

# **Interim Report**

## **Level 4**

### **Automated Step-by-Step Visual Performance Guide Generation from Sinhala Demonstration Videos**

Group Name: The TechNovas

#### **Group Members**

<b>Index Number</b>	<b>Name</b>
204104H	Kularathna M.D.S.A.
204137K	Nethmini S.A.R.
204041K	Dilshan K.G.A.P.

Faculty of Information Technology

University of Moratuwa

2025

# **Interim Report**

## **Level 4**

### **Automated Step-by-Step Visual Performance Guide Generation from Sinhala Demonstration Videos**

Group Name: The TechNovas

#### **Group Members**

Index Number	Name
204104H	Kularathna M.D.S.A.
204137K	Nethmini S.A.R.
204041K	Dilshan K.G.A.P.

Supervised by: Dr. L. Ranathunga

Faculty of Information Technology

University of Moratuwa

2025

## **Abstract**

With the increasing availability of demonstration videos online, users often struggle to quickly access and comprehend essential information. This challenge is particularly significant for Sinhala demonstration videos, which are fewer in number and lack structured guides to highlight key details. As a result, Sinhala-speaking audiences face difficulties in efficiently extracting and understanding crucial steps, ingredients, or instructions from such videos.

This project addresses the absence of structured visual performance guides for Sinhala demonstration videos. The proposed solution employs speech-to-text technology to generate accurate transcriptions of the video content. Using Natural Language Processing (NLP), key points are extracted and organized into meaningful sections. Additionally, image processing techniques identify and extract relevant visual snapshots from specific timestamps, ensuring a synchronized alignment between text and visuals. The final output is a well-structured performance guide that presents essential information concisely and effectively.

The implementation process consists of multiple phases: first, transcribing Sinhala audio using speech-to-text technology; second, refining and structuring the transcriptions using NLP; third, extracting key visual elements through image processing; and finally, synchronizing the textual instructions with the selected visuals to generate a comprehensive performance guide. The system will primarily focus on demonstration videos related to cooking and technology, with a maximum duration of 10 minutes.

This solution enhances accessibility for Sinhala-speaking users, enabling them to efficiently extract vital information without the need to watch entire videos. The project leverages existing technologies such as open-source Speech-to-Text APIs, NLP tools, and image processing techniques to deliver a practical and user-friendly tool for content summarization. By addressing a critical gap in content accessibility, this project offers a novel approach to improving the usability of Sinhala demonstration videos.

## Table of Contents

Chapter 1	1
Introduction	1
1.1 Introduction	1
1.2. Background & Motivation	1
1.3. Problem in Brief	2
1.4. Aim & Objectives	2
1.4.1 Aim	2
1.4.2 Objectives	2
1.5 Proposed Solution	3
1.5.1 Module 1: Correction and Meaningful Refinement of Transcriptions to enhance Clarity and Accuracy	3
1.5.2 Module 2: Extraction of Meaningful Instructions, Summarization and Conversion to Written Language	3
1.5.3 Module 3: Action/Object Detection and Image Selection Aligned with Instructions.	4
1.6 Summary	4
Chapter 2	5
Literature Review	5
2.1 Chapter Overview	5
2.2 Literature Review on Module 1 -Correction and Meaningful Refinement of Transcriptions to enhance Clarity and Accuracy	5
2.2.1 Introduction	5
2.2 Review of Other's Work	5
2.3 Literature Review on Module 2 -Extraction of Meaningful Instructions, Summarization and Conversion to Written Language	7
2.3.1 Introduction	7
2.3.2 Review of Other's Work	7
2.4 Literature Review on Module 3 -Frame Aligned Cooking Instructions via Action and Object Detection	8
2.4.1 Introduction	8
2.4.2 Review of Other's Work	9
Chapter 3	13
Technology adapted	13
3.1 Chapter Overview	13
3.2 Module 1 -Correction and Meaningful Refinement of Transcriptions to enhance Clarity and Accuracy	13

3.3 Module 2 -Extraction of Meaningful Instructions, Summarization and Conversion to Written Language	14
3.4 Module 3 -Frame Aligned Cooking Instructions via Action and Object Detection	16
3.5 Summary	16
Chapter 4	18
Approach	18
4.1 Chapter Overview	18
Chapter 5	24
Analysis and Design	24
5.1 Chapter Overview	24
5.2 Module 1: Correction and Meaningful Refinement of Transcriptions to enhance Clarity and Accuracy	24
5.3 Module 2: Extraction of Meaningful Instructions, Summarization and Conversion to Written Language	27
5.4 Module 3: Action/Object Detection and Image Selection Aligned with Instructions.	31
Chapter 6	34
Implementation	34
6.1 Chapter Overview	34
6.2 Datasets	34
6.3 Coding	37
Chapter 7	39
Discussion	39
7.1 Chapter Overview	39
7.2 Module 1: Correction and Meaningful Refinement of Transcriptions to Enhance Clarity and Accuracy	39
7.3 Module 2: Extraction of Meaningful Instructions, Summarization, and Conversion to Written	39
7.4 Module 3: Action/Object Detection and Image Selection Aligned with Instructions This module	40
References	41
<i>Appendix: Individuals Contribution to the Project</i>	43

## List of Figures

Figure 1 - How food changes appearance during cooking. In this sequence a cucumber is being finely chopped.....	9
Figure 2 - state identification problem definition .....	10
Figure 3-Samples from the Cooking State Recognition Challenge dataset .....	10
Figure 4- Examples of images of ingredients dataset. ....	11
Figure 5-Architectural Diagram-Module 1 .....	24
Figure 6 -Architectural Diagram-Module 2 .....	28
Figure 7-Architectural Diagram-Module 3.....	31
Figure 8-corrected transcribed text. ....	34
Figure 9-extracted key instructions.....	35
Figure 10-performance guide dataset.....	36
Figure 11 -audio extraction and transcription code. ....	37
Figure 12-frame extraction code .....	38

## Introduction

### 1.1 Introduction

As more demonstration videos become available online across various platforms, users often find it hard to access and understand the information. With so many videos to choose from, it can be overwhelming and frustrating for people to watch an entire video just to see if it contains the specific help they need. This issue is even more noticeable for Sinhala demonstration videos because there are fewer of them available compared to English videos. As a result, people who speak Sinhala may have a harder time finding the information they need. When they do find these videos, they might not have the time to watch the entire video or fully understand the key points. Many Sinhala videos lack structured guides that highlight important actions, ingredients, or instructions. As a result, viewers can miss critical details or waste time re-watching content.

The problem addressed in this project is the absence of performance guides for Sinhala demonstration videos. Visual Performance guides provide viewers with accurate, synchronized summaries that combine text and visual cues. These guides would allow users to quickly understand and follow the most important parts of the demonstration, without needing to watch the entire content.

To solve this, we aim to create a solution using speech-to-text technology combined with Natural Language Processing to generate accurate transcriptions and extract key information from Sinhala cooking tutorial videos [1]. By using image processing techniques, we will obtain key visuals by selecting the most suitable frames [2]. This will make sure that the visuals correspond to the instructions being delivered. Afterwards, we will synchronize both images and text in the correct order. This will create a clear visual performance guide that matches the transcript with the corresponding visuals, making it easier to understand and more engaging.

This solution uses various technologies to improve accessibility for Sinhala video content, saving time for viewers while ensuring they capture the most important details in very little time. It offers a practical and efficient way to extract useful information from demonstration videos.

### 1.2. Background & Motivation

The internet is a very familiar place to everyone for sharing knowledge and information on countless topics, including cooking, education, and skills training. It allows people to access demonstrations and resources from anywhere in the world. This makes learning more convenient and accessible for everyone. However, users often face challenges when consuming long-form video content, especially when looking for specific information. For Sinhala-speaking audiences, this problem is made worse by

the lack of effective tools that allow easy extraction of key information from video content. Currently, many users either miss important details or spend significant time re-watching segments to understand the critical parts of a demonstration.

Watching videos in Sinhala can be time-consuming, and users often find it difficult to quickly pinpoint key actions, ingredient lists, or step-by-step instructions. This lack of easily accessible performance guides or summaries is a major downside for viewers who want an efficient way to follow demonstrations. In contrast, English-language content has more developed solutions, including advanced video summaries and synchronization tools that do not exist or work well for Sinhala content.

The motivation for this project stems from the growing demand for efficient content summarization and the unique need to cater to Sinhala-speaking audiences who are underserved in this area. By addressing this gap, we aim to provide a solution that not only saves time but also improves user experience. Our team plans to use technologies such as Natural Language Processing, speech-to-text technology, Image processing technology and synchronization technologies, to deliver a solution to this problem. The proposed solution will use these technologies to offer concise, accurate visual performance guides from Sinhala demonstration videos.

### **1.3. Problem in Brief**

The main issue this project tackles is the absence of effective performance guides for Sinhala demonstrations available online. Specifically, viewers of Sinhala demonstrations do not have access to a tool that can automatically extract key information such as ingredients, steps, and important visuals, and present this information in an accurate, synchronized text-visual format. This gap in content accessibility makes it difficult for Sinhala-speaking users to engage with video content efficiently. This can result in a time-consuming and often frustrating experience.

By developing a solution that generates text, extracts key points [3] and creates visual summaries using speech-to-text, NLP technologies and image/video processing technologies this project looks to address this communication and accessibility gap for Sinhala video content [1].

### **1.4. Aim & Objectives**

#### **1.4.1 Aim**

To automatically generate visual performance guides from Sinhala demonstration videos, allowing viewers to quickly grasp key information through synchronized text and visuals.

#### **1.4.2 Objectives**

To achieve the above aim, our objectives are as follows.



1. Convert Sinhala audio to text, correct it and generate meaningful content.
2. Identify key points and create step-by-step instructions.
3. Capture and enhance key visuals to focus on important steps.
4. Generate a document by combining relevant visuals and instructions.
5. Enhance text accuracy, image selection, and performance guide quality using machine learning techniques.

## **1.5 Proposed Solution**

In this project, we propose a solution to generate visual performance guides from Sinhala demonstration videos, by using speech-to-text technology, Natural Language Processing algorithms, Image Processing and synchronization tools. The system will transcribe, summarize and obtain the steps mentioned in the demonstration video. Afterwards, it will obtain the correct visuals from the video which correspond to the steps that we have obtained earlier. Finally, we will synchronize the text and images we have derived in order to create a step-by-step visual performance guide as the end product. This solution will focus on demonstration videos in the cooking and technology-related domains. Videos considered will have a maximum duration of 10 minutes.

To solve the problem, we have suggested 3 modules.

### **1.5.1 Module 1: Correction and Meaningful Refinement of Transcriptions to enhance Clarity and Accuracy**

The aim of this module is to improve the clarity and accuracy of transcribed content by correcting errors and refining language. It ensures that transcriptions are coherent, contextually accurate, and free from ambiguities. This enhances overall data quality, making it suitable for subsequent processing, analysis, and automated evaluation tasks.

### **1.5.2 Module 2: Extraction of Meaningful Instructions, Summarization and Conversion to Written Language**

This module focuses on extracting essential instructions from transcribed text, summarizing the extracted important points, and converting them into clear written language. The objective is to identify the key points and create step by step instructions required for the performance guide.

### **1.5.3 Module 3: Action/Object Detection and Image Selection Aligned with Instructions.**

This module focuses on action and object recognition within the video content. The process involves breaking down video clips into distinct actions while simultaneously identifying relevant ingredients and tools. Based on the extracted information, instruction generation converts recognized actions and objects into simple, structured steps. These generated instructions are then mapped to corresponding Sinhala instructions derived from the audio. Additionally, the best representative frame from each video segment is selected to visually align with the generated instructions.

## **1.6 Summary**

This chapter provides an introduction to our research and presents the background and motivation for selecting the chosen area of study. Next, it outlines the core problem that the research aims to address, followed by a clear explanation of the overall aim and specific objectives. Additionally, three key modules that form essential components of the research framework are introduced, highlighting their significance in achieving the research goals.

