Braille Character Recognition Using Deep Learning Techniques

AUTHOR 1: Ms.Asiya Farseen
Dept of CSE,
University Visvesvaraya College of engineering.

AUTHOR 2: Dr Kumaraswamy S,
Assistant Professor,
Dept of CSE,
University Visvesvaraya College of engineering .

Abstract: Braille Character Recognition (BCR) is a technique for finding and identifying Braille documents that are saved in images like jpeg, jpg, tiff, or gif files, and then converting the text into a machine-readable form like a text file. BCR transforms an image's pixel representation into its corresponding character representation. Numerous advantages of braille recognition make working at institutions and schools for the blind and visually impaired easier on a daily basis. Since it would take a very long time to look through, read, or rewrite all of the paper forms and documents that are used in many schools to educate blind pupils by hand, it makes sense to try to automate this process. OCR is a technology that transforms scanned, printed, and handwritten image files into legible, editable text documents. The OCR system converted the file from the image it got by comparing the characters. using the collection of OCR-stored databases. Character recognition is being employed in many significant applications, such as speed track, which the government has used to monitor speeding vehicles. It is possible for blind persons to read text using character recognition. It has given the community's blind members new hope.

1 INTRODUCTION

The embossed Braille alphabet, developed in 1824, has long been the writing and reading system of choice for blind people. Due to recent technological advancements, blind people today have a wide range of new alternatives for information transmission and receiving. However, their primary forms of communication remain writing and reading Braille. But Braille is frequently employed to make it easier for sighted and blind people to communicate with one another. Sighted teachers occasionally encounter this issue, especially when working with blind students and navigating Braille textbooks and student work.

There are 63 possible characters that can be encoded in a Braille text because each letter or other character has many bulging points (1 to 6) that are arranged in a 2x3 grid. In comparison to two columns of points in a single sign, the spacing between consecutive symbols is a little bit greater. Every Braille document has the same character height, character width, character spacing, line spacing, and character placement. The majority of the currently used Braille recognition techniques heavily rely on this fact. This severely restricts their usefulness, though. For these algorithms to assure precise alignment of the Braille page image, either a scanner or particular photography settings are needed.

Louis Braille created the Braille language, also known as night writing, which he created exclusively for persons who are blind or visually challenged in order to teach them [4]. There are six dots per line in braille. The visually challenged may read and write a variety of characters using these diverse dot patterns. Previously, a slate and stylus were used to read and write Braille. Modern society has access to cutting-edge technologies that improve people's quality of life. More people utilise touchscreen devices, and between 2009 and 2014, the use of mobile screen readers increased dramatically [5].

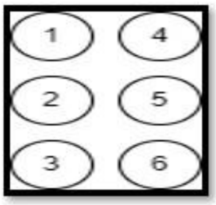

FIGURE 1. Braille Cell

2 Related Work

Convolutional neural networks [3] were presented first. ConvNet and a gradient-based learning algorithm are combined in the well-known model LeNet-5 to recognise documents. Gradient-based learning techniques are excellent for categorising handwritten characters because they create a decision surface for spotting highdimensional patterns. One flattening convolutional layer, two sets of average pooling layers, and two sets of convolutional layers make up the LeNet-5 architecture. The network's classification section is made up of one softmax classifier, two fully connected layers, and two. A typical ConvNet is made up of the three core operations of convolution, activation, and pooling. ConvNet requires an image's width, height, and depth as three input parameters. An image from the CIFAR-10 dataset, for instance, has a depth of 3 and a size of 32'32.

ConvNet models have been created in large numbers over the past ten years, including AlexNet, VGGNet, GoogLeNet, ResNet, and others. To support different classification types, each model includes a number of iterations and structures. Designing ResNet models has been proposed [8]. A convergence deterioration problem arises as deeper neural networks evolve more quickly. As a network's depth rises, the model's accuracy rapidly degrades, although overfitting is not to blame for this decline. The model's training error may increase when the network's layers are increased. The outcomes demonstrate that VGG-18 completed training in 5 minutes with an accuracy of 80% , While VGG-34 trained in 8 minutes and achieved an accuracy of 72%. VGG-18 (18 layers) and VGG-34 successfully met the CIFAR-10 dataset's classification accuracy requirements (34 layers). ResNet was therefore used to tackle this issue.

