JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

TEXT TO VOICE CONVERSION FOR VISUALLY IMPAIRED

[1] M. Naga Keerthi, [2] Singupuram Sravan Kumar,
[1]Assistant Professor, [2]MCA 2nd year,
[2]Master of Computer Applications,
[2]Sanketika Vidya Parishad Engineering College, Visakhapatnam, India

ABSTRACT

A Text-to-speech synthesizer is an application that converts text into spoken word, by analyzing and processing the text using Natural Language Processing (NLP) and then using Digital Signal Processing (DSP) technology to convert this processed text into synthesized speech representation of the text. Here, we developed a useful text-to-speech synthesizer in the form of a simple application that converts inputted text into synthesized speech and reads out to the user which can then be saved as an mp3 file. The development of a text to speech synthesizer will be of great help to people with visual impairment and make making through large volume of text easier.

Keywords

OpenCV , Text-to-speech, Tesseract, RNN, Bi-directional LSTM .

1.1 INTRODUCTION

Machine Learning is a branch of Artificial Intelligence that enables machines to naturally learn and improve based on their experiences. It is described as the branch of science that enables computers to learn without being explicitly programmed. Machine learning has the potential to unlock the value of consumer and business data and enable companies to make decisions that keep them ahead of the competition. It is being used in various sectors like Health care, hospitality, travel, manufacturing, financial services, energy etc. Machine learning helps people by assisting them and quickly completing the tasks for them.

1 .2 SCOPE OF THE PROJECT

There are many deep learning algorithms for text detection and recognition that can solve the above problem. Here in the first model OpenCV is used for text detection and Tesseract for recognition purpose is used and second model is based on RNN and Bi-directional LSTM. Only the text region is detected and predicted instead of detecting the non textual region like in the conventional approaches in the natural scene images. The final goal of text detection i.e., word or text line level detection is predicted. The model abandons unnecessary intermediate steps like candidate proposal, text region formation and word partition and allows for end-to-end training and optimization.

2.1 DEEP LEARNING

The structure of one deep learning model differs from another in the number of layers used. The process of identifying the important aspects from voice, text, images is known as feature extraction. The feature extraction is done by applying convolutions over the inputs. The convolutions are nothing but some mathematical operations performed over the inputs with the help of the kernels. If an image is considered as an input, then the image is represented in the form of matrix. Where the elements of the matrix represent the pixel intensities of each point on the image. The resultant output is known as feature map. The number of convolutions applied differs from one model to another model.

2.2 TEXT DETECTION

Recently, extracting and understanding textual information embedded in natural scenes have become increasingly important and popular, which is evidenced by the unprecedented large numbers of participants of the International conference on document analysis and recognition (ICDAR) series contests. It is a pre requisite of the subsequent processes which plays a critical role in the whole procedure of textual information extraction and understanding. The process of identification of the probability of presence of text in the image is known as text detection. It gives the bounding boxes around the text in the image.

2.3 TEXT RECOGNITION

Recognizing text in images is a research problem that has attracted significant interest in the last few years due to its numerous potential applications in document image analysis, image retrieval, scene understanding, visual assistance, and so on. Early work focused on images of printed text, which can be interpreted with traditional sliding-window based Optical Character Recognition (OCR) techniques.

In the last few years, great progress has been made with the development of methods that use deep convolutional neural networks (CNNs) to spot text in natural scene images that is to both locate text target regions and then recognize the words in these regions.

3.1 EXISTING SYSTEMS

In the paper 'Image to Speech Conversion for Visually Impaired', the authors Asha , Thota, Bera, Fatima Shaik[2] explain about the device which can detect efficiently from any complex background. The basic framework is an embedded system that captures an image, extracts the image that contains text and then converts that text to corresponding speech. A series of image pre-processing steps are used to find the text and remove the background. Here, one of the disadvantages is that when edge detection is performed some of the alphabets can be miss-detected and this leads to false output generation by the OCR.

3.2 PROPOSED SYSTEMS

Primary education is generally imparted to students with visual impairment by the means of Braille books in most schools. During primary classes education for the visually impaired stresses on proficiency in Braille language, while basic knowledge in computers is also provided to the Braille is a tactile code used by persons with visual impairment to read and write in any language, from English, Chinese to mathematics and music. Braille devices allows persons with visual impairment to read and produce content in Braille Sight is previewing its eSight Go, a wearable device that allows users to gain a level of independence. According to the company, eSight will be available on the market starting in Q4 2023.

3.3 FEASIBILITY STUDY

While using older techniques before the existence of the deep learning era for text detection and recognition like Stroke width transform (SWT), Maximally stable extremal regions (MSER) based methods generally seek character candidates via edge detection or extremal region extraction. In SWT the output is not accurate as the texts are not detected properly.

