


Article

Learning at Your Fingertips: An Innovative IoT-Based AI-Powered Braille Learning System

Ghazanfar Latif ^{1,*}, Ghassen Ben Brahim ¹, Sherif E. Abdelhamid ^{2,*} , Runna Alghazo ³, Ghadah Alhabib ¹ and Khalid Alnujaidi ¹

- ¹ Department of Computer Science, Prince Mohammad Bin Fahd University, Khobar 34754, Saudi Arabia; gbrahim@pmu.edu.sa (G.B.B.); 202000093@pmu.edu.sa (G.A.); 202002530@pmu.edu.sa (K.A.)
² Department of Computer and Information Sciences, Virginia Military Institute, Lexington, VA 24450, USA
³ Department of Education, Health, & Behavioral Studies (EHBS), University of North Dakota, Grand Forks, ND 58202, USA; runna.alghazo@und.edu
* Correspondence: glatif@pmu.edu.sa (G.L.); abdelhamidse@vmi.edu (S.E.A.)

Abstract: Visual impairment should not hinder an individual from achieving their aspirations, nor should it be a hindrance to their contributions to society. The age in which persons with disabilities were treated unfairly is long gone, and individuals with disabilities are productive members of society nowadays, especially when they receive the right education and are given the right tools to succeed. Thus, it is imperative to integrate the latest technologies into devices and software that could assist persons with disabilities. The Internet of Things (IoT), artificial intelligence (AI), and Deep Learning (ML)/deep learning (DL) are technologies that have gained momentum over the past decade and could be integrated to assist persons with disabilities—visually impaired individuals. In this paper, we propose an IoT-based system that can fit on the ring finger and can simulate the real-life experience of a visually impaired person. The system can learn and translate Arabic and English braille into audio using deep learning techniques enhanced with transfer learning. The system is developed to assist both visually impaired individuals and their family members in learning braille through the use of the ring-based device, which captures a braille image using an embedded camera, recognizes it, and translates it into audio. The recognition of the captured braille image is achieved through a transfer learning-based Convolutional Neural Network (CNN).

Keywords: braille learning; education for the blind; internet of things (IoT); deep learning; smart device; convolutional neural networks; transfer learning



Citation: Latif, G.; Brahim, G.B.; Abdelhamid, S.E.; Alghazo, R.; Alhabib, G.; Alnujaidi, K. Learning at Your Fingertips: An Innovative IoT-Based AI-Powered Braille Learning System. *Appl. Syst. Innov.* **2023**, *6*, 91. <https://doi.org/10.3390/asi6050091>

Academic Editor: Dorota Temple

Received: 16 July 2023

Revised: 11 September 2023

Accepted: 28 September 2023

Published: 11 October 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Braille is the universal form of literacy for the blind and visually impaired. Braille bridges the communication gap between the visually impaired and their surroundings. It is the only textual representation that the blind and the visually impaired can understand. A major challenge faced by the blind and the visually impaired is their need to learn the braille language to be able to read and learn in general. They require a dedicated instructor and one-to-one supervision to learn braille. Visually impaired individuals living in small cities or rural areas find it challenging to learn the language due to the limited number of educational institutes and special education schools present in these locations. There is also a shortage of resources provided for enhancing the learning environment of these individuals. Furthermore, the process of learning braille is both time-consuming and requires specialized personnel due to the need for a specialized instructor to guide and assist the blind in learning. Therefore, the aforementioned challenges cause many visually impaired individuals to feel discouraged and unmotivated to learn. If an automated translation system exists that is proficient in braille, then it will accelerate the learning process for the visually impaired because computers can process information and translate braille text much faster than humans [1]. This means that a visually impaired individual

would be able to read a braille text much faster by having the automated IoT-based system directly translate a braille document to audio.

Braille recognition systems translate printed braille to its textual and natural language representations. The dotted format of braille documents is the starting point for any automated recognition system. The system should be able to capture the dotted format of the braille language, translate it to the corresponding alphabet letter, and combine the letters to recognize words. There are different braille systems for different languages; thus, the development of an automated braille translating system should target a particular language or should at least have the capability to choose between languages for bilingual users. The Automatic braille recognition system could use computer vision and machine learning models like Conventional Neural Networks (CNN), Decision Trees (DT), K-nearest neighbor (KNN), and Support Vector Machines (SVM) for this purpose. Deep learning (DL) and computer vision are extensively used for pattern recognition and image classification [2–4].

The main goal of this work is to develop an automated system based on artificial intelligence for Arabic and English bilingual individuals who are visually impaired. The main reason for the exact choice of these two languages is that English is taught in many Arabic-speaking countries; thus, most individuals in Arabic-speaking countries—especially in the Middle East—are bilingual (proficient in Arabic and beginner to intermediate in English). Thus, the goal is to develop a device that will assist in the following ways:

1. Teach visually impaired individuals braille with a learn-at-your-own-pace methodology without the need for professional Braille instructors;
2. Teach braille to the parents of visually impaired individuals so that they can in turn teach braille to their children;
3. In terms of recognizing braille characters, assist visually impaired individuals in reading braille documents and books at much faster speeds and with a high accuracy level.

This research work offers a 4-fold contribution consisting of these objectives:

1. Presents an extensive survey of existing techniques to detect Braille in different languages;
2. Designs an IoT-based system that can fit on the ring finger, simulating the real-life experience of a visually impaired person;
3. Develops an ML-based model to recognize and translate Arabic and English braille into audio using deep learning techniques with transfer learning;
4. Creates a new bilingual Arabic–English braille dataset, which is to be expanded using data augmentation techniques;
5. Perform a performance evaluation study of the entire system with regard to accuracy and effectiveness.

The research topic is significant because the visually impaired lack access to both educational centers that have Braille translation systems and instructors for the learning process. It is estimated by the INEI that only 23.9% of visually impaired individuals manage to complete their education, thus indicating the need for a system that supports translation from Braille to text for the integration of the visually impaired into their communities. The implementation of language translation systems is crucial to restrict the communication gap, and performing further research is important for providing open sources on how to build translators. Sometimes, a person may wish to learn braille to teach it or to communicate with someone with visual disabilities. This improves the daily life activities of the visually impaired [5].

The rest of the paper is organized as follows: Section 2 discusses a review of the recent studies, Section 3 explains the methodology proposed, the experimental results are discussed in Section 4, and the research work is concluded in Section 5.

2. Review of Recent Studies

In [1], researchers suggest a deep learning scheme for character detection with a position-free touchscreen-based input methodology. This device translates braille input into

natural language by simply tapping on the dots of each character. The dataset used in this research is composed of 1258 photographs of sizes 64×64 with two categories: Category-A (a–m) and Category-B (n–z). The dataset was obtained from a screen interface for Android devices. The input braille text is processed and entered into the Convolution Neural Network (CNN). Two CNN techniques were used: transfer learning and the sequential model. The recognition is achieved using a deep learning model trained using the gathered braille dataset. The classification evaluation was carried out using DL techniques such as the GoogleNet Inception model, achieving an accuracy of 95.8%, and the sequential model, achieving a total accuracy of 92.21%.

In [6], the authors proposed a touchscreen to detect Urdu braille characters using ML methods. The dataset obtained from the National Special Education School is composed of 39 classes sorted into three groups with 13 classes in each group, 144 cases for each class resulting in 5616 cases in total. The letters are input into the screen. The methodology uses a Reconstruction Independent Component Analysis (RICA)-based feature extraction model. The highest-achieving classifier was the support vector machine (SVM) with a yielded accuracy of 99.73% accuracy. However, other robust ML techniques were used such as K-nearest neighbors (KNN) and decision trees (DT) for comparison purposes. The evaluation was conducted in terms of total accuracy, true positive rate, true negative rate, false positive rate, positive predictive value, negative predictive value, and area under the receiver operating curve. Unfortunately, this study is only limited to Grade 1 Urdu braille and does not include Grade 2 Urdu braille with speech and text responses.

In [7], the authors suggest using RICA-based feature extraction methods and automated tools to extract English braille alphabets. The proposed methodology uses a Grade 1 English braille dataset obtained from a touchscreen from the National Special Education School along with a position-free braille text entry technique to produce synthetic data to generate a dataset composed of 2512 cases. The dataset comprises 26 braille English letters and is divided into two classes: class 1 (1–13) and class 2 (14–26). For character recognition, Decision Trees (DT), Support Vector Machine (SVM), and K-nearest neighbor (KNN) with PCA-based feature extraction methods and Reconstruction Independent Component Analysis (RICA) were implemented. RICA outperformed PCA and the SVM classifier also achieved an accuracy of 99.85%. Sequential methods and RF methods yielded the highest accuracy with a value of 90.01%. The performance was evaluated based on total accuracy, true positive rate, true negative rate, false positive rate, positive predictive value, negative predictive value, and area under the receiver operating curve. The accuracy achieved is 100% for classes such as a, c, d, h, i, j, p, u, w, and k, 99.87 and 99.60% for other classes such as b, f, q, s, t, and v. The study is only suitable to Grade 1 English character braille and cannot be implemented with restricted computation power. The study also does not use DL methods such as CNN and GoogleNet to enhance the outcome.

