic issues related to HMMs:

1. **Evaluation** – calculating the probability of an observation sequence for a given model, solved through the Forward algorithm.
2. **Decoding** – finding the most likely state sequence that produced a specific observation sequence, solved by the Viterbi algorithm.
3. **Learning** – setting the model parameters to optimize the probability of an observation set, done through the Baum-Welch re-estimation algorithm (a type of Expectation-Maximization).

The tutorial not only discusses the mathematical underpinnings of the algorithms but also offers practical advice on their implementation, complexity, and limitations. It highlights the strength of HMMs in modeling time-varying speech and variable-length input, which makes them very effective for tasks such as isolated digit recognition.

One of the strengths of the tutorial is its application-oriented focus. Rabiner shows how HMMs are used in practical speech recognition applications, such as isolated word and connected word recognition systems, speaker-independent modeling, and digit string recognition. He also touches upon problems such as model topology, training data needs, and feature vector quantization—central issues for any HMM-based system.

Baum et al. [2] talk about one of the first and most significant theoretical advances in the statistical modeling of sequential data [2]. Released in 1970, the work introduced what is now popularly referred to as the Baum-Welch algorithm—an Expectation-Maximization (EM)-based algorithm for estimating Hidden Markov Models' (HMMs) parameters.

The authors formalize re-estimation of model parameters (transition and emission probabilities) such that it ensures non-decreasing likelihood of observed data under the model. This iterative procedure converges to a local maximum, and thus it is applicable for unsupervised learning of HMMs from unlabeled observation sequences. The algorithm works by calculating forward and backward probabilities and employing them to update the model in probabilistically consistent fashion.

The Baum-Welch algorithm solves a central problem in speech recognition: how to train models when the hidden state sequence underlying them is unknown. In contrast to deterministic or template-matching methods employed by early systems, Baum's method enables systems to learn adaptively temporal and acoustic variations from observation. It finds particular application in speech tasks in which segmentation and labeling of states are impossible.

In addition, this paper is seminal in that it provides the learning foundation of the whole HMM infrastructure. Although Rabiner's tutorial gives us the implementation guide, Baum et al.'s paper gives us the theoretical guarantee that model parameters can be successfully learned with no direct supervision—a critical component in constructing scalable speech recognition systems.

The 1967 paper by Andrew J. Viterbi [3] introduced what is now known as the Viterbi algorithm, a dynamic programming method for finding the most likely sequence of hidden states in a Markov process. Although initially designed for the decoding of convolutional error-correcting codes used in communication systems, the algorithm's recursiveness and optimality with respect to maximum likelihood criteria positioned it well for application in a large number of domains, among them speech recognition.

For Hidden Markov Models (HMMs), this gives a solution to the decoding problem—finding the most likely state path that could have generated a particular sequence of observations. This is especially important in speech recognition applications where we want not only to identify what was uttered, but also the timing alignment of phonetic units or words in the speech signal.

It uses the recursive approach of calculating each state's maximum probability path in each time step and maintaining back-pointers such that it is able to construct the optimal sequence of states effectively. It keeps it computationally efficient to compute the best match over large sequences of observations in polynomial time versus exponential time needed to explore every possible path. The use of the Viterbi algorithm integrates so that the recognition system is accurate and computationally efficient, particularly for real-time or embedded environments.

In 1986, Rabiner et al. show one of the first large-scale uses of Hidden Markov Models (HMMs) for real-world speech recognition applications [4]. The paper compares and analyzes HMM-based and template-based methods for connected-digit recognition and provides useful insights into the advantages and disadvantages of both.

Unlike isolated digit recognition, connected-digit tasks involve modeling coarticulation—where surround digits have a mutual acoustical influence on each other. To deal with this, the authors apply whole-word HMMs directly trained on connected-digit utterances. By so doing, segmentation of ongoing speech into isolated digits is avoided, allowing for more smooth and accurate recognition.

The system utilizes discrete vector quantization of cepstral features and a level-building algorithm in decoding sequences of digits. Training is conducted using segmental k-means and re-estimation methods, while time alignment scoring and Viterbi algorithm are used in evaluation.