Thresholding is the most basic technique for finding points. A dynamic local threshold is utilised by Zhang and Yoshino [4]. In order to recognise dots on double-sided braille and differentiate between the front side and reverse side dots, Antonacopoulos and Bridson [5] segmented images into portions of three classes: bright, dark, and background. To achieve this, they employ static thresholds based on the typical light level in the area. The SVM and the Haar detector are both used by Renqiang Li et al. Morgavi and Morando [7] employ a straightforward neural network to locate objects [6]. Venugopal-Wairagade [8] completes circle detection using the Hough transform. HOG and SVM were used by Perera and Wanniarachchi [9], RLi et al. [6], and R. Li et al. [10] for the initial dot detection and HOG, LBP, and SVM for the second dot detection after grid restoration.

The following steps involve restoring the grid to which the points are attached and, if necessary, making up for the sheet's rotation. You can do this by applying linear regression [9], the Hough Transform [5], or [11] while rotating the image step-by-step and analysing the coordinates distribution density. Points are occasionally searched again using knowledge of their probable placements after the image has been de-skewn [5], [10].

Despite the fact that some techniques ([10] and others) call for grid deformation, which implies a change Based on the pitch between individual lines, it is presumed that the grid lines are parallel and straight over the entire sheet. The works listed above often use images captured by scanners since their fundamental method of operation involves clipping Braille points to the grid. The grid lines in this instance can be converted into vertical and horizontal ones simply by de-skewing. Few articles ([4], [8]) clearly state the goal of OBR on smartphone images, although theTechniques they outline still need a sheet with a rectangular grid on it. Convolutional neural networks (CNN) have recently achieved great progress in the field of photo identification, despite deep learning and neural networks being sparingly utilised for object-based recognition (OBR). Only a small number of research that have been published have used fully coupled neural networks. Morgavi, Morando [7], Ting Li et al. [12], and others used the picture segmentation points to create their models and Subur et al. [13] employ basic neural networks to calculate the symbol value. When reading braille on both sides, Kawabe and others [11] utilise it to separate front from back points. At CVPRW 2020, R. Li et al. presented their segmentation neural network research using a modified UNet architecture. The Braille characters were identified by determining the regions that the Braille characters filled using a neural network. The locations of individual symbols must be determined in a subsequent post-processing stage using the segmentation findings.

To complete our thesis, we looked at multiple papers produced by numerous writers that were pertinent to the subject we were focusing on. We found these articles to be really helpful. Using Bangla handwriting as the basis for character recognition has led to many noteworthy developments in the field. A. Roy Bhattacharya, one of the creators of the Bangla Handwritten Character and Digit Classification U.Pal, accomplished some of the best work. They created the foundation for several significant contributions to this field of study.

Pal et alstudy's [4] employed topological and statistical properties of numerals along with qualities that were taken from the concept of water overflow from reservoirs in order to construct a technique. On the recognition accuracy test, Pal used the technique on a range of people and received a score of 91.98. He did not, however, particularly address certain crucial elements that are crucial for real-world applications, such as response time and recognition reliability. Any character recognition research must take dependability into account because it shows how the error rate and recognition rate are related.

Das et al.[6] have performed comparable work on the recognition of Bangla characters. By progressing from the most common to the least common characters and accounting for the inherent complexity of the Optical Character Recognition for Bangla Character Recognition, this study provided a method for identifying compound character classes. By doing this, they created a framework for progressively learning more classes of compound characters, in addition to Basic characters, from those that occur more frequently to those that occur less frequently. After three-fold cross-validating the data, the experiment revealed an average recognition rate of 79.25, with space for expansion and future growth.

