Detecting Braille Images into Braille codes

Anonymous submission

Paper ID

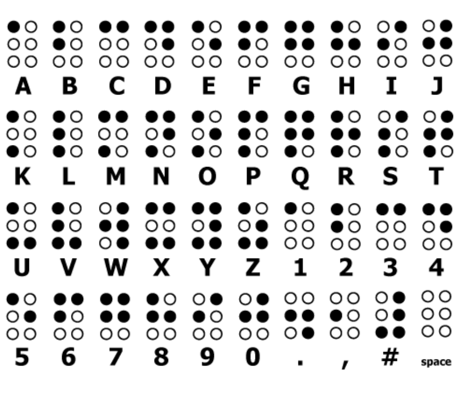
Abstract

This report gives a brief description of our analysis and exploration on how braille in different languages can be made accessible for the visually impaired and the visually acute. The visually impaired rely on braille to communicate and read information. However, not everyone is trained to understand braille which results in a gap in communication while sharing information and ideas. Moreover, there are over 133 braille languages. This makes the process of learning and understanding braille more complex. A literature review was also conducted to analyse the existing state of art to make braille more accessible. This pointed towards the language centrism towards only English braille. Thus, we have elaborated on ways through which translation of braille can be made more universal despite its varied meanings as per the language context. An elaboration on the contribution of this work and the challenges that can be faced while implementing it has also been done. It concludes with a summary of the report.

1. Introduction

Braille is a tactile code used by the visually impaired people to read and write text. It is touch dependent. A binary string of 6 letters is represented in form of a 2*3 matrix with the 1's and 0's represented with raised dots. Currently, there are 133 braille languages.[4]

Figure 1. English Braille Key



1.1. Problem Statement

Figure 2. Hindi Braille Key

texts accessible to the visually impaired. More so, currently the number of braille languages is so high that it is difficult to keep a standard rule of understanding. Therefore, the several braille languages and the lack of training in braille for the majority of people makes communication difficult and inefficient between the visually impaired and visually acute people.

This project aims at trying to generate a standard binary coded string from the image of a given braille pattern. Following this, these binary coded strings can be mapped to their corresponding translation in any particular language. This shall simplify the complexity caused due to the many braille languages which essentially follow the same algorithm of translation. Moreover, this shall help create a platform that can make understanding braille for all people, whether or not trained in braille, easier and makes training people in Braille much easier.

1.2. Challenges

The following are the present challenges in catering to the current disadvantageous situation of the visually impaired.

1. Acquiring the dataset of all languages in braille will be difficult to execute.

2. The dataset of all languages will be very lengthy and repetitive since all combinations will have multiple interpretations in different languages.

3. The computation and assembly of redundant data meaning different things will be complicated. There is no consistent or universal approach to solving it.

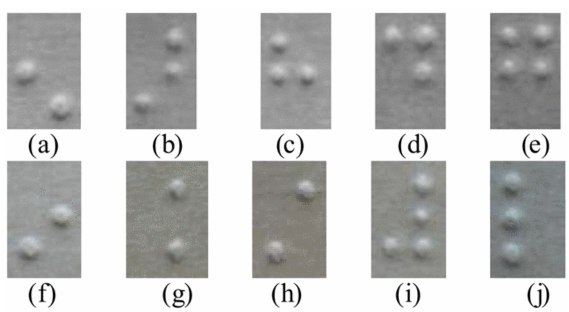
1.3. Motivation

of the visually impaired and the visually acute. Hence, as a group of students who wish to take a concern, we decided to go ahead with exploring braille and how computer vision can make it more accessible. We believe, through this project we will be able to add more to the current disadvantageous context of the visually impaired to make it better.