Authors in [8] recommend using a Histogram of Oriented Gradient Features and a Support-Vector Machine (SVM) for braille recognition and feature extraction. The method can translate Sinhala braille to Sinhala language and English braille to the English language. The images are processed, segmented, and then recognized using HOG feature extraction methods and the SVM classifier method. The study uses two types of HOG feature extraction methods: a cell size of 4×4 , and another one of 2×2 . The dataset is composed of both scanned handwritten and computer-generated braille text. The methodology can process Grade 1 English characters as well as some Grade 2 characters. The yielded accuracy was 99%. The authors report that higher processing time was needed in the case of 2×2 cells compared to 4×4 cells.

Reference [9] advocates for using a Semantic Retrieval System to assist visually impaired individuals in mathematical studies. The methodology begins with translating a query math formula in braille into MathML code, and then the structural and semantic meaning is obtained from the MathML expression to produce a multilevel tree. The feature extraction method used is the conventional vector model. Afterward, in the classification stage, the K-nearest neighbors method is used to choose a multilevel similarity measure to

compare between expressions. Lastly, the query produced is translated to braille mathematical expressions. The dataset was created using MathType and consists of 6925 mathematical equations and expressions from five languages: Hebrew, Japanese, Tifinagh, Arabic, and Latin. For each language, 1385 different types of equations were written. This study used Latin to test the performance of the methodology.

Authors in [10] have used a novel approach of the CNN extraction method to translate Bangla handwritten text to Bangla braille notation. The study used an object detection model, Faster-RCNN to draw boundaries over Bangla cells and then used 10 CNN models for classification. Faster-RCNN is a fast and efficient algorithm. The CNN models used include VGG16, DenseNet201, ResNet152V2, MobileNet, and ZFNet. The CNN models were trained and tested using the Microsoft Azure ML platform for calculation using Standard_NV48s_v3. Results show that the highest achieving accuracy CNN model was VGG16 with a value of 95%. The methodology was implemented using Python v3, Keras, and TensorFlow libraries. Furthermore, the dataset was collected from external resources of handwritten Bangla, the images were resized using a canny edge detection, and a median filter was applied to decrease the noise and threshold. Afterward, it is converted to black and white. The dataset comprises 105 classes with 157,500 photographs where 80% were kept for training and 20% were kept for testing. Each class comprises 1500 photographs. Unfortunately, this study covered a limited number of conjunctions and many of the 300 conjunctions of the Bangla language were not considered.

In [11], the paper recommends using machine learning (ML) for character recognition of Hindi handwritten documents to translate to braille text. The pages are first transformed into a printable form and then converted to braille using UTF-8 codes. The dataset used is composed of 92,000 images and for each of the 46 characters, 2000 images are used for classification. However, vowels and Matras are discarded from the dataset. Additionally, the author uses a Histogram of oriented gradient features of Hindi characters to extract features. The segmented letters are then classified using an SVM classifier for character recognition. To produce higher levels of accuracy, the resolution of the image should be greater than 300 dpi. Further, the range of accuracies achieved is between 87.667% to 97.667%. This study is unique because it tackles a language with limited resources. The results showed that the classifier failed to predict the letters "HA" and "DHA", which is considered a limitation of the proposed model. However, because the cell size used was upgraded to 4×4 the average accuracy increased from 94.65% to 95.56%.

In [12], the study encourages using a Convolution Neural Networks (CNN) system to classify images of braille and translate them to English characters. The dataset used is composed of 14,378 braille photographs. The 3-major steps in the conversion process are pre-processing segmentation and image classification. In pre-processing, the method uses grayscale conversion, contrast adjustment, finding circles, and inverting colors. Segmentation is divided into line segmentation and cell segmentation. For image classification, DL algorithms and CNNs are used with nine different layers including, an input layer, convolutional filter, max pooling, and output layer, etc. The paper yields a high accuracy with a value of 96.37% for a 500-image containing dataset size. Furthermore, the scheme possesses high-performance characteristics due to the implementation of deep learning and not only simple neural networks.

Authors in [13] consider a deep learning-based model that combines the CNN model to detect characters and transformer models to recognize words. The results showed that the proposed model achieves high performance in terms of accuracy in detecting characters and words reaching 98.6% and 96.7%, respectively.

In [14], the authors proposed a hardware device to aid visually impaired individuals. This device combines the use of long short-term memory (LSTM) along with Raspberry Pi and the convolutional neural network (CNN). The proposed system recognizes numbers, letters, dots, and punctuation. Performance-wise, the system achieved a high level of accuracy, reaching 98%.

Artificial intelligence (AI) and Deep Learning (ML) have been used in research to assist students with disabilities as well as in other fields such as the medical field, sign language, and handwritten text classification. In [15], the authors proposed an automated AI-based system for assisting the deaf and hard of hearing to communicate with their surrounding community. Using Random Forest (RF), the authors reported an accuracy of 92.15%. In [2], the author proposed a system for assisting the deaf and hard of hearing using deep learning (DL). They reported an accuracy of 97.6%. In [3], the authors proposed an automatic AI-based system for the automatic recognition of multi-lingual handwritten digits using novel structural features. They reported an accuracy of 96.15%. There are many more examples of AI and ML being used to automate and develop automatic systems in many fields including medicine, agriculture, education, etc. The continued pursuit of optimal solutions will develop over time until the optimal solutions are reached and developed into patented devices that could actually be used and assist in making people's lives better.

Attempts were also made to design a model to perform a reverse operation of what this current research aims. For instance, the authors in [16] designed a CNN-based model to recognize real-time Arabic speech and eventually translate it into Arabic text then convert it into Arabic braille characters. The model works on digits and is yet to be improved to include alphabets. An accuracy performance of 84% was achieved when adding the ReLU activation function to the CNN model.

3. Proposed Methodology

The proposed system shown in Figure 1 is designed to be compact, portable, and fitting on the tip of a finger. Equipped with a digital camera, it is capable of capturing images of the braille dots for processing. The dimensions of each braille dot are determined based on the tactile resolution of a person's fingertips. The dot's height measures approximately 0.5 mm (0.02 inches), with a vertical and horizontal spacing of 2.5 mm (0.1 inches) between dot centers and a spacing of 3.75 mm (0.15 inches) between adjacent cells. A standard braille document measures 11 × 11.5 inches with each line having between 40 and 43 cells.

Figure 1 shows a detailed workflow of the proposed system. During the AI software processing phase, the captured image with the help of a button will be segmented to exclude the region of the image that does not contain braille dots. The IoT system follows a series of image processing steps, including edge detection, binary conversion, hole fitting, and image filtering. Preprocessing methods are used to reduce noise and enhance the visibility of the dots. The system also performs segmentation to allow for individual identification of the letters. During the next step, the image is resized to 16 × 16 pixels. In the braille system, each letter is represented by a single cell consisting of six dots arranged in two columns and three rows. Once the image has been extracted, the system undergoes training to classify the braille characters based on their corresponding classes. Each letter or number is associated with a specific class, allowing for accurate mapping. The performance of the algorithms used to train the models is evaluated in terms of accuracy, positive and negative predicted values, and other relevant metrics. It is important to note that misclassification errors may arise due to challenges encountered during noise removal, variations in braille dot sizes, and the process of segmentation.