4 SPECIFICATIONS

3.1 HARDWARE REQUIREMENTS

1. Raspberry pi 3b+
2. USB Camera
3. Adapter
4. IR sensor
5. Headphone
6. Laptop

3.2 SOFTWARE REQUIREMENTS

1. Raspbian OS
2. Python software
3. VNC viewer

4 ARCHITECTURE

4.1 system architecture

In this step the image of the text is captured using raspberry pi camera or an HD webcam with high resolution. The acquired image is then applied to the image preprocessing step for reduction of unwanted noise. In image processing, it is defined as the action of retrieving an image from some source, usually a hardware-based source for processing it is first step in the workflow sequence because without an image, no processing is possible.

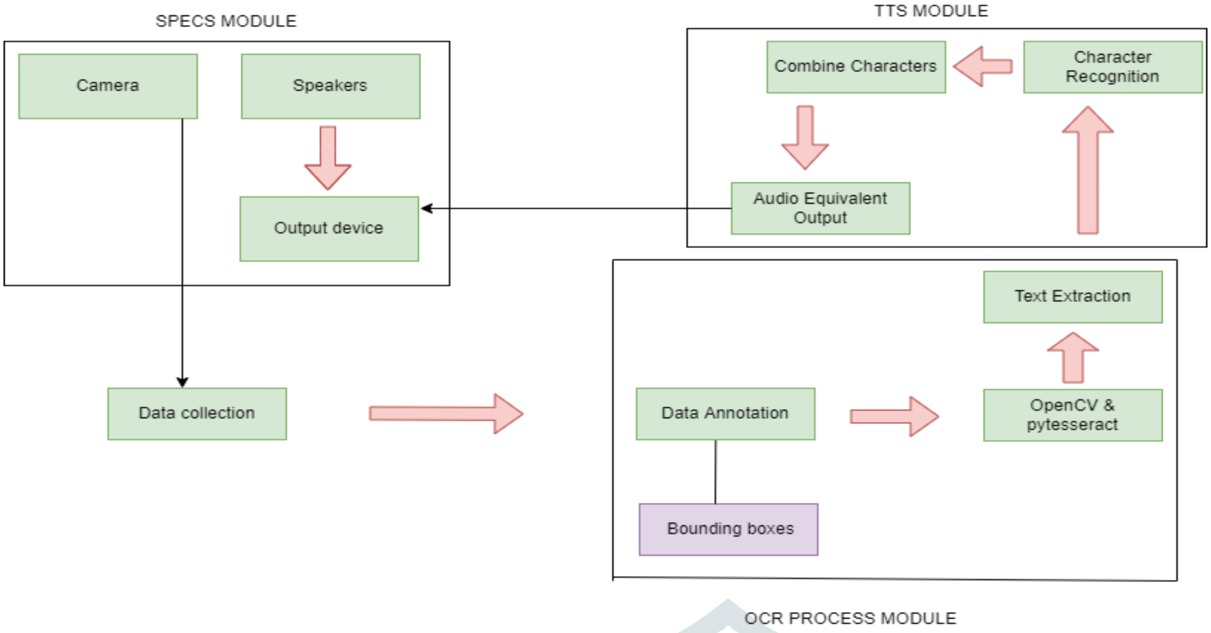


FIGURE 1: System Architecture

4.2 DATAFLOW DIAGRAMS

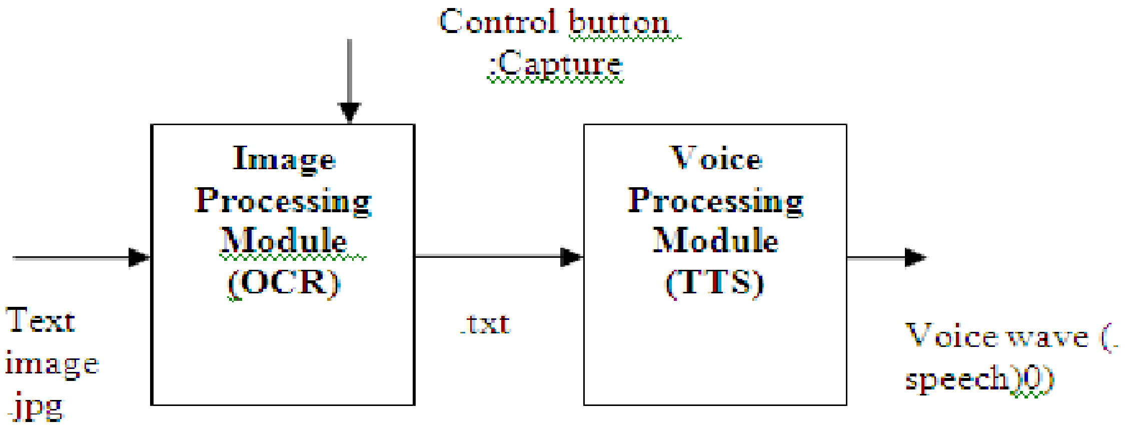


FIGURE 2: Data Flow Diagram

Text to speech (TTS) synthesis is the automatic conversion of text into speech. Generally, TTS system consists of two phases. The first is text analysis, where the input text is transcribed into a phonetic or some other linguistic representation. The second one is the generation of speech waveforms.