The upcoming chapters of this report will explore various aspects of the study in detail. Chapter 2 will present an extensive literature review, examining related research within the context of the three modules introduced. Chapters 3, 4, and 5 will sequentially discuss the technologies utilized, the development methodology adopted, and the analysis and design of the proposed system. Chapter 6 will focus on the implementation phase, providing information about how the proposed system was developed. Finally, Chapter 7 will offer a discussion and conclusion, summarizing the research findings and contributions, and the report will conclude with references and an appendix.

# Literature Review

## 2.1 Chapter Overview

This chapter provides a detailed review of existing solutions and research related to generating automated performance guides from demonstration videos. It explores the problems identified by previous studies, the approaches they have employed, and the limitations of their methods. A summary of the current methodologies for transcription, key point identification, and visual guide generation is provided to establish a foundation for the proposed solution in this project. This section also addresses gaps in the literature and highlights the importance of integrating speech-to-text, natural language processing, and machine learning in improving the quality and accessibility of performance guides.

## 2.2 Literature Review on Module 1 -Correction and Meaningful Refinement of Transcriptions to enhance Clarity and Accuracy

### 2.2.1 Introduction

Text obtained by transcribing the audio extracted from Sinhala cooking tutorial videos may contain several errors, which may make it difficult to continue the process of creating performance guides. These errors can significantly hinder the subsequent stages of creating comprehensive performance guides, where clarity and accuracy are essential.

This section reviews existing research focused on refining transcriptions to produce coherent and meaningful text suitable for further processing. The review highlights the methods used to correct transcription errors, improve textual clarity, and maintain contextual accuracy. It also examines the effectiveness of different techniques and tools, comparing their performance and identifying existing gaps that future research could address.

### 2.2 Review of Other's Work

The first system focuses on detecting grammatical errors in Sinhala sentences, particularly addressing subject-verb agreement [4]. It identifies the subject, verb, and object in a sentence through a translation-based approach, converting Sinhala text into English [4] for easier error analysis. However, limitations exist in terms of translation accuracy, as complex sentences or idiomatic expressions may be misinterpreted, affecting the analysis. Additionally, challenges arise in correctly identifying subjects and verbs in sentences with irregular structures or polysemy. The reliance on predefined rules for gender and animacy classification may lead to errors in verb conjugation, and

context-dependent ambiguities may further complicate the process. These issues suggest a need for more reliable, context-aware techniques to improve the system's overall performance.

Another system [5] integrates a hybrid approach for grammatical error detection and correction in Sinhala text by combining rule-based and machine learning techniques. It uses a part-of-speech tagger and a morphological analyzer to process sentences, followed by a pattern recognition module to identify grammatical structures [5]. Errors are detected using predefined grammar rules, and corrections are suggested using a decision-tree model. However, the rule-based model might not cover all grammatical nuances, and the speech-to-text conversion can lead to inaccuracies due to phonetic variations or noise.

Sinhala grapheme-to-phoneme conversion models [6] can also be useful when correcting transcriptions. When transcribing audio or speech to text, errors can occur due to misinterpretation of sounds, especially if the system struggles with differentiating similar-sounding words or letters in Sinhala.

A system for spelling error detection and correction in Sinhala uses a mix of both phonetic similarity and statistical language models and relies on unigram, bigram, and trigram frequencies from a large corpus [7]. It suggests possible corrections based on grapheme similarity. However, it's not without limitations. It struggles when handling complex sentences, context-aware disambiguation, and OOV words. Other systems use neural models for spelling correction, using techniques like sequence-to-sequence, language modeling, and sequence labeling [8]. These models help automatically detect and correct spelling errors by learning patterns in large datasets. However, limitations include the requirement for substantial training data and the dependency on language-specific models.

There are reports of systems that restore punctuation in unpunctuated speech transcripts using a two-stage fine-tuning approach with a BERT model [9]. It first trains on movie subtitles using n-gram similarity for relevant data sampling, then fine-tunes on noisy ASR phone transcripts. However, domain differences, annotation complexity, and real-time deployment challenges limit performance.

A Sinhala speech recognition system has focused on automatic speech recognition and machine translation for Sinhala, offering features such as noise removal, speaker diarization, and audio-to-text conversion [10]. However, limitations include the scarcity of resources for Sinhala language models, difficulties with handling mixed-language inputs, and challenges in grammar correction. SinMorphy is a Sinhala morphological analyzer that breaks down Sinhala words into their base forms, adding morphemes to produce different grammatical variations [11]. The system incorporates rules for handling nouns, verbs, adjectives, and particles, which are used to create various word forms. However, SinMorphy has limitations, including difficulties in handling

compound verbs and words that may not be in its vocabulary. Additionally, handling irregular forms and exceptions remains a challenge.

## **2.3 Literature Review on Module 2 -Extraction of Meaningful Instructions, Summarization and Conversion to Written Language**

### **2.3.1 Introduction**

The extraction of meaningful instructions from transcribed text is an important step in creating structured and actionable guides, particularly when dealing with instructional content such as Sinhala cooking tutorials. Transcriptions, while offering a textual representation of spoken language, often contain disorganized or fragmented information that needs to be processed and transformed into clear, actionable steps. This process is particularly challenging with Sinhala, as the language's unique syntax, idiomatic expressions, and informal speech patterns can complicate the extraction of instructions.

This section reviews existing research on methods and technologies used to extract and refine instructions from transcribed Sinhala text. The focus is on identifying key techniques that have been used to improve the quality of the extracted instructions, ensuring that the final output is both accurate and user-friendly. The review also highlights the challenges associated with processing Sinhala text and discusses the potential for future advancements in this area.

### **2.3.2 Review of Other's Work**

The first project proposes a Sinhala tokenizer [12] which uses two main approaches for text processing, punctuation-based tokenization and dependent word tokenization. The punctuation-based approach handles 15 punctuation marks by analyzing their specific use cases and applying rule-based mechanisms to accurately tokenize them [12]. The dependent word tokenization method focuses on combining words that are contextually related, which reduces word fragmentation and enhances meaning retention. However, the system faces limitations, such as difficulty handling domain-specific terms, complex sentence structures, and new or slang words not present in the training corpus. Additionally, its reliance on rule-based methods may fail to capture semantic context effectively, which can affect the accuracy of key point extraction.

A dynamic stopwords removal system automatically generates a stopwords list for the Sinhala language using a hybrid approach combining term frequency and inverse document frequency [13]. This method processes over 90,000 documents to identify stopwords, which is crucial for natural language processing tasks [13]. However, the approach has limitations, such as the reliance on a manual cut-off point, which may not be optimal, and the need for further refinement in practical applications.

There is also work which uses a simplified Lesk algorithm for Word Sense Disambiguation [14] using Sinhala WordNet. It compares the glosses of target word senses with the surrounding context, ignoring stop words to improve accuracy and efficiency. However, limitations include the absence of a morphological analyzer for Sinhala, affecting overlap accuracy, and a limited sense-annotated corpus due to the recent development of Sinhala WordNet. Another project has focused on Sinhala text simplification [15] using baseline models like mBART, mT5, and pivot-based methods, alongside ITTL-based models that incorporate auxiliary tasks such as translation, paraphrasing, and HRL text simplification [15]. It addresses zero-shot learning challenges by creating synthetic datasets. However, limitations include the absence of large parallel datasets, computational constraints, and potential inaccuracies from translation and paraphrase mining processes.

Another research on various text summarization algorithms [3] focuses on TF-IDF, Text Rank, and Seq2Seq for extractive summarization in Sinhala. TF-IDF evaluates the importance of words based on their frequency, while Text Rank uses a graph-based approach to rank sentences for summarization. Seq2Seq, a neural network model, is primarily used for abstractive summarization but can be adapted for extractive tasks [3]. Limitations include the need for large datasets and the challenge of accurately handling the linguistic nuances of Sinhala, which lags behind more widely researched languages.

Another system focuses on Sinhala text categorization using rule-based classification algorithms [16], specifically addressing the challenges of Sinhala text pre-processing. It explores various algorithms such as OneR, ZeroR, PART, Decision Table, JRip, and Ridor for classification. However, its limitations include the scarcity of publicly available Sinhala datasets, reliance on manually classified data, and the absence of freely available stop word lists and stemming libraries for Sinhala.

## **2.4 Literature Review on Module 3 -Frame Aligned Cooking Instructions via Action and Object Detection**

### **2.4.1 Introduction**

This section focuses on action and object recognition within video content, specifically in cooking and food preparation. The process involves breaking down video clips into distinct actions such as peeling, cutting, mixing, and heating while simultaneously identifying relevant ingredients and tools. Based on the extracted information, instruction generation converts recognized actions and objects into simple, structured steps. These generated instructions are then mapped to corresponding Sinhala instructions derived from the audio to ensure consistency and accuracy. Additionally, the best representative frame from each video segment is selected to visually align with the generated instructions, creating a structured and synchronized performance guide.

This approach enhances accessibility and comprehension, allowing users to quickly grasp essential steps without watching the entire video.

#### 2.4.2 Review of Other's Work

The task of action and object recognition in cooking videos has become a focal point in the intersection of computer vision and food preparation. Various methods have been proposed to address the challenges involved in detecting and classifying actions and objects in cooking scenarios. This literature review explores key research that contributes to the development of techniques used in cooking action recognition and food ingredient identification.

Ryoo et al. introduced iVAT, an interactive video annotation tool designed to facilitate the recognition of cooking actions. The tool enables efficient annotation of cooking videos by decomposing actions into specific tasks such as stirring, chopping, and frying. This method helps in improving the training of recognition models for cooking-related actions, as demonstrated in Figure.1.1, where the process of action recognition and annotation is visualized [17].



*Figure 1 - How food changes appearance during cooking. In this sequence a cucumber is being finely chopped*

The approach is significant for large-scale action recognition tasks, particularly in video datasets where labeled annotations are required for model training.

Aboubakr et al. tackled the problem of food localization within cooking videos. Their approach focused on recognizing and localizing food items in videos to create a more structured representation of the cooking process [18]. This work aids in identifying key ingredients that are part of specific cooking actions, contributing to the broader task of cooking process understanding. Similarly, Bagdanov et al. explored the use of multiple kernel learning (MKL) for improving cooking action recognition, which combines different feature representations to enhance recognition accuracy. Their work highlighted the importance of combining various feature learning strategies to better capture the diverse nature of cooking actions [19].

Venkataramanan et al. proposed a generative model called Cook-Gen, which models cooking actions robustly from recipes. This model uses both ingredients and their associated actions to generate cooking sequences, offering a generative approach to cooking action recognition [20]. The authors demonstrated that this model could help in recognizing novel cooking actions that were not present in the training data, further advancing the flexibility and scalability of action recognition systems.

Object recognition in cooking-related images was addressed by Jelodar et al , who focused on identifying object states during food preparation. Their work, shown in Figure.1.2, illustrates how object states change during the cooking process. By analyzing cooking images, the study identified key states of ingredients that are essential for generating cooking instructions and monitoring cooking progress [21].

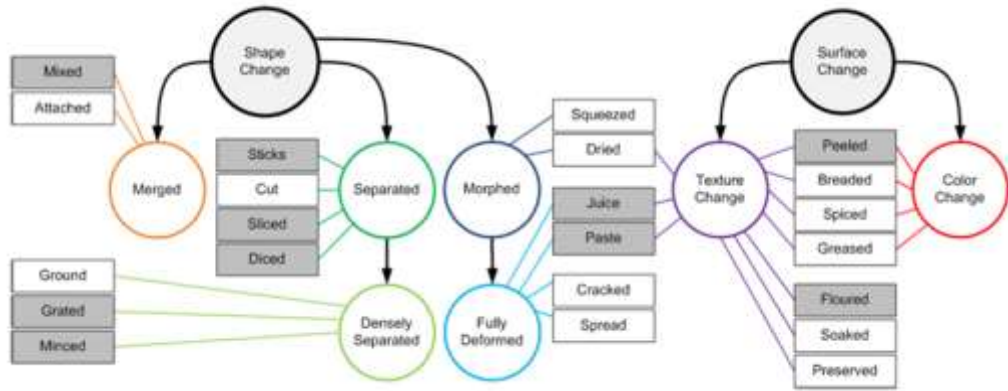


Figure 2 - state identification problem definition

This approach provides valuable insights into how cooking objects transition through various stages, which is crucial for creating step-by-step cooking guides.