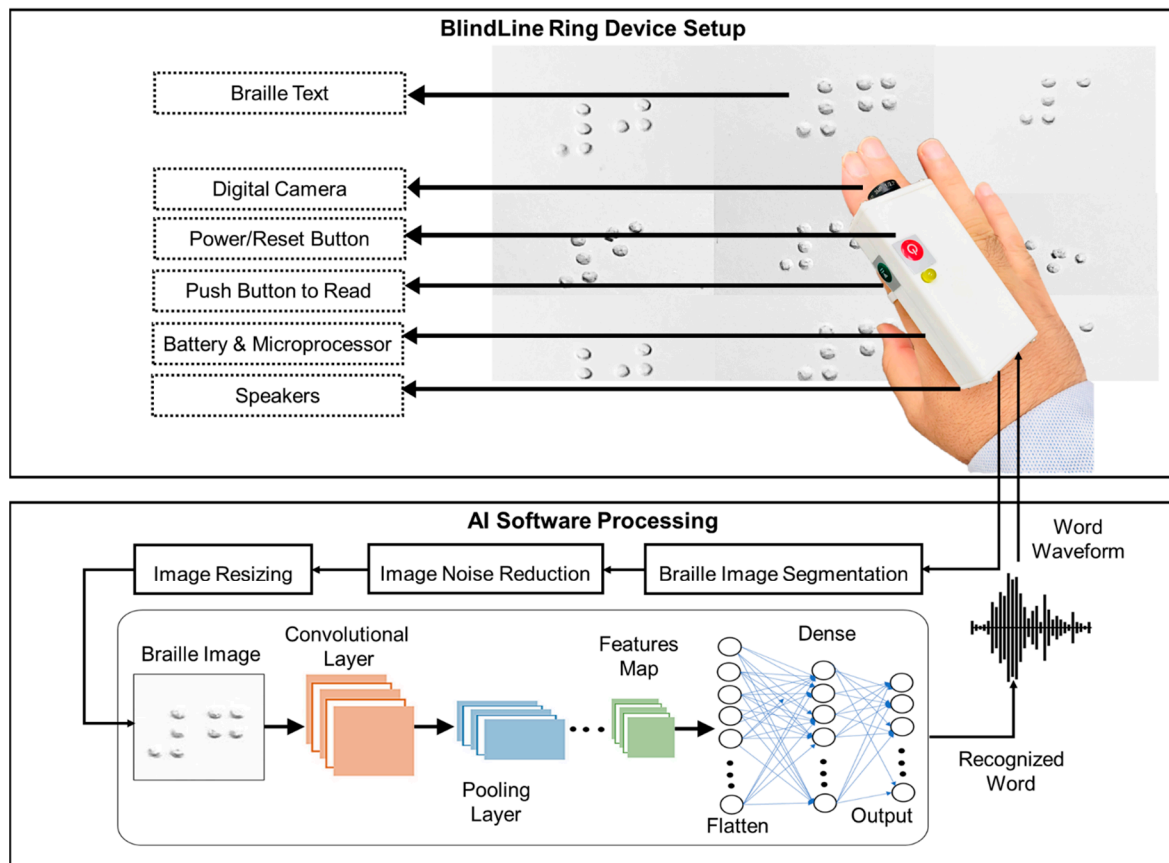


Figure 1. Workflow of the proposed system for the IoT-based braille language learning on finger tip.

3.1. Experimental Dataset

This research was conducted on a new built dataset containing images of Arabic and English braille characters. The dataset is used as an input to test the validity and efficiency of the proposed methodology and is composed of 28 Arabic characters (from “أ” to “ي”), and 26 English characters (from “a” to “z”) as shown in Figures 2 and 3, respectively. The different augmentation methods are applied to the collected images including width-height shift, rotation, and brightness which change the shift, rotational, and brightness values, accordingly. English braille dataset is composed of ‘A’ to ‘Z’ English alphabetical letters and comprises 500 labeled images for each class which is deemed sufficient for the training, validation, and testing of the model for braille dots. Similarly, the 26 Arabic characters dataset was also augmented to have 500 labeled images of each character’s class used for the training while another 15 non-augmented images of each character were used for testing. The images were cropped individual letters and the image name contains the number of the image, the character alphabet, and the type of data augmentation. The images in the dataset possessed different brightness for better machine learning training and character recognition. It is important to mention that the detection of braille characters may be challenging due to their small size, the minimized visual contrast with their background, similarity between characters. Our dataset design involved printing braille letters on single-sided A4 embossed paper in blue and white, creating the images. These images were captured using smartphone cameras, ensuring diversity by varying lighting conditions, colors, angles, and heights. To optimize processing, the images were converted to grayscale, and resized to 256 pixels.

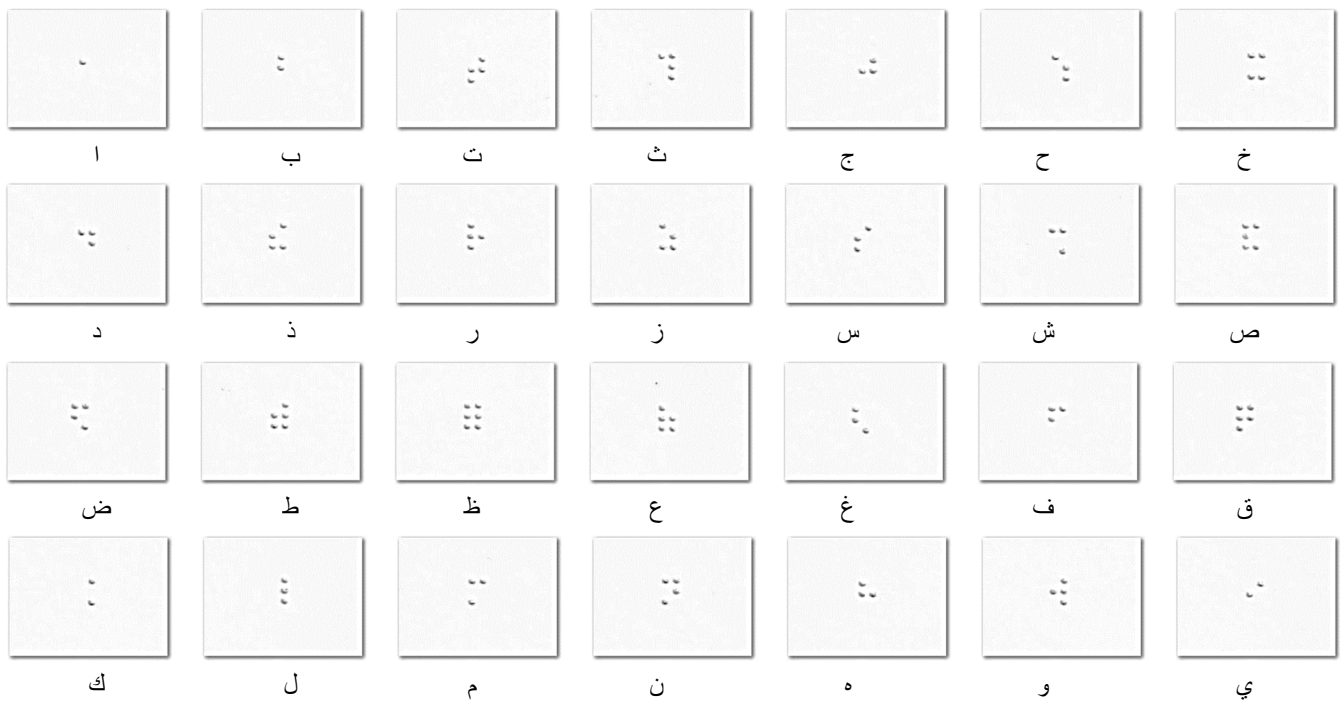


Figure 2. Sample braille for the Arabic braille language for Arabic alphabets.

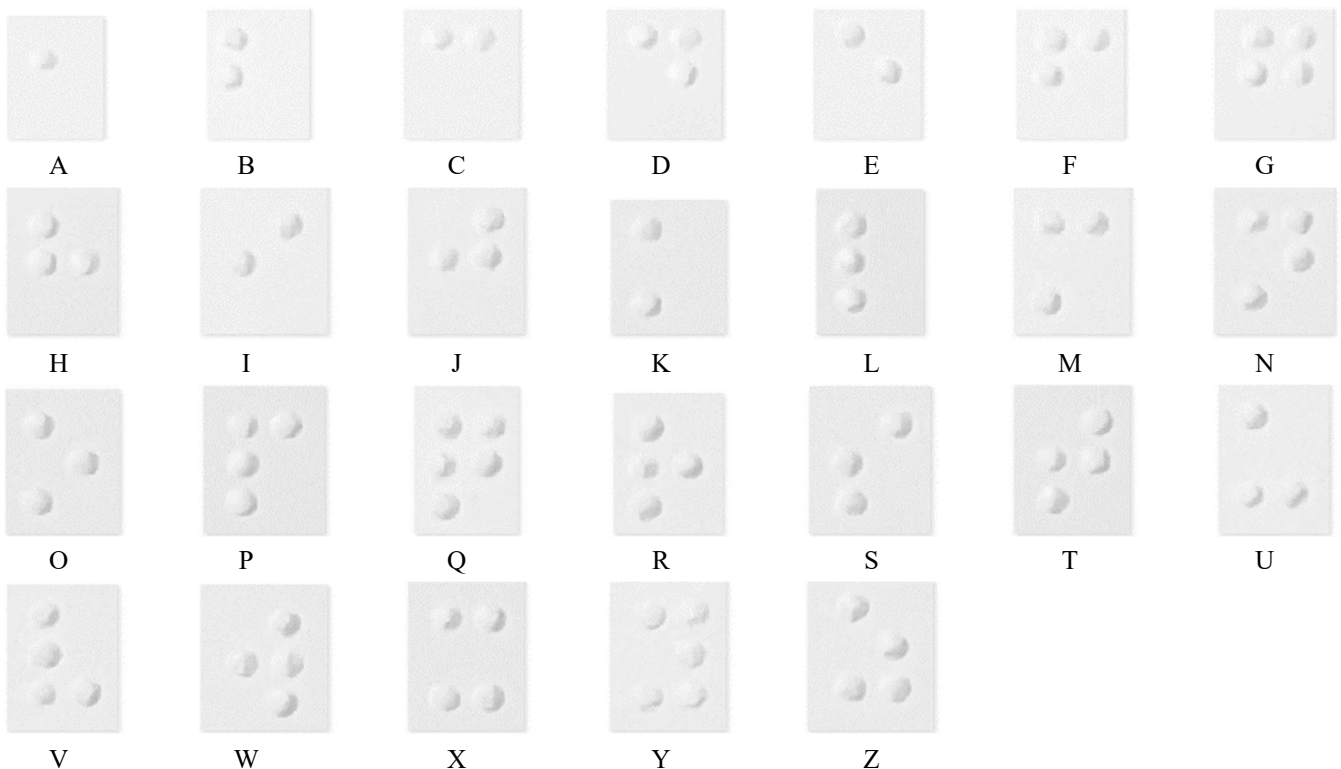


Figure 3. Sample braille for the Arabic braille language for English alphabets.