We read numerous papers and research about the braille during the course of our investigation, which helped to support our theory. Hossain et al. completed a substantial piece of work in this area. [2] In the experiment described in this article, a DFA (Determination Finite Automaton) design for a Bangla to braille translator machine was proven, and an expression was created using a structured and state elimination approach. Our understanding of braille was significantly enhanced by Anupam's additional explanation. [9] According to his article, the 6-dot braille was laborious and slow, therefore he created an 8-dot class value braille pattern for English characters that could represent up to 256 different symbols.

Finding Braille points, restoring the character grid, adjusting for image rotation, grouping the points into characters, and lastly decoding the characters are the essential processes in optical braille recognition. All algorithms are taken into account using this methodology in the survey papers [2], [3]. Thresholding is the most basic technique for finding points. A dynamic local threshold is utilised by Zhang and Yoshino [4].Antonacopoulos and Bridson's [5] method of detecting dots on double-sided braille and differentiating between the front side and reverse side dots segments images into areas of three classes: bright, dark, and background. In order to do this, they use static thresholds that are based on the local lighting conditions. Perera and Wanniarachchi [9] employ par HOG and SVM for initial dot detection and post-grid restoration final dot detection. Adaboost, the Haar detector, R. Li, et al., and [6] The Haar detector and Adaboost are both in [10]. The following steps involve restoring the grid to which the points are attached and, if necessary, making up for the sheet's rotation. You can do this by applying linear regression [9], the Hough Transform [5], or [11] while rotating the image step-by-step and analysing the coordinates distribution density.Points are periodically re-searched using knowledge of their expected locations after the image has been de-skewn [5], [10].

The aforementioned works usually make use of photos from scanners since they heavily rely on snapping Braille points to the grid. The grid lines in this instance can be converted into vertical and horizontal ones simply by de-skewing. Few articles ([4], [8]) clearly state the goal of OBR on smartphone images, although the techniques they describe still require the existence of a sheet with a rectangular grid.

Convolutional neural networks (CNN) have recently achieved great progress in the field of photo identification, despite deep learning and neural networks being sparingly utilised for object-based recognition (OBR). Only a small number of research that have been published have used fully coupled neural networks. Basic neural networks are used by Morgavi, Morando [7], Ting Li et al. [12], and Subur et al. [13] to determine the symbol value using the points identified by picture segmentation. Two-sided braille must be recognised in order to distinguish between front and rear points, according to Kawabe et al. [11] employ it.R.Li et al. presented their segmentation neural network research at CVPRW 2020 utilising a adaptable UNet architecture. They identified the regions using a neural network. that the Braille characters filled in order to identify the Braille characters.

In a further post-processing stage, the locations of individual symbols must be established based on the segmentation results. B Techniques for measuring optical braille recognition precisely There are at least two difficulties in comparing algorithms, even while different research offer varying numeric accuracy values of suggested algorithms, sometimes rather high:

was no open dataset to compare various algorithms until recently. It is extremely difficult to compare published results of different investigations because the quality values provided in papers were measured using proprietary datasets rather than published databases.Only the point recognition phase often includes quality indicators in works that employ the whole pipeline described above (i.e., points detection-grid restoration-grouping points into symbols-and-decoding). The performance of these algorithms cannot be compared to that of approaches like ours that do not include a separate point recognition stage.

Li et al. [6] provided the first DSBI dataset with braille text that was made available to the general audience. The train (26 pages) and test (88 pages) sets make up the book's 114 pages of scanned, two-sided braille texts. As the pages are scanned, they are all evenly aligned. For the front and back sides, as well as the rotation required to align the grid vertically and horizontally, calculations have been conducted to construct a grid of points. The annotation includes a list of braille letters that are connected to the nodes of this grid, the rotation angle, and the locations of the vertical and horizontal grid lines after rotation. Although all texts are in Chinese, this dataset can be utilised in any language because the Braille alphabet is global.

