A MACHINE LEARNING ASSISTED THERAPY TOOL FOR COMMUNICATION DISORDERED CHILDREN

A PROJECT REPORT

Submitted by

S. SNEHAA

17CSR192

B. SWATHI

17CSR209

V. VENKATESH

17CSR220

in partial fulfilment of the requirements for

the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMMUNICATION AND COMPUTER SCIENCES

Estd : 1984

KONGU ENGINEERING COLLEGE

(Autonomous)

PERUNDURAI ERODE – 638 060

DECEMBER 2020

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
KONGU ENGINEERING COLLEGE

(Autonomous)
PERUNDURAI ERODE – 638060
DECEMBER 2020

BONAFIDE CERTIFICATE

This is to certify that the Project Report entitled **A MACHINE LEARNING ASSISTED THERAPY TOOL FOR COMMUNICATION DISORDERED CHILDREN** is the bonafide record of project work done by **S.SNEHAA (Register no: 17CSR192), B.SWATHI (Register no: 17CSR209), V.VENKATESH (Register no: 17CSR220)** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Engineering in **Computer Science and Engineering** of Anna University, Chennai during the year 2020 - 2021.

SUPERVISOR **HEAD OF THE DEPARTMENT**
(Signature with seal)

Date:

Submitted for the end semester viva voice examination held on ___________________________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

KONGU ENGINEERING COLLEGE

(Autonomous)

PERUNDURAI ERODE – 638060

APRIL 2020

DECLARATION

We affirm that the Project report titled **A MACHINE LEARNING ASSISTED THERAPY TOOL FOR COMMUNICATION DISORDERED CHILDREN** being submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering is the original work carried by us. It has not formed the part of any other project or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

S.SNEHAA

(Reg.No.:17CSR192)

B.SWATHI

(Reg.No.:17CSR209)

V.VENKATESH

Date:

(Reg.No.:17CSR220)

I certify that the declaration made by the above candidates is true to the best of my knowledge.

Name & Signature of the Supervisor with seal

Date:

ABSTRACT

Communication is an important way of expressing yourself. But there are people who face Communication disorders. These days many children, from toddlers to teenagers are significantly affected by communicative disorders. There are mainly three types of communicative disorders. Namely language disorders, speech production disorders and oral motor/swallowing/feeding disorders. These problems being unidentified, and not being treated may cause problems for the children, like emotional and behavioral problems. There are tools and schools to treat these kind of disorders. Machine Learning techniques are proven to be effective for those diagnostic problems in today’s world. There is no tool which addresses the problem as a whole.

Using machine learning techniques we can help in treating the disorder. With the voice recording of the child, we can evaluate and monitor voice impairments in children by creating an automatic classification system which can determine the percentage of correctness in the child’s voice. Therapy tools available in the market focuses only on assisting the therapist but not to measure the improvement of the child. Hence, this proposal aims to include a novel and innovative ML based method in giving therapy for children having communicative disorders.

ACKNOWLEDGEMENT

Owing deeply to the supreme, we extend our thanks to the Almighty, who has blessed us to come out successfully with our project. We take pleasure to express our deep sense of gratitude to our beloved parents.

We express our sincere thanks and gratitude to our beloved correspondent **Thiru.P.Sachithanandan** for giving us the opportunity to pursue this course. We are extremely thankful with no words of formal nature to the dynamic Principal **Dr.V.Balusamy MTech, PhD** for providing the necessary facilities to complete our work.

We would like to express our sincere gratitude to our respected Head of the Department

Dr.N.Shanthi M.E., PhD., for providing necessary facilities.

We extend our thanks to **Dr.E.Gothai M.E., Ph.D** the project coordinator for her encouragement and valuable advice that made us to carry out the project work successfully.

We extend our gratitude to our supervisor **Dr.C.S.Kanimozhi Selvi M.E., Ph.D.,** for her valuable ideas and suggestions, which have been very helpful in the project. We are grateful to all the faculty members of the Computer Science and Engineering Department, for their support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

SSA	Sarva Shiksha Abhiyan
SLPs	Speech-Language Pathologists
DSM	Diagnostic and Statistical Manual of Mental Disorders
ICT	Information and Communication Technologies
CNN	Convolution Neural Network
SpLD	Specific Learning Disorder
ASD	Autism Spectrum Disorder
AAC	Augmentative and Alternative Communication
RIFF	Resource Interchange File Format

CHAPTER 1

INTRODUCTION

Communication is one's ability to express themselves to others in an effective way. Skillful communication is essential to succeed in all human activities. Communication disorders are most common problem seen in children with poor educational achievement. Individuals with communication disorders may unable to participate fully and competently in everyday interpersonal, learning, and occupational situations. Limited participation in those life activities due to their problem leads to associated emotional and behavioral problems.

These children are trained and cared in SSA schools. SSA is a programme for Universal Elementary Education. This programme is also an attempt to provide an opportunity for improving human capabilities to all children through provision of community owned quality education in a mission mode. In our country educational and vocational skills training to these persons become tough because of limited resources and funding. Though SSA schools are dedicated for the special children, they have limited resources and facilities. Hence it is very difficult to provide individual care to each and every child with special needs, and the children remains under sublimated care and training. Identification of voice disorders has a fundamental role in our life nowadays.

We can help parents and pathologists to evaluate and monitor voice impairments in children by creating an automatic classification system which can give therapy to the child's disorder. Many efforts have been devoted to improving the effectiveness of voice and emotion recognition. A computer based Education tool will alleviate the problems.

Nowadays, the use of mobile devices in the health-care sector is increasing significantly. Mobile technologies offer not only forms of communication for multimedia content (e.g. clinical audio-visual notes and medical records) but also provide promising solutions for people who desire the detection, monitoring and treatment of their health conditions anywhere and at any time. Mobile health systems can contribute to make patient care faster, better and cheaper. Several pathological conditions can benefit from the use of mobile technologies.

1.1. MACHINE LEARNING TECHNIQUES

Machine Learning is a concept which allows the machine to learn from examples and experience. Learning can classified as supervised and unsupervised. Supervised learning trains a model on known input and output audio data so that it can predict future outputs. Unsupervised learning finds hidden patterns or intrinsic structures in input data. Machine learning plays a vital role in health care, because of its ability to process a large number of audio datasets beyond human's capability and convert the analyzed data into clinical insights.

Classification algorithm is used for classify the autistic children, under which the given audio data fall upon, the classification algorithms that have been used are Multilayer perceptron (ANN) and Convolution Neural Network (DNN) is also used for predicting the training accuracy and testing accuracy. This figure 1.1 represents the techniques used in the machine learning .

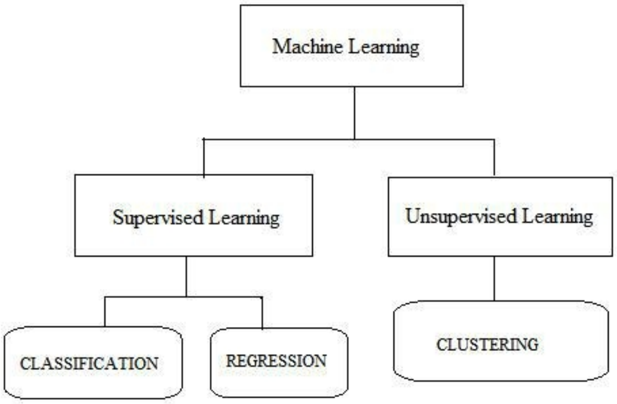


Figure 1.1. Machine learning techniques

1.1EXISTING SYSTEM

Speech therapy is the assessment and treatment of communication problems and speech disorders. It is performed by speech-language pathologists (SLPs), which are often referred to as speech therapists. Speech therapy techniques are used to improve communication. Therapists use a variety of tools and technologies to assist with evaluation, diagnosis and rehabilitation of individuals both young and old. These include articulation therapy, language intervention activities, and others depending on the type of speech or language disorder. Also, there are techniques like Voice Synthesizer, Analytical Software and Tablet Computer are the frequently used tools by the physicians.

1.2 OBJECTIVE

Therapy tools available in the market focuses only on assisting the therapist but not to measure the improvement of the child. Hence, this proposal aims to include a novel and innovative ML based application which will help in providing therapy for children having communicative disorders.

CHAPTER 2

LITERATURE REVIEW

Witsawakiti, et al. [2006] have described an e-training tool designed to help hearing impaired children learn and practice words in Thai language more correctly. The tool uses speech to overcome the limitations of the traditional face-to-face speech therapy and introduces background, main contents and knowledge structure of hearing impaired student's tutorials in pronunciation practice. Its professional cites benefits for hearing impaired students that increase the opportunity to prepare themselves for greater ability and competition in a globalization. The objective of this research was to study the effects of interactive multimedia tutorials for hearing impaired students in pronunciation practice.

Konstantinidis, et al. [2009] have evaluated the implementation of ACALPA platform uses an affective avatar, synthesized speech and multimedia content aiming at supporting and facilitating the teacher-child interaction. Enabling affective technologies are visited and a number of possible exploitation scenarios are illustrated. Emphasis is placed in covering the continuous and long term needs of autistic persons by unobtrusive and ubiquitous technologies with the engagement of an affective speaking avatar. A personalized prototype system facilitating these scenarios is described. In addition the feedback from educators for autistic persons is provided for the system in terms of its usefulness, efficiency and the envisaged reaction of the autistic persons, collected by means of an anonymous questionnaire. Results illustrate the clear potential of this effort in facilitating a very promising autism intervention.