Major experimental findings indicated that HMM-based systems were superior to template-based approaches, especially in speaker-independent and multi-speaker scenarios. The system was able to achieve up to 98.4% digit string accuracy with speaker-trained conditions and performed well with unknown speakers. The paper also indicates that utilizing connected-digit training data instead of isolated digit samples greatly enhances recognition rates—a principle that can be applied directly to larger vocabulary tasks as well.

This work is particularly germane to this project because it sets up the dominance of statistical HMM models in digit recognition and highlights the importance of data-driven training. The methods outlined here - e.g., cepstral feature extraction, model re-estimation, and maximum-likelihood decoding—are the technical heart of the HMM pipeline.

Wilpon, Mikkilineni, and Gokcen recount the operational problems and breakthroughs in applying HMM-based speech recognition systems in business and working environments in their 1990 paper "Speech Recognition: From the Laboratory to the Real World" [5]. Previous studies on HMMs concentrated more on experimental laboratory settings, but this paper talks about the bridge between research-quality models and deployable field products—a critical benchmark in speech recognition history.

The authors outline the application of stand-alone and connected-digit recognition systems in AT&T's telephony applications, such as automated credit card validation and directory services. These systems employ front-end signal processing methods—such as DC shift removal, pre-emphasis, and cepstral feature extraction—subsequent to discrete vector quantization and HMM-based pattern recognition. Computational efficiency is emphasized through the application of Digital Signal Processors (DSPs) and efficient algorithms such as Viterbi decoding.

Another significant contribution of this work is its emphasis on speaker-independent recognition. The system is trained on the data from many speakers and tested in actual use cases where users might voice under different acoustic conditions (e.g., presence of background noise, distortion of telephone lines). For preserving accuracy under these limitations, the authors use noise-robust modeling techniques, adaptive codebooks, and improved token-passing decoders.

Experimental evidence reported in the paper confirms that real-time recognition with greater than 95% accuracy can be attained even under mismatched, noisy speaker conditions. This amounts to empirical evidence that HMMs—when combined with effective implementation techniques—can scale from prototype implementations to deployed systems.

The 1990 paper by Wilpon et al. discusses the use of HMMs for solving keyword spotting—a harder variant of isolated word recognition where desired words have to be identified in continuous, unconstrained speech flows [6].

In contrast to traditional speech recognition systems, which operate on the premise of well-defined word boundaries and clean utterances, keyword spotting systems have to deal with spontaneous speech, varying contexts, and strong background speech. In order to solve these problems, the authors introduce an architecture involving two sets of HMMs: one for target keywords and another for filler (non-keyword) speech. The filler models are "garbage" detectors that pick up the variability of non-target utterances and minimize false positives.

Major technical contributions are:

- Employment of multiple observation sequences (e.g., static and delta cepstral features),
- Employment of forced alignment and confidence scoring,

- Employment of Viterbi-based alignment and scoring to separate target and filler regions in the audio stream.

Experiments on telephone-quality speech reveal that the system was 99.3% accurate for isolated keyword inputs and 95.1% accurate in unconstrained environments, indicating the robustness of HMMs even without explicit segmentation. The capability to process overlapping and unstructured speech input indicates that HMMs can be applied beyond isolated tasks to more fluid and naturalistic situations.

The 1986 tutorial paper "An Introduction to Hidden Markov Models" [7] by Rabiner et al. provides a transparent and understandable base for comprehending the underlying fundamentals of Hidden Markov Models (HMMs) and how they are being used in speech recognition. Although Rabiner's 1989 tutorial is the most referenced tutorial due to mathematical completeness, the previous work assumes a critical importance in bringing HMMs into the public knowledge of practitioners and engineers joining the profession.

The authors first situate the requirement of HMMs for speech processing. Classic template-based techniques did not have enough capacity to deal with the time-varying and random aspects of speech, especially where there was speaker variation and background noise. Rabiner and Juang provide support for HMMs as means to characterize speech signals in terms of strings of probabilistic states that produce observable outputs. This "doubly stochastic process" view encapsulates the state dynamics that are hidden and the acoustic observation, creating a coherent framework for recognition.