3.2. Convolutional Neural Network-Based Transfer Learning

Convolutional Neural Network (CNN) is an algorithm widely used in computer vision and deep learning. The algorithm takes an image as input and assigns significance to several objects in that image to distinguish one from the other. CNN algorithm requires minimal pre-processing compared to other classification methodologies. The CNN-based

models are generally divided into three major layers: the convolutional layer, the pooling layer, and the fully connected layer [17,18]. The algorithm begins with reducing the image into an easily processed form while preventing the loss of significant features. This aids the creation of an architecture that can learn features and is scalable to interpret new datasets.

In the convolutional layer, the Kernel/Filter, K , is the element performing the convolution operation in the first part of the layer. The filter traverses the image by moving to the right until it covers the full width and then down until it covers all pixels. The goal of convolution operations is to extract high-level features of an image. The results are of two types: dimensionality is either increased or stays the same by applying the same padding or convolved features reduced in dimensionality by applying valid padding. CNN may have multiple convolutional layers. The first layer captures the low-level features and with additional layers, it adapts the high-level features. This builds a system that can interpret images.

The next layer is the pooling layer, where the spatial size of the convolved feature is reduced to minimize the computational power necessary for data processing through dimensionality reduction. The pooling layer extracts rotational and positional invariant dominant features for model training. Pooling has two types: average pooling and max pooling. Average pooling computes the average of all the values from the section of the image covered by the kernel. On the contrary, max pooling selects the maximum value from the section covered by the kernel and implements a noise suppressant.

The convolutional layer and the pooling layer compose the i th layer of a CNN. Each architecture has a unique number of layers depending on the complexity of the image. Increasing the number of layers assists in capturing additional low-level details but requires more computational power. Now, the model can interpret the features and complete the first stage of the architecture to then move to the next stage and feed the classification model. In the third and last stage or the fully-connected layer, the image is flattened into a column vector and is fed into the neural network. The model then differentiates between significant and insignificant features and classifies them using the SoftMax classification technique. With each layer, the model increases in complexity and can identify more sections of a photo. Earlier layers extract simpler features and later ones extract more elements used to identify the object [19].

ConvNet includes several architectures such as LeNet, AlexNet, DenseNet, GoogleNet, and VGGNet [20]. These models are widely adopted as transfer learning to retrain the models with the new datasets for different applications. AlexNet is an extension of LeNet with a deeper architecture. It has eight layers in total: five convolutional layers and three fully connected layers. All layers are connected to a ReLU activation function. AlexNet employs data augmentation and dropout techniques to prevent overfitting due to excessive parameters.

DenseNet can be considered an extension of ResNet, where the output of a previous layer is added to a subsequent layer. DenseNet proposes concatenating the outputs of previous layers with subsequent layers, which enhances the distinction in the input of succeeding layers, thereby increasing efficiency. DenseNet significantly reduces the number of parameters in the learned model. For this research, the DenseNet-201 architecture was used. It has four dense blocks, each followed by a transition layer except for the last block, which is followed by a classification layer. A dense block contains several sets of 1×1 and 3×3 convolutional layers, while a transition block contains a 1×1 convolutional layer and a 2×2 average pooling layer. The classification layer in DenseNet-201 consists of a 7×7 global average pool followed by a fully connected network with 28 outputs based on the 28 Arabic braille letters.

GoogleNet architecture is based on inception modules, which perform convolution operations with different filter sizes at the same level. This increases the width of the network. The architecture has 27 layers (22 layers with parameters) and nine stacked inception modules. At the end of the inception modules, a fully connected layer with a SoftMax loss function serves as the classifier for the 28 classes of Arabic braille letters.

3.3. Fine-Tuned VGG16 Architecture

Large-scale visual data classification is usually performed using VGG16 and VGG19 CNN architectures. VGG16 is a CNN that could be combined with transfer learning for the classification process [21]. VGG16 is divided into three parts: convolutional layers which utilize filters for feature extraction from images, pooling layers for reducing spatial size, thereby decreasing the number of parameters and computations, and fully connected layers for final classification. When combining VGG16 with transfer learning, the model is expected to become more accurate, faster, and require less training time. This is a result of the fact that VGG16 is already pre-trained on large datasets and thus can detect particular features. Transfer learning allows leveraging the VGG16 pre-trained weights thereby increasing efficiency.

Small convolutional filters are used in the VGG16 architecture to increase network depth. The input is of size $224 \times 224 \times 3$, where 3 refers to 3 color channels. As depicted in Figure 4, the input images go through the convolutional layers along with the small receptive field of size 3×3 and the max pooling layers. As shown in Figure 4, the first two sets of VGG utilize conv3-64 followed by a conv3-128 layer, using the ReLU activation function. The remaining three sets use conv3-256, conv3-512, and conv3-512, respectively, also utilizing the ReLU activation function. A stride of 2 and 2×2 always accompanies the convolutional layers in VGG16 and VGG19, while varying the number of channels between 64 to 512. It should be noted that the only difference between VGG19 and VGG16 is the presence of 16 convolutional layers. The fully connected layer usually has outputs representing the number of classes and in this case, it has 28 outputs corresponding to the 28 Arabic braille letters.

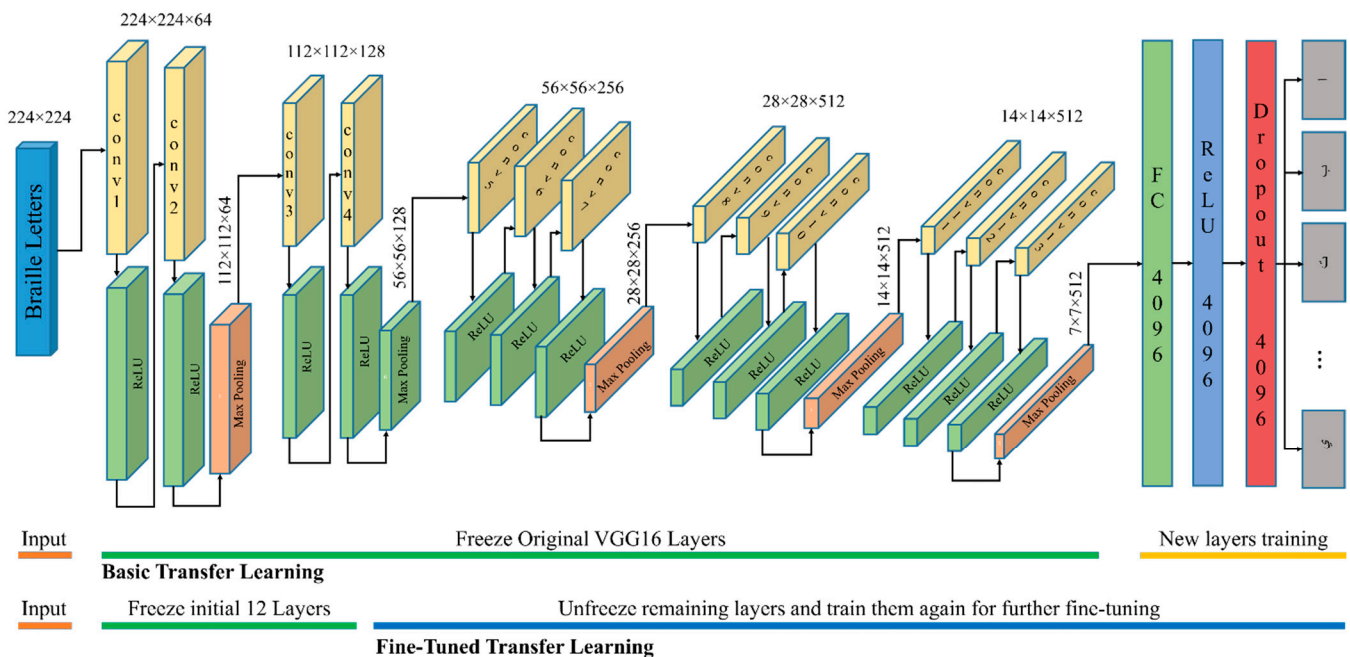


Figure 4. Fine-tuned VGG16 architectures for braille language detection.

