Assisting partially blind people using Machine Learning

Soukhya Boreddy
Data Science
Stevens Institute of Technology
Hoboken, New Jersey
Sboreddy@stevens.edu

Vinoth Kumar kolluru
Data Science
Stevens Institute of Technology
Hoboken, New Jersey
Vkolluru@stevens.edu

Sathya Narayanan Sriram
Data Science
Stevens Institute of Technology
Hoboken, New Jersey
ssriram1@stevens.edu

***Abstract*—This project is proposed to help the visually impaired people using machine learning by converting the text in image into Voice. Visual impairment is a major disability which most of the visually challenged people are facing as the vision is the basis for most of the tasks, the automation of this system would be extremely advantageous and beneficial to visually challenged personalities.**

***Index Terms*—Segmentation, Text-Recognition, Pre and Post Processing, Neural Network, Sigmoid Function.**

I. Introduction

Big Data analytics, known in the business world for its valuable use in controlling, managing and contrasting large datasets that can be applied to benefit visually challenged people, Machine learning/ Deep learning model will be developed with huge data sets which consists of multiple handwritten documents and Word text documents where the model is trained with all essential data to convert all image text form to Audio. Automated tools for the service of visually impaired people, text to speech (or TTS) has emerged as a preferred tool for many technology service providers for improving customer service. In simple language, a TTS tool converts written text into natural speech that can be heard and understood by visually challenged persons. They can listen to the audio and understand the purpose of the document without seeing them using naked eye. Visual impairment is a major disability which most of the visually challenged people are facing as the vision is the basis for most of the tasks, the automation of this system would be extremely advantageous and beneficial to visually challenged personalities.

II. Description of Datasets

By analyzing the available health data set, and by considering the different forms of data (Text, image, various fonts of data), we will classify the data into testing, training data. By training, the data with appropriate machine learning algorithms will implement the project. As we were looking for the data which consists of all alphabets and digits we decided to use EMNIST dataset.The data Cleaning process is required to identify and remove errors and also to avoid duplication in data. Thus reliable data sets can be used as input to work on a solution. Data cleaning increases the quality of the training data for analytics and enables us to result in precise results or perfect decision making.

Image or Object Detection is a computer technology that processes the image and detects objects in it. People often confuse Image Detection with Image Classification. Although the difference is rather clear. If you need to classify image items, you use Classification. But if you need to locate them, for example, find out the number of objects in the picture, you should use Image Detection. Image recognition is the ability of Artificial Intelligence to detect the object, classify, and recognize it. The last step is close to the human level of image processing. The best example of image recognition solutions is the face recognition – say, to unblock your smartphone, you have to let it scan your face. So first of all, the system has to detect the face, then classify it as a human face and only then decide if it belongs to the owner of the smartphone. As you can see, it is a rather complicated process. We implemented an image recognition concept to recognize handwritten letters to text, which is part of our project, and that is converted audio to help partially blind people.

A. Abbreviations and Acronyms :

- ASCII - American Standard Code for Information Interchange
- MNIST - Modified National Institute of Standards and Technology database
- EMNIST – Extended Modified National Institute of Standards and Technology database
- OCR – Optimal Character recognition
- TTS – Text to Speech
- CSV – A Comma-separated values file
- JPEG - Joint Photographic Expert Group
- MP3 - MPEG Audio Layer-3

B. List of Packages

- Pandas
- Matplotlib
- Numpy
- Utils
- Tensor flow
- Tensorflow.keras
- PIL (Python Imaging Library)

- CSV (to import any CSV format data)
- Sys (to import System Specific parameters and functions)

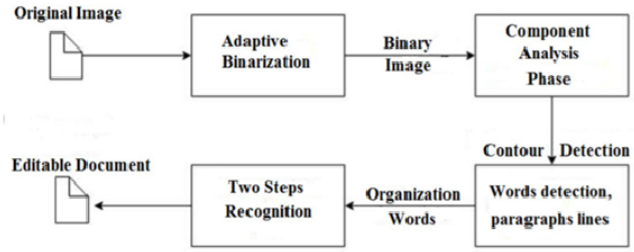

Fig. 1. Example of a figure caption.