For our stand-alone digit recognition application, this tutorial is an essential stepping stone. It connects general probabilistic notions with practical signal modeling and introduces a layered organization for recognition pipelines that closely fits your system design: cepstral feature extraction, quantization, model training, and Viterbi-based prediction. The focus on statistical sequence modeling and temporal segmentation is a direct support of our implementation of speaker-independent, digit-level HMMs.

III. METHODOLOGY

This chapter describes the theoretical framework and implementation strategy for our isolated digit recognition system based on Hidden Markov Models (HMMs). The method is an extension of the standard HMM pipeline, with enhancements in preprocessing, feature extraction, and reliable statistical modeling. Mathematical underpinnings of every algorithm used are also introduced.

3.1 Historical Methodology

Historically, conventional isolated digit speech recognition systems have made use of template matching or dynamic time warping (DTW) techniques. These systems used prerecorded reference patterns of spoken digits and compared them with test inputs on the basis of similarity measures. Though useful in restricted environments, such systems were highly susceptible to speaker variation, channel noise, and rate of speaking.

With the arrival of Hidden Markov Models (HMMs), a probabilistic approach was developed that significantly improved robustness and scalability. In the conventional HMM-based pipeline, isolated word recognition is performed as follows:

- **Preprocessing:** The audio is preprocessed by cleaning through DC shift removal and amplitude normalization.
- **Feature Extraction:** Speech frames are subjected to a Hamming windowing and computation of cepstral coefficients (classically on LPC-derived features).
- **Vector Quantization (VQ):** The cepstral features are quantized as discrete symbols from a pre-defined codebook.
- **Model Training:** Each digit has an independent HMM. Training is generally obtained with the Baum-Welch algorithm for parameter estimation.
- **Recognition:** Viterbi is utilized to calculate the most probable state sequence of probabilities and project the test sample into the maximum likelihood digit model.

3.2 Proposed Methodology

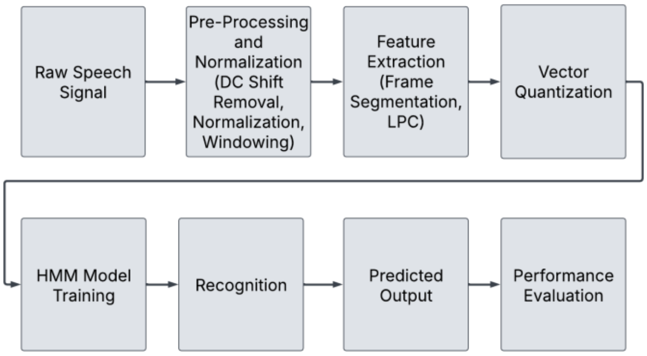

Raw Speech Signal
Pre-Processing and Normalization (DC Shift Removal, Normalization, Windowing)
Feature Extraction (Frame Segmentation, LPC)
Vector Quantization
HMM Model Training
Recognition
Predicted Output
Performance Evaluation

Fig.: 3.2.1 Pipeline for the system

Our approach extends the traditional HMM methodology with judicious design decisions in preprocessing, feature extraction, and model setup to improve accuracy and speaker-independence. The system targets isolated English digits (0–9) and is comprised of the following pipeline in detail:

3.2.1 Data Collection

The dataset used for this project is the Free Spoken Digit Dataset (FSDD) from Kaggle which is an open-source dataset. It consists of 6 speakers, with each speaker having 50 utterances per digit (3000 total samples).

3.2.2 Preprocessing

The audio recording goes through a series of normalization processes:

- **DC Shift Removal:** Computation of the average amplitude and its subtraction for correcting DC bias. [9]

$$x_{\text{corrected}}(n) = x(n) - \mu \text{, where } \mu = \frac{1}{N}\sum_{n=1}^{N} x(n)$$

- **Amplitude Normalization:** Adjusts all samples to within a constant amplitude range to provide speaker and recording uniformity. [9]

$$x_{\text{norm}}(n) = \frac{x(n)}{max(|x(n)|)}$$

- **Framing:** There is 320 sample (20 ms) framing of each recording with 60% overlap. [8]

- **Windowing:** Signal discontinuities at frame boundaries are minimized by applying a Hamming Window to each frame [8]

3.2.3 Feature Extraction

We apply LPC-based cepstral analysis to obtain significant features from each frame:

Autocorrelation is calculated for every frame.