4. Results and Discussions

The original and augmented datasets were used in the experiments in order to increase the overall size of the dataset. Various metrics were used in order to evaluate the performance of the proposed methodology. These include recall, precision, accuracy, and F1 measure [22]. In the proposed model, the idea is to freeze the top twelve layers and unfreeze the remaining layers to retrain the unfrozen layers. The decision to freeze the initial layers and retrain the later layers was made to balance pre-trained knowledge while adapting to our specific task. The determination of optimal layers for freezing and retraining was based on systematic experimentation, aiming for a balance between prior knowledge and

task-specific adaptation. This approach was applied to various deep learning models including VGG19, VGG16, DenseNet, AlexNet, GoogleNet, and LeNet. The proposed approach was applied to the combined dataset and to both the Arabic and English braille letters. In order to compare the performance of the proposed approach, the first experiment was performed using the original freeze weight of the original CNN models applied to the Arabic braille language dataset. The results are shown in Table 1. It should be noted that each letter has 500 images being used. These are divided into 300 images (60%) of each letter for training, 100 (20%) for validation, and 100 (20%) for testing. This percentage was used for all letters in both the Arabic and English braille alphabets. The experiments are performed for 30 epochs with a batch size of 512 with Adam optimizer and a learning rate of 0.001.

Table 1 shows the results of the experiments of the original CNN models using freeze weight applied to the Arabic braille language dataset. The results indicated that the best accuracy was achieved using GoogleNet with an average value of 98.63% and 98.4%, 98.4%, and 98.1% for precision, recall, and F1-measure, respectively. The lowest accuracy was reported for the GoogleNet algorithm with an average accuracy of 94.50%.

Table 1. Experimental results of freeze weights of the original CNN models for the Arabic braille language data.

	Accuracy	Precision	Recall	F1 Measure
VGG19	98.31%	0.982	0.983	0.983
VGG16	98.63%	0.984	0.984	0.981
DenseNet	95.32%	0.953	0.952	0.953
AlexNet	98.27%	0.976	0.977	0.977
GoogleNet	94.50%	0.942	0.943	0.942
LeNet	85.48%	0.849	0.853	0.851

Figure 5, Figure 6, Figure 7, Figure 8, Figure 9, and Figure 10 show a comparison of the training and testing validation accuracies for VGG19, VGG16, DenseNet, AlexNet, GoogleNet, and LeNet, respectively. The comparison shows that both training and testing validation accuracies approach 100% as expected. These results indicate that the overfitting and the under-fitting problems were accounted for in this research with no under-fitting or overfitting problems reported. This is further proven and shown in Figure 11, Figure 12, Figure 13, Figure 14, Figure 15, and Figure 16 which show the comparison of the training and testing validation losses for VGG19, VGG16, DenseNet, AlexNet, GoogleNet, and LeNet, respectively. The comparison shows that both training and testing validation losses approached zero as expected.

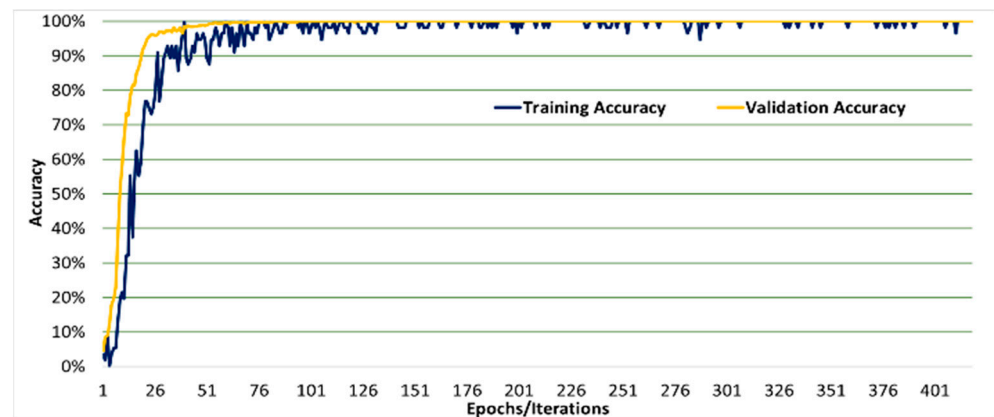


Figure 5. VGG19 accuracy learning curves for training and validation.

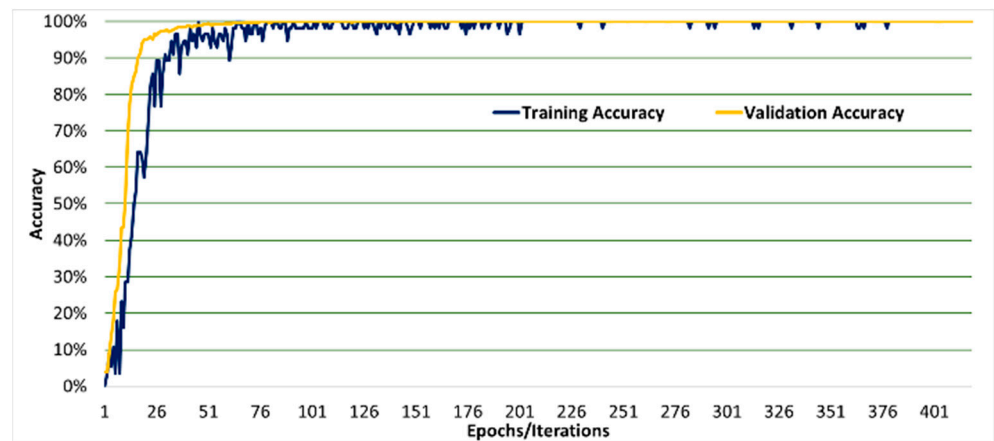


Figure 6. VGG16 accuracy learning curves for training and validation.

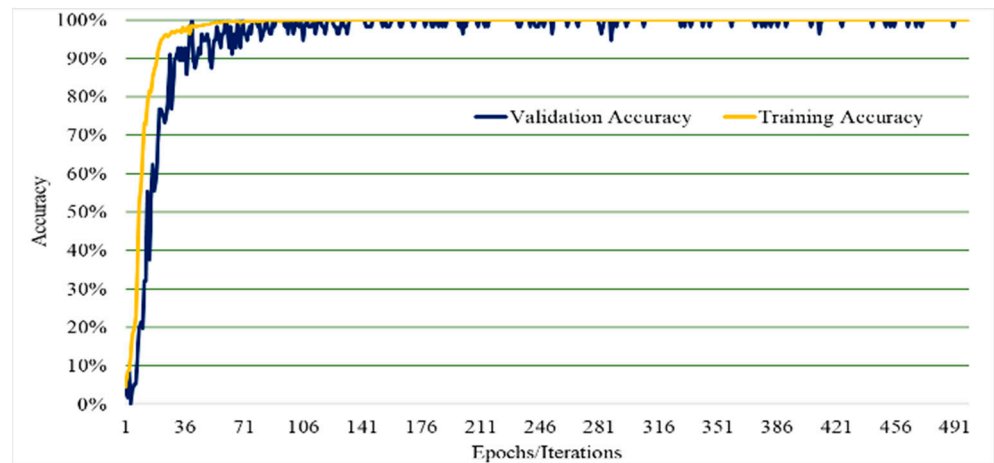


Figure 7. DenseNet accuracy learning curves for training and validation.

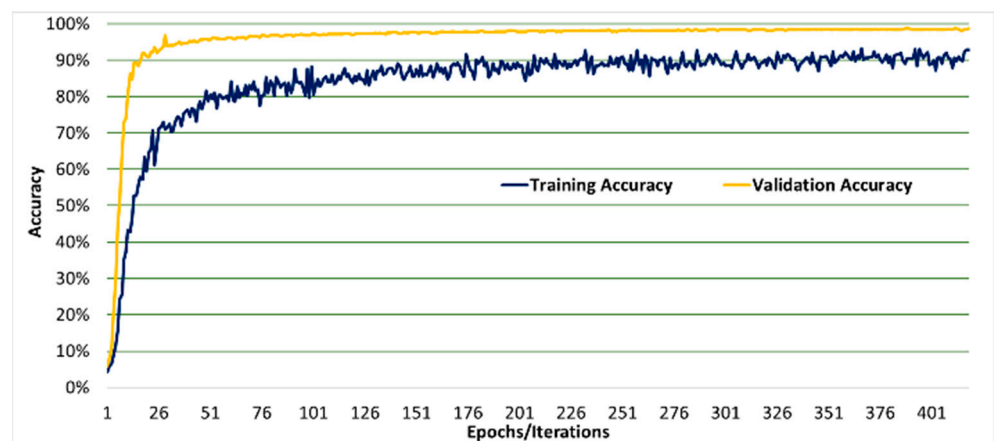


Figure 8. AlexNet accuracy learning curves for training and validation.