5. ALGORITHM

5.1 ML ALGORITHM

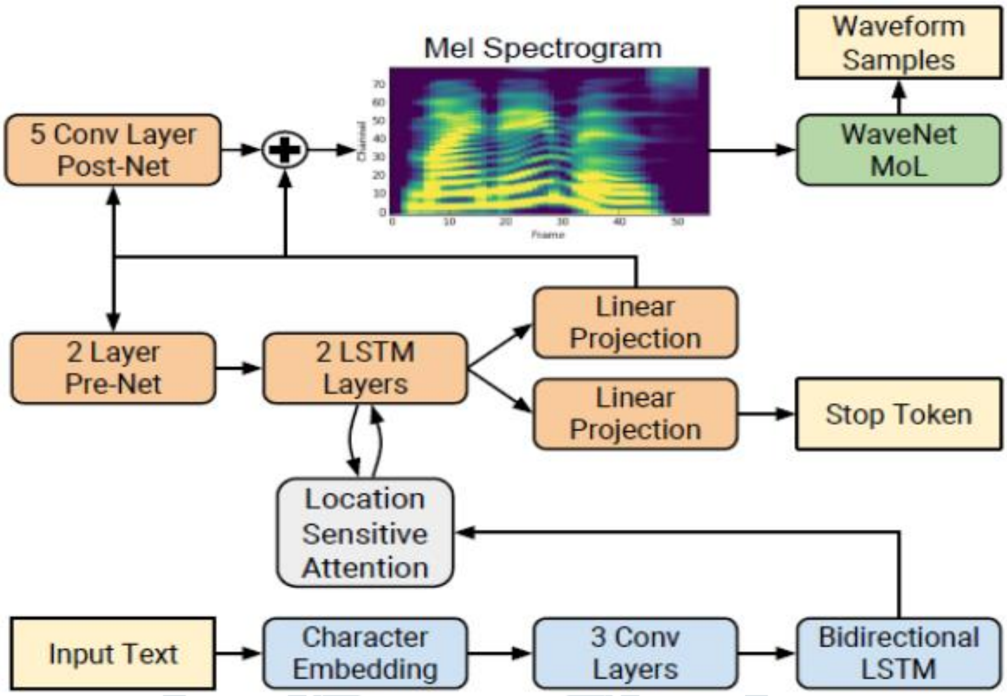


FIGURE 3: ML algorithm

The ML algorithm establishes the connection between phonemes and sounds, giving them accurate intonations. The system uses a sound wave generator to create a vocal sound. The frequency characteristics of phrases obtained from the acoustic model are eventually loaded into the sound wave generator. Text-to- speech synthesizer (TTS) is the technology which lets computer speak to you. The TTS system gets the text as the input and then a computer algorithm which called TTS engine analyses the text, pre-processes the text and synthesizes the speech with some mathematical models.

5.2 Module for ML ALGORITHM

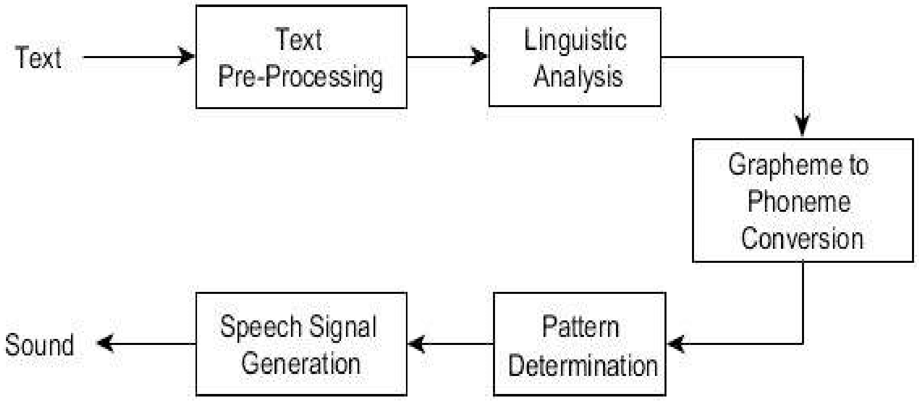


FIGURE 4: Module of ML algorithm

Text to speech (TTS) synthesis is the automatic conversion of text into speech. Generally, TTS system consists of two phases. The first is text analysis, where the input text is transcribed into a phonetic or some other linguistic representation. The second one is the generation of speech waveforms.