- LPC coefficients are obtained via the Levinson-Durbin algorithm.
- Cepstral coefficients are calculated from LPCs and liftered with a raised sine window to enhance discrimination.
- Energy coefficients are optionally added for robustness.

Each frame yields a 12-dimensional feature vector describing the phonetic nature of the speech signal.

3.2.4 Vector Quantization

To discretize the continuous feature vectors:

- A 32-sized codebook is created by applying the LBG algorithm [11] to training data.
- A feature vector is mapped to the closest codebook entry by using the Tokhura distance [10], which perceptually weights cepstral dimensions.

$$D(\mathbf{c}, \mu_k) = \sum_{i=1}^{12} w_i\left(c_i - \mu_{k,i}\right)^2$$

where w_i are predefined Tokhura weights and μ_k is the k^{th} codeword in the codebook.

- The process converts the feature sequence into a discrete observation sequence appropriate for discrete HMMs.

3.2.5 Hidden Markov Model Definitions

Every digit (0–9) is represented by a 5-state left-to-right HMM with the following configuration:

- **Initial state distribution (π):** Uniform probability over states.

$$\pi_i = \text{ initial state probability}$$

- **Transition matrix (A):** Left-to-right transitions only (no backward jumps).

$$A = \{a_{ij}\}$$

- **Observation matrix (B):** 32-symbol discrete probability distributions.

$$B = \{b_j(k)\}$$

3.2.6 HMM Algorithms

Successful training of Hidden Markov Models involves estimation of the parameters of the model—transition probabilities, observation probabilities, and initial state distributions—such that the model has the best explanation for a given set of observed sequences. Three fundamental algorithms for HMM training and testing are applied in this project: Forward algorithm, Viterbi algorithm, and Baum-Welch algorithm.

1. Forward Algorithm

The Forward algorithm [1] is used to compute the probability of an observation sequence from an HMM. It computes the overall probability that a sequence of observations came from a model by adding over all possible paths of hidden states. This algorithm is critical for making test samples comparable with trained digit models, and for calculating the likelihood required in other algorithms such as Baum-Welch.

2. Viterbi Algorithm

The Viterbi algorithm [3] is a decoding algorithm that finds the most probable sequence of hidden states (also referred to as the state path) which may have generated a particular sequence of observations. As opposed to the Forward algorithm, which determines total probability on all paths, Viterbi determines the best single path. It is particularly suited to aligning observed sequences with model states and is used in a variation of HMM training known as Viterbi re-estimation.

3. Baum-Welch Algorithm

Baum-Welch [2] is an EM-based unsupervised learning algorithm. The Baum-Welch algorithm learns HMM parameters by maximizing the observation data likelihood. With a given initial model parameter estimation, the algorithm alternates between refining these estimates by determining how likely each segment in the sequence was formed by each state and updating the parameters based on this. It is particularly valuable when the state sequence for the hidden state is unknown, as it is often the case in speech recognition systems.

3.2.7 Recognition and Decision

After the Hidden Markov Models (HMMs) for the digits (0–9) are trained, recognition is the process of analyzing an unknown speech sample and assigning it one of the ten digits. The recognition process is most important to the performance of the system and includes three primary stages: observation sequence generation, model scoring, and digit selection.

1. Observation Sequence Generation

The input speech signal goes through the same pre-processing and feature extraction processes as in training:

- The audio is windowed, framed, and normalized.
- Cepstral coefficients are extracted for each frame.
- Each frame's feature vector is quantized to the nearest codeword from the pre-generated codebook through vector quantization. This creates a sequence of discrete observation symbols that represent the speech signal in a form consistent with the HMMs trained.

2. Likelihood Computation

For each of the ten trained HMMs (one per digit), the system computes the probability that the given observation sequence was generated by the model. This is done using the Forward algorithm, which computes probabilities over all feasible state paths, leaving the total likelihood score per digit model.

Alternatively, in order to analyze or debug, the Viterbi algorithm may be used to find the most likely state path and the associated probability. However, for recognition purposes, the Forward algorithm is preferable since it can compare all possible sequences of states, rather than just the most likely one.