The use of Vision Transformers (ViTs) in cooking state recognition was explored by Khan et al, who applied ViTs to analyze cooking videos. ViTs have proven effective in processing image data for action and state recognition due to their ability to capture long-range dependencies in the data [22].



Figure 3-Samples from the Cooking State Recognition Challenge dataset



In Figure.1.3, the authors demonstrate how their ViT model recognizes cooking states, achieving high accuracy in identifying different cooking stages such as boiling, frying, or sautéing. This work offers a promising direction for improving cooking state recognition, especially in real-time applications.

Zhu and Dai focused on identifying food ingredients from dish images using deep learning techniques. They applied a convolutional neural network (CNN) to classify food items based on visual features from dish images. This research highlighted the challenges of ingredient identification due to variations in cooking styles and presentation. Figure.1.4 presents a detailed performance comparison of different models, showing the impact of deep learning on ingredient identification accuracy. This approach complements action recognition by providing a way to automatically identify the ingredients used in cooking actions [23].

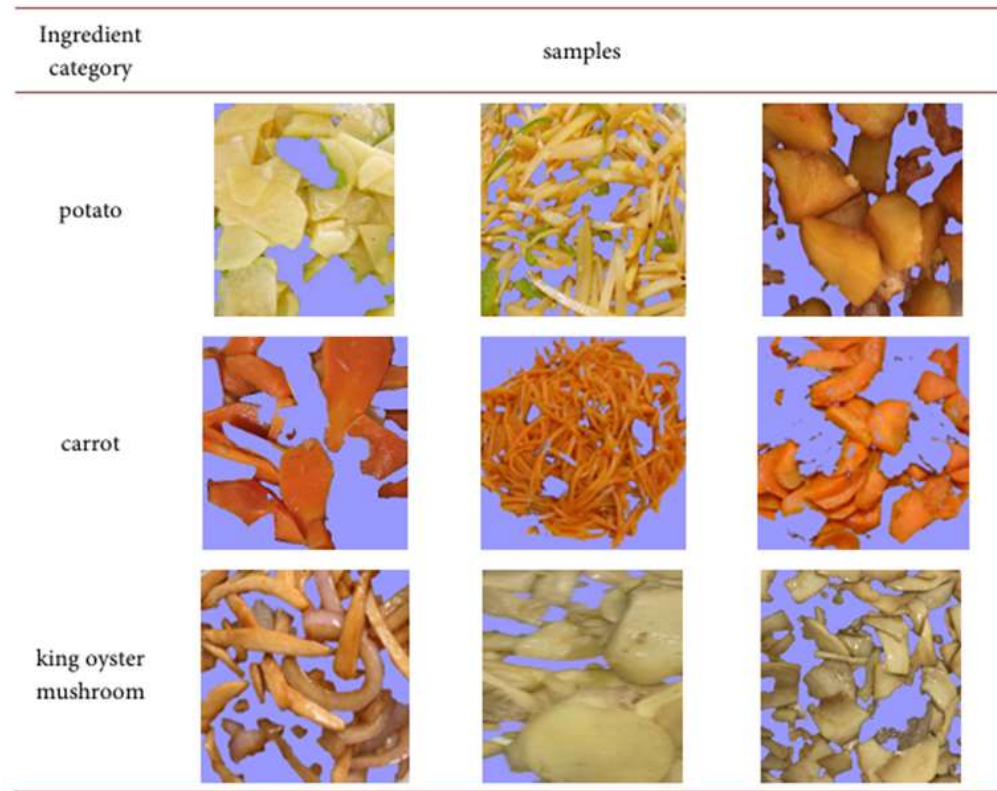


Figure 4- Examples of images of ingredients dataset.

Finally, Álvarez et al. proposed a cooktop sensing approach based on YOLO (You Only Look Once), an object detection algorithm. Their work emphasized the need for real-time object detection on cooking surfaces to identify food items and cooking tools during food preparation [24]. This research complements other action recognition studies by offering a precise method for detecting objects in cooking videos. Hossain et al. conducted a systematic literature review on deep learning techniques for video processing, including applications to cooking videos. Their review provides a comprehensive overview of the methods and challenges in using deep learning for action and object recognition in videos. The study underscores the importance of

applying advanced deep learning models to video data to improve accuracy in dynamic environments like cooking [25].

In addition to action and object recognition, optimizing the selection of frames from cooking videos is crucial for generating a concise summary of the cooking process. Krylov et al. proposed an algorithm for choosing the best frame in a video stream, which is particularly useful for summarizing video content and selecting key moments that represent the cooking action effectively [26]. Their algorithm evaluates various frames and selects the most informative one, which can be used as a reference for generating cooking instructions or summarizing the video content.

The body of research in cooking action recognition, food ingredient identification, and cooking state recognition is rapidly advancing. By combining generative models, object detection algorithms, and deep learning techniques, these studies contribute significantly to the development of automated systems for cooking process analysis and instruction generation. Together, they offer a multi-faceted approach to recognizing actions and objects in cooking videos, with the potential to improve both user experience and efficiency in food preparation applications.

### Technology adapted

#### 3.1 Chapter Overview

This chapter outlines the technologies selected for implementing our solution and explains the rationale behind each choice. Since our solution comprises three distinct modules, each will be implemented using different approaches and machine learning techniques depending on their specific requirements.

#### 3.2 Module 1 -Correction and Meaningful Refinement of Transcriptions to enhance Clarity and Accuracy

To solve the problem of correcting errors in transcribed Sinhala text from speech-to-text services, several advanced natural language processing and machine learning techniques have been adapted. These techniques were chosen based on their ability to handle specific error groups and improve transcription accuracy, ensuring that the final output is both linguistically and contextually correct.

1. **Google Speech Recognition API:** The first step in this process involves extracting the audio from the video and transcribing it into text using the Google Speech Recognition API. This API is a powerful tool for transcribing speech to text and is particularly useful in situations where speech patterns are complex, such as in Sinhala. While it offers high accuracy in transcribing audio, there are still challenges, such as phonetic similarities and transcription errors. Therefore, the next steps focus on addressing these issues.
2. **Edit Distance Algorithms:** For the detection and correction of spelling and character-level errors, the distance algorithm is used. This algorithm calculates the minimum number of edits (insertions, deletions, substitutions) needed to transform one string into another. It is ideal for detecting small spelling variations or missing characters, which are common in speech-to-text conversions. By comparing the transcribed word with a dictionary, it identifies the closest match based on the edit distance.
3. **Sinhala Morphological Analyzer:** To address word form errors, such as incorrect verb tenses or word forms, a Sinhala morphological analyzer is used. Sinhala morphology involves identifying the root word and its correct form. This tool is suitable because it considers language-specific rules, which are necessary for languages like Sinhala, where word forms change based on tense, number, and other grammatical features. It uses pattern recognition to analyze and correct the form of words in the transcription (Weerasinghe, 2013).

4. **BERT and FastText Models for Contextual Correction:** For detecting and correcting completely different words, gibberish, or nonsense words, a combination of BERT and FastText models is used. BERT is particularly suitable for contextual analysis, as it takes the entire sentence into account and can predict the most appropriate word based on context. FastText, on the other hand, is effective for handling out-of-vocabulary words and finding semantically similar words. Together, they allow the system to identify and correct words that are completely off-track in terms of meaning and context.
5. **Language Detection Models for Foreign Words:** The algorithm to detect and correct foreign words in the transcription uses a simple language detection model. This model can identify English words and map them to their Sinhala equivalents using a transliteration dictionary. This approach is effective because it avoids relying on complex language models for this specific task. It focuses instead on a rule-based mapping system to provide accurate transliteration.
6. **Transformers for Punctuation and Sentence Structure:** To address punctuation and sentence structure errors, transformer models, such as BERT, are adapted for punctuation prediction. The BERT model is fine-tuned to predict punctuation based on the surrounding words and provides grammatically correct output.
7. **Word Pair Matching Models:** For checking and correcting unmatching consecutive words, a statistical or machine learning model that checks word pair probabilities is used. This model examines the likelihood of two consecutive words occurring together in natural Sinhala language. By using a language corpus, this model is able to make sure that word pairs in the transcription match both grammatically and contextually.
8. **Sentence Splitting Algorithms:** To handle long sentences and split them into smaller, more readable sentences, a sentence segmentation algorithm based on punctuation and sentence length is used. This method is efficient because it focuses on linguistic breaks, such as conjunctions, commas, and periods.

### **3.3 Module 2 -Extraction of Meaningful Instructions, Summarization and Conversion to Written Language**

To address the task of processing Sinhala-language video transcriptions and creating an effective performance guide, several natural language processing techniques and technologies are adapted. The primary objective is to extract and classify actionable instructions while removing irrelevant content. It is also to make sure that the final output is concise, accurate, and instructional.

1. **Sentence Splitting and Tokenization:** The first step involves splitting the transcription into individual sentences and breaking each sentence into words. For Sinhala, this step requires custom segmentation due to the unique nature of the language's punctuation and syntax. While general NLP libraries

such as NLTK and spaCy offer sentence segmentation, their handling of Sinhala-specific punctuation may be inadequate. Therefore, custom rules based on Sinhala linguistic patterns are used for accurate sentence splitting. This allows for better handling of special punctuation marks such as "!" and "?" in Sinhala. Tokenization follows, using space-based splits to handle Sinhala words accurately. With tokenization, essential terms are captured for further analysis.

2. **Stop Word Removal:** To increase the relevance of the extracted words, a custom stop word list for Sinhala is used. Common words like "අපි", "එක", and "හා" are removed, allowing the focus to shift towards more meaningful words like action verbs and ingredients. This is an important step as it reduces the noise in the data, which is often problematic in Sinhala because libraries like NLTK may not provide sufficient support for specific word removal in Sinhala.
3. **Feature Extraction and Classification:** The next step involves the identification of specific sentence types, such as ingredients, actions, tips, and explanations. NLP-based feature extraction techniques are used to classify sentences according to their function. For example, the detection of units like "ග්‍රෑම්" (grams) or verbs like "එකතු කරනවා" (add) allows for the identification of ingredient-related and action-related sentences. Custom algorithms look for patterns specific to Sinhala grammar and vocabulary. It makes sure that the classification corresponds with the linguistic structure of the language.
4. **Simplification and Sentence Summarization:** After classifying and tagging the sentences, the goal is to simplify and combine similar sentences for clearer instructions. For example, multiple sentences detailing the addition of ingredients are condensed into a single concise instruction. This process benefits from sentence summarization techniques that extract key details and combine them into a more understandable format. This makes the instructional content more efficient and easier to follow.
5. **Written Language Conversion:** Since spoken Sinhala is more informal than written Sinhala, a conversion process is used to adapt the language from spoken to formal instructional style. Phrases like "කරගන්න" (do) are replaced with more formal equivalents such as "කර ගත යුතුයි" (should be done). This guarantees that the language is consistent with instructional norms, while increasing readability and clarity for the end user.
6. **Final Cleaning and Formatting:** The final step involves checking if there is consistent spacing, correct punctuation, and proper grammar. Custom scripts are used to clean the text to make sure there are no extraneous spaces and that punctuation marks such as full stops and commas are correctly placed. Additionally, the sentence structure is refined to align with standard Sinhala writing norms.

### 3.4 Module 3 -Frame Aligned Cooking Instructions via Action and Object Detection

Technology Adapted Advanced technologies have been adopted to solve the complex problem of generating accurate, contextually meaningful, and well-structured visual performance guides from Sinhala demonstration videos. These technologies were carefully selected to address challenges related to action recognition, object detection, instruction generation, Sinhala transcription mapping, and frame selection. Each technique plays a crucial role in ensuring that the final output is both linguistically and visually coherent.