6 DATASET

6.1 Dataset Collection

IAM dataset is collected in which it contains an image of some handwritten text and its corresponding target is the string present in the image. It consists of variable length ground-truth targets. This dataset is used across many Optical character recognition (OCR) benchmarks. Only English texts are present in this dataset. The database contains forms of unconstrained handwritten text, which were scanned at a resolution of 300dpi and saved as PNG images with 256 Gray levels. It provides samples of a complete form, a text line and some extracted words. All forms and also all extracted text lines, words and sentences are available for download as PNG files, with corresponding XML meta-information included into the image files.

6.2 SAMPLE IMAGE PREVIEW

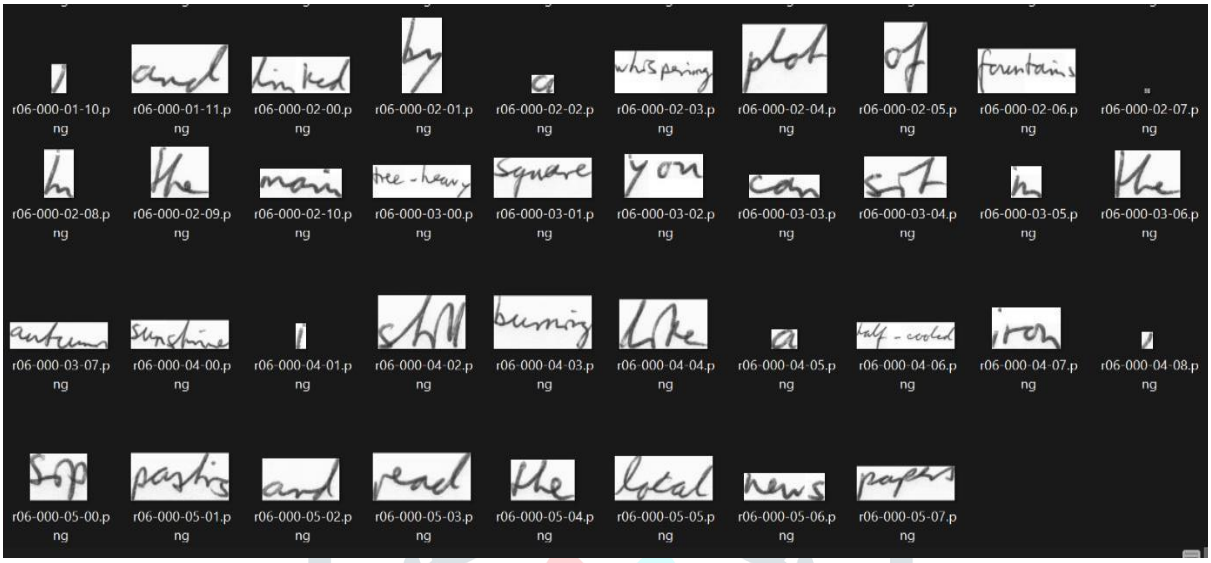
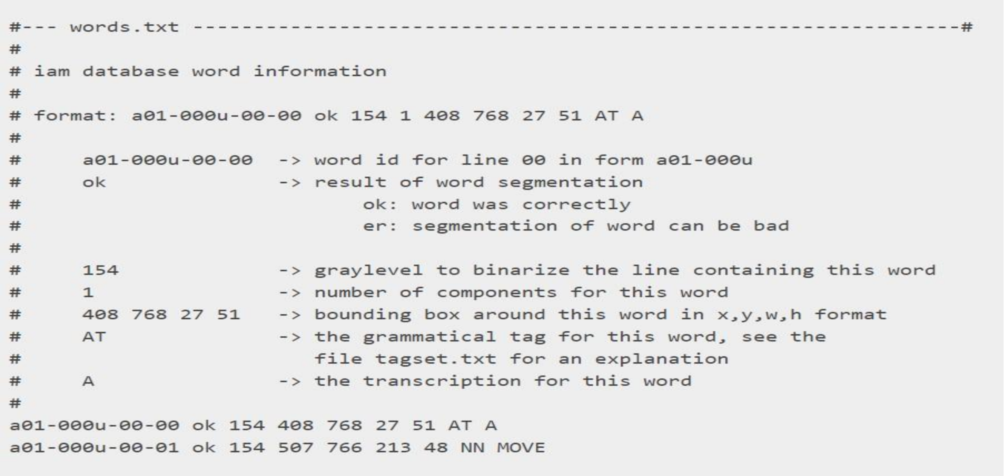


FIGURE 5: Sample Image preview

The KNFB Reader is a print to speech application that runs on iOS or Android mobile device. The app enables the camera to take pictures of printed material, rapidly convert the images into text, and read the text aloud using high quality text-to-speech, TTS.