Although the training set is small (just 26 pages), you can still examine various methods because this dataset appears to be insufficiently vast and diversified to successfully train recognition algorithms. The authors of [6] evaluated the precision of their method in comparison to approaches based on picture segmentation techniques, Haar features, and Adaboost (Viola Jones [15]), as well as their own methodology (Antonacopoulos et al., 2004 [5]). Among others, Li [10] They only provide accurate point detection, though. The accuracy results for their technology, which were determined at the character level rather than the dot level, were only made public by
[14].

3 proposed system

We don't split the dots merging into characters, grid restoration, and dots detection procedures like other methods do. Instead, we swiftly recognise complete Braille symbols while also recognising them using the aforementioned object detection CNN. We use a formula where 1 = 1 if the i-th point is present in the character and 0 otherwise to assign each character a class label from 1 to 63. The Optical Braille Recognition task looks for many things that are nearly the same modest size and have a specified width to height ratio, which is the difference between the two tasks. We have optimised NMS operations within the ResNet architecture to shorten execution times. There was just one "output to class + box subnet" at the layer level, and the feature map cells were 16x16. At least one grid cell is guaranteed to cover each Braille character. Each grid cell contained simply one anchor, which was roughly the size of the anticipated character. These changes have resulted in a calculation time reduction of more than 5 times without noticeably lowering recognition quality.

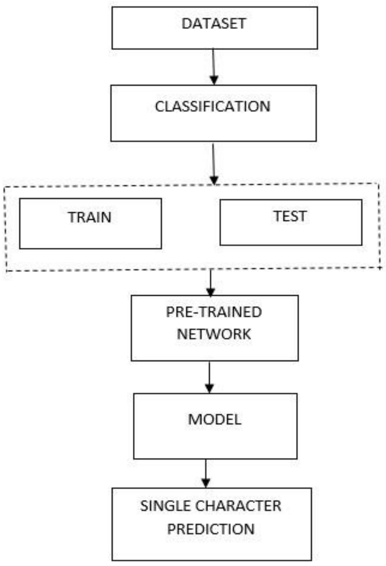

FIGURE 2.Process flow diagram for recognising braille characters.

4 METHODOLOGY :

In this project, the idea of transforming Braille characters to ordinary text is presented. The project's objectives can be divided into four major categories:

- Make a system that recognises Braille characters and translates them into English text using a digital image processing technique.
- By lowering errors, picture preparation methods can improve the accuracy of Braille recognition.
- Create a reliable matching and extracting algorithm to improve the recognition system.
- Offer a library of Python codes and templates for all design-related stages that are required.

4.1 Pre -Trained Models :

The pre-trained model contains the weights and biases that describe the characteristics of the dataset(s) it was trained on. Several different forms of data can regularly be used to learned features. For example, a model trained on a sizable dataset of bird images will have learned features like edges or horizontal lines that you could apply to your dataset.

ResNet and Inception, which provide great performance at a relatively low computational cost, have been important in enabling the significant improvements in image recognition performance over the past few years. The Inception architecture is combined with residual connections in Inception-ResNet.

- RSNET 50 ResNet is a type of convolutional neural network (CNN) that was created as a result of the 2015 study "Deep Residual Learning for Image Recognition" by He Kaiming, Zhang Xiangyu, Ren Shaoqing, and Sun Jian. CNNs are frequently used in computer vision applications. ResNet-50 is the name of a 50-layer convolutional neural network (48 convolutional layers, one MaxPool layer, and one average pool layer). Residual neural networks (RNNs) are artificial neural networks (ANNs) that construct networks using residual blocks.
- Inception-ResNet-v2 The convolutional neural network Inception-ResNet-v2 was trained using more than a million images from the ImageNet dataset. The 164-layer network can identify images into 1000 different object categories, including a wide variety of animals, the keyboard, mouse, and pencil. As a result, the network has gathered thorough feature representations for many different photos. With an input image size of 299 by 299 pixels, the network outputs a list of predicted class probabilities.