1) Model Creation: We have developed the model which is trained with all essential data to convert all images, text forms to audio. This can be done using state of the art text audio translation techniques. Visually impaired people can listen to the sound and understand the purpose of the document without seeing them using the naked eye. Recognition of massive set of handwritten alphabets and characters there are classification technologies or methods which are based on pattern matching can be used, and they belong to a statistical technique where the input pattern is compared with an already stored model and classified into class of reference model which has the close match or which has less or minor distance concerning already available input pattern.Text detection is done using Optical Character Recognition (OCR). Optical character recognition OCR can transform a 2-D picture or Image file to a text file. There are several stages to recognize the image file; They are pre-processing of the image file, then the text localization then Alphabet/character segmentation and recognition and final post-processing. Thus characters are segmented and recognized using OCR technology.

(Thus characters are segmented and recognized using OCR technology, which can be implemented using high-level language Python it's main aim is to identify and capture all the unique alphabets and words from an image file that is in written text characters. OCR engines can be built using Machine learning /deep learning models. Using this deep technology model and large/massive datasets publicly available, models achieve state of the art accuracies on given tasks. Neural machine translation techniques using neural network/ machine learning and deep learning models can be developed, whereas encoders and decoders predict the words that should appear in the image file)

III. Project Implementation:

In the Implementation phase Following we have converted input images into csv and feed them for the model to generate the required output of the image converted into text format.The conversion of image to CSV is done by converting the image into 2D numpy arrays and saving them as txt file with .CSV extension. This CSV file is feeded into the model and the output, a text file is generated. For image to Audio conversion

```
def correct_percentage_on_test(predictions):
    correct = 0
    incorrect = 0

    for x in range(len(test_y)):
        if np.argmax(predictions[x]) == np.argmax(test_y[x]):
            correct += 1
        else:
            incorrect += 1

    # print(correct, incorrect)
    return (correct / (correct + incorrect)) * 100
```

```
correct_percentage_on_test(final_model_predictions)
```

```
86.89893617021276
```

Fig. 2. Model Accuracy: 86.89.

the output generated in the first generated model is fed into "tesseract", which is an optical character recognition engine. After recognizing the text, we use GTTS package, which is "Google Text-to-Speech", a python library.

```
import pytesseract
from gtts import gTTS
from PIL import Image

picture = Image.open('pic.jpg')
mytext = pytesseract.image_to_string(picture)
language = 'en'
myobject = gTTS(text=mytext, lang=language, slow=False)
myobject.save("pic.mp3")
```

Fig. 3. Text to Image Conversion

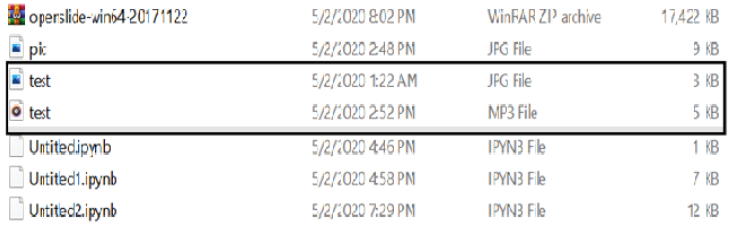

Fig. 4. Text converted to Image

From the above screenshot we can say the desired output is obtained i.e., test.jpg file has been converted into test.Mp3 file. Thus, the developed model is extremely advantageous and beneficial to partially visually challenged personalities. In future work, we would extend the dataset and transform all other languages text into audio form

IV. Rectified Linear Unit

ReLU: The Rectified Linear Unit has become very popular in the last few years. It computes the function f(x)=max(0,x).In other words, the activation is simply thresholded at zero.

There are several pros and cons to using the ReLUs: It was found to greatly accelerate. the convergence of stochastic