6.3 PREVIEW OF DATASET



```
#--- words.txt ---------------------------------------------------------------#
#
# iam database word information
#
# format: a01-000u-00-00 ok 154 1 408 768 27 51 AT A
#
#     a01-000u-00-00  -> word id for line 00 in form a01-000u
#     ok              -> result of word segmentation
#                            ok: word was correctly
#                            er: segmentation of word can be bad
#
#     154             -> graylevel to binarize the line containing this word
#     1               -> number of components for this word
#     408 768 27 51   -> bounding box around this word in x,y,w,h format
#     AT              -> the grammatical tag for this word, see the
#                        file tagset.txt for an explanation
#     A               -> the transcription for this word
#
a01-000u-00-00 ok 154 408 768 27 51 AT A
a01-000u-00-01 ok 154 507 766 213 48 NN MOVE
```

FIGURE 6 : PREVIEW OF DATASET

6.4 DATA PREPROCESSING

Cleaning of the validation and test labels is done. provides different pre-processing layers but here character level pre-processing labels is used. For ex: Suppose there are two labels say “cat” and “dog”, then the character vocabulary should be {a, c, d, g, o, t} (without any special tokens). String Lookup layer is used for this purpose. Initially map strings from a vocabulary to integer indices. This layer uses a table-based lookup to convert a set of arbitrary texts into an integer output, with optional out-of-vocabulary management. Attributes that it takes are token and vocabulary.

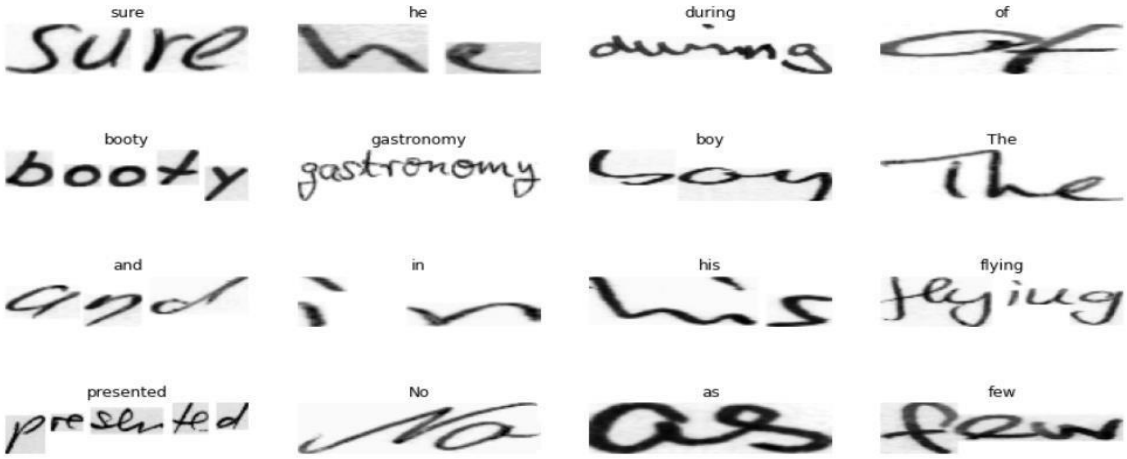


FIGURE 7: RESULTS AFTER RESHAPING THE IMAGE

7 LIBRARIES

7.1 MODEL 1

7.1.1 OPENCV

Computer vision is a method for understanding how images and videos are stored, as well as manipulating and retrieving data from them. Artificial Intelligence is built on the foundation of computer vision. Self-driving cars, robotics, and photo editing apps all rely heavily on computer vision. OpenCV is an open-source computer vision, machine learning, and image processing toolkit. Python, C++, Java, and other programming languages are supported by OpenCV. It can analyse images and videos to recognise features, faces, and even human handwriting. When it's combined with other libraries, like Numpy which is a highly efficient library for numerical operations, the number of weapons in your arsenal grows, as any operation that Numpy can do may be merged with OpenCV. The OpenCV array structure can be processed by Python for analysis. A vector space is employed and executes mathematical operations on these features to identify visual patterns and their various features.

7.2 MODEL 2

7.2.1 Tensoflow

TensorFlow is an open source numerical computing library. It was developed by Google and is available under the Apache 2.0 open source licence. Although there is access to the underlying C++ API, the API is ostensibly for the Python programming language. TensorFlow, unlike other numerical libraries for Deep Learning, like as Theano, was developed for use in both research and production systems, including RankBrain in Google search and the fun DeepDream project.

7.2.3 MatPlotdb

In most cases, a Python matplotlib script is constructed so that only a few lines of code are required to produce a visual data plot. Two APIs are overlaid by the matplotlib scripting layer:

- The pyplot API is a tree of Python code objects, with matplotlib at the top.
- pyplot is a collection of Object-Oriented API objects that can be constructed with more flexibility than pyplot.

7.2.3 Numpy

Jim Hugunin created Numeric, the forerunner of NumPy. Num array, a new package with some extra features, was also built. Numerical python (Numpy) is a library which consists the objects of multidimensional arrays and a group of routines in order to process those arrays. Various mathematical and logical operations can be performed on the arrays using the Numpy library.