4.2 Model Optimization : Adam Optimizer

The Adam optimizer, an improved version of stochastic gradient descent, may be used in future deep learning applications including computer vision and natural language processing. Adam made his debut in 2014. It was first presented at the ICLR 2015 conference, a prestigious convocation of deep learning professionals. It is a technique for optimization that can take the place of stochastic gradient descent. The name derives from adaptive moment estimate. The optimizer, known as Adam, modifies the gradient's first and second moments to change the learning rate for each weight in the neural network. Adam is the name of the optimizer and is not an acronym.

IMAGE ENHANCEMENT : Since noise distorts the bulk of Braille images, deep models developed on noisy images will struggle to recognise Braille. Therefore, it is essential to enhance image quality before feeding it to any deep CNN model. The steps for improving an image are as follows:

- each image will be represented as an array of arrays after reading, therefore we concatenate them to one to build a list of all the images in the directory using a nxm matrix. Instead of using RGB, transform a grayscale image to a black-and-white one with only one pixel parameter. 0 denotes black and 255 denotes white.

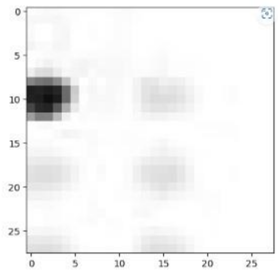

FIGURE 3 .Image after Enhancement

4.3 UNET

The UNET architecture for biological image segmentation was developed in 2015 at the University of Freiburg in Germany by Olaf Ronneberger et al. It is now one of the techniques for semantic segmentation problems that is most often employed. The neural network was developed with a smaller training sample set in mind and is entirely convolutional. Jonathan Long and coworkers created Fully Convolutional Networks for Semantic Segmentation (FCN) in (2014).

- Network Architecture for UNET: The U-shaped encoder-decoder network architecture, or UNET, is composed of four encoder blocks and four decoder blocks connected by a bridge. The encoder network at each encoder block doubles the number of filters (feature channels) and halves the spatial dimensions (contracting path). The number of feature channels is also cut in half, while the spatial dimensions are increased by two, thanks to the decoder network.
- Encoder: A succession of encoder blocks are used by the encoder network, which also functions as a feature extractor, to develop an abstract representation of the input image. Each encoder block has two 33-convolution layers, followed by a ReLU (Rectified Linear Unit) activation function. The ReLU activation function increases the network's nonlinearity, resulting in better generalisation of the training data. The output of the ReLU sends a skip connection to the relevant decoder block. The feature maps' spatial dimensions (height and width) are then divided in half by a 22 max-pooling. By reducing the number of trainable parameters, the computational cost is decreased.

- Bypass Connections: These skip links provide additional information that strengthens the decoder's capacity to generate semantic features. Additionally, they act as a rapid link that makes it possible for gradients to go unhindered to the lower layers. Simply said, skip connections help backpropagation of gradients run more smoothly, which helps the network acquire better representation. • Bridge By tying up the network of encoders and decoders, the bridge completes the information flow. A ReLU activation function follows each of the two 33 convolutions in this algorithm.
- Network Decoder After receiving the abstract representation, the decoder network creates a semantic segmentation mask. The first stage of the decoder block uses a 22 transpose convolution. The relevant skip connection feature map from the encoder block is then concatenated with it. Due to the network's depth, these skip connections offer functionality from prior layers that is occasionally lost. A ReLU activation function is then employed after two 33 convolutions.
- The output of the final decoder is placed through a 1:1 convolution with sigmoid activation. The sigmoid activation function provides the segmentation mask for the pixel-wise classification.