Bastanfard, et al. [2010] have developed a software system which facilitates the interaction and synchronization of language learning activities for Persian hearing-impaired children. Shortcomings in the current process of speech therapy motivated us to develop a software system for Persian children with articulation disorders. The main aim of the system is to facilitate the interaction and synchronization of language learning activities conducted both at home and in speech therapy centers during the therapy process. A creative, entertaining combination of multimedia material is incorporated into the system. The methodology is practical and cost-effective and can be extended to other relatively uncommon languages.The preliminary test results show the successful integration of the designed system into the convenient method used by speech therapists in Iran.

Toki, E. I., &amp; Pange [2010] introduced an e-learning system for improving articulation in Greek preschoolers. Over the past decade speech and language therapy has taken an interesting turn towards the use of Information Communication Technologies (ICTs) for diagnosis of disorders and delivery of therapy. In many cases ICTs have worked as assistive tools to therapists, while in others as sole providers of therapy, especially in remote areas. In this work authors provide a brief overview of the most representative articles for applications and assistive technologies used for assessment and intervention purposes in Speech Therapy according to the type of disorders. The results of the study on this software showed that children not only improved their articulation but they also increased on language activities success by acquiring new vocabulary.

Amandeep Kaur and Jubilee Padmanabhan [2017] have highlighted the need and importance of early identification of the students with specific learning disorder, they also focus on the various tools and techniques for the screening of SpLD; national and international level programs and policies and school based interventions that can facilitate the learning. For better understanding of every aspect of SpLD is very essential for the teachers, as he/ she has the responsibility towards such students being specially able children and it is necessary to guide and train them in proper direction. While highlighting the need and importance of early identification of the students with specific learning disorder, this work will focus on the various tools and techniques for the screening of SpLD; national international level programs and policies and school based interventions that can facilitate the learning which can give wings to the dreams of such students.

Kanwajit Kaur1 & S. Pany [2017] have reviewed the computer based interventions which were used to improve social skills of autism spectrum disorder children. This work review addresses two systematic research questions: How the computer based intervention is used or developed and the effectiveness of computer based intervention for autism spectrum disorder children in improvement of social skills. Therefore, the specific objectives of this work are described as; to review the computer based interventions which were used to improve social skills of autism spectrum disorder children; and to analyse the findings of the previous work. The analysis of different studies revealed that computer based games are popularly used to improve the social skills of the ASD children and it is also observed that computer based interventions proved to be the useful interventions to improve the social skills of autism spectrum disorder children.

Sai Aishwarya Ramani and Amudhu Sankar [2017] focused on the development of a prototype “ISpeak” for Augmentative and Alternative Communication (AAC) which includes all forms of communication (other than speech) that are used to express thoughts, needs and ideas. People with severe speech or language difficulties rely on AAC to supplement existing speech. People with severe speech or language difficulties rely on AAC to supplement existing speech. This increases social interaction, school performance, and feeling of sel festeem. Studies in the past have mentioned greater results in the usage of AAC. This study focused on the development of a prototype AAC “ISpeak” which is cost efficient and can be easily maintained. This increases social interaction, school performance, and feeling of self esteem. Studies in the past have mentioned greater results in the usage of AAC.

CHAPTER 3

SYSTEM REQUIREMENTS

3.1HARDWARE REQUIRMENTS:

Processor	DELL
Processor Speed	1.80 GHz
Hard Disk	1 TB
RAM	4GB

Table 3.1 Hardware Requirements

3.2 SOFTWARE REQUIREMENTS:

Programming language	Python 3.5
Software	Anaconda Navigator 1.9.7 (Jupyter Notebook 6.0.3) Android Studio 3.5.3
Operating system	Window 10

Table 3.2 Software Requirements.

3.3 SOFTWARE DESCRIPTION

3.3.1 Python

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. Python source code is also available under the GNU General Public License (GPL). It provides constructs that enable clear programming on both small and large scales. It features a dynamic type system and automatic memory management, supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library. Python is open source software and has a community-based development model. Python and C Python are managed by the non-profit Python Software Foundation.

Following are the features of Python:

1. **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the developers to learn the language quickly.
2. **Easy-to-read** − Python code is more clearly defined and visible to eyes.
3. **Easy-to-maintain** − Python's source code is fairly easy to maintain.
4. **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
5. **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
6. **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
7. **Databases** − Python provides interfaces tall major commercial databases.

8. **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

9. **Scalable**− Python provides a better structure and support for large programs than shell scripting.

3.3.2 Anaconda Framework:

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. Package versions are managed by the package management system conda. Anaconda distribution is used by over 15 million users and includes more than 1500 popular data-science packages suitable for Windows, Linux, and MacOS.

3.3.2.1 Jupyter Notebook:

The Jupyter Notebook is an interactive computing environment that enables users to author notebook documents that include: - Live code - Interactive widgets - Plots - Narrative text - Equations - Images – Video. These documents provide a complete and self-contained record of a computation that can be converted to various formats and shared. The Jupyter Notebook combines three components:

1 **The notebook web application**: An interactive web application for writing and running code interactively and authoring notebook documents.

2 **Kernels**: Separate processes started by the notebook web application that runs users' code in a given language and returns output back to the notebook web application. The kernel also handles things like computations for interactive

widgets, tab completion and introspection.

3 **Notebook documents**: Self-contained documents that contain a representation of all content visible in the notebook web application, including inputs and outputs of the computations, narrative text, equations, images, and rich media representations of objects. Each notebook document has its own kernel.

3.3.2.2 TensorFlow

TensorFlow is an open source library for numerical computation and large-scale machine learning. TensorFlow bundles together a slew of machine learning and deep learning (aka neural networking) models and algorithms and makes them useful by way of a common metaphor. It uses Python to provide a convenient front-end API for building applications with the framework, while executing those applications in high-performance C++.

TensorFlow is the best library of all because it is built to be accessible for everyone. Tensorflow library incorporates different API to build at scale deep learning architecture like CNN or RNN. TensorFlow is based on graph computation; it allows the developer to visualize the construction of the neural network with Tensorboard. This tool is helpful to debug the program. Finally, Tensorflow is built to be deployed at scale. It runs on CPU and GPU.

3.3.3 Android Studio

Android Studio is the integrated development environment (IDE) for Android application development. It is based on the IntelliJ IDEA, a Java integrated development environment for software, and incorporates its code editing and developer tools.

CHAPTER 4

PROPOSED SYSTEM

4.1.WORKING MODEL OF PROPOSED SYSTEM

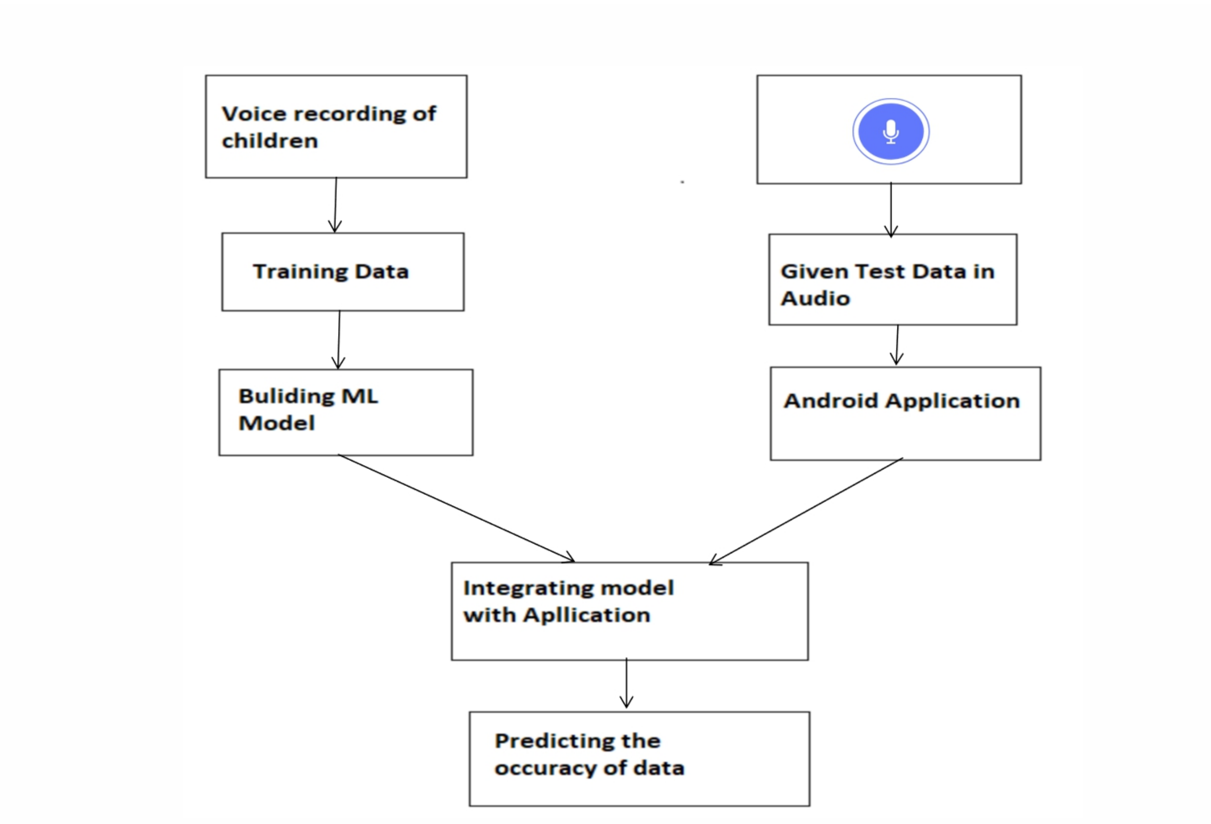

Voice recording of children
Training Data
Buliding ML Model
Given Test Data in Audio
Android Application
Integrating model with Apllication
Predicting the occuracy of data

