# ABSTRACT

Image to audio conversion technology is a transformative innovation that has made a significant impact on the lives of blind individuals, enabling them to access media and information that was previously inaccessible. This technology converts visual information into audio format, providing a means for the blind community to engage with various forms of media, art, and information. This article provides a comprehensive overview of image to audio conversion technology, discussing its history, applications, benefits, challenges, and future developments. We examine the importance of accessibility for the blind community and how image to audio conversion technology promotes independence, inclusion, and equal access to information. The applications of image to audio conversion technology are extensive, spanning across various fields such as education, media, art, navigation, healthcare, finance, and gaming. In education, this technology has transformed the learning experience for deaf students by providing access to visual aids in audio format. In the media, image to audio conversion technology has made films, television shows, and online content accessible to blind individuals. In the arts, this technology has opened new avenues for deaf individuals to appreciate and create visual art. Despite its many benefits, image to audio conversion technology also has its challenges and limitations, including accuracy, potential information loss during the conversion process, and implementation costs. However, future developments in this field are likely to focus on enhancing accuracy, speed, and accessibility through advancements in machine learning and artificial intelligence. Overall, image to audio conversion technology is a significant step forward in improving accessibility for the blind community and has the potential to transform the way we interact with various forms of media, art, and information.

## TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF ACRONYMS AND ABBREVIATIONS

# CHAPTER 1
# INTRODUCTION

## 1.1 Introduction

Visual impairment is one of the biggest limitations for humanity, especially in this day and age when information is communicated a lot by text messages (electronic and paper based) rather than voice. The device we have proposed aims to help people with visual impairment. In our planet of 7.4 billion humans, 285 million are visually impaired out of whom 39 million people are completely blind, i.e. have no vision at all, and 246 million have mild or severe visual impairment (WHO, 2011).

It has been predicted that by the year 2020, these numbers will rise to 75 million blind and 200 million people with visual impairment. As reading is of prime importance in the daily routine (text being present everywhere from newspapers, commercial products, sign boards, digital screens etc.) of mankind, visually impaired people face a lot of difficulties.

According to the development in today's technology towards the computer vision, digital camera and portable computers it is feasible to develop a camera-based technology that combines computer vision technology with other commercial products such as OCR systems. Reading is very essential in today's society. Everywhere the printed text is in the form of Reports, bank statements, receipts, restaurant menu's etc. so the blind users face a difficulty in reading these forms. In order to reduce the frustrated problem the method Text to Voice Adaption Using Portable Camera is referred. The method which is already existed a carries major drawback in size and not portable.

To reduce this drawback, we choose an embedded platform raspberry pi (Model 3) which acts as a mini Cohere the camera is interfaced to the raspberry pi board and the captured images is processed to the Rpi board. ROI method is used to localize and recognize the text. The text codes from the ROI are recognized by the Optical Character recognition (OCR) and the captured image is processed using the python programming language. The text from the OCR is compared with the text in the Open CV library to identifying the orientations and edge pixels. Therefore the captured images are converted in to the text. The text codes are processed to the pyttsx3 library and it is output to blind users in speech. And in addition, we add an ultrasonic sensor to alert the blind users by the speaker in avoiding the obstacles.

To extract the hand-held object from the camera image, this system going to develop a motion-based method to obtain a region of interest (ROI) of the object. Then, perform text recognition only that ROI. [2] detecting text in natural scenes with stroke width transform, to post process the image, support vector machines (SVM) had been proposed to do classification on the extracted features. Some kernel functions which are being tested are second degree polynomial, radial basis function (RBF), exponential radial basis function (ERBF), sigmoid, and odd-order Bsp line. RBF and ERBF.

To solve the common aiming problem we have implemented motion-based method to detect the objects of interest. Text extractions are done using stroke orientation and distribution of edge pixels. The text characters are recognized using Optical Character Recognition, the text codes are transformed as speech for blind persons. Our future work will extend the text localization algorithm with further more features and we will address the human interface issues associated with text reading by the blind user.

# CHAPTER 2
# LITERATURE REVIEWS

1. **Sneha.C. Madre, S.B. Gundre, "OCR Based Image Text to Speech Conversion Using MATLAB", Second International Conference on Intelligent Computing and Control Systems (ICICCS), 2019**

   The proposed system is inexpensive and allows visually impaired people to hear the text. This project's main concept is optical character recognition, which is used to convert text characters into audio signals. The text is preprocessed before being used for character recognition by segmenting each character. Following segmentation, the letter is extracted and the file containing the text is resized. The audio signal is then generated from the text file. All of the aforementioned processes will be carried out using MATLAB16.

2. **P Rohit, M S Vinay Prasad, S J Ranganatha Gowda, D R Krishna Raju, Imran Quadri, "Image Recognition Based Smart Aid For Visually Challenged People", International Conference on Communication and Electronics Systems (ICCES), 2020**

   Our paper describes the creation of a real-time system based on object detection, classification, and position estimation in an outdoor environment to provide visually impaired people with voice output-based scene perception. The system is inexpensive, lightweight, simple, and easy to wear. The module is built into the stick, and the pi-camera is used to take the photo, with a controller to move the camera in the desired direction. The useful insights obtained from the feedback are then used to modify the system to better suit the user's needs. For efficient feature representation, the object detection and classification framework employ a multi-modal fusion-based mask RCNN with motion, sharpening, and blurring filters. The image recognition classifies the detected objects as well as their positions.

3. **Sujata Deshmukh, Praditi Rede, Sheetal Sharma, Sahaana Iyer, "Voice-Enabled Vision For The Visually Disabled", International Conference on Advances in Computing, Communication, and Control (ICAC3), 2022**
   We proposed a unified system for extracting text from images and converting it into an audio track in the target language. This method can help the visually impaired sense the attitude and demeanor of the person they are dealing with. All the aforementioned tasks associated with this application are carried out by issuing commands. This system is unique in that it reads handwritten papers, analyses personality based on them, and provides audio output to the blind. This technology enables visually impaired people to read and comprehend First Information Reports (FIRs), business cards, chats, doctor's prescriptions, invoices, addresses, and other documents.

4. **Sai Aishwarya Edupuganti, Vijaya Durga Koganti, Cheekati Sri Lakshmi, Ravuri Naveen Kumar, Ramya Paruchuri, "Text and Speech Recognition for Visually Impaired People using Google Vision", 2nd International Conference on Smart Electronics and Communication (ICOSEC), 2021**
   This study assists visually impaired and elderly people in detecting text in order to identify the medicine. The researchers propose to create an application that will assist visually impaired people in scanning images and converting the detected text into voice messages. Google vision library is used to create an application for the Android platform that primarily contains three important functionalities: text recognition, text detection, and text-to-speech conversion. The medicine image is scanned using an in-built camera.

5. **Abhishek Mathur, Akshada Pathare, Prerna Sharma, Sujata Oak, "AI**

**based Reading System for Blind using OCR", 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), 2019**

OCR is a mechanism that converts typed, handwritten, or printed text images into machine encoded text. This system will assist you in taking a picture or scanning a document that is present with the user using the phone's camera. The image will be scanned, and the application will read the text written in English and convert the output to speech format. The Text to Speech Module is used to generate the speech output. The goal of delivering the output in the form of voice/speech is to serve the visually impaired with the information on the document.

6. **Vaibhav V. Mainkar, Tejashree U. Bagayatkar, Siddhesh K. Shetye, Hrushikesh R. Tamhankar, Rahul G. Jadhav, Rahul S. Tendolkar, "Raspberry pi based Intelligent Reader for Visually Impaired Persons", 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), 2020**

   In this project, the researchers use the Raspberry Pi Camera to take pictures, which are then converted into scan images for further processing using the Imagemagick software. The Imagemagick software produces a scanned image, which is then fed into the Tesseract OCR (Optical Character Recognition) software, which converts the image to text. TTS (Text to Speech) engine was used to convert text to speech. The experimental results show that analyzing different captured images will be more beneficial to blind people.

7. **Javavrinda Vrindavanam, Raghunandan Srinath, Anisa Fathima, S. Arpitha, Chaitanya S Rao, T. Kavya, "Machine Learning based approach to Image Description for the Visually Impaired", Asian Conference on Innovation in Technology (ASIANCON), 2021**

The need for the paper stems from the fact that the interaction points for visually impaired people are becoming increasingly limited in an increasingly digitized world, and accessing digital media through an image describer can be an accomplice for the visually impaired. Images that are unseen by the visually impaired are processed, appropriate descriptions are generated, and the audio output is converted. In contrast to traditional methods such as Computer Vision and Convolutional Neural Networks (CNN), the paper employs the Inception Resnet - V2 model as the feature extractor and decoder (GRU-RNN) along with the Bahdanau attention model to generate a text description of the image, which is then converted to audio using Google Text-to-Speech converter.