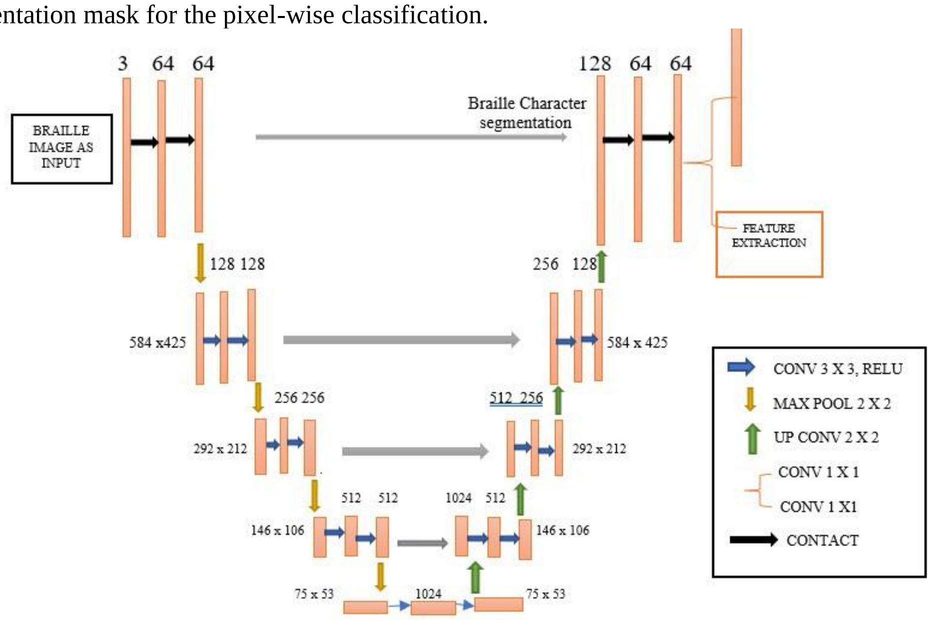

FIGURE 4. U-net architecture

5 RESULTS

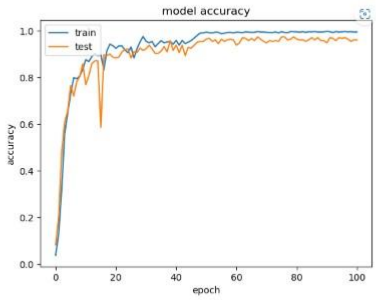

FIGURE 5. A Model Accuracy graph

Model accuracy is defined as the ratio of the total number of predictions made to the total number of classifications a model correctly predicts. It's one approach to assess a model's effectiveness, but it's not the only one.

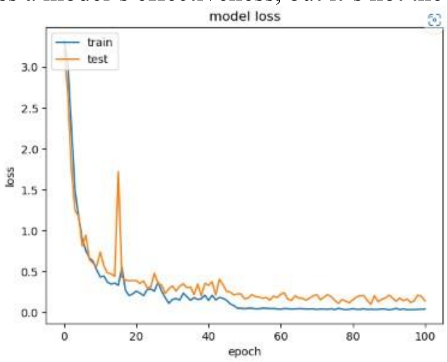

FIGURE 5. A Model Loss Graph

Loss is a metric for how poorly a model performed in a single instance of prediction. If the model's forecast is true, the loss is zero; otherwise, it is larger. The goal of training a model is to identify a collection of weights and biases that, on average, have low loss across all cases.

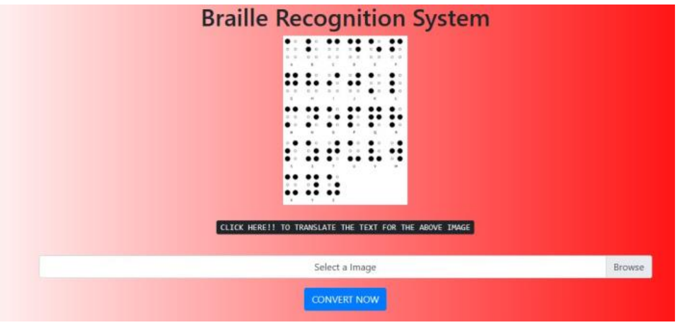

FIGURE 6. A Single Character Prediction The above picture shows the output for single character prediction