1. **Action recognition:** Deep learning-based models such as Convolutional Neural Networks (CNNs) and Transformer-based architectures are used to analyze and classify different cooking actions from the video. Pretrained models like I3D (Inflated 3D ConvNet) and SlowFast Networks are considered to accurately detect and segment actions such as chopping, mixing, frying, and baking. This ensures that each cooking step is identified correctly for further processing.
2. **Object recognition:** This is achieved using advanced object detection models such as YOLO (You Only Look Once) and Faster R-CNN, which enable the identification of ingredients and kitchen tools present in each frame. These models help associate actions with the correct objects, such as detecting a knife during a cutting action or recognizing a frying pan during a frying action. To further refine object recognition, image augmentation techniques are applied to improve accuracy across diverse video inputs.
3. **Instruction Generation:** Once actions and objects are identified, instruction generation is performed using Natural Language Processing (NLP) techniques. A rule-based or transformer-based model, such as T5 (Text-to-Text Transfer Transformer), is used to generate structured cooking instructions based on recognized actions and objects. These generated instructions are then mapped to their corresponding Sinhala translations.
4. **Video processing and frame selection:** OpenCV and FFmpeg are utilized to break the video into individual frames and extract the most relevant ones for each instruction. A ranking algorithm evaluates frames based on clarity, action visibility, and object presence to ensure that the most informative frame is selected. The final selected frames are then paired with their respective instructions to create a structured instructional guide.

### 3.5 Summary

By integrating deep learning-based action and object recognition, rule-based NLP, machine translation models, and frame selection techniques, the system effectively

processes and organizes Sinhala-language demonstration videos and transcriptions. The use of custom stop word lists, classification algorithms customized for Sinhala language features, and formal language conversion ensures higher accuracy, better readability, and contextually meaningful performance guides. This approach addresses the unique challenges posed by Sinhala language structure, syntax, and speech-to-text transcription, making the final output accurate, relevant, and accessible to Sinhala-speaking users.

# Approach

### 4.1 Chapter Overview

This chapter discusses the approach adopted for implementing the proposed solution. Each sub-module will be thoroughly examined, with an explanation of the methods applied.

### 4.2 Module 1: Correction and Meaningful Refinement of Transcriptions to enhance Clarity and Accuracy

In this module, we aim to improve the accuracy and reliability of Sinhala speech-to-text transcriptions through a multi-step error correction approach. It integrates advanced natural language processing techniques, machine learning algorithms, and rule-based systems..

**Inputs:** The input consists of audio files in Sinhala which are extracted from video files of Sinhala cooking tutorials. These audio files are processed using the Google Speech Recognition API to convert speech into text with timestamps. This raw transcription is then analyzed for potential errors, such as spelling mistakes, incorrect word forms, joined or split words, gibberish, and foreign language usage.

**Process:** The core process involves applying different algorithms for each error group identified in the transcriptions:

1. **Spelling and Character-Level Errors:** I am using an edit distance algorithm that calculates the minimum number of operations (insertions, deletions, substitutions) needed to transform one word into another. This helps identify minor spelling errors and fix them by matching transcribed words to dictionary entries.
2. **Word Forms Errors:** Using a Sinhala morphological analyzer, I break down words into their components, identify their root forms, and correct any verb tense or word form errors based on linguistic patterns.
3. **Errors Regarding Incorrectly Joined Words:** By checking the transcribed text against a dictionary of valid words, I identify incorrectly joined or split words. I then apply language models to reassemble the words correctly.
4. **Completely Different Words or Gibberish:** To handle completely incorrect words or gibberish, we use a combination of BERT and FastText models. BERT helps with contextual correction by checking if a word replacement fits within the sentence, while FastText provides semantically similar word suggestions when a word does not exist in the Sinhala lexicon.



5. Foreign Words: A language detection model identifies English words in the transcribed text, and a rule-based mapping algorithm transliterates them into their closest Sinhala equivalents.
6. Punctuation and Sentence Structure Errors: A fine-tuned BERT transformer model helps predict and insert missing punctuation marks.
7. Unmatching Consecutive Words: A probability model checks for low-probability word pairs and indicates errors in word usage or order. These pairs are reordered or replaced to improve sentence structure.
8. Long Sentence Splitting: The algorithm detects sentences that exceed a predefined length threshold and splits them into smaller, more manageable segments. It uses punctuation and conjunctions as natural split points for improved readability.

Outputs: The final output is a corrected version of the initial transcription, with all errors addressed in the following manner: accurate spelling, proper word forms, corrected word boundaries, contextually appropriate words, correct punctuation, well-structured sentences, and appropriate transliterations of foreign words.

Technology Implementation:

- Google Speech Recognition API for transcribing audio into text
- BERT for contextual error correction and punctuation prediction
- FastText for semantic similarity and word validation
- Sinhala Morphological Analyzer for detecting and correcting word form errors
- Edit Distance Algorithm for handling spelling and character-level errors
- Rule-Based Mapping Algorithm for transliterating foreign words into Sinhala

By using these technologies together, I make sure that the system can correct a wide range of transcription errors. Therefore, it is suitable for producing high-quality, contextually accurate Sinhala transcriptions.

Through a combination of NLP techniques and algorithms, this approach significantly improves the accuracy of Sinhala speech-to-text transcriptions and addresses common transcription errors and produces more reliable, readable text outputs.

### 4.3 Module 2: Extraction of Meaningful Instructions, Summarization and Conversion to Written Language

In this module, I am developing an expert system to extract and structure key instructional content from Sinhala-language video transcriptions. The objective is to create a clean and structured performance guide from raw transcriptions and get rid of unnecessary information while retaining only the essential ingredients, action steps, and instructions. This will improve the efficiency of content delivery for instructional videos, such as cooking tutorials.

Inputs: Raw Sinhala transcription of an instructional video, which may contain greetings, descriptions, tips, action steps, ingredients, and calls to action.

Process:

1. **Sentence Splitting (Segmentation):** Transcription text is split into individual sentences based on punctuation marks specific to Sinhala (such as full stops and question marks).
2. **Tokenization (Breaking into Words):** Sentences are tokenized into individual words to identify key terms and actions, using a combination of traditional libraries (like NLTK) and custom rules specific to Sinhala.
3. **Stop Word Removal:** A custom stop-word list for Sinhala is used to remove common, less meaningful words to focus on the core content.
4. **Feature Extraction (Identify Sentence Type):**

Sentences are classified as:

- **INTRO** for greetings and introductory phrases.
- **INGREDIENT** for mentions of materials or tools required.
- **ACTION** for the core instructional steps.
- **DESCRIPTION** for detailed explanations.
- **TIP** for optional suggestions or alternatives.
- **ENGAGEMENT** for social media calls to action.

This classification is based on predefined rules for recognizing sentence patterns.

5. **Sentence Classification and Tagging:** Each sentence is labeled with its type (e.g., **INGREDIENT**, **ACTION**) and tagged for relevance (**KEEP** or **REMOVE**).
6. **Sentence Filtering:** Only sentences tagged as **INGREDIENT** and **ACTION** are retained, as they contain the required information needed for the final guide.
7. **Sentence Summarization:** Multiple sentences conveying the same information (e.g., different quantities of ingredients) are combined into a single concise instruction.

8. Sinhala Writing Style Conversion: Restructured content in speaking style is transformed into formal, written Sinhala. The resulting text follows the syntax and grammatical rules typically used in instructional content.
9. Final Cleaning and Formatting: The final output is cleaned for consistency in punctuation, spacing, and grammar for readability and professionalism.

Outputs:

A clean, set of instructions in formal written Sinhala that only includes:

- Ingredients (with their respective quantities).
- Action steps (formatted with formal language).
- No extraneous information such as greetings, tips, or calls to action.

Technology:

- Natural Language Processing (NLP): Used for sentence segmentation, tokenization, and stop word removal. Libraries like spaCy and NLTK will be used with custom rules for Sinhala.
- Custom Algorithms: For identifying key sentence categories (e.g., INGREDIENT, ACTION) and filtering out irrelevant content (e.g., INTRO, TIP).
- Text Processing Libraries: To ensure correct formatting and grammar for the final guide in written Sinhala.

By adopting this approach, the system simplifies the content extraction process and makes sure that the transcribed text is converted into clear, actionable instructions, all while accommodating the nuances of the Sinhala language.

#### **4.4 Module 3: Frame Aligned Cooking Instructions via Action and Object Detection**

In this module focuses on developing a system that automates the generation of structured, step-by-step cooking instruction guides from Sinhala video demonstrations. By leveraging action and object recognition techniques, this system ensures that cooking instructions are clear, visually supported, and aligned with transcribed Sinhala speech. This solution enhances accessibility, particularly for users who prefer visual guidance and structured learning while cooking.

Inputs

Video files of Sinhala cooking demonstrations  
Sinhala instructions derived from the audio

## Process

### 1. Preprocessing

Before extracting meaningful content, the system preprocesses the input videos to enhance clarity and extract key information:

**Video Enhancement:** Improves video quality by reducing noise and stabilizing frames.

**Frame Extraction:** Selects key frames at relevant timestamps for further analysis.

### 2. Action and Object Recognition

The core of the system lies in recognizing distinct cooking actions and identifying objects such as ingredients and tools:

**Action Recognition:** Utilizes deep learning models like CNNs or Transformer-based architectures to detect cooking-related activities (e.g., washing, peeling, cutting, frying).

**Object Detection:** Employs YOLO or Faster R-CNN to recognize ingredients and kitchen tools associated with each step, ensuring accurate instruction generation.

### 3. Instruction Generation

After recognizing actions and objects, the system constructs step-by-step instructions in a structured format:

**Combining Actions and Objects:** The system formulates instructions using a consistent pattern: Action + Object + (Optional Additional Details) (e.g., Cut ingredient 1 into thin slices).

**Simplification and Optimization:** Ensures instructions are clear and concise while maintaining accuracy.

### 4. Mapping with Sinhala Instructions

To ensure consistency between the generated instructions and the transcribed Sinhala speech, the system applies:

**Text-Matching Algorithms:** Compares generated instructions with transcriptions for alignment.

**Machine Translation Models:** In cases where an exact match is unavailable, translation models like MarianMT or Google Translate API convert generated text into Sinhala.

### 5. Frame Selection

Selecting an appropriate frame for each instruction enhances the clarity of visual guidance:

Frame Evaluation: Assesses the extracted frames based on clarity and relevance.

Best Frame Selection: Identifies the most informative frame to represent each instruction, ensuring a clear visual reference for users.

## Outputs

Step-by-step cooking instructions extracted from video content

Mapped Sinhala transcriptions aligned with each instruction

Relevant video frames corresponding to each cooking step

## Technology Implementation

- To achieve these objectives, the system integrates a combination of machine learning and deep learning techniques:
- Action Recognition: CNN or Transformer-based models
- Object Detection: YOLO or Faster R-CNN for identifying ingredients and tools
- Instruction Mapping: Text-matching algorithms and machine translation models
- Frame Selection: Frame ranking algorithms for choosing the best representative frames

By systematically combining these technologies, this approach enables the automated generation of structured and visually supported cooking instructions. The system ensures that users can easily follow Sinhala cooking demonstrations with enhanced accessibility, efficiency, and comprehension.

## 4.5 Summary

This chapter outlines a three-module approach for generating accurate Sinhala cooking guides from video transcriptions. It covers transcription correction using NLP and machine learning, extraction of key instructions with sentence classification and summarization, and alignment of cooking steps with video frames through action and object recognition. The integrated technologies ensure clear, contextually accurate, and visually supported instructional content.

## Analysis and Design

### 5.1 Chapter Overview

The system can be divided into multiple modules, each performing specific tasks. The flow from one module to another ensures the creation of a flawless performance guide at the end of the process.

### 5.2 Module 1: Correction and Meaningful Refinement of Transcriptions to enhance Clarity and Accuracy

The module aims to process Sinhala transcriptions from video content, identify errors, and correct them systematically using a series of algorithms targeting specific error groups. This section outlines the design, with a focus on the modules, their responsibilities, and how they interact within the system.