8. **R. Prabha, M. Razmah, G. Saritha, RM Asha, Senthil G. A, R. Gayathiri, "Vivoice - Reading Assistant for the Blind using OCR and TTS", International Conference on Computer Communication and Informatics (ICCCI), 2022**
   In this paper, the researchers intended to assist blind people in recognizing various texts and identifying various objects in their surroundings. The images are processed with OpenCV, which is written in Python and uses the Tesseract OCR library. The extracted texts are voiced using a Text-to-Speech synthesizer. The software used for text-to-speech conversion is eSpeak. The final output is delivered to visually impaired people via earphones. Another application is the use of Natural Language Processing Algorithms to search for the required product by providing it as an input to the device. The device looks for the product and alerts the visually impaired person to it. This device allows visually impaired people to reduce their reliance on other people and their other senses while still meeting their daily needs.

9. **S. Durgadevi, K. Thirupurasundari, C. Komathi, S.Mithun Balaji, "Smart Machine Learning System for Blind Assistance", International**

**Conference on Power, Energy, Control and Transmission Systems (ICPECTS), 2021**

The necessary data input is obtained using an image classification technique in order to access machine learning techniques. Using a camera, the objects in the vicinity of blind people are captured as images. It can precisely detect every object within a certain distance. The captured images are then converted into audio signals, which are then used to assist the blind. As a result, a user-friendly flexible guiding mechanism is created to assist blind people.

10. **Sandeep Musale, Vikram Ghiye, "Smart reader for visually impaired", 2nd International Conference on Inventive Systems and Control (ICISC), 2018**

    Typically, many people had visual impairments. Written transcripts are visible forms of information that are inaccessible to many blind and visually impaired people unless they are represented in a non-visual format such as Braille. A smart reader is required for an effective system for visually impaired people. MATLAB's OCR (Optical Character Recognition) functions convert images to text. The smart reader system for the visually impaired is proposed in this paper. A novel audio-tactile user interface that assists the user in reading information is proposed here.

# CHAPTER 3
# SYSTEM ANALYSIS

## 3.1 EXISTING SYSTEM

One such system is the Image Captioning system, which uses a combination of computer vision and NLP techniques to generate a textual description of an image. This textual description can then be converted into an audio format using text-to-speech synthesis techniques.

The Image Captioning system works by first analyzing the content of an image using computer vision algorithms to identify objects, scenes, and other visual features. This information is then combined with contextual knowledge and semantic rules to generate a textual description of the image. Natural Language Processing (NLP) techniques are used to analyze the textual description and to convert it into an audio format that can be easily understood by humans.

Another system is the Deep-Speaker Embeddings system, which uses deep learning techniques to convert an image into an audio representation. This system extracts features from the image using a deep convolutional neural network (CNN), and then uses an NLP algorithm to generate a corresponding audio signal. The resulting audio signal can be used to convey information about the content of the image, such as its color, texture, and shape.

Overall, there are many different approaches to converting images to audio using NLP algorithms, and the specific approach used will depend on the application and the specific requirements of the system.

There are several disadvantages to existing systems for image to audio conversion using NLP algorithms. Some of these disadvantages include:

**Accuracy:** Existing systems for image to audio conversion may not always produce accurate results. This is because the accuracy of the system depends on the quality of the image analysis and the NLP algorithms used. In some cases, the system may misinterpret the image, resulting in inaccurate audio descriptions.

**Limited vocabulary:** Existing systems for image to audio conversion may have a limited vocabulary, which can limit the types of descriptions that can be generated. This can make it difficult to accurately describe complex or abstract images.

**Difficulty with abstract concepts:** NLP algorithms may have difficulty with abstract concepts, such as emotions or ideas. This can make it challenging to accurately describe images that convey these types of concepts.

**Time-consuming:** Image to audio conversion using NLP algorithms can be time-consuming, particularly for large or complex images. This can make it challenging to use the system in real-time applications.

**Dependence on high-quality images:** The accuracy of the image analysis and NLP algorithms used in existing systems is dependent on the quality of the input image. Low-quality images may produce inaccurate results, which can limit the usefulness of the system.

Overall, while existing systems for image to audio conversion using NLP algorithms are useful, they still have several limitations that need to be addressed. Future research will be needed to improve the accuracy, speed, and flexibility of these systems.

## 3.2 PROPOSED SYSTEM

A proposed system for image to audio conversion using CNN-LSTM algorithm would involve the following steps:

Preprocessing: The system would begin by preprocessing the input image to extract relevant features. This would involve using a convolutional neural network (CNN) to extract visual features from the image.

**Encoding:** The visual features would then be encoded into a sequence of vectors using an LSTM (Long Short-Term Memory) network. The LSTM network would be trained to learn the relationships between the visual features and the corresponding audio descriptions.

**Decoding:** The encoded sequence of vectors would be passed through a decoder network, which would generate the corresponding audio signal. This would involve using a text-to-speech synthesis technique to convert the encoded sequence of vectors into an audio waveform.

**Postprocessing:** Finally, the resulting audio signal would be post-processed to improve its quality and clarity. This might involve techniques such as noise reduction, equalization, and amplification.

This proposed system has several advantages over existing systems that use NLP algorithms for image to audio conversion. First, the use of a CNN-LSTM algorithm allows the system to capture more complex relationships between the visual features and the corresponding audio descriptions, which can improve the accuracy and quality of the generated audio. Second, the use of a CNN-LSTM algorithm can also improve the speed and efficiency of the system, allowing it to be used in real-time applications. Finally, the proposed system is not limited by vocabulary or abstract concepts, as the audio descriptions are generated directly from the visual features of the input image.

## 3.3 FEASIBILITY STUDY

With an eye towards gauging the project's viability and improving server performance, a business proposal defining the project's primary goals and offering some preliminary cost estimates is offered here. Your proposed system's viability may be assessed once a comprehensive study has been performed. It is essential to have a thorough understanding of the core requirements of the system at hand before beginning the feasibility study. The feasibility research includes mostly three lines of thought:

- Economical feasibility

- Technical feasibility
- Operational feasibility
- Social feasibility

### 3.3.1 ECONOMICAL FEASIBILITY

The study's findings might help upper management estimate the potential cost savings from using this technology. The corporation can only devote so much resources to developing and analysing the system before running out of money. Every dollar spent must have a valid reason. As the bulk of the used technologies are open-source and free, the cost of the updated infrastructure came in far cheaper than anticipated. It was really crucial to only buy customizable products.

### 3.3.2 TECHNICAL FEASIBILITY

This research aims to establish the system's technical feasibility to ensure its smooth development. Adding additional systems shouldn't put too much pressure on the IT staff. Hence, the buyer will experience unnecessary anxiety. Due to the low likelihood of any adjustments being necessary during installation, it is critical that the system be as simple as possible in its design.

### 3.3.3 OPERATIONAL FEASIBILITY

An important aspect of our research is hearing from people who have actually used this technology. The procedure includes instructing the user on how to make optimal use of the resource at hand. The user shouldn't feel threatened by the system, but should instead see it as a necessary evil. Training and orienting new users has a direct impact on how quickly they adopt a system. Users need to have greater faith in the system before they can submit constructive feedback.

### 3.3.4 SOCIAL FEASIBILITY

During the social feasibility analysis, we look at how the project could change the community. This is done to gauge the level of public interest in the endeavor. Because of established cultural norms and institutional frameworks, it is likely that a certain kind of worker will be in low supply or nonexistent**.**

## 3.4 REQUIREMENT SPECIFICATION

### 3.4.1 HARDWARE REQUIREMENTS

| | |
|---|---|
| Processor | : Pentium Dual Core 2.00GHZ |
| Hard disk | : 120 GB |
| RAM | : 2GB (minimum) |
| Keyboard | : 110 keys enhanced |

### 3.4.2 SOFTWARE REQUIREMENTS

| | |
|---|---|
| Operating system | : Windows7 (with service pack 1), 8, 8.1 and 10 |
| Language | : Python |

## 3.5 LANGUAGE SPECIFICATION– PYTHON

Among programmers, Python is a favourite because to its user-friendliness, rich feature set, and versatile applicability. Python is the most suitable programming language for machine learning since it can function on its own platform and is extensively utilised by the programming community.