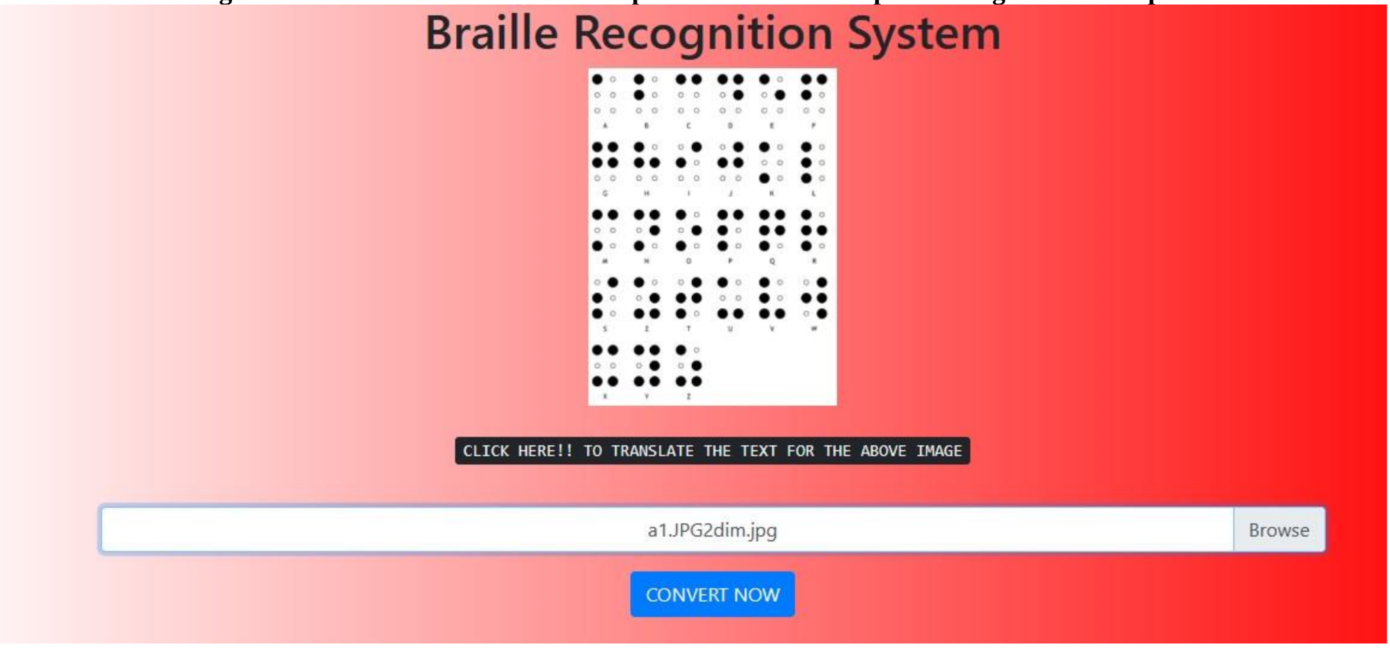

FIGURE 7. A sentence Prediction The above picture shows the output for sentence prediction by input the braille image

6 CONCLUSIONS

The suggested Braille recognition technique has shown to be extremely effective and accurate. Despite contradictions and perspective distortions on the displayed text-filled sheet, the system is still able to recognise messages. The proposed framework, which eliminates the need for picture de-skewing, Braille layout rules, or Braille dots detection, enables direct Braille letter identification and recognition in the original Braille images. The Braille Character Dataset was downloaded from kaggle.com at https://www.shanks0465/datasets.

ACKNOWLEDGMENT

Authors would like to thanks,Dr. Kumaraswamy S(Assistant Professor),Dept. of CSE for their constant encouragement towards the realization of this work.