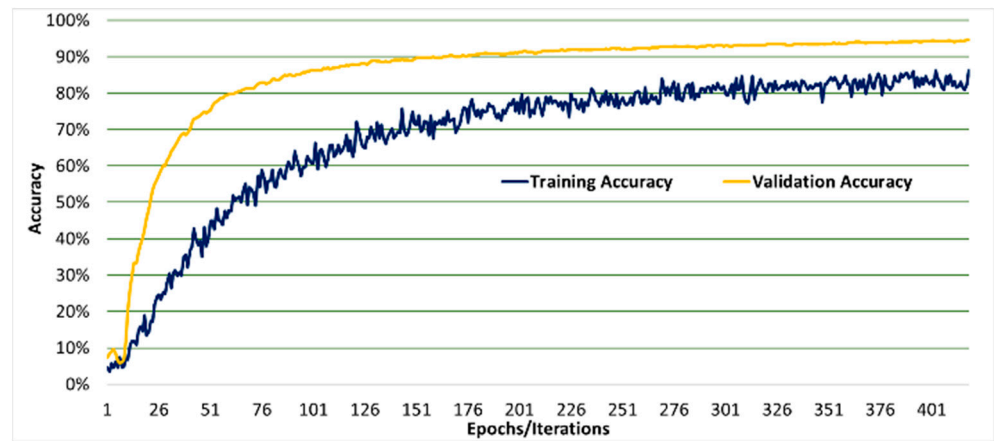


Figure 9. GoogleNet accuracy learning curves for training and validation.

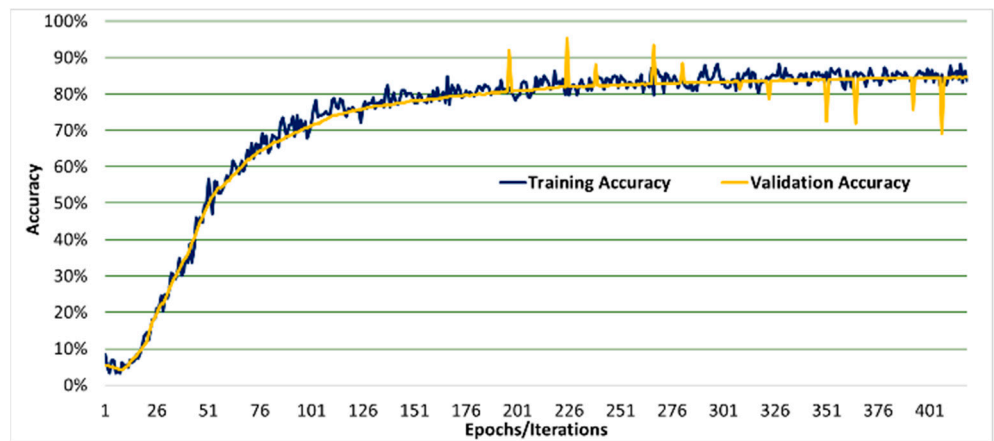


Figure 10. LeNet accuracy learning curves for training and validation.

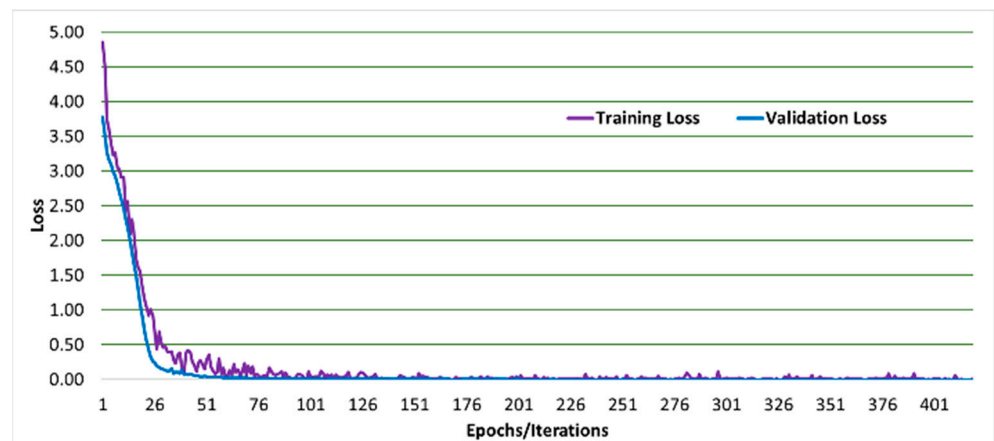


Figure 11. VGG19 loss learning curves for training and validation.

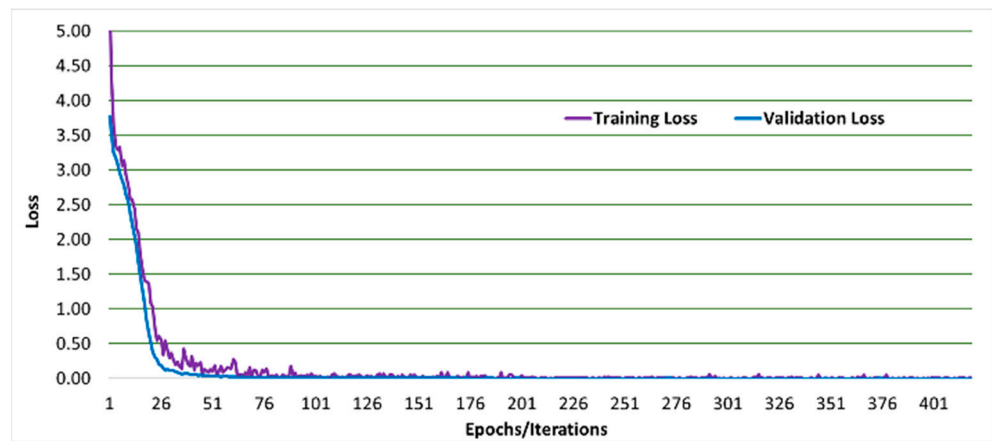


Figure 12. VGG16 loss learning curves for training and validation.

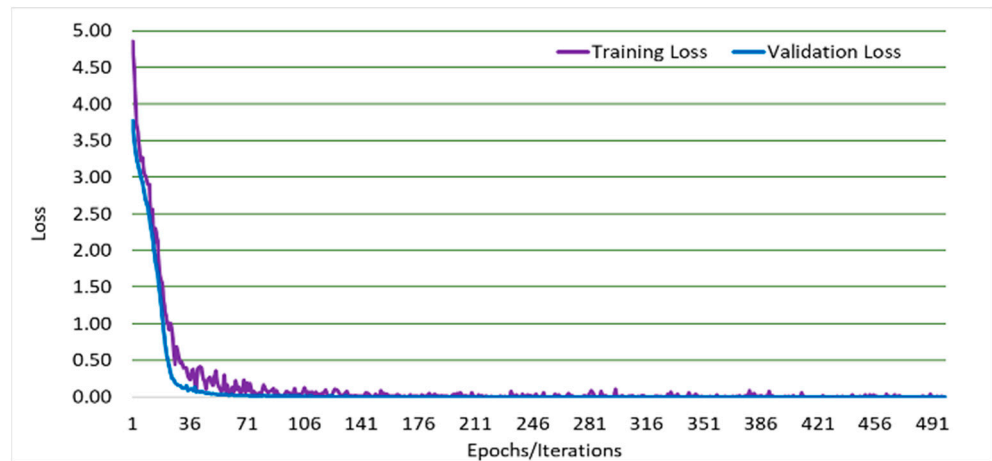


Figure 13. DenseNet loss learning curves for training and validation.

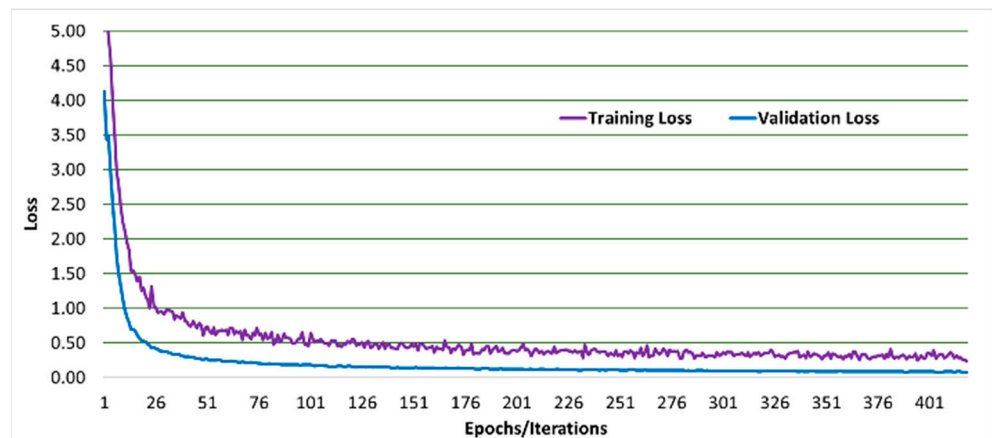


Figure 14. AlexNet loss learning curves for training and validation.