Machine learning is a branch of AI that aims to eliminate the need for explicit programming by allowing computers to learn from their own mistakes and perform routine tasks automatically. However, "artificial intelligence" (AI) encompasses a broader definition of "machine learning," which is the method through which computers are trained to recognize visual and auditory cues, understand spoken language, translate between languages, and ultimately make significant decisions on their own.

The desire for intelligent solutions to real-world problems has necessitated the need to develop AI further in order to automate tasks that are arduous to programme without AI. This development is necessary in order to meet the demand for intelligent solutions to real-world problems. Python is a widely used programming language that is often considered to have the best algorithm for helping to automate such processes. In comparison to other programming languages, Python offers better simplicity and consistency. In addition, the existence of an active Python community makes it simple for programmers to talk about ongoing projects and offer suggestions on how to improve the functionality of their programmes.

## ADVANTAGES OF USING PYTHON

Following are the advantages of using Python:

- **Variety of Framework and libraries:**

A good programming environment requires libraries and frameworks. Python frameworks and libraries simplify programme development. Developers can speed up complex project coding with prewritten code from a library. PyBrain, a modular machine learning toolkit in Python, provides easy-to-use algorithms. Python frameworks and libraries provide a structured and tested environment for the best coding solutions.

- **Reliability**

Most software developers seek simplicity and consistency in Python. Python code is concise and readable, simplifying presentation. Compared to other programming languages, developers can write code quickly. Developers can get community feedback to improve their product or app. Python is simpler than other programming languages, therefore beginners may learn it quickly. Experienced developers may focus on innovation and solving real-world problems with machine learning because they can easily design stable and trustworthy solutions.

- **Easily Executable**

Developers choose Python because it works on many platforms without change. Python runs unmodified on Windows, Linux, and macOS. Python is supported on all these platforms, therefore you don't need a Python expert to comprehend it. Python's great executability allows separate applications. Programming the app requires only Python. Developers benefit from this because some programming languages require others to complete the job. Python's portability cuts project execution time and effort.

# CHAPTER 4
# SYSTEM DESIGN

## 4.1 SYSTEM ARCHITECTURE

This graphic provides a concise and understandable description of all the entities currently integrated into the system. The diagram shows how the many actions and choices are linked together. You might say that the whole process and how it was carried out is a picture. The figure below shows the functional connections between various entities.

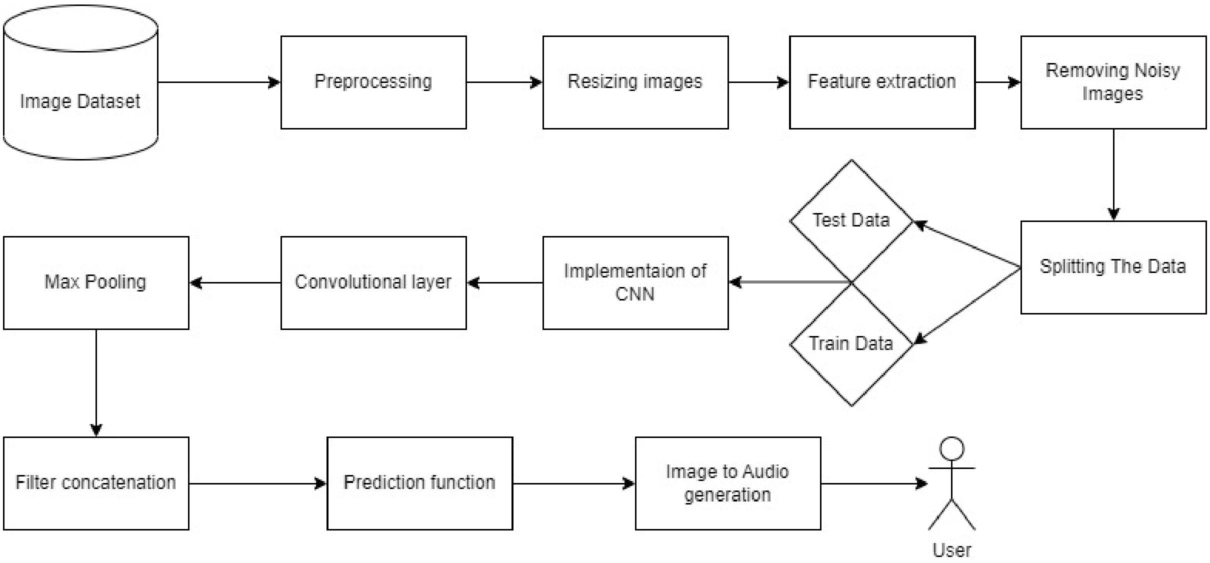


**Fig 4.1 – Architecture Diagram**

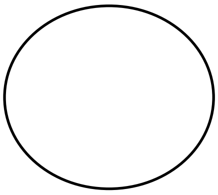
## 4.2 DATA FLOW DIAGRAM

To illustrate the movement of information throughout a procedure or system, one might use a Data-Flow Diagram (DFD). A data-flow diagram does not include any decision rules or loops, as the flow of information is entirely one-way. A flowchart can be used to illustrate the steps used to accomplish a certain data-driven task. Several different notations exist for representing data-flow graphs. Each data flow must have a process that acts as either the source or the target of the information exchange. Rather than utilizing a data-flow diagram, users of UML often substitute an activity diagram. In the realm of data-flow plans, site-oriented data-flow plans are a subset. Identical nodes in a data-flow diagram and a Petri net can be thought of as inverted counterparts since the semantics of data memory are represented by the locations in the network. Structured data modeling (DFM) includes processes, flows, storage, and terminators.

## Data Flow Diagram Symbols

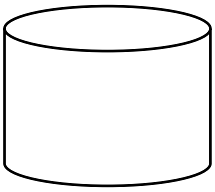
### Process

A process is one that takes in data as input and returns results as output.



### Data Store

In the context of a computer system, the term "data stores" is used to describe the various memory regions where data can be found. In other cases, "files" might stand in for data.



### Data Flow

Data flows are the pathways that information takes to get from one place to another. Please describe the nature of the data being conveyed by each arrow.

### External Entity

In this context, "external entity" refers to anything outside the system with which the system has some kind of interaction. These are the starting and finishing positions for inputs and outputs, respectively.

### DATA FLOW DIAGRAM

The whole system is shown as a single process in a level DFD. Each step in the system's assembly process, including all intermediate steps, are recorded here. The "basic system model" consists of this and 2-level data flow diagrams.