#### Description of Module

As described in the previous chapter, this module focuses on converting speech to corrected, understandable text.

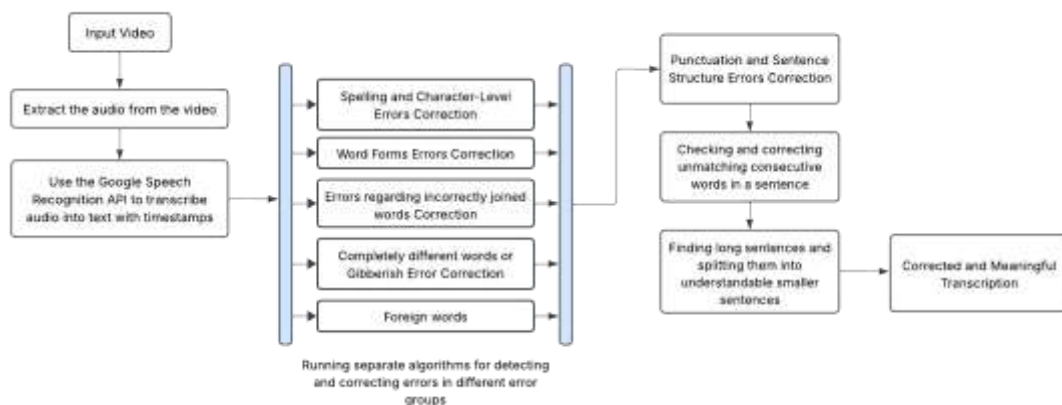


Figure 5-Architectural Diagram-Module 1

#### Step 1: Extract the Audio from the Video

The first step in the process involves extracting the audio from the video source. This is a necessary step as it isolates the spoken content, enabling the system to focus only on the audio for transcription. Tools like FFmpeg or other media processing libraries can be used to extract the audio from the video file, saving it in a suitable audio

format such as WAV or MP3. Once the audio is extracted, it can be passed into the speech recognition API, where the transcription process will begin.

### **Step 2: Use Google Speech Recognition API to Transcribe Audio into Text with Timestamps**

In the second step, the extracted audio is passed to the Google Speech Recognition API, which transcribes the audio into text. This API is highly effective for converting speech into written words, and it also provides timestamps, which are essential for aligning the transcription with the corresponding segments of the audio. This step provides the necessary text data that will be used for further analysis and correction in the subsequent steps.

### **Step 3: Running Separate Algorithms for Detecting and Correcting Errors in Different Error Groups**

After transcription, the system processes the text by running several algorithms to detect and correct errors in various categories. This is done to obtain an output which is as accurate and readable as possible. The errors are divided into five key groups: spelling and character-level errors, word forms errors, incorrectly joined words, completely different or gibberish words, and foreign words. Each error group is tackled separately with specialized algorithms designed to identify the specific types of mistakes associated with that group. The system uses a combination of algorithms, including edit distance calculations, morphological analysis, and machine learning models, to detect and fix each type of error.

#### **Error Group 1: Spelling and Character-Level Errors**

Spelling and character-level errors are common in speech-to-text transcriptions, where words are often misspelled due to phonetic similarities or typographical mistakes. To detect and correct these errors, the system uses an algorithm that calculates the minimum edit distance between the transcribed word and valid words in the dictionary. The edit distance algorithm identifies the smallest number of insertions, deletions, or substitutions required to convert one word into another. Once the incorrect word is identified, the algorithm finds the closest match from the dictionary. It then replaces the error if the distance is within an acceptable threshold.

#### **Error Group 2: Word Forms Errors**

Word forms errors occur when words are used in an incorrect form, such as improper verb tenses or incorrect word endings. This step uses a Sinhala morphological analyzer to break down each word into its base components. The analyzer identifies the root word and checks for its correct form in the dictionary. It applies linguistic

rules to predict the proper word form based on context. For example, a verb form may need to be corrected from a plural form to a singular one or from one tense to another.

### **Error Group 3: Errors Regarding Incorrectly Joined Words**

Incorrectly joined words occur when words are run together due to mispronunciations. The system addresses this issue by checking each word against a dictionary of valid words. If a word doesn't match, the system tries to split the word into smaller, valid components. By checking for smaller valid words that could be joined together correctly, the algorithm identifies and corrects the errors. With this step, misheard or mistyped words are properly separated and reconstructed.

### **Error Group 4: Completely Different Words or Gibberish**

Sometimes, transcription errors result in words that are either completely different from the intended words or are nonsensical (gibberish). To address these, the system uses advanced models like BERT and FastText. BERT analyzes the entire sentence context to determine if the word fits semantically, while FastText helps to find semantically similar words when no valid match exists. The system first tokenizes the sentence, checks if the word exists in a lexicon, and if not, uses FastText to find alternatives. BERT further validates these alternatives by analyzing the context of the sentence. Finally, the system selects the word with the highest confidence score as the best correction.

### **Error Group 5: Foreign Words**

Foreign words, such as English words that are mistakenly transcribed, pose a challenge when transcribing Sinhala audio. This step detects such words and transliterates them into Sinhala. The system uses a language detection model to identify foreign words and a rule-based mapping algorithm to convert these words into their closest Sinhala equivalents. This involves matching each segment of the English word with corresponding Sinhala characters or syllables. For instance, an English word like “computer” might be transliterated to “කම්පියුටර්” in Sinhala.

### **Step 4: Punctuation and Sentence Structure Errors**

Punctuation errors, such as missing commas, full stops, or incorrect placement of question marks, are addressed in this step. Punctuation plays an important role in sentence clarity and readability. The system uses transformers like BERT, fine-tuned for punctuation prediction tasks, to insert the correct punctuation marks. It identifies errors by analyzing sentence structure and context to make sure that the right punctuation is placed at appropriate locations, such as at the end of sentences or between clauses. This step improves the overall clarity and readability of the transcribed text.



### **Step 5: Checking and Correcting Unmatching Consecutive Words in a Sentence**

In this step, the system checks whether consecutive words in a sentence form a valid and meaningful phrase. If the combination of two consecutive words doesn't match grammatically or semantically, it indicates an error that needs correction. The system uses a model that checks the probability of two words occurring together in a sentence. If the probability is low, it suggests that the words don't typically belong together and need to be replaced or reordered. This makes sure that sentences are grammatically correct and the meaning is preserved.

### **Step 6: Finding Long Sentences and Splitting Them into Understandable Smaller Sentences**

Long sentences can be difficult to process and understand, so this step focuses on detecting lengthy sentences and splitting them into shorter, more understandable segments. The system uses an algorithm to identify sentences that exceed a predefined length threshold. By analyzing punctuation marks and conjunctions within the sentence, the system identifies natural break points to split the sentence into smaller, more manageable parts. This improves readability and ensures that the text is easier for users to understand.

The steps in this module play an important role in refining and improving the transcription process. They make sure that the final output is as accurate, readable, and contextually appropriate as possible. The combination of speech recognition, error detection, and correction algorithms creates a reliable module for handling various transcription challenges.

## **5.3 Module 2: Extraction of Meaningful Instructions, Summarization and Conversion to Written Language**

This module aims to process Sinhala speech transcriptions and refine them to create a set of instructions. The system focuses on identifying key instructions and filtering out irrelevant information to improve the clarity and accuracy of the final performance guide. Below is the top-level design and the various modules that contribute to this process.

### **Description of Module**

As described in the previous chapter, this module focuses on extracting important instructions from the transcribed text and writing it in written sinhala style.

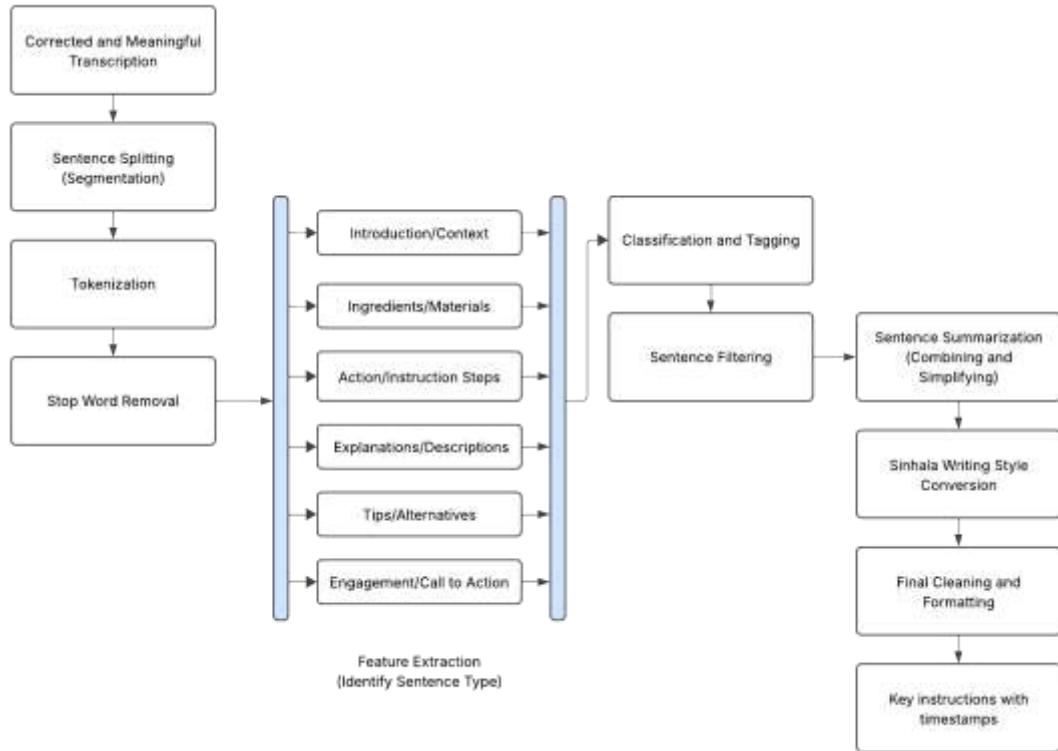


Figure 6 -Architectural Diagram-Module 2

### Step 1: Sentence Splitting (Segmentation)

In this initial step, the transcription is split into individual sentences to facilitate more focused analysis and processing. This can be accomplished by identifying punctuation marks, such as full stops or question marks, to segment the text. For the Sinhala language, custom punctuation rules or natural language processing libraries like NLTK or spaCy might be necessary, as they may not handle Sinhala-specific punctuation correctly. By dividing the text into sentences, each part becomes easier to handle and understand.

Output: A list of separate sentences.

Example Input: "ඔබ්න අද අපි වොකලට පුඩින් එකක් කරන්න යන්නේ. පළවෙනිම කෝන් ෆ්ලා ගන්න."

Example Output: ["ඔබ්න අද අපි වොකලට පුඩින් එකක් කරන්න යන්නේ", "පළවෙනිම කෝන් ෆ්ලා ගන්න"]

### Step 2: Tokenization (Breaking into Words)

Once the text is split into sentences, the next step involves breaking each sentence into individual words. Tokenization is essential for identifying key elements like action verbs or important terms that can help categorize the sentence's meaning.

Libraries such as NLTK can be used, but additional adjustments may be needed to handle Sinhala effectively.

Output: A list of words per sentence.

Example Input: "කෝන් ලො එකතු කරන්න"

Example Output: ["කෝන්", "ලො", "එකතු", "කරන්න"]

### **Step 3: Stop Word Removal**

Stop words, such as "අපි", "එක", and "හා", are common words that don't add significant meaning to a sentence and can clutter the analysis. To make the processing more efficient, these words should be removed. While some libraries, like NLTK, offer stop word lists, a custom stop word list for Sinhala would be more effective.

Output: Cleaner, more meaningful words.

