

Braille Character Recognition Model

For this project, I'm aiming to build a model that can recognize Braille characters from images and translate them into plaintext. This application would provide accessibility benefits by allowing visually impaired individuals and others to easily convert Braille text into readable text.

Choice of Dataset:

I will use publicly available datasets from Kaggle, such as Braille character image datasets, to train my model. I will use [\[1\]](#) to train for single-character classification and later [\[2\]](#) for more complex tasks like word/sentence detection in natural settings (with rotation/noise/low contrast). [\[3\]](#) [\[4\]](#) also have annotated images. Additionally, I may augment the dataset by gathering real-life examples of braille and labelling them.

Methodology: Broadly, the project is planned to advance in the following steps:

1. Data Preprocessing:

- Convert images to grayscale for uniformity.
- Resize images to a fixed dimension to maintain consistency in training.
- Apply image augmentation techniques like rotation, contrast adjustment, and noise addition to improve model robustness.
- Ensure Braille characters correctly labelled for supervised learning.

2. Machine Learning Model:

- I will begin with a simple Convolutional Neural Network (CNN) to recognize Braille characters.
- If necessary, I will explore pre-trained models and understand what parameters/techniques are used to improve accuracy.
- The model will classify each Braille character, which will then be mapped to its corresponding text representation.
- Research into alternative approaches like Optical Character Recognition (OCR) techniques

3. Evaluation Metric:

- Since this is a classification problem, I will evaluate the model using accuracy, precision, recall to analyze confusion matrices and identify misclassifications.
- If needed, benchmark the model against existing solutions to measure improvement.

Application: To make my model accessible, I will integrate it into a simple web application with the following features:

- **User Input:** Users can upload an image containing Braille text.
- **Processing:** The application will process the image, run it through the trained model, and extract the corresponding plaintext output.
- **Output:** The translated text will be displayed on the screen. Additionally, a text-to-speech (TTS) feature can be integrated to read the output aloud, further enhancing accessibility.

For deployment, I plan to use Flask for the backend, and a frontend built with Javascript/React. By the end of this project, I aim to create a fully functional and accurate Braille-to-text recognition system that enhances accessibility for those relying on Braille reading systems.

In the future, I might look into adding an NLP model to help with low-certainty predictions (e.g., finding similar words) to improve classification accuracy or use an ensemble approach (combining outputs of multiple models) to enhance performance.

References:

1. <https://www.kaggle.com/datasets/shanks0465/braille-character-dataset>
2. lu li qiong, Wu Dong, Xiong Jianfang. Natural Scene Braille Character Recognition Dataset[DS/OL]. V1. Science Data Bank, 2024[2025-02-19].
<https://cstr.cn/31253.11.sciencedb.j00188.00007>. CSTR:31253.11.sciencedb.j00188.00007.
<https://www.scidb.cn/en/detail?dataSetId=b1df1a601acc47a6984aafa8f3ab8e92>
3. <https://universe.roboflow.com/braille-lq5eh/braille-detection>
4. https://universe.roboflow.com/thesisbr/br_tbc/browse?queryText=&pageSize=50&startingIndex=0&browseQuery=true