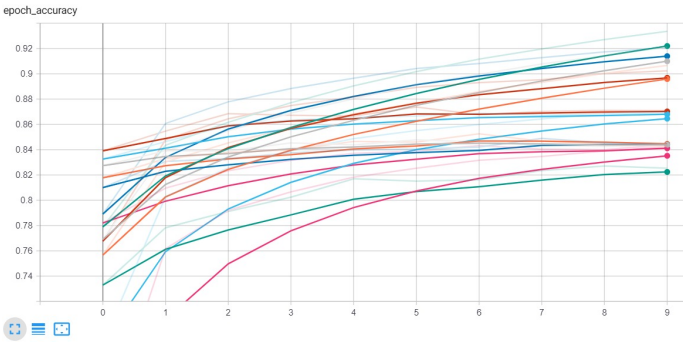

Fig. 5. Epoch Accuracy

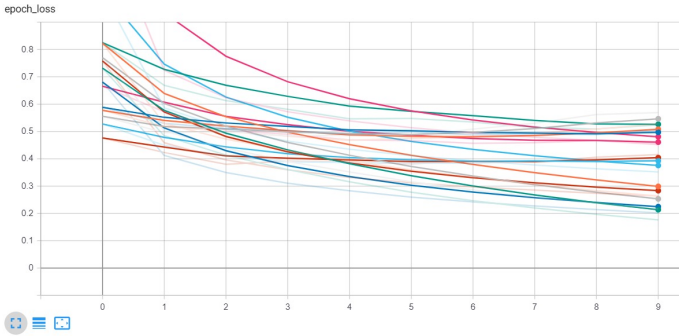

Fig. 6. Epoch Loss

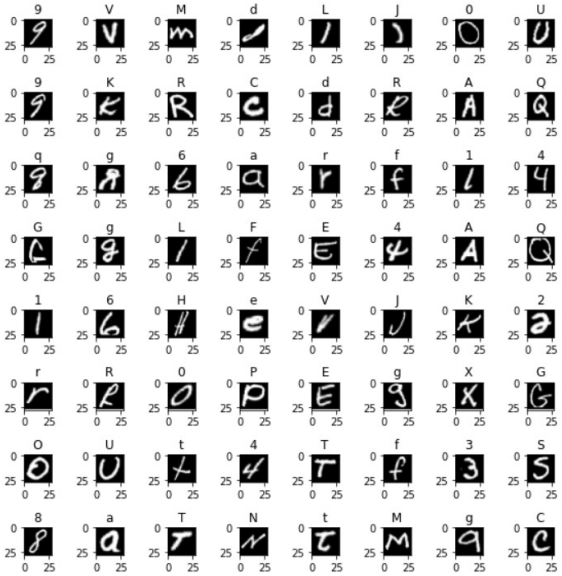

Fig. 7. Sample Output of Handwritten to Text

gradient descent compared to the sigmoid/tanh functions.It is argued that this is due to its linear, non-saturating form. Compared to tanh/sigmoid neurons that involve expensive operations (exponentials, etc.), the ReLU can be implemented by simply thresholding a matrix of activations at zero.(-) Unfortunately, ReLU units can be fragile during training and can "die". For example, a large gradient flowing through a ReLU neuron could cause the weights to update in such a way that the neuron will never activate on any datapoint again. If this happens, then the gradient flowing through the unit will forever be zero from that point on. That is, the ReLU units can irreversibly die during training since they can get knocked off the data manifold. For example, you may find that as much as 40 percent of your network can be "dead" (i.e. neurons that never activate across the entire training dataset) if the learning rate is set too high. With a proper setting of the learning rate this is less frequently an issue.

V. Sigmoid Function

The sigmoid nonlinearity has the mathematical form

Sigmoid(x)=1/(1+ex)

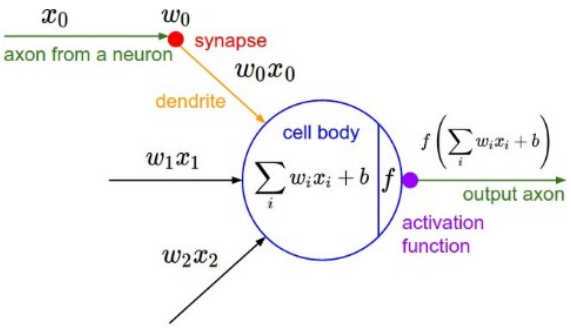

Fig. 8. Sigmoid Function

It takes a real-valued number and "squashes" it into a range between 0 and 1. In particular, large negative numbers become 0 and large positive numbers become 1. The sigmoid function has seen frequent use historically since it has a nice interpretation as the firing rate of a neuron: from not firing at all (0) to fully-saturated firing at an assumed maximum frequency

VI. Convolution Neural Network

The dataset of handwritten digits developed by the Mixed National Institute of Standards and Technology (MNIST) is the most well-suited dataset for benchmarking the performance of Convolutional Neural Networks (CNNs). However, the MNIST dataset is overused, easily solved, and not representative of up to date computer vision projects and tasks. For these reasons, the next-generation Fashion-MNIST dataset is chosen as the vehicle with which to test the results of information that was trained on the validation accuracy of a CNN. Accomplishing this optimization problem necessitates subsetting the info into training, validation, and test sets. The shuffled and split data is used to coach the Convolution