Example Input: ["අපි", "කෝන්", "ලො", "එකතු", "කරන්න"]

Example Output: ["කෝන්", "ලො", "එකතු", "කරන්න"]

### **Step 4: Feature Extraction (Identify Sentence Type)**

In this step, the text is categorized into different types, such as greetings, ingredient lists, action steps, and more. Each type has specific characteristics, and recognizing these helps in determining which parts of the transcription are essential for the final guide. For example, sentences that offer greetings or introductions should be removed, while sentences with ingredients and instructions should be retained. Action steps and explanations should also be classified for further use.

Output: A classification of sentences into categories like INTRO, INGREDIENT, ACTION, DESCRIPTION, TIP, and ENGAGEMENT.

Algorithm Idea: Use keywords and phrases such as "අද මං", "කෝප්ප", "කරන්න", and "සබ්ජුයිබ්" to identify sentence categories.

### **Step 5: Classification and Tagging**

Each sentence is now tagged with a label based on its category. The classification identifies whether a sentence is an instruction (ACTION), an ingredient (INGREDIENT), or an introductory phrase (INTRO). Once labeled, the sentences are also tagged with a decision to either "keep" or "remove", depending on their relevance to the core instructions. This process helps prioritize key information.

Output: Tagged sentences with labels such as INTRO, INGREDIENT, ACTION, DESCRIPTION, TIP, and ENGAGEMENT.

## Step 6: Sentence Filtering

After the sentences are classified and tagged, the next task is to filter out non-essential sentences. Only sentences tagged as **INGREDIENT** and **ACTION** should be kept, as these are critical for the process. Sentences classified under **INTRO**, **DESCRIPTION**, **TIP**, or **ENGAGEMENT** are removed because they do not provide direct instructions or important information. This filtering makes sure that only actionable content remains for the final guide.

Output: A list of sentences that are essential to the process, with all unnecessary sentences removed.

## Step 7: Sentence Summarization (Combining and Simplifying)

In this step, related sentences are combined and simplified to create clear, concise instructions. Multiple sentences that describe the same action or ingredient are merged into a single, more straightforward statement. This helps avoid redundancy and ensures that the instruction is easy to follow. Quantities and measurements should be extracted and formatted into a single sentence for clarity.

Output: Simplified sentences that contain key instructions.

Example Input: "ගුම් 30", "මේස හැඳි තුනක්"

Example Output: "එයට කෝන් ෆ්ලා ගුම් 30 (මේස හැඳි තුනක්) සහ වොකලට පව්ඨර් ගුම් 20 (මේස හැඳි දෙකහමාරක්) එකතු කරගන්න"

## Step 8: Sinhala Writing Style Conversion

The spoken and written styles of Sinhala differ, with the written style being more formal and instructional. In this step, informal phrases used in spoken language, like "කරගන්න", are replaced with formal equivalents such as "කර ගත යුතුයි". Additionally, sentence structures are adjusted to fit the norms of written Sinhala.

Output: Revised sentences that follow the formal written style of Sinhala.

Example: "එයට කෝන් ෆ්ලා එකතු කරගන්න" becomes "එයට කෝන් ෆ්ලා එකතු කළ යුතුයි".

## Step 9: Final Cleaning and Formatting

The final step involves making sure the transcription follows consistent punctuation, spacing, and grammar rules. The text should have no extra spaces between words, and punctuation should be used properly. The sentence structure must also conform to Sinhala grammar norms, so that the instructions are clear and readable. This step improves the final output and makes sure it is professionally formatted.

Output: Clean, well-formatted text that follows proper punctuation and grammar rules.

## Summary

The module involves a multi-step process to transform a Sinhala transcription into clear, concise instructions for a performance guide. It includes sentence segmentation, tokenization, stop word removal, feature extraction, and classification to categorize and filter essential sentences. The final steps involve summarizing, converting to a formal writing style, and cleaning the text for consistency.

## 5.4 Module 3: Action/Object Detection and Image Selection Aligned with Instructions.

The module focuses on analyzing video demonstrations to extract meaningful actions, recognize the ingredients and tools used, generate structured instructions, and synchronize them with Sinhala transcriptions.

### Description of Module

As described in the previous chapter, this module focuses on finding the best image that corresponds to the extracted key instructions.

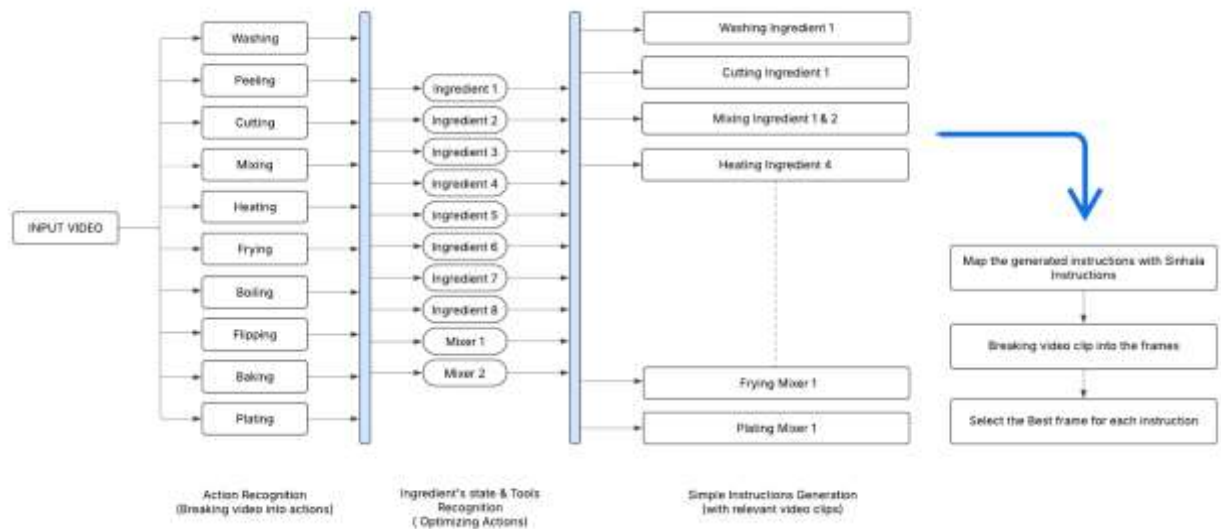


Figure 7-Architectural Diagram-Module 3

### **Step 1: Preprocessing**

In the preprocessing stage, the video is prepared for further analysis by improving its quality and extracting relevant data. The first task involves improving the video's clarity by removing noise and ensuring it is suitable for processing. This is followed by applying speech-to-text processing, where the audio content is transcribed into text to extract spoken instructions. The speech-to-text model helps isolate the spoken words that will guide the system in generating actionable instructions, forming the foundation for the subsequent stages.

### **Step 2: Action and Object Recognition**

Once the video is preprocessed, the system moves to the action and object recognition stage, where deep learning models are employed to detect cooking actions and objects. A model such as a Convolutional Neural Network or a Transformer-based architecture is trained to recognize various actions (e.g., chopping, stirring) being performed in the video. Simultaneously, object detection techniques like YOLO (You Only Look Once) or Faster R-CNN are used to identify ingredients and cooking tools present at each step. This stage's output is critical, as it supplies the action and object information necessary for constructing the final instructions.

### **Step 3: Instruction Generation**

In this stage, the recognized actions and objects are synthesized into structured instructions using Natural Language Processing techniques. The goal is to combine the action and object into a coherent instruction that is easy to follow. These instructions are generated in a simple and standardized format: Action + Object + (Optional Additional Details), for example, "Chop the carrot into small pieces." The system utilizes NLP algorithms to ensure that the instructions are clear, grammatically correct, and provide enough detail for the user to execute the cooking tasks.

### **Step 4: Sinhala Instruction Mapping**

The Sinhala instruction mapping process is aimed at ensuring the generated instructions are localized for Sinhala-speaking users. The system first checks whether a direct match exists between the generated instructions and pre-existing Sinhala translations using text-matching techniques. If an exact match is found, the corresponding Sinhala translation is assigned to the instruction. If no match is found, a machine translation model such as MarianMT or the Google Translate API is used to translate the instruction into Sinhala. This step ensures that all instructions are presented in the appropriate language, making the system accessible to a broader audience.

### **Step 5: Frame Extraction and Selection**

To improve the user experience, the video is broken down into frames at key timestamps that correspond to critical moments in the cooking process. Each frame is then evaluated by a ranking algorithm, which considers factors such as visual clarity, relevance, and alignment with the instruction. The goal is to identify the most visually informative frame for each step of the process. Once the ranking is completed, the best frame is selected and paired with the corresponding instruction. This step ensures that users receive a clear, visually supportive guide, allowing them to easily follow the cooking process by referencing both the textual instructions and the relevant video frame.

# Chapter 6

## Implementation

### 6.1 Chapter Overview

This chapter delves into the application of the technologies and models discussed in the previous chapter, providing a detailed overview of the models and their usage.

### 6.2 Datasets

We have prepared multiple datasets in order to train models required for different aspects of our project.

#### 1. Datasets for corrected text from transcribed text

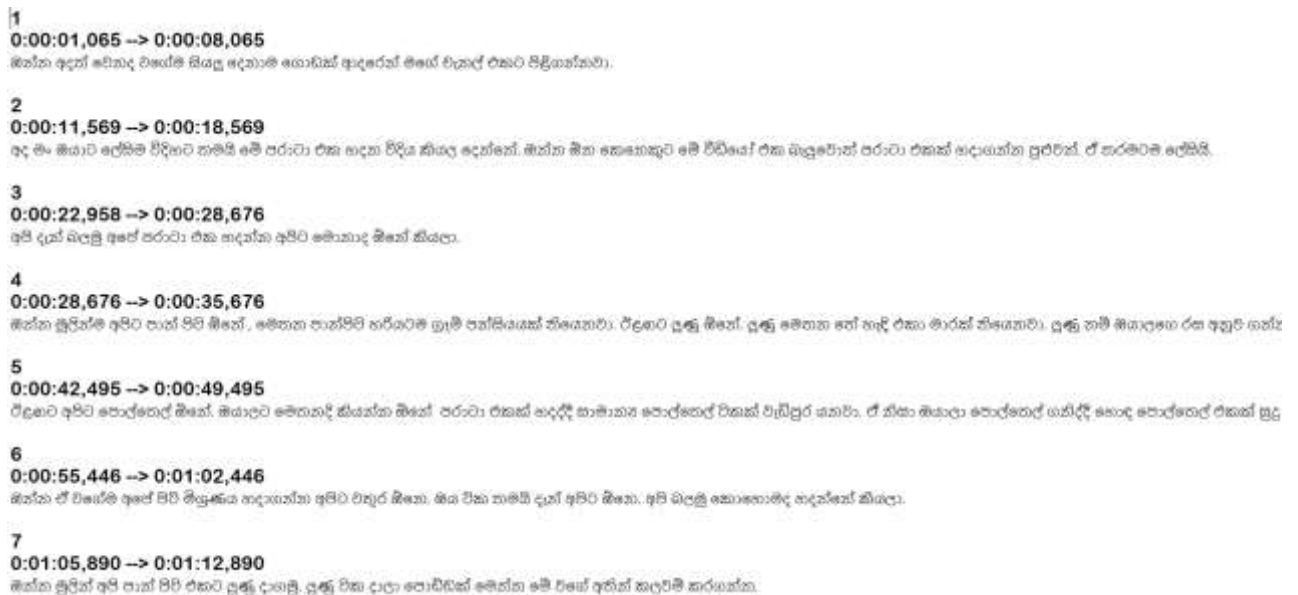


Figure 8-corrected transcribed text.



## 2. Datasets for key instructions

4  
0:00:28,676 --> 0:00:35,676  
පාත්පිටි ග්‍රාමී පන්සියයක් සහ ලුණු තේ හැඳි එකා මාරක් ගත යුතුයි.

5  
0:00:42,495 --> 0:00:49,495  
හොඳ පොල්තෙල් එකක් හෝ සුදු පොල්තෙල් එකක් ගත යුතුයි.