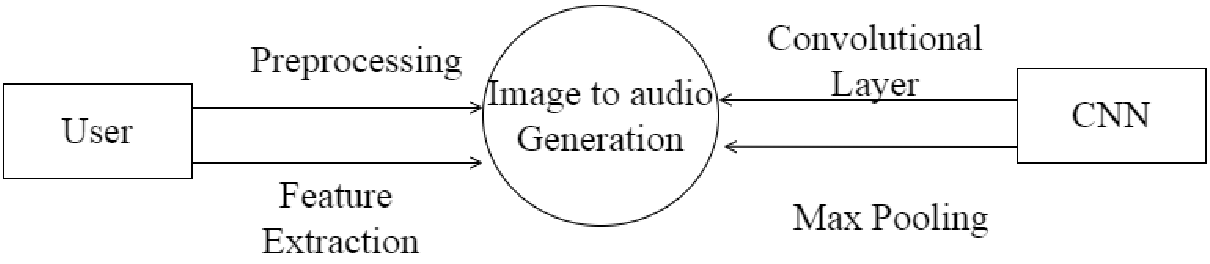


**Fig 4.2 – Data Flow Diagram Level 0**

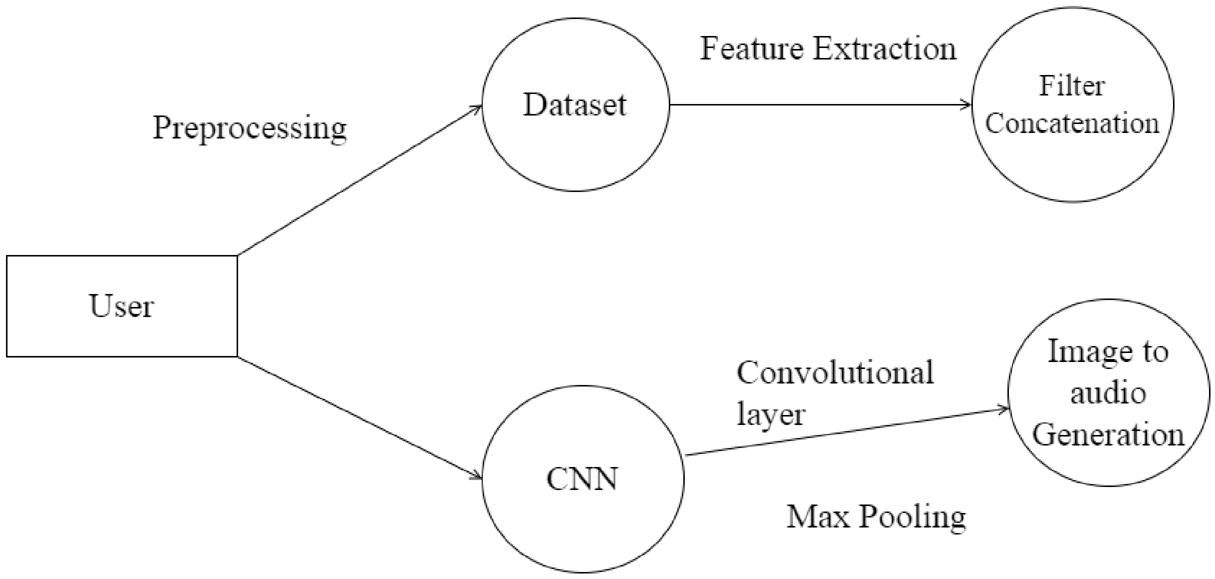


**Fig 4.3 – Data Flow Diagram Level 1**

## 4.3 ENTITY RELATIONSHIP DIAGRAM

- **Definition**

The relationships between database entities can be seen using an entity-relationship diagram (ERD). The entities and relationships depicted in an ERD can have further detail added to them via data object descriptions. In software engineering, conceptual and abstract data descriptions are represented via entity-relationship models (ERMs). Entity-relationship diagrams (ERDs), entity-relationship diagrams (ER), or simply entity diagrams are the terms used to describe the resulting visual representations of data structures that contain relationships between entities. As such, a data flow diagram can serve dual purposes. To demonstrate how data is transformed across the system. To provide an example of the procedures that affect the data flow.

**1. One-to-One**

Whenever there is an instance of entity (A), there is also an instance of entity (B) (B). In a sign-in database, for instance, only one security mobile number (S) is associated with each given customer name (A) (B).

**2. One-to-Many**

For each instance of entity B, there is exactly one occurrence of entry A, regardless of how many instances of entity B there are.

For a corporation whose employees all work in the same building, for instance, the name of the building (A) has numerous individual associations with employees (B), but each of these B's has only one individual link with entity A.

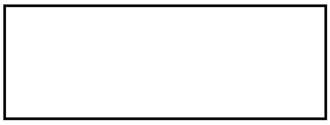
**3. Many-to-Many**

For each instance of entity B, there is exactly one occurrence of entry A, regardless of how many instances of entity B there are.

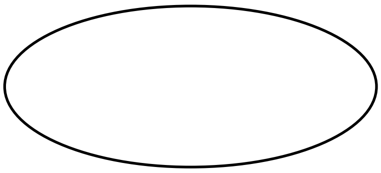
In a corporation where everyone works out of the same building, entity A is associated with many different Bs, but each B has only one A.

**SYMBOLS USED**

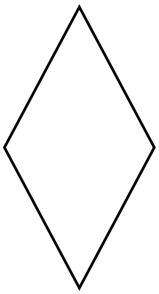
External Entity



Attribute



Relationship



Data Flow

**Fig 4.4 – Entity Relationship Diagram**

## 4.4 USE-CASE DIAGRAM

The possible interactions between the user, the dataset, and the algorithm are often depicted in a use case diagram. It's created at the start of the procedure.

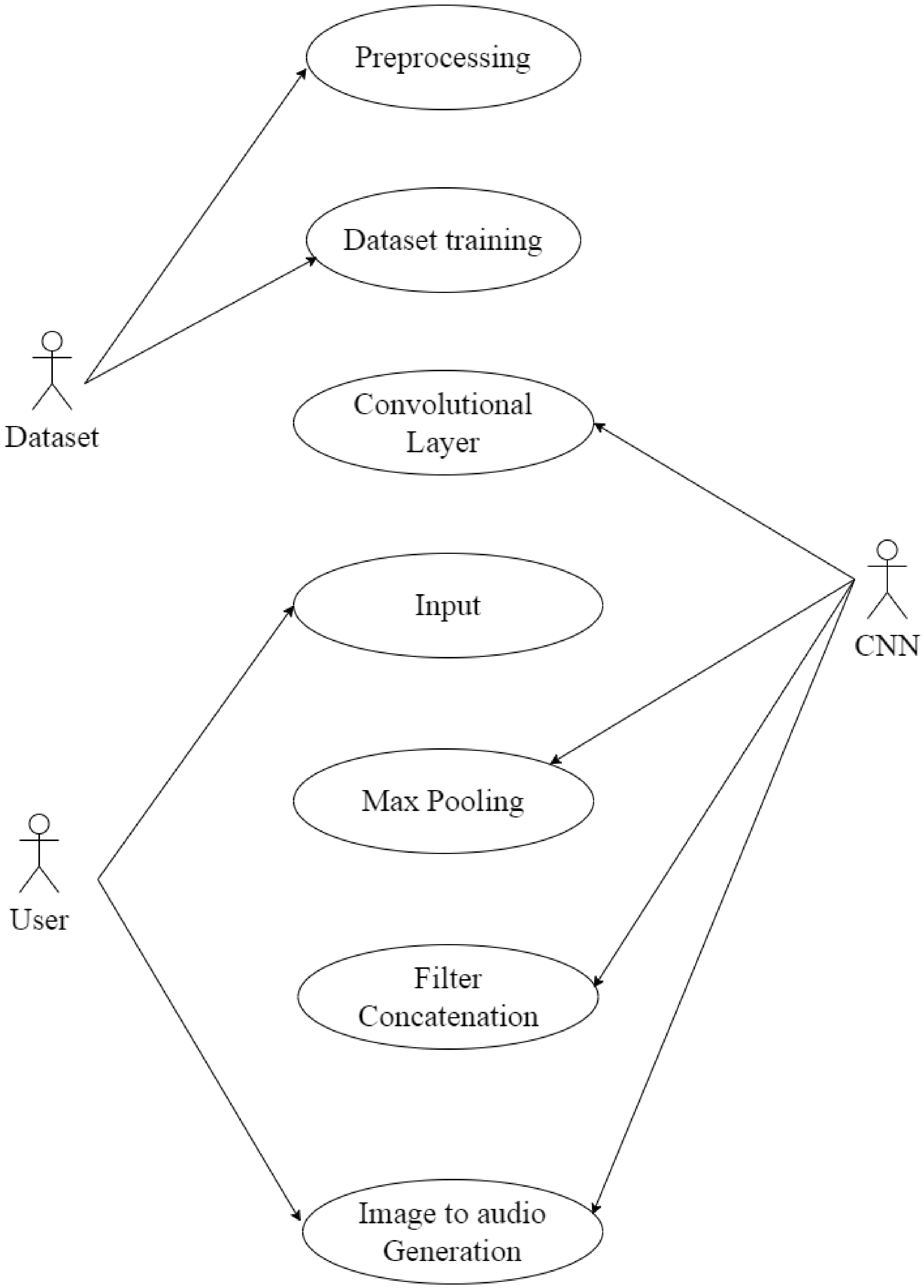


**Fig 4.5 – Use-Case Diagram**

## 4.5 ACTIVITY DIAGRAM

An activity diagram, in its most basic form, is a visual representation of the sequence in which tasks are performed. It depicts the sequence of operations that make up the overall procedure. They are not quite flowcharts, but they serve a comparable purpose.

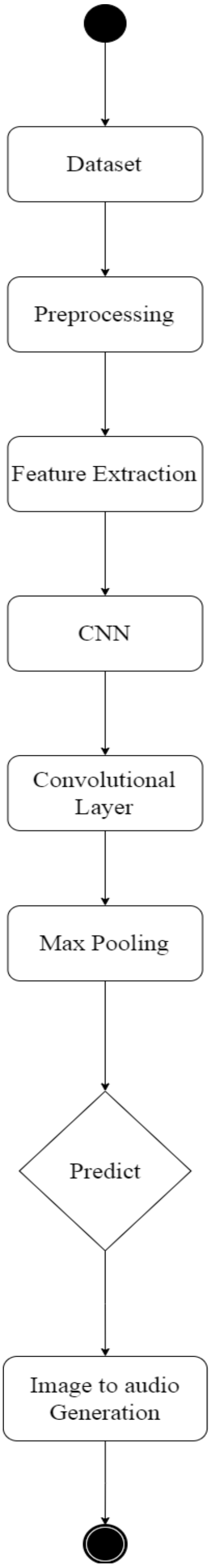


**Fig 4.6 – Activity Diagram**

## 4.6 SEQUENCE DIAGRAM

These are another type of interaction-based diagram used to display the workings of the system. They record the conditions under which objects and processes cooperate.