Figure 4.1. Workflow of the Application

4.2. MODULE DESCRIPION

The aim of the proposed work is to predict the amount of correctness in the speech signal. Machine learning algorithms have been applied to classify the correctness percentage on the basis of audio taken from the observation from the audio recordings. The modules in the proposed system are

1. Data collection
2. Feature Selection
3. Building a ML model
    - 3.1 Data Preprocessing
        - 3.1.1. Padding
        - 3.1.2.Spectogram
        - 3.1.3One-hot-encoding
    - 3.2 Model Fitting with CNN
    - 3.3 Get a tflite file
4. Converting the model to tflite model
5. App Development
6. Integrating Model with application

1. Data collection

Voice recording of children aged between 6-14yrs is collected. Recordings consist of children saying small words. Then this audio files are padded, to 1 sec audio files. Then they are converted into wave forms for data pre-processing. In the dataset, the audio is in the Waveform Audio File Format (.wav) format. The wav file is an instance of a Resource Interchange file format (RIFF). The RIFF is a generic file container format for storing the data. Each audio data are collected had to be labeled so that the machine learning algorithm can be applied and both input & output data is an important part for pre-processing.

2. Feature Selection

Feature selection is also called variable selection or attribute selection. It is the automatic selection of attributes in your data (such as columns in tabular data) that are most relevant to the predictive modeling problem you are working on. Our Real time dataset collected from children consists of recordings. The children are asked to speak out a few words which are recorded and is used for training. It contains recordings of both normal and disordered audio recordings.

The phoneme's shape can be accurately predicted by identifying the shape of the voice, because the shape of the voice path determines the variations of the sound wave. The key to understanding speech is that a human are filtered by the shape of the vocal tract, including tongue and teeth etc., The shape of the vocal tract (speech tube) manifests itself in the envelope of a short time power spectrum, the work of MFCC's accurately represents this envelope. An audio signal is constantly changing, so the audio signal does not change much to simplify the things that consider in the short time scales. So, that the signal is divided into frames. If the frame is much shorter, don't have enough samples to get a reliable spectrum. The next step is to calculate the power spectrum of each frame.

3. Building a ML model

Machine learning consists of algorithms that can automate analytical model building. Using algorithms that iteratively learn from data, machine learning models facilitate computers to find hidden insights from Big Data without being explicitly programmed where to look. To create a successful machine learning model, you need to follow some steps: formulate the question, gather and clean the data, visualise the data, train the algorithm, evaluate the result based on the requirements. A machine learning model is a file that has been trained to recognize certain types of patterns. You train a model over a set of data, providing it an algorithm that it can use to reason over and learn from those data. The three stages to build the hypotheses in machine learning are model building, model testing and applying model.

3.1 Data Preprocessing

Data preprocessing is a crucial step that helps enhance the quality of data to promote the extraction of meaningful insights from the data. In simple words, data preprocessing is a data mining technique that transforms raw data into an understandable and readable format. The useful information that can be derived from it directly affects the ability of our model to learn; therefore, it is extremely important that we preprocess our data before feeding it into our model.

3.1.1 Padding

A “PAD” stands for Passive Attenuation Device. The circuitry in active mics may overload if the incoming signal from the capsule is too strong, causing audio signal distortion. Pads reduce signal levels before the active amplification process in order to avoid overloading the microphone circuitry. The recorded audios will be of different length. The audio files need to be trimmed to 1 sec length each. Padding is done to organize the wave files.

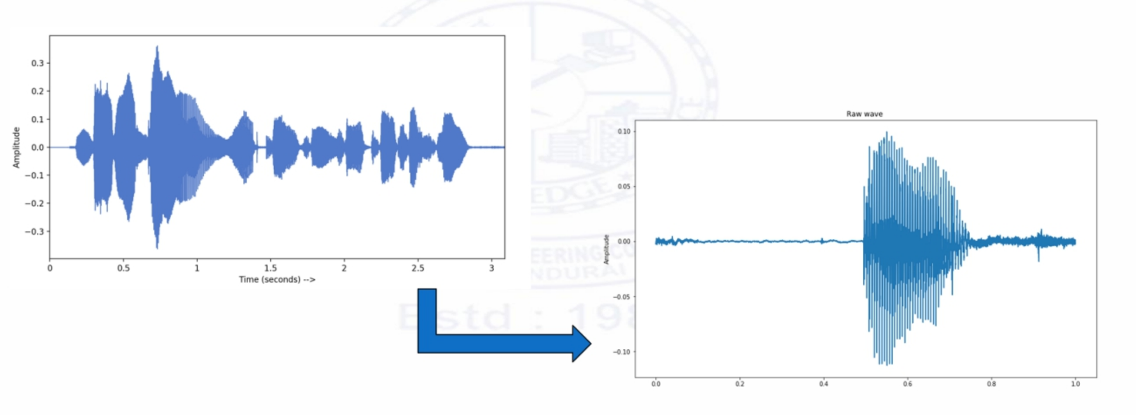


Figure 4.2. Padding a wave file

3.1.2 Spectogram

A spectrogram is a visual way of representing the signal strength or loudness, of a signal over time at various frequencies present in a particular waveform.A spectrogram is usually depicted as a heat map.Data is converted into short term Fourier transform. FFT converts signals such that we can know the amplitude of the given frequency at a given time. Using FFT we can determine the amplitude of various frequencies playing at a given time of an audio signal. The vertical axis shows frequencies , and the horizontal axis shows the time of the clip. Features can be obtained from a spectrogram by converting the linear frequency axis, as shown above, into a logarithmic axis. The resulting representation is also called a log-frequency spectrogram.

3.1.3 One-hot encoding

Lot of algorithms cannot work directly with categorical data.We need a way to convert categorical data into a numerical form and our machine learning algorithm can take in that as input.That categorical data is defined as variables with a finite set of label values. That most machine learning algorithms require numerical input and output variables. That an integer and one hot encoding is used to convert categorical data to integer data. A one hot encoding is a representation of categorical variables as binary vectors.This first requires that the categorical values be mapped to integer values.Then, each integer value is represented as a binary vector that is all zero values except the index of the integer, which is marked with a 1.

3.2. Model Fitting with CNN

Since CNNs are hungry for images, we want to transform the sound into an image. We have plot the audio signals with respect to time and frequency. These spectrograms now become an image representation of our spoken words. We make the network go through 30 epochs. You need to feed the pixels of the image in the form of arrays to the input layer of the neural network. The hidden layers- convolution layer, the ReLU layer, and pooling layer- carry out feature extraction by performing different calculations and manipulations. Finally, there’s a fully connected layer that identifies the word.

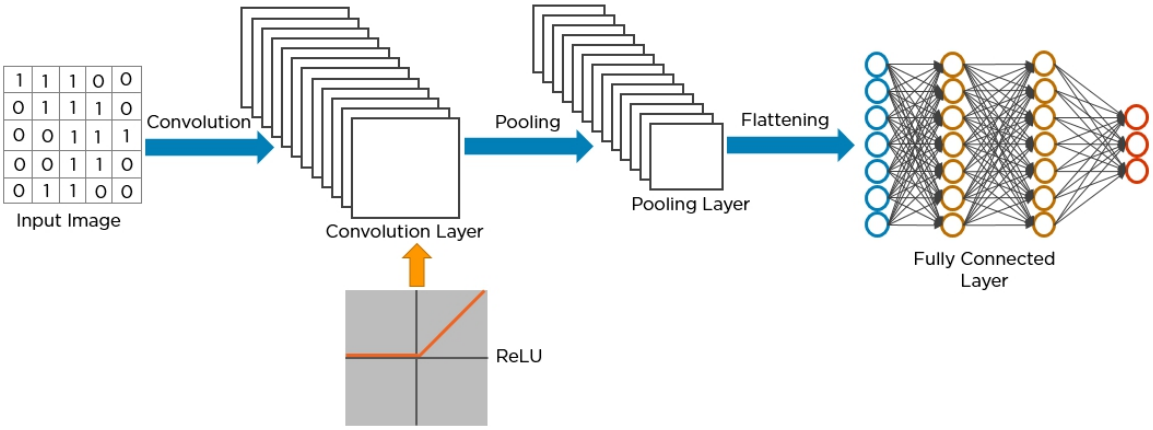


Figure 4.3. Convolutional Neural Networks

The process in the CNN model: The pixels from the image are fed to the convolutional layer that performs the convolution operation/It results in a convolved map. The convolved map is applied to a ReLU function to generate a rectified feature map .The image is processed with multiple convolutions and ReLU layers for locating the features. Different pooling layers with various filters are used to identify specific parts of the image. The pooled feature map is flattened and fed to a fully connected layer to get the final output.We finally get a tflite file built - digitsnet.tflite.

Convolutional Neural Networks

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.