1.4. Contribution

Through our explorations of the current state of art, we realised that the current methods of braille conversion to words are not inclusive of all languages. Language plays a very essential role in communication. The current models and interventions deal more with the English language Braille. Through this project, we wish to make an attempt to cater to many more braille languages so that braille becomes more accessible and efficient. This could also serve as a training tool for braille learners.Through this project, we will also bring the visually impaired on the same page as the visually acute people by mending the gap between their textual presentation and understanding of information.

Figure 3. Embossed Braille



1.5. Summary

This project aims at catering to the gap between information presentation and retrieval between the visually impaired and visually acute people. Currently, the visually impaired use braille to read and write information. However, due to the lack of training in braille for the visually acute people, the communication can be inefficient and ineffective. This results in a disadvantageous situation for the visually impaired. More so, there are over 133 braille languages. This enhances the aforementioned problem area. Braille is not commonly taught and the number of different braille languages makes the process of making it universally accessible more complicated. Hence, we wish to cater to this gap and complexity by focusing on the basic functionality of braille and making it language agnostic. By developing a dataset of the various combinations of possible braille patterns and generating a 6 digit binary string from the recognised pattern, we can map the generated string to its corresponding meaning in the language selected. This shall make the conversion of braille universal for all languages and reduce the complexity caused due to the many different braille languages while making it accessible for all.

2. Literature Survey

2.1. Commercial Text and Braille Converters

We have found quite a few text to braille converters but very few actually convert Braille to text.

- **Text to Braille converter**: Displays the Braille as the user types characters. This converter is OS independent and the language used is java. It only concentrates on conversion from English to Braille
- **Win Braille**: Win Braille includes standard Windows image control and the unique feature to convert images to tactile graphic format online and can be used by someone with little to no knowledge on Braille.
- **Braille Master**: Braille Master package comes with both Windows and DOS versions. A large print facility suitable for partially sighted persons is also included in this package.
- **Cipher Braille Translator**: Cipher is a text to Braille program that converts text documents into a format suitable for producing Braille documents, through the use of a Braille printer. The user can edit, save, use style templates and enable translation rules.
- **Supernova**: Supernova is a window-based magnifier, screen reader and a Braille system that supports the conversion of text to speech, Braille displays and note-takers.

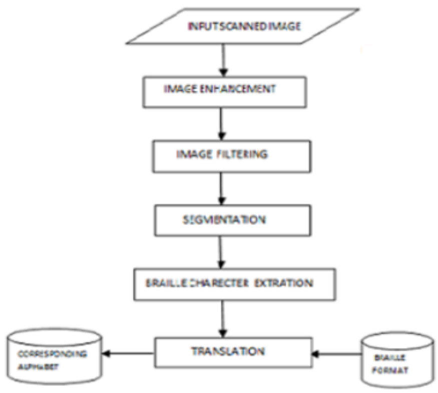
2.2. Educational Research Papers in the Field

The following are the research papers we could find relating to our topic.

1. **Conversion of Braille to Text in English, Hindi and Tamil Languages**

   This[3] paper explores how to convert braille text to English, Hindi and Telugu text. Hence the different mapping of English, Hindi and Tamil are all considered. This paper does this without the use of Deep Learning through only enhancing and converting the dots into data using prepossessing. This is done through a process of Image Enhancement followed by Image filtering and then Segmenting the Braille Cells finished by Extraction of Text from pattern vector.

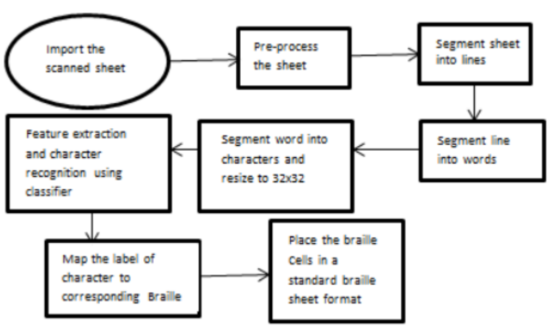
Figure 4. Block diagram for technique for [3]



2. **Braille Transliteration Of Hindi Handwritten Texts Using Machine Learning For Character Recognition**

Here[1] the researchers have tried to understand handwritten Hindi text and subsequently convert it to Braille through once again segmenting the image into lines and subsequently words and characters. Then they use a feature extraction and character recognition to identify the meaning of each letter. The phonetic equivalents from the English Braille are used to represent Indian language texts. This also states how Hindi Braille sometimes uses multiple characters for one letter and this may cause problems.