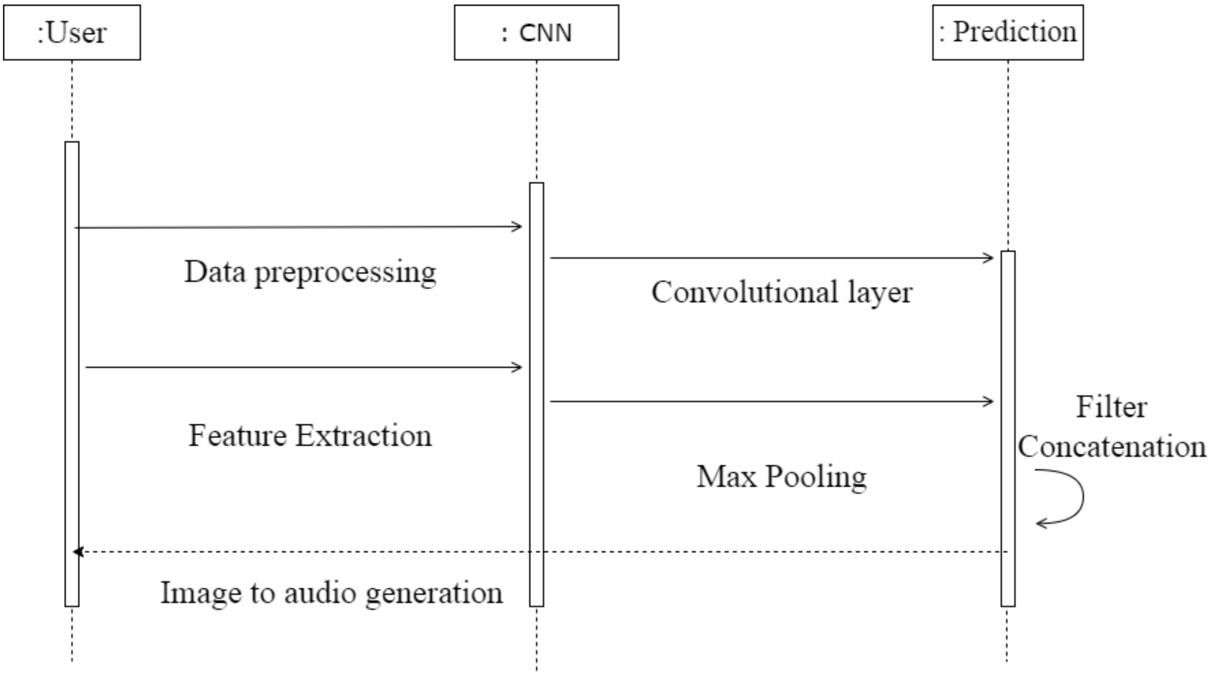


**Fig 4.7 – Sequence Diagram**

## 4.7 CLASS DIAGRAM

In essence, this is a "context diagram," another name for a contextual diagram. It simply stands for the very highest point, the 0 Level, of the procedure. As a whole, the system is shown as a single process, and the connection to externalities is shown in an abstract manner.

- A + indicates a publicly accessible characteristic or action.
- A - a privately accessible one.
- A # a protected one.
- A - denotes private attributes or operations.

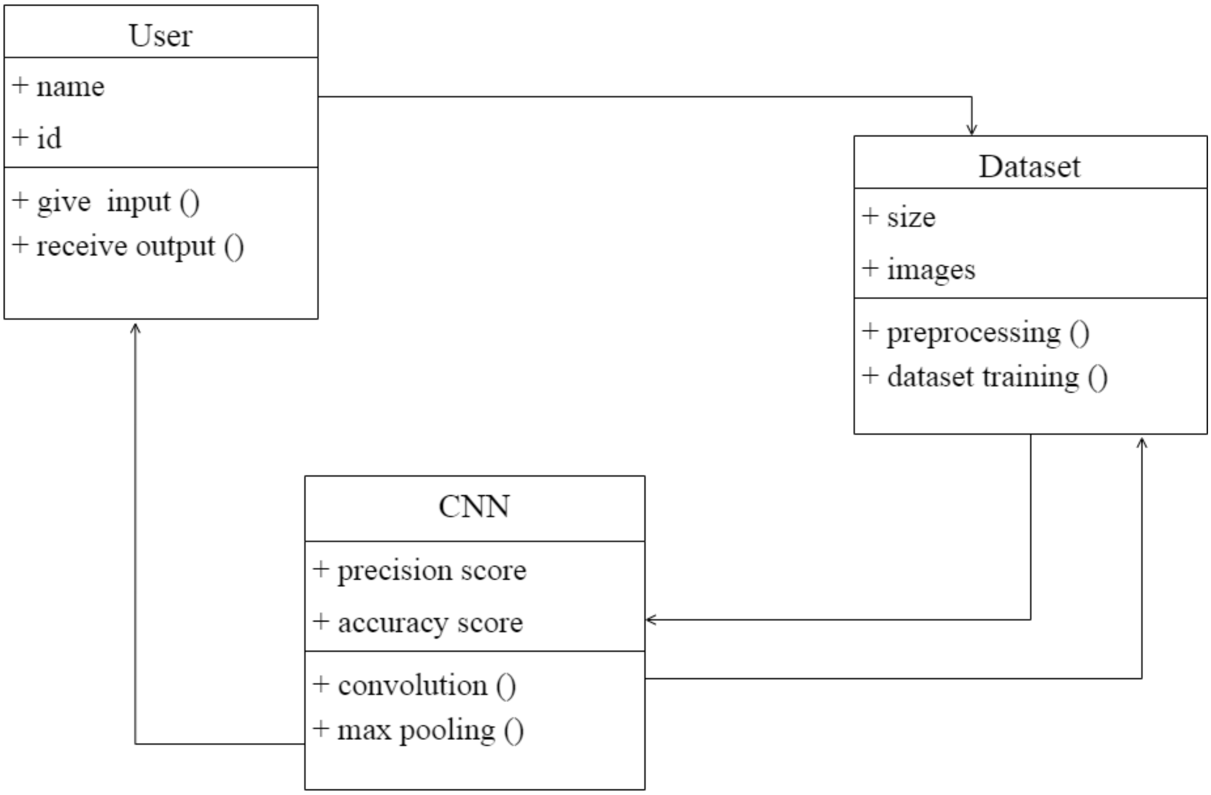


**Fig 4.8 – Class Diagram**

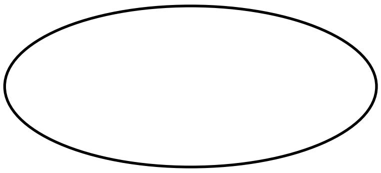
## 4.8 ER DIAGRAM

The abbreviation ER refers to a connection between two entities. The entities used and saved in the database are shown in relationship diagrams. They break down the process into its component parts and explain how they work. Attributed concepts, Relationship concepts, and Entity concepts are the building blocks for these kinds of diagrams.