Neural Network. The validation set is employed to tune the hyperparameters of the model. The CNN has various componential layers that are employed to hunt out the minimum epochs for training, also as minimum error and good accuracy. The foremost components of CNN are the convolutional, pooling, and dense layers. The convolutional layer extracts feature selection and learning from given clothing images. The pooling layer reduces dimensionality to forestall overfitting and increase computational performance. The dense layers compute the weights for each node of the labels to perform image classification. Optimization of the training data size is realized through two metrics. The model checkpoint provides a metric wherein a loss function is monitored throughout epoch training. If the given loss function doesn't improve, then the model is saved thanks to the simplest model. The primary stopping method halts training if the given loss function being monitored has not improved from the previous epoch. Utilizing both the model checkpoint and also the initial stopping method maximizes accuracy while simultaneously minimizing training time for CNN.

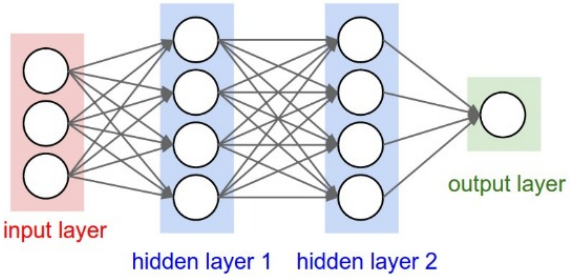

Fig. 9. Neural Network

We will evaluate our results based mostly on the accuracy of character recognition for each of the different algorithms we implement. Qualitatively, this will translate these. For the convolutional neural network architectures that we are experimenting with, we will also take into account the performance as a possible factor given the time period of our project.

TensorFlow can help you build neural network models to automatically recognize images. These are typically Convolutional Neural Networks (CNN). There are two approaches to TensorFlow image recognition:

- Classification—train the CNN to recognize categories like cats, dogs, cars, or anything else. The system classifies the image as a whole, based on these categories.
- Object Detection—more powerful than classification, it can detect multiple objects in the same image.

Neural Networks are essentially mathematical models to solve an optimization problem. They are made of neurons, the basic computation unit of neural networks. A neuron takes an input(say x), does some computation on it(say: multiply it with a variable w and adds another variable b) to produce a value (say; z= wx+b). This value is passed to a nonlinear function called activation function(f) to produce the final output(activation) of a neuron. There are many kinds of activation functions. One of the popular activation functions is Sigmoid.

Fig. 10. Neural Network with 3 Hidden Layers

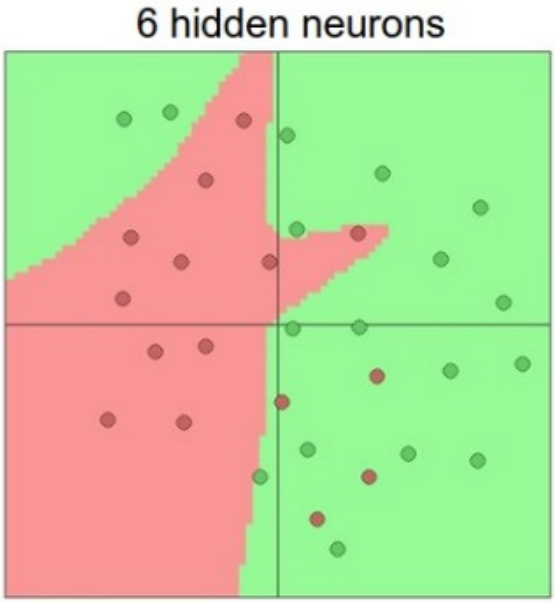

Fig. 11. Neural Network with 6 Hidden Layers

In the diagram above, we can see that Neural Networks with more neurons can express more complicated functions. However, this is both a blessing (since we can learn to classify more complicated data) and a curse (since it is easier to overfit the training data). Overfitting occurs when a model with high capacity fits the noise in the data instead of the (assumed) underlying relationship. For example, the model with 20 hidden neurons fits all the training data but at the