An image is nothing but a matrix of pixel values. In basic binary images, the method might show an average precision score while performing prediction of classes but would have little to no accuracy when it comes to complex images having pixel dependencies throughout.

A ConvNet is able to successfully capture the Spatial and Temporal dependencies in an image through the application of relevant filters. The architecture performs a better fitting to the image dataset due to the reduction in the number of parameters involved and re-usability of weights. In other words, the network can be trained to understand the sophistication of the image better.

In the image, we have an RGB image which has been separated by its three color planes — Red, Green, and Blue. There are a number of such color spaces in which images exist — Grayscale, RGB, HSV, CMYK, etc.

The role of the ConvNet is to reduce the images into a form which is easier to process, without losing features which are critical for getting a good prediction. This is important when we are to design an architecture which is not only good at learning features but also is scalable to massive datasets.

Convolutional Layer — The Kernel

The element involved in carrying out the convolution operation in the first part of a Convolutional Layer is called the Kernel/Filter, K. The Kernel shifts 9 times because of Stride Length = 1 (Non-Strided)**,** every time performing a matrix multiplication operation between K and the portion P of the image over which the kernel is hovering.

Image Dimensions = 5 (Height) x 5 (Breadth) x 1 (Number of channels, eg. RGB)
The filter moves to the right with a certain Stride Value till it parses the complete width. Moving on, it hops down to the beginning (left) of the image with the same Stride Value and repeats the process until the entire image is traversed.

In the case of images with multiple channels (e.g. RGB), the Kernel has the same depth as that of the input image. Matrix Multiplication is performed between Kn and In stack ([K1, I1]; [K2, I2]; [K3, I3]) and all the results are summed with the bias to give us a squashed one-depth channel Convoluted Feature Output.

The objective of the Convolution Operation is to extract the high-level features such as edges, from the input image. ConvNets need not be limited to only one Convolutional Layer. Conventionally, the first ConvLayer is responsible for capturing the Low-Level features such as edges, color, gradient orientation, etc. With added layers, the architecture adapts to the High-Level features as well, giving us a network which has the wholesome understanding of images in the dataset, similar to how we would.

There are two types of results to the operation — one in which the convolved feature is reduced in dimensionality as compared to the input, and the other in which the dimensionality is either increased or remains the same. This is done by applying Valid Padding in case of the former, or Same Padding in the case of the latter.

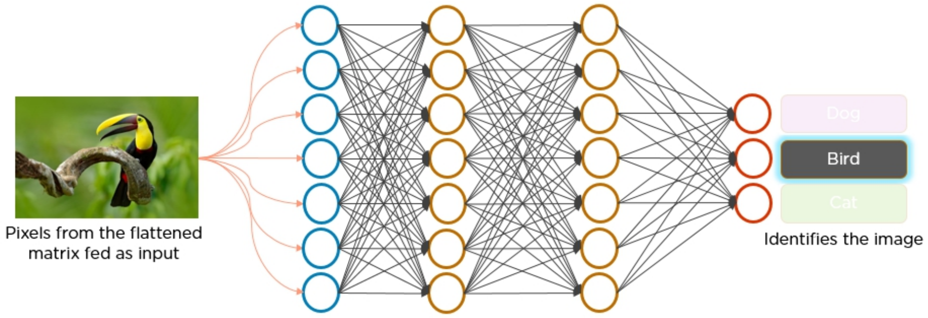


Figure 4.4. Convolutional Layer

ReLU Layer

The ReLU layer applies the function f(x) = max(0, x) to all of the values in the input volume. In basic terms, this layer just changes all the negative activations to 0. This layer increases the nonlinear properties of the model and the overall network without affecting the receptive fields of the conv layer.

The images are naturally non-linear.When you look at any image, you'll find it contains a lot of non-linear features. The rectifier serves to break up the linearity even further in order to make up for the linearity that we might impose an image when we put it through the convolution operation. For example, if we take a black and white image, by putting the image through the convolution process, or in other words, by applying a feature detector to it, the result will be composed of pixels that vary from white to black with many shades of gray in between.

In general, what the rectifier function does to an image like this is remove all the black elements from it, keeping only those carrying a positive value (the grey and white colors). The essential difference between the non-rectified version of the image and the rectified one is the progression of colors. If you look closely at the first one, you will find parts where a white streak is followed by a grey one and then a black one. After we rectify the image, you will find the colors changing more abruptly. The gradual change is no longer there. That indicates that the linearity has been disposed of.

Pooling Layer

The Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training of the model.

There are two types of Pooling: Max Pooling and Average Pooling. Max Pooling returns the maximum value from the portion of the image covered by the Kernel. On the other hand, Average Pooling returns the average of all the values from the portion of the image covered by the Kernel.

Max Pooling also performs as a Noise Suppressant**.** It discards the noisy activations altogether and also performs de-noising along with dimensionality reduction. On the other hand, Average Pooling simply performs dimensionality reduction as a noise suppressing mechanism. Hence, we can say that Max Pooling performs a lot better than Average Pooling**.**

The Convolutional Layer and the Pooling Layer, together form the i-th layer of a Convolutional Neural Network. Depending on the complexities in the images, the number of such layers may be increased for capturing low-levels details even further, but at the cost of more computational power.

After going through the above process, we have successfully enabled the model to understand the features. Moving on, we are going to flatten the final output and feed it to a regular Neural Network for classification purposes.

Fully Connected Layer

Fully Connected Layer is simply, feed forward neural networks. Fully Connected Layers form the last few layers in the network. The input to the fully connected layer is the output from the final Pooling or Convolutional Layer, which is flattened and then fed into the fully connected layer.

The output of convolution/pooling is flattened into a single vector of values, each representing a probability that a certain feature belongs to a label. The input values flow into the first layer of neurons. They are multiplied by weights and pass through an activation function (typically ReLu). They then pass forward to the output layer, in which every neuron represents a classification label. The fully connected part of the CNN network goes through its own backpropagation process to determine the most accurate weights. Each neuron receives weights that prioritize the most appropriate label. Finally, the neurons “vote” on each of the labels, and the winner of that vote is the classification decision.

4. Converting the model to tflite model

As we will deploy our model to a mobile device, we want our model to be as small and as fast as possible. Quantization is a common technique often used in on-device machine learning to shrink ML models. By using quantization, we often traded off a bit of accuracy for the benefit of having a significantly smaller model. Converting models reduces their file size and introduces optimizations with a little trade-offs. Here, in this project we finally get digitsnet.tflite file. We place it in the assets folder of the android project.

5. Integrating application with ML model

The whole model is compressed into a .tflite format to support the android application. This mobile application will act as a diagnostic and therapeutic app and can provide assistance to SSA teachers in diagnosing and delivering therapy, and increase the time a child is exposed to therapy at relatively low cost. By the use of this application, the children can improve the communication skill, learning by repeated exposure. Parents can use this app at any time of their convenience to train the children. It can reduce the therapeutic expense for the parents and reduce the difficulty of the special teacher in diagnosing and training and will bring better improvement in child's life.

6. App Development:

The main objective of the app -OPENTALK is to bring the user an interactive multimedia Application. The OPENTALK was developed in both Tamil and English using JAVA in Android Studio-3.5.3. The welcome screen prompts the user to select the preferred language which proceeds to the personal details screen. This page gets the basic details of the children The Final assessment is based on the assigned scores to the questions.

CHAPTER 5

RESULTS AND DISCUSSIONS

The app is a mobile speech recognition application written in Dart. It uses Flutter as mobile UI toolkit and TFLite as mobile machine learning framework for doing inference. Depending on the user's recorded voice it predicts which word out of 15 was recorded. Therefore, methods exposed by the TFLite binary are called with the help of the bindings.

Accuracy

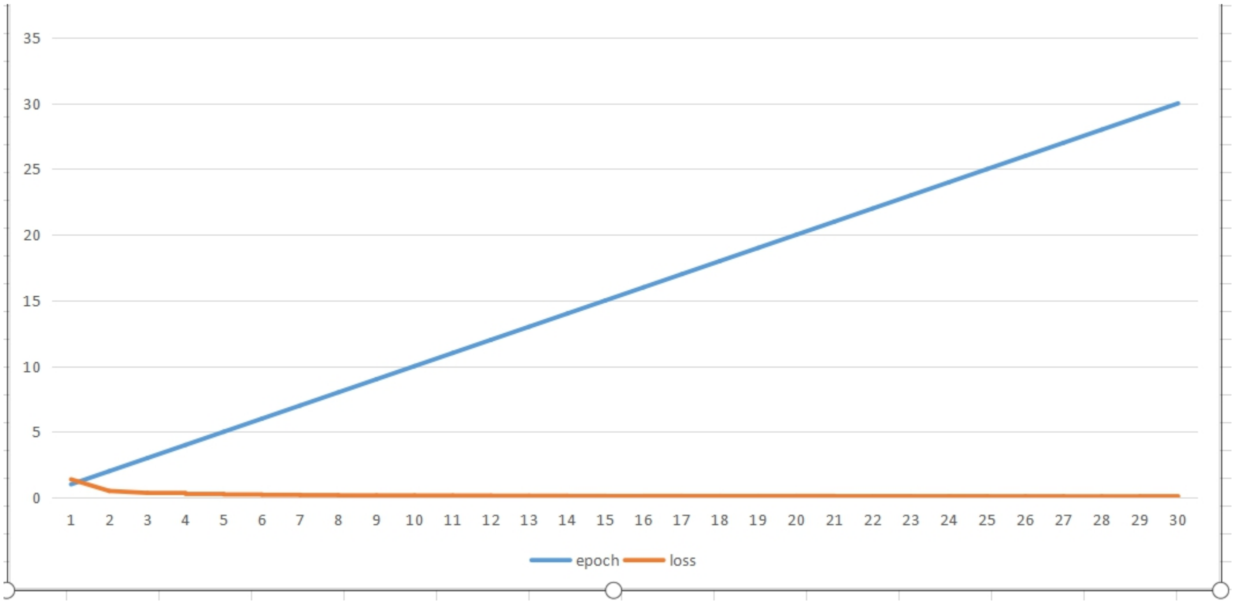
Accuracy is one such metric for evaluating classification models. It is the fraction of predictions our model got right. Mathematically, Accuracy may be defined as the ratio of the number of correct predictions to the total number of predictions. Our model yield a quite fair percentage (95%) of accuracy which reveals that most of the predictions for the type of disorder are correctly shown out.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy for our model is calculated to be 95%.