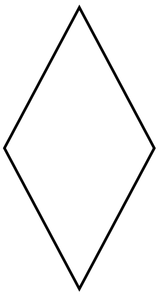
## SYMBOLS USED

External Entity

Attribute



Relationship



Data Flow

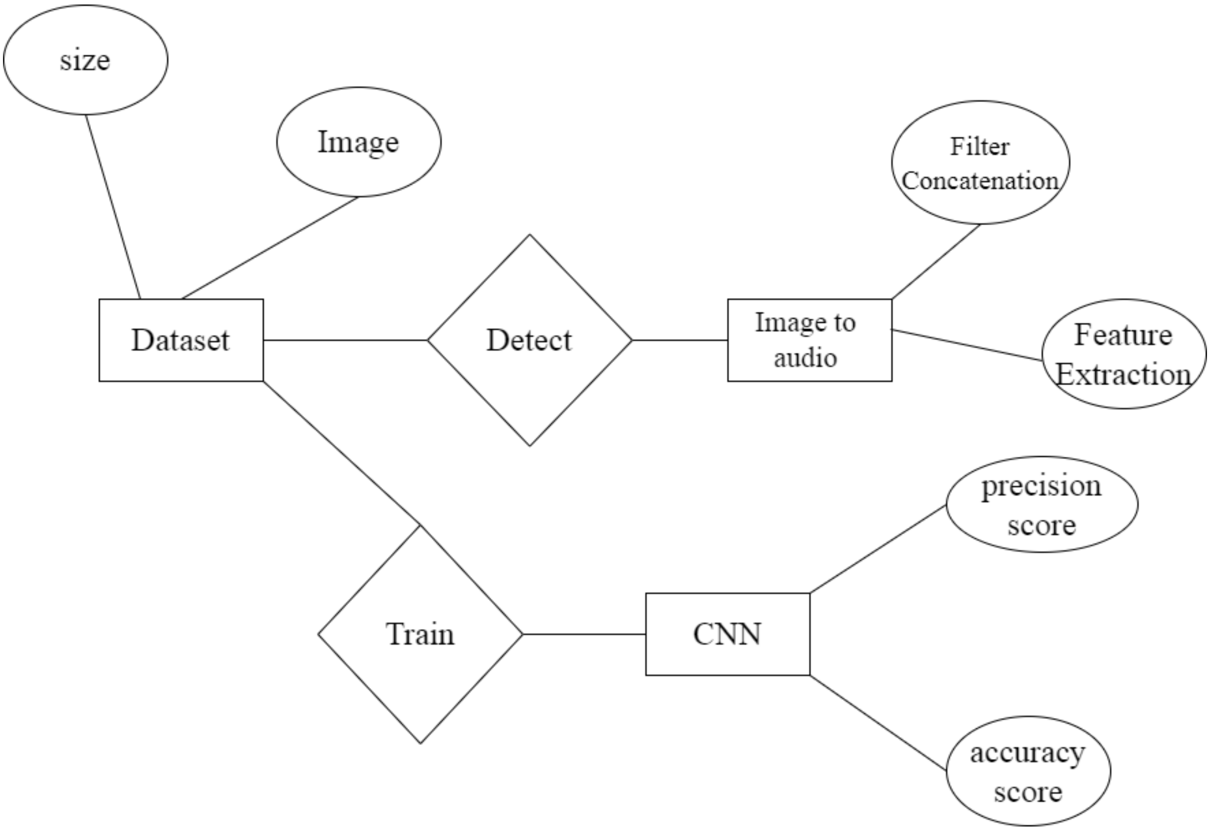


**Fig 4.9 – ER Diagram**

# CHAPTER 5
# MODULE DESCRIPTION

## 5.1 MODULE 1: IMAGE PRE-PROCESSING

The image pre-processing module is the first one in the CNN-LSTM image to audio conversion system. For the CNN-LSTM to be trained efficiently, this module is essential in getting the input picture data ready. A number of methods and algorithms are employed in the pre-processing module to improve the quality of the input photos, extract valuable characteristics, and normalise the pixel values to a set scale.

The pre-processing module's first responsibility is to resize the photos to a particular resolution. In order to properly train the CNN-LSTM, this step makes sure that all of the images are the same size. Standardizing the picture size will assist the CNN-LSTM in recognising patterns and features from the images. Images of varied sizes may present difficulties in the network training.

The pre-processing module's function of contrast enhancement is also crucial. Contrast enhancement techniques are used to make visual characteristics more visible and assist the CNN-LSTM in locating important features that can be used to generate audio. Histogram equalisation, adaptive histogram equalisation, and gamma correction are a few well-liked methods for enhancing contrast. These methods work by enhancing the contrast and altering the intensity values of the pixels in the image.

The images may also be subjected to noise reduction techniques to get rid of any extraneous or distracting data that could harm CNN-LSTM training. Gaussian blur, the median filter, and the bilateral filter are common approaches. These filters function by minimising the amount of noise in the image and smoothing it out.

Another crucial duty in the pre-processing module is feature extraction. In order to convert the image to audio, this method entails locating and extracting significant elements. Depending on the task at hand and the nature of the image data, several features may be extracted. Edge detection, texture analysis, and colour analysis are a few methods of feature extraction that are frequently utilised. In order to generate reliable audio, the CNN-LSTM needs to be able to recognise key patterns and characteristics in the visual data.

Another essential stage in the pre-processing module is normalisation. The process of normalisation involves converting the image's pixel values to a uniform scale. To ensure that the CNN-LSTM can be trained successfully, this step is required. Standardization, which adjusts the pixel values to have a mean of zero and a standard deviation of one, or min-max scaling, which scales the pixel values to a range between 0 and 1, are two examples of normalisation procedures.

In conclusion, the CNN-LSTM image to audio conversion system relies heavily on the image pre-processing module. The pre-processing programme improves the input photos' quality, extracts pertinent information, and scales the pixel values to a uniform scale. For the CNN-LSTM to learn and recognise significant patterns and characteristics in the image data that may be used for precise audio production, this module is crucial. The pre-processing module aids in the CNN-performance LSTM's optimization by giving the CNN-LSTM clean and pertinent data to train on.

## 5.2 MODULE 2: CNN-LSTM ARCHITECTURE

The CNN-LSTM architecture module is the second component of the image to audio conversion system using CNN-LSTM. In order for the CNN-LSTM to learn the mapping between the input images and associated audio signals, the structure and parameters of the CNN-LSTM must be defined by this module.

Determining the number of LSTM layers, the number of pooling layers, the size of the filters, the number of convolutional layers, and the number of filters per layer are all tasks included in the CNN-LSTM architecture module. Applying a series of filters to the input image allows convolutional layers to learn features from the previously processed image data. The filters are made to recognise particular motifs or characteristics in the visual data, such as edges, forms, or textures. The size of the filters and the number of filters per layer are hyperparameters that can be adjusted to enhance the CNN-performance. LSTM's In order to minimise the dimensionality of the data and avoid overfitting, pooling layers are employed to downsample the output of the convolutional layers. The maximum value within a local region of the input data is chosen by the max pooling layer, which is the most popular kind of pooling layer.

The output of the convolutional layers is processed by LSTM layers, while the audio signal is produced by LSTM layers. The LSTM layer architecture can describe the temporal dependencies between the input image and audio signals and is built to handle sequential data. The CNN-performance LSTM's can be improved by adjusting the number of LSTM layers and the number of neurons in each layer, which are both hyperparameters.

Tasks like regularisation, dropout, and activation functions might potentially be included in the CNN-LSTM architecture module. By including a penalty term in the loss function, regularisation approaches stop overfitting. In order to avoid overfitting, the CNN-LSTM is trained with a technique called dropout that randomly removes some of the neurons. In order to describe complicated interactions between the input and output data, activation functions are employed to bring non-linearity into the CNN-LSTM.

Using a set of tagged images and the matching audio labels, the CNN-LSTM is trained after the CNN-LSTM architecture has been defined. By modifying its biases and weights using an optimization approach like gradient descent, the CNN-LSTM learns to map the input image data to the audio labels.

In conclusion, a crucial part of the CNN-LSTM image to audio conversion system is the CNN-LSTM architectural module. The CNN-structure LSTM's and parameters are described in this module, along with how the CNN-LSTM learns to produce audio signals from the pre-processed image data. The CNN-LSTM architecture module enables the CNN-LSTM network to train effectively and understand the intricate mapping between the input visuals and related audio signals

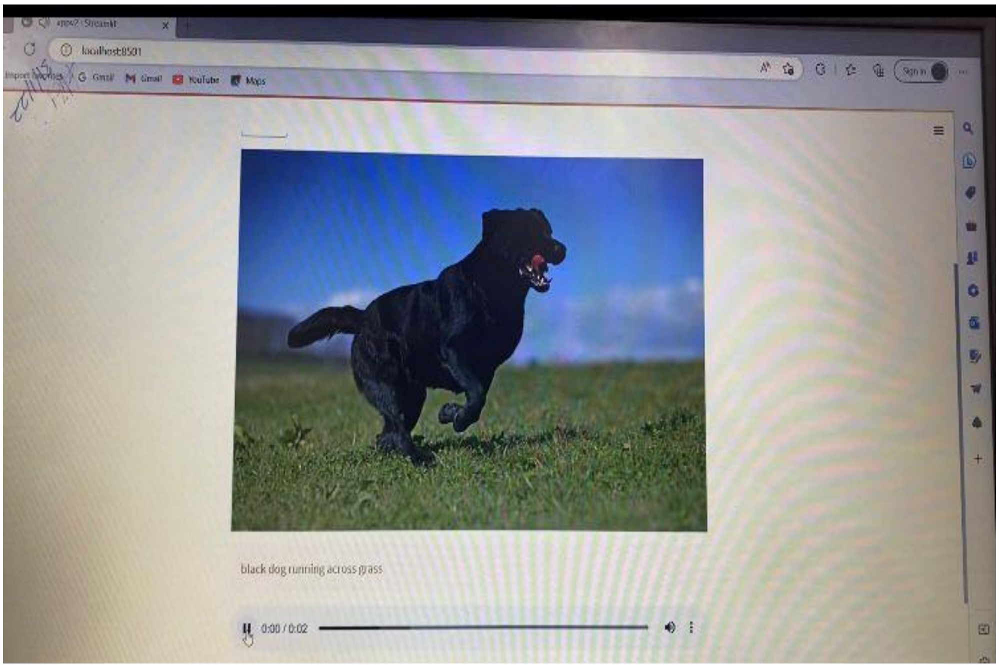
## 5.3 MODULE 3: POST PRE-PROCESSING

A series of feature vectors representing the likelihood of producing each audio sample is the CNN-output. LSTM's This series of feature vectors is examined by the post-processing module, which then creates the final audio signal.