Figure 5. Flow chart of the algorithm in [1]



3. **A Deep Learning Method for Braille Recognition**

This[2] is an exploration on how we can convert braille Image to its corresponding text through the use of deep learning. It uses Stacked Denoising Auto Encoder (SDAE) to solve the problems of automatic feature extraction and dimension reduction in Braille recognition. This states that deep learning techniques are suitable for the recognition of Braille, since they could automatically extract relational features without interference of subjective ideas from human experience and meanwhile boost recognition accuracy in the premise of reduced digital image processing. One problem it did state was that they did not consider double sided Braille pages.

3. Methodology

For binary based braille classification, we take use of a dataset containg all possible braille letters. The dataset contains 1,872 images of braille characters, there are 63 classes of braille characters corresponding to each of the possible numbers that can be made. The sizes of the images are 288px x 432px. The characters are printed having various colors, brightness, angles and blurs.

For preprocessing, OpenCV was used to import the images, and resize them into 32px x 32px images, then normalize the pixels to 0-1 (pixel value divided by 255). The processed data will be randomly shuffled and separated into training set, test set, and validation set.

Here we chose ResNet-152v2 model for the trials, changing the number of trainable layers foe every instance. The sizes of the images are quite small; so, we test whether a more deeply trained network can perform better than a shallower one. We flattened the incoming data and ran it through two dense layers with a 0.3 dropout in between each. Finally, we set 63 outputs classes for the last linear fully connected layer.

The experiments were conducted using a laptop, the models were trained by GPU for time saving and efficiency. For meta-parameters, we set the learning rate as 0.001 initially, which reduced by a factor of 0.4 upon reaching a plateau, each model takes 30 epochs of iterations. The loss function for model convergence is cross entropy loss function. The optimizer for models is Adam optimization algorithm.

For evaluating the different models, training and validation losses are used for evaluating the convergence of the models during training and validation. Each model's time consumption is recorded for the purpose of comparisons. Precision, Recall, and F1 score are used as the metrics to show how models work.

4. Experiments

When only training the top 5 layers of ResNet we observe a max accuracy of 84%, this took a total of 13 minutes. It had a precision of 0.92, recall of 0.16 and F1 score of 0.03.

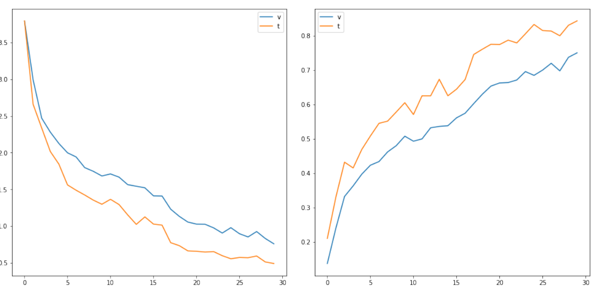


Figure 6. Loss and Accuracy for ResNet trained on 5 layers

When only training the top 10 layers of ResNet we observe a max accuracy of 90%, this took a total of 14 minutes. It had a precision of 0.94, recall of 0.016 and F1 score of 0.03.