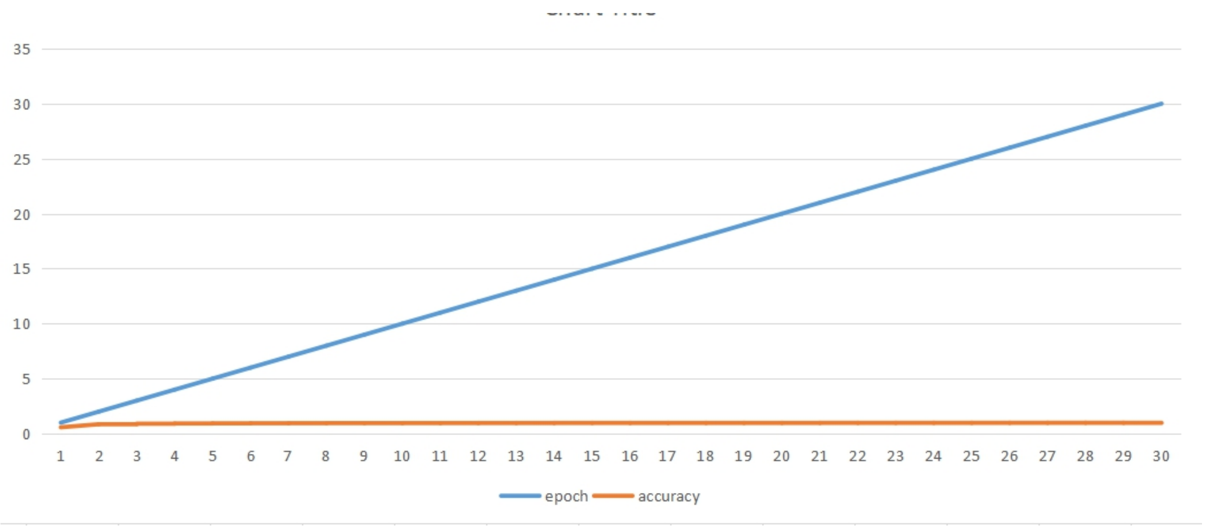
Epoch vs loss

Loss has reduced from first to 30 epoch.



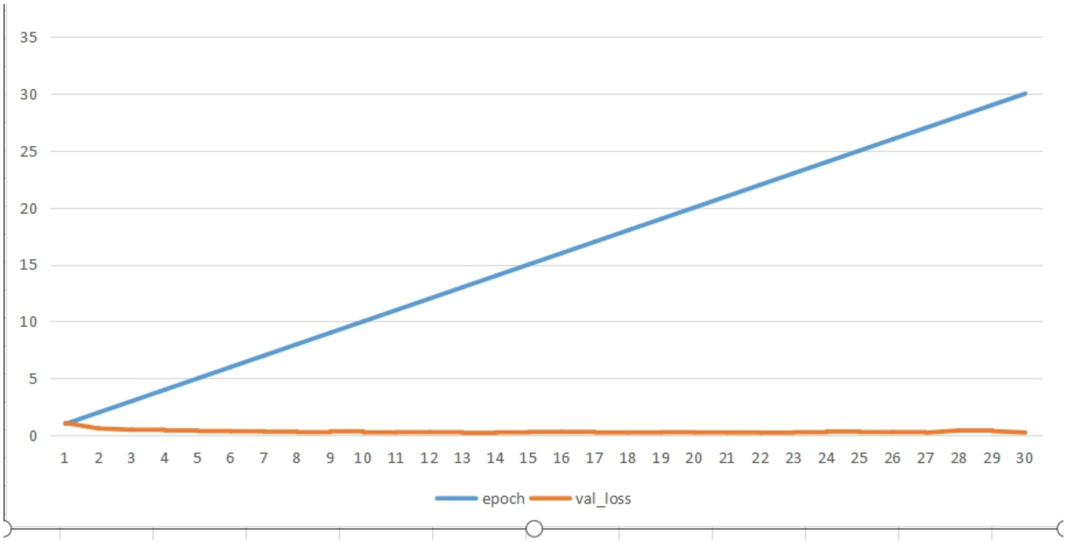
Epoch vs Accuracy

Accuracy has been constant throughout the project. At the 30th epoch its the highest of 98%.



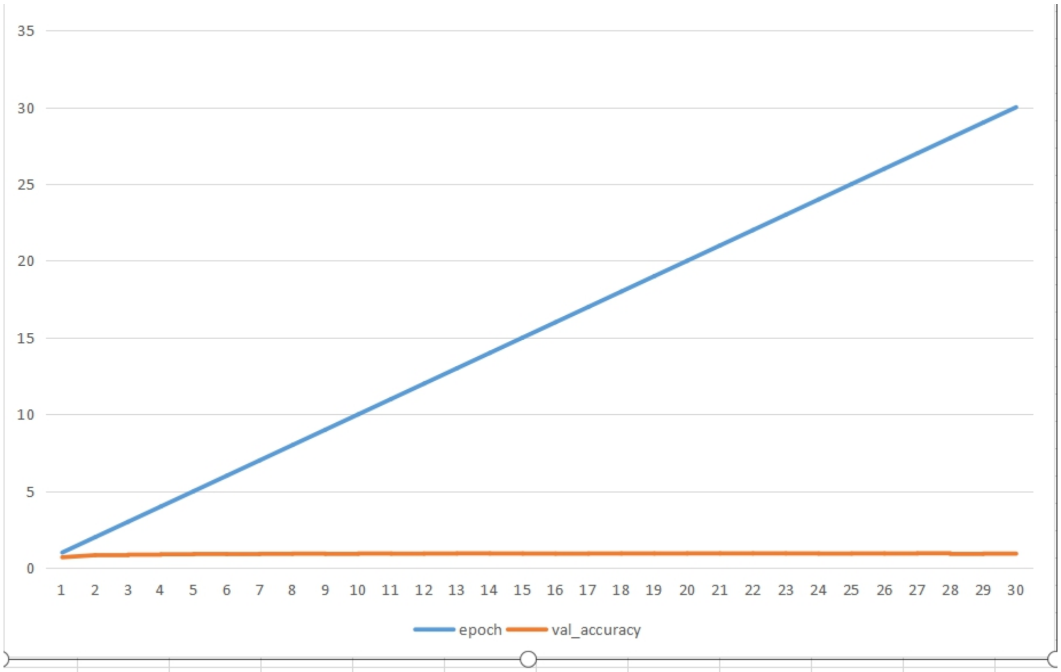
Epoch vs val_lass

There was both ups and downs in the graph. At the end the graph went down.