3. Decision Making

After computing probability scores for all digit models, maximum likelihood classification is performed by the system. The digit with the highest probability matched model is the output predicted. The method assumes equally probable a priori models (uniform class priors), and that is fine with well-balanced data sets such as isolated digit recognition.

4. Output Interpretation

The predicted digit is then returned to the user and the accuracy of the system is measured along with other metrics like precision, recall and F1-score (macro).

IV. RESULTS

5.1 Dataset Overview

The dataset used in this project is the Free Spoken Digit Dataset (FSDD) from Kaggle, which is an open-source dataset. It contains spoken digit utterances recorded from six distinct speakers: George, Jackson, Lucas, Nicolas, Theo, and Yweweler. Each of these speakers captured 50 utterances for each digit (0–9), which amounts to 500 utterances from each speaker and 3,000 utterances in total.

Training and Testing Split

For each speaker, the dataset was split into 35 samples per digit i.e., 350 samples for training set and 15 samples per digit i.e., 150 samples for testing set.

In this speaker-dependent setup, separate models were trained for each speaker. Testing was also done per speaker on their own unseen test samples. Testing was also performed per speaker based on their own unseen test samples.

Evaluation Metrics

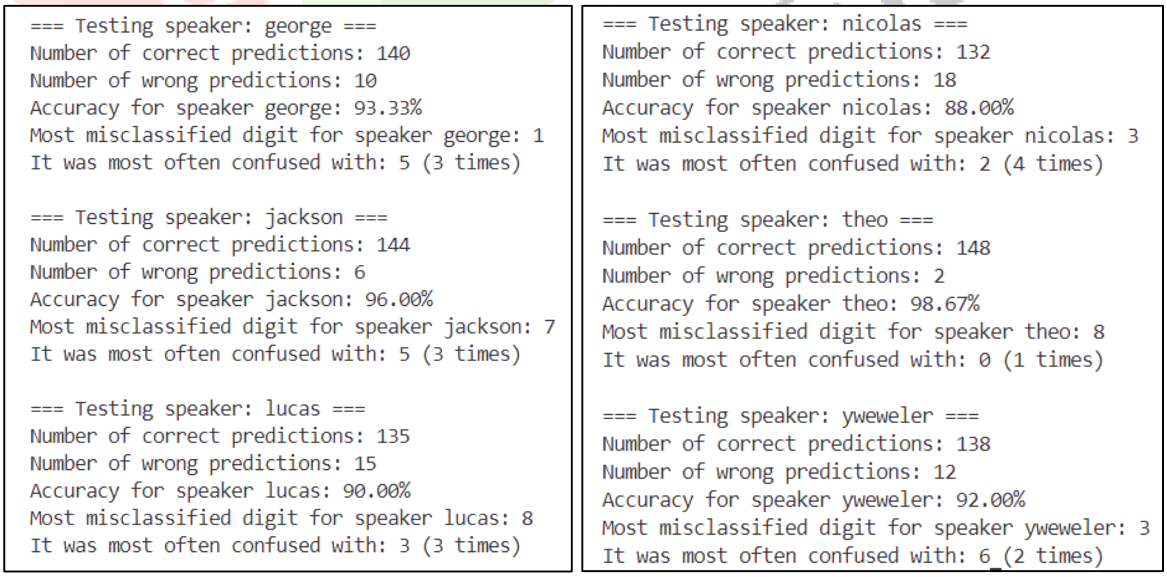
Each speaker-specific model's performance was measured using:

- **Overall Accuracy:** The ratio of correctly predicted digits to the total number of test samples.
- **Confusion Matrix:** For visualizing per-digit classification accuracy and finding most common misclassifications.

All the evaluation metrics were computed on a per-speaker basis to investigate inter-speaker variability and digit-specific accuracy.