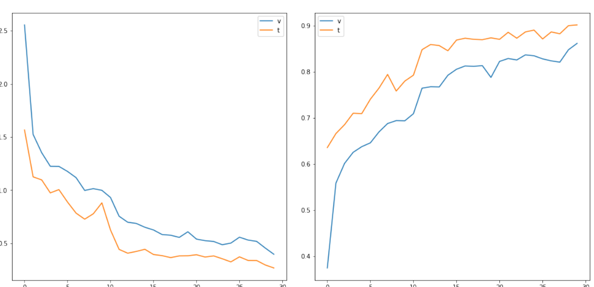


Figure 7. Loss and Accuracy for ResNet trained on 10 layers

When only training the top 15 layers of ResNet we observe a max accuracy of 94%, this took a total of 24 minutes. It had a precision of 0.97, recall of 0.16 and F1 score of 0.03.

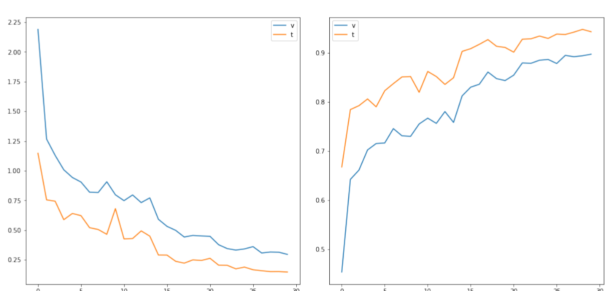


Figure 8. Loss and Accuracy for ResNet trained on 15 layers

We can also compare the different models based on these parameters. Table 1 shows the accuracy of three models. 15 trainable layers model(M15) has the highest accuracy 94%, and accuracy of 10 trainable layers(M10) has slightly down to 90%, and 5 trainable layers(M5) has the lowest accuracy of 84%. As the model becomes deeper, it takes more time for training, 10 minutes for training 5 trainable layers, 14 minutes for 10 trainable layers, and 24 minutes for 15 trainable layers.

Table 1. Comparison of Accuracy, Error and time take for each model

Trainable Layers	Accuracy	Error	Time Taken
5	84	0.44	10
10	90	0.27	14
15	94	0.14	23

In table 2 we can observe the precision, recall and F1 scores of all the models. We can observe that the Precision recall and F1 scores of M15 are higher as compared to M10 which themselves are higher than M5.

Table 2. Comparison of Precision, Recall and F1 Score for each model

Trainable Layers	Precision	Recall	F1 Score
5	0.61	0.95	0.74
10	0.65	0.95	0.77
15	0.68	0.95	0.79

5. Conclusion

Unlike traditional techniques using image processing, deep neural networks make use of noises in the images and that's how classifiers are trained. This allows the model to have the ability to detect and then classify braille characters with decent accuracy.

ResNet allows the images to rotate by angles, and then classify the characters into the right corresponding letters, this takes into consideration the various discrepancies in the data.

As observed, the design of the residual block in ResNet enables the losses to be controlled in deeper networks as can be seen by increasing the layers. The overall accuracies lie between 80 % -95 %. The highest accuracy is reported at ResNet with 15 layers.

Therefore, making use of ResNet model, we can build a model that obtains integers corresponding to an image input of braille. This integer can further be simply mapped to a letter of a chosen language. This helps us accomplish the objective stated previously by making braille more accessible for both visually impaired and acute users.

References

[1] Vinod Jha and K Parvathi. "Braille Transliteration of hindi handwritten texts using machine learning for character recognition". In: *Int J Sci Technol Res* 8.10 (2019), pp. 1188–1193.

[2] Ting Li, Xiaoqin Zeng, and Shoujing Xu. "A deep learning method for Braille recognition". In: *2014 International Conference on Computational Intelligence and Communication Networks*. IEEE. 2014, pp. 1092–1095.

[3] S Padmavathi, S Sphoorthy Reddy, D Meenakshy, et al. "Conversion of braille to text in English, Hindi and Tamil languages". In: *arXiv preprint arXiv:1307.2997* (2013).

[4] Wikipedia contributors. *Braille*. [Online; accessed 25-march-2022]. 2022. URL: https://en.wikipedia.org/wiki/Braille.