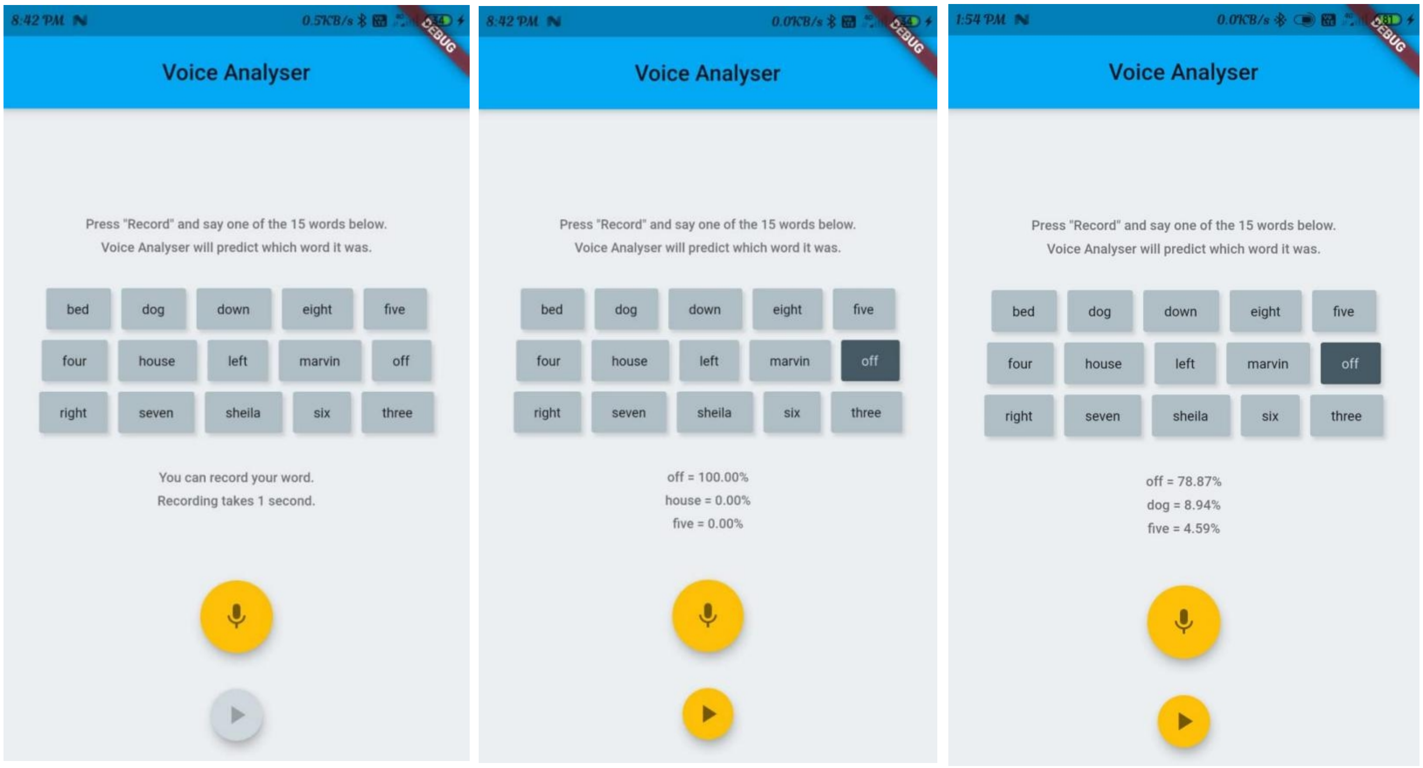
Epoch vs val_accuracy

The graph was constant through out 30 epochs.



Result

This app is able to predict simple recorded words depending on the user's voice by using a converted voice classification model under the hood. Once the application is started, the user can record one of the 15 words and immediately gets the prediction, which word was recorded. Hence, the word with the highest confidence is highlighted and the top three predictions are displayed. By pressing the play button the user can hear the previously recorded voice again.



Press "Record" and say one of the 15 words below.
Voice Analyser will predict which word it was.

You can record your word.
Recording takes 1 second.

```
off = 100.00%
house = 0.00%
five = 0.00%
```

```
off = 78.87%
dog = 8.94%
five = 4.59%
```

a) Screen in which audio can be recorded

b) Display for the correctness in recording

c) Display for the correctness in recording

Figure 5.1. Android application

CHAPTER 6

CONCLUSION AND FUTURE WORK

Android based application was developed for the welfare of children. Machine Learning based Convolutinal Neural Networks is created. Dataset consists of recordings of children saying few words, the dataset consists of both normal and disordered daata. Almost 30 records is given as an input to train the constructed model. Validated/Test data is given finally to check the accuracy of the constructed model. The accuracy is found out to be 95%. It will prompt the parents in early identification of the disorder and treating it as early as possible, with low cost.

This work will be further extended by developing one more voice based application for children side. In the next project we will have the children to speak A-Z, and small sentences. ML model will be developed and trained with audio dataset. Children can record long sentences and the correctness can be predicted accurately. This application will help the special school teachers and will bring better improvement in child's life. This application prevents the high level of severity in voice and can be treated at its early stage.

APPENDIX 1

CODE

```
import pathlib
from typing import Dict, List, Tuple


import numpy as np
import pandas as pd
import scipy.io.wavfile
import sklearn.preprocessing
import structlog
import tensorflow as tf
from tensorflow.keras.activations import relu
from tensorflow.keras.layers import (
    Activation,
    BatchNormalization,
    Conv1D,
    Dropout,
    Input,
    MaxPool1D,
    Reshape,
)
```

```
from tensorflow.keras.models import Model
from tensorflow.keras.regularizers import l2
from tqdm import tqdm

LOG = structlog.get_logger()



class SpeechModelTFLiteConverter:
    SAMPLE_RATE = 16000
    WINDOWS_SIZE = 128
    EPSILON = 1e-7
    N_WORDS = 15
    EPOCHS = 30

    def __init__(self) -> None:
        cwd = pathlib.Path.cwd()
        data_dir = cwd / "python_files/data"
        self.train_file_names_path = data_dir / "train_files.csv"
        self.validation_file_names_path = data_dir / "validation_files.csv"
        self.audio_dir = data_dir / "audio"
        self.overlap = self.WINDOWS_SIZE
        self.time_samples = self.SAMPLE_RATE
        assets_dir = cwd / "xain_voice_recognizer/assets"
        self.tflite_file_path = assets_dir / f"digitsnet.tflite"
        self.metrics_path = data_dir / "metrics.csv"

    def calculate_log_spectrogram(self, audio: np.ndarray) -> np.ndarray:
        number_of_frequencies = int(self.WINDOWS_SIZE / 2) + 1
        window = np.hanning(self.WINDOWS_SIZE)
        log_spectrogram = np.empty((self.time_samples - 1, number_of_frequencies))

        for i in range(1, self.time_samples):
            start = int((i - 1) * self.WINDOWS_SIZE / 2)
            end = int((i + 1) * self.WINDOWS_SIZE / 2)
            func = audio[start:end]
            transformed_signal = np.absolute(
                np.fft.fft(func * window)[:number_of_frequencies]
            )
            log_spectrogram[i - 1, :] = np.log(self.EPSILON + transformed_signal)

        return log_spectrogram

    def pad_audio(self, samples: np.ndarray) -> np.ndarray:
        if len(samples) >= self.SAMPLE_RATE:
            padded_samples = samples
        else:
```

```
        padded_samples = np.pad(
            samples,
            pad_width=(self.SAMPLE_RATE - len(samples), 0),
            mode="constant",
            constant_values=(0, 0),
        )
    return padded_samples


def process_vaw_files(
        self, files_sample: pd.DataFrame
) -> Tuple[np.ndarray, List[str]]:
    resized_overlap = self.overlap + 1
    n_samples = len(files_sample)
    x_array = np.empty((n_samples, self.time_samples - 1, resized_overlap))
    y_list = []

    LOG.info("processing audio files")
    for i, row in tqdm(files_sample.iterrows(), total=n_samples):
        file_path = self.audio_dir / row["class"] / row["filename"]
        _, samples = scipy.io.wavfile.read(file_path)
        padded_samples = self.pad_audio(samples)
        spectrogram = self.calculate_log_spectrogram(padded_samples)
        # add samples and truncate them when too long
        x_array[i, :, :] = spectrogram[:, :resized_overlap]
        y_list.append(row["class"])

    return x_array, y_list

@staticmethod
def encode_ys(
        y_train: List[str], y_validation: List[str], all_words: pd.DataFrame
) -> Tuple[np.ndarray, np.ndarray]:
    encoder = sklearn.preprocessing.OneHotEncoder(
        handle_unknown="ignore", sparse=False
    )
    encoder.fit(all_words)

    y_train_transformed = encoder.transform(pd.DataFrame(y_train))
    y_val_transformed = encoder.transform(pd.DataFrame(y_validation))

    return y_train_transformed, y_val_transformed

def conv_1d_time_stacked_model(self, input_size: Tuple[int, int]) -> Model:
         def _context_conv(tensor: tf.Tensor, num_filters: int, k: int) -> tf.Tensor:
        tensor = Conv1D(
            num_filters,
            k,
            padding="valid",
            dilation_rate=1,
            kernel_regularizer=l2(0.00002),
            use_bias=False,
        )(tensor)
        tensor = BatchNormalization()(tensor)
        tensor = Activation(relu)(tensor)
        return tensor

    def _reduce_conv(tensor: tf.Tensor, num_filters: int, k: int) -> tf.Tensor:
        tensor = Conv1D(
            num_filters,
            k,
            padding="valid",
            use_bias=False,
            kernel_regularizer=l2(0.00002),
        )(tensor)
        tensor = BatchNormalization()(tensor)
        tensor = Activation(relu)(tensor)
        tensor = MaxPool1D(pool_size=3, strides=2, padding="valid")(tensor)
        return tensor
```

```
input_layer = Input(shape=input_size)
tensor = input_layer

tensor = _context_conv(tensor, num_filters=16, k=1)
tensor = _reduce_conv(tensor, num_filters=32, k=3)
tensor = _context_conv(tensor, num_filters=32, k=3)

tensor = _reduce_conv(tensor, num_filters=64, k=3)
tensor = Dropout(0.1)(tensor)
tensor = _context_conv(tensor, num_filters=64, k=3)

tensor = _reduce_conv(tensor, num_filters=128, k=3)
tensor = Dropout(0.1)(tensor)
tensor = _context_conv(tensor, num_filters=128, k=3)

tensor = _reduce_conv(tensor, num_filters=249, k=3)
tensor = _context_conv(tensor, num_filters=249, k=3)

tensor = Dropout(0.1)(tensor)
tensor = Conv1D(self.N_WORDS, 9, activation="softmax")(tensor)
tensor = Reshape([-1])(tensor)
```