5.2 Accuracy results

Recognition accuracy of every speaker model is presented in Table X. The models were as accurate as 88.00% to 98.67% for all the speakers, reflecting good digit classification with speaker-dependent training. Theo was the most accurate at 98.67%, and Nicolas had the lowest accuracy at 88.00%. Most of the models had over 90% accuracy, indicating good performance in general.



```
=== Testing speaker: george ===
Number of correct predictions: 140
Number of wrong predictions: 10
Accuracy for speaker george: 93.33%
Most misclassified digit for speaker george: 1
It was most often confused with: 5 (3 times)

=== Testing speaker: jackson ===
Number of correct predictions: 144
Number of wrong predictions: 6
Accuracy for speaker jackson: 96.00%
Most misclassified digit for speaker jackson: 7
It was most often confused with: 5 (3 times)

=== Testing speaker: lucas ===
Number of correct predictions: 135
Number of wrong predictions: 15
Accuracy for speaker lucas: 90.00%
Most misclassified digit for speaker lucas: 8
It was most often confused with: 3 (3 times)
```

```
=== Testing speaker: nicolas ===
Number of correct predictions: 132
Number of wrong predictions: 18
Accuracy for speaker nicolas: 88.00%
Most misclassified digit for speaker nicolas: 3
It was most often confused with: 2 (4 times)

=== Testing speaker: theo ===
Number of correct predictions: 148
Number of wrong predictions: 2
Accuracy for speaker theo: 98.67%
Most misclassified digit for speaker theo: 8
It was most often confused with: 0 (1 times)

=== Testing speaker: yweweler ===
Number of correct predictions: 138
Number of wrong predictions: 12
Accuracy for speaker yweweler: 92.00%
Most misclassified digit for speaker yweweler: 3
It was most often confused with: 6 (2 times)
```

Fig. 5.2.1: Testing Output

Table 5.2.1: Accuracies for each of the speaker-dependent models

Speaker Model	Accuracy (%)
George	93.33
Jackson	96
Lucas	90
Nicolas	88
Theo	98.67

5.3 Confusion Matrices

To get a better idea of how well models are performing digit by digit, confusion matrices were generated for each speaker (see Figures 5.3.1 - 5.3.3). These figures tell us how often each digit was correctly predicted and where the misclassifications occurred. These matrices also show any speaker patterns or weaknesses.