8 CNN ARCHITECTURE

8.1 BASIC CNN ARCHITECTURE

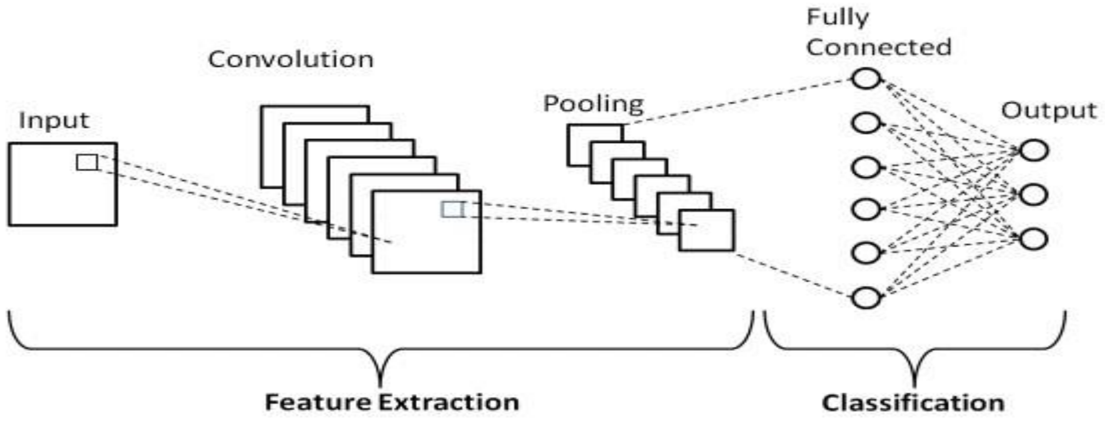


FIGURE 8: Basic CNN architecture Design

Modern neural networks use non-linear function as their activation function to fire the neuron. The most used Non-linear activation functions are Sigmoid Activation Function, Tanh, ReLU (Rectified Linear Units), Leaky ReLU, ELU (Exponential Linear Units), SoftMax.

8.2 Bi-directional Long Short Term Memory (Bi-LSTM)

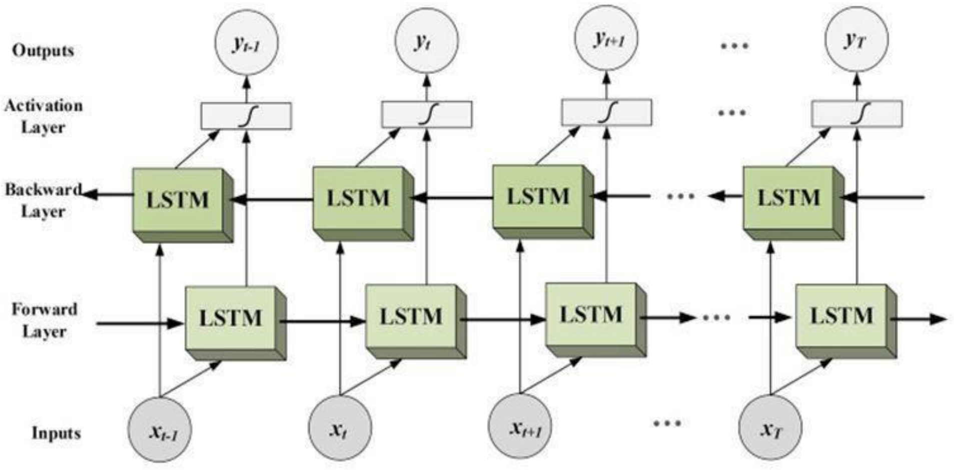
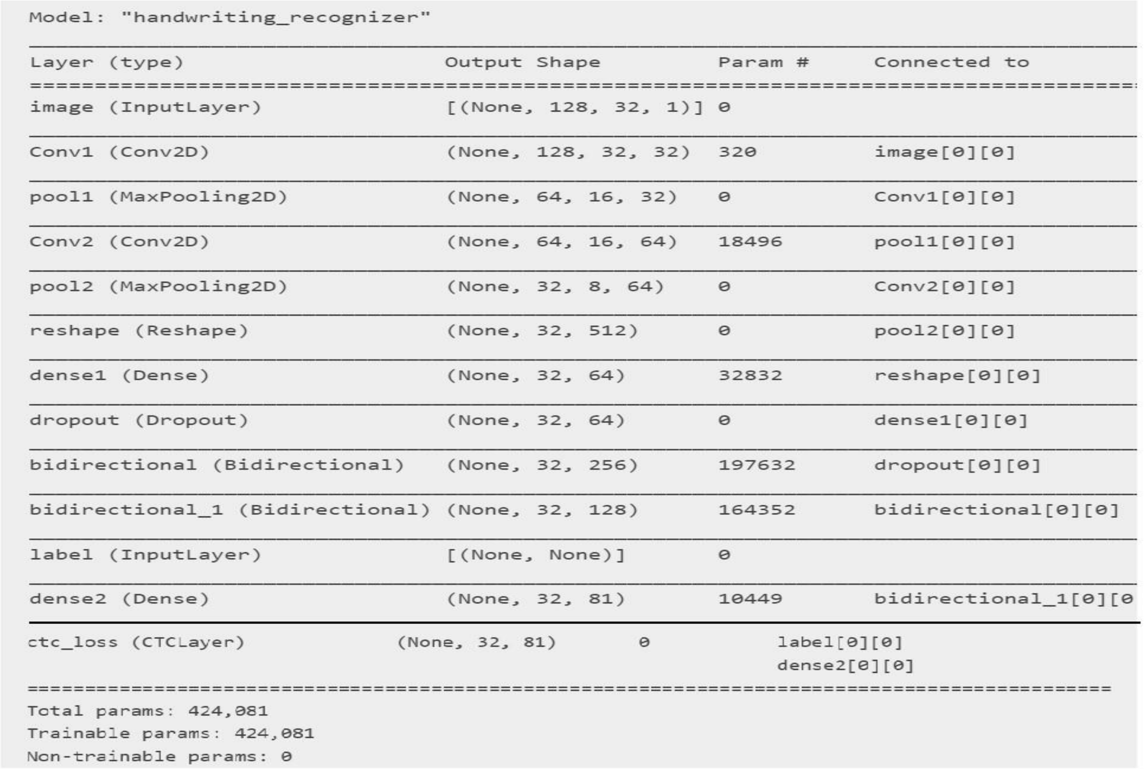