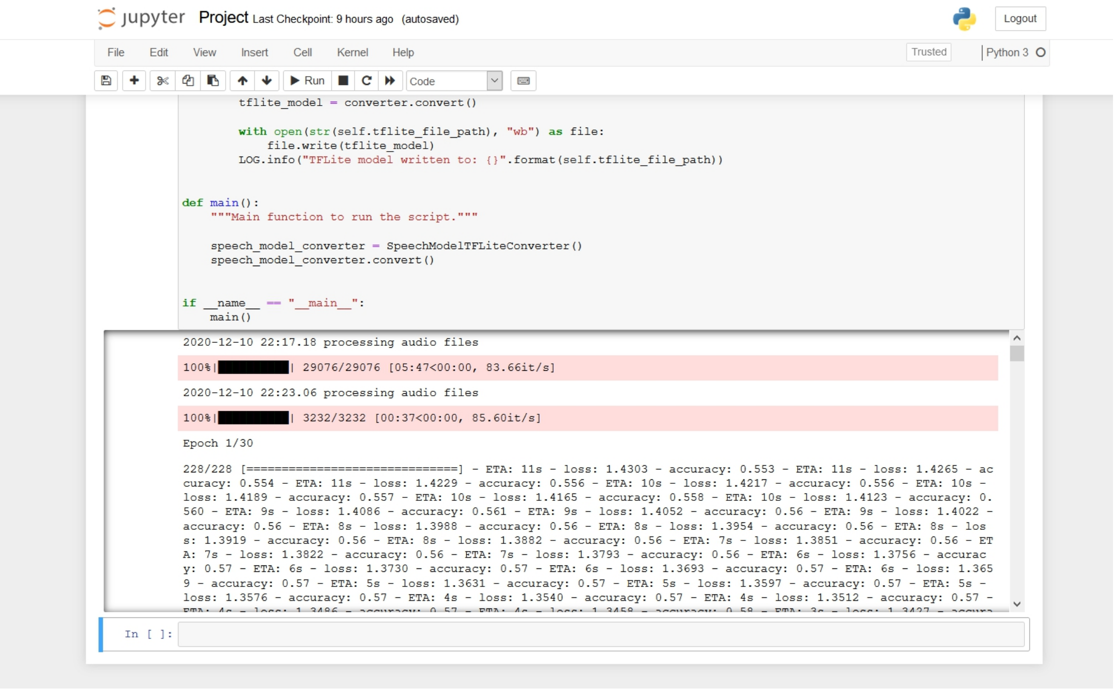
```
        model = Model(input_layer, tensor, name="conv_1d_time_stacked")
        model.compile(
            loss="categorical_crossentropy", optimizer="Adam", metrics=["accuracy"]
        )
        return model

    def prepare_data(self) -> Dict[str, np.ndarray]:
        train_sample_files = pd.read_csv(self.train_file_names_path, index_col=0)
        val_sample_files = pd.read_csv(self.validation_file_names_path, index_col=0)
        x_train, y_train, = self.process_vaw_files(train_sample_files)
        x_val, y_val = self.process_vaw_files(val_sample_files)

        words_1_15 = train_sample_files["class"].unique().tolist()
        all_words = pd.DataFrame(words_1_15)
        y_train_encoded, y_val_encoded = self.encode_ys(y_train, y_val, all_words)

        data = {
            "x_train": x_train,
            "y_train": y_train_encoded,
            "x_validation": x_val,
            "y_validation": y_val_encoded,
        }
        return data
```

```
def fit_model(self, data: Dict[str, np.ndarray]) -> Model:
    spectrogram_shape = data["x_train"].shape[1:]
    recognizer = self.conv_1d_time_stacked_model(spectrogram_shape)

    history = recognizer.fit(
        data["x_train"],
        data["y_train"],
        batch_size=128,
        epochs=self.EPOCHS,
        verbose=1,
        validation_data=(data["x_validation"], data["y_validation"]),
        shuffle=True,
    )
    history_df = pd.DataFrame(history.history)
    history_df.to_csv(self.metrics_path, index=False)

    return recognizer

def convert(self) -> None:
    data = self.prepare_data()
    recognizer = self.fit_model(data)
    converter = tf.lite.TFLiteConverter.from_keras_model(recognizer)
```

```
        converter.optimizations = [tf.lite.Optimize.DEFAULT]

        tflite_model = converter.convert()

        with open(str(self.tflite_file_path), "wb") as file:
            file.write(tflite_model)
        LOG.info("TFLite model written to: {}".format(self.tflite_file_path))


def main():
    speech_model_converter = SpeechModelTFLiteConverter()
    speech_model_converter.convert()


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

APPENDIX 2

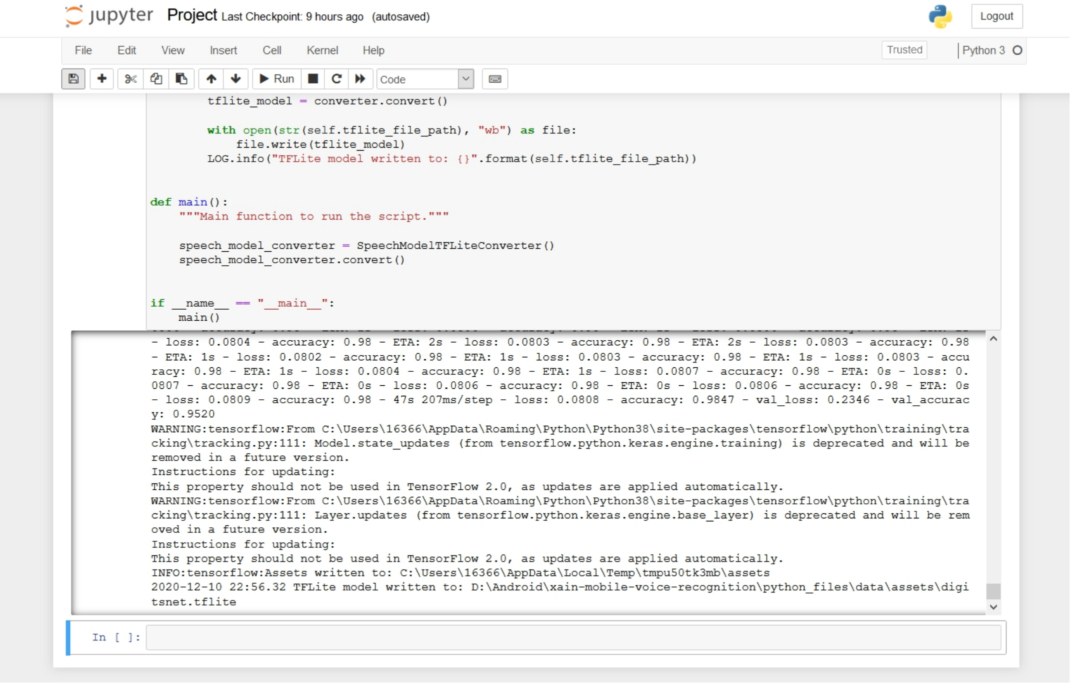
OUTPUT



```
        tflite_model = converter.convert()

        with open(str(self.tflite_file_path), "wb") as file:
            file.write(tflite_model)
        LOG.info("TFLite model written to: {}".format(self.tflite_file_path))


def main():
    """Main function to run the script."""

    speech_model_converter = SpeechModelTFLiteConverter()
    speech_model_converter.convert()


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

```
2020-12-10 22:17.18 processing audio files

100%|██████████| 29076/29076 [05:47<00:00, 83.66it/s]

2020-12-10 22:23.06 processing audio files

100%|██████████| 3232/3232 [00:37<00:00, 85.60it/s]

Epoch 1/30

228/228 [==============================] - ETA: 11s - loss: 1.4303 - accuracy: 0.553 - ETA: 11s - loss: 1.4265 - ac
curacy: 0.554 - ETA: 11s - loss: 1.4229 - accuracy: 0.556 - ETA: 10s - loss: 1.4217 - accuracy: 0.556 - ETA: 10s -
loss: 1.4189 - accuracy: 0.557 - ETA: 10s - loss: 1.4165 - accuracy: 0.558 - ETA: 10s - loss: 1.4123 - accuracy: 0.
560 - ETA: 9s - loss: 1.4086 - accuracy: 0.561 - ETA: 9s - loss: 1.4052 - accuracy: 0.56 - ETA: 9s - loss: 1.4022 -
accuracy: 0.56 - ETA: 8s - loss: 1.3988 - accuracy: 0.56 - ETA: 8s - loss: 1.3954 - accuracy: 0.56 - ETA: 8s - los
s: 1.3919 - accuracy: 0.56 - ETA: 8s - loss: 1.3882 - accuracy: 0.56 - ETA: 7s - loss: 1.3851 - accuracy: 0.56 - ET
A: 7s - loss: 1.3822 - accuracy: 0.56 - ETA: 7s - loss: 1.3793 - accuracy: 0.56 - ETA: 6s - loss: 1.3756 - accurac
y: 0.57 - ETA: 6s - loss: 1.3730 - accuracy: 0.57 - ETA: 6s - loss: 1.3693 - accuracy: 0.57 - ETA: 6s - loss: 1.365
9 - accuracy: 0.57 - ETA: 5s - loss: 1.3631 - accuracy: 0.57 - ETA: 5s - loss: 1.3597 - accuracy: 0.57 - ETA: 5s -
loss: 1.3576 - accuracy: 0.57 - ETA: 4s - loss: 1.3540 - accuracy: 0.57 - ETA: 4s - loss: 1.3512 - accuracy: 0.57 -
ETA: 4s - loss: 1.3486 - accuracy: 0.57 - ETA: 4s - loss: 1.3458 - accuracy: 0.58 - ETA: 3s - loss: 1.3427 - accura
```