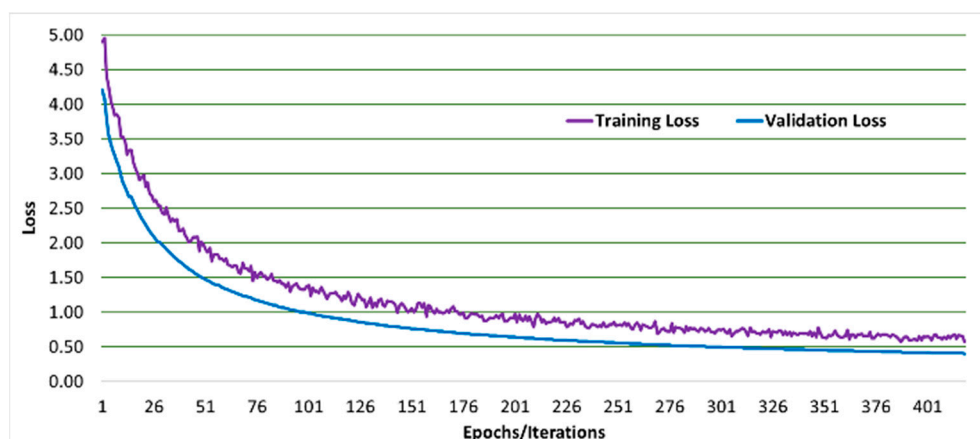


Figure 15. GoogleNet loss learning curves for training and validation.

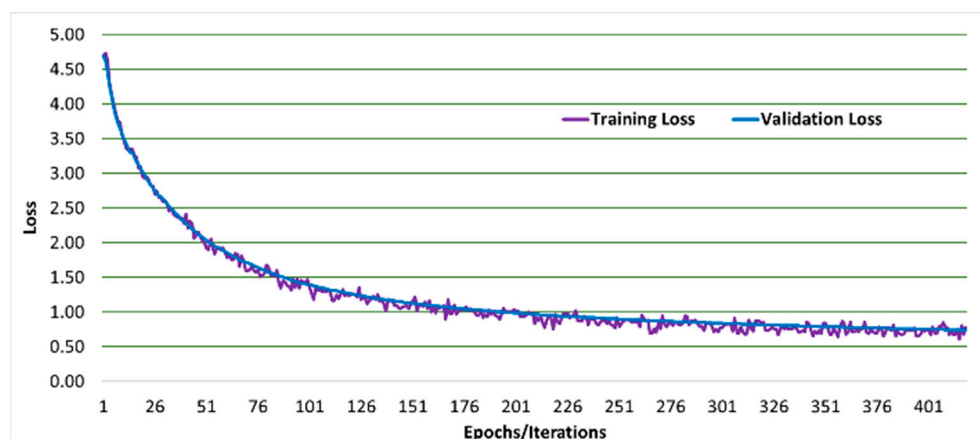


Figure 16. LeNet loss learning curves for training and validation.

The experiments were then repeated on the same optimized deep learning algorithms using the proposed non-freeze weights approach with the Arabic braille language dataset, as shown in Table 2. The accuracy increased significantly, with the best accuracy achieved by the VGG16 with an average accuracy of 99.68%, a precision of 98.36%, a recall of 97.96%, and an F1-measure of 98.16%. The lowest accuracy was still reported for GoogleNet with an average accuracy of 98.70%. Note that with non-freeze weights, the accuracy increased by 6.55% compared with the highest reported accuracy in Table 1. It should be clear here that the experiment was performed on the Arabic braille language dataset without augmentation.

Table 2. Experimental results of the proposed non-freeze weight-based CNN models for the Arabic braille language data.

	Accuracy	Precision	Recall	F1 Measure
VGG19	99.28%	0.991	0.992	0.992
VGG16	99.68%	0.993	0.991	0.994
DenseNet	96.51%	0.960	0.964	0.965
AlexNet	99.13%	0.989	0.988	0.989
GoogleNet	98.70%	0.984	0.981	0.981
LeNet	86.23%	0.858	0.860	0.861

The experiment is then repeated using the optimized deep learning algorithms using the proposed non-freeze weight approach but this time on the combined Arabic braille language dataset with the augmented dataset. The results are shown in Table 3. The results

indicate yet another increase in accuracy due to expanding the dataset size. The increase of 0.3% is actually significant as compared to results in Table 2 and dramatically significant as compared to results in Table 1 where the difference is 1.35%. The increase in accuracy is extremely important because this a proposed system that will serve for assistive learning for the visually impaired and they have no way of comparing the audio translation with the original unless they go to the traditional time-consuming touch-and-feel approach. Table 3 shows that the highest reported accuracy was again achieved using VGG16 with an average accuracy of 99.98%, precision of 99.4%, recall of 99.5%, and F1-measure of 99.7%. The lowest accuracy is again reported using th GoogleNet with an average value of 88.5%.

Table 3. Experimental results of the proposed non-freeze weight-based CNN models for the Arabic braille language augmented data.

	Accuracy	Precision	Recall	F1 Measure
VGG19	99.58%	0.995	0.994	0.995
VGG16	99.98%	0.994	0.995	0.997
DenseNet	98.62%	0.981	0.982	0.980
AlexNet	99.45%	0.993	0.991	0.989
GoogleNet	88.50%	0.883	0.879	0.881
LeNet	85.51%	0.850	0.851	0.851

The confusion matrix-based comparison obtained for the various experiments performed above with the best-performing VGG16 model is shown in Figures 17–19. These are the confusion matrices for the experiments performed on the Arabic braille language dataset. Figure 17 shows the confusion matrix for the basic VGG16 applied to the Arabic braille language dataset. It is noticed that even with the basic VGG16, the accuracy is high but it can be optimized to achieve better results because the application we are targeting is for the specific purpose of assistive learning technology for the visually impaired. Thus, an optimal solution can only be achieved as we approach approximately 100% on various complex datasets of the Arabic braille language dataset. Figure 18 shows the confusion matrix using the Optimized VGG16 model with the proposed transfer learning approach. It is noticed that the confusion matrix showed better results but still can stand for improvement for the optimal solution. Therefore, Figure 19 shows the confusion matrix using the optimized VGG16 along with the proposed transfer learning approach, which resulted in a further increase of accuracy.

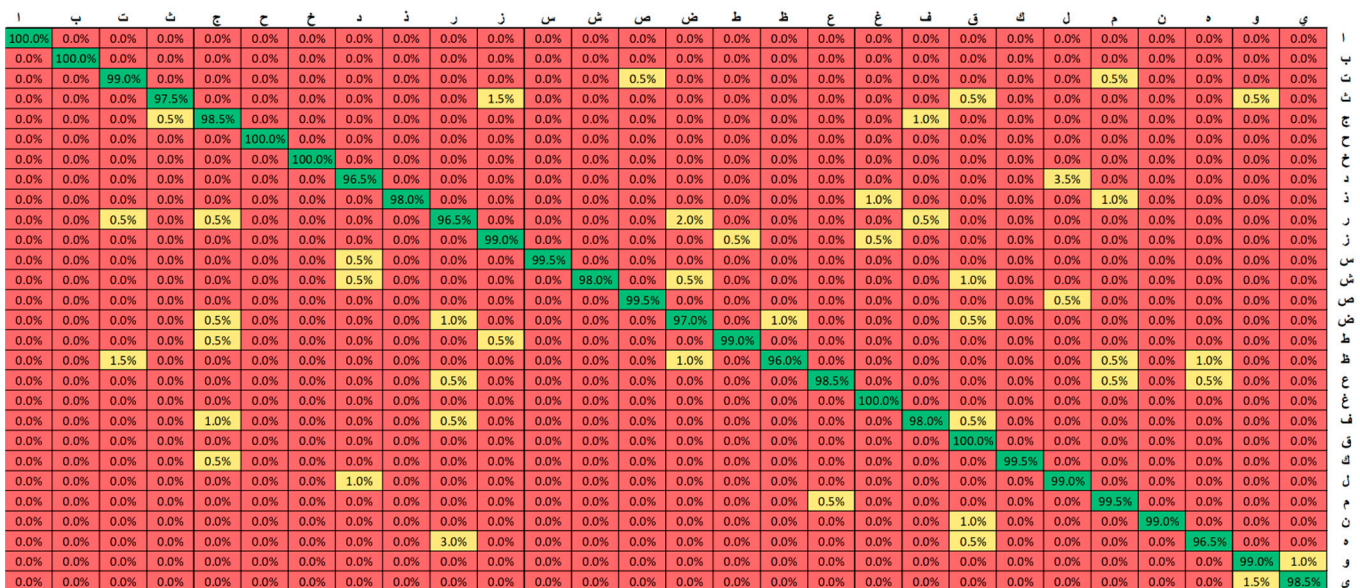


Figure 17. Confusion matrix for 28 Arabic braille letters classification using VGG16 basic transfer learning.

Table 4. Experimental results of the proposed non-freeze weight-based CNN models for the English braille language 26 letters augmented data.