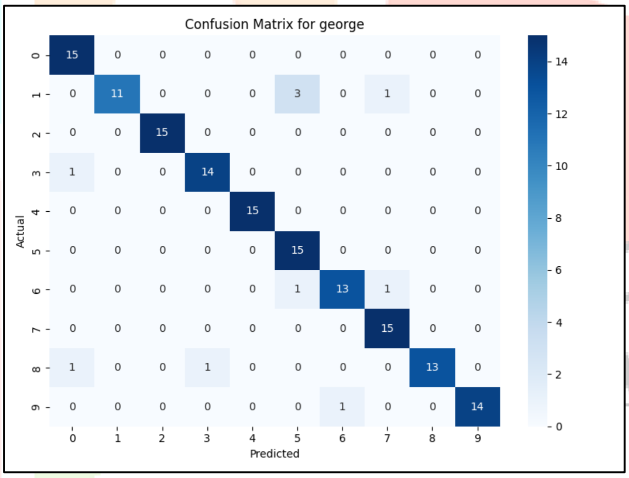


Fig. 5.3.1: Confusion matrix for speaker George

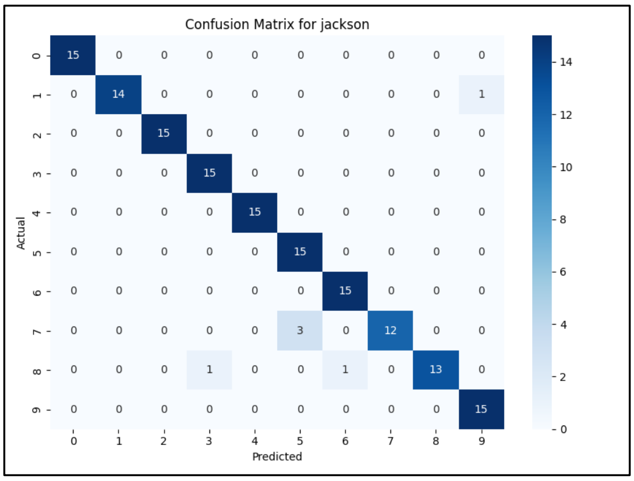


Fig. 5.3.2: Confusion matrix for speaker Jackson

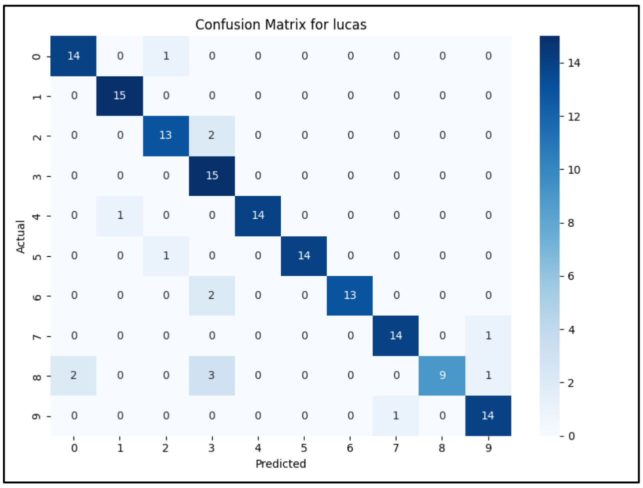


Fig. 5.3.3: Confusion matrix for speaker Lucas

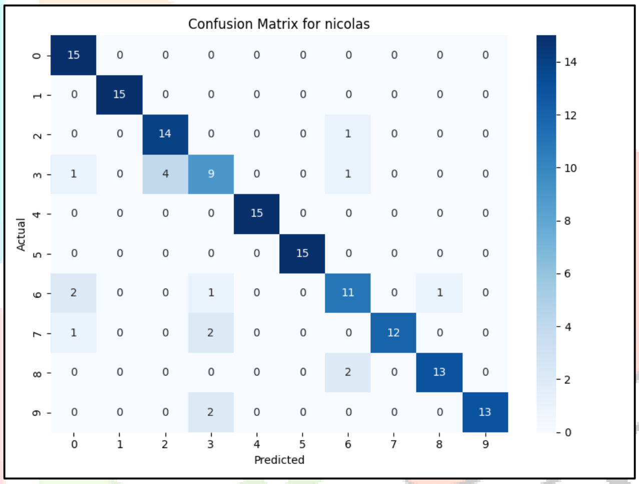


Fig. 5.3.4: Confusion matrix for speaker Nicolas

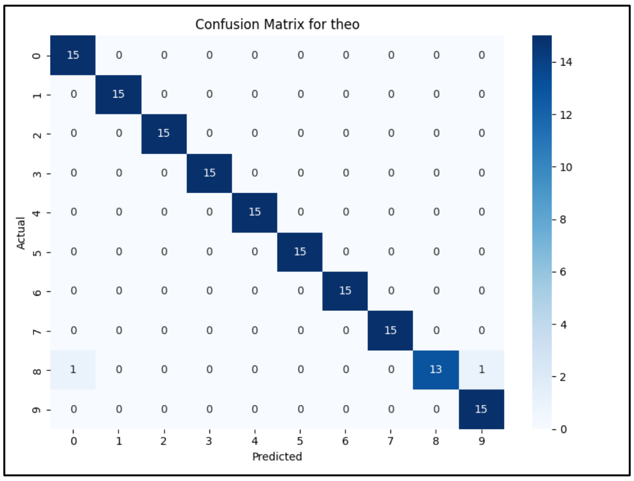


Fig. 5.3.5: Confusion matrix for speaker Theo

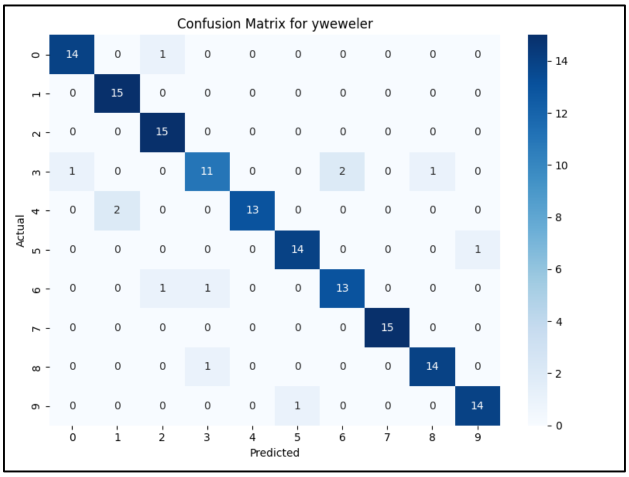


Fig. 5.3.6: Confusion matrix for speaker Yweweler

5.4 Additional Evaluation Metrics

Apart from accuracy and confusion matrices, we checked each model with a speaker using precision, recall, and F1-score. These measures give a finer idea of the performance of the model, especially with regard to how well it can recognize each digit without generating false positives or negatives.