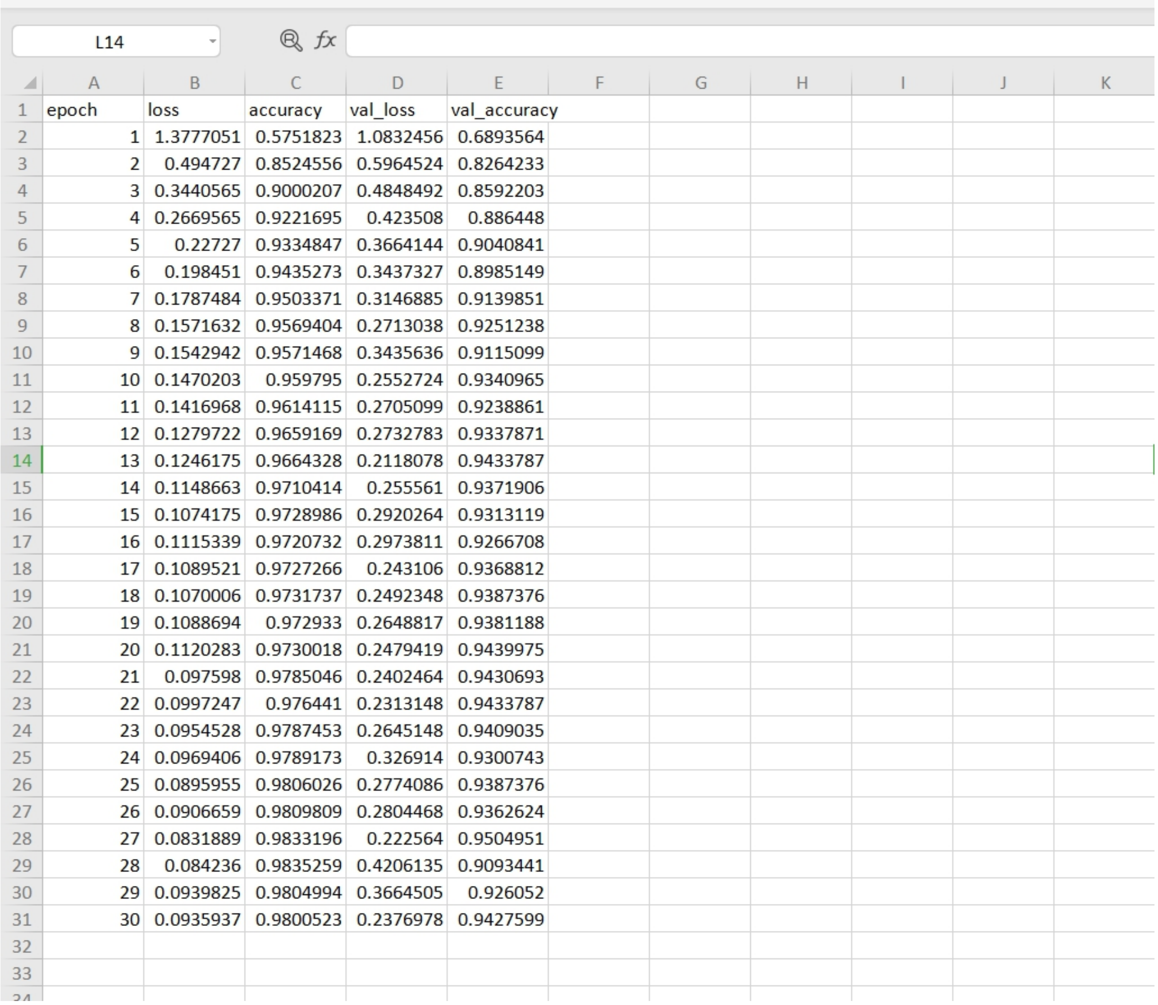
```
In [ ]:
```



```
        tflite_model = converter.convert()

        with open(str(self.tflite_file_path), "wb") as file:
            file.write(tflite_model)
        LOG.info("TFLite model written to: {}".format(self.tflite_file_path))


def main():
    """Main function to run the script."""

    speech_model_converter = SpeechModelTFLiteConverter()
    speech_model_converter.convert()


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

```
- loss: 0.0804 - accuracy: 0.98 - ETA: 2s - loss: 0.0803 - accuracy: 0.98 - ETA: 2s - loss: 0.0803 - accuracy: 0.98
- ETA: 1s - loss: 0.0802 - accuracy: 0.98 - ETA: 1s - loss: 0.0803 - accuracy: 0.98 - ETA: 1s - loss: 0.0803 - accu
racy: 0.98 - ETA: 1s - loss: 0.0804 - accuracy: 0.98 - ETA: 1s - loss: 0.0807 - accuracy: 0.98 - ETA: 0s - loss: 0.
0807 - accuracy: 0.98 - ETA: 0s - loss: 0.0806 - accuracy: 0.98 - ETA: 0s - loss: 0.0806 - accuracy: 0.98 - ETA: 0s
- loss: 0.0809 - accuracy: 0.98 - 47s 207ms/step - loss: 0.0808 - accuracy: 0.9847 - val_loss: 0.2346 - val_accurac
y: 0.9520
WARNING:tensorflow:From C:\Users\16366\AppData\Roaming\Python\Python38\site-packages\tensorflow\python\training\tra
cking\tracking.py:111: Model.state_updates (from tensorflow.python.keras.engine.training) is deprecated and will be
removed in a future version.
Instructions for updating:
This property should not be used in TensorFlow 2.0, as updates are applied automatically.
WARNING:tensorflow:From C:\Users\16366\AppData\Roaming\Python\Python38\site-packages\tensorflow\python\training\tra
cking\tracking.py:111: Layer.updates (from tensorflow.python.keras.engine.base_layer) is deprecated and will be rem
oved in a future version.
Instructions for updating:
This property should not be used in TensorFlow 2.0, as updates are applied automatically.
INFO:tensorflow:Assets written to: C:\Users\16366\AppData\Local\Temp\tmpu50tk3mb\assets
2020-12-10 22:56.32 TFLite model written to: D:\Android\xain-mobile-voice-recognition\python_files\data\assets\digi
tsnet.tflite
```

```
In [ ]:
```



epoch	loss	accuracy	val_loss	val_accuracy
1	1.3777051	0.5751823	1.0832456	0.6893564
2	0.494727	0.8524556	0.5964524	0.8264233
3	0.3440565	0.9000207	0.4848492	0.8592203
4	0.2669565	0.9221695	0.423508	0.886448
5	0.22727	0.9334847	0.3664144	0.9040841
6	0.198451	0.9435273	0.3437327	0.8985149
7	0.1787484	0.9503371	0.3146885	0.9139851
8	0.1571632	0.9569404	0.2713038	0.9251238
9	0.1542942	0.9571468	0.3435636	0.9115099
10	0.1470203	0.959795	0.2552724	0.9340965
11	0.1416968	0.9614115	0.2705099	0.9238861
12	0.1279722	0.9659169	0.2732783	0.9337871
13	0.1246175	0.9664328	0.2118078	0.9433787
14	0.1148663	0.9710414	0.255561	0.9371906
15	0.1074175	0.9728986	0.2920264	0.9313119
16	0.1115339	0.9720732	0.2973811	0.9266708
17	0.1089521	0.9727266	0.243106	0.9368812
18	0.1070006	0.9731737	0.2492348	0.9387376
19	0.1088694	0.972933	0.2648817	0.9381188
20	0.1120283	0.9730018	0.2479419	0.9439975
21	0.097598	0.9785046	0.2402464	0.9430693
22	0.0997247	0.976441	0.2313148	0.9433787
23	0.0954528	0.9787453	0.2645148	0.9409035
24	0.0969406	0.9789173	0.326914	0.9300743
25	0.0895955	0.9806026	0.2774086	0.9387376
26	0.0906659	0.9809809	0.2804468	0.9362624
27	0.0831889	0.9833196	0.222564	0.9504951
28	0.084236	0.9835259	0.4206135	0.9093441
29	0.0939825	0.9804994	0.3664505	0.926052
30	0.0935937	0.9800523	0.2376978	0.9427599

REFERENCES

1. Witsawakiti et al (2006) have described an e-training tool designed to help hearing impaired children learn and practice words in Thai language more correctly. The tool uses speech to overcome the limitations of the traditional face-to-face speech therapy.
2. Konstantinidis, et al (2009) have evaluated the implementation of ACALPA platform uses an affective avatar, synthesized speech and multimedia content aiming at supporting and facilitating the teacher-child interaction.
3. Bastanfard et al (2010) have developed a software system which facilitates the interaction and synchronization of language learning activities for Persian hearing-impaired children..
4. Toki, E. I., &amp; Pange(2010) introduced an e-learning system for improving articulation in Greek preschoolers.The results of the study on this software showed that children not only improved their articulation but they also increased on language activities success by acquiring new vocabulary.
5. Schipor, O. A et al (2012) presented a tool which contains Children Manager, 3D Articulator Model, and Homework Manager all installed on the child's PC aiming at improving pre-school children's articulation.
6. Robles-Bykbaev et al(2014) have developed an ecosystem of intelligent ICT tools consists of an expert system to support speech and language pathologists, doctors, students, patients and their relatives.

.