6  
0:00:55,446 --> 0:01:02,446  
පිටි මිශ්‍රණය සාදා ගැනීම සඳහා වතුර අවශ්‍ය වේ .

7  
0:01:05,890 --> 0:01:12,890  
පළමුව පාත් පිටි එකට ලුණු දමා අතින් කලවම් කළ යුතුයි.

8  
0:01:13,033 --> 0:01:20,033  
ඉන්පසු පොල්තෙල් මේස හත්දක් දමා කලවම් කළ යුතුයි.

Figure 9-extracted key instructions.

### 3. Datasets for Performance Guides

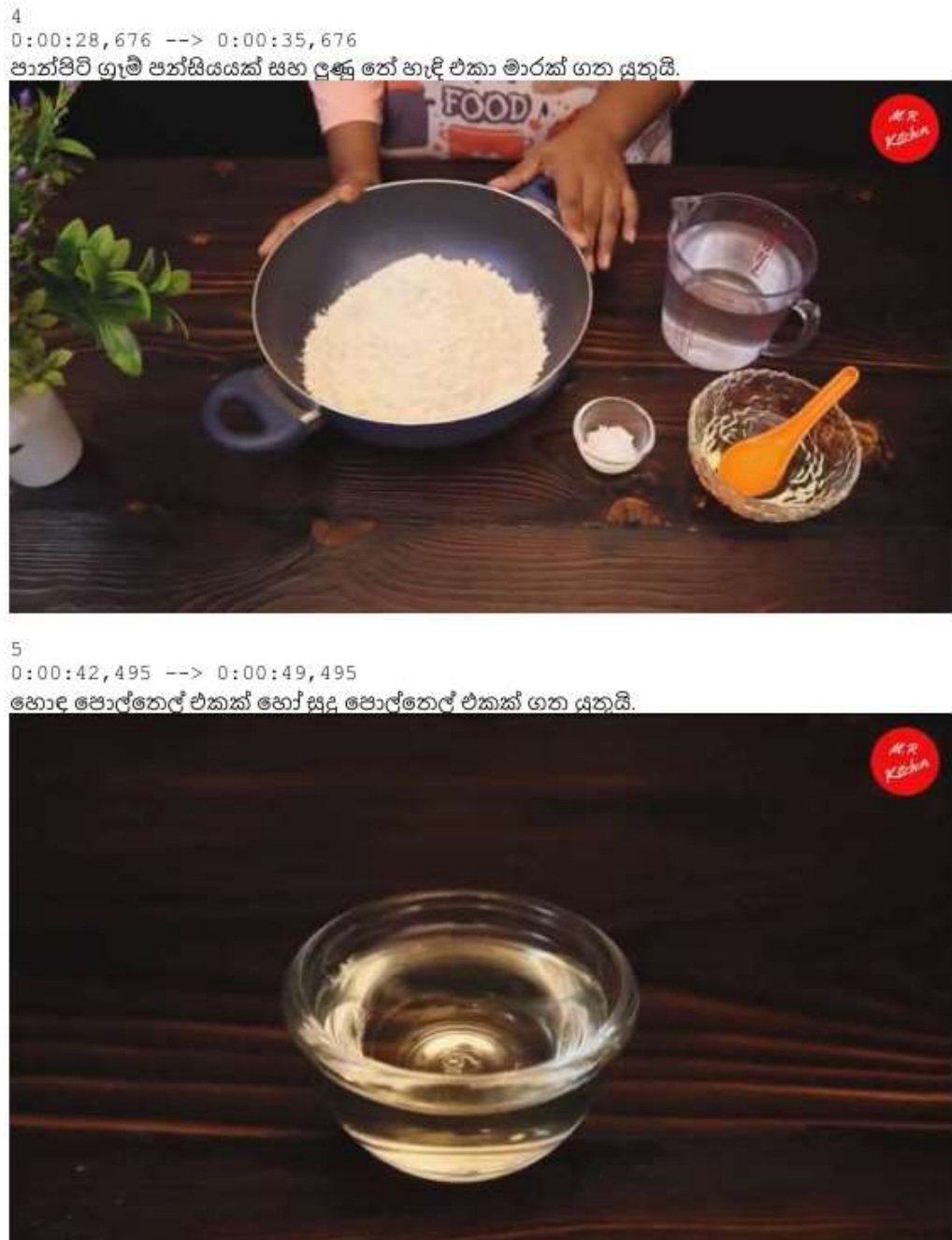
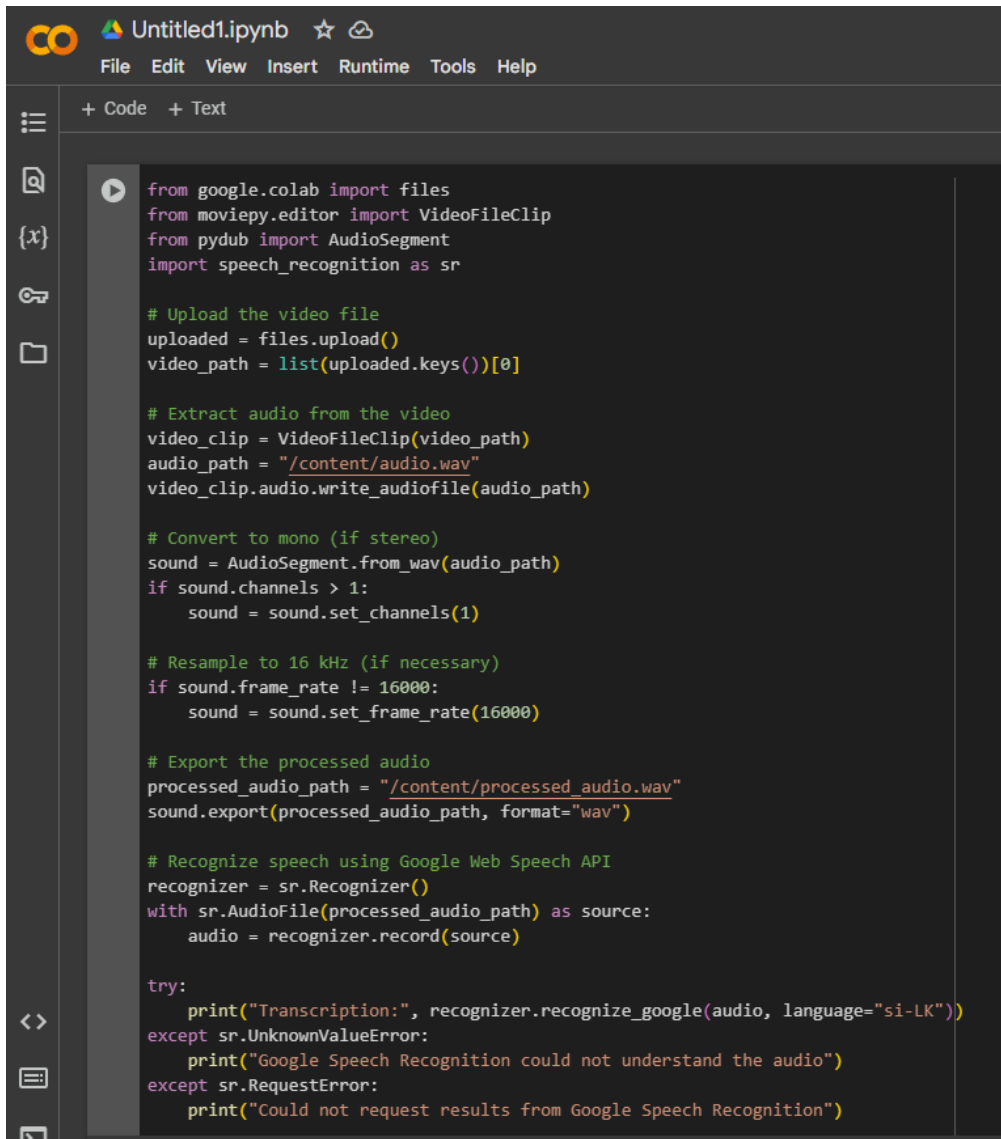


Figure 10-performance guide dataset.

## 6.3 Coding

We have used codes for different purposes throughout the project. Some examples are given below.

1. Code to extract audio from a video and transcribe it.

The image shows a Jupyter Notebook window titled 'Untitled1.ipynb'. The interface includes a top menu bar with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. Below the menu is a toolbar with icons for file operations and a '+ Code' button. The main area contains a single code cell with the following Python code:

```
from google.colab import files
from moviepy.editor import VideoFileClip
from pydub import AudioSegment
import speech_recognition as sr

# Upload the video file
uploaded = files.upload()
video_path = list(uploaded.keys())[0]

# Extract audio from the video
video_clip = VideoFileClip(video_path)
audio_path = "/content/audio.wav"
video_clip.audio.write_audiofile(audio_path)

# Convert to mono (if stereo)
sound = AudioSegment.from_wav(audio_path)
if sound.channels > 1:
    sound = sound.set_channels(1)

# Resample to 16 kHz (if necessary)
if sound.frame_rate != 16000:
    sound = sound.set_frame_rate(16000)

# Export the processed audio
processed_audio_path = "/content/processed_audio.wav"
sound.export(processed_audio_path, format="wav")

# Recognize speech using Google Web Speech API
recognizer = sr.Recognizer()
with sr.AudioFile(processed_audio_path) as source:
    audio = recognizer.record(source)

try:
    print("Transcription:", recognizer.recognize_google(audio, language="si-LK"))
except sr.UnknownValueError:
    print("Google Speech Recognition could not understand the audio")
except sr.RequestError:
    print("Could not request results from Google Speech Recognition")
```

Figure 11 -audio extraction and transcription code.

## 2. Code to extract frames from a video

```
1  import cv2
2  import os
3
4  video_path = "input/VID_001_action01.mp4"
5  output_folder = "frames"
6
7  # Create output directory if it doesn't exist
8  os.makedirs(output_folder, exist_ok=True)
9
10 # Open the video file
11 video = cv2.VideoCapture(video_path)
12 # Get total frame count and FPS
13 frame_count = int(video.get(cv2.CAP_PROP_FRAME_COUNT))
14 fps = video.get(cv2.CAP_PROP_FPS)
15 print(f"Video loaded: {video_path}")
16 print(f"Total Frames: {frame_count}, FPS: {fps}")
17
18 # Frame extraction loop
19 frame_index = 0
20 while True:
21     ret, frame = video.read()
22     if not ret: # Exit loop when no more frames
23         break
24     # Save frame as image
25     frame_filename = os.path.join(output_folder, f"frame_{frame_index:04d}.jpg")
26     cv2.imwrite(frame_filename, frame)
27
28     print(f"Saved: {frame_filename}")
29     frame_index += 1
30
31 video.release()
32 print(f"All frames saved to {output_folder}")
```

Figure 12-frame extraction code .

### Discussion

#### 7.1 Chapter Overview

In this chapter, we have provided a summary of each chapter along with how our solution differs from similar works.

#### 7.2 Module 1: Correction and Meaningful Refinement of Transcriptions to Enhance Clarity and Accuracy

In this module, we focused on increasing the accuracy and reliability of Sinhala speech-to-text transcriptions. The process involved multiple stages of error correction, addressing issues such as spelling mistakes, incorrect word forms, gibberish, foreign language usage, punctuation errors, and sentence structure problems. Key techniques, such as the edit distance algorithm for spelling errors, the Sinhala morphological analyzer for word forms, BERT for contextual correction, and FastText for semantic validation, were used to improve the transcriptions. The module provided a more accurate output by correcting these various types of transcription errors for better clarity and readability. This approach differs from similar works by incorporating a unique combination of NLP techniques, such as transliterating English words and also breaking down complex sentences into smaller sentences. It guarantees that the final transcriptions are not only correct but also contextually meaningful, making them suitable for further processing in the instructional guide.