The post-processing module may carry out operations like denoising, spectrogram creation, and inverse Fourier transform in order to do this. Denoising aids in removing any noise that might be present in the feature vector sequence. In order to effectively manipulate and process the audio input, the sequence of feature vectors is represented in the frequency domain using spectrogram creation. The final audio signal is created by converting a series of feature vectors from the frequency domain to the time domain using the inverse Fourier transform.

The post-processing module may also perform tasks including spectrum smoothing, dynamic range compression, and waveform normalisation. To make sure that the amplitude of the audio signal is within a particular range, waveform normalisation is performed. By modifying the audio signal's dynamic range, dynamic range compression can improve its clarity and lower distortion. To lessen any abrupt transitions or abnormalities that might be present in the audio signal, spectral smoothing is applied.

In general, the post-processing module is a crucial component of the CNN-LSTM image to audio conversion system. It examines the CNN-output LSTM's and changes it into a useful illustration of the audio signal. Several methods to improve the final audio signal's quality, such as denoising, spectrogram creation, and spectrum smoothing, may be included in the post-processing module. The post-processing module makes sure that the audio signal produced appropriately reflects the supplied visual data and is of excellent quality.



black dog running across grass

# CHAPTER 6
# TESTING

Discovering and fixing such problems is what testing is all about. The purpose of testing is to find and correct any problems with the final product. It's a method for evaluating the quality of the operation of anything from a whole product to a single component. The goal of stress testing software is to verify that it retains its original functionality under extreme circumstances. There are several different tests from which to pick. Many tests are available since there is such a vast range of assessment options.

**Who Performs the Testing:** All individuals who play an integral role in the software development process are responsible for performing the testing. Testing the software is the responsibility of a wide variety of specialists, including the End Users, Project Manager, Software Tester, and Software Developer.

**When it is recommended that testing begin**: Testing the software is the initial step in the process. begins with the phase of requirement collecting, also known as the Planning phase, and ends with the stage known as the Deployment phase. In the waterfall model, the phase of testing is where testing is explicitly arranged and carried out. Testing in the incremental model is carried out at the conclusion of each increment or iteration, and the entire application is examined in the final test.

**When it is appropriate to halt testing:** Testing the programme is an ongoing activity that will never end. Without first putting the software through its paces, it is impossible for anyone to guarantee that it is completely devoid of errors. Because the domain to which the input belongs is so expansive, we are unable to check every single input.

## 6.1 TYPES OF TESTING

There are four types of testing:

### Unit Testing

The term "unit testing" refers to a specific kind of software testing in which discrete elements of a program are investigated. The purpose of this testing is to ensure that the software operates as expected.

### Test Cases

1. Test if the software is able to correctly recognize individual sign language gestures in a given image.
2. Verify if the software is able to correctly classify each recognized gesture into appropriate categories.
3. Test if the software is able to correctly map each recognized gesture to appropriate audio output.
4. Verify if the software is using appropriate algorithms and techniques to perform image recognition and audio conversion tasks.

### Integration Testing

The programme is put through its paces in its final form, once all its parts have been combined, during the integration testing phase. At this phase, we look for places where interactions between components might cause problems.

### Test Cases

1. Test if the software is properly integrating with any image recognition software or libraries used for recognizing sign language gestures.
2. Verify if the software is properly integrating with any audio conversion software or libraries used for converting recognized gestures into audio output.

3. Test if the software is properly passing data between the different components of the application, such as image recognition and audio conversion.
4. Verify if the software is able to handle any conflicts or errors that may arise during the integration process.

**Functional Testing**

One kind of software testing is called functional testing, and it involves comparing the system to the functional requirements and specifications. In order to test functions, their input must first be provided, and then the output must be examined. Functional testing verifies that an application successfully satisfies all of its requirements in the correct manner. This particular kind of testing is not concerned with the manner in which processing takes place; rather, it focuses on the outcomes of processing. Therefore, it endeavours to carry out the test cases, compare the outcomes, and validate the correctness of the results.

**Test Cases**

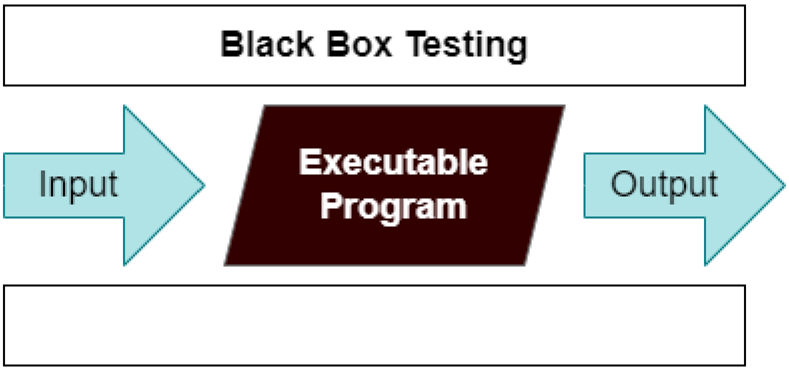
1. Verify if the software is able to detect the image of sign language and convert it into audio output.
2. Test if the software is able to recognize and process the images with various hand positions and facial expressions of the person signing.
3. Verify if the audio output is clear and understandable, without any distortion or noise.
4. Test if the software is able to differentiate between different signs and provide appropriate audio output.

## 6.2 TESTING TECHNIQUES

There are many different techniques or methods for testing the software, including the following:

## BLACK BOX TESTING

During this kind of testing, the user does not have access to or knowledge of the internal structure or specifics of the data item being tested. In this method, test cases are generated or designed only based on the input and output values, and prior knowledge of either the design or the code is not necessary. The testers are just conscious of knowing about what is thought to be able to do, but they do not know how it is able to do it.



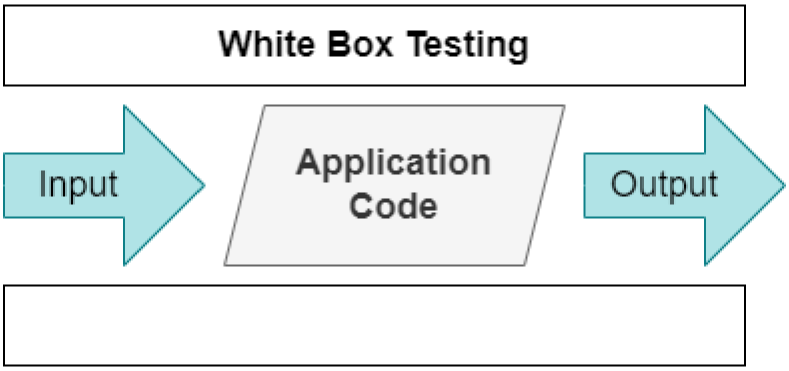
For example, without having any knowledge of the inner workings of the website, we test the web pages by using a browser, then we authorise the input, and last, we test and validate the outputs against the intended result.

**Test Cases**

1. Verify if the software is able to take an image as input and generate an audio output.
2. Test if the software is able to recognize different signs and gestures from the input image.
3. Verify if the software is able to provide the correct audio output for each sign and gesture.
4. Test if the software is able to recognize multiple signs in a single image and provide the correct audio output for each sign.