Table. 5.4.1: Precision, Recall and F1-Score per speaker (Macro Averaged)

Speaker	Precision (%)	Recall (%)	F1-Score (%)
George	94.16	93.33	93.27
Jackson	96.46	96	95.95
Lucas	91.69	90	89.95
Nicolas	88.72	88	87.99
Theo	98.75	98.67	98.64
Yweweler	92.11	92	91.89

These findings are consistent with overall accuracy findings, with all the speakers showing uniformly high performance, highest F1-score for Theo (98.64%) and lowest for Nicolas (87.99%). The close similarity between precision and recall values suggests that the models are performing well in balancing false positives against false negatives.

V. CONCLUSION & FUTURE WORK

The aim of this project was to develop a spoken digit recognition system with the Free Spoken Digit Dataset (FSDD) and Hidden Markov Models (HMMs). There were six individual models trained, one for each speaker, on 35 training utterances per digit and 15 for testing.

The tests showed high accuracy among all speakers ranging from 88% to 98.67%. Analysis using confusion matrices and other evaluation metrics showed that the models were efficient in recognizing the isolated digits.

The results verify that HMMs work well for speech recognition especially when sufficient training data is available for each speaker.

Hidden Markov Models were very useful for this task because of a number of reasons:

- **Sequential Modeling:** HMMs naturally model time-series data and are therefore good for speech signals changing over time.
- **Efficiency:** HMMs are much less computationally demanding than contemporary deep learning models thus making them suitable for small scale datasets or real-time applications.

To extend the system further, the following extensions can be pursued:

- Replace discrete input observations with MFCCs or other continuous representation of features.
- Explore deeper learning methods (e.g., RNN, CNN) to achieve potentially higher performance on big data.
- Measure model robustness in noisy settings or with in-the-wild background interference.

VI. References

[1] L. R. Rabiner, "*A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*," Proceedings of the IEEE, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[2] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "*A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains*," Annals of Mathematical Statistics, vol. 41, no. 1, pp. 164–171, 1970.

[3] A. J. Viterbi, "*Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm,*" IEEE Transactions on Information Theory, vol. 13, no. 2, pp. 260–269, Apr. 1967.

[4] L. R. Rabiner, J. G. Wilpon, and B. H. Juang, "*A Model-Based Connected-Digit Recognition System Using Either Hidden Markov Models or Templates*," Computer Speech & Language, vol. 1, no. 3–4, pp. 167–197, 1986.

[5] J. G. Wilpon, R. P. Mikkilineni, and S. Gokcen, "*Speech Recognition: From the Laboratory to the Real World*," AT&T Technical Journal, vol. 69, no. 5, pp. 14–30, 1990.

[6] J. G. Wilpon, L. R. Rabiner, C. H. Lee, and E. R. Goldman, "*Automatic Recognition of Keywords in Unconstrained Speech Using Hidden Markov Models*," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 38, no. 11, pp. 1870–1878, Nov. 1990.

[7] L. R. Rabiner and B. H. Juang, "*An Introduction to Hidden Markov Models*," IEEE ASSP Magazine, vol. 3, no. 1, pp. 4–16, Jan. 1986.

[8] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals and Systems*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 1997.

[9] B. Gold and N. Morgan, *Speech and Audio Signal Processing: Processing and Perception of Speech and Music*, 1st ed. New York, NY, USA: Wiley, 2000.

[10] K. Tokhura, “Speech feature extraction: the auto-correlation model and LPC cepstrum,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-26, no. 6, pp. 523–529, Dec. 1978.

[11] Y. Linde, A. Buzo, and R. M. Gray, “An Algorithm for Vector Quantizer Design,” *IEEE Transactions on Communications*, vol. 28, no. 1, pp. 84–95, Jan. 1980.