	Accuracy	Precision	Recall	F1 Measure
VGG19	99.60%	0.991	0.994	0.994
VGG16	99.92%	0.995	0.994	0.995
DenseNet	97.46%	0.969	0.971	0.968
AlexNet	98.37%	0.970	0.970	0.97
GoogleNet	98.21%	0.976	0.978	0.977
LeNet	86.79%	0.864	0.863	0.863

5. Conclusions

Individuals with disabilities should continue to receive the utmost support since with the right education and tools they have proved themselves to be valuable members of the community. They contributed in many fields and history has recorded many famous individuals with disabilities and persons with visual impairment. They became famous because they achieved things that persons without disabilities and persons with full eyesight have not been able to achieve. Therefore, society must continue to support persons with disabilities to achieve their full potential. With the advancements in technology, many devices can be developed to assist persons with disabilities in the education field to enhance their education, learning, and knowledge. These technology-enhanced devices can assist them to learn or speed up their learning process. In this paper, we proposed an AI-based device that can automatically translate Arabic and English braille to the corresponding audio. This device can serve either Arabic-speaking individuals, English-speaking individuals, or bilingual individuals. There are many benefits to this device, including but not limited to teaching visually impaired individuals the braille language, teaching the relatives of the visually impaired individual the braille language, or assisting visually impaired individuals who already know braille to read braille documents/books at much faster speeds. The proposed system optimized deep learning models along with transfer learning. The main idea of the proposed system is to optimize the deep learning algorithms and then freeze the first portion of layers and unfreeze the second portion of the layers to allow the systems to retrain and update the weights accordingly. This resulted in an enhanced accuracy for both the Arabic language braille and English language braille. In addition, increasing dataset size and complexity allows for better performance. Therefore, augmentation was performed for both the Arabic and English braille language datasets to increase the dataset sizes. The increased sizes of datasets using the proposed method resulted in even higher optimal accuracies.

Future work in this field will include field testing the device to receive actual feedback from individuals with visual impairments. The system will continue to be enhanced based on the feedback from individuals with visual impairments and their relatives.

Author Contributions: G.L. conducted this research; G.B.B. worked on the methodology and analysis; S.E.A. and G.A. did the initial writing; R.A. reviewed the paper and fixed grammar; K.A. helped with the literature review. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Prince Mohammad bin Fahd Futuristic Studies Research Grant 2022. This work was also supported in part by the Commonwealth Cyber Initiative, an investment in the advancement of cyber R&D, innovation, and workforce development. For more information about CCI, visit <https://cyberinitiative.org>.

Data Availability Statement: The data used in this research was newly created which can be acquired on request by sending email to glatif@pmu.edu.sa.

Acknowledgments: All authors acknowledge the support from the Prince Mohammad Bin Fahd University for providing the computational resources to conduct this research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shokat, S.; Riaz, R.; Rizvi, S.S.; Abbasi, A.M.; Abbasi, A.A.; Kwon, S.J. Deep Learning Scheme for Character Prediction with Position-Free Touch Screen-Based Braille Input Method. *Hum.-Cent. Comput. Inf. Sci.* **2020**, *10*, 41. [[CrossRef](#)]
2. Latif, G.; Mohammad, N.; AlKhalaf, R.; AlKhalaf, R.; Alghazo, J.; Khan, M. An Automatic Arabic Sign Language Recognition System Based on Deep CNN: An Assistive System for the Deaf and Hard of Hearing. *Int. J. Comput. Digit. Syst.* **2020**, *9*, 715–724. [[CrossRef](#)]
3. Khan, S.; Rahmani, H.; Shah, A.; Bennamoun, M. *A Guide to Convolutional Neural Networks for Computer Vision*; SpringerLink: Berlin/Heidelberg, Germany, 2018.
4. Alufaisan, S.; Albur, W.; Alsedrah, S.; Latif, G. Arabic Braille Numeral Recognition Using Convolutional Neural Networks. *Springer eBooks* **2021**, *9*, 87–101.
5. Tiendee, S.; Lerdsudwichai, C.; Thainimit, S.; Sinthanayothin, C. The Method of Braille Embossed Dots Segmentation for Braille Document Images Produced on Reusable Paper. *Int. J. Adv. Comput. Sci. Appl.* **2022**, *13*, 163–170. [[CrossRef](#)]
6. Shokat, S.; Riaz, R.; Rizvi, S.S.; Khan, I.; Paul, A. Detection of Touchscreen-Based Urdu Braille Characters Using Machine Learning Techniques. *Mob. Inf. Syst.* **2021**, *2021*, 1–16. [[CrossRef](#)]
7. Shokat, S.; Riaz, R.; Rizvi, S.S.; Khan, I.; Paul, A. Characterization of English Braille Patterns Using Automated Tools and RICA Based Feature Extraction Methods. *Sensors* **2022**, *22*, 1836. [[CrossRef](#)]
8. Perera, T.D.S.H.; Wanniarachchi, W.K.I.L. Optical Braille Recognition Based on Histogram of Oriented Gradient Features and Support-Vector Machine. *Int. J. Eng. Sci. Comput.* **2018**, *8*, 19192–19195.
9. Asebriy, Z.; Raghay, S.; Bencharef, O. An Assistive Technology for Braille Users to Support Mathematical Learning: A Semantic Retrieval System. *Symmetry* **2018**, *10*, 547. [[CrossRef](#)]
10. Sufiun, A.; Jabiullah, M.I. A Novel Approach of CNN Patterns Extraction for Bangla Handwriting to Bangla Braille Notation. *Int. J. Eng. Adv. Res.* **2021**, *3*, 1–15.
11. Jha, V.; Parvathi, K. Braille Transliteration of hindi handwritten texts using machine learning for character recognition. *Int. J. Sci. Technol. Res.* **2019**, *8*, 1188–1193.
12. Prakash, S.; Thomas, S.; Gopalan, S.M. An Effective Approach of English Braille to Text Conversion for Visually Impaired Using Machine Learning Technique. *EasyChair Prepr.* **2023**, *9908*, 1–9.
13. Souza, M.D.; Preetham, S.; Varun, S.M.; Vardhan, N.; Venkatraman, G. Braille Character Recognition Using Deep Learning Strategy Image Processing and Computer Vision. *Int. Res. J. Mod. Eng. Technol. Sci.* **2023**, *5*, 6385–6389.
14. Chellaswamy, C.; Geetha, T.S.; Hariharan, K.; Archana, K.; Babitharani, S. Deep Learning-Based Braille Technology for Visual and Hearing Impaired People. In Proceedings of the 2023 International Conference on Smart Systems for Applications in Electrical Sciences, Tumakuru, India, 7–8 July 2023; pp. 1–8.
15. Fogarassy-Neszly, P.; Pribeanu, C. Multilingual text-to-speech software component for dynamic language identification and voice switching. *Stud. Inform. Control.* **2016**, *25*, 335–342. [[CrossRef](#)]
16. Bhatia, S.; Devi, A.; Alsuwailam, R.I.; Mashat, A. Convolutional Neural Network Based Real Time Arabic Speech Recognition to Arabic Braille for Hearing and Visually Impaired. *Front. Public Health* **2022**, *10*, 898355. [[CrossRef](#)] [[PubMed](#)]
17. Latif, G.; Alghmgham, D.A.; Maheswar, R.; Alghazo, J.; Sibai, F.; Aly, M.H. Deep Learning in Transportation: Optimized Driven Deep Residual Networks for Arabic Traffic Sign Recognition. *Alex. Eng. J.* **2023**, *80*, 134–143. [[CrossRef](#)]
18. Mohammed, A.S.; Hasanaath, A.A.; Latif, G.; Bashar, A. Knee Osteoarthritis Detection and Severity Classification Using Residual Neural Networks on Preprocessed X-ray Images. *Diagnostics* **2023**, *13*, 1380. [[CrossRef](#)]
19. Saleem, M.A.; Senan, N.; Wahid, F.; Aamir, M.; Samad, A.; Khan, M. Comparative Analysis of Recent Architecture of Convolutional Neural Network. *Math. Probl. Eng.* **2022**, *2022*, 1–9. [[CrossRef](#)]
20. Tao, Y.; Xu, M.; Lu, Z.; Zhong, Y. DenseNet-Based Depth-Width Double Reinforced Deep Learning Neural Network for High-Resolution Remote Sensing Image Per-Pixel Classification. *Remote Sens.* **2018**, *10*, 779. [[CrossRef](#)]
21. Latif, G.; Morsy, H.A.; Hassan, A.; Alghazo, J. Novel Coronavirus and Common Pneumonia Detection from CT Scans Using Deep Learning-Based Extracted Features. *Viruses* **2022**, *14*, 1667. [[CrossRef](#)]
22. Qu, Y.; Tang, W.; Feng, B. Paper Defects Classification Based on VGG16 and Transfer Learning. *J. Korea TAPPI* **2021**, *53*, 5–14. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to person or property resulting from any ideas, methods, instructions or products referred to in the content.