## WHITE BOX TESTING

During this kind of testing, the user is aware of the internal structure and details of the data item, or they have access to such information. In this process, test cases are constructed by referring to the code. Programming is extremely knowledgeable of the manner in which the application of knowledge is significant. White Box Testing is so called because, as we all know, in the tester's eyes it appears to be a white box, and on the inside, everyone can see clearly. This is how the testing got its name.



As an instance, a tester and a developer examine the code that is implemented in each field of a website, determine which inputs are acceptable and which are not, and then check the output to ensure it produces the desired result. In addition, the decision is reached by analyzing the code that is really used.

### Test Cases

1. Verify if the software is using appropriate algorithms and techniques to recognize sign language gestures in the input image.
2. Test if the software is using appropriate methods to convert the recognized gestures into audio output.
3. Verify if the software is handling exceptions and errors appropriately, such as when an input image cannot be recognized.
4. Test if the software is using appropriate data structures and algorithms to efficiently handle multiple input images.

# CHAPTER 7
# CONCLUSION AND FUTURE ENHANCEMENTS

## 7.1 CONCLUSION

In conclusion, image to audio conversion technology is an important tool that can significantly improve the lives of blind individuals by providing them with access to information and media that they would otherwise be unable to access. This technology has the potential to transform various fields such as education, media, art, navigation, healthcare, finance, and gaming, by providing equal access to all individuals, regardless of their hearing abilities. Despite the challenges and limitations, such as accuracy and implementation costs, image to audio conversion technology has enormous potential benefits. The technology is continuously evolving with advancements in machine learning and artificial intelligence, thus providing more accurate and efficient results. Image to audio conversion technology has the power to promote independence, inclusion, and equal access to information for the deaf community, leading to better educational, professional, and social outcomes. It has the potential to increase employment opportunities, improve communication, and reduce social isolation, among other benefits. Moreover, image to audio conversion technology has broader implications for society, as it promotes accessibility and inclusion for all individuals, regardless of their abilities. It is important to continue investing in the development of this technology to ensure its effectiveness and accessibility for all. In conclusion, image to audio conversion technology is a transformative innovation that has the potential to change the way we interact with visual content, and promote greater accessibility and inclusion for the deaf community.

## 7.2 FUTURE ENHANCEMENTS

Currently, only one image can be uploaded at a time; work is being done to allow users to upload multiple images at once. The audio generated is per page, so if you have multiple pages, you'll get many mp3 format, so depending on the order in which the image is uploaded, the audio files can be combined into a single.mp3, making it easier for users to download.

The program is currently limited to the English language; however, a user may upload an image in another language and the program will fail. Tesseract-OCR can recognize a variety of languages. As a result, in the future, it may be a goal to implement various languages in the project.

# CHAPTER 8

# APPENDIX 1

## CODING

```
import streamlit as st
from keras.models import load_model
import numpy as np
from keras.layers import Dense, LSTM, TimeDistributed, Embedding, Activation, RepeatVector,Concatenate
from keras.models import Sequential, Model
import cv2
from keras.preprocessing.sequence import pad_sequences
from tqdm import tqdm
from tensorflow.keras.applications.resnet50 import ResNet50
import pyttsx3
from PIL import Image
from gtts import gTTS
import re

#engine = pyttsx3.init()
st.header('Image to Audio Conversion')
vocab = np.load('vocab.npy', allow_pickle=True)

vocab = vocab.item()

inv_vocab = {v:k for k,v in vocab.items()}
```

```
print("+"*50)
print("vocabulary loaded")


embedding_size = 128
vocab_size = len(vocab)
max_len = 40


image_model = Sequential()

image_model.add(Dense(embedding_size, input_shape=(2048,),
activation='relu'))
image_model.add(RepeatVector(max_len))


language_model = Sequential()

language_model.add(Embedding(input_dim=vocab_size,
output_dim=embedding_size, input_length=max_len))
language_model.add(LSTM(256, return_sequences=True))
language_model.add(TimeDistributed(Dense(embedding_size)))


conca = Concatenate()([image_model.output, language_model.output])
x = LSTM(128, return_sequences=True)(conca)
x = LSTM(512, return_sequences=False)(x)
x = Dense(vocab_size)(x)
```

```
out = Activation('softmax')(x)
model = Model(inputs=[image_model.input, language_model.input], outputs = out)

model.compile(loss='categorical_crossentropy', optimizer='RMSprop', metrics=['accuracy'])

model.load_weights('mine_model_weights.h5')

print("="*150)
print("MODEL LOADED")

resnet = ResNet50(include_top=False,weights='imagenet',input_shape=(224,224,3),pooling='avg')


#resnet = load_model('model.h5')

print("="*150)
print("RESNET MODEL LOADED")

def main():
    uploaded_file = st.file_uploader('Upload a Image', type=['jpg', 'png'])
    if uploaded_file is not None:
        with open('./Images/input.png', 'wb') as f:
            f.write(uploaded_file.getvalue())

    if st.button('Detect'):
```

```
image = cv2.imread('./Images/input.png')
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

image = cv2.resize(image, (224,224))

image = np.reshape(image, (1,224,224,3))



incept = resnet.predict(image).reshape(1,2048)

print("="*50)
print("Predict Features")


text_in = ['startofseq']

final = ''

print("="*50)
print("GETING Captions")

count = 0
while tqdm(count < 20):

    count += 1

    encoded = []
    for i in text_in:
```

```
                encoded.append(vocab[i])

            padded = pad_sequences([encoded], maxlen=max_len, padding='post', truncating='post').reshape(1,max_len)

            sampled_index = np.argmax(model.predict([incept, padded]))

            sampled_word = inv_vocab[sampled_index]

            if sampled_word != 'endofseq':
                final = final + ' ' + sampled_word

            text_in.append(sampled_word)


        final_string = re.sub(r'[^\w\s]','', final)
        img = Image.open('./Images/input.png')
        st.image(img)
        st.warning(final_string)
        audio = gTTS(final_string, lang="en")
        audio.save('output.mp3')

        audio_file = open('output.mp3', 'rb')
        audio_bytes = audio_file.read()

        st.audio(audio_bytes, format='audio/mp3')
```

```
if __name__=='__main__':
    main()
```

# CHAPTER 10
# REFERENCES

[1] Sneha.C. Madre, S.B. Gundre, “OCR Based Image Text to Speech Conversion Using MATLAB”, Second International Conference on Intelligent Computing and Control Systems (ICICCS), 2019

[2] P Rohit, M S Vinay Prasad, S J Ranganatha Gowda, D R Krishna Raju, Imran Quadri, “Image Recognition Based Smart Aid For Visually Challenged People”, International Conference on Communication and Electronics Systems (ICCES), 2020

[3] Sujata Deshmukh, Praditi Rede, Sheetal Sharma, Sahaana Iyer, “Voice-Enabled Vision For The Visually Disabled”, International Conference on Advances in Computing, Communication, and Control (ICAC3), 2022

[4] Sai Aishwarya Edupuganti, Vijaya Durga Koganti, Cheekati Sri Lakshmi, Ravuri Naveen Kumar, Ramya Paruchuri, “Text and Speech Recognition for Visually Impaired People using Google Vision”, 2nd International Conference on Smart Electronics and Communication (ICOSEC), 2021

[5] Abhishek Mathur, Akshada Pathare, Prerna Sharma, Sujata Oak, “AI based Reading System for Blind using OCR”, 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), 2019

[6] Vaibhav V. Mainkar, Tejashree U. Bagayatkar, Siddhesh K. Shetye, Hrushikesh R. Tamhankar, Rahul G. Jadhav, Rahul S. Tendolkar, “Raspberry pi based Intelligent Reader for Visually Impaired Persons”, 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), 2020

[7] Javavrinda Vrindavanam, Raghunandan Srinath, Anisa Fathima, S. Arpitha, Chaitanya S Rao, T. Kavya, “Machine Learning based approach to Image Description for the Visually Impaired”, Asian Conference on Innovation in Technology (ASIANCON), 2021

[8] R. Prabha, M. Razmah, G. Saritha, RM Asha, Senthil G. A, R. Gayathiri, “Vivoice - Reading Assistant for the Blind using OCR and TTS”, International Conference on Computer Communication and Informatics (ICCCI), 2022

[9] S. Durgadevi, K. Thirupurasundari, C. Komathi, S.Mithun Balaji, “Smart Machine Learning System for Blind Assistance”, International Conference on Power, Energy, Control and Transmission Systems (ICPECTS), 2021

[10] Sandeep Musale, Vikram Ghiye, “Smart reader for visually impaired”, 2nd International Conference on Inventive Systems and Control (ICISC), 2018