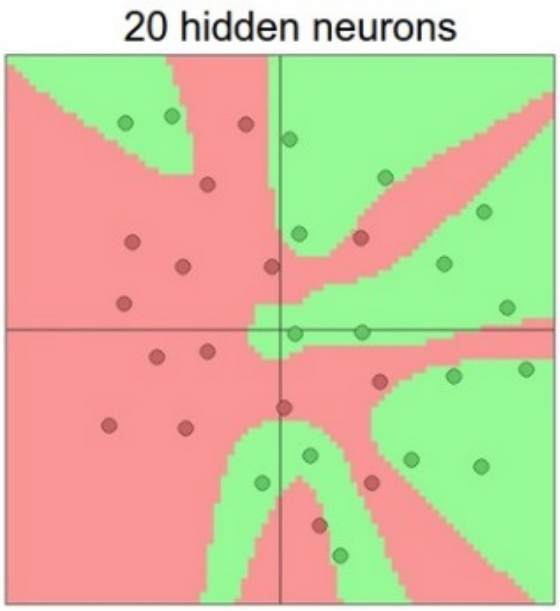

Fig. 12. Neural Network with 20 Hidden layers

cost of segmenting the space into many disjoint red and green decision regions. The model with 3 hidden neurons only has the representational power to classify the data in broad strokes. It models the data as two blobs and interprets the few red points inside the green cluster as outliers (noise). In practice, this could lead to better generalization on the test set.

The neuron which uses sigmoid function as an activation function will be called Sigmoid neuron. Depending on the activation functions, neurons are named and there are many kinds of them like RELU, TanH etc(remember this). One neuron can be connected to multiple neurons, like this: The weights are the property of the connection, i.e. each connection has a different weight value while bias is the property of the neuron. This is the complete picture of a sigmoid neuron which produces output y.

If you stack neurons in a single line, it's called a layer; which is the next building block of neural networks.The neurons in 1 layer of the network where input data is passed to the network. Similarly, the last layer is called the output layer. The layers in between the input and output layer are called hidden layers. The networks which have many hidden layers tend to be more accurate and are called deep networks and hence machine learning algorithms which use these deep networks are called deep learning.

VII. Conclusion

Considering that Image Detection, Recognition, and Classification technologies are only in their early stages, we can expect great things to be happening in the near future. Imagine a world where computers can process visual content better than humans.The OCR systems are based on three main rules–integrity, purposefulness, and adaptability. First, the observed object has always to be considered as one entity comprising many interrelated parts. Interpretation of data must always serve some purpose. And finally, the OCR program has to be capable of self-learning. How easy our lives would be when AI could find our keys for us, and we would not need to spend precious minutes on a distressing search so we came up with our project assisting partially blind people by using image detection and recognition to detect handwritten text and convert the text into audio form which is audible to disable persons. This project was carried out in two stages. First was converting the handwritten data into a text document. And the second stage was further converting that Text document into an audio file. The complete project was carried out with an accuracy rate of 86.89. This project focuses on converting English language data into audio format. The future work of this project involves converting handwritten documents in various languages into text and further into audio format.

Acknowledgment

We wish to express our deep gratitude and sincere thanks to my Professor Mr. Shucheng Yu for his encouragement and provided required information for this research project. We sincerely appreciate his generosity by taking us into his fold for which we shall remain indebted to him. We take this opportunity to express our deep sense of gratitude to his invaluable guidance, ongoing encouragement, enormous motivation, which has sustained our efforts at all the stages of project development.

References

[1] V. Wan, Y. Agiomyrgiannakis, H. Silen, and J. Vit, "Google's next-generation realtime unit selection synthesizer using sequence-to-sequence LSTM-based autoencoders," Proc. Interspeech, 2017, pp. 1143–1147, 2017.

[2] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, et al., "Tacotron: Towards end-to-end speech synthesis," Proc. Interspeech, 2017, 2017.

[3] T. Nose, J. Yamagishi, T. Masuko, and T. Kobayashi, "A style control technique for HMM-based expressive synthesis," Information and Systems vol. 90, no. 9, pp. 1406– 1413, 2007.

[4] Baldi, P. and Brunak, S. (2002). Bioinformatics: A Machine Learning Approach. Cambridge, MA: MIT Press. This book offers a good coverage of machine learning approaches - especially neural networks and hidden Markov models in bioinformatics.

[5] Jordan, M. (2003). Probabilistic Graphical Models. Professor Jordan has kindly shared a pre-publication draft. This text has an excellent coverage of generative and discriminative probabilistic models for classification.

[6] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86, 2278–2324