References

1. Dubus, J. P., Benjelloun, M., Devlaminck, V., Wauquier, F., Altmayer, P. ”Image processing techniques to perform an autonomous system to translate relief Braille into black-ink, called: Lectobraille.” Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE, 1988.
2. Udayashankara, V. ”A review on software algorithms for optical recognition of embossed braille characters.” IJCA. Vol. 81. No. 3. 2013.
3. Isayed, Samer, and Radwan Tahboub. ”A review of optical Braille recognition.” 2015 2nd World Symposium on Web Applications and Networking (WSWAN). IEEE, 2015.
4. Zhang, Shanjun, and Kazuyoshi Yoshino. ”A braille recognition system by the mobile phone with embedded camera.” Second International Conference on Innovative Computing, Information and Control (ICICIC 2007). IEEE, 2007.
5. Antonacopoulos, A., and Bridson, D. ”A robust Braille recognition system.” International Workshop on Document Analysis Systems. Springer, Berlin, Heidelberg, 2004.
6. Morgavi, G., and Morando. M. “A neural network hybrid model for an optical braille recognition.” International Conference on Signal, Speech and Image Processing 2002 (ICOSSIP 2002). 2002.
7. Venugopal-Wairagade, G. ”Braille recognition using a camera-enabled smartphone.” Int J Eng Manuf 4 (2016): 32-39.
8. Li, R., Liu, H., Wang, X., Qian, Y. ”Effective Optical Braille Recognition Based on Two-Stage Learning for Double-Sided Braille Image.” Pacific Rim International Conference on Artificial Intelligence. Springer, Cham, 2019.

9. Kawabe, H., Shimomura, Y., Nambo, H., Seto, S.”Application of deep learning to classification of braille dots for restoration of old braille books.” International Conference on Management Science and Engineering Management. Springer, Cham, 2018.
10. Li, Ting, Xiaoqin Zeng, and Shoujing Xu. ”A deep learning method for Braille recognition.” 2014 International Conference on Computational Intelligence and Communication Networks. IEEE, 2014.
11. Subur, Joko, Tri Arief Sardjono, and Ronny Mardiyanto. ”Braille Character Recognition Using Find Contour And Artificial Neural Network.” JAVA Journal of Electrical and Electronics Engineering 14.1 (2016).
12. Li, R., Liu, H., Wang, X., Xu, J., Qian, Y. ”Optical Braille Recognition Based on Semantic Segmentation Network With Auxiliary Learning Strategy.” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2020.
13. Viola, Paul, and Michael Jones. ”Rapid object detection using a boosted cascade of simple features.” Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001. Vol. 1. IEEE, 2001.
14. LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., Jackel, L. D. ”Backpropagation applied to handwritten zip code recognition.” Neural computation 1.4 (1989): 541-551.
15. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. ”Imagenet classification with deep convolutional neural networks.” Communications of the ACM 60.6 (2017): 84-90.
16. Girshick, R., Donahue, J., Darrell, T., Malik, J. ”Rich feature hierarchies for accurate object detection and semantic segmentation.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.
17. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., Berg, A. C. ”Ssd: Single shot multibox detector.” European conference on computer vision. Springer, Cham, 2016.
18. Fu, C. Y., Liu, W., Ranga, A., Tyagi, A., Berg, A. C. ”Dssd: Deconvolutional single shot detector.” arXiv preprint arXiv:1701.06659 (2017).
19. Redmon, J., Divvala, S., Girshick, R., Farhadi, A. ”You only look once: Unified, real-time object detection.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
20. Redmon, Joseph, and Ali Farhadi. ”YOLO9000: better, faster, stronger.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
21. Lin, T. Y., Goyal, P., Girshick, R., He, K., Dollar, P. ”Focal loss for dense object detection.” Proceedings of the IEEE international conference on computer vision. 2017.
22. Diederik P. Kingma and Jimmy Lei Ba. Adam : A method for stochastic optimization. 2014. arXiv:1412.6980v9