FIGURE 9: Bi-LSTM diagram

8.3.1 Model Building

Input image to the model is of size 128*32. Model consists of Convolutional layers, pooling layers with a stride = 2, dense layer, and at last RNN layers are also added along with the CTC layer as CTC loss is used as an endpoint layer. Down sampled feature maps are 4x smaller. No. of filters in the last layer is 64. Reshaping is done before passing the output to the RNN part of the model.

8.3.2 SUMMARY OF THE MODEL



```
Model: "handwriting_recognizer"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
image (InputLayer)              [(None, 128, 32, 1)] 0
__________________________________________________________________________________________________
Conv1 (Conv2D)                  (None, 128, 32, 32)  320         image[0][0]
__________________________________________________________________________________________________
pool1 (MaxPooling2D)            (None, 64, 16, 32)   0           Conv1[0][0]
__________________________________________________________________________________________________
Conv2 (Conv2D)                  (None, 64, 16, 64)   18496       pool1[0][0]
__________________________________________________________________________________________________
pool2 (MaxPooling2D)            (None, 32, 8, 64)    0           Conv2[0][0]
__________________________________________________________________________________________________
reshape (Reshape)               (None, 32, 512)      0           pool2[0][0]
__________________________________________________________________________________________________
dense1 (Dense)                  (None, 32, 64)       32832       reshape[0][0]
__________________________________________________________________________________________________
dropout (Dropout)               (None, 32, 64)       0           dense1[0][0]
__________________________________________________________________________________________________
bidirectional (Bidirectional)   (None, 32, 256)      197632      dropout[0][0]
__________________________________________________________________________________________________
bidirectional_1 (Bidirectional) (None, 32, 128)      164352      bidirectional[0][0]
__________________________________________________________________________________________________
label (InputLayer)              [(None, None)]       0
__________________________________________________________________________________________________
dense2 (Dense)                  (None, 32, 81)       10449       bidirectional_1[0][0
__________________________________________________________________________________________________
ctc_loss (CTCLayer)             (None, 32, 81)       0           label[0][0]
                                                                 dense2[0][0]
==================================================================================================
Total params: 424,081
Trainable params: 424,081
Non-trainable params: 0
```

8.3.3 EVALUATION METRIC

Edit distance is used as an evaluation metric. It is one of the most widely used metric for evaluating OCR models. At first segregating the validation images and their labels is done. Now a call back to monitor the edit distances is created. Single batch is created and converted its labels to sparse tensors. Predictions are also made and converted them to sparse tensors. Edit distance is computed individually and average is taken out. In each step edit distance is calculated and is returned.

9 CONCLUSION

This project will provide a great help to the blind people and gives voice output to them so that every time they need not require help of some other person to read any text for them. Even if the person does not know the braille they can easily understand the text. Before learning their tactile writing system i.e., braille script this protocol which is being developed will be of great use to them. Two models have been developed i.e., OpenCV and the other one is using the dataset. The dataset model can identify various handwritten texts easily. Using this prototype is very easy for them.

10 FUTURE SCOPE

A wearable camera need to the attached to the spectacles or a small device need to be developed for them so that by using the camera they can place the text that need to be recognized. After the recognition of the text, it is send as an voice output for them so for that external speakers are required which makes easy for them to hear the voice. So these two devices need to be developed with a minimal cost so that each and everyone could afford it easily. Coming to the second model just a sample model is built to identify handwritten text. In order to help them in their academics for studying any notes or textbook certain dataset containing those images need to be created so that the text can be recognized and can be read to them via voice so that they can read any content easily nd clearly. Even other models can be tried and checked or some alternative ways need to be identified to improve the performance of training the model and to get more accurate output than this.

11 REFERENCES

[1] Qiu, X., Chen, S., Li, R., Wang, D., & Lin, X. (2021). A Post-processing method for text detection based on geometric features. *IEEE Access*, *9*, 36620-36633.

[2] Naiemi, F., Ghods, V., & Khalesi, H. (2021). A novel pipeline framework for multi oriented scene text image detection and recognition. *Expert Systems with Applications*, *170*, 114549.

[3] Kantipudi, M. V. V., Kumar, S., & Kumar Jha, A. (2021). Scene Text Recognition Based on Bidirectional LSTM and Deep Neural Network. *Computational Intelligence and Neuroscience*, *2021*.

[4] Deng, G., Ming, Y., & Xue, J. H. (2021). RFRN: A recurrent feature refinement network for accurate and efficient scene text detection. *Neurocomputing*, *453*, 465-481.

[5] Wan, Q., Ji, H., & Shen, L. (2021). Self-attention based Text Knowledge Mining for Text Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 5983-5992).

[6] Al-Radhi, M. S., Csapó, T. G., & Németh, G. (2021). Noise and acoustic modeling with waveform generator in text-to-speech and neutral speech conversion. *Multimedia Tools and Applications*, *80*(2), 1969-1994**.**

[7] Harizi, R., Walha, R., Drira, F., & Zaied, M. (2021). Convolutional neural network with joint stepwise character/word modeling based system for scene text recognition. *Multimedia Tools and Applications*, 1-16.

[8] Chen, Y., Shu, H., Xu, W., Yang, Z., Hong, Z., & Dong, M. (2021). Transformer text recognition with deep learning algorithm. *Computer Communications*, *178*, 153-160.

[9] Long, S., He, X., & Yao, C. (2021). Scene text detection and recognition: The deep learning era. *International Journal of Computer Vision*, *129*(1), 161-184.

[10] Jayakumari, & Vyas. (2020). *Advances in Communication Systems and Networks*. Springer Singapore. Assistive technology to the blind(From page 309-320).

[11] Chu, J., Zhang, Y., Li, S., Leng, L., & Miao, J. (2020). Syncretic-NMS: A merging nonmaximum suppression algorithm for instance segmentation. *IEEE Access*, *8*, 114705114714.

[12] Huang, Y., Sun, Z., Jin, L., & Luo, C. (2020). EPAN: Effective parts attention network for scene text recognition. *Neurocomputing*, *376*, 202-213.

[13] Naiemi, F., Ghods, V., & Khalesi, H. (2020). Scene text detection using enhanced Extremal region and convolutional neural network. *Multimedia Tools and Applications*, *79*(37), 27137-27159.

[14] Geetha, M., Pooja, R. C., Swetha, J., Nivedha, N., & Daniya, T. (2020). Implementation of text recognition and text extraction on formatted bills using deep learning. *Int J Contrl Automat*, *13*(2), 646-651.

[15] Zuo, L. Q., Sun, H. M., Mao, Q. C., Qi, R., & Jia, R. S. (2019). Natural scene text recognition based on encoder-decoder framework. *IEEE Access*, *7*, 62616-62623.

[16] Mohanty, S., Dutta, T., & Gupta, H. P. (2018, August). An efficient system for hazy scene text detection using a deep CNN and patch-NMS. In *2018 24th International Conference on Pattern Recognition (ICPR)* (pp. 2588-2593). IEEE.

[17] Basheer, S., Anbarasi, M., Sakshi, D. G., & Vinoth Kumar, V. (2020). Efficient text summarization method for blind people using text mining techniques. *International Journal of Speech Technology*, *23*(4), 713-725.

[18] Zhang, C., Ding, W., Peng, G., Fu, F., & Wang, W. (2020). Street view text recognition with deep learning for urban scene understanding in intelligent transportation systems. *IEEE Transactions on Intelligent Transportation Systems*, *22*(7), 4727-4743.

12. BIBILIOGRAPHY

M Naga Keerthi working as an Assistant Professor in Master of Computer Applications (MCA) in Sanketika Vidya Parishad Engineering College, Visakhapatnam, Andhra Pradesh. With 7 years' experience in computer science, accredited by NAAC with her areas of interests in C, Java, Operating System, DBMS, Web Technologies, Software Engineering.

Singupuram Sravan Kumar is studying his 2nd year, Master of Computer Applications in Sanketika Vidya Parishad Engineering College, affiliated to Andhra University, accredited by NAAC. As a part of academic project, he chooses Text To Voice Conversion For Visually Impaired. A full-fledged project along with code has been submitted for Andhra University as a result of a desire to comprehend, in completion of his MCA.