#### 7.3 Module 2: Extraction of Meaningful Instructions, Summarization, and Conversion to Written

This module aims to create a set of instructions by extracting important content from Sinhala video transcriptions. It utilized a series of NLP techniques, including sentence segmentation, tokenization, and stop-word removal, followed by custom algorithms for classifying sentences into categories like ingredients, actions, and tips. After filtering out irrelevant content such as greetings or tips, the system summarized the necessary action steps and ingredients into concise instructions. These instructions are then transformed from informal spoken Sinhala into a more formal written format suitable for instructional content. This module's innovation lies in its ability to analyze transcribed text and create instruction sets while following the linguistic structure of formal Sinhala, which has not been widely explored in existing solutions.

#### **7.4 Module 3: Action/Object Detection and Image Selection Aligned with Instructions This module**

In this module, the proposed system streamlines the process of extracting structured cooking instructions from video demonstrations by focusing on action and object recognition, instruction generation, Sinhala instruction mapping, and frame selection. By utilizing deep learning techniques, the system effectively identifies key cooking actions (e.g., cutting, mixing, frying) and associated objects, ensuring accurate step-by-step instruction generation. Mapping these generated instructions with Sinhala transcriptions enhances linguistic consistency, making the content more accessible to Sinhala-speaking users. The selection of the most relevant video frames further improves comprehension, providing users with a clear visual reference for each step. However, challenges such as variations in video angles, overlapping objects, and ambiguous actions may affect accuracy. Future improvements could involve refining object recognition models and incorporating contextual understanding to enhance instruction quality. Overall, this approach ensures an efficient transformation of unstructured video content into structured and visually supported instructional guides.

## References

- [1] “PR6941-48.pdf.” Accessed: Feb. 23, 2025. [Online]. Available: <http://viduketha.nsf.gov.lk:8585/slsipr/PR6941/PR6941-48.pdf>
- [2] “(PDF) A Review on Image & Video Processing,” *ResearchGate*, Oct. 2024, Accessed: Feb. 23, 2025. [Online]. Available: [https://www.researchgate.net/publication/228612963\\_A\\_Review\\_on\\_Image\\_Video\\_Processing](https://www.researchgate.net/publication/228612963_A_Review_on_Image_Video_Processing)
- [3] “(PDF) Automated Text Summarization of Sinhala Online Articles.” Accessed: Feb. 23, 2025. [Online]. Available: [https://www.researchgate.net/publication/372952200\\_Automated\\_Text\\_Summarization\\_of\\_Sinhala\\_Online\\_Articles](https://www.researchgate.net/publication/372952200_Automated_Text_Summarization_of_Sinhala_Online_Articles)
- [4] “(PDF) Grammar Error Correction for Less Resourceful Languages: A Case Study of Sinhala,” in *ResearchGate*, doi: 10.1109/ICIIS58898.2023.10253578.
- [5] H. M. U. Pabasara and S. Jayalal, “Grammatical error detection and correction model for Sinhala language sentences,” in *2020 International Research Conference on Smart Computing and Systems Engineering (SCSE)*, Sep. 2020, pp. 17–24. doi: 10.1109/SCSE49731.2020.9313051.
- [6] A. Wasala, R. Weerasinghe, and K. Gamage, “Sinhala Grapheme-to-Phoneme Conversion and Rules for Schwa Epenthesis,” in *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, Sydney, Australia: Association for Computational Linguistics, Jul. 2006, pp. 890–897. Accessed: Feb. 23, 2025. [Online]. Available: <https://aclanthology.org/P06-2114/>
- [7] “(PDF) A Data-Driven Approach to Checking and Correcting Spelling Errors in Sinhala,” *ResearchGate*, Oct. 2024, Accessed: Feb. 23, 2025. [Online]. Available: [https://www.researchgate.net/publication/235931937\\_A\\_Data-Driven\\_Approach\\_to\\_Checking\\_and\\_Correcting\\_Spelling\\_Errors\\_in\\_Sinhala](https://www.researchgate.net/publication/235931937_A_Data-Driven_Approach_to_Checking_and_Correcting_Spelling_Errors_in_Sinhala)
- [8] “(PDF) Sinhala Spell Correction - A Novel Benchmark with Neural Spell Correction,” *ResearchGate*. Accessed: Feb. 23, 2025. [Online]. Available: [https://www.researchgate.net/publication/354477123\\_Sinhala\\_Spell\\_Correction\\_-\\_A\\_Novel\\_Benchmark\\_with\\_Neural\\_Spell\\_Correction](https://www.researchgate.net/publication/354477123_Sinhala_Spell_Correction_-_A_Novel_Benchmark_with_Neural_Spell_Correction)
- [9] X.-Y. Fu, C. Chen, M. T. R. Laskar, S. Bhushan, and S. Corston-Oliver, “Improving Punctuation Restoration for Speech Transcripts via External Data,” in *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, W. Xu, A. Ritter, T. Baldwin, and A. Rahimi, Eds., Online: Association for Computational Linguistics, Nov. 2021, pp. 168–174. doi: 10.18653/v1/2021.wnut-1.19.
- [10] “(PDF) SINHALA SPEECH RECOGNITION SYSTEM FOR JOURNALISTS IN SRILANKA,” in *ResearchGate*, Accessed: Feb. 23, 2025. [Online]. Available: [https://www.researchgate.net/publication/346624775\\_SINHALA\\_SPEECH\\_RECOGNITION\\_SYSTEM\\_FOR\\_JOURNALISTS\\_IN\\_SRILANKA](https://www.researchgate.net/publication/346624775_SINHALA_SPEECH_RECOGNITION_SYSTEM_FOR_JOURNALISTS_IN_SRILANKA)
- [11] “(PDF) SinMorphy: A Morphological Analyzer for the Sinhala Language,” in *ResearchGate*, Sep. 2024. doi: 10.1109/MERCon52712.2021.9525636.
- [12] S. Y. Senanayake, K. T. P. M. Kariyawasam, and P. S. Haddela, “Enhanced Tokenizer for Sinhala Language,” in *2019 National Information Technology Conference (NITC)*, Oct. 2019, pp. 84–89. doi: 10.1109/NITC48475.2019.9114420.
- [13] “Dynamic Stopword Removal for Sinhala Language | IEEE Conference

- Publication | IEEE Xplore.” Accessed: Feb. 23, 2025. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9114476>
- [14] “(PDF) A Word Sense Disambiguation Technique for Sinhala,” in *ResearchGate*, doi: 10.1109/ICAIET.2014.42.
  - [15] “SiTSE: Sinhala Text Simplification Dataset and Evaluation.” Accessed: Feb. 23, 2025. [Online]. Available: <https://arxiv.org/html/2412.01293v1>
  - [16] “Effectiveness of rule-based classifiers in Sinhala text categorization | IEEE Conference Publication | IEEE Xplore.” Accessed: Feb. 23, 2025. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8285655>
  - [17] S. Bianco *et al.*, “Cooking Action Recognition with iVAT: An Interactive Video Annotation Tool,” in *Image Analysis and Processing – ICIAP 2013*, A. Petrosino, Ed., Berlin, Heidelberg: Springer, 2013, pp. 631–641. doi: 10.1007/978-3-642-41184-7\_64.
  - [18] N. Aboubakr, R. Ronfard, and J. L. Crowley, “Recognition and Localization of Food in Cooking Videos,” in *Joint Workshop on Multimedia for Cooking and Eating Activities and Multimedia Assisted Dietary Management*, Stockholm, Sweden, Jul. 2018. doi: 10.1145/3230519.3230590.
  - [19] “(PDF) On the use of MKL for cooking action recognition,” *ResearchGate*, Oct. 2024, doi: 10.1117/12.2041939.
  - [20] R. Venkataramanan *et al.*, “Cook-Gen: Robust Generative Modeling of Cooking Actions from Recipes,” Jun. 01, 2023, *arXiv*: arXiv:2306.01805. doi: 10.48550/arXiv.2306.01805.
  - [21] A. B. Jelodar, M. S. Salekin, and Y. Sun, “Identifying Object States in Cooking-Related Images,” Oct. 30, 2018, *arXiv*: arXiv:1805.06956. doi: 10.48550/arXiv.1805.06956.
  - [22] A. M. Khan, A. Ashrafee, R. Sayera, S. Ivan, and S. Ahmed, “Rethinking Cooking State Recognition with Vision Transformers,” in *2022 25th International Conference on Computer and Information Technology (ICCIT)*, Dec. 2022, pp. 170–175. doi: 10.1109/ICCIT57492.2022.10055869.
  - [23] “Food Ingredients Identification from Dish Images by Deep Learning.” Accessed: Feb. 23, 2025. [Online]. Available: <https://www.scirp.org/journal/paperinformation?paperid=108663>
  - [24] “Cooktop Sensing Based on a YOLO Object Detection Algorithm - PMC.” Accessed: Feb. 23, 2025. [Online]. Available: <https://pmc.ncbi.nlm.nih.gov/articles/PMC10007026/>
  - [25] “Video Processing Using Deep Learning Techniques: A Systematic Literature Review | IEEE Journals & Magazine | IEEE Xplore.” Accessed: Feb. 23, 2025. [Online]. Available: <https://ieeexplore.ieee.org/document/9563948>
  - [26] Smart Engines Service LLC *et al.*, “Algorithm for choosing the best frame in a video stream in the task of identity document recognition,” *Computer Optics*, vol. 45, no. 1, Feb. 2021, doi: 10.18287/2412-6179-CO-811.



## ***Appendix: Individuals Contribution to the Project***

### **204104H**

#### **Module 1: Correction and Meaningful Refinement of Transcriptions to Enhance Clarity and Accuracy**

My primary responsibility was to refine transcriptions and correct errors to enhance clarity and accuracy. The transcriptions were obtained from audio or video recordings, and my task was to identify areas where the text was unclear, incorrect, or misleading. I applied my linguistic and technical knowledge to ensure that the transcriptions made sense contextually, while preserving the original meaning.

One of the key challenges I encountered was dealing with audio recordings that contained heavy accents, background noise, or multiple speakers talking over each other. These factors often made transcribing difficult, leading to inaccuracies in the initial transcriptions. To address this, I made changes to the transcription code to ensure that the final output was both accurate and easy to understand.

Through this process, I not only improved my skills in transcription and editing but also developed a deeper understanding of how nuances in spoken language can affect written content. I learned the importance of context when refining transcriptions and how to strike a balance between accuracy and readability. By the end of the project, I felt more confident in my ability to improve the quality of transcriptions, ensuring that they were not only accurate but also clear and accessible to a wide audience.

### **204137K**

#### **Module 2: Extraction of Meaningful Instructions, Summarization, and Conversion to Written Language**

For this module, my role was to extract meaningful instructions from raw content, summarize the key points, and convert them into clear, actionable written instructions. The content we worked with included a range of technical and instructional material, which required careful analysis to distill the essential information without losing important details.

The main challenge I faced was dealing with complex, jargon-heavy material that was difficult to understand. It was essential to ensure that the final written instructions were simple, concise, and user-friendly. To tackle this, I used a step-by-step approach, breaking down each instruction into digestible sections, removing unnecessary jargon, and rephrasing complex ideas into simpler language.

This module taught me the importance of clear communication, especially when translating complex information into easy-to-understand instructions. I learned how to identify the core elements of an instruction and how to present them in a way that could be easily followed by someone unfamiliar with the topic. I also gained valuable experience in summarization and content structuring, which will

be useful in my future work. The process not only improved my writing skills but also enhanced my ability to convey information efficiently and effectively.

### **204041K**

#### **Module 3: Frame Aligned Cooking Instructions via Action and Object Detection**

In this module, my role focused on aligning cooking instructions with action and object detection. The aim was to analyze video data to identify actions and objects in cooking instructions. Then I had to align these objects/actions with the extracted key instructions.

A significant challenge in this module was the accurate detection of objects and actions in the video content. The videos were sometimes poorly lit, and certain actions were hard to distinguish due to the camera angle or motion blur.

This module was a learning experience in combining video analysis with natural language processing. I gained a better understanding of how machine learning models can be used to identify specific actions and objects within dynamic video content. Additionally, I learned about the challenges of working with multimedia data, particularly how to address issues like poor video quality